TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Scalable and secure RFID data discovery

Miguel Filipe Leitão Pardal

Supervisor: Doctor José Manuel da Costa Alves Marques
Co-Supervisor: Doctor Sanjay Emani Sarma

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Jury final classification: Pass

Jury

Chairperson: Chairman of the IST Scientific Board
Members of the Committee:
Doctor Sanjay Emani Sarma
Doctor José Manuel da Costa Alves Marques
Doctor Luís Eduardo Teixeira Rodrigues
Doctor Miguel Nuno Dias Alves Pupo Correia
Doctor André Ventura da Cruz Marnoto Zúquete
Doctor António Manuel Raminhos Cordeiro Grilo

2014

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Scalable and secure RFID data discovery

Miguel Filipe Leitão Pardal

Supervisor: Doctor José Manuel da Costa Alves Marques
Co-Supervisor: Doctor Sanjay Emani Sarma

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

Jury final classification: Pass

Jury

Chairperson: Chairman of the IST Scientific Board
Members of the Committee:
Doctor Sanjay Emani Sarma, Professor
Massachusetts Institute of Technology, USA
Doctor José Manuel da Costa Alves Marques, Professor Catedrático
do Instituto Superior Técnico, da Universidade de Lisboa
Doctor Luís Eduardo Teixeira Rodrigues, Professor Catedrático
do Instituto Superior Técnico, da Universidade de Lisboa
Doctor Miguel Nuno Dias Alves Pupo Correia, Professor Associado
do Instituto Superior Técnico, da Universidade de Lisboa
Doctor André Ventura da Cruz Marnoto Zúquete, Professor Auxiliar
da Universidade de Aveiro
Doctor António Manuel Raminhos Cordeiro Grilo, Professor Auxiliar
do Instituto Superior Técnico, da Universidade de Lisboa

Funding Institutions
Fundação para a Ciência e a Tecnologia (FCT)

2014

Abstract

The combined use of Enterprise Resources Planning (ERP) and Supply Chain Management (SCM) systems has greatly improved the efficiency of supply chains. Further improvements require a deeper connection between the virtual and physical worlds. Automatic identification technologies, like radio-frequency identification (RFID), allow identification data about tagged physical objects to be collected by readers deployed across locations in the supply chain. This data is stored and managed using traceability systems to allow efficient answers to queries like *Track* and *Trace*. A practical traceability system should perform adequately for the large number of physical objects flowing in the supply chain (address the *scale* problem); and it should protect the sensitive business data from unauthorized access (address the *data visibility* problem).

The original contributions of this dissertation are: quantitative cost models that compare traceability systems for given supply chain scenarios; and visibility restriction mechanisms that can be used to define and enforce supply chain data access control policies. The analytic models take supply chain and target system parameters and compute cost estimates for data capture and queries, even when many implementation details are not available. The visibility restriction mechanisms are capable of identifying assets and stating the existence of records and the data access conditions, even if some of the supply chain partners are not known in advance. The policies are authored in RDF format with a distributed data model; and are enforced in a security infrastructure based on the XACML standard. The results are illustrated with examples from several industries and a case study in the Pharmaceutical supply chain.

Resumo

Os sistemas ERP (*Enterprise Resources Planning*) e SCM (*Supply Chain Management*) trouxeram grandes melhorias ao funcionamento das cadeias de fornecimento. No entanto, para continuar a melhorar é necessária uma maior ligação entre os mundos virtual e físico. A identificação automática por rádio-frequência (*RFID*) permite que dados de identificação de objectos físicos etiquetados possam ser recolhidos continuamente por leitores instalados em localizações relevantes (fábricas, armazéns, centros de distribuição, etc). Os dados são guardados e geridos por sistemas de rastreabilidade que permitem dar resposta a interrogações tais como *Localizar* e *Rastrear*. Um sistema de rastreabilidade prático deve ter um desempenho adequado ao grande número de objectos físicos que circulam na cadeia de fornecimento (*escala*); e deve proteger os dados de negócio de acessos não autorizados (*visibilidade de dados*).

As contribuições originais desta dissertação são: os modelos de avaliação quantitativa que permitem a comparação de sistemas de rastreabilidade; e os mecanismos de restrição de visibilidade que permitem definir e aplicar políticas de controlo de acessos aos dados. Os modelos analíticos partem de parâmetros da cadeia de fornecimento e do sistema para estimar os custos de captura e interrogação de dados, mesmo quando alguns dos detalhes de implementação não estão disponíveis. Os mecanismos de restrição de visibilidade permitem identificar os activos, declarar a existência de registos e as condições de acesso aos dados, mesmo quando alguns dos parceiros de negócio não são conhecidos à partida. As políticas são criadas no formato RDF com um modelo de dados distribuídos; e são aplicadas usando uma infra-estrutura com base na norma XACML. Os resultados são ilustrados com exemplos de várias indústrias e com um caso de estudo na indústria Farmacêutica.

Keywords & Palavras Chave

Keywords

- Information System
- Business-to-Business
- Supply Chain Management
- Traceability
- Automatic Identification
- Scalability
- Security
- Access control
- Internet of Things
- Pharmaceutical Supply Chain

Palavras Chave

- Sistema de Informação
- Negócio-para-Negócio
- Gestão de Cadeia de Fornecimento
- Rastreabilidade
- Identificação Automática
- Escalabilidade
- Segurança
- Controlo de Acessos
- Internet das Coisas
- Cadeia de Fornecimento Farmacêutica

Acknowledgments

I would like to thank my advisor, Professor José Alves Marques, for his exceptional dedication to my research, for his keen and wise insight, and for his pragmatic approach that helped me keep course during the whole journey.

I would also like to thank my co-advisor, Professor Sanjay Sarma, for the opportunity to collaborate with great people. Professor Sanjay is "curiosity embodied'", and his enthusiasm and sharp focus are passed on to his students in a great way.

I also thank all the members of the evaluation committee for taking their time to read my work and for their comments that have surely improved it.

A word of thanks is also due to the administrative assistants that connected me to the busy schedules of my advisers: Lena Simões, Sandrina, Olga; Cintia Castro, Emilia Arimah.

I thank the MSc students that I had the pleasure of co-advising: Guilherme Pereira, Nuno Rodrigues, Ricardo Carapeto, João Leitão, Carlos Perdigão, and last, but not least, Nuno Correia. Thanks also to Mário Romano and Link Consulting.

I thank my school, Instituto Superior Técnico (IST), Universidade de Lisboa, and the Massachusetts Institute of Technology (MIT), for giving me the working conditions necessary to achieve my goals.

Thanks to my IST colleagues and thanks to the MIT colleagues that have welcomed me during my visits: Christian Floerkemeier, Rahul Bhattacharyya, Isaac Ehrenberg, Sumeet Kumar, Ed Schuster, and Stephen Ho.

I was supported by a PhD fellowship from the Portuguese Foundation for Science and Technology FCT (SFRH/BD/45289/2008). I am grateful to my country for providing all the great education opportunities that I had during my life.

Thanks to the IEEE RFID and GS1 standards communities. I acknowledge the participants in the GS1 Discovery Services work-group and in the Event-based traceability work-group, with a special mention to Dr. Mark Harrison, whose calm perspective and deep technical knowledge were inspirational in many ways. I also acknowledge Dirk Rodgers for his insightful comments on the US Pharma supply chain, and the HDMA for providing the EDI guidelines and other reference materials.

I acknowledge Christine Robson of IBM for her helpful answers to some questions regarding the model developed to compare the Theseos and TraceSphere systems.

I acknowledge all the open-source programmers that have developed libraries that I have used for programming. Thanks also to Ingo Kegel for a complementary JProfiler license that allowed me to use a great tool.

I leave my dear friends and family for last.

Thanks to Miguel e Marta Panão, Zé João, Ricardo Ataíde, Nuno e Sofia Costa e Silva, Zé e Maria João Martins, Mary and Juergen, Andrea, Pato and Ester, Padre Duarte da Cunha, Padre Ruy Corrêa Leal, Padre Giovanni Musazzi, Padre José Luís Costa, Padre Nazário Kuatouta, and many others.

To the little ones: Tiago, Salvador, Raquel, Mariana, and all the teens from Nossa Senhora do Cabo, with a special mention to Alexandre; for making the future present and worth fighting for.

Ana, Sérgio and João for being there cheering me up along the way. Uncles and Aunts Pardal and Leitão for being present at the important moments. Mom Guida, thanks for the safe shelter during the storms. Grandma Suzete, thanks for all the prayer.

To my dear wife, Joana, there are not enough words to express the kinship, the love, the incentives, the laughs, the hugs, the... everything. You have earned the title "orient-amor" for sure.

To the faithfully departed, especially my grandfathers and my father Vitó, your love endures forever: *"In the days of my youth I was told what it means to be a man. Now I've reached that age I've tried to do all those things the best I can."*

To God for all of the above and what else lies ahead. Obrigado N'Anjos!

Lisbon, July 2nd 2014
Miguel

"The road goes ever on and on
Down from the door where it began.
Now far ahead the road has gone,
And I must follow, if I can,
Pursuing it with eager feet,
Until it joins some larger way,
Where many paths and errands meet."

– J. R. R. Tolkien, "The Lord of the Rings".

Contents

List of Figures

List of Tables

1 Introduction

The world around us intertwines countless supply chains and each one delivers physical goods from producer to consumer. Supply chains cross geographic, political and organizational boundaries. They are very large, open systems with multiple players and multiple authorities. They are not small, closed systems where everything can be managed centrally. An example supply chain is presented in Figure 1.1. In it, many organizations are involved and play different roles. The dashed arrows represent the flow of physical goods, and the solid lines represent contractual relationships.

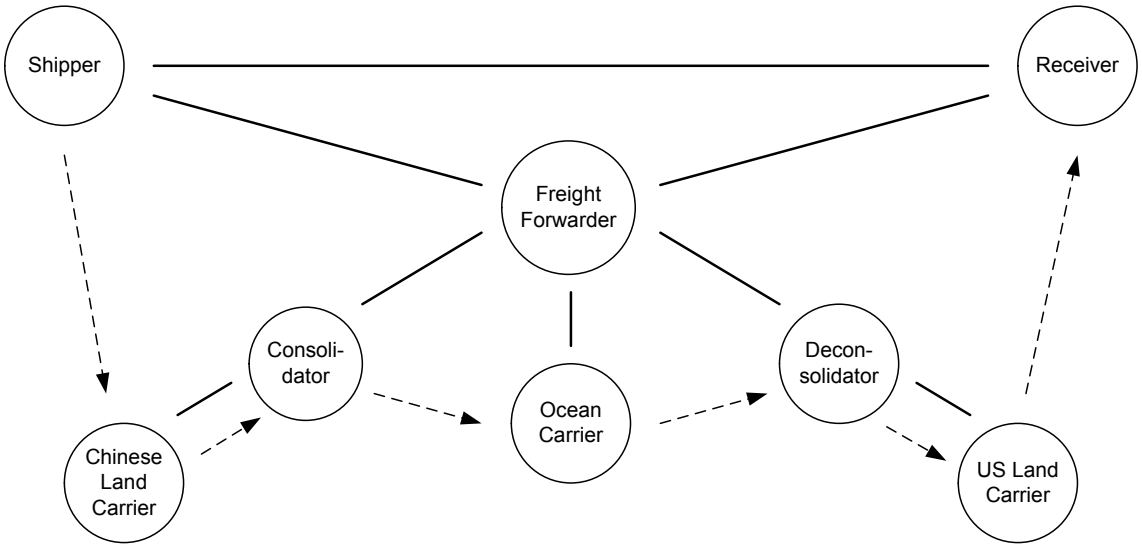

Figure 1.1: Transoceanic supply chain, supplying goods from China to the USA.

According to the Supply Chain Council [2007], a *supply chain* is formed by multiple companies – called *trading partners* – who are involved in the life-cycle of a product, from its origin to its delivery. The supply chain can be extended to include after-sale services and after-life recovery for recycling. Li and Ding [2007] describe some of the most common business arrangements for collaboration between trading partners: third-party logistics (3PL); vendor managed inventory (VMI); collaborative planning, forecasting, and replenishment (CPFR); and supply network (SN). Most industries depend on the management of the supply chain [Schuster et al., 2007], and some examples are: Automotive, Aerospace, Consumer Goods, and Pharmaceutical.

1.1 Information systems for supply chains

Information systems play a key role in this world-wide choreography of people and goods. The Internet is used to connect all the trading partners in the supply chains. The most relevant enterprise information systems related to supply chains are ERP and SCM. The overall business goal of SCM systems working together with ERPs is to optimize the flows of physical objects, i.e. to move the right amount of products in the least amount of time and at the lowest cost [Laudon and Laudon, 2011].

ERP (Enterprise Resource Planning) systems integrate the key internal business processes of a company into a single software system with the purpose of improving coordination and decision making.

SCM (Supply Chain Management) systems manage the relationships of the company with its suppliers, to optimize the planning, sourcing, manufacturing, and delivery of products and services. SCM can be used both for planning and execution. *Planning* enables the company to model its existing supply chain, generate demand forecasts for products, and develop optimal sourcing and manufacturing plans. *Execution* manages the flow of products through distribution centers and warehouses to ensure that products are delivered to the right locations in the most efficient manner.

SCM solutions have brought significant business improvements to companies but there are limits caused by *inaccurate* or *untimely* information. The physical world is in constant change and the world representation in the information system needs to keep up with it. To further improve solutions it is necessary to "sense" what is moving along the supply chain in greater detail. The *sense-control loop* of the information system needs to be tightened to achieve a deeper connection between the physical and the virtual worlds [Williams and Sanchez, 2007].

1.2 Automatic identification

Identification technologies are pivotal to collect better data about the physical objects. Linear bar-codes are the established technology for this purpose, but two-dimensional bar-codes and radio-frequency identification (RFID) [Finkenzeller and Muller, 2010] are the state of the art technologies. In particular, RFID is better at capturing data about physical goods than bar-code scanning and manual input because it has greater reading ranges and does not require line-of-sight to the object [Günther et al., 2008].

Figure 1.2 represents the use of RFID in a supply chain. Each object of interest is assigned an unique serial number and the number is stored in an RFID tag. At the most relevant physical locations of the supply chain – where items are shipped, handled or received – RFID readers automatically detect and identify the tags attached to the objects of interest and produce event data – *what* was sighted, *when* and *where*. The unique number is the *key* to the traceability data about the object.

RFID is not a new technology – Landt [2005] provides a good historical account of how it originated – but recent improvements in passive tags and UHF frequencies have made it more capable for use in supply chains. Some benefits of RFID for supply chains, observed by Weinstein [2005] and Derakhshan et al. [2007], are:

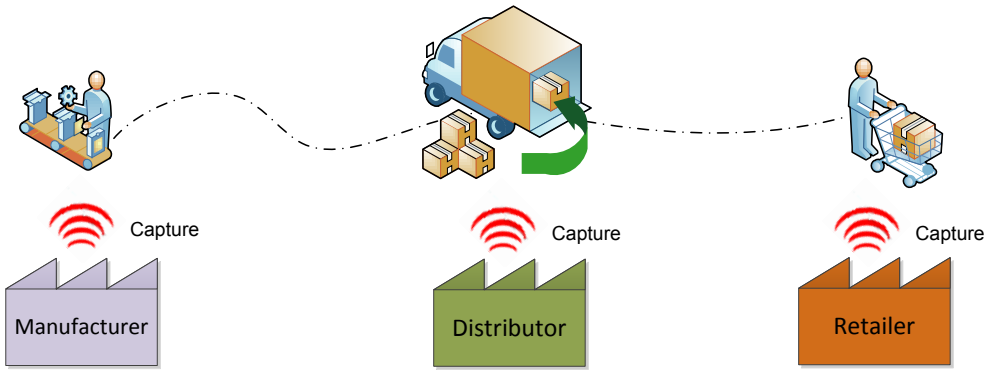

Figure 1.2: RFID data is captured and stored across the supply chain.

- More automation – RFID does not require line-of-sight access, has greater read range than bar-codes, allows simultaneous readings, and requires less human intervention;
- More safety – RFID item tracking provides more data about products in the supply chain; sensors can observe conditions and detect if environmental changes may have altered the item[1];
- Better inventory management:
  - Increased accurateness – RFID provides a comprehensive view of inventory throughout the supply chain, allowing more opportunities to optimize processes;
  - Faster replenishment – the detection of almost out-of-stock products can automatically trigger new orders.

RFID can be used to significantly improve the quality of supply chain data, making it more *accurate* and more *timely*, as intended.

1.3 The need for traceability

Traceability data is the result of observations made and collected at disperse physical locations in the supply chain. Data is stored and can later be retrieved using the unique identifiers as keys. This retrieval process is called *RFID data discovery*. The collected data can then be used to answer *traceability queries* [Agrawal et al., 2006] like the following:

- **Track query**: What is the current location of the physical object?
- **Trace query**: What is the location history of the object?

The answer to *track* is to know where the object is. It is a *downstream* view, usually from the perspective of the first company in the supply chain. The answer to *trace* is a complete

[1] Bhattacharyya et al. [2011] provide a specific example of a passive tag capable of temperature sensing.

supply chain pedigree (history) of a given object. It is an *upstream* view from the perspective of the current holder of the object in the supply chain. Ziekow and Günther [2010] and Wu et al. [2011] provide more detailed formulations of traceability queries.

Traceability systems can bridge the physical and virtual worlds and provide more accurate and up-to-date data to ERP and SCM systems so they can better achieve their business function. A *traceability system* is defined as an information system that manages the life-cycle of traceability data i.e. it collects and stores the data, and provides means to query the data. The next Section elicits actual user requirements.

1.3.1 BRIDGE project survey

The need for traceability data was documented in a requirements survey done in the context of the European Union BRIDGE project[2]. The survey [BRIDGE, 2007] was answered by 15 companies that were EPCglobal subscribers at the time[3]. The survey answers are summarized next.

Respondents listed supply chain efficiency, product authentication and safety issues as the most important *business drivers* for traceability. They wanted to be able to predict the location of objects, and they also needed exception reports for misplacements, duplicates, etc. These results confirm the intended use for the traceability systems.

Participants in the survey also expected to be able to track goods at item-level, single-case-level, and pallet-level. This means that aggregation events need to be captured and stored.

Respondents expected that the service would be updated and queried on each shipping and receiving event. The updates to the service should be available within 1 minute and preferably within 1 second. Simple query results should be returned within 1 second but complex query results could take longer and be returned asynchronously. This makes the expected performance level for the service very demanding, as traceability systems have to keep up with the pace of business.

The availability level of the service will have to be high because many critical business processes will rely on it. For this reason, the traceability system should be available 24 hours a day, 7 days a week.

The majority of participants in the survey expected that only the unique identifier would be stored on the RFID tag memory. This means that solutions for supply chains should not rely on advanced tag capabilities.

Respondents expected to store traceability data at multiple repositories, and almost all expected that their trading partners would also keep data about items. The majority expected to share data with their trading partners through dedicated repositories whereas the minority expected to share directly using their ERP systems.

[2]BRIDGE – Building Radio frequency IDentification solutions for the Global Environment – http://www.bridge-project.eu/

[3]EPCglobal – http://www.gs1.org/epcglobal – is part of GS1, the organization responsible for the bar-codes used around the world. EPCglobal defines the most relevant RFID standards for supply chain applications collectively called the EPC framework. An EPC (Electronic Product Code) is an unique serial number that can be stored in an RFID tag or in a bar-code.

Regarding the hosting of the service, the majority of answers stated that traceability services should be provided by certified providers, on a competitive commercial basis. This result shows that trading partners realize that some sort of Business-to-Business (B2B) platform is required to achieve supply chain wide traceability. B2B requires interoperability standards because otherwise there would have to be many point-to-point agreements between trading partners and that would not scale well to large systems. B2B also requires some form of trust in the service providers that may have partial or total access to the exchanged data.

The authors of the survey noted that there was a mismatch between the data that respondents wanted to obtain from the traceability service and the data that they were prepared to provide. This result points towards the need to control the access to traceability data.

1.4 The need for access control

The capability that a traceability system offers to other systems is to provide *visibility* of objects in a supply chain. It can answer – or assist in answering – traceability queries. The ultimate goal is usually referred to as the *total visibility supply chain* [Bose and Pal, 2005] where all trading partners in the supply chain can access all data about all goods, all the way from producer to consumer. This may give the impression that more visibility is always better. However, this utopia of seeing everything at all times misses the point. Business systems are supposed to provide added value to companies and not to satisfy unbound inquisitiveness. A pragmatic approach is needed, so traceability systems should be assessed with careful cost-benefit analysis. Companies should aim to achieve the capability to access data about *relevant* goods in *relevant* points in the supply chain, taking into account the cost of storing and retrieving that data. Deciding what is relevant is specific to a business context. For instance, if there are legal obligations, then the relevance is determined by regulations.

Besides the cost issue, sometimes visibility can expose too many internal details. Each trading partner can gain from exchanging data with other companies, but there is information such as the levels of demand, inventory, and supplier identities that should be kept to restricted circles of trust. There needs to be data access control intrinsic to the system design.

There is another issue relevant to the access control solution. The chain-of-custody[4] for each object of interest is unknown at the beginning of its life and is only revealed as the object makes its way along the supply chain. These *emergent object paths* mean that there is no way to anticipate which trading partners will be involved. This creates the need for access control mechanisms that can handle authorizations according to the flow of physical objects to accommodate new trading partners as they appear.

1.4.1 BRIDGE project survey (continued)

The BRIDGE [2007] survey also detailed which kinds of data companies would be willing to share in a traceability system. Most of them are only willing to share data under contract, with trading partners that are known to them. Overall, the study suggests that data sharing

[4] A physical object flowing in a supply chain defines both a *chain-of-custody* – sequence of trading partners that took possession of the good – and *chain-of-ownership* – sequence of trading partners that held property rights over the object.

should be limited to specific applications and for predefined purposes. However, the findings are not uniform across industries. In the Automotive industry there seems to be more reluctance to share item-level data. In the Aerospace, on contrast, there seems to be more willingness to share item-level data about the parts. Also, in the Consumer Goods industry there is willingness to share data, including pallet identification numbers and shipping notices. In the Pharmaceutical industry there is promptness to share relevant product data downstream, should a pedigree be required by law. Trust and negotiation power were found to be relevant to explain these differences. *Trust* is a decisive and important element of business relationships in supply chains. It allows savings because less inspections of incoming and outgoing objects are required. However, trust is hard to build and it is harder to build in longer supply chains because more trading partners are involved. Audit trails are a mechanism that can increase the trust in the system, by providing logs for all the changes. The *negotiation power* of a specific trading partner can also affect the willingness to share data. For example, a big retailer that trades with many small suppliers may coerce them to share data as a precondition for business.

1.4.2 Study by Eurich et al.

A study by Eurich et al. [2010] further investigated the reasons for the unwillingness to share traceability data. The authors performed a total of 16 interviews with organizations from Belgium, Germany, Switzerland and the USA. They found that sharing item-level data offers potential rewards but with significant risks. The potential *rewards* described in the study are:

- Precise tracking and tracing of products for electronic pedigrees;
- Targeted recalls of products;
- Improved vendor managed inventory and continuous replenishment programs;
- Automated counterfeit detection;
- Fewer out-of-stock occurrences;
- Identification of shrinkage;
- Theft detection.

Some of the *risks* identified in the study are:

- Reconstruction of strategic decisions;
- Threat to be penalized for unfair behavior;
- Revelation of distribution channels;
- Exposure to the development of a competing product;
- Weakening of the bargaining power after disclosure of purchase or supply volume;
- Difficulty in justifying the price;
- Loss of know-how.

The rewards are associated with the already discussed benefits of traceability. The risks are mostly related to giving competitors advantages that they currently do not possess, namely, by revealing internals of how the company adds value to their products. Figure 1.3 summarizes the study results for the present data sharing by companies and the willingness to share in the future. It shows that willingness is expected to grow, but the majority of companies are still very defensive about item-level data.

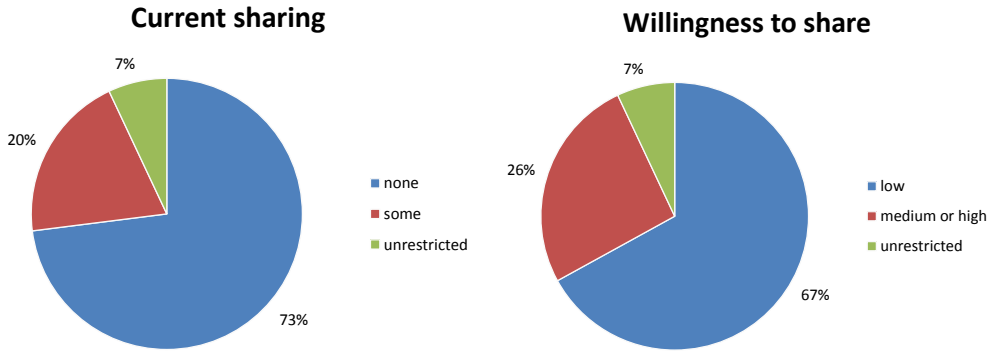

Figure 1.3: Eurich et al. [2010] study results for present sharing (left) and willingness to share in the future (right).

The authors of the study propose the following ways to minimize the risks and promote the use of traceability systems:

- A traceability system should have *fine-grained access control*;
- *Multiparty computation* should be used when possible, to allow the computation of useful results without revealing data;
- *Trusted third parties* should be intermediaries in data sharing. Only the trusted third parties have access to the raw data and propagate the results to the authorized participants.

Even if all these suggestions are implemented it is unlikely that total visibility will ever be a reality because of all the risks mentioned earlier. For example, Choi and Sethi [2010] have shown that there are business contexts where a manufacturer can profit more by providing low quality data to retailers of its products.

1.5 The need for scalability

The global scale of many supply chains with many companies implies that the service will have to be highly distributed with several locations scattered around the world. The scale of the system must be adequate to handle the expected volume of observation captures and queries. The scalability is closely related to the performance of the system that has to be accurately predicted and measured as the system grows in size. A system is said to be *scalable* if it can handle increasing numbers of users and resources without suffering a noticeable loss of performance or increase in administrative complexity [Neuman, 1994]. In the case of a traceability system, its scale can be measured in terms of the number of objects being tracked, queries being issued, and trading partners involved in the supply chain [Ilic et al., 2011].

1.6 Research goal

The presented BRIDGE [2007] and Eurich et al. [2010] surveys established that scale and security – *cost* and *visibility* – are the most important traceability system concerns. They are interrelated, as discussed by Burbridge and Harrison [2009]: visibility requirements may need to be relaxed to compose a more cost effective service, or, the scale of the service may be limited by the acceptable security practices.

This dissertation defines **practical visibility** as the capability to access data about *relevant* goods in *relevant* points in the supply chain, taking into account the *cost* of storing and retrieving that data, with *access control* defined by the data owners. Deciding what is relevant is determined by the supply chain participants. Therefore, finding the "right" traceability system architecture depends on the specific traceability need being considered.

Practical visibility entails the need to perform cost-benefit analysis with *quantification* i.e. the use of metrics to consider alternatives and compare trade-offs. This is a research challenge that has not been sufficiently addressed yet. The EPC framework [Traub et al., 2010] proposes a comprehensive suite of standards to build a traceability system, but it does not quantify the suitability of a solution to a specific business problem. Evdokimov et al. [2010] qualitatively compared traceability systems, by examining the software quality characteristics to check if the functional requirements were met, but they did not provide quantitative means to compare the suitability of an approach to a specific problem. Finally, the survey presented in Section 2.5 classified many traceability system proposals but none of the proposals provides tools to compare different solutions.

The research question for this work can be stated as:

> ¿ What is the best traceability system for a given supply chain problem, considering the cost of storing, retrieving, and protecting the data? ?

The *objective* of the research is to provide tools to allow the comparison of solution alternatives, using metrics whenever possible.

The *methodology* for research followed two lines:

1. The *cost assessment* line surveyed the existing proposals for traceability systems and built models to assess their performance and scalability, with increasing detail.
2. The *visibility* line surveyed data visibility requirements and built mechanisms for expressing and enforcing data sharing policies.

