Assessment of Visibility Restriction Mechanisms for RFID Data Discovery Services

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

[†]Department of Computer Science and Engineering
Instituto Superior Técnico, Technical University of Lisbon, Portugal
Email: miguel.pardal@ist.utl.pt, jose.marques@link.pt

[‡]Auto-ID Labs, Institute for Manufacturing,
University of Cambridge, UK
Email: mark.harrison@cantab.net

***Abstract*—RFID is a technology that enables the automated capture of observations of uniquely identified physical objects as they move through supply chains. Discovery Services provide links to repositories that have traceability information about specific physical objects. Each supply chain party publishes records to a Discovery Service to create such links and also specifies access control policies to restrict who has visibility of link information, since it is commercially sensitive and could reveal inventory levels, flow patterns, trading relationships, etc.**

The requirement of being able to share information on a need-to-know basis, e.g. within the specific chain of custody of an individual object, poses a particular challenge for authorization and access control, because in many supply chain situations the information owner might not have sufficient knowledge about all the companies who should be authorized to view the information, because the path taken by an individual physical object only emerges over time, rather than being fully pre-determined at the time of manufacture. This led us to consider novel approaches to delegate trust and to control access to information.

This paper presents an assessment of visibility restriction mechanisms for Discovery Services capable of handling emergent object paths. We compare three approaches: enumerated access control (EAC), chain-of-communication tokens (CCT), and chain-of-trust assertions (CTA). A cost model was developed to estimate the additional cost of restricting visibility in a baseline traceability system and the estimates were used to compare the approaches and to discuss the trade-offs.

I. Introduction

RFID tags [1] and also some optical barcode technologies such as DataMatrix and DataBar allow each physical object to be uniquely identified e.g. via an Electronic Product Code (EPC). The unique identifier is required to associate the highly granular traceability data with each corresponding object. This information trail needs to be retrieved to answer traceability queries [2], such as:

- What is the current location of the object? (Track)
- What is the location history of the object? (Trace)

The traceability information is not centralized but instead is stored in a number of distributed data repositories owned and managed by the various companies participating in a given supply chain.

The EPC Network architecture provides a suite of global open standards [3] for the capture and sharing of traceability information. The EPC Information Services (EPCIS) standard [4] enables physical event data to be exchanged between companies using a standardized information model and query interface, irrespective of differences in the implementation of the underlying database. At the top of the EPC Network architecture there is a placeholder for Discovery Services (DS) that enable business applications to locate multiple sources of information to answer traceability queries. DS provide links to EPCIS repositories that have event data about a specific physical object. These links are pro-actively created by each company that wants to make the association between the unique identifier of a physical object and the Uniform Resource Locator (URL) address of a EPCIS repository. Because the traceability information is commercially sensitive, each company also publishes access control policies to restrict who has visibility of link information. A company can benefit from sharing information with trusted business partners but can also be harmed if information is exposed to competitors. This means that information providers and consumers have to be authenticated in a trusted way and that the information access must be authorized effectively. An industry survey [5] further emphasizes the need for information protection.

Access control for traceability information is different from traditional authorization [6] because in many supply chain scenarios the information owner will not have prior knowledge about which companies should be authorized to view the data about a specific individual object. The path taken by an individual object - the 'object path' - emerges over time as customers and intermediate distributors place orders for goods and ship those goods downstream through the supply chain. The dynamic object path characteristic led us to consider the following *visibility restriction* approaches:

- Enumerated access control (EAC);
- Chain-of-communication tokens (CCT);
- Chain-of-trust assertions (CTA).

We developed an analytical cost estimation model to compare these approaches. The system's actions are represented as operations that store or retrieve data, process data, or transfer data over the network. The costs of these operations are summed by category to produce the estimates. We modeled data capture and query operations first and then the visibility restriction operations.

In the next section we briefly discuss related work on DS systems and other traceability information systems. Then we proceed to describe in detail the visibility restriction mechanisms, followed by the cost model and baseline estimates. We then present and discuss the results and compare the approaches. The paper ends with the conclusions and plans for future work.

II. Related work

A DS is a facilitator service, enabling information consumers to discover links to information providers. However there are other proposals to build traceability systems.