Assessing the cost of a traceability system is not simple, since most systems do not exist yet or are too big to deploy for testing. Implementing actual systems was not considered as a viable approach because of the need for a large investment before a significant deployment was achieved that could produce significant measurements. Previous work [Perdigão and Pardal, 2010] used simulation for preliminary and exploratory testing, showing that it is useful but also too costly to build. The approach finally followed was to build analytic models to estimate the cost of capturing and querying data, starting from supply chain and target system descriptions.

Assessing visibility required the specification and implementation of traceability data visibility restriction approaches that needed to be tested for correctness and performance. The research focused on access control mechanisms capable of handling emergent object paths. The end result includes a fine-grained discretionary policy implementation that can express the data visibility requirements of specific traceability applications.

1.7 Overview

RFID extends the reach of supply chain information systems in such a way that it will soon be possible and economically feasible to tag valuable physical objects and then to track and trace them throughout the supply chain, enabling many novel and useful applications.

The notion of *practical visibility* is proposed to guide the design of traceability systems. The cost-benefit trade-offs need to consider the scale and security of the target system.

The research described in this dissertation presents original contributions: **cost models** to predict computation and communication effort of traceability systems architectures; and **visibility policies** to make sure that the data is consumed only by authorized partners in the supply chain because, without trusted visibility policies, most trading partners will not be willing to participate in a traceability system.

The dissertation is set out in the following way.

Chapter 2 presents an extensive survey of traceability systems. It provides technical details of the identification, information, and discovery subsystems. It proposes a classification that summarizes more than twenty proposals. Additionally, it contains a threat assessment of discovery systems that confirms information disclosure (data visibility) as a critical security concern and presents existing protection approaches.

Chapter 3 presents an analytic model that takes supply chain and target system parameters to estimate the cost of data capture and traceability queries.

Chapter 4 describes the visibility restriction approaches that were considered and modeled using an extended version of the cost model that included specific information from system messages.

Chapter 5 describes how the visibility policies were actually implemented and their performance evaluated. An extensible data visibility policy implementation, called Chain-of-Trust Assertions (CTA), is proposed and described in detail.

Chapter 6 presents an in-depth case study in the Pharmaceuticals industry, that validated the use of the cost models and the expressiveness of the CTA visibility restriction mechanism.

Finally, Chapter 7 presents conclusions and future work.

1.8 Related publications

The results presented in this dissertation were published in the following articles, reviewed by peers in the RFID and information systems research communities.

Chapter 1 and 2:

Miguel L. Pardal and José Alves Marques "Towards the Internet of Things: An Introduction to RFID Technology", 4th International Workshop on RFID Technology - Concepts, Applications, Challenges – IWRT 2010.

Chapter 2 and 3:

Miguel L. Pardal and José Alves Marques "Cost Model for RFID-based Traceability Information Systems", IEEE International Conference on RFID Technology and Applications – IEEE RFID TA 2011

Chapter 4:

Miguel L. Pardal, Mark Harrison and José Alves Marques, "Assessment of Visibility Restriction Mechanisms for Discovery Services", IEEE International Conference on RFID – IEEE RFID 2012.

Chapter 5:

Miguel L. Pardal, Mark Harrison, Sanjay Sarma and José Alves Marques, "Enforcing RFID Data Visibility Restrictions Using XACML Security Policies", IEEE International Conference on RFID Technology and Applications – IEEE RFID TA 2012.

Miguel L. Pardal, Mark Harrison, Sanjay Sarma and José Alves Marques, "Performance Assessment of XACML Authorizations for Supply Chain Traceability Web Services", 8th International Conference on Next Generation Web Services Practices – NWeSP 2012.

Chapter 6:

Miguel L. Pardal, Mark Harrison, Sanjay Sarma and José Alves Marques, "Expressive RFID data access policies for the Pharmaceuticals supply chain", IEEE International Conference on RFID – IEEE RFID 2013.

Miguel L. Pardal, Mark Harrison, Sanjay Sarma and José Alves Marques, "Access Control Policies for Traceability Information System", International Journal of Computer Information Systems and Industrial Management Applications – IJCISIM, Volume 6, 2014.

2 Architecture of Traceability Systems

Traceability systems capture and manage traceability data with the purpose of answering queries such as Track (*Where is the object of interest?*) and Trace (*Where has the object been?*). Individual parts of a traceability system may appear basic but the composition of the overall system is not trivial.

This Chapter discusses the architecture of a traceability system by decomposing it in three subsystems, each with a specific purpose. The *identification subsystem* specifies the automatic identification technology required for the collection of traceability data. The *information subsystem* specifies what happens to the data once it is captured, namely, how data is stored and managed. Finally, the *discovery subsystem* specifies the process that locates the data required to answer the traceability queries.

The EPCglobal framework [Traub et al., 2010] is used to illustrate each subsystem because the EPC standards specify a complete traceability system. The final part of the Chapter presents an extensive survey of traceability system proposals, a threat assessment, and a review of available protections for such systems.

2.1 The GS1 EPCglobal framework

GS1[1] is an international non-profit association dedicated to the development and implementation of standards to improve supply chains. It is also the organization that oversees barcode use in the world. GS1 *identifiers* are widely used and are very useful for supply chains. Two of the most important GS1 identifiers are the GTIN (Global Trade Item Number) and the GLN (Global Location Number). A GTIN is used to identify any item upon which there is a need to retrieve predefined information and that may be priced or ordered or invoiced at any point in a supply chain. A GLN is used for location: physical, functional or legal entities requiring a permanent identification, such as a company, department, or warehouse. These identifiers can be extended with serial numbers to create unique identifiers for individual products and locations. The serialized identifiers are called SGTIN (Serialized GTIN) and SGLN (Serialized GLN), respectively. Appendix B provides a brief overview of GS1 identifiers.

The Electronic Product Code (EPC) standard defines globally unique identifiers for items in the supply chain. The original creator of the Electronic Product Code (EPC) technology for RFID was the Massachusetts Institute of Technology (MIT) Auto-ID Center. In 2003, EPCglobal was formed as the successor organization to manage the EPC standards along with the Auto-ID Labs – at MIT, Cambridge UK, St. Gallen, KAIST, Fudan, Adelaide, and Keio – to do research on the EPC technology. EPCglobal is now a part of GS1.

[1] **G**lobal **S**tandards **1** – http://www.gs1.org/

The GS1 EPCglobal framework [Traub et al., 2010] [Thiesse et al., 2009] is a comprehensive set of standards that define the hardware, software, and data for a traceability system. The EPC framework is designed to facilitate the exchange of information and physical goods between trading partners in a supply chain.

The EPC framework components are represented in detail in Figure 2.1. They include: tags, readers, data repositories, applications and services. The data flow through the components is the following: readers query tags and read their memory contents producing observations. The observations are collected, filtered and used to generate events that add more context. Finally, applications consume events.

The framework specifies interfaces, not implementations, leaving room open for different implementations. For instance, the *Fosstrak* project[2], authored by Floerkemeier et al. [2007], is an open-source implementation of the EPC ALE, IS, and LLRP software standards.

The EPC standards are presented next in more detail, organized by traceability subsystem: identification, information, and discovery.

2.2 Identification subsystem

The technology focus of the identification subsystem is on RFID [Finkenzeller and Muller, 2010] that allows physical objects to be monitored with much greater granularity than using bar-code technologies.

2.2.1 RFID technology

RFID technology is at a mature stage of development, with increasing adoption in many industries [Schuster et al., 2007]. Figure 2.2 shows an example of a RFID reader with two antennas and a UHF tag. RFID readers collect data statements such as: "Object O was seen at time T, place L." and "Object O was aggregated into pallet P".

RFID is not a single technology but a suite of technologies. The choice of the "right" tags and readers must be made considering the application requirements and the working environment. The best suited tags for use in supply chains are passive UHF tags because they are the least expensive per unit – which is important given the high volumes of objects to track – and they support greater reading ranges. Appendix C discusses the operating principles of RFID in more detail.

2.2.2 Tag standards

The *tag* standards specify how tags should operate and how to store data on them. The most important standard is called UHF Class 1 Gen 2 [EPCglobal, 2008a].

The basic RFID data element is the EPC code, a globally unique number, typically 96 bits long. It is formed by a header, a company identifier, an object class identifier, and a serial number.

[2]Fosstrak – Free and Open Source Software for TRAcK and trace – http://www.fosstrak.org/

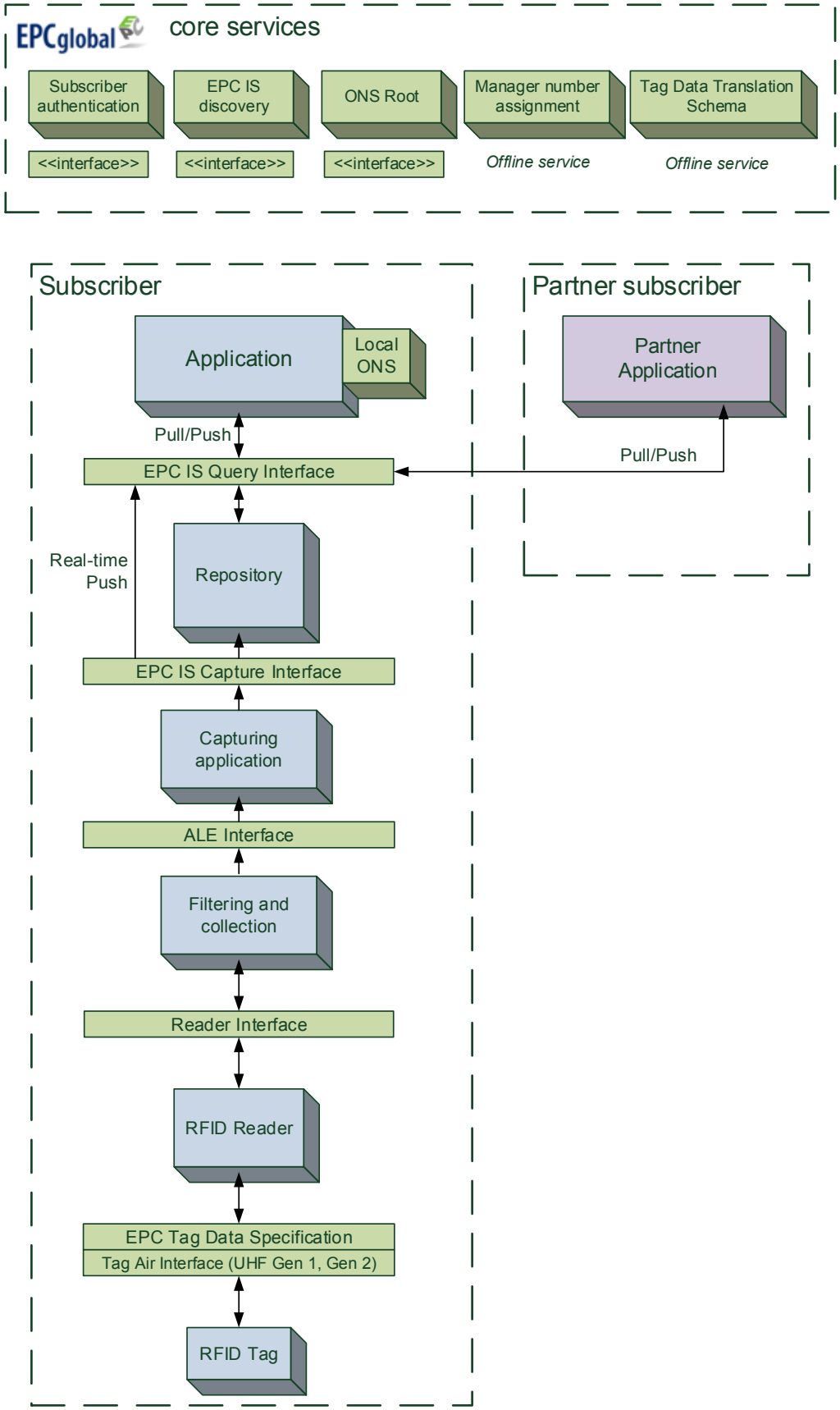

Figure 2.1: Overview of the EPC Architecture Framework.

Figure 2.2: RFID antennas, reader, and tag (images courtesy of Alien Technologies).

The Tag Data Standards (TDS) [EPCglobal, 2010b] and the Tag Data Translation (TDT) [EPCglobal, 2009b] define the EPC tag data, including how that data is encoded on the EPC tag itself – the EPC Tag Encodings – as well as how it is encoded for use in information systems – the EPC Uniform Resource Identifier (URI) Encodings. It also indicates how coding systems such as the GS1 family of codes – GTIN, GLN – should be embedded within the EPC.

2.2.3 Reader standards

The *reader* standards define communication protocols between RFID readers and tags. The Reader Protocol (RP) [EPCglobal, 2006] specifies the interaction between a device capable of reading – and possibly writing – tags, and application software. The Low-Level Reader Protocol (LLRP) [EPCglobal, 2007b] provides greater control to clients over the use of the radio-frequency channel and the tag features that are protocol-specific. The Reader Management (RM) [EPCglobal, 2007d] interface deals with the configuration and control of operating status of reader deployments.

2.3 Information subsystem

After data is collected using tags and readers, it is up to the information subsystem to handle the data. It specifies what happens to captured data so it can later be retrieved by the discovery subsystem. Its main functions are: connecting to readers, filtering data, and capturing events.

The readers and the information systems are scattered in disperse locations and are connected using Internet Protocol (IP) networking technology [Tanenbaum, 2002].

2.3.1 RFID data

An *RFID observation* is a primitive event that holds the literal values of *What*, *When* and *Where*. Observations are instantaneous and atomic in the sense that they are either totally captured or not at all. Observations are read-only i.e. they cannot be updated or deleted. Even small RFID deployments can generate gigabytes of observations a day because of continuous tag readings [Derakhshan et al., 2007]. Observations are also inaccurate due to errors, such

as duplicate or missing reads. Read rates are in the 60-70% range for real-world deployments reported by Floerkemeier and Lampe [2004] and by Jeffery et al. [2006].

An *event* is the result of one or more observations with business relevance and context [Etzion and Niblett, 2010]. An event adds *Who* and *Why* values and specifies *What*, *When*, and *Where* in a more meaningful way. Event capture usually requires situation context – how is data being collected e.g. the physical object is at an entry gate – and business context – what is the meaning of the event to the business process e.g. the physical object being read corresponds to a purchase order being received.

Floerkemeier et al. [2007] summarize the challenges in RFID data capture:

- False negative caused by interference or absorption by objects;
- False positive reads when tags are detected outside the typical range of a reader;
- Heterogeneous readers with different computing and networking capabilities;
- Read collisions that require reader coordination to be avoided;
- Limited bandwidth available per channel that limits the data transfer rate between readers and tags.

2.3.2 Filtering and collection standard

The EPC Application Level Events (EPC ALE) [EPCglobal, 2009a] specifies a software interface through which client applications may access filtered, consolidated data from multiple sources. The data is processed using basic rules like "wild-cards". ALE needs to be fast with low complexity processing to be able to cope with high volume data streams.

Floerkemeier et al. [2007] list the required operations on RFID data:

- Accumulation by time, space, count:
  - Items detected in the last minute, appearing/disappearing items;
  - Items detected by multiple readers at same location;
  - Product type totals;
- Filtering: data may be filtered into subsets based on tag identity, tag memory, reader antenna, reader identifier, etc;
- Dissemination: data captured by a reader is of interest to multiple applications inside and outside of the company. There is a need to support asynchronous and synchronous messaging because different applications may require different response latency.

EPC ALE deals with real-time processing – *what is happening* – while the next standard, EPC IS, deals explicitly with historical data – *what happened*.

2.3.3 Information Services standard

The EPC Information Services (EPC IS) [EPCglobal, 2007a] specification is concerned with recording business events that can serve as the basis for a wide variety of enterprise-level information processing tasks. It supports internal data capture and external data sharing about the movement and status of the physical goods. EPC IS is responsible for capturing data coming from EPC ALE, adding more business context, and making the data available for queries or subscriptions. The specification indicates how events and queries should be structured and the technology bindings that should be used for data exchange with partner systems. The EPC IS has the following components:

- Capture Interface: interprets captured EPC event data;
- Repository: stores EPC event data persistently;
- Query interface: interprets queries, and retrieves the requested data. It includes:
  - Control interface: accepts "on demand" queries that are answered immediately, and "standing" queries that are subscriptions for future events;
  - Callback interface: delivers query results that must be delivered upon capture, bypassing the repository.

The EPC IS data model is presented in Table 2.1 and it shapes the way in which an EPC IS "sees'" the world. There are four different event types in the data model. An *ObjectEvent* corresponds to an observation of an EPC code. A *QuantityEvent* can be used to include only the product type and the number of objects when individual identification is not needed. An *AggregationEvent* represents an aggregation – or disaggregation – of a group of items. Finally, a *TransactionEvent* links EPCs to a specific business transaction, such as a Purchase Order (PO).

Besides event data, the EPC IS repository includes *Master Data* that makes use of data vocabularies to refer to business locations, process steps, and transactions. There is a Common Business Vocabulary standard [EPCglobal, 2010a] but there are also vocabularies specific to industries or companies.

2.4 Discovery subsystem

After the data is collected and organized, it should be retrievable when needed. The discovery subsystem defines the data *lookup* mechanisms that retrieve data to answer queries. Agrawal et al. [2006] define the following traceability queries:

- **Track/Recall query**: What is the current location of the physical object?
- **Trace/Pedigree query**: What is the location history of the object?
- **Aggregation/Bill-of-Materials (BoM) query**: What are the components of the object?

The lookup process for traceability uses EPC unique serial numbers that are structured and hierarchical. They are composed of several parts: country of origin, company issuing the identifier, product class, and serial number. The function of ONS, described in the next Section, is to translate an EPC code into a service address.

		EPCISEvent (base class)	Object-Event	Aggregation-Event	Quantity-Event	Transaction-Event	
When?	eventTime	●	●	●	●	●	Time of event observation
	recordTime	❍	❍	❍	❍	❍	Time of event registration
	eventTimezone-Offset	●	●	●	●	●	Time zone information
What?	epcList		●			●	List of observed EPCs
	parentID			❍		❍	Containing object ID
	childEPCs			●			Contained objects IDs
	epcClass				●		Product type
	quantity				●		Number of observed objects
	action		●	●		●	Life-cycle phase of the EPCs
Where?	readPoint		❍	❍	❍	❍	Reader name
	bizLocation		❍	❍	❍	❍	Location name
Why?	bizStep		❍	❍	❍	❍	Business process step
	disposition		❍	❍	❍	❍	State of the objects
	bizTransactionList		❍	❍	❍	●	Associated transactions

Table 2.1: EPC IS event types and attributes (● = mandatory, ❍ = optional) [Thiesse et al., 2009].

2.4.1 Object Name Service standard

There are two main approaches to lookup names in distributed systems:

- *Identifier*-based, illustrated by DNS (Domain Name System) [Albitz and Liu, 2006], and used extensively in the Internet;
- *Attribute*-based, exemplified by LDAP (Lightweight Directory Protocol) [Arkills, 2003], and used internally in many organizations.

The Object Naming Service (ONS) [EPCglobal, 2008b] is an *identifier*-based naming system that resolves EPC product codes into data and services about the product provided by the company that issued the EPC code. The ONS architecture and protocols are based on DNS.

Figure 2.3 illustrates how an ONS resolver works. First, the tag is detected by a reader (step 1). The raw EPC is translated into an URI (steps 2 and 3) and then into a DNS domain name, using the company prefix (step 4). The ONS resolver issues a DNS query starting with the ONS root 'onsepc.com' (step 5) that retrieves a NAPTR record with an address and a service type (step 6). The result can be an address for an EPC IS repository or a web page with information about the object (steps 7 and 8). For example, a DNS NAPTR record can be returned, containing a service type and a value e.g. EPC+ws `http://epc-is.example.com/epc.wsdl`.

The serial number is discarded in the resolving process. ONS only gives access to the original EPC issuer company and it provides no data about the current and past whereabouts of a tagged physical object. This means that ONS cannot be used for lookup of data about individual physical objects, but only for class-level data. This creates the need for a discovery standard to retrieve serial-level data.

Another concern with the current ONS standard is that it assumes a centralized naming authority – just like DNS. However, since ONS is such a critical resource, potentially affecting a

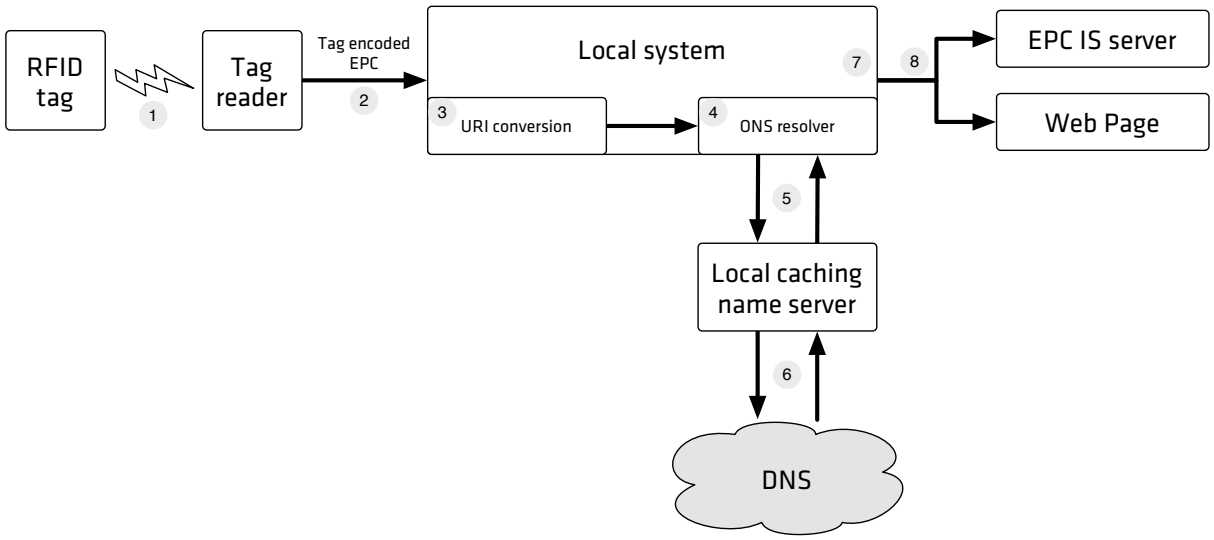

Figure 2.3: ONS resolver data flow.

large number of supply chains, some countries may not be willing to concede the capability of identifying items to a foreign nation. There are proposals for federated ONS [Balakrichenan et al., 2010] with multi-polar authority that may result in a more balanced sharing of administrative power.

2.4.2 Discovery Service standard

At the top of the EPC framework, there is a *place-holder* for a Discovery Service (DS) standard. According to GS1, the *tentative* responsibilities of EPC DS are:

- To provide a means to locate all EPC IS services that may have data about a specific EPC;
- Provide a cache for selected EPC IS data;
- Enforce data access authorization.

Being so, a DS is a possible architecture that allows applications to locate multiple sources of data to answer traceability queries. While the EPCglobal standard is still under development there have been studies and proposals about other possible architectures to solve the serial-level data lookup problem. There are several possible architectures arrangements to consider.

In a *link traversal* or *follow-the-chain* approach, the lookup locates the issuer's company via an ONS query and then follows the onward links from one company to the next. This may not be possible if one of the links is broken, as illustrated in Figure 2.4.

In a *directory* approach, the lookup queries a directory to find links to companies with data about the object. This approach works even if some of companies are temporarily or permanently unreachable, as illustrated in Figure 2.5. A directory-based discovery service has two information flow alternatives: it can use a *push* model [Cantero et al., 2008], where data is

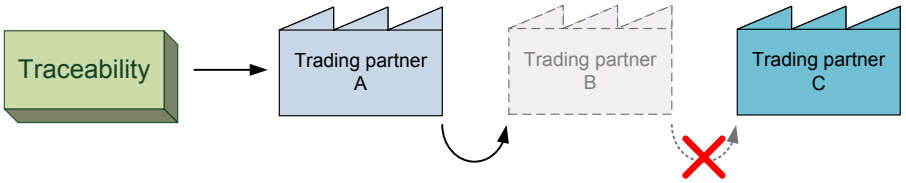

Figure 2.4: Discovery using link traversal.

published directly to the directory; or a *pull* model [Kürschner et al., 2008], where queries are issued to the directory that then forwards them to the companies.

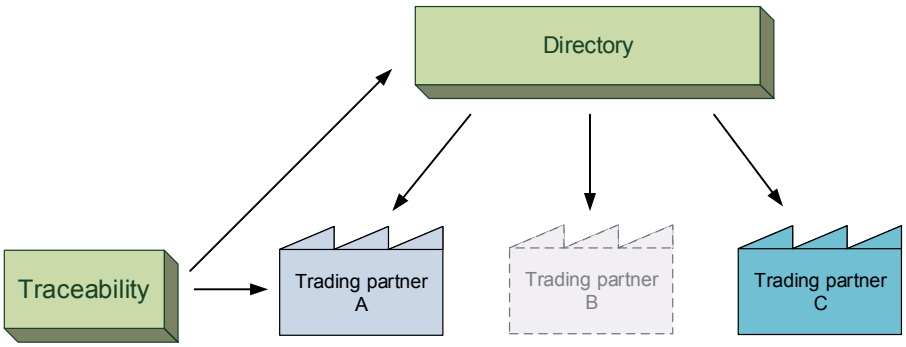

Figure 2.5: Discovery using a directory.

A prototype was implemented to explore the DS design issues and is described in Appendix E. Still, a survey of proposals was required to get a broader view of the design alternatives.

2.5 Data discovery proposal survey

A DS is just one of the possible approaches to build a traceability system. The overall goal of serial-level lookup is to facilitate RFID data exchange between trading partners in a supply chain. There are several architecture proposals, but it is unclear which is the best for a given supply chain problem. Also, there is no way to check for overlaps in architectural approaches and data handling procedures.

This Section presents an updated version of a survey conducted by Pardal and Marques [2011]. More than thirty publications were found and were consolidated into a total of twenty proposals. Still there were too many systems and no effective way to compare them. The different proposals were classified using the criteria of *centralization* and *data integration*. These criteria were adapted from a proposal by Do et al. [2006]. The *centralization* criterion considers the reliance on special nodes for data capture and query processing. A centralized system has nodes with special functions whereas in a decentralized system all nodes are functionally equivalent. The *data integration* criterion considers where data is physically stored. Data can

be *copied* to specific locations – materialized integration – or *referenced* – virtual integration. Combining the criteria, there are four distinct approaches and their respective proposals are listed next. Figure 2.6 summarizes the survey results graphically and a representative proposal is highlighted in each quadrant.

- **Unstructured peer-to-peer (UP2P) approach** (virtual, decentralized):
  - ePedigree [Huang et al., 2007],
  - **Theseos** [Cheung et al., 2007] – has several distributed data stores and answers queries recursively meaning that each node can manage its data set and enforce its own data access policies;
- **Metadata integration (MDI) approach** (virtual, centralized):
  - GS1 PoC [Tearnen, 2005],
  - Verisign DS [Verisign, 2008],
  - IBM PoC [Beier et al., 2006],
  - PTSP [Cao et al., 2007],
  - EPCISDS [Lee et al., 2008],
  - EPCDS [Worapot et al., 2010],
  - ADS [Müller et al., 2010],
  - Afilias ESDS [Young, 2008],
  - **BRIDGE Directory** [Cantero et al., 2008] – relies on centralized services to store data links and is closely aligned with the EPCglobal architecture,
  - UniSalento DS [Barchetti et al., 2010],
  - SLS [Polytarchos et al., 2010],
  - BRIDGE Query Relay [Kürschner et al., 2008],
  - UniPR DS [Rizzi et al., 2012],
  - EPCIS caching [Song et al., 2006],
  - TraceSphere [Robson et al., 2007],
  - IOTA [Laurence et al., 2010];
- **Structured peer-to-peer (SP2P) approach** (materialized, decentralized):
  - LoTR [Wakayama et al., 2007],
  - UniKoeln DS [Schoenemann et al., 2009],
  - **OIDA** [Fabian, 2009] – relies on a Peer-to-Peer (P2P) network with a hashing algorithm for fully decentralized data placement in nodes. It uses a DHT (Distributed Hash Table) [Balakrishnan et al., 2003] that is fully decentralized and has potential for high scalability,
  - InnoSem [Schoenemann et al., 2009],
  - WWAI [Do et al., 2006];

- **Data integration (DI) approach** (materialized, centralized):
  - **ID@URI** [Framling et al., 2007] – uses a product-agent architecture where all data concerning the item is forwarded to a central data store, managed by the product's manufacturer.

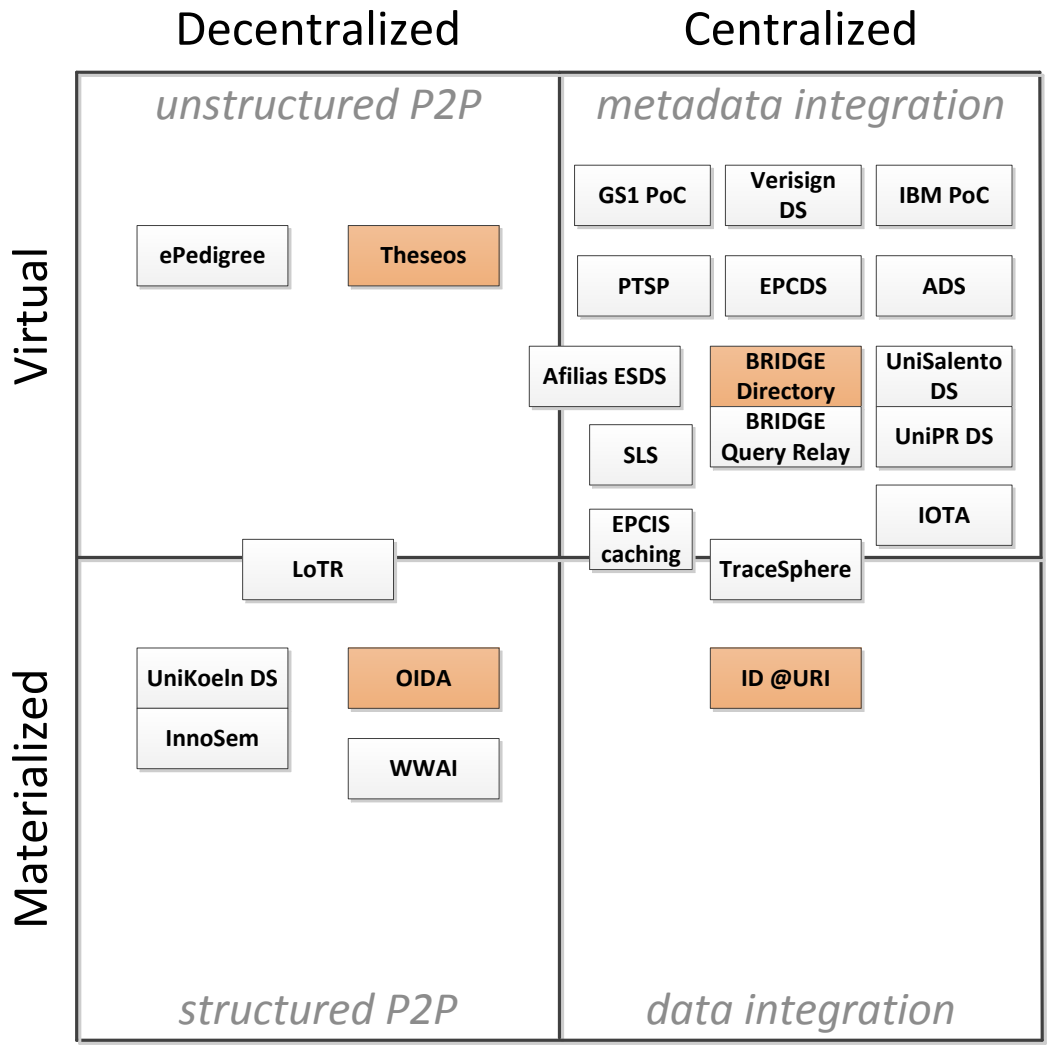

Figure 2.6: Data discovery proposal classification.

2.6 Security of discovery services

A system is said to be *secure* if it defines a policy stating what should be protected, and implements mechanisms to protect its valuable assets against improper use [Tanenbaum and van Steen, 2007]. The protection depends on the *value* of the assets, in this case, the traceability data. A traceability system requires identification, authentication, and authorization of the trading partners [BRIDGE, 2007].

This Section presents a threat assessment and a protection survey for the discovery subsystem of a traceability system. This assessment is aimed at understanding the risks that the threats pose, and the available mitigation that can reduce those risks.

The protection proposals that were found apply to the MDI architecture.

2.6.1 Threats

A *threat* is a potential violation of the security policy and a *vulnerability* is a flaw or weakness in a mechanism's design, implementation, operation, or management that can be exploited to breach the system.

To elicit threats to the EPC framework, Garcia-Alfaro et al. [2008] used the Microsoft Threat Analysis and Modelling (TAM) that is a part of Security Development Life-cycle (SDL) [Howard and Lipner, 2006].

STRIDE[3] (that is part of TAM and is also recommended by the OWASP[4]) was used to classify the threats and the ETSI[5] methodology [ETSI, 2003] was followed to assign severity degrees to the threats – *critical*, *major*, and *minor* – according to their likelihood of occurrence, their possible impact upon the target systems, and the risk that they represent.

The threat modeling analysis identified the following threats:

- *Spoofing of identities* threat: the attacker impersonates the ONS server associated to a company, by using a man-in-the-middle attack for example, in order to intercept queries addressing products associated to a company; The attacker impersonates an external application and executes a dictionary attack in order to generate random queries that target the ONS instances associated to the lookup service utilized by a company.
- *Data tampering* threat: the attacker intercepts queries sent from an external application to the ONS instance associated to a company and responds with false URLs.
- *Repudiation* threat: illegal changes made to the ONS component of the lookup service cannot be detected because there are no integrity checks.
- *Information disclosure* threat: ONS operations are based on a clear text protocol which uses domain names constructed using some field values of corresponding EPCs. It means that the use of these domain names without additional countermeasures leads to a leakage of data, such as manufacturers and product classes.
- *Denial of service*: ONS can be flooded with requests, rendering the the service inoperable.
- *Elevation of privilege*: ONS has no authentication procedures for the exchange of information, so configuration problems or programming flaws on the systems can be exploited by attackers to elevate their privileges and expose protected resources.

[3] **STRIDE** stands for **S**poofing identity, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, and **E**levation of privilege.

[4] Open Web Application Security Project – https://www.owasp.org/

[5] European Telecommunications Standards Institute

The underlying reason for most of these vulnerabilities consists of the fact that DNS is a highly exposed service [Atkins and Austein, 2004] that provides no way of authenticating a client, the server, or the information provided. These weaknesses directly transfer to ONS.

The ETSI classification is presented in Table 2.2. The 'spoofing', 'information disclosure', and the 'elevation of privilege' threats are ranked at the *critical* level; and 'tampering', 'repudiation', and 'denial of service' threats were considered as *major*.

Threat	Motivation	Difficulty	Likelihood	Impact	Risk
Spoofing of identities	*High*	*Solvable*	*Possible*	*High*	***Critical***
Tampering with data	*Moderate*	*Solvable*	*Possible*	*Medium*	*Major*
Repudiation	*Moderate*	*Solvable*	*Possible*	*Medium*	*Major*
Information disclosure	*High*	*Solvable*	*Possible*	*High*	***Critical***
Denial of service	*Moderate*	*Solvable*	*Possible*	*Medium*	*Major*
Elevation of privilege	*High*	*Strong*	*Unlikely*	*High*	***Critical***

Table 2.2: STRIDE table for the EPC framework [Garcia-Alfaro et al., 2008].

2.6.2 Protections

In this Section, a literature review of protection proposals is presented for the ONS, IS, and DS. Attackers can intercept unprotected ONS, IS, and DS queries and its contents expose data about product and raw material flows, and other sensitive business information.

ONS: The main approach to address the security shortcomings of DNS is called DNSSEC (DNS Security Extensions) [Friedlander et al., 2007]. It introduces two independent procedures: TSIG (Transaction Signature) that provides mutual authentication between two DNS servers by using shared secrets; and, RRsets (Resource Records sets) that provide authenticity and data integrity for DNS information by using public-key cryptography to sign sets of records. Global ONS information integrity could only be assured by DNSSEC in the long run, if the Internet community as a whole adopts it too. However, this is still not the case.

The main privacy enhancing strategy for ONS proposed by Fabian et al. [2005] is obfuscating the source IP of the ONS query by using anonymous mixes that collect ONS queries from different sources and mix them together to obfuscate the source of individual queries.

IS: The communications with EPC IS can be protected by the use of TLS/SSL (Transport Layer Security/Secure Sockets Layer) [Dierks and Rescorla, 2008] or WS-Security [Lawrence et al., 2006].

Grummt and Müller [2008] propose a rule-based, context-aware policy language for describing access rights on large sets of EPC IS events. The Auto-ID Authorization Language (AAL) language leverages the structure of IS data. The traceability data sets are called AAL Shares. Each one defines a subset of IS events of a specific event type. Rules can refer to the content of the events and to contextual information such as business relationships and the current time. The approach uses query rewriting into several sub-queries in order to achieve flexible row, column, cell based restrictions specified by rule-based policy definitions. The

policy language realizes that companies will most probably prefer defining subsets of events to be shared using dynamic rules instead of static access control lists.

DS: The communications with DS can also be protected with TLS/SSL or WS-Security.

Worapot et al. [2010] propose visibility policies for data in a DS directory. The supported policies are: all, none, in chain visibility.

Shi et al. [2012] implemented SecDS, a directory DS, and used an attribute-based access control (ABAC) to implement fine-grained access policies. They assume that DS is a trusted server where a trusted authentication mechanism is implemented. They provide support for three kinds of visibility policies: whole-stream policy, up-stream policy, and down-stream policy. In order to reduce the cost of users' queries, ABAC policies are transformed into a fine-grained access control (FGAC) policies which use SQL predicates to express the users' privileges. The authors implemented a prototype of the SecDS system that showed that it is practical.

Shi et al. [2012] also proposed SecTTS, a DS with a *pull* model, where a relay policy allows each supply chain partner to define different relay policies for different data sets. The DS in SecTTS is semi-trusted and is considered as "honest-but-curious". For "honest", DS will correctly follow the designated protocol specification; for "curious", DS will infer information of supply chain partners from its own operation, namely, which EPCs have been handled by a given company. The authors propose a fuzzy relay model. For example, for EPC urn:epc:id:sgtin:0649588:183409.22072006078 which represents sensitive information, a set-based relay policy is published with the generalized EPC urn:epc:id:-sgtin:0649588:183409.2207*******. In this case, the DS only has $1/10^7$ confidence that the company has handled the product with the stated EPC. In this approach, companies have to be willing to answer queries related to objects that they do not know. However, this apparent problem, provides *plausible deniability* to data owners if they do not wish to respond to some traceability queries. The authors implemented a prototype of SecTTS and conducted performance experiments and the results validated the design as practical.

2.7 Conclusion

The EPC framework was used as an illustrative example of the identification, information and discovery subsystems that make up a traceability system but the EPC Discovery Service (DS) is still unspecified.

The data discovery proposal survey studied over twenty proposals. To summarize the many published system proposals, they were classified using two architecture criteria: centralized versus decentralized solutions, and data copying versus data referencing. Centralization produces a clear distinction because it identifies the need for special roles that require more trust and that are suited for third party service providers. Since the data handling was also very sensitive, the data integration approach (copy versus reference) was also used as a distinction between systems.

The larger cluster of proposals was found at the MDI architecture quadrant, because companies prefer to keep their data under control, and MDI offers a limited trust approach. Perhaps there is also influence from the tentative EPC DS standard. The other – smaller – cluster is at the SP2P architecture quadrant. Here it is the scale concern that dominates and a

DHT approach is used by several proposals. These survey clusters support the claim made in Chapter 1 that security and scale are the core concerns for a traceability system. MDI is a flexible architecture and can accommodate both concerns. A trusted third party can intermediate data sharing between trading partners, and can use DHT internally to achieve better scalability, if desired.

The MDI architecture security was analyzed with a security threat assessment and a protection proposal survey to verify if it could actually play a trust broker role, as suggested. The protection survey confirmed that there are indeed solutions that can take advantage of an intermediary third party. The survey also showed that fine-grained access control is viable and can have adequate performance. The translation of an higher-level security policy to a lower-level policy for enforcement is a proven practice that can yield positive results. It allows the specification of policies in a more meaningful way for the asset owners and enables formulations that are more suited to enforcement infrastructures.

3 Assessing Cost

This Chapter presents a cost model that allows a comparison of the data capture and query cost of different traceability system architectures. The costs being estimated are response times and storage volumes.

A graph model is proposed to represent the relevant characteristics of the supply chain being considered. Then, the target system model is defined to compute the costs without relying on implementation details that are not available. The model is applied to two example supply chains and the results are discussed at the end of the Chapter.

3.1 Traceability cost model

An analytic cost model approach was chosen to estimate the performance because it allows a comparison of different architectures without requiring actual system implementations. The supply chain and the information system are represented using parameters. The developed model is based on previous work by Murthy and Robson [2008] to compare two systems: Theseos and TraceSphere. The model abstracted out the characteristics of the two systems and built a query execution model to compare the performance.

The developed model estimates total system cost from the cumulative processing and communication costs of data capture and query execution [Pardal and Marques, 2011]. The challenge is finding the parameters that capture the essential characteristics of the supply chain and of the system.

3.1.1 Modeling the supply chain

Each object path in the supply chain can be represented with a graph, as illustrated in Figure 3.1. A *vertex* is a node corresponding to a company. An *edge* represents a connection between companies. The supply chain graph model is described in detail in Appendix D.

The cost model's parameters represent characteristics of the supply chain and are presented in Table 3.1.

The number of nodes in the chain (n) is the number of vertices in the supply chain graph. The average number of item records per node (r) is the average of the number of events recorded on each node about a particular item. The average length of the chain graph (z) is the average of the length of all item paths.

The average depth (b) and children per node (c) parameters are used to characterize an average product BoM tree. The number of components for a level is given by c^b with b starting at 0 for the root node. The accumulated number of components is given by $\sum_{i=0}^{b} c^i$.

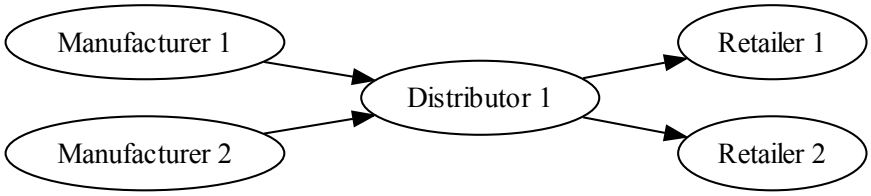

Figure 3.1: Supply chain graph.

Name	Symbol	Unit	Default value
Nodes	n	vertex	3
Avg. item records per node	r	item record	1
Avg. length	z	vertex	3
Avg. BoM depth	b	level	z
Avg. children per BoM level	c	node	2

Table 3.1: Supply chain parameters.

3.1.2 Example supply chains

To exercise the cost model there was a need to find example supply chains and define values for their parameters. A "short and broad'" chain and a "long and narrow" chain were considered, and are represented in Figure 3.2. The parameters that are not mentioned below retain the default values defined in Table 3.1.

Automotive: The generic *Auto* supply chain is "short and broad". It has 700 companies and the chain is 6 levels deep, with 3 components per level, on average.

$n = 700, z = 6, c = 3$

Pharmaceutical: The generic *Pharma* supply chain is "long and narrow". It has 4000 companies and the chain is 12 levels deep, with 2 components per level, on average.

$n = 4000, z = 12, c = 2$

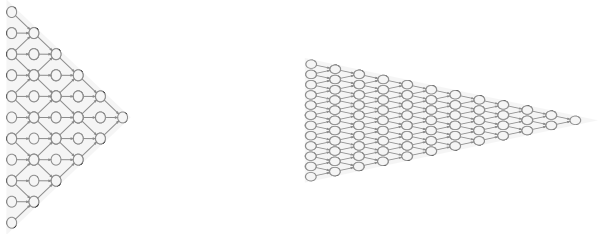

Figure 3.2: "Short and broad" (left) versus "long and narrow" (right) supply chain graphs.

3.1.3 Modeling the system

After the chain parameters were defined, the next challenge was to represent the target system with just enough detail.

The system parameters are presented in Table 3.2.

Name	Symbol	Unit	Default value
Bandwidth	β beta	bps	1 000 000 000
Processing speed	γ gamma	bps	1 000 000 000
Seek time	θ theta	s	0.001
Message size	μ mu	bit	100 000
Item record size	δ delta	bit	100 000

Table 3.2: System parameters.

The system default values roughly represent the capabilities of current technology: gigabit processing and gigabit networking.

The following simplifying assumptions were assumed. System parameters are the same for every node. All messages have the same size. All item records have the same size. Messages and received item records can be processed in main memory. All data stores are append-only. The time cost of accessing the data store to retrieve a record is independent of store size and independent of record size. The time cost of storing a record can be ignored, because it can be done asynchronously, using otherwise idle time.

3.1.4 Cost formulae

The time cost is measured in *seconds* and is denoted by C whereas the storage cost is measured in *bits* and is denoted by S.

The cost of processing a message (C_{MP}) is the message size divided by the processing speed.

$$C_{MP} = \frac{\mu}{\gamma}$$

The cost of transferring a message over the network (C_{MT}) is the message size divided by the bandwidth.

$$C_{MT} = \frac{\mu}{\beta}$$

The cost of a lookup (C_L) is a constant.

$$C_L = \theta$$

The cost of a one-way message (C_M) is the sum of the cost of sending, transferring, and receiving the message.

$$C_M = 2 \cdot C_{MP} + C_{MT}$$

The cost of a message exchange (C_{MX}) is the sum of the cost of the request and of the response. A data lookup cost can be added when response data has to be fetched.

$$C_{MX} = 2 \cdot C_M$$

The previous message cost definitions can be extended to include a payload of k item records. These are especially relevant when accumulate records are carried around. For a message exchange, k is the sum of item records in the request and response payloads.

$$C_{MP}(k) = \frac{\mu + k \cdot \delta}{\gamma}$$

$$C_{MT}(k) = \frac{\mu + k \cdot \delta}{\beta}$$

$$C_M(k) = 2 \cdot C_{MP}(k) + C_{MT}(k)$$

$$C_{MX}(k) = C_M(0) + C_M(k)$$

The storage cost of k item records ($S(k)$) is the multiplication of the item record size δ.

$$S(k) = k \cdot \delta$$

These basic cost formulae allow representing processing and communication costs in an abstract way. These basic formulae are composed together in the next Sections, assuming specific architectures.

3.2 Architectures

The model represents the four system categories presented in Section 2.5: Meta-Data Integration (MDI), Data Integration (DI), Unstructured Peer-to-Peer (UP2P), and Structure Peer-to-Peer (SP2P). The representative systems were used as source of details required for the modeling, in particular, to draw the presented UML (Unified Modeling Language) collaboration diagrams [Fowler, 2003] that provide a succinct representation of the data flows and allow the definition of the cost formulae. The developed cost model for each approach is presented in the next Sections.

3.2.1 Meta-data integration approach

In the MDI approach, the system is centralized to integrate meta-data about the location of data sources. This approach is exemplified by the *BRIDGE Directory* system. The cost of DS search is considered negligible because the result can be cached for a long time.

Capture cost: When a company receives a product, r EPC IS records are created locally. A single publication is sent to the DS. According to Figure 3.3, the time cost of an item's data

capture ($C_{mdi\,capture}$) is a message exchange with an item record for each company in the supply chain where the product is detected.

$$C_{mdi\,capture} = z \cdot C_{MX}(1)$$

The storage cost of an item's data capture ($S_{mdi\,capture}$) is the sum of the cost of EPC IS records with the cost of the DS publication.

$$S_{mdi\,capture} = z \cdot \big(S(r) + S(1)\big)$$

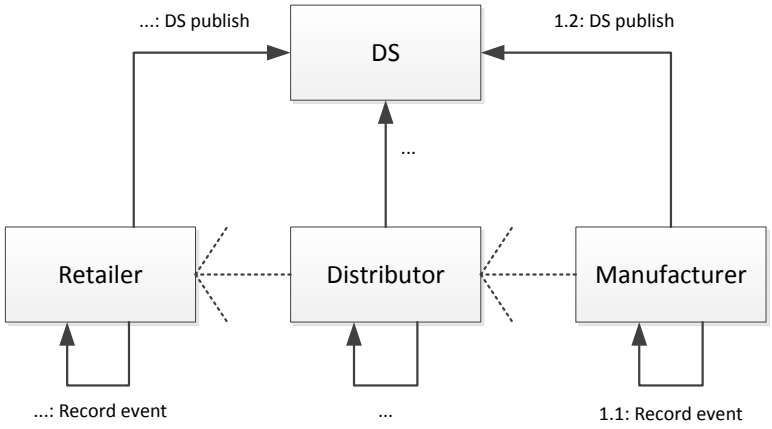

Figure 3.3: Meta-data integration capture.

Track query cost: To start, the DS search cache is contacted to locate the suitable DS instance. A query is issued to the DS and the IS location is returned. The asker contacts the IS with the most recent sighting of the object, as depicted in Figure 3.4. The time cost of a track query ($C_{mdi\,track}$) is the sum of the cost of querying the DS with the cost of querying the IS.

$$C_{mdi\,track} = 2 \cdot \big(C_{MX}(1) + C_L\big)$$

Trace query cost: A trace query is issued to the DS and a list of IS locations are returned. The asker contacts each IS for all records. The time cost of a trace query ($C_{mdi\,trace}$) is the sum of the cost of querying the DS with the cost of querying each IS.

$$C_{mdi\,trace} = \big(C_{MX}(z) + C_L\big) + z \cdot \big(C_{MX}(r) + C_L\big)$$

BoM query cost: A BoM query starts with a DS query to track the first observation of the product. Then, the asker contacts the IS for all aggregation records. The BoM query continues recursively The time cost of a BoM query ($C_{mdi\,BoM}$) is the sum of the cost of the DS track queries for each component in the BoM tree with the cost of the IS aggregation queries.

$$C_{mdi\,BoM} = \sum_{i=0}^{b} \Big(c^i \cdot \big(C_{MX}(1) + C_L\big)\Big) + \sum_{i=0}^{b} \Big(c^i \cdot \big(C_{MX}(c) + C_L\big)\Big)$$

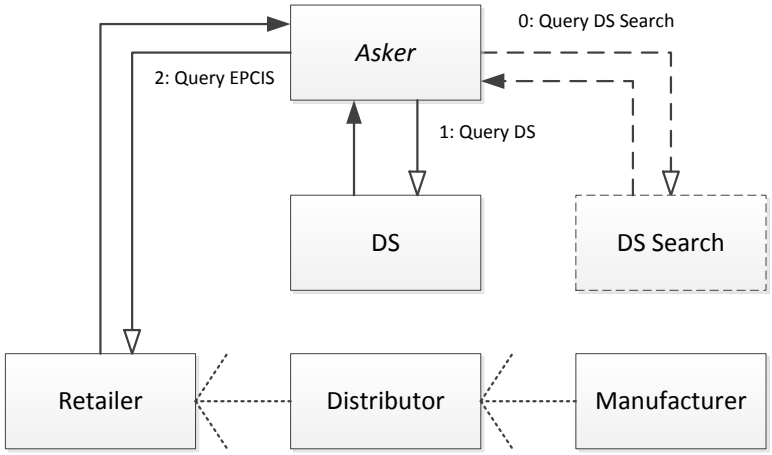

Figure 3.4: Meta-data integration track query.

3.2.2 Data integration approach

In the DI approach the system is centralized to integrate all data. This approach is exemplified by the *ID@URI* system.

Capture cost: The capture records are sent to the Manufacturer for storage. The time cost of an item's data capture ($C_{di\,capture}$), illustrated in Figure 3.5, is a message exchange with an item record for each company where the product passes outside of the Manufacturer.

$$C_{di\,capture} = (z-1) \cdot r \cdot C_{MX}(1)$$

The storage cost ($S_{di\,capture}$) is the sum of all item records stored at the Manufacturer.

$$S_{di\,capture} = z \cdot S(r)$$

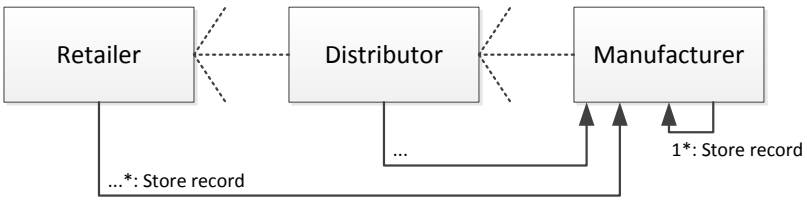

Figure 3.5: Data integration capture.

Track query cost: The time cost of a track query ($C_{di\,track}$), represented in Figure 3.6, is the cost of exchanging a message with the Manufacturer with a single result item record.

$$C_{di\,track} = C_{MX}(1) + C_L$$

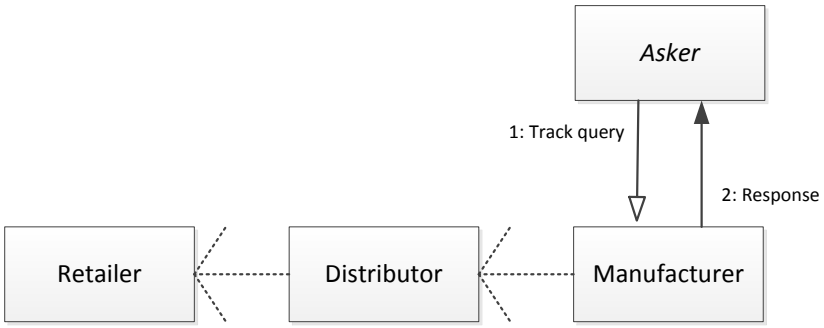

Figure 3.6: Data integration track query.

Trace query cost: The trace query is also sent directly to the Manufacturer. The time cost trace query ($C_{di\,trace}$) is the cost of exchanging a message with the Manufacturer with a list of result item records.

$$C_{di\,trace} = C_{MX}(z \cdot r) + C_L$$

BoM query cost: The BoM query is sent to the Manufacturer of each component. The time cost of a BoM query ($C_{di\,BoM}$) is the cost of exchanging a message with the Manufacturer of each component to get the child component list.

$$C_{di\,BoM} = \sum_{i=0}^{b} c^i \cdot \left(C_{MX}(c) + C_L\right)$$

3.2.3 Unstructured peer-to-peer approach

In the UP2P approach the system is decentralized and data is distributed across the capture nodes. This approach is exemplified by the *Theseos* system. The cost of determining the address of the next node was considered to be a single lookup.

Capture cost: Data is captured along the supply chain, as represented in Figure 3.7, and no communication is required, so there is no communication cost for an item's data capture ($C_{up2p\,capture}$).

$$C_{up2p\,capture} = 0$$

The storage cost of an item's data capture ($S_{up2p\,capture}$) is the sum of the cost of all item records stored along the chain.

$$S_{up2p\,capture} = z \cdot S(r)$$

Track query cost: The time cost of a track query is ($C_{up2p\,track}$), represented in Figure 3.8, the sum of the cost of propagating the query to the node where the item is with the cost of propagating back the response. There is a lookup at each node required for the forwarding.

$$C_{up2p\,track} = z \cdot \left(C_M + C_L\right) + z \cdot C_M(1)$$

Trace query cost: The time cost of a trace query is ($C_{up2p\,trace}$) the sum of the cost of

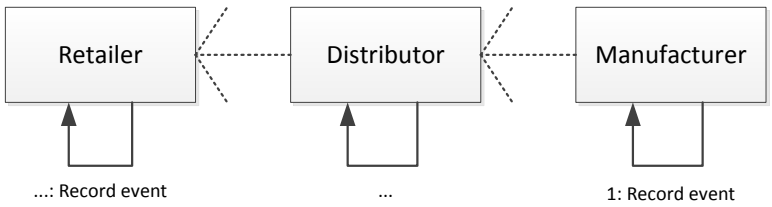

Figure 3.7: Unstructured P2P capture.