Evdokimov et al. [7] produced a qualitative comparison of traceability information systems examining the characteristics of functional requirements using a framework based on an ISO standard for software quality.

Pardal and Alves Marques [8] surveyed over twenty traceability information system proposals, and summarized them into four categories, according to two criteria: distribution (centralized versus decentralized solutions) and data integration (data copying versus data referencing). The four categories were compared using a simple quantitative cost model that estimated total system cost. The performance estimates for typical Pharmaceutical and Automobile supply chains showed that the Metadata Integration (MDI) architecture had the second-best overall performance and provided an additional indirection level that could be used to address other solution concerns, such as visibility restrictions.

The combination of DS with EPCIS is a concrete implementation of the MDI architecture, and there are freely available implementations of both systems. Fosstrak [9] is an open-source EPCIS implementation and BRIDGE Directory of Resources [10] [11] is a DS implementation. In the MDI architecture, a DS provider can play the role of a trusted third party [12] and assist in information sharing operations.

III. Visibility restriction

Visibility restriction mechanisms define how restrictions are stated and enforced. They should be *expressive* to allow compact sharing statements, for example, they should assume default values to avoid repetitive expressions (e.g. the information owner should be provided access to its data by default). They should also be *correct* i.e. formally verifiable at the conceptual level, and auditable by external parties at the implementation level.

In the work presented in this paper we make the visibility restriction approaches *quantifiable* to allow objective comparisons relative to a baseline. Both the expressiveness and correctness aspects are out of scope. Other relevant concerns such as performance and scale [5] are also out of scope.

A. Mechanisms

We compare three distinct approaches: enumerated access control (EAC), chain-of-communication tokens (CCT), and chain-of-trust assertions (CTA). Each approach can be seen, for comparison purposes, as a different formulation of the same canonical data structure: a four-dimensional matrix defined by tuples of 'information owner', 'action', 'trading partner' and 'physical object'. Each cell represents a data access right: the 'owner' grants 'action' rights to the 'partner' over data about the 'object'.

1) Enumerated access control: EAC represents the more traditional access control mechanism based on access control lists (ACL) [6] or similar data structures, that keep the access rights indexed by the object identifier.

In this approach, there is an ACL that holds identifiers for trading partners that have access to information owned by a company about a given object. The ACL is maintained at the DS, but a local copy is maintained in each EPCIS to also protect its records. To share information, the information owner adds trading partners to the ACL. For audit purposes, the changes to the list should be logged to allow reconstruction of list state at any point in time.

2) Chain-of-communication tokens: CCT represents a capabilities mechanism [6] because the access rights are kept within the object reference. When a reference - token - is shared, the access rights are also shared. Access rights can 'follow the chain' if the shipping company sends the token to the destination along with the physical object.

A token is a binary data structure consisting of two parts: identifier and secret. The id part is used to identify the token. The secret part is used to authorize access to data. The token is propagated along the chain by electronic communication e.g. within an Advance Shipping Notice (ASN) message or embedded in a special RFID tag. The token must be presented by the querying party. The DS issues the token and also keeps a copy and uses it to protect DS records. The token is also used to protect event data in EPCIS. To create new visibility scopes, new tokens are created and used at each node in the object path. However, a single token can be used along the chain to protect all of the traceability records. This may be interesting in some business scenarios. For audit purposes, the presented token values at the times of publishing and querying should be logged.

3) Chain-of-trust assertions: CTA represents a potentially more expressive mechanism that expresses the access rights using logical statements issued by the multiple companies. When information is requested, the logical formulae are evaluated to make an access decision. The semantics of the statements can express conditions like *reciprocal trust* meaning that a company is willing to share information with a partner if (and only if) the partner is willing to do the same.

In this approach, a cell from the canonical matrix is stated in an assertion format: trust(owner, action, partner, object). These assertions are sent to the DS, signed by the author, and can be revoked later, if necessary. DS provides access to partners for which there is an explicit unbroken chain of trust assertions back to the owner of the information. A local copy of the assertions is kept in EPCIS to allow protection and local verification of the chain of trust. To share information, the information owner should add assertions for the desired partners. For audit purposes, all assertions should be logged along with the certificates required to verify them.

B. Sharing policies

Companies have to decide *when* to authorize information sharing. We consider two policies: ‘upfront’ and ‘on demand’.

1) Upfront sharing: In this case the trading partners are pro-active regarding the sharing operations because future queries are considered likely for all objects. For each individual object, they grant access to their immediate upstream and downstream partners.

A use case where ‘upfront’ sharing might make sense is in a *Pharmaceutical pedigree* application. In this case, it is likely that a legal requirement mandates that the product trace for every individual object should be provided to the customer. Therefore, queries will be issued for all objects that are subject to pedigree legislation.

2) On demand sharing: In this case the sharing effort at capture time is reduced because future queries for all objects are considered less likely. The flipside is that additional requests and decisions to share information are required and have to be mediated by DS, creating additional burden for it.

A use case where ‘on demand’ sharing makes sense is in an *Automobile recall* application for defective parts. A recall is issued only for a small proportion of objects. Therefore, queries will be issued only for recalled objects, rather than for the majority of objects.

C. Infrastructure

All described visibility restriction approaches assume an underlying security infrastructure that provides secure communications and identity verification.

The infrastructure relies on a Public Key Infrastructure (PKI) [13] to store digital certificates.

The authentication and authorization are *externalized*, i.e. kept outside of the main application code, to allow consistent use of policies across all applications, and providing consistent logging for auditing processes. The policies can express the visibility restriction approaches described earlier and make them interchangeable so they can co-exist in a working system.

The externalized security architecture uses standard technologies, namely SAML (Security Assertion Markup Language) [14] and eXtensible Access Control Modeling Language (XACML) [15].

Using SAML means that there are standard formats and exchange protocols for identity claims. Claims can be used to transfer attributes other than identity.

Using XACML means that there are standard formats and processing models for security policies along with interfaces for administration, enforcement and evaluation of policies.

Hebig et al. [16] describe an integration of the mentioned security technologies in a working infrastructure.

IV. Assessment Framework

Our overall assessment approach is depicted in Figure 1:

- Model a supply chain;
- Model the candidate solution’s workflows, resulting from functional, partition and security aspects;
- Use a cost model to compute estimates;
- Validate and calibrate the model using measurements.

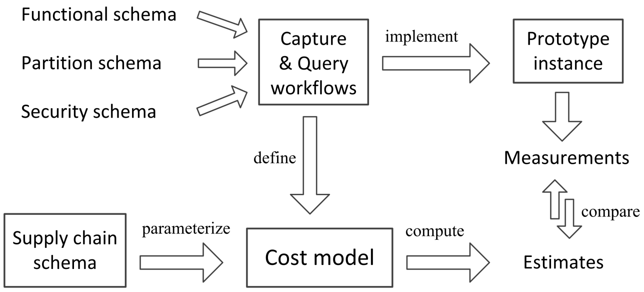

Fig. 1. Assessment framework.

The supply chain nodes can be generated from statistics or can be based on an actual scenario. The nodes are represented as a graph that can be defined *bottom-up* by the union of object paths [8] or *top-down* using definitions in the Supply Chain Modeling Language (SCML) [17].

A supply chain scenario is defined by a set of trading partners, a set of physical objects, and a corresponding set of object paths, as depicted in Figure 2. For our purposes, we consider that each trading partner has a single EPCIS instance and that there is a single DS shared by all the trading partners in the chain.

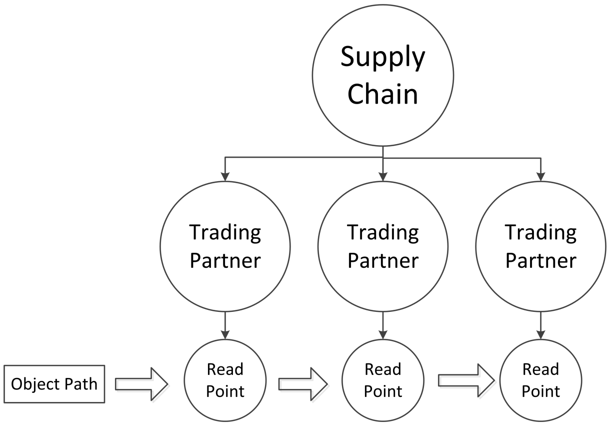

Fig. 2. Supply chain scenario.