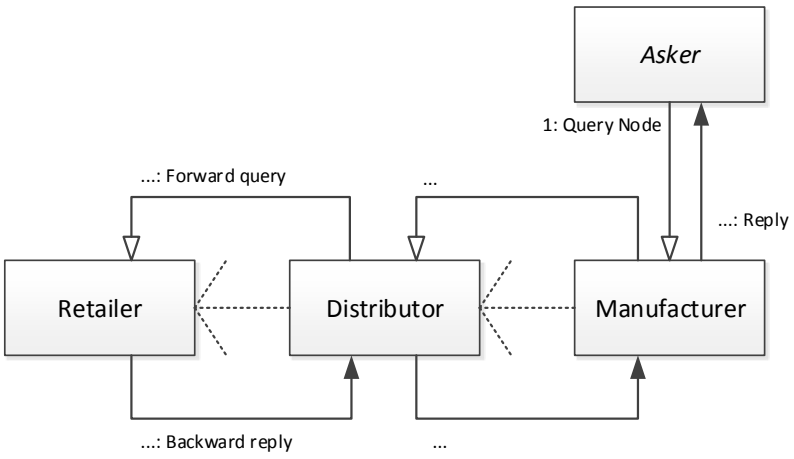

Figure 3.8: Unstructured P2P track query.

propagating the query to the node where the item is with the cost of propagating back the response with the accumulated trace records.

$$C_{up2p\,trace} = z \cdot \big(C_M + C_L\big) + \sum_{i=1}^{z} \big(C_M(i \cdot r)\big)$$

BoM query cost: The time cost of a BoM query ($C_{up2p\,BoM}$) is the sum of the cost of propagating the forward queries (one message for each component) with the cost of propagating back the response with the accumulated BoM tree. There is a great accumulation of records as the result goes through the nodes, back to the origin.

$$C_{up2p\,BoM} = \sum_{i=0}^{b} \big(c^i \cdot (C_M + C_L)\big) + \sum_{i=0}^{b} \Big(c^i \cdot C_M\big(\sum_{j=i}^{b} c^{(b-j)}\big)\Big)$$

3.2.4 Structured peer-to-peer approach

In the SP2P approach the system is decentralized and data is distributed according to a hashing algorithm to form a DHT (Distributed Hash Table). This approach is exemplified by the *OIDA* system. It is assumed that there is one DHT node for each supply chain node and that the hashing algorithm distributes the data evenly across the DHT nodes. The nodes join the DHT and never leave and the cost of joining the DHT is negligible. The item records are all kept on the same DHT node, indexed by a unique item identifier. The number of message hops to put or get a value is the logarithm of the number of DHT nodes

Capture cost: When a company receives a product, it contacts a known DHT node to store the data record. The hash algorithm determines the destination node and the destination is reached through a series of forwarding messages (hops). The time cost of data capture ($C_{sp2p\,capture}$), represented in Figure 3.9, is the sum of the cost of the message exchange with a DHT node with the cost of the message hops required to reach the node where data will be stored, and with the cost of the acknowledgement message.

$$C_{sp2p\,capture} = z \cdot r \cdot \Big(C_{MX}(1) + log(n) \cdot \big(C_M(1) + C_L\big) + C_M\Big)$$

The storage cost of an item's data capture ($S_{sp2p\,capture}$) is the sum of the cost of all item records stored in a single DHT node plus the hop pointers.

$$S_{sp2p\,capture} = z \cdot \Big(S(r) + S\big(log(n)\big)\Big)$$

Track query cost: When a track query is issued, a known DHT node is contacted to retrieve the data record. The hash algorithm determines the location node and it is reached through a series of forwarding messages (hops). Figure 3.10 illustrates the query processing. The time cost of a track query ($C_{sp2p\,track}$) is the sum of the cost of the message exchange with a DHT node with the cost of the message hops required to reach the node where data is stored with the cost of the response message containing the single item record.

$$C_{sp2p\,track} = C_{MX}(1) + log(n) \cdot \big(C_M + C_L\big) + C_M(1)$$

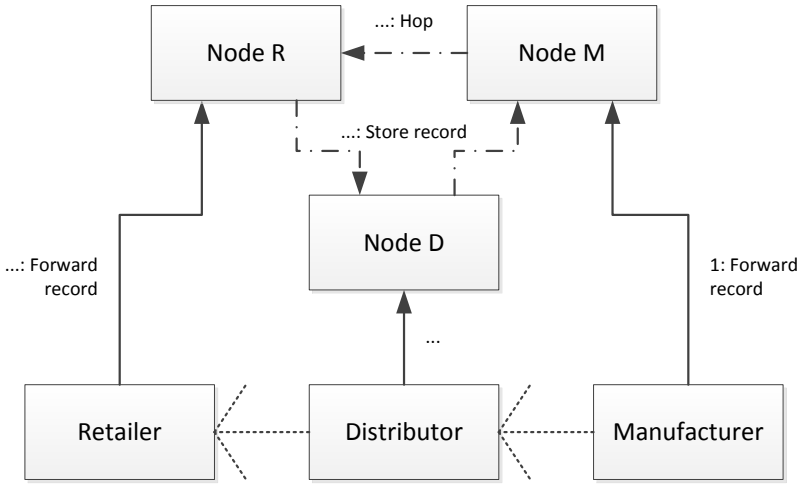

Figure 3.9: Structured P2P capture.

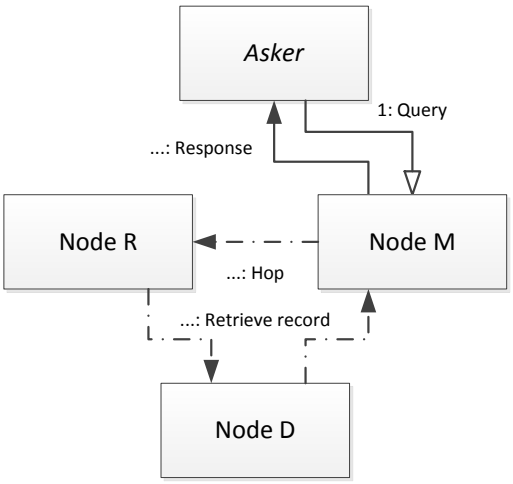

Figure 3.10: Structured P2P track query.

Trace query cost: The time cost of a trace query ($C_{sp2p\,trace}$) is the sum of the cost of the message exchange with a DHT node with the cost of the message hops required to reach the node where data is stored, with the cost of the response message containing the item records.

$$C_{sp2p\,trace} = C_{MX}(z \cdot r) + log(n) \cdot \left(C_M + C_L\right) + C_M(z \cdot r)$$

BoM query cost: The time cost of a BoM query ($C_{sp2p\,BoM}$) is the cost of exchanging a message with the DHT for each component to get the child component list.

$$C_{sp2p\,BoM} = \sum_{i=0}^{b} \left(c^i \cdot \left(C_{MX}(c) + log(n) \cdot (C_M + C_L) + C_M(c)\right)\right)$$

3.3 Comparing traceability systems

The cost model was used to compute estimates and generate plots for two distinct chains, described earlier in Section 3.1.2: "short and broad" (*Auto*), and, "long and narrow" (*Pharma*). These supply chains have distinct parameter values that are likely to contrast the estimates for each architecture.

Figures 3.11, 3.12, 3.13, and 3.14 present the plots. Each Figure has one line plotted for each architecture. The top plot is always the *Auto* supply chain and the bottom plot is always the *Pharma* supply chain. The vertical axes show the time cost and the horizontal axes show the number of items being considered.

To put the absolute time values in perspective, a "work day" scale is used, where 8 hours correspond to 28 800 seconds. An operation that takes more than 1% of that time – 288 seconds, 4.8 minutes – is considered as too costly for practical purposes. This limit is arbitrary but defines a threshold grounded on a business suitable time scale.

The data capture results are presented in Figure 3.11. UP2P has the best possible performance, because data is captured and stored locally and no network communication is required. DI and MDI have very good performance because they capture the data in one or two steps, respectively. SP2P has the worst performance because of the DHT hops. The cost is sensitive to the chain length parameter, so the costs are greater in the longer *Pharma* chain.

Figure 3.12 presents the results for the track query. DI yields the best performance, since the query is answered in a single request. MDI comes second-best because there is one additional level. The UP2P and SP2P approaches are the most expensive but are still practical because they are well below the defined threshold. Comparing the two supply chains, DI and MDI cost the same because they do not change with the length. The same cannot be said for UP2P and SP2P because, the longer the chain, the costlier the query.

Figure 3.13 shows the results for the trace query. The trace query is more expensive than the corresponding track query as expected, since the track answer is a small subset of the trace response. DI and MDI have comparable performance, with low response times, because they benefit from having data and meta-data concentrated. SP2P and UP2P have the worst performance that degrades with longer chains but the times are still within the practical threshold.

Finally, Figure 3.14 presents the BoM query results. DI is the best approach to return BoM results, since all data is concentrated. MDI and UP2P have comparable performance, and SP2P is the worst. Supply chain length has a very significant impact. Here, the *Pharma* results are approximately 100 times more costly than the *Auto*. Worst still, BoM results are above the practical time threshold, making the computation of a complete BoM from scratch impractical. This means that the BoM data should be stored incrementally, during the product life-cycle.

The DI approach has the *best overall* performance. The MDI approach is *second-best* and provides an additional indirection level that can be used to address security concerns. In this sense, MDI and DI are the most suited to being services provided by third parties.

The UP2P is the approach that allows the most control of data ownership, so it might appeal to some companies. However, the follow-the-chain approach cannot compensate for broken links and this makes it impractical for law abiding scenarios.

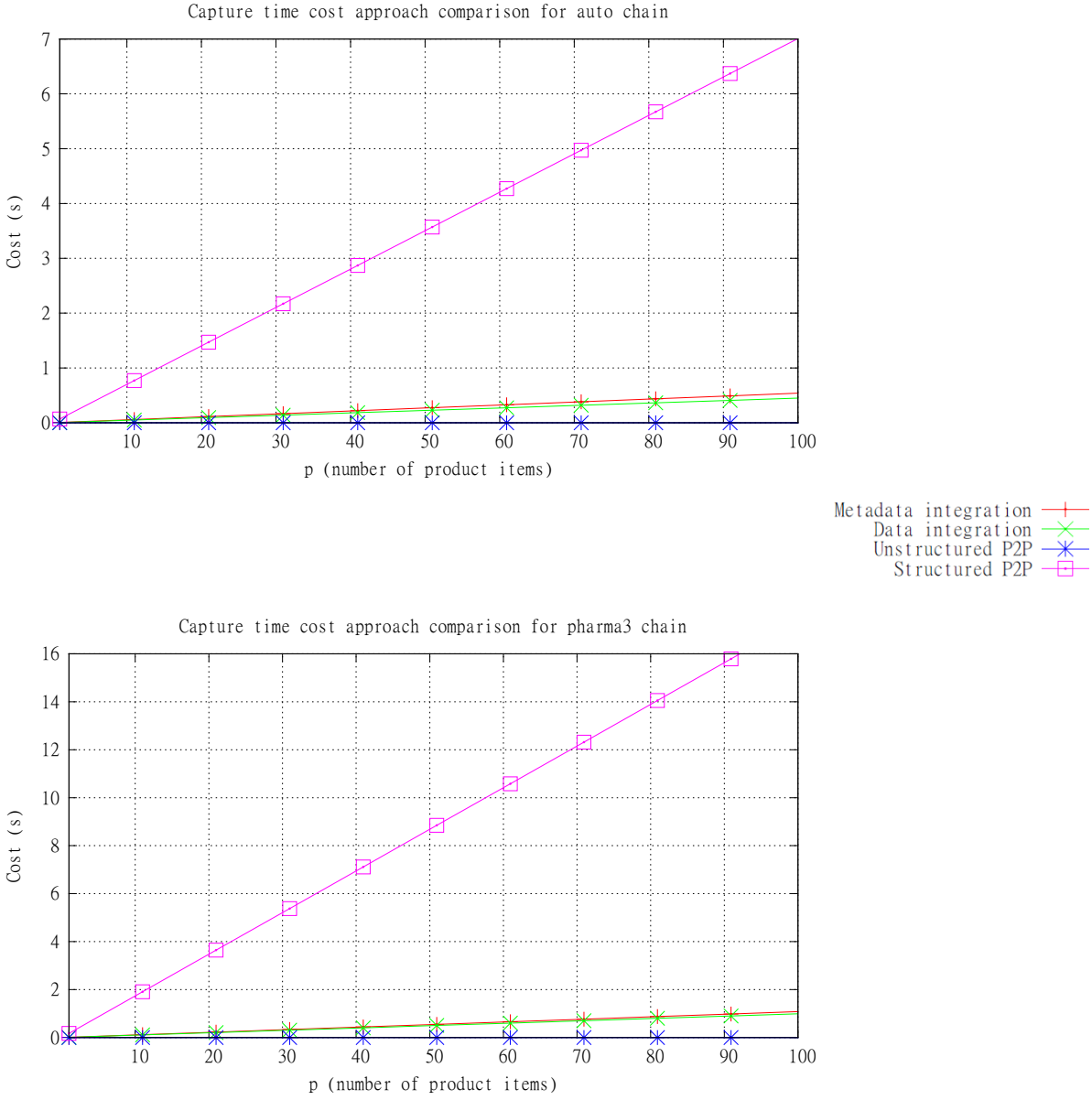

Figure 3.11: Estimated cost for data capture.

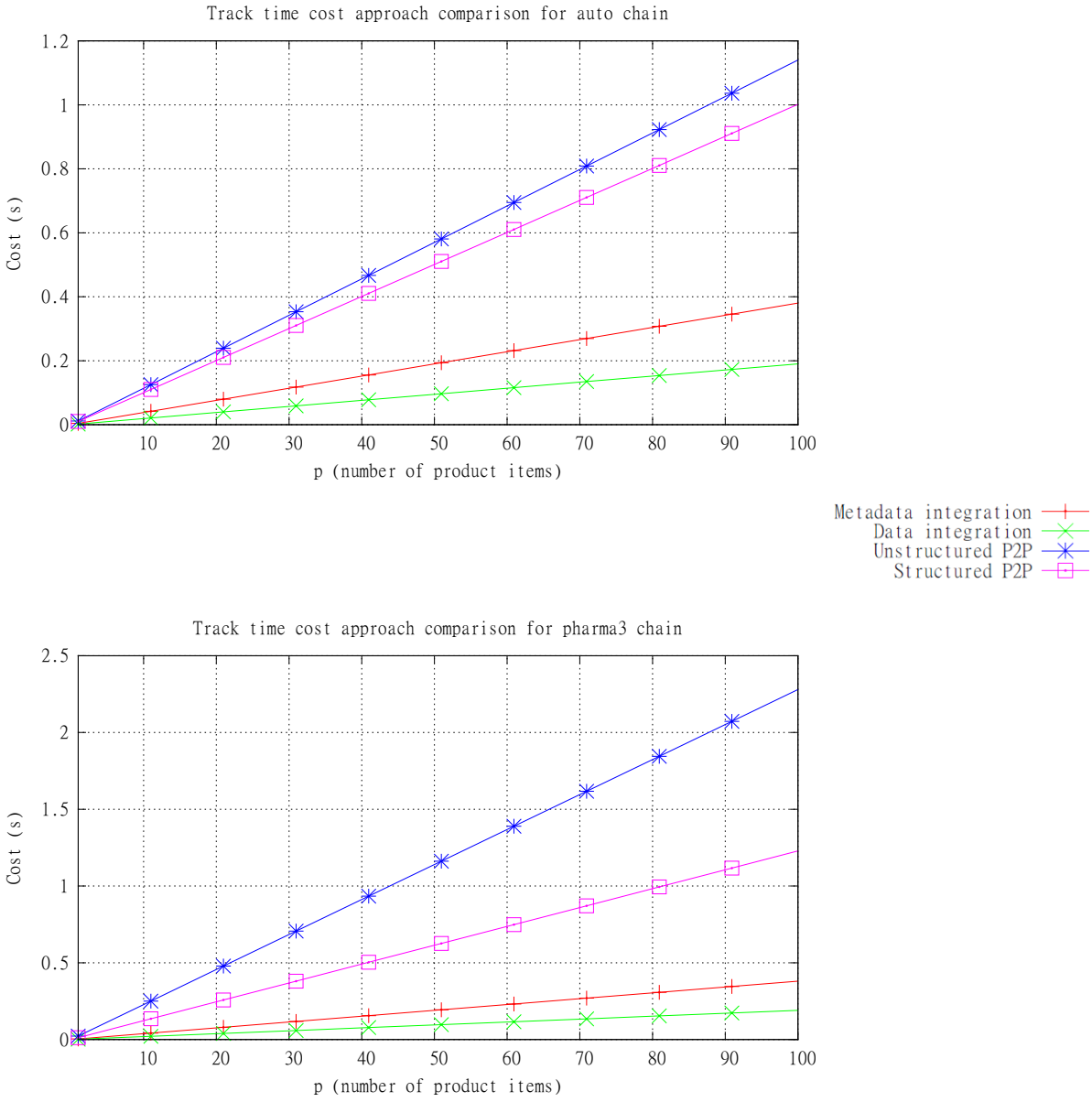

Figure 3.12: Estimated cost for track query.

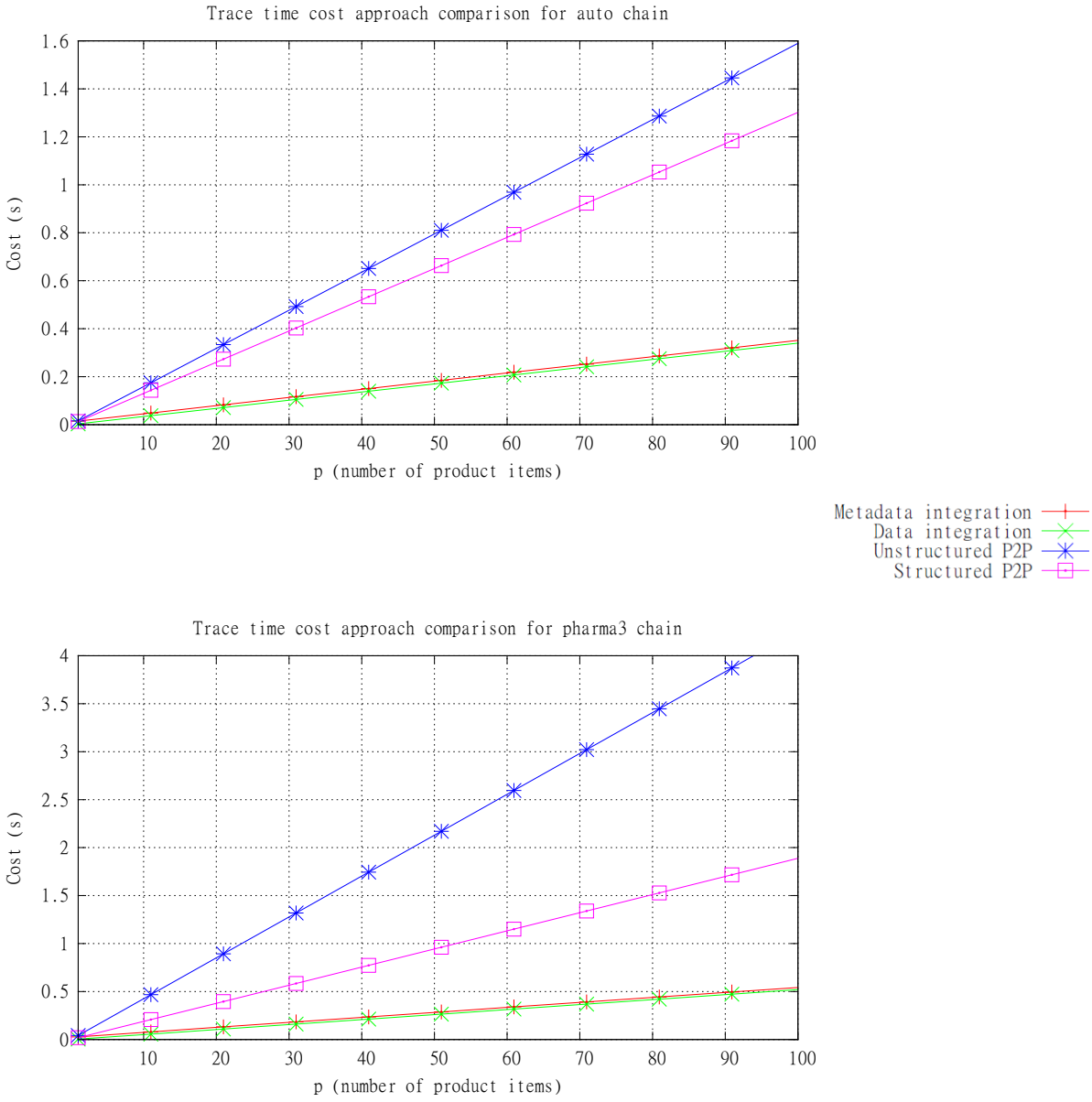

Figure 3.13: Estimated cost for trace query.

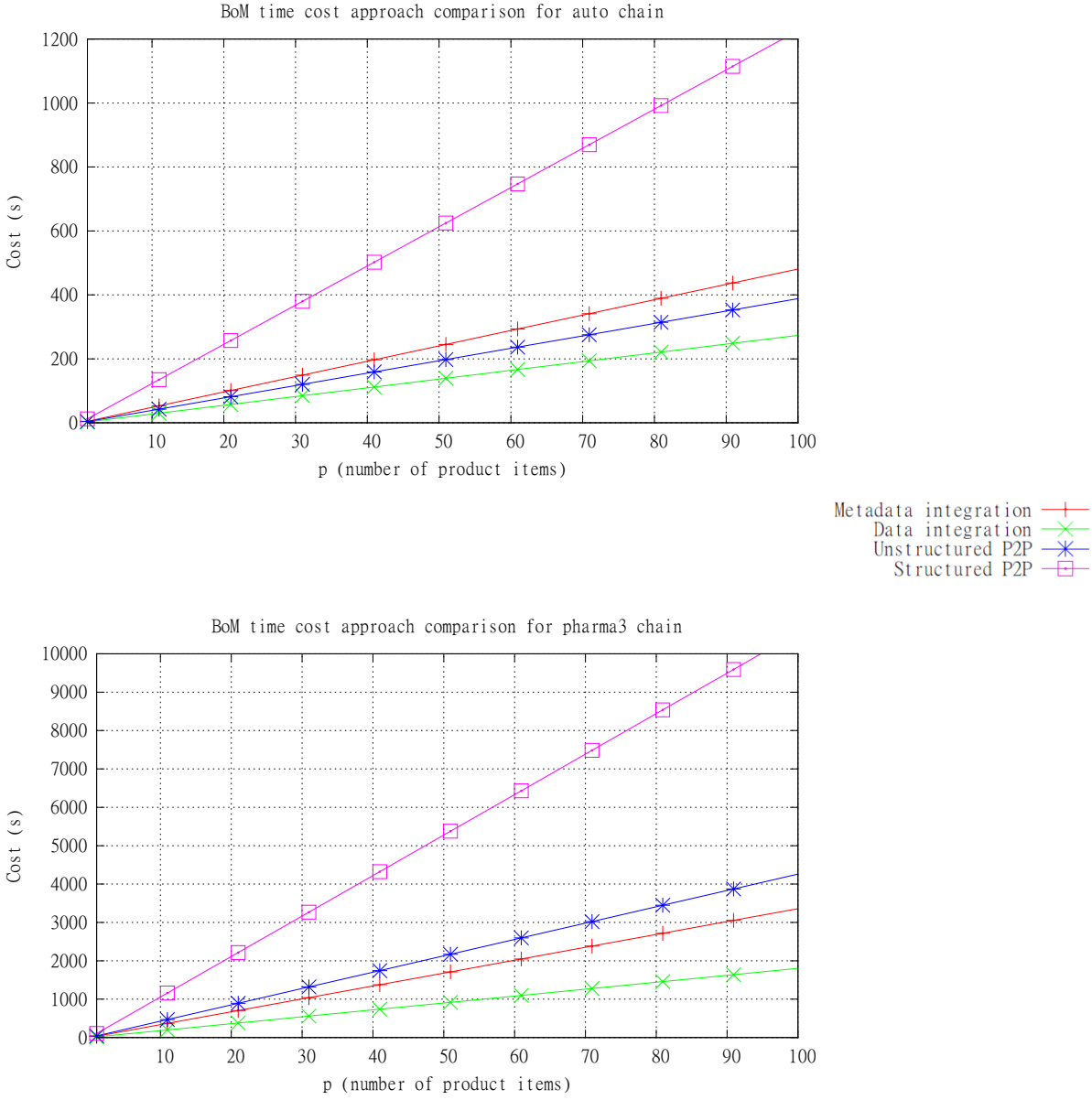

Figure 3.14: Estimated cost for BoM query.

The SP2P approach is always costlier than DI and MDI, and is only slightly better than UP2P on track and trace. Because of the use of DHT technology, SP2P has the most potential to scale, but on the downside, it provides higher response latency times. Also, in SP2P, data is scattered by a hashing algorithm and quickly goes out of control of the data owner.

Overall, the UP2P and SP2P approaches are more expensive for all kinds of query.

3.4 Conclusion

The cost model outputs were evaluated using parameters from *Auto* and *Pharma* supply chains. The specific supply chain domain being addressed is relevant to the choice of system architecture because the layout and the product features change the parameters that, in turn, significantly affect the estimated costs. The supply chain length z, in particular, is a very influential parameter. This was visible, for instance, in Figure 3.14 where the ordering of approaches changed for a different supply chain. This shows that the best traceability solution depends on the specific case being considered. Given this fact, an assessment tool is useful for designing traceability systems.

For the continuation of the dissertation only track and trace queries are considered. BoM queries are dropped because it has been shown that they need a special-purpose system to achieve a practical performance level.

The developed cost model can quantitatively compare traceability system architectures. Each architecture was modeled using cost formulae. The absolute cost values produced by the model correspond to idealized machine costs and cannot be easily translated to actual system cost but they do allow architectures to be compared. The cost measure is important to realize if the traceability system is worth being built. However, data visibility restriction is very important and is not being addressed by this model.

4 Assessing Visibility

A *traceability data set* contains events owned by a company about a specific item and it is an asset that requires protection against unauthorized access attempts. Protecting data requires policies that have to be defined and enforced. This adds overheads to the traceability system. This Chapter presents an assessment of visibility restriction approaches capable of handling *emergent object paths* i.e. there is no prior knowledge about who should be authorized to access the data because the path that each physical object will take in the supply chain is not known in advance. The approaches are modeled using an *extended cost calculator* that uses more details of the target system to estimate the additional cost of restricting visibility in a traceability system.

The comparison of traceability system architectures made in Chapter 3 showed that the MDI architecture had the second-best overall performance and provided an additional indirection level that can be used to address security concerns. Since visibility is a critical concern, the MDI approach was chosen as the reference architecture for modeling the visibility restriction approaches.

The EPC DS + IS combination is a concrete implementation of the MDI architecture, represented in Figure 4.1. The DS stores meta-data about the repositories containing data and IS repositories store the detailed data, generated by the objects tagged with EPC codes, as they move along the supply chain. The architecture is *semi-centralized* – DS instances are few compared to IS instances and play a "special" role – and traceability data is *referenced* (not copied).