The information system is specified in three aspects: the *functional scheme* of the components outlines what are the parts of the system and how they interact; the *partition scheme* handles how data is distributed within and across instances of the system; the *security scheme* encompasses the security infrastructure and the visibility restriction approaches described in this paper.

When the schemes are combined, we specify a system that can be modeled to compute cost estimates or can be implemented and instrumented to produce measurements. The estimates can then be compared to measurements, allowing the model to be validated and calibrated.

A. System modeling

A system is modeled for cost estimation by identifying its operations. Each operation has input and output and these data structures are modeled from actual implementations whenever possible. For EPCIS the modeling was based on Fosstrak [9] and for DS it was based on the BRIDGE design [18].

The basic cost formulae are based on the model by Murthy and Robson [19] and the following assumptions hold:

1) Bandwidth, processing speed, latency are the same for every node.
2) Messages and received object records can be processed in main memory.
3) All data stores are append-only.
4) The time cost of accessing the data store to retrieve a record is independent of size.
5) The time cost of storing a record can be ignored, because it can be done asynchronously.

B. Cost computation

The cost model uses a *blackboard* data structure [20] represented in Figure 3. There are board contributors that observe the posts that are placed on the board, and that can add more posts. A post can represent a cost parcel. Each contributor only looks at each post once. The board is fully expanded when all contributors view all posts and add nothing new. At the end of the expansion the cost totals are computed from all posts.

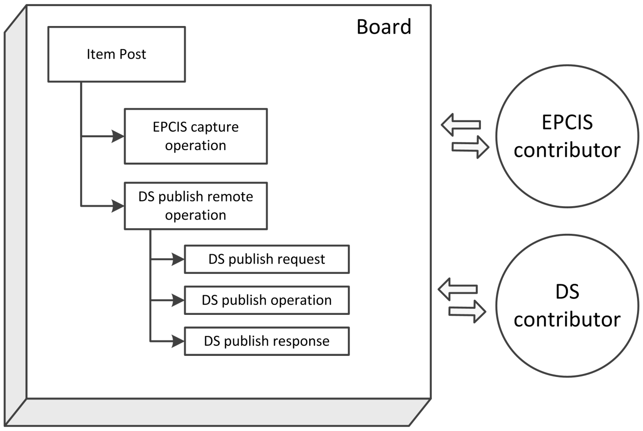

Fig. 3. Cost computation board.

There are three types of cost: storage (data size), processing (time) and networking (time). Not all posts have direct cost, some are merely placeholder posts that signal the need for additional expansion.

The cost calculation board follows a *cause-effect* logic: something happens - represented by a board post - and triggers other posts. This allows each board contributor to concern itself with only a subset of effects at a time. The main advantage of this data structure for cost calculation is that it is easy to add costs of cross-cutting concerns, like security, without having to model the sequential flow of actions. Also, board contributors can easily be enabled or disabled to test different conditions e.g. add/remove network latency.

The cost parcels are stored in a cost tree that separates storage, processing and network costs. For each kind of cost, there are several levels of detail, as shown in Figure 4.

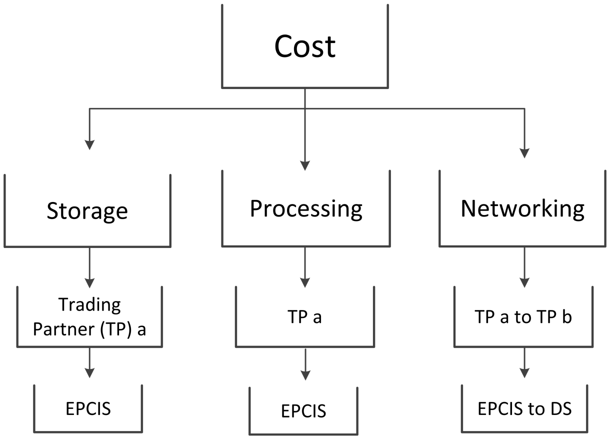

Fig. 4. Cost ‘buckets’. Each bucket keeps part of the total computed cost.

Finally, the cost results are exported in tabular data format that can be readily recognized and used by most analysis and plotting tools.

A cost computation example starts with a statistic description of a supply chain scenario. An average chain is generated with one object path. At this step a specific chain scenario could be provided, with as many companies and object paths as desired. The cost calculation board is initialized and a single post is placed there. In this example, it is an item post, representing that an object is passing on the supply chain. The EPCIS contributor adds one EPCIS capture post for each read point. The DS contributor adds one DS publish for each trading partner. Since DS publish is a remote operation, the Remote Procedure Call (RPC) contributor adds posts to send the request, execute the operation and send back the response. The process continues until there are no further posts. EPCIS captures have storage and processing costs, but since they are local to a trading partner they have no networking costs. DS publish has storage, processing and network communication/lookup costs. After the board is complete (see Figure 3), all the posts are analyzed and the total costs are computed.

C. Visibility restriction modeling

We defined a meta-model for visibility restriction that defines board posts corresponding to *initialize* (InitShare), *request* access (RequestShare), *share* (Share), and *enforce* access (EnforceShare) to physical object information. Each visibility restriction approach - EAC, CCT, and CTA - defines its own effects for these placeholder posts.

EAC is initialized with the creation of remote and local ACLs for each object. A request for new access is mediated by the DS and decided by the ACL owner. The sharing is done by adding a new company to the ACL. The enforcement is done by checking if the querying party is in the ACL.

CCT is initialized with the creation of token for each object. A request for new access is mediated by DS and decided by a token holder. The sharing is achieved by sending the token. The enforcement checks if the token is valid.

CTA does not require initialization. A request for new access is mediated by DS and decided by the data owner. The sharing is accomplished by publishing new assertions. The enforcement checks it the existing assertions logically grant access to the querying party.

The main differences between the approaches are that in CCT the request for new access can bypass DS (the token can be shared directly) and CTA does not require initialization.

V. Evaluation

A. Baseline

The baseline represents the base system cost without information sharing costs. We have considered the following three chains to gain a broader perspective on the relative values of the results:

- Short chain - 3 companies;
- Medium chain - 6 companies;
- Long chain - 12 companies.

Each chain is linear i.e. there are no path branches/forks.

1) Storage: Figure 5 shows the storage costs for each chain. The only operation with storage costs is the capture, storing EPCIS events and DS records. The cost of audit logs is not being considered. We can see that the storage cost grows linearly with the chain length, as expected.

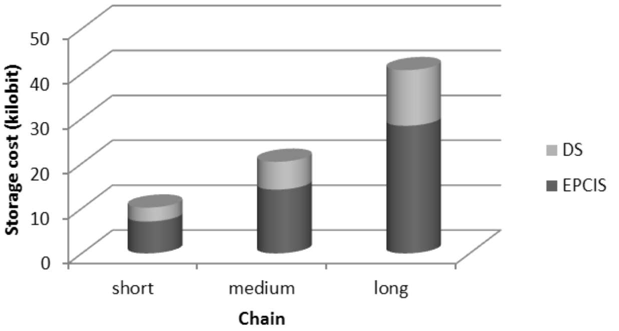

Fig. 5. Storage cost baseline.

A typical value of 10ms was assumed for data seek. This value is bounded by typical average access times to secondary memory devices, like hard disks.

2) Processing: Figure 6 shows the processing time for each operation: data capture, track query, and trace query. The data capture cost is much lower because data writing is done asynchronously i.e. the operation does not have to wait for write completion. The queries, on the other hand, have to wait for the data seek completion. Notice also that the track query cost is independent of chain length because only the EPCIS with the most recent record is contacted after the DS query.

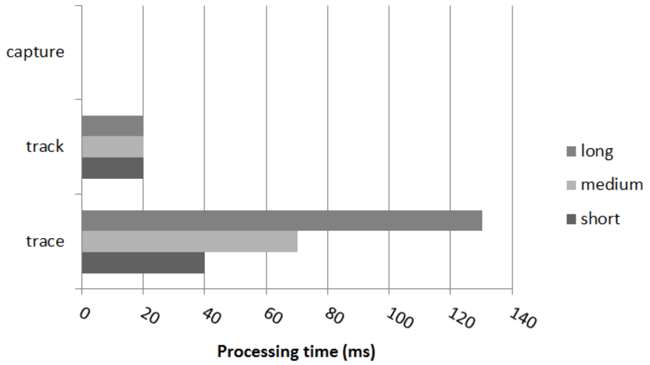

Fig. 6. Processing cost baseline.