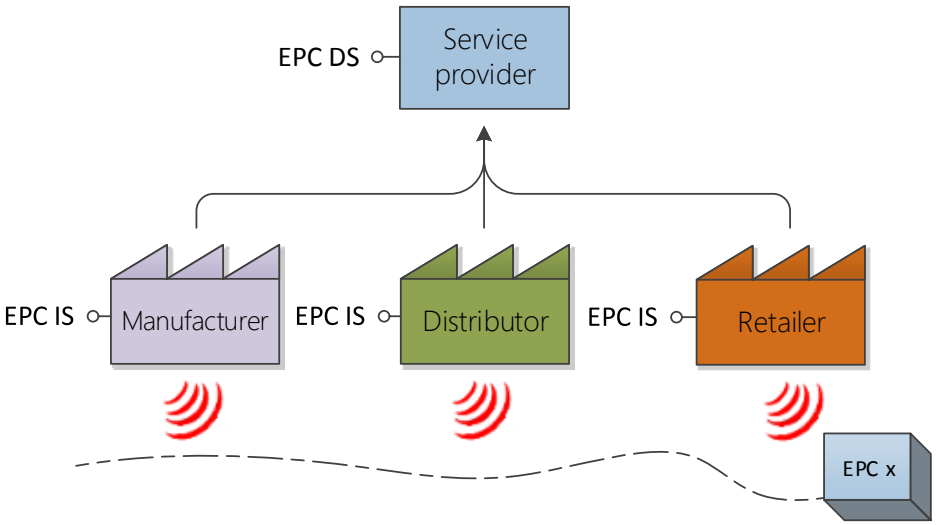

Figure 4.1: MDI architecture with EPC DS and IS.

4.1 Visibility restriction approaches

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

There is a canonical data structure for specifying access control: 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"*. There can be several approaches that specify access control. The following approaches were considered for visibility restriction of traceability data sets:

- Enumerated Access Control (EAC);
- Chain-of-Communication Tokens (CCT);
- Chain-of-Trust Assertions (CTA).

The first two correspond to the classical approaches to access control: Access Control Lists and Capabilities [Sandhu and Samarati, 1994], respectively, while the last one corresponds to an assertion-based formulation that is intended to be more expressive.

4.1.1 Enumerated Access Control

One natural way to control access to a resource is to have a list of parties that can access the data. In this case, each traceability data set is protected by an Access Control List (ACL). An ACL keeps the access rights indexed by the object identifier, and is represented in Figure 4.2. It has an owner and several entries that define authorized user-action pairs. Figure 4.3 shows the available ACL operations.

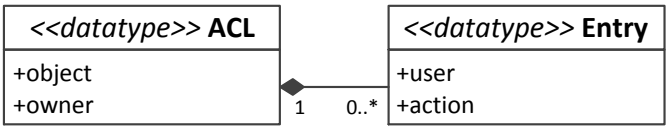

Figure 4.2: Access control list data structure.

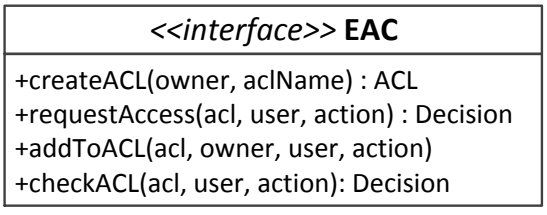

Figure 4.3: EAC interface operations.

4.1.2 Chain-of-Communication Tokens

Chain-of-Communication Tokens (CCT) is an adapted Capability mechanism. CCT represents access rights within object references called tokens. Figure 4.4 presents the contents of a token: there is a public identifier and a secret that is used as a 'password'. The remaining data – owner, resource, and action – identify the purpose of the token. A token can be read as: "the *owner* grants the right to perform the *action* on the *resource* to the token holder". A token protects a traceability data set. Figure 4.5 shows the token operations.

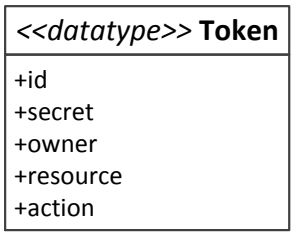

Figure 4.4: Authorization token data structure.

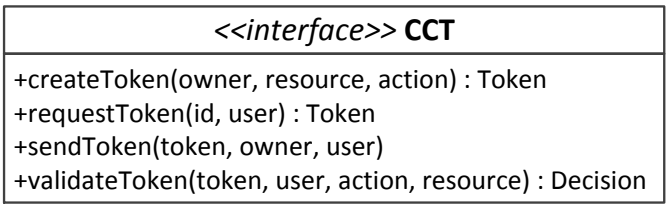

Figure 4.5: CCT interface operations.

This approach was proposed by Ilic et al. [2007] to leverage existing organizational relationships to provide access based on physical possession. When a token is shared, the access rights are also shared. This way, access rights can 'follow the chain' if the shipping company sends the token to the destination along with the physical object. To create new visibility scopes, new tokens can be created and used at each node in the object path. A single token can be used along the chain to protect all of the traceability records, if a global scope is desired for the data.

4.1.3 Chain-of-Trust Assertions

Chain-of-Trust Assertions (CTA) represents a potentially more expressive mechanism that expresses the access rights using logical statements. A system is said to be *extensible* if new features can be added to it with limited effort and existing features are preserved [Bass et al., 2003]. Assertions allow extensibility because more assertions can be defined and add new meanings.

Figure 4.6 shows an assertion represented as text. It contains the assertion name 'trust' and its arguments: the data *owner* identification, the *action*, the *resource* identifier, and, finally, the *partner* identification. When information is requested, the logical formulae are evaluated to make an access decision. Figure 4.7 shows the assertion operations.

```
trust(owner, action, resource, partner)
```

Figure 4.6: Textual representation of assertion.

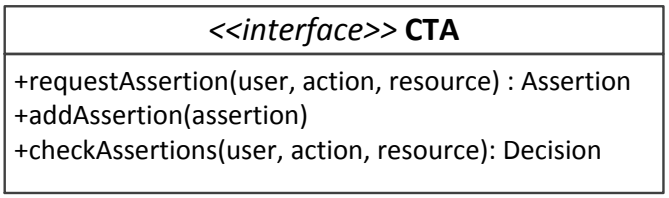

Figure 4.7: CTA interface operations.

Each assertion is a formal representation of a condition and is a building block of a policy. The idea to use assertions was influenced by the DS proposal by Traub and Sarma [2007]. This proposal was implemented early in the research and showed that assertions were a practical approach. The DS prototype is described in Appendix E.

4.2 Cost calculator

A specialized version of the analytic cost estimation model – the cost calculation board – was developed to compare the visibility restriction approaches [Pardal et al., 2012]. While the previous model used average values as parameters, this one can take actual descriptions, because individual messages are represented and accounted for.

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 4.8.

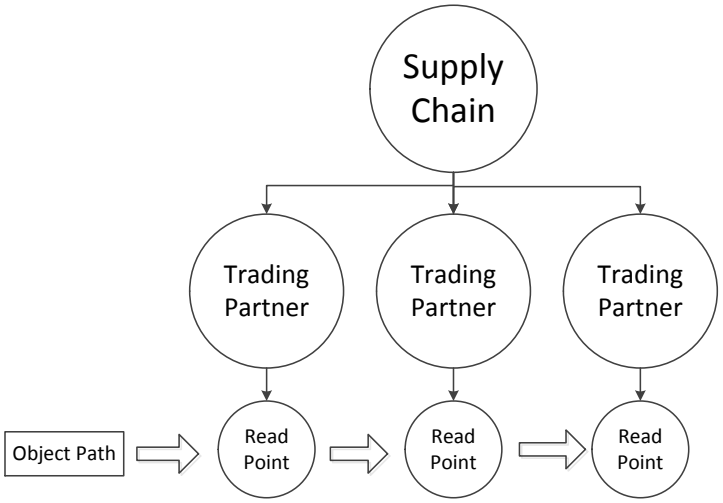

Figure 4.8: Supply chain scenario.

A target system is modeled for cost estimation by identifying its operations. Each operation has input and output and these data structures are modeled from implementation specifics

of the Fosstrak EPC IS and BRIDGE DS design. For the assessment, each trading partner is considered to have a single IS instance and there is a single DS instance shared by all the trading partners in the supply chain.

The cost calculation uses a *black-board* data structure [Beigl et al., 2007], represented in Figure 4.9, to implement 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. Board contributors can easily be enabled or disabled to test different conditions e.g. add/remove overheads.

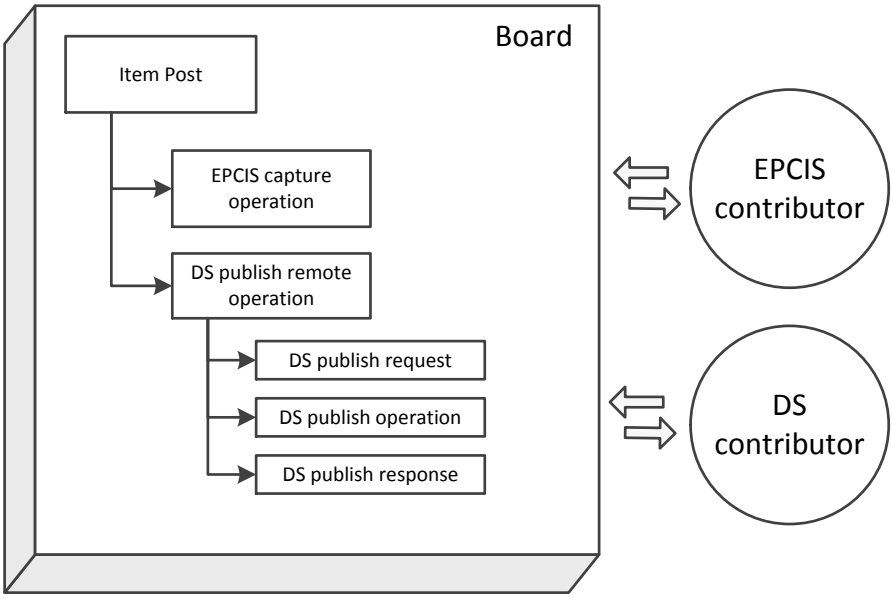

Figure 4.9: Cost computation board.

A board post represents a cost parcel. There are three types of cost: *storage* (data size), *processing* (time) and *networking* (time). There are board contributors that observe the posts that are placed on the board, and that can add more posts. 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. Not all posts have direct cost, some are merely place-holder posts that signal the need for additional expansion (Section 4.2.1 presents a concrete example of this technique).

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 4.10. 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 analysis and plotting tools.

4.2.1 Modeling the visibility restriction approaches

A meta-model for visibility restriction defines board posts corresponding to *initialize* (InitShare), *request* access (RequestShare), *share* (Share), and *enforce* access (EnforceShare) to

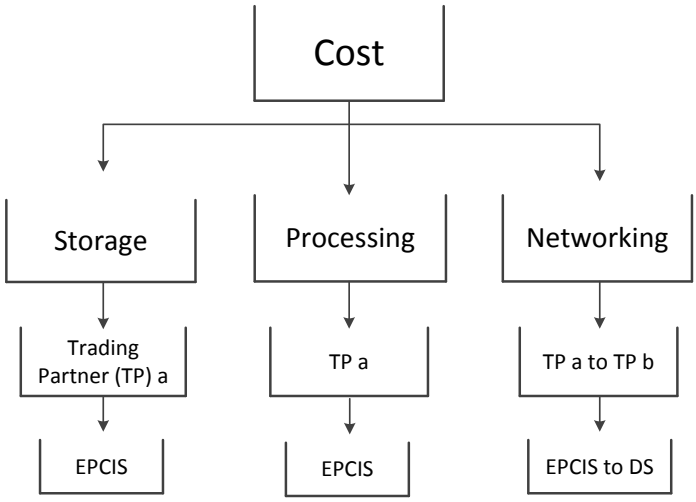

Figure 4.10: Cost 'buckets'.

physical object information. Each visibility restriction approach – EAC, CCT, and CTA – defines its own effects for these place-holder posts, listed in Table 4.1. The operations are presented in `object.methodCall()` format to show where they are executed: 'is' operations are local to a company repository while the 'ds', 'owner', and 'partner' operations require remote communication.

Post	**EAC**	**CCT**	**CTA**
initShare	`ds.createACL()` `is.createACL()`	`ds.createToken()` `is.storeToken()`	
requestShare	`ds.reqAccess()` `owner.decideReq()`	`ds.reqToken()` `owner.decideReq()`	`ds.reqAssertion()` `owner.decideReq()`
share	`ds.addToACL()` `is.addToACL()`	`partner.sendToken()`	`ds.addAssertion()` `is.addAssertion()`
enforceShare	`is.checkACL()`	`is.validateToken()`	`is.checkAssertions()`

Table 4.1: Comparison of visibility restriction implementations.

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 and the enforcement checks 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 and the enforcement confirms 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 and the enforcement verifies that the existing assertions grant access.

4.3 Comparing visibility restriction approaches

The evaluation baseline presented the system without visibility restriction costs. The approaches defined in Section 4.1 are considered in the assessment presented next.

4.3.1 Baseline

The supply chain length z was found, in Section 3.3, to be one of the most influential parameters. For the following assessment, three generic supply chains variants were considered: a short chain with average length of 3 companies; medium chain with 6 companies; and long chain with 12 companies. Each chain is linear meaning that there are no path branches/forks.

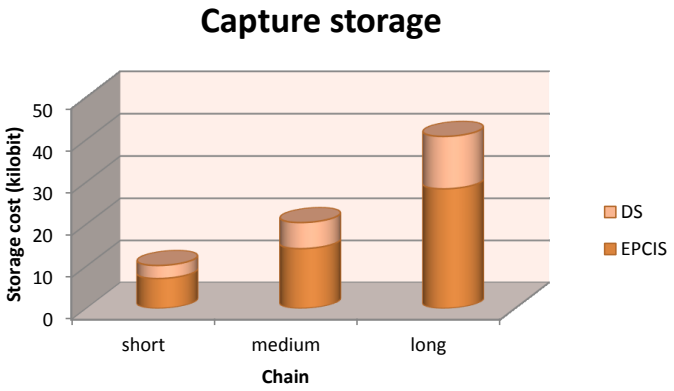

Figure 4.11: Storage cost baseline.

Figure 4.11 shows the *storage costs* for the only operation with storage costs: the data capture, storing IS events and DS records. The cost of audit logs is not being considered. The storage cost grows linearly with the chain length, as expected.

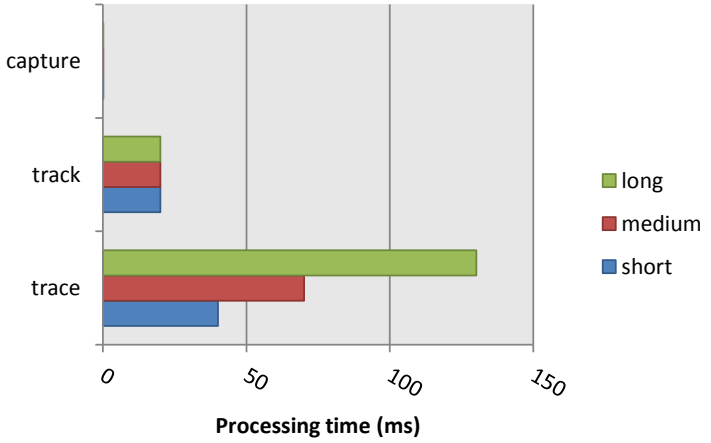

Figure 4.12: Processing cost baseline.

Figure 4.12 shows the *processing time cost* for each operation: data capture, track query, and trace query. A typical value of 10 ms was assumed for data seek[1].

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 IS with the most recent record is contacted after the DS query.

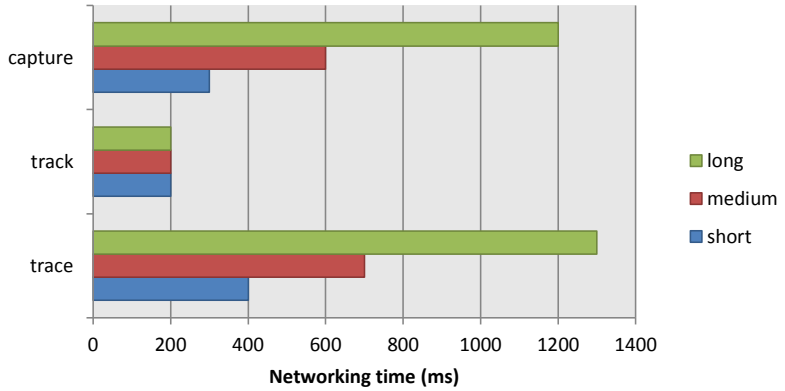

Figure 4.13: Networking cost baseline.

Figure 4.13 shows the *networking time cost*. A typical value of 100 ms latency for round-trip was assumed[2]. 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 IS query). The time cost of networking is dominated by the latency because messages are small in size.

Since the track queries cost do not depend on supply chain length, the remaining assessment is done considering only trace queries.

4.3.2 Overheads for XML and TLS/SSL

The baseline considers that data structures are binary using basic integer and string types. However, there are interoperability advantages in using Web Services [Alonso et al., 2004] that encode the data in XML (eXtensible Markup Language). The XML overhead was modeled using values measured by Juric et al. [2006] that state 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 TLS/SSL (Transport Layer Security/Secure Sockets Layer) [Dierks and Rescorla, 2008]. Again, according to Juric et al. [2006], SSL makes messages 1.08 times larger and response time 1.40 times longer. Despite these overhead values, the increments are small when compared with the latency values[3]. For this reason, the use of both XML and SSL can be considered practical.

[1] This value is bounded by typical average access times to secondary memory devices, like hard disks.

[2] It corresponds to the average of 'ping' response times from servers of the universities across the world that have Auto-ID Labs.

[3] The plots of the processing and networking cost with XML and SSL are not presented because the differences

4.3.3 Visibility restriction results

By default, companies decide to authorize information sharing 'on-demand' i.e. a request is made when someone needs access to the information.

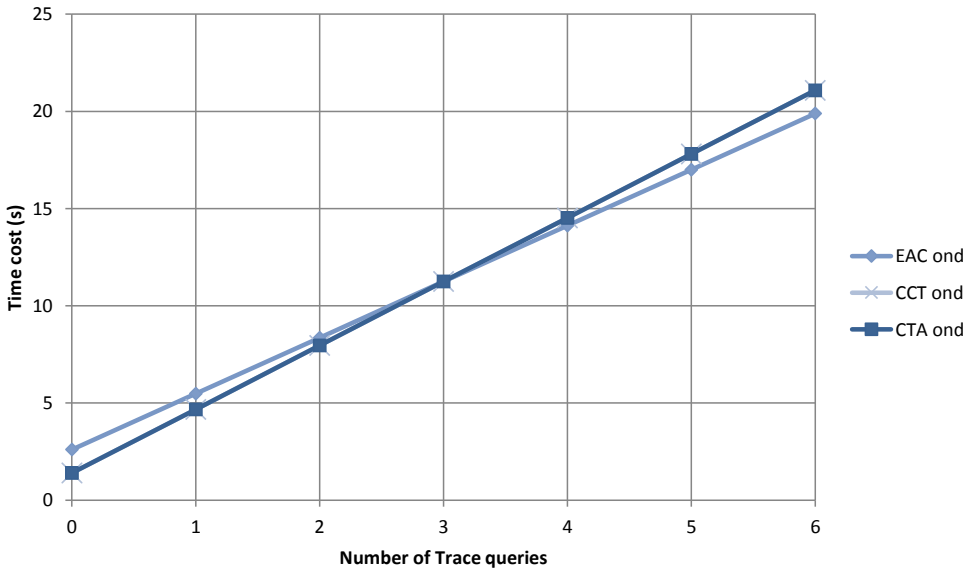

Figure 4.14: Visibility approach comparison for 'on demand sharing'.

For comparing EAC, CCT, and CTA the prediction results for the chains were averaged. 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). Figure 4.14 presents the comparison results. The differences are not significant. CTA and CCT have the best performance up to three queries per object, then EAC is better. CTA, CCT are slightly better. But the differences to EAC are very small.

4.3.4 Comparing with up-front data sharing

With 'on demand sharing', presented in the previous Section, the sharing effort at capture time is reduced because future queries for all objects are considered less likely. The flip-side is that additional requests and decisions to share information are required and have to be mediated by DS, creating an additional burden for it.

An alternative option was evaluated, called 'upfront sharing'. In this case, as soon as the trading partner sends the object, it also issues an authorization that may or may not be actually necessary. The idea is to save time by anticipating future data uses. For each individual object, the data owner grants access to its immediate upstream and downstream partners. The query is always issued by the last company in the object's path.

With 'upfront sharing' the trading partners are pro-active regarding the sharing operations because future queries are considered likely for all objects.

---

are not visible.

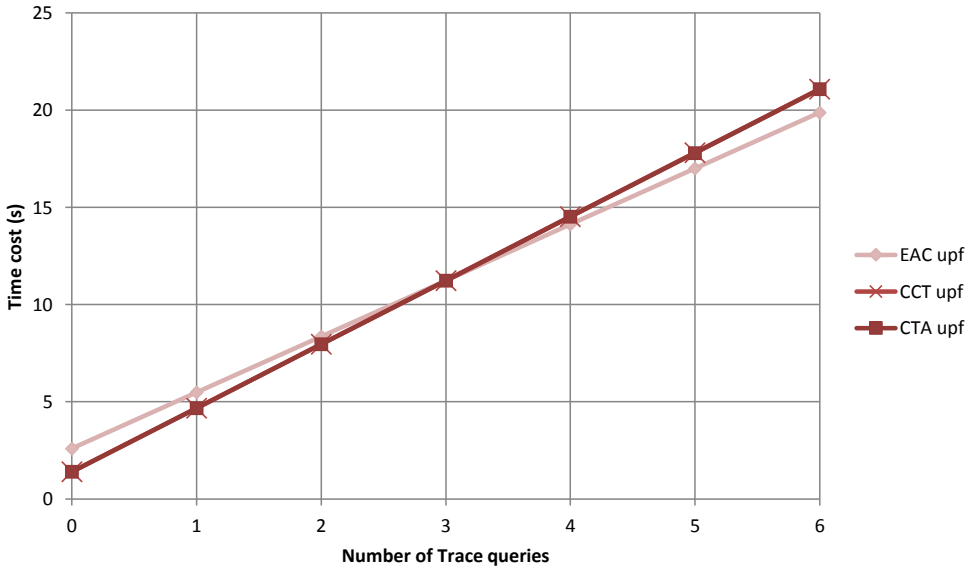

Figure 4.15: Visibility approach comparison for 'upfront sharing'.

Figure 4.15 presents the comparison results for 'upfront' (abbreviated 'upf') EAC, CCT, and CTA. CTA and CCT have the best performance up to three queries per object, then EAC is better, but the difference is not significant. The results are nearly identical to 'on-demand' sharing.

For Figure 4.16 the best 'upf' and the best 'ond' were picked (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 time cost.

To compare 'on demand' with 'upfront' the break-even point is 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. 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.

4.4 Conclusion

In this Chapter the visibility restriction approaches were defined and assessed. A specialized cost model was developed using more implementation details from EPC DS and IS. However, the results show no significant cost differences. Overall, the estimates were found to be too similar. This lack of resolution in predicted results means that the specialized cost model may be insufficient for the assessment, and an actual implementation is required to properly assess the visibility restriction approaches.

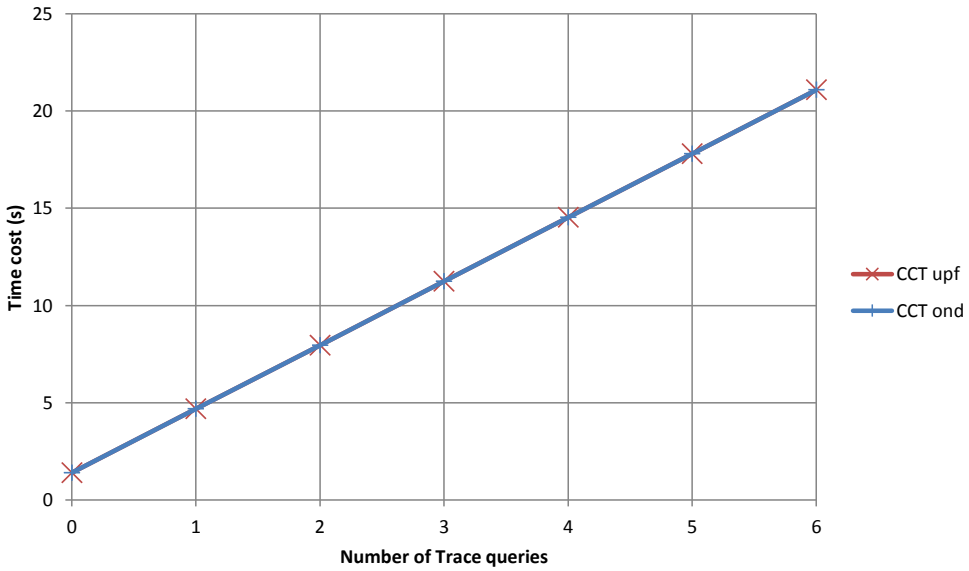

Figure 4.16: Comparison between CCT 'upfront' and CCT 'on demand'.

5 Implementing Visibility

The EAC, CCT, and CTA visibility restriction approaches presented and assessed in the previous Chapter appear to be equally capable of achieving the desired access control, but the actual performance remains to be evaluated. This Chapter describes the visibility restriction implementations done on the Java[1] programming platform and the measurements of their execution performance.

Like in the previous Chapter, the visibility restriction implementation is applied to the MDI architecture. The policies are authored by the data owners and used both at EPC DS and IS, as depicted in Figure 5.1.

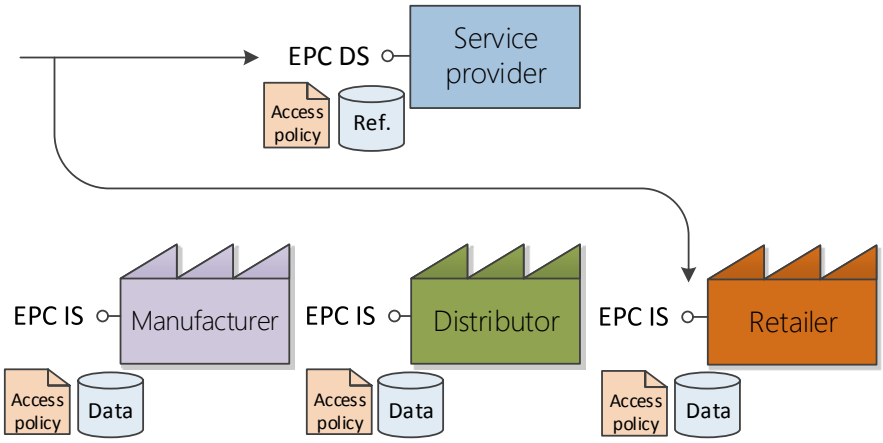

Figure 5.1: Authorization policies protect both EPC DS and IS.

5.1 Supply chain authorization implementations

The Supply Chain Authorization (SCAz) Application Programming Interface (API) allows companies participating in a supply chain to express authorizations using concepts such as item and trading partner [Pardal et al., 2012a] [Pardal et al., 2012b]. Figure 5.2 presents the SCAz operations: *init* for initializing, *share* for granting access, *request* to ask for access, and *enforce* to verify permission.

An alternative implementation of SCAz was developed for each one of the visibility restrictions presented in Section 4.1: EAC, CCT, and CTA. Figure 5.3 shows how the classes

[1] http://www.java.com

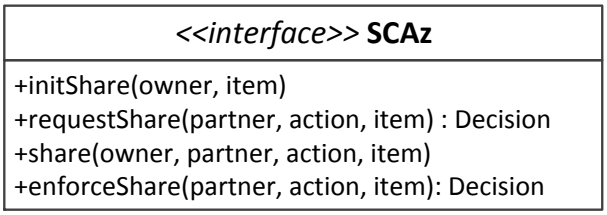

Figure 5.2: SCAz interface operations.

relate to the interfaces: each class implements a mechanism-specific API and implements the SCAz interface, allowing the same authorization needs to be mapped transparently to distinct implementations. As long as the SCAz interface is used, the implementations are interchangeable.