3) Networking: Figure 7 shows the networking time cost. A typical value of 100ms latency for round-trip was assumed, after averaging the 'ping' response times from servers of major universities across the world.

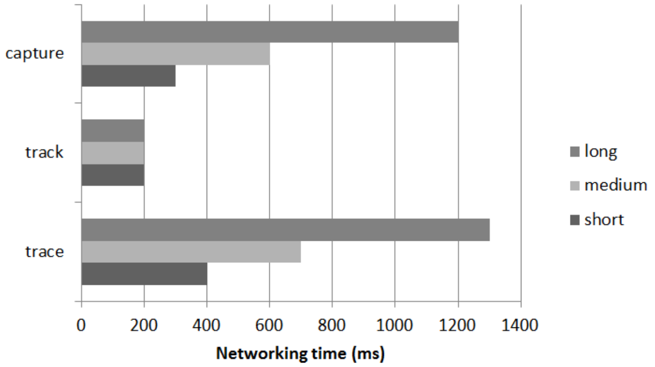

Fig. 7. Networking cost baseline.

The networking time cost increases linearly with the object path's length for both capture and trace queries, but it is the same for the track query because only two remote calls are needed in all cases (one DS query and one EPCIS query). The time cost of networking is dominated by the latency, because messages are small in size.

4) Overheads: The baseline considers that data structures are binary using basic integer and string types. However, there are several advantages in using SOAP-based Web Services [21] as mandated by the EPC standards. The data is encoded in eXtensible Markup Language (XML) and this allows greater interoperability. We modeled the XML overhead using values measured by Juric et al. [22] that states that a SOAP message is, on average, 4.3 times larger than a binary message, and that response time is, on average, 9 times longer.

The security infrastructure adds the overhead of a security channels using Secure Sockets Layer (SSL). According to

Juric et al. [22], SSL makes messages 1.08 times larger, and response time 1.40 times longer.

We do not present plots of the processing and networking cost with XML and SSL because the overheads are barely visible next to the latency values.

B. Sharing

For comparing EAC, CCT, and CTA the results for the short, medium and long chains were averaged. We consider only 'trace' queries in the comparison. We also consider that the query is always issued by a company in the object path to enable potential benefits of the 'upfront' sharing policy. We decided, by convention, to always pick the first company in the object's path as the querying party.

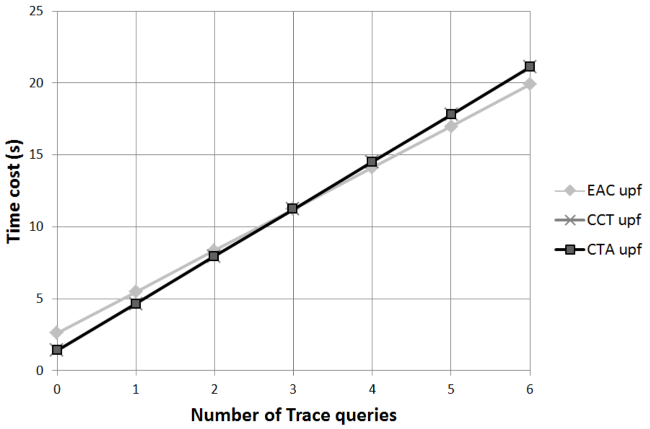

Fig. 8. Visibility approach comparison for 'upfront sharing'.

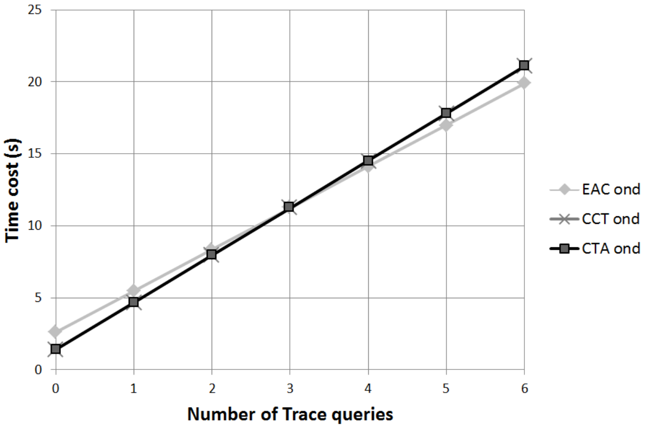

Fig. 9. Visibility approach comparison for 'on demand sharing'.