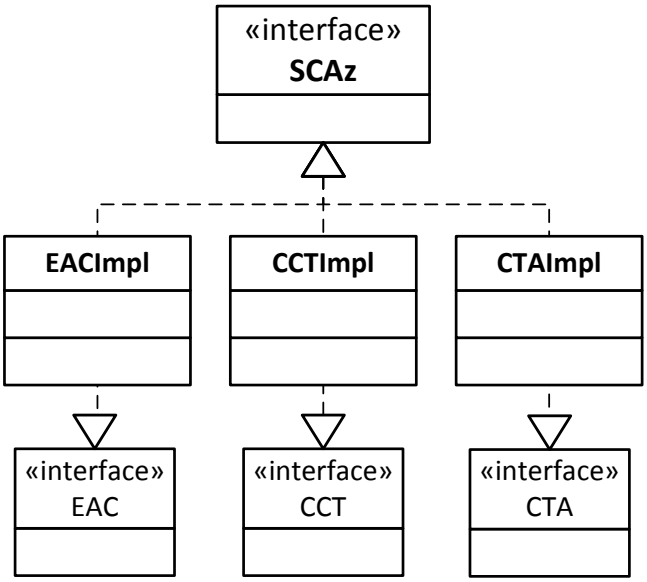

Figure 5.3: SCAz, EAC, CCT, and CTA interfaces and classes.

SCAz is used to capture authorizations in a way that is understandable by supply chain users and then it is translated to XACML so that it can be deployed and enforced in a standard infrastructure, with audit trail that can increase the overall trust in the system. XACML is described in detail in Appendix F.

5.1.1 EAC implementation

The EAC implementation uses an Access Control List (ACL) library available on the Java virtual machine in the package `sun.security.acl`. Each traceability data set – *the events owned by a company C about an item i* – is protected by an ACL.

The master ACL is maintained at the DS. A local copy is also maintained at each IS, to also protect its records. The data owner contacts DS to add new partners to the ACL and the changes are propagated to the IS. For audit purposes, the changes to the list should be logged to allow reconstruction of list state at any point in time.

5.1.2 CCT implementation

The CCT implementation uses custom-built Java code to represent tokens. The token identifier is a `java.util.UUID` string and the secret is composed by 128 bits generated with `java.security.SecureRandom`. A token protects a traceability data set.

The token is propagated along the chain by electronic communication e.g. within an Advance Shipping Notice (ASN) message or embedded in a special purpose RFID tag. 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 EPC IS. The data owner sends the token directly or via DS to its partners to authorize them. For audit purposes, the presented token values at the times of publishing and querying should be logged.

5.1.3 CTA implementation

CTA expresses access rights of traceability data sets as logical statements, called *assertions*, that are issued by the data owners. The CTA implementation uses the Apache Jena RDF library[2] to represent assertions as RDF (Resource Description Framework) statements [Manola and Miller, 2004]. RDF and other Linked Data technologies are presented in Appendix G.

Figures 5.4 and 5.5 show a simple CTA policy stated in RDF classes and properties, expressed as subject-predicate-object tuples. In the example, 'policy0' created by 'company0' (the data owner) grants read access to 'record0' about 'item0' to 'company0' and 'company1' (the partner). The extensibility can be achieved by adding new properties e.g. `cta:grantsWrite` to grant *write* access.

```
: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 .
```

Figure 5.4: CTA Policy in RDF Turtle syntax.

These assertions are sent to the DS. A local copy of the assertions is kept in EPC IS to allow protection and local verification of the chain of trust. To share data, the information owner should add assertions for the desired partners with whom it wants to share. For audit purposes, all assertions should be logged.

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

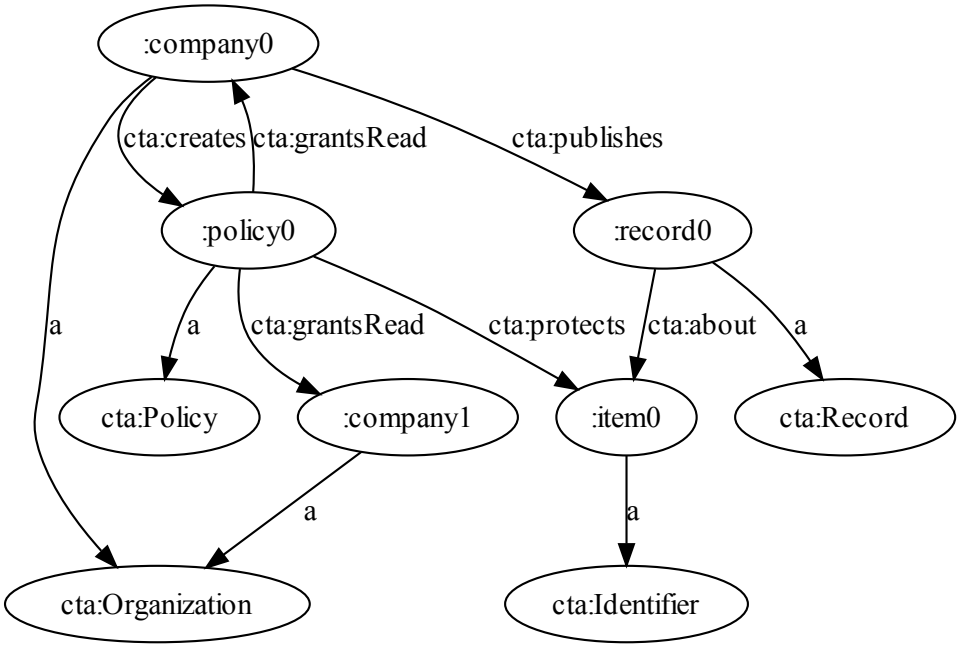

Figure 5.5: CTA Policy graph.

5.1.4 Conversion to XACML

The conversions described next are based on previous work by Karjoth et al. [2008] and they allow SCAz policy instances to be represented in the standard XACML format, as intended.

To convert from EAC, each item data set has a single ACL and that ACL corresponds to a single XACML policy. The item identifier is used as policy name and as target. The rule combining algorithm is 'first-applicable' meaning that the outcome of the first matched rule will be the access decision. For each entry in the source ACL, a 'Permit' rule is generated for each subject-action pair. Also a 'Deny' rule is generated. Lastly, there is a "Deny all" rule for all other attempts.

To convert from CCT, each token corresponds to one XACML policy that expresses its permissions. The token identifier is used as policy name and as target. For each capability encoded in the token there is a 'Permit' rule that checks if the action-resource pair and the secret are correct. In the end there is a "catch" rule to deny access for all other attempts using the token.

To convert from a CTA policy to a XACML policy, the RDF statements are navigated as follows: first the objects of "*policy* protects *identifier*" 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 "*policy* grantsRead *organization*" 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.

5.2 Performance assessment

The aim of the performance assessment was to compare the visibility restriction implementations and evaluate if their performance is suitable for use in supply chains.

The policies were defined using the common SCAz 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 [Huonder, 2010].

5.2.1 Assessment tool

The SCAz assessment tool was used to perform test runs on the 'raw' EAC, CCT and CTA implementations and on their XACML-equivalent forms.

Figure 5.6 represents the data flow associated with the tool. The initial input contains scenario statistics, such as the 'chain length' and the 'number of items', and the 'random seed'. The random number generator is used to generate item paths and to pick the trading partners. The authorization requests are also generated with equal probability of the expected outcome being 'Permit' or 'Deny'.

The access decision is computed using the 'raw' implementation of either EAC, CCT or CTA to determine the expected outcome. Then the policies and requests are translated to XACML, and a job is sent to the PDP, where the policy files are unmarshaled[3] and deployed, the requests are unmarshaled and evaluated, and the response is marshaled[4]. The measurements are collected and final statistics are computed.

The SCAz decisions were compared with the XACML decisions. If any inconsistent response is returned – e.g. the implementation says 'Deny' but XACML PDP says 'Permit' – then the discrepancy is detected to ensure that only correct responses are used.

5.2.2 Experiments

The performance of policy loading and policy evaluation is measured to see if the response times are acceptable. To assess the performance two experiments were designed: 'companies' and 'items'.

The test machine was a Quad-core CPU[5] 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 absolute values of the presented results will differ in another server, but the relative performance should be similar. The experiments were repeated several times – a least 30 times each – so that the sampling distribution could be considered 'normal' according to the Central Limit Theorem [Ross, 2009] to obtain statistically meaningful values.

Items: Figure 5.7 presents a plot of the 'request evaluation' time for increasing number of items. EAC using Java's ACL and CTA using Apache Jena's RDF handle the loads much better with results below 0.1 ms. The performance of CCT implemented with custom code is clearly

[3]Unmarshal: convert from XML text file to Java objects in-memory.
[4]Marshal: convert from Java objects in-memory to XML text file.
[5]Intel Core 2 Quad Central Processing Unit Q8300

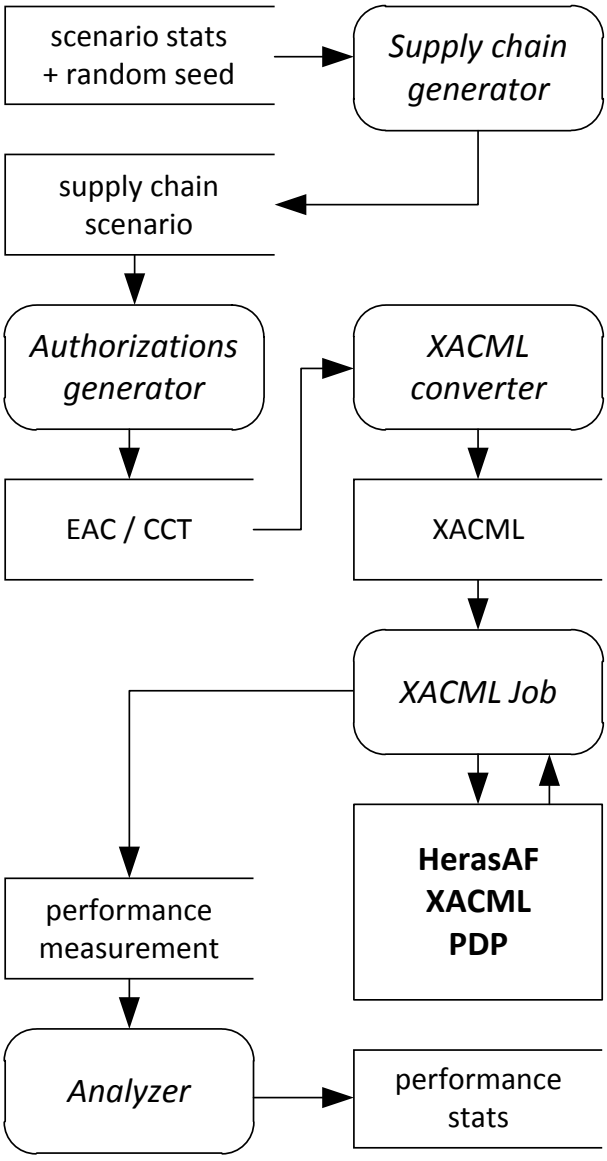

Figure 5.6: SCAz tool data flow diagram.

the worse. The reason for this difference, as observed with a profiling tool[6], was that the CCT implementation 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.

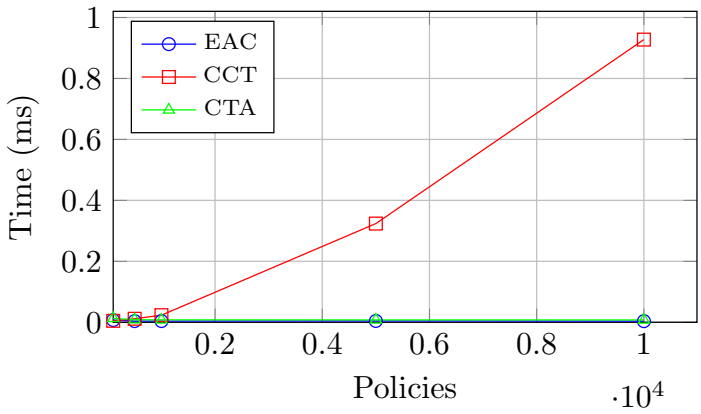

Figure 5.7: Raw EAC, CCT and CTA evaluation time with increasing number of items.

Figure 5.8 presents a plot of the 'request evaluation' time, again for increasing number of item policies converted to XACML format. The performance of CCT and CTA are the best. EAC is worse but on the same order of magnitude. The differences present in the 'raw' implementation of CCT are not visible in the XACML implementation.

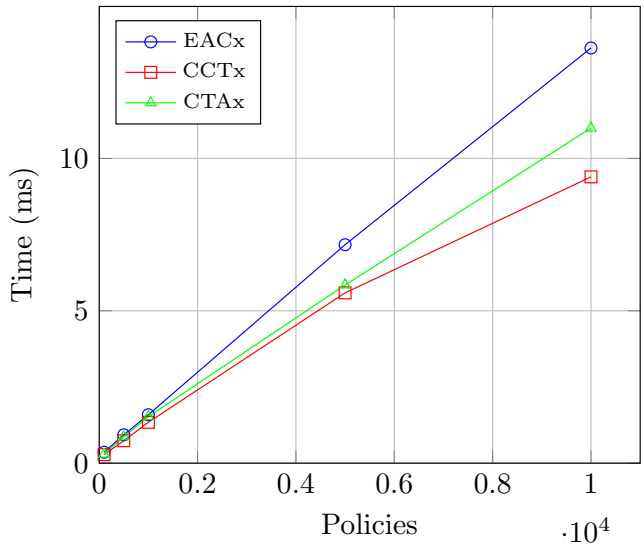

Figure 5.8: XACML EAC, CCT and CTA evaluation time with increasing number of items.

Raw versus XACML: Comparing the y-axis of Figures 5.7 and 5.8 it is visible that the XACML performance overheads are very significant. Table 5.1 presents each XACML time divided by the corresponding 'raw' time for chains handling increasing numbers of items. A *400-fold overhead* was observed, on average. Also, the performance overhead increases with the number of deployed policies.

[6] http://www.ej-technologies.com/products/jprofiler/overview.html

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 5.1: XACML overhead with increasing number of item policies.

For item collections larger than 10^4, the policy loading times became too high, and meaningful results could not be produced. This limitation might pose problems for large item volumes (e.g. 10^6) [Ilic et al., 2011].

Companies Again, three supply chains with different lengths z were considered: short (3), medium (6), and long (12). The length of the supply chain directly impacts the number of companies to be authorized in the experiment.

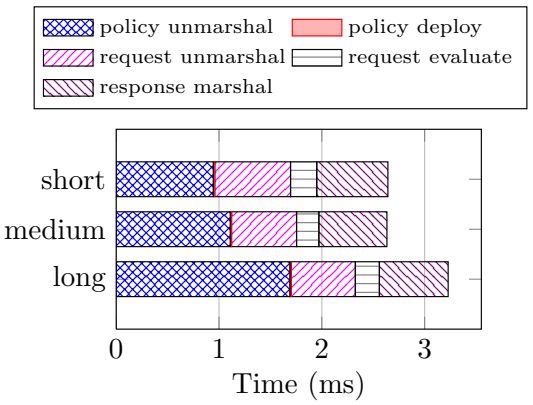

Figure 5.9: XACML EAC processing time breakdown for request evaluation.

Figure 5.9 shows the processing time for the three supply chains using EAC. The number of authorized partners in a policy increases with the item path length, making the generated policy size also increase. The average response time grows with the chain length, but below linear for the considered chains. For CCT (not shown), each XACML policy has constant size because the token and its authorized actions have constant size. This improves the performance, making CCT faster than EAC.

5.2.3 Discussion

The comparison of visibility restriction approaches is summarized in Table 5.2. The estimated performance (Perf.) for all approaches is very similar and does not have a significant impact on overall cost. This confirms the results presented in Chapter 4. However, the expressive potential (Express.) of the approaches is different: EAC and CCT have predefined semantics and are, in this sense, limited; CTA uses assertions and its semantics can be extended.

	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 5.2: Summary comparison of visibility restriction approaches.

5.3 Conclusion

The visibility approaches were assessed in this Chapter. The different SCAz implementations – EAC, CCT, and CTA – were compared using the common API to allow the same visibility constraints to be mapped transparently to distinct implementations; and then converted to the standard XACML format. The data sharing policies were correctly translated and enforced. All three mechanisms expressed exactly the same visibility restrictions and no significant performance differences were found. XACML is a useful policy enforcement standard with suitable performance whereas SCAz can be much more intuitive to express the intended restrictions for supply chain data visibility.

The Chain-of-Trust Assertions (CTA) implementation has similar performance to the other approaches based on access control lists and capabilities. The specification of CTA policies uses RDF to achieve open-ended expressiveness and a distributed data model. The enforcement of CTA policies uses XACML to provide a standardized enforcement infrastructure that enables the use of trusted management and audit tools. Considering that the performance is similar to EAC and CCT, and that CTA is expressive and extensible, it should be the choice for expressing supply chain authorizations. However, the stated expressiveness of CTA remains to be evaluated in an actual case study.

6 Case Study

There are growing concerns that illegitimate products are penetrating the Pharma(ceuticals) supply chain and there are proposals in many countries to apply RFID to solve this problem. The *goal* is strengthening the security of the supply chain through enhanced data visibility control. In this Chapter, the cost models introduced in Chapters 3 and 4 are used to compare candidate solutions to achieve more security in the *Pharma* supply chain: Point-of-Dispense Authentication, Network-based electronic Pedigree, and Document-based electronic Pedigree.

The case study is also used to evaluate the expressiveness of the CTA visibility restriction implementation introduced in Section 5.1.3, as considered necessary by relevant industry trading partners.

6.1 Security of the Pharmaceutical supply chain

The current traceability technology choice for the *Pharma* industry is that item-level identifiers are encoded using printed two-dimensional bar-codes – GS1 DataMatrix [GS1, 2010] – and container-level identifiers are stored in RFID tags. The GS1 identifier architecture provides a data carrier compatibility layer [GS1, 2012] that will allow a gradual transition from bar-codes to RFID tags in the future. Marking products with unique identifiers is necessary but not sufficient to achieve traceability in the supply chain.

There have been documented cases [Eban, 2005] of illegitimate drugs re-entering the legitimate *Pharma* supply chain. The HDMA[1] Fact Book [HDMA, 2013] reports that counterfeit drug cases are also on the rise. This endangers patients and reduces the public's trust in brand names. More security is needed in the legitimate supply chain to prevent re-introduction of pharmaceuticals.

Past problems reported have led larger wholesalers to make a *pledge* to increase the security of the supply chain: only buy drug supplies directly from the manufacturers. Nevertheless, there are *vulnerabilities*:

- Wholesalers can ignore their pledge;
- Counterfeit or stolen products are returned in replacement of legitimate products through a wholesaler; and
- Wholesalers or pharmacists/pharmacies can be the criminals.

[1] The **H**ealthcare **D**istribution **M**anagement **A**ssociation is an organization representing primary health care distributors in the U.S.

The next Section discusses the protections being proposed to prevent illegitimate products from entering the supply chain through these vulnerabilities.

6.1.1 Protections

Around the world, there is a shared understanding of the challenges facing *Pharma* supply chains. Nevertheless there are different views of the solution. In the European Union (EU) [EFPIA et al., 2012] the proposed solution is *Point-of-Dispense Authentication (PoD)* that verifies the authenticity of products at both ends of the supply chain: manufacturers and pharmacies. In the United States of America (US) the proposed solution is *electronic Pedigree (eP)* that records the chain-of-ownership of the products. Both PoD and eP require that the products be identified with unique serial numbers.

In a PoD (Point-of-Dispense) approach, only the two ends in the *Pharma* supply chain need to collaborate: the manufacturers and the pharmacies. The PoD system keeps track of unique serial numbers commissioned by manufacturers and consumed at the point of dispensing to a patient. Checks at the middle of the supply chain are optional.

In a eP (electronic Pedigree) approach, the chain-of-ownership is tracked and its legitimacy is checked by each new owner as the drug moves down the supply chain. There are document-based and network-centric approaches to eP.

DPMS (Drug Pedigree Messaging Standard) [EPCglobal, 2007c] is a GS1 standard for *Document-based eP* (DeP) that was specifically created to assist the *Pharma* supply chain with creating an interoperable system to trace drugs in a way that complies with existing drug pedigree laws. DPMS documents are self-contained and show the chain-of-ownership of a given product. Security is achieved with nested digital signatures [Biskup, 2009].

A special EPC IS event set can be defined for *Network-centric eP* (NeP). These events are captured and stored in EPC IS repositories and then used later for pedigree validation. The current NeP proposals state that the number of repositories should be limited and well-known, entailing a semi-centralized architecture. In this case, there is no need for a Discovery Service (DS).

6.2 US Pharmaceutical Supply Chain

The US *Pharma* supply chain was selected for the case study because it is well documented, with recent statistics collected in the latest edition of the HDMA [2013] Fact Book .

The great majority of drugs dispensed in US pharmacies are initially sold by a manufacturer to a distributor, who then sells them to the dispensing pharmacy[2]. The *Pharma* supply chain has around 1,400 manufacturers, 70 distributors, and 166,000 pharmacies. More than 90% of the volume of drugs passing through the supply chain goes through only three distributors. The vast majority of drugs sold in the U.S. pass through only a single wholesaler on its way from the manufacturer to the pharmacy, making the average chain length z equal to 3. This

[2]Chain pharmacies have their own internal distribution networks and often buy high volume products directly from the manufacturers.

is shorter than the characterizations made in Section 3.1.2, because only the distribution of the finished drugs is being considered.

The average distribution center handles about 50,000 stock keeping units (SKU), where each code is used to identify each unique product classes or items for sale.

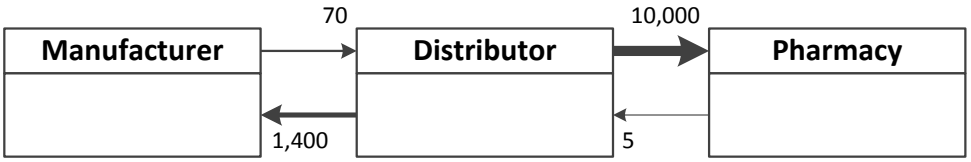

Figure 6.1: US Pharmaceutical supply chain associations with typical cardinalities.

Figure 6.1 represents the "connections" between trading partners that represent both business relationships and network connections [Rodgers, 2011]. The typical *manufacturer* wants to maximize the availability of its products, so it works with as many distributors as it can handle (up to 70 connections). The typical *distributor* (wholesaler) sources from most manufacturers (1,400) and sells to a large number of pharmacies, whether as a primary source or as a secondary source. The three largest distributors each sell and deliver to about 10,000 pharmacies. The *pharmacy* buys their drugs from a primary distributor and, if that distributor is out-of-stock, from secondary sources (up to 5 sources).

6.2.1 Electronic Data Interchange

In the US *Pharma* supply chain the trading partners communicate electronically. The HDMA has published EDI (Electronic Data Interchange) guidelines that are followed by trading partners to exchange structured business data in electronic form. The guidelines are based on the ASC X12 variant of EDI, the most widely used in North America. The following transactions (messages) are relevant for this work because they transmit item identifiers that can be extracted and used later to define traceability data visibility policies: 810 – Invoice; 850 – Purchase Order (PO); 855 – Purchase Order Acknowledgment; and 856 – Advance Ship Notice (ASN). For example, an 856 ASN document informs a trading partner about a shipment of goods arriving at a location.

6.3 Pharmaceutical Supply Chain Authorizations

The data visibility requirements are described next, derived from an industry NeP pilot made in 2012. New authorization assertions are proposed next to comply with the requirements of the *Pharma* supply chain.

The supply chain visibility policy in current use throughout the US *Pharma* supply chain is "one up-one down". Each trading partner knows who they bought the products from ("one up") and who they sold the products to ("one down"). Each trading partner does not know where the drugs came from prior to their immediate supplier.

This policy protects business confidentiality but is insufficient to protect consumers from illegitimate products. Data about an individual physical object should be shared by the

companies in its chain-of-custody, at least. More protection requires upstream traceability data that must be explicitly authorized to be accessed.

6.3.1 Pilot project

A NeP pilot project was implemented by a service provider (GHX[3]) involving a manufacturer (Abbott Laboratories[4]), a distributor (McKesson[5]) and a dispenser (Veterans Administration hospital[6]) [Rodgers, 2012]. Surveys were conducted to establish the functionalities and visibility policies required by the participating companies [Basta, 2011].

Regarding visibility:

- The *pharmacy* sees all the chain-of-custody events only for the product they receive, from the manufacturer through any intermediaries in the supply chain;
- The *distributor* can see all the history prior to acquisition of the drug and its own events. It can see the pharmacy's receive event. From that point on, the information is filtered;
- The *manufacturer* can see its own events and the receiving event of the distributor, but after that all the information is carefully filtered to remove the company location identifiers and none of the downstream consistency checks are shown;
- The *service provider* holds all of the data but the ownership rights – and policy authoring rights – remain with the trading partner who generated each event. Every member of the supply chain owns the events that they contribute. The service provider provides the service of data sharing, as specified by the data owner's policies.

There is a need to express delegated and transitive trust because it is likely that the manufacturer does not know the final destination of the products. Also, a partner may require additional conditions for sharing data. And there must be ways to define trust for sets of products and sets of partners, otherwise the administrative burden of authorizing individual items can quickly become overwhelming.

6.4 Policy building blocks

The CTA implementation presented in Section 5.1.3 uses RDF and it allows new statements to be added. Special predicates can be designed to express dynamic chain upstream/downstream conditions, allowing data sharing. It can also support the delegation of administrative rights from one organization to another. These constructs can also be combined to express rich data sharing conditions, as required by the NeP pilot project. Each company can define its own *trust circle*, like the one in Figure 6.2. Extension assertions can be used to represent trust relationships.

[3] http://www.ghx.com/
[4] http://www.abbott.com
[5] http://www.mckesson.com/
[6] http://www.va.gov/health/

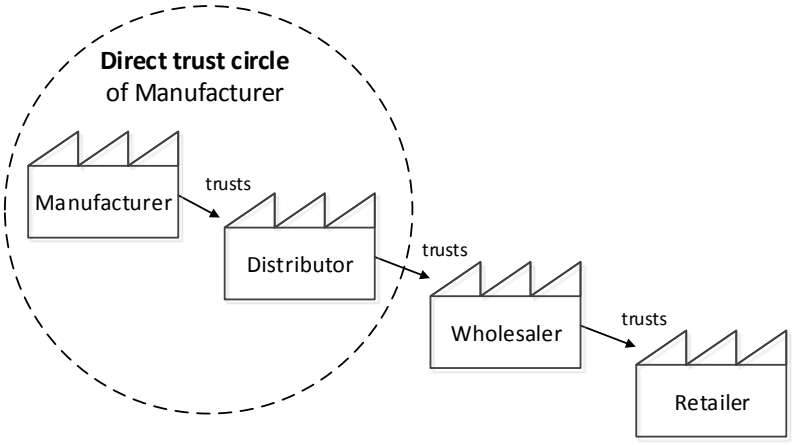

Figure 6.2: Direct trust circle of a Manufacturer.

These extension assertions use supply chain context provided directly by trading partners or inferred from existing documents and product flows in the supply chain. For example, an Advanced Shipping Notice (ASN) document informs a trading partner about a shipment of goods arriving soon to a specific location.

6.4.1 Delegated trust

For trading partners whom the creator of the data record does know directly, a possible solution is for the creator of the data record to grant explicit read access rights and to give *delegation rights* to some of these partners, to allow them to grant further access rights to additional partners that they know and trust.

To share *downstream*, the Manufacturer publishes a record and creates a policy that delegates rights to the Distributor to create additional policies to the appropriate downstream partners, eventually reaching the Retailer.

To share *upstream*, the Retailer publishes a record and creates a policy that delegates rights to an upstream partner instead of a downstream partner. The process continues upstream, until the policy created by Distributor grants read access to the Manufacturer.