1) Upfront sharing: Figure 8 presents the comparison results for 'upfront' (abbreviated 'upf') EAC, CCT, and CTA. The total cost is a result of the sum of the cost of data capture with the cost of one or more trace queries (x-axis). CTA and CCT have the best performance up to 3 queries per object, then EAC is better, but the difference is not significant.

2) On demand sharing: Figure 9 presents the comparison results for 'on demand' ('ond') EAC, CCT, and CTA. Once again, the differences are not significant. CTA and CCT have the best performance up to 3 queries per object, then EAC is better. More surprisingly, the results are nearly identical to 'upfront' sharing.

3) On demand versus upfront: For Figure 10 we picked one of the best 'upf' and one of the best 'ond' - CCT in both cases. Again the differences are very small because both policies end up having the same number of remote operations, and the latency dominates the cost.

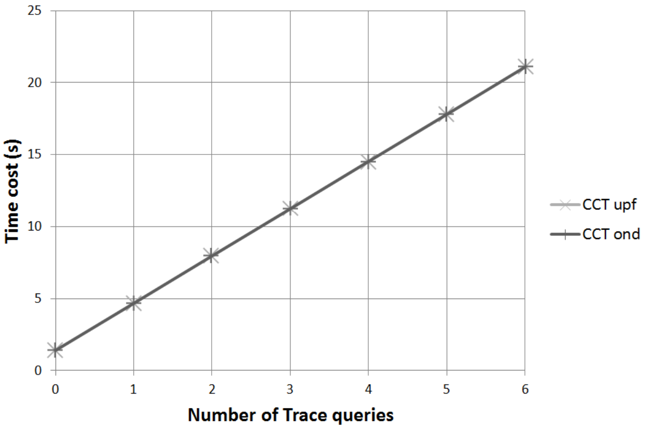

Fig. 10. Visibility approach comparison between CCT 'upfront' and CCT 'on demand'.

To compare 'on demand' with 'upfront' we look for the break-even point where the number of queries makes one of the approaches preferable. Theoretically, close to zero in the x-axis is the area of interest for 'on demand' cases. For 'upfront' cases, the area of interest is one on the x-axis or a higher value, depending on how many trace history checks are expected to be performed by intermediate parties within the supply chain. However, given that the estimates are identical for 'on demand' and 'upfront', the conclusion is that 'on demand' is always preferable, even when future queries are certain.

VI. Conclusions

We presented approaches to express and enforce visibility restrictions required by trading partners to be willing to participate in traceability systems. We developed a cost model that produced estimates for different approaches, highlighting the differences between processing at the capture stage and at the query stage.

The time cost of processing and networking is dominated by the latencies. This means that the time cost total can be roughly approximated by summing just the costs of the network transfer delays and the database seek operations.

Using our model, we concluded that there is no significant difference in using 'upfront' versus 'on demand' information sharing policies because it is hard to guess the future query needs. This is aligned with the notion of emergent path discussed in the beginning of the paper.

The comparison of visibility restriction approaches is summarized in Table I.

	DS role	Sharing	Perf.	Express.
EAC	ACL master copy	Add/remove from ACL via DS	OK	Limited
CCT	Token issuer	Send token to partner directly or via DS	OK	Limited
CTA	Assertion repository	Assert/negate statement via DS	OK	Extensible

TABLE I
SUMMARY COMPARISON OF VISIBILITY RESTRICTION APPROACHES.

DS plays a central role in all approaches but in CCT the sharing of authorization token can be done directly, without DS intervention. The estimated performance (Perf.) for all approaches is very similar and does not have a significant impact on overall cost. The expressive potential (Express.) of the approaches is different: EAC and CCT have pre-defined semantics and are, in this sense, limited; CTA uses logic and the semantics can be extended with additional statements.

A. Future Work

The *expressiveness* of visibility restriction approaches will be further researched. The canonical access matrix is sparse and considerable efficiencies can be obtained by representing it in more compact ways using object groupings (batches) or company sets (groups). We will explore combinations and variations of the presented approaches.