Partners in the middle of the supply chain, such as distributors and wholesalers might also publish records and create policies that delegate rights to partners further upstream (their suppliers) or further downstream (their customers).

The extension to add support for the *delegation* of administrative rights from one organization to another is shown in Figures 6.3 and 6.4: 'company0' delegates the ability to grant access to 'company1', and 'company1' grants read access to 'company2'. 'company0' does not have to know 'company2'. Access is granted if there is an explicit unbroken chain of trust assertions leading back to the owner of the data.

To verify the delegated trust, first the 'cta:delegates' predicates are checked to see if they chain forward; then the 'cta:grantsRead' predicates are checked to see if access is granted by anyone in the established trust circle.

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

:company0   cta:creates      :policy0  .
:policy0    cta:protects     :item0  .
:policy0    cta:delegates    :company1  .
:policy0    cta:grantsRead   :company2  .
```

Figure 6.3: CTA delegation extension (type definition predicates omitted).

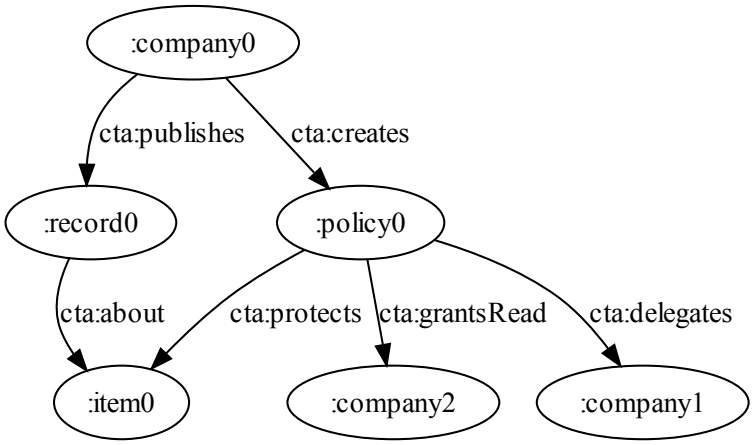

Figure 6.4: CTA delegation extension graph.

6.4.2 Transitive trust

Trust for data regarding a specific item sometimes needs to express dynamic chain upstream/downstream conditions, allowing data sharing between partners that did not have previous interactions. The *transitive* predicates are represented in Figure 6.5 and 6.6. By issuing the 'cta:trustChain' predicate, 'company0' allows 'company1' to access 'record0' about 'item0' because it published 'record1' about the same item.

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

:company0   cta:creates      :policy0 .
:policy0    cta:protects     :item0 .
:policy0    cta:trustChain   :item0.

:company1   cta:publishes    :record1 .
:record1    cta:about        :item0 .
```

Figure 6.5: CTA chain trust transitivity extension (type definitions omitted).

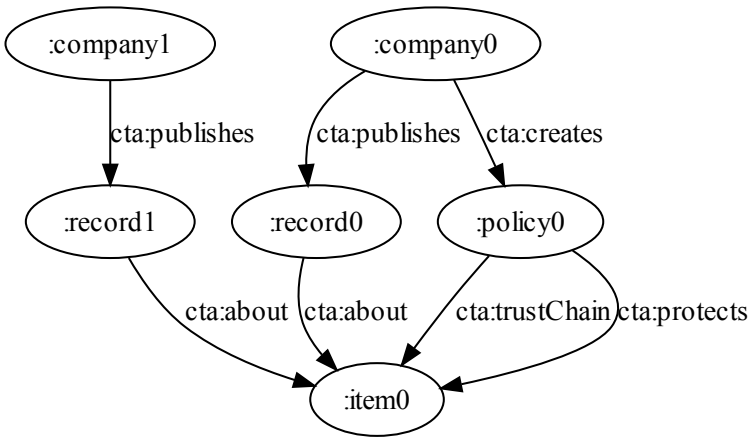

Figure 6.6: CTA chain trust transitivity extension graph.

6.4.3 Conditional trust

Trading partners can issue *conditional* assertions, like reciprocal trust: *"I trust you if you trust me"*. The reciprocal trust predicates are represented in Figures 6.7 and 6.8. The 'cta:grantsReadRecipr' issued by 'company0' is only effective if a similar predicate is issued granting conditional access to records about the same item by 'company1'.

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

:company0   cta:creates             :policy0  .
:policy0    cta:protects            :item0  .
:policy0    cta:grantsReadRecipr    :company1.

:company1   cta:creates             :policy1  .
:policy1    cta:protects            :item0  .
:policy1    cta:grantsReadRecipr    :company0.
```

Figure 6.7: CTA reciprocal trust extension (type definitions omitted).

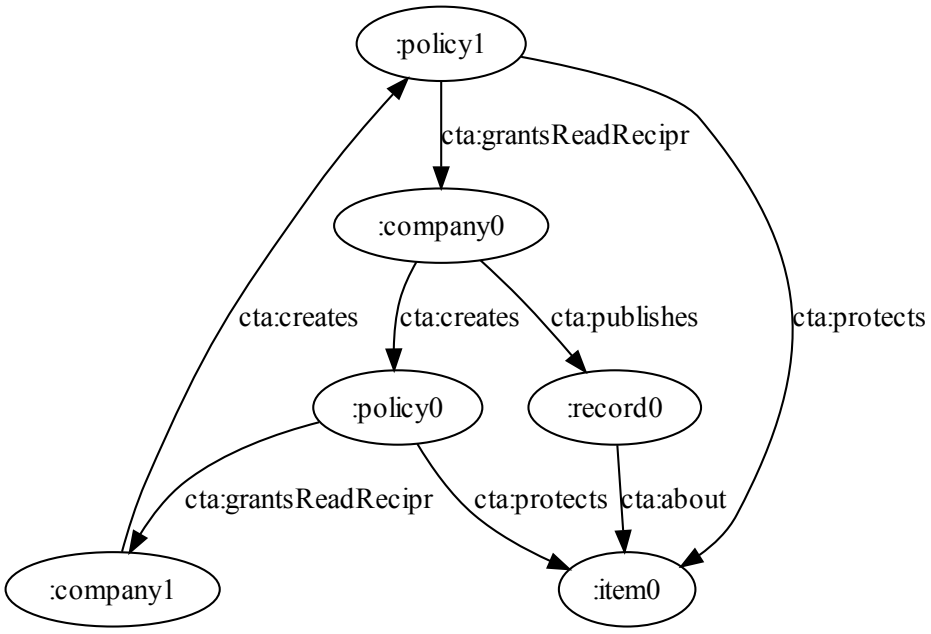

Figure 6.8: CTA reciprocal trust extension graph.

6.4.4 Bulk trust

So far the visibility policies have addressed individual items and individual companies. However, considerable efficiency can be obtained by representing object groups (lots) and company sets (groups).

There are three ways of modeling relationships with cardinality greater than one in RDF. The first, and simplest, is to define multiple values for a predicate. The second uses 'head' and 'rest' predicates to create a linked list and is intended for closed, ordered collections. The third uses types and special ordinal predicates to specify the items that belong to the collection. There are ordered and unordered collections, called *Sequence* and *Bag*, respectively.

Figures 6.9 and 6.10 represent 'lot0' that contains three items and 'group0' that contains two companies. For the *product lot*, multiple predicate values were used because it is a simpler, less verbose approach that is suitable for relationships without attributes. The lot object can be further characterized, with predicates for it. For the *trading partner group*, a 'Bag' was used because it provides an identity to the collection and allows further characterization of the relationship. Also the cardinality of the relationship is expected to be much smaller than the lots that can reach thousands of items.

```
:lot0    cta:inLot   :item0.
:lot0    cta:inLot   :item1.
:lot0    cta:inLot   :item2.