In our model, the time cost of the operations is dominated by the latencies in the network and in the data store accesses. We have plans to implement a visibility restriction *prototype* to calibrate the cost estimates by comparing them to actual measurements. In particular we want to verify that the visibility restriction costs were not underestimated. We will use our cost computing model to assess real-world case studies. We hope that having an actual business context can further validate our assumptions regarding traceability query types and frequencies.

Further ahead, we will extend our assessment framework to measure the performance impact of adding *domain-specific rules* to the traceability operations. The rules will try to leverage recurring data access patterns due to physical world (time-space) and business (document and process) realities.

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] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. John Wiley & Sons, Ltd, 2003.

[2] R. Agrawal, A. Cheung, K. Kailing, and S. Schonauer, "Towards traceability across sovereign, distributed RFID databases," in *International Database Engineering and Applications Symposium (IDEAS)*, 2006.

[3] K. Traub, F. Armenio, H. Barthel, P. Dietrich, J. Duker, C. Floerkemeier, J. Garrett, M. Harrison, B. H. J. Mitsugi, J. Preishuber-Pfluegl, O. Ryaboy, S. Sarma, K. Suen, and J. Williams, *The EPCglobal Architecture Framework 1.4*, GS1 Std., December 2010. [Online]. Available: http://www.epcglobalinc.org/standards/architecture/

[4] EPCglobal, *EPC Information Services (EPCIS) 1.0.1 Specification*, GS1 Std., September 2007. [Online]. Available: http://www.epcglobalinc.org/standards/epcis

[5] BRIDGE, "Requirements document of serial level lookup service for various industries," University of Cambridge and AT4 wireless and BT Research and SAP Research and ETH Zurich and GS1 UK, Tech. Rep., August 2007.

[6] R. Sandhu and P. Samarati, "Access control: principle and practice," *Communications Magazine, IEEE*, vol. 32, no. 9, pp. 40 –48, September 1994.

[7] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of Discovery Service architectures for the Internet of Things," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2010, pp. 237–244.

[8] M. L. Pardal and J. A. Marques, "Cost model for RFID-based traceability information systems," in *IEEE International Conference on RFID Technology and Applications*, September 2011.

[9] 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. [Online]. Available: http://www.fosstrak.org/publ/FosstrakIEEESystems.pdf

[10] J. Cantero, M. Guijarro., G. Arrebola, E. Garcia, J. Banos, M. Harrison, and T. Kelepouris, "Traceability applications based on Discovery Services," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2008, pp. 1332–1337.

[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] T. Burbridge and M. Harrison, "Security considerations in the design and peering of RFID discovery services," in *International IEEE Conference on RFID*, Orlando, USA, 2009, pp. 249–256.

[13] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IEFT, Internet Engineering Task Force Std., May 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[14] 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. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/

[15] B. Parducci, H. Lockhart, and E. Rissanen, *eXtensible Access Control Markup Language (XACML) Version 3.0*, OASIS Std., August 2011. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html

[16] 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. Conf. Web Services ICWS 2009*, 2009, pp. 551–558.

[17] D. Chatfield, T. Harrison, and J. Hayya, "XML-based supply chain simulation modeling," in *Winter Simulation Conference*, vol. 2, December 2004, pp. 1485 – 1493 vol.2.

[18] BRIDGE, "High level design for Discovery Services," University of Cambridge and AT4 wireless and BT Research and SAP Research, Tech. Rep., August 2007.

[19] K. Murthy and C. Robson, "A model-based comparative study of traceability systems," in *Proceedings of the International Conference on Information Systems, Logistics and Supply Chain (ILS)*, May 2008, madison, Wisconsin.

[20] M. Beigl, , M. Beuster, D. Rohr, T. Riedel, C. Decker, and A. Krohn, "S2B2: Blackboard for transparent data and control access in heterogeneous sensing systems," in *4th International Conference on Networked Sensing Systems (INSS)*, June 2007, pp. 126–129.

[21] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Springer Verlag, 2004. [Online]. Available: http://www.inf.ethz.ch/personal/alonso/WebServicesBook

[22] M. B. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko, "Comparison of performance of Web Services, WS-Security, RMI, and RMISSL," *Journal of Systems and Software*, vol. 79, no. 5, pp. 689 – 700, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121205001329