:group0  cta:group  [
     a          rdf:Bag;
     rdf:_1     :company0;
     rdf:_2     :company1
].
```

Figure 6.9: CTA bulk trust for product lot and company group.

6.5 Assessment

The PoD, NeP, and DeP proposals for the US *Pharma* supply chain were assessed using the cost model. They are classified in Figure 6.11 according to data integration and centralization. Data integration specifies if the system copies data (*copy*) or refers to it (*refer*). Centralization specifies if the system has special nodes (*centralized*) or not (*decentralized*). In this particular case, referrals to data stored in other trading partners were not considered an acceptable option because the companies need to be able to prove that they are fulfilling the legal requirement of protecting the products using their own systems or using service providers with a Service Level Agreement (SLA).

PoD is centralized and copies identifier usage data to a special repository node. NeP is also centralized but copies more data – EPC IS events – to the pedigree repository. DeP is decentralized because the DPMS pedigree records are not stored in any special node, and the accumulated data is copied along the supply chain.

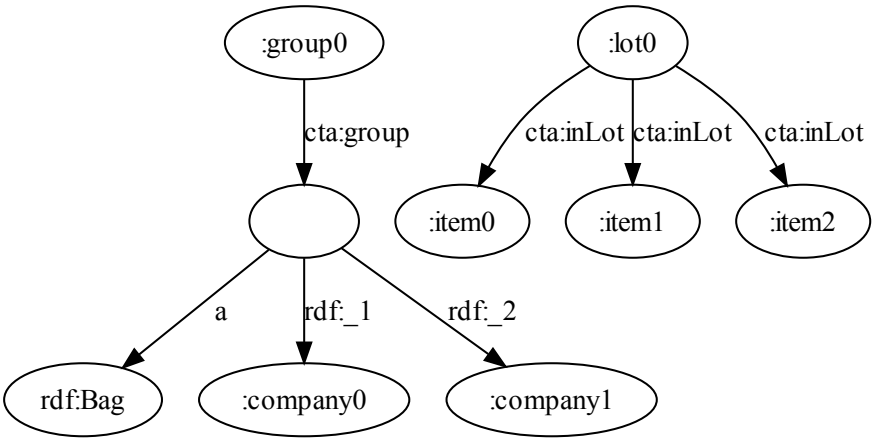

Figure 6.10: CTA bulk trust graph.

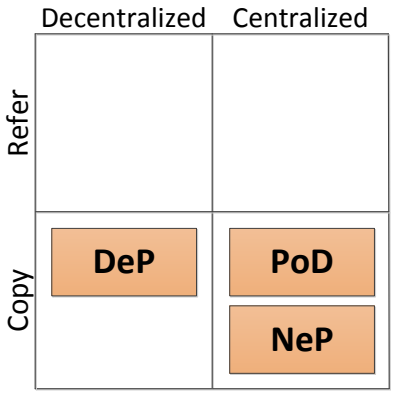

Figure 6.11: Pharmaceutical traceability system classification.

Each of the solutions – PoD, DeP, and NeP – is represented next. A Public-Key Infrastructure (PKI) [Housley et al., 1999] is represented in all approaches because cryptographically strong authentication is required before product identifier data is published by a manufacturer or queried by a pharmacy.

6.5.1 Point-of-Dispense Authentication

Figure 6.12 represents a PoD solution, including the cardinalities of the data exchange connections. The PoD repository keeps product instance data.

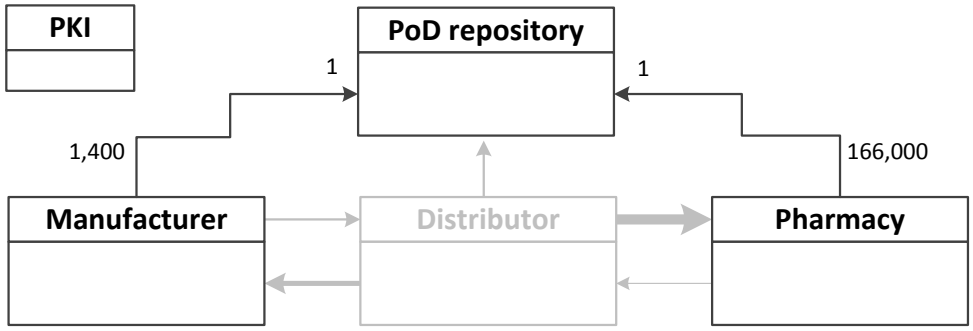

Figure 6.12: PoD data exchange connection cardinalities.

As represented in Figure 6.12, on commission of a new product (id), the manufacturer sends a message to the PoD repository to register new identifiers. On sale of a product, the pharmacy sends a message to the PoD repository to verify that the identifier is still unsold. There can be additional identity checks, usually triggered by a random test or by a specific suspicion. On suspicion, the trading partner (e.g. a wholesaler) checks if the identifiers belong to the expected manufacturer and are fit for sale.

6.5.2 Network-based electronic Pedigree

Figure 6.13 represents a NeP solution, including the cardinalities of the data exchange connections. A semi-centralized model is assumed, where each trading partner connects with a small number of service providers (just one in the Figure). The NeP repository implements the EPC IS interfaces, extended with a pedigree checking service.

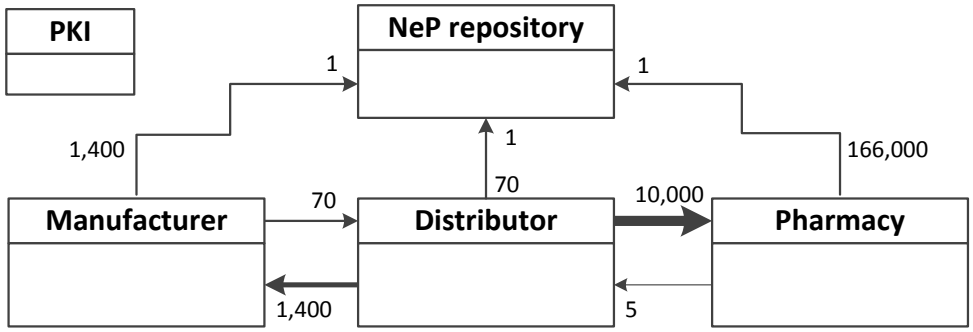

Figure 6.13: NeP data exchange connection cardinalities.

For outgoing products, an EPC IS event must be published (e.g. with business step 'shipping'). For incoming products[7], a query is issued to the checking service that will apply the relevant pedigree regulations. When the physical products actually arrive, and EPC IS event is published with the business step of 'receiving'. Aggregation events (required to keep track of container transports) are being omitted for simplification.

6.5.3 Document-based electronic Pedigree

Figure 6.14 represents a DeP solution, including the cardinalities of the data exchange connections. There is no centralized support service in this case.

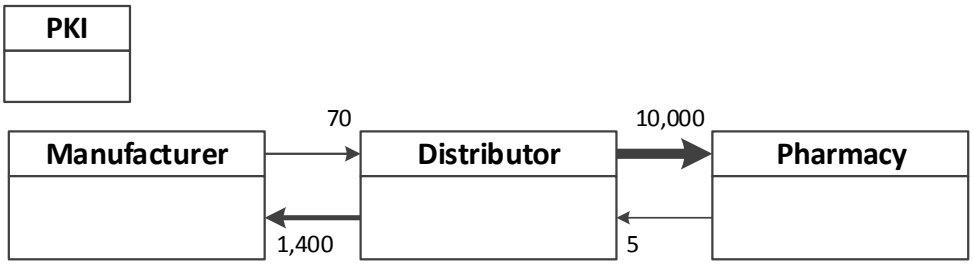

Figure 6.14: DeP data exchange connection cardinalities.

For outgoing products, a new record is appended to the pedigree, and a new digital signature is added. For incoming products, the public key certificates must be retrieved from the PKI to verify the pedigree.

6.5.4 Estimates

The cost model's parameters are presented in Table 6.1 and capture characteristics of the supply chain and target system.

Name	Symbol	Unit	Value
Avg. chain length	z	vertex	3
Message size	μ mu	bit	10^5
Item record size	δ delta	bit	10^5
Bandwidth	β beta	bps	10^9
Processing speed	γ gamma	bps	10^9
Seek time	θ theta	ms	1

Table 6.1: Common parameters.

According to the HDMA Fact Book, a typical Pharmaceuticals Distribution Center handles an average of 100,000 items per day. The cost model produced estimates assuming this number of items as the central value in the plots.

Figure 6.15 represents the storage costs. The data volumes of tens of gigabytes per working day for the distribution center are within the reach of available technology. PoD stores less

[7]The pedigree check can be triggered as soon as a 856 ASN is received.

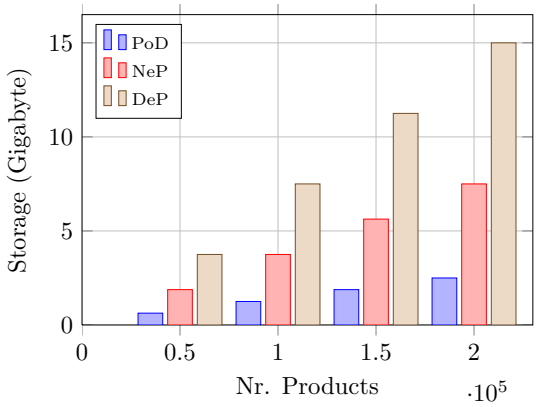

Figure 6.15: Total storage cost for capture.

information than NeP because it only keeps track of identifier usage whereas NeP records more detailed events. The DeP solution has the greatest overall storage requirements, significantly greater than NeP and PoD, because the partial pedigree records are kept at each trading partner.

Figures 6.16 and 6.17 represent the time costs of the data capture and query operations, respectively. The time cost considers both time spent on processing and time spent on network communications. The considered time values spent (hundreds of seconds) are small for a work day with 8 hours (28,800 seconds), as defined in Section 3.3. The cost of the capture is smaller for the PoD, because only two operations are needed, at both ends of the supply chain. The cost for PoD is independent of the chain length. NeP spends time communicating with the central repository. DeP spends (a little less) time communicating with the next trading partner in the chain. In both cases, the cost is dependent on the length of the supply chain (in this case, $z = 3$). The cost of query is smaller for DeP, because only local records have to be retrieved. The cost of the PoD is next because it only queries the identifier state. NeP has more cost because it retrieves the complete pedigree record from the repository. Alternatively, it could return just a check result. In that case, the cost would be similar to PoD.

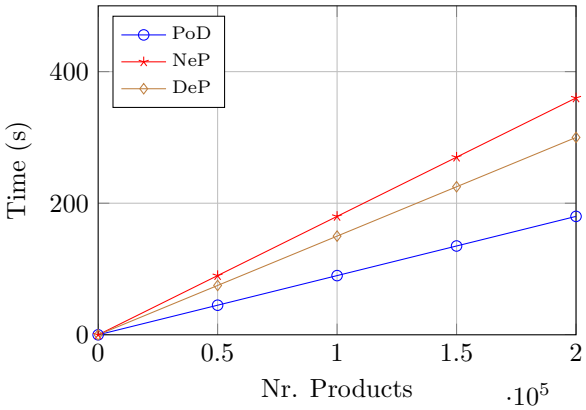

Figure 6.16: Total time cost for capture.

Secure data exchange connections are required for all solutions; in PoD and NeP to connect to the centralized services; in NeP to connect the trading partners in the object's path. The trust required for the digital signatures means that each trading partner needs an account in a PKI.

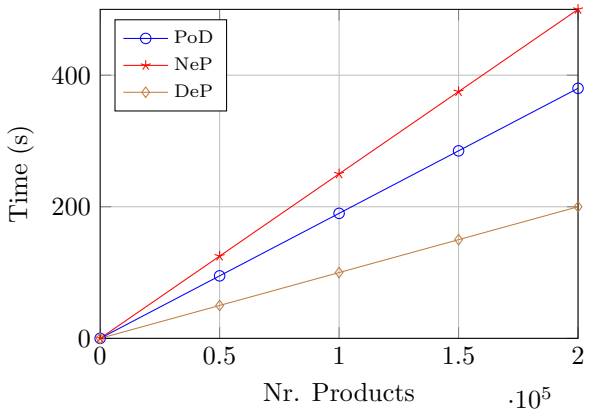

Figure 6.17: Total time cost for query.

The cost of set-up involves the creation of a key pair, and the emission and sharing of public key certificates. This procedure can be added to the existing accreditation procedures already practiced by most companies, as described in the HDMA Fact Book.

Table 6.2 shows the assessment of the required secure connections upstream, downstream and other, for the PoD solution. NeP is the same in this regard. Table 6.3 assesses the DeP case. The sub-total is the number of connections for a single instance. The total is the number of connections for the whole US *Pharma* supply chain (the multiplier values are derived from the cardinalities of the relationships between entities presented in Figures 6.12, 6.13, and 6.14).

	up	**down**	**other**	**sub-total**	**total**
Manufacturer	0	0	1	1	1 400
Distributor	0	0	1	1	70
Pharmacy	0	0	1	1	166 000
					167 470

Table 6.2: PoD and NeP required secure connections.

	up	**down**	**other**	**sub-total**	**total**
Manufacturer	0	70	0	70	98 000
Distributor	1 400	10 000	0	11 400	798 000
Pharmacy	5	0	0	5	830 000
					1 726 000

Table 6.3: DeP required secure connections.

Clearly the centralized approaches – PoD and NeP – require less set-up effort for most trading partners, because less key exchanges are necessary whereas choosing DeP represents a *tenfold* increase in the number of required secure connections.

6.6 Conclusion

In this Chapter, a case study was used to validate the cost models introduced in Chapters 3 and 4, by using them to compare candidate solutions to achieve more security in the *Pharma* supply chain. The produced estimates show that PoD is the most lightweight approach and stores less information. It also requires less secure data exchange connections. This approach however does not aid in criminal investigations, because the chain-of-ownership is not retained. DeP stores the most data – partial pedigrees – across the chain, and also requires ten times more secure connections. It is the most expensive solution but it does provide documentation that can be useful for criminal investigations and prosecutions, when needed. NeP is a middle ground between PoD and DeP. It stores more data than PoD, but assuming a semi-centralized architecture, it minimizes the number of secure data exchange connections. An advantage of NeP is that the same infrastructure – EPC IS – can be used with different legal pedigree regulations, and also for other traceability purposes, including recalls. The recommendation is that the security system can start with a PoD approach and then evolve to a full pedigree.

Extensions to the CTA visibility restriction implementation introduced in Section 5.1.3 were proposed and qualitatively evaluated. The pilot requirements were met using the proposed extensions of delegated, transitive, conditional, and bulk trust. CTA proved to be expressive enough to satisfy the requirements of a real-world case.

7 Conclusion

RFID is one of the technologies that allows things and places in the physical world to automatically generate data for information systems. Eventually every object of interest in the world will be connected to the network, creating an *Internet of Things*:

> The future Internet of Things links uniquely identifiable things to their virtual representations in the Internet containing or linking to additional information on their identity, status, location or any other business, social or privately **relevant information** at a financial or non-financial pay-off that exceeds the **efforts of information provisioning** and offers information access to **non-predefined participants**.
>
> Uckelmann et al. [2011]

There are many research challenges to be addressed before the Internet of Things vision can become a reality [Aitenbichler et al., 2010] [Atzori et al., 2010], but traceability will surely be a core functionality. Traceability systems store and manage the data so that it can be discovered and put to good use, providing answers to traceability queries, like Track/Recall, Trace/Pedigree, and Aggregation/Bill-of-Materials (BoM).

This dissertation focused on the scale and security aspects of traceability systems:

- Chapter 1 elicited the requirements for traceability data (Section 1.3), suitable access controls (Section 1.4), and scalability (Section 1.5);
- A full traceability system was described in detail in Chapter 2. Architecture proposals and security protections were also surveyed;
- Chapter 3 presented an analytic model to compare the cost of data capture and queries for traceability system architectures without relying on implementation details. The Meta-Data Integration (MDI) architecture was found to have the second-best estimated performance and being more suited to security solutions with trusted third parties;
- Chapter 4 detailed the MDI architecture with message-level details to assess visibility restriction approaches;
- The implementation of the visibility restriction approaches and the performance measurements were reported in Chapter 5 and the Chain-of-Trust Assertions (CTA) approach was found to have adequate performance and greater expressive potential;
- The case study of Chapter 6 confirmed the usefulness of the cost models and confirmed that CTA is expressive enough for real-world use cases.

7.1 Contributions

The research presented in this dissertation contributes to the development of the Internet of Things in two concrete ways:

1. The CTA visibility restriction implementation provides a way to keep **"relevant information"** protected by default but accessible to the authorized parties that use it to improve their business. It also provides access to **"non-predefined participants"** through the delegated and transitive trust assertions. CTA is expressive, extensible, and suitable to semantic processing – it uses Linked Data in RDF format for representation – and can be enforced in a standard infrastructure – it can be converted to the XACML policy format.
2. The cost models allow for the estimation of the **"efforts of information provisioning"** without the need for fully specified systems, allowing early planning and decision-making. This will prove very useful, since the scale of the Internet of Things is expected to greatly surmount that of the current Internet – according to Fleisch [2010], it will grow from the current 10^9 to 10^{12} network nodes.

7.2 Future work

This research has shown concrete examples of how the visibility restriction implementation and the cost models can contribute to the design of better traceability systems. There are future work opportunities for the visibility policies, cost models, and traceability systems at large.

7.2.1 Visibility policies

In its current form, CTA implies a significant administrative burden to keep the authorizations 'synchronized' with the physical object flows. To alleviate this problem, the authorization layer should be connected with the B2B layer (e.g. the EDI described in Section 6.2.1) to allow automatic issuing of assertions for sharing data with the current trading partners. Authorizations should also be integrated with the respective modules of ERP and SCM solutions, like the *SAP Authorization Concept* [Linkies and Off, 2006].

The performance of XACML policy evaluation needs to be significantly improved because currently the overheads are very significant, and authorizations will need to apply to more than the 10^4 items used in the assessments. More work is also needed to find the best XACML formulations for the CTA extensions proposed in the case study (Section 6.4).

7.2.2 Cost models

The cost models can be further developed into a design tool to provide early insight for system architects. It can take inputs from industry surveys and domain experts to produce performance estimates that can be plotted graphically. It can also assist in the future development of traceability standards.

More detailed models can be built based on workload profiles [Herrero-López, 2012] and queueing theory [Gross et al., 2008], for example.

Despite the usefulness of the presented results, the tool outputs require further empirical validation from more case studies. The comparison of estimates with actual performance data would help to improve the model and increase the confidence in the results. However, getting actual data for research is a hard task, as many companies do not possess it or are unwilling to provide it. Many attempts to obtain such data for this research were not successful.

7.2.3 Traceability systems

At the present, the MDI architecture is deemed as the most promising for the deployment of traceability data services. However, even with a trusted service provider in place, the willingness to share results will not be total.

Trust circles can be used to increase the willingness to participate in traceability systems. The least known trading partners can be kept in outer trust circles, and they can gradually build trust to access data in inner circles. These circles are intended to serve as lines of defense in authentication. The more detailed access control policies – that already expose some sensitive information – would only be applied to trading partners that are already in the data owner's inner trust circles.

A *distributed architecture*, like UP2P discussed in Section 2.5, allows even greater control by the data owners. The RDF and XACML approaches need to be applied in this architecture to further verify their suitability for architectures other than MDI.

Despite the best design efforts, in the end, there will still companies unwilling to share the data. In these cases, providing *remote validation functions* i.e. functions that use data to produce a desired output without exposing the underlying data; may provide a way to circumvent data sharing resistance. A concrete example is the PoD approach in the case study (described in Section 6.5.1) that provided a way to verify the authenticity of products without exposing the raw data.

Finally, considering the whole traceability system, more data attributes can be shared and made available by the system, like expiry dates and temperature readings, for example. This additional data can be used increase the safety and quality of the products in the supply chain. In the long term, traceability systems will evolve into a *"World Programming Interface (WPI)"* that will provide more and more functions to access relevant facts about the physical world.

A Bibliography

Agrawal, R., A. Cheung, K. Kailing, and S. Schonauer (2006). Towards Traceability across Sovereign, Distributed RFID Databases. In *International Database Engineering and Applications Symposium (IDEAS)*, Delhi, India.

Aitenbichler, E., A. Behring, D. Bradler, M. Hartmann, L. A. Martucci, M. Mühlhäuser, D. S. Ries, D. D. Schnelle-Walka, D. Schreiber, J. Steimle, and T. Strufe (2010, June). Shaping the Future Internet. *Ubiquitous Computing and Communication Journal Special Issue for Future Internet of People, Things and Services (IOPTS) eco-systems workshop*. ISSN: 1992-8424.

Albitz, P. and C. Liu (2006, May). *DNS and BIND* (5th ed.). O'Reilly Media, Inc. ISBN: 978-0596100575.

Allemang, D. and J. Hendler (2011, June). *Semantic Web for the Working Ontologist, Second Edition: Effective Modeling in RDFS and OWL* (2nd ed.). Morgan Kaufmann. ISBN: 978-0123859655.

Alm, C. and R. Illig (2010). Translating High-Level Authorization Constraints to XACML. In *6th World Congress Services (SERVICES-1)*, Miami, FL, USA, pp. 629–636. IEEE. ISBN: 978-0769541297.

Alonso, G., F. Casati, H. Kuno, and V. Machiraju (2004). *Web Services: Concepts, Architectures and Applications*. Springer Verlag. ISBN: 978-3540440086.

Arkills, B. (2003, March). *LDAP Directories Explained: An Introduction and Analysis*. Addison-Wesley Professional. ISBN: 978-0201787924.

Atkins, D. and R. Austein (2004, August). RFC 3833 – Threat Analysis of the Domain Name System (DNS). IEFT.

Atzori, L., A. Iera, and G. Morabito (2010, October). The Internet of Things: A survey. *Computer Networks – The International Journal of Computer and Telecommunications Networking 54*(15), 2787–2805. ISSN: 1389-1286.

Baier, D., V. Bertocci, K. Brown, S. Densmore, E. Pace, and M. Woloski (2013, March). *A Guide to Claims-Based Identity and Access Control: Authentication and Authorization for Services and the Web – 2nd edition*. Microsoft. ISBN: 978-1621140023.

Balakrichenan, S., A. Kin-Foo, and M. Souissi (2010). Federated ONS Architecture for the Internet of Things - A Functional Evaluation. In *Internet of Things*. Springer.

Balakrishnan, H., M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica (2003, February). Looking up data in P2P systems. *Communications of the ACM 46*, 43–48.

Barchetti, U., A. Bucciero, M. D. Blasi, L. Mainetti, and L. Patrono (2010). RFID, EPC and B2B convergence towards an item-level traceability in the pharmaceutical supply chain. In *IEEE International Conference on RFID Technology and Applications (RFID-TA)*, Guangzhou, China, pp. 194–199.

Bass, L., P. Clements, and R. Kazman (2003). *Software Architecture in Practice – 3rd edition*. Addison-Wesley Professional. ISBN: 978-0321815736.

Basta, N. (2011, April). Healthcare Exchange Bids for Prototyping a Track-and-Trace System. *Pharmaceutical Commerce*.

Beier, S., T. Grandison, K. Kailing, and R. Rantzau (2006, December). Discovery Services – Enabling RFID Traceability in EPCglobal Networks. In *International Conference on Management of Data (COMAD)*, Delhi, India.

Beigl, M., , M. Beuster, D. Rohr, T. Riedel, C. Decker, and A. Krohn (2007, June). S2B2: Blackboard for Transparent Data and Control Access in Heterogeneous Sensing Systems. In *4th International Conference on Networked Sensing Systems (INSS)*, Braunschweig, Germany, pp. 126–129.

Bhattacharyya, R., D. Deavours, C. Floerkemeier, and S. Sarma (2011, April). RFID Tag Antenna Based Temperature Sensing in the Frequency Domain. In *IEEE International Conference on RFID*, Orlando, FL, USA, pp. 70–77.

Biskup, J. (2009). *Security in Computing Systems - Challenges, Approaches and Solutions*. Springer. ISBN: 978-3540784425.

Bose, I. and R. Pal (2005). Auto-ID: managing Anything, Anywhere, Anytime in the Supply Chain. *Communications of the ACM 48*(8), 100–106.

Brachman, R. and H. Levesque (2004). *Knowledge Representation and Reasoning*. Morgan Kaufman. ISBN: 978-1558609327.

BRIDGE (2007, August). Requirements document of serial level lookup service for various industries. Technical report, University of Cambridge and AT4 wireless and BT Research and SAP Research and ETH Zurich and GS1 UK. Building Radio Frequency IDentification for the Global Environment.

Brucker, A. D., I. Hang, G. Lückemeyer, and R. Ruparel (2012, June). SecureBPMN: modeling and enforcing access control requirements in business processes. In *17th ACM Symposium on Access Control Models and Technologies*, SACMAT, Newark, NJ, USA, pp. 123–126. ACM.

Burbridge, T. and M. Harrison (2009, April). Security Considerations in the Design and Peering of RFID Discovery Services. In *IEEE International Conference on RFID*, Orlando, FL, USA, pp. 249–256.

Butler, B., B. Jennings, and D. Botvich (2010, October). XACML policy performance evaluation using a flexible load testing framework. In *17th ACM Conference on Computer and Communications Security*, CCS, Chicago, IL, USA, pp. 648–650.

Cadenhead, T., V. Khadilkar, M. Kantarcioglu, and B. Thuraisingham (2012, June). A cloud-based RDF policy engine for assured information sharing. In *17th ACM symposium on Access Control Models and Technologies*, SACMAT, Newark, NJ, USA, pp. 113–116. ACM.

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

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

Cao, Y., D. Wang, and H. Sheng (2007, September). PTSP: a lightweight EPCDS platform to deploy traceable services between supply-chain applications. In *1st Annual RFID Eurasia*, Istanbul, Turkey, pp. 1–5. IEEE.

Cheung, A., K. Kailing, and S. Schonauer (2007, April). Theseos: A Query Engine for Traceability across Sovereign, Distributed RFID Databases. In *23rd IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, pp. 1495–1496.

Choi, T.-M. and S. Sethi (2010). Innovative quick response programs: a review. *International Journal of Production Economics 127*, 1–12. ISSN: 0925-5273.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein (2009). *Introduction to Algorithms*. MIT Press. ISBN: 978-0262033848.

Cruellas, J. C., G. Karlinger, D. Pinkas, and J. Ross (2003, February). XML advanced electronic signatures (XAdES). W3C.

Derakhshan, R., M. E. Orlowska, and X. Li (2007, March). RFID Data Management: Challenges and Opportunities. In *IEEE International Conference on RFID*, Orlando, FL, USA, pp. 175–182.

Dierks, T. and E. Rescorla (2008, August). RFC 5246 – The Transport Layer Security (TLS) Protocol – Version 1.2. IEFT.

Do, H.-H., J. Anke, and G. Hackenbroich (2006). Architecture evaluation for distributed Auto-ID systems. In *17th International Workshop on Database and Expert Systems Applications (DEXA)*, Krakow, Poland, pp. 30–34.

Eastlake, D., J. Reagle, and D. Solo (2002, February). XML-Signature Syntax and Processing. W3C.

Eban, K. (2005). *Dangerous Doses: How Counterfeiters Are Contaminating America's Drug Supply*. Houghton Mifflin Harcourt. ISBN: 978-0151010509.

EFPIA, GIRP, and PGEU (2012, June). European Stakeholder Model (ESM) - ensuring patients have access to safe medicines. European Federation of Pharmaceutical Industries and Associations.

EPCglobal (2006, June). Reader Protocol 1.1.

EPCglobal (2007a, September). EPC Information Services (EPCIS) 1.0.1 Specification.

EPCglobal (2007b, August). Low Level Reader Protocol (LLRP) 1.1.

EPCglobal (2007c, January). Pedigree 1.0.

EPCglobal (2007d, May). Reader management 1.0.1.

EPCglobal (2008a, May). EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz 1.2.0.

EPCglobal (2008b, May). Object Name Service (ONS) 1.0.1.

EPCglobal (2009a, March). Application Level Events (ALE) Specification 1.1.1.

EPCglobal (2009b, June). Tag Data Translation (TDT) 1.4.

EPCglobal (2010a, October). Core Business Vocabulary Standard.

EPCglobal (2010b, August). Tag Data Standards 1.5.

ETSI (2003). Methods and protocols for security; part 1: Threat analysis. Technical Specification ETSI TS 102 165-1 V4.1.1. ETSI.

Etzion, O. and P. Niblett (2010, August). *Event Processing in Action*. Manning. ISBN: 978-1935182214.

Eurich, M., N. Oertel, and R. Boutellier (2010, December). The impact of perceived privacy risks on organizations' willingness to share item-level event data across the supply chain. *Journal of Electronic Commerce Research 10*(3-4), 423–440. ISSN: 1572-9362.

Evdokimov, S., B. Fabian, S. Kunz, and N. Schoenemann (2010, June). Comparison of Discovery Service Architectures for the Internet of Things. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, Newport Beach, CA, USA, pp. 237–244.

Fabian, B. (2009, June). Implementing secure P2P-ONS. In *IEEE International Conference on Communications (ICC)*, Dresden, Germany, pp. 1–5.

Fabian, B., O. Günther, and S. Spiekermann (2005, July). Security Analysis of the Object Name Service. In *International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SECPerU)*, Santorini Island, Greece.

Ferrini, R. and E. Bertino (2009, June). Supporting RBAC with XACML+OWL. In *14th ACM Symposium on Access Control Models and Technologies*, SACMAT, Stresa, Italy, pp. 145–154.

Finin, T., A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham (2008). ROWLBAC: representing role based access control in OWL. In *13th ACM Symposium on Access Control Models and Technologies*, SACMAT, Estes Park, CO, USA, pp. 73–82.

Finkenzeller, K. and D. Muller (2010, August). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication* (3rd ed.). Wiley. ISBN: 978-0470695067.

Fleisch, E. (2010, January). What is the Internet of Things? An Economic Perspective. Technical report, ETH Zurich / University of St. Gallen Auto-ID Labs.

Floerkemeier, C. and M. Lampe (2004, April). Issues with RFID usage in ubiquitous computing applications. In *2nd International Conference on Pervasive Computing (PERVASIVE)*, Linz / Vienna, Austria.

Floerkemeier, C., C. Roduner, and M. Lampe (2007, December). RFID Application Development with the Accada Middleware Platform. *IEEE Systems Journal, Special Issue on RFID Technology 1*(2), 82–94. ISSN: 1932-8184.

Fowler, M. (2003, September). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Boston, MA, USA: Addison-Wesley. ISBN: 978-0321193681.

Fowler, M., D. Rice, M. Foemmel, E. Hieatt, R. Mee, and R. Stafford (2002, November). *Patterns of Enterprise Application Architecture*. Addison-Wesley. ISBN: 978-0321127426.

Framling, K., T. Ala-Risku, M. Karkkainen, and J. Holmstrom (2007). Design Patterns for Managing Product Life Cycle Information. *Communications of the ACM 50*(6), 75–79.

Friedlander, A., A. Mankin, W. D. Maughan, and S. D. Crocker (2007, June). Dnssec: a protocol toward securing the internet infrastructure. *Communications of the ACM 50*(6), 44–50.

Garcia-Alfaro, J., M. Barbeau, and E. Kranakis (2008, May). Analysis of Threats to the Security of EPC Networks. In *6th Annual Communication Networks and Services Research Conference (CNSR)*, Halifax, Nova Scotia, Canada, pp. 67 –74.

Gross, D., J. F. Shortle, J. M. Thompson, and C. M. Harris (2008, August). *Fundamentals of Queueing Theory* (4th ed.). Wiley. ISBN: 978-0471791270.

Grummt, E. and M. Müller (2008). Fine-grained access control for epc information services. In C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern, and S. Sarma (Eds.), *The Internet of Things*, Volume 4952 of *Lecture Notes in Computer Science*, pp. 35–49. Springer Berlin / Heidelberg. 10.1007/978-3-540-78731-0_3.

GS1 (2010). The Value and Benefits of the GS1 System of Standards. Technical report, GS1.

GS1 (2012, August). RFID Bar Code Interoperability. Technical report, GS1.

Günther, O., W. Kletti, and U. Kubach (2008). *RFID in Manufacturing*. Springer. ISBN: 978-3540764540.

Harris, S. and A. Seaborne (2012). SPARQL 1.1 Query Language. W3C.

HDMA (2012-2013). *HDMA Fact Book*. Healthcare Distribution Management Association.

Hebig, R. N., C. Meinel, M. Menzel, I. Thomas, and R. Warschofsky (2009, July). A Web Service Architecture for Decentralised Identity- and Attribute-Based Access Control. In *IEEE International Conference Web Services (ICWS)*, Miami, FL, USA, pp. 551–558.

Herrero-López, S. (2012). *Large-scale simulator for global data infrastructure optimization*. Ph. D. thesis, Massachusetts Institute of Technology. Dept. of Civil and Environmental Engineering.

Housley, R., W. Ford, W. Polk, and D. Solo (1999, January). RFC 2459 – Internet X.509 Public Key Infrastructure. IEFT.

Howard, M. and S. Lipner (2006, June). *The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software*. Microsoft Press. ISBN: 978-0735622142.

Huang, D., M. Verma, A. Ramachandran, and Z. Zhou (2007, March). A Distributed ePedigree Architecture. In *IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, Sedona, AZ, USA, pp. 220–230.

Huonder, F. (2010, July). Conflict Detection and Resolution of XACML Policies. M. Sc. thesis, University of Applied Sciences Rapperswil.

Ilic, A., A. Grössbauer, F. Michahelles, and E. Fleisch (2011). Understanding Data Volume Problems of RFID-enabled Supply Chains. *Business Process Management Journal 16*(6), 904–916. ISSN: 1463-7154.

Ilic, A., F. Michahelles, and E. Fleisch (2007, May). The Dual Ownership Model: Using Organizational Relationships for Access Control in Safety Supply Chains. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, Volume 2, Niagara Falls, Ontario, Canada, pp. 459–466.

Jeffery, S. R., M. Garofalakis, and M. J. Franklin (2006, September). Adaptive Cleaning for RFID Data Streams. In *32nd International Conference on Very Large Data Bases (VLDB)*, Seoul, Korea, pp. 163–174.

Juric, M. B., I. Rozman, B. Brumen, M. Colnaric, and M. Hericko (2006). Comparison of performance of Web Services, WS-Security, RMI, and RMI-SSL. *Journal of Systems and Software 79*(5), 689–700. ISSN: 0164-1212.

Karjoth, G., A. Schade, and E. V. Herreweghen (2008, December). Implementing ACL-Based Policies in XACML. In *Annual Computer Security Applications Conference (ACSAC)*, Anaheim, CA, USA, pp. 183–192.

Kürschner, C., C. Condea, O. Kasten, and F. Thiesse (2008). Discovery Service Design in the EPCglobal Network, Towards Full Supply Chain Visibility. *The Internet of Things, Lecture Notes in Computer Science 4952*, 19–34. ISBN: 978-3540787303.

Landt, J. (2005). The history of RFID. *IEEE Potentials 24*(4), 8–11.

Laudon, K. and J. Laudon (2011, January). *Management Information Systems - 12th edition*. Prentice Hall. ISBN: 978-0273789970.

Laurence, A., J. L. Moulec, J. Madelaine, and I. Bedini (2010, June). Experiments of Discovery Services Interconnection. In *International Workshop on RFID Technology (IWRT)*, Funchal, Madeira, Portugal.

Lawrence, K., C. Kaler, A. Nadalin, R. Monzillo, and P. Hallam-Baker (2006, February). Web Services Security: SOAP Message Security 1.1. OASIS.

Lee, G., J. Shin, D. Park, and H. Kwon (2008, December). Discovery Architecture for the Tracing of Products in the EPCglobal Network. In *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, Volume 2, Shanghai, China, pp. 553–558.

Li, Y. and X. Ding (2007, March). Protecting RFID Communications in Supply Chains. In *2nd ACM Symposium on Information, Computer and Communications Security*, ASIACCS, Singapore, pp. 234–241. ACM.

Linkies, M. and F. Off (2006, June). *SAP Security and Authorizations*. SAP Press. ISBN: 978-1592290628.

Liu, A., F. Chen, J. Hwang, and T. Xie (2010, December). Designing Fast and Scalable XACML Policy Evaluation Engines. *IEEE Transactions on Computers* (99), 1802–1817. ISSN: 0018-9340.

Manola, F. and E. Miller (2004). RDF Primer. W3C.

Müller, J., J. Oberst, S. Wehrmeyer, J. Witt, A. Zeier, and H. Plattner (2010, January). An Aggregating Discovery Service for the EPCglobal Network. In *43rd Hawaii International System Sciences Conference (HICSS)*, Honolulu, HI, USA, pp. 1–9.

Murthy, K. and C. Robson (2008, May). A model-based comparative study of traceability systems. In *International Conference on Information Systems, Logistics and Supply Chain (ILS)*, Madison, WI, USA.

Neuman, B. C. (1994). Scale in distributed systems. *Readings in Distributed Computing Systems - IEEE Computer Society Press*, 463–489.

ODIN (2009, May). RFID Tag Pricing Guide. Technical report, ODIN Technologies.

Papakonstantinou, V., M. Michou, I. Fundulaki, G. Flouris, and G. Antoniou (2012). Access control for RDF graphs using abstract models. In *17th ACM symposium on Access Control Models and Technologies*, SACMAT, Newark, NJ, USA, pp. 103–112. ACM.

Pardal, M. L., M. Harrison, and J. A. Marques (2012, April). Assessment of Visibility Restriction Mechanisms for RFID Data Discovery Services. In *IEEE International Conference on RFID*, Orlando, FL, USA, pp. 7.

Pardal, M. L., M. Harrison, S. Sarma, and J. A. Marques (2012a, November). Enforcing RFID Data Visibility Restrictions Using XACML Security Policies. In *IEEE International Conference on RFID Technology and Applications*, Nice, France.

Pardal, M. L., M. Harrison, S. Sarma, and J. A. Marques (2012b, November). Performance Assessment of XACML Authorizations for Supply Chain Traceability Web Services. In *8th International Conference on Next Generation Web Services Practices (NWeSP)*, São Carlos, Brazil.

Pardal, M. L. and J. A. Marques (2011, September). Cost Model for RFID-based Traceability Information Systems. In *IEEE International Conference on RFID Technology and Applications*, Sitges, Barcelona, Spain.

Parducci, B., H. Lockhart, and E. Rissanen (2013, January). eXtensible Access Control Markup Language (XACML) Version 3.0.

Perdigão, C. and M. L. Pardal (2010, June). EPC Virtual Lab: Experiments using an RFID location simulator. In Q. Z. Sheng, A. Mitrokotsa, S. Zeadally, and Z. Maamar (Eds.), *4th International Workshop on RFID Technology - Concepts, Applications, Challenges (IWRT)*, Funchal, Madeira, Portugal, pp. 107–112. SciTePress.

Polytarchos, E., S. Eliakis, D. Bochtis, and K. Pramatari (2010). Evaluating Discovery Services Architectures in the Context of the Internet of Things. In D. C. C. Ranasinghe, Q. Z. Z. Sheng, and S. Zeadally (Eds.), *Unique Radio Innovation for the 21st Century*, pp. 203–227. Springer Berlin Heidelberg. ISBN: 978-3642034619.

Rizzi, A., P. Simonazzi, and R. Vitulli (2012, February). Design, Implementation and In-Field Testing of an Original Discovery Service for EPC Network Infrastructure. *Data Collection*.

Robson, C., Y. Watanabe, and M. Numao (2007, April). Parts Traceability for Manufacturers. In *IEEE 23rd International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, pp. 1212–1221.

Rodgers, D. (2011, May). U.S. Pharma Supply Chain Complexity. *RxTrace*.

Rodgers, D. (2012, November). The Significance of the Abbott, McKesson and VA Pilot. *RxTrace*.

Ross, S. M. (2009). *Introduction to Probability and Statistics for Engineers and Scientists - 4th edition*. Academic Press. ISBN: 978-0123756862.

Sandhu, R. and P. Samarati (1994, September). Access Control: Principle and Practice. *IEEE Communications Magazine 32*(9), 40–48. ISSN: 0163-6804.

Schläger, C., M. Sojer, B. Muschall, and G. Pernul (2006, September). Attribute-Based Authentication and Authorisation Infrastructures for E-Commerce Providers. In K. Bauknecht, B. Pröll, and H. Werthner (Eds.), *E-Commerce and Web Technologies*, Volume 4082 of *Lecture Notes in Computer Science*, pp. 132–141. Springer Berlin Heidelberg. ISBN: 978-3540377436.

Schoenemann, N., K. Fischbach, and A. Manteuffel (2009, December). Flexible Semantic Services to Facilitate Innovative and Dynamic Ubiquitous Supply Chain Networks. In *2nd International Conference on Computer Science and its Applications (CSA)*, Jeju Island, Korea, pp. 1–5.

Schoenemann, N., K. Fischbach, and D. Schoder (2009, June). P2P Architecture for Ubiquitous Supply Chain Systems. In *17th European Conference on Information Systems (ECIS)*, Verona, Italy.

Schuster, E. W., S. J. Allen, and D. L. Brock (2007). *Global RFID: The value of the EPCglobal Network for Supply Chain Management*. Springer. ISBN 978-3540356547.

Shi, J., Y. Li, W. He, and D. Sim (2012). SecTTS: A secure track & trace system for RFID-enabled supply chains. *Computers in Industry 63*(6), 574 – 585. ISSN: 0166-3615.

Shi, J., D. Sim, Y. Li, and R. Deng (2012, February). SecDS: a secure EPC discovery service system in EPCglobal network. In *2nd ACM Conference on Data and Application Security and Privacy*, CODASPY, San Antonio, TX, USA, pp. 267–274.

Song, S., T.-K. Shim, and J.-H. Park (2006, October). Proxy based EPC Track & Trace Service. In *IEEE International Conference on e-Business Engineering (ICEBE)*, Shanghai, China, pp. 528–531.

Supply Chain Council (2007). Design-chain operations reference model. Technical report, Supply Chain Council.

Tanenbaum, A. S. (2002). *Computer Networks* (4th ed.). Prentice Hall. ISBN: 978-0130661029.

Tanenbaum, A. S. and M. van Steen (2007). *Distributed Systems - principles and paradigms* (2nd ed.). Prentice Hall. ISBN: 978-0132392273.

Tearnen, P. (2005, September). EPCglobal US Discovery Service Proof of Concepts. EPCglobal US Conference.

Thiesse, F., C. Floerkemeier, M. Harrison, F. Michahelles, and C. Roduner (2009). Technology, Standards, and Real-World Deployments of the EPC Network. *IEEE Internet Computing 13*(2), 36–43. ISSN: 1089-7801.

Traub, K., F. Armenio, H. Barthel, P. Dietrich, J. Duker, C. Floerkemeier, J. Garrett, M. Harrison, B. Hogan, J. Mitsugi, J. Preishuber, J. Preishuber-Pfluegl, O. Ryaboy, S. Sarma, K. Suen, and J. Williams (2010, December). The EPCglobal Architecture Framework 1.4. EPCglobal.

Traub, K. and S. Sarma (2007, August). Framework for Multi-Party Data Exchange. Technical report, EPCglobal. Working draft.

Turkmen, F. and B. Crispo (2008). Performance evaluation of XACML PDP implementations. In *ACM Workshop on Secure Web Services*, SWS, Fairfax, VA, USA, pp. 37–44. ACM.

Uckelmann, D., M. Harrison, and F. Michahelles (Eds.) (2011). *Architecting the Internet of Things*. Springer. ISBN: 978-3642191565.

Verisign (2008). The EPCglobal Network: Enhancing the Supply Chain. Technical report, Verisign.

Verma, M. (2004, January). XML Security: The XML Key Management Specification. *IBM Developer Works*.

Vollbrecht, J., P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence (2000, August). RFC 2904 – AAA Authorization Framework. IEFT.

Wakayama, S., Y. Doi, S. Ozaki, and A. Inoue (2007, March). Cost-effective Product Traceability System Based on Widely Distributed Databases. *Journal of Communications 2*(2), 45–52. ISSN: 1796-2021.

Weinstein, R. (2005). RFID: a technical overview and its application to the enterprise. *IT Professional 7*(3), 27–33.

Williams, J. R. and A. Sanchez (2007). Supply chain realms with data streams and location services. In *EU RFID 2007 Academic Convocation*. Auto-ID Laboratory, MIT.

Worapot, J., Y. Li, and A.-I. Somjit (2010, June). Design and implement of the EPC Discovery Services with confidentiality for multiple data owners. In *IEEE International Conference on RFID Technology and Applications (RFID-TA)*, Guangzhou, China, pp. 19–25.

Wu, Y., D. Ranasinghe, Q. Z. Sheng, S. Zeadally, and J. Yu (2011, October). RFID Enabled Traceability Networks: A Survey. *Distributed and Parallel Databases 29*(5–6), 397–443. ISSN: 0926-8782.

Yavatkar, R., D. Pendarakis, and R. Guerin (2000, January). RFC 2753 – A Framework for Policy-based Admission Control. IEFT.

Young, M. (2008, August). Extensible supply-chain discovery service concepts. IEFT.

Ziekow, H. and O. Günther (2010, November). Sharing RFID and Complex Event Data among Organizations. *Information Systems Frontiers 12*(5), 541–549. ISSN: 1387-3326.

B GS1 identification system

This appendix briefly provides details about the GS1 identifiers.

B.1 Common identifier components

GS1 defines a 3 digit code that can be used as a *country code* or as a *special code*.

Example country code prefixes:

- 000 - 019, 060 - 099 USA and Canada
- 300 - 379 France and Monaco
- 400 - 440 Germany
- 450 - 459, 490 - 499 Japan
- 460 - 469 Russia
- 500 - 509 United Kingdom
- 560 Portugal
- 690 - 695 China
- 754 - 755 Canada
- 760 - 769 Switzerland and Liechtenstein
- 789 - 790 Brazil
- 800 - 839 Italy, San Marino and Vatican City
- 840 - 849 Spain and Andorra
- 880 South Korea
- 890 India
- 930 - 939 Australia
- 940 - 949 New Zealand

Every country in Europe is on the GS1 system. There are around 200 countries in the world and there are enough free number ranges for countries still out of the system.

Special ranges prefixes:

- 020 - 029 Restricted distribution (GS1 Member Organization defined)
- 030 - 039 U.S. drugs (see U.S. National Drug Code)
- 040 - 049 Restricted distribution (GS1 Member Organization defined)
- 050 - 059 Coupons
- 200 - 299 Restricted distribution (GS1 Member Organization defined)
- 950 Head Office - used for several special applications and bi-lateral agreements. As an example, prefix 9509999 has been allocated by GS1 to the United Nations International Drug Control Programme UNDCP in 1995.
- 977 Serial publications International Standard Serial Number (ISSN)
- 978 - 979 Bookland International Standard Book Number (ISBN)
- 9790 International Standard Music Number (ISMN)
- 980 Refund receipts
- 981 - 982 Common Currency Coupons
- 990 - 999 Coupons

GS1 identifiers use the *Company Prefix* assigned to the organization. The Company Prefix provides a way for GS1 Member Organizations (MO) to uniquely and globally identify things like trade items, logistic units, locations, parties, and assets.

The Company Prefix has varying length. In bar-codes it appears with usually 4 or 5 digits. For radio-frequency tags it can range from 6 to 12 digits.

A UPC (Universal Product Code) company prefix can be converted to a GS1 Company Prefix by adding a leading zero.

B.2 Identifiers

A *GTIN (Global Trade Item Number)* is used to identify any item upon which there is a need to retrieve predefined information and that may be priced or ordered or invoiced at any point in a supply chain.

A separate unique GTIN is required whenever any of the predefined characteristics of a product are different in any way that is relevant to the trading process. The guiding principle is: if the customer is expected to distinguish a new trade item from an old trade item, and purchase accordingly, a new GTIN should be assigned.

GTINs have four numbering structures: GTIN-8, GTIN-12, GTIN-13 and GTIN-14. All GTIN can be formatted using 14 digits and leading zeros. The last digit is *always* used as a check digit.

A *SGTIN (Serialized GTIN)* is the combination of a GTIN with a unique serial number to create a unique identifier for individual trade items.

A *GLN (Global Location Number)* is used for location: physical, functional or legal entities requiring a permanent identification, such as a company, department, or warehouse. A GLN has 14 digits, including: company prefix, location reference and 1 check digit. Within the GS1 system, high capacity data carriers use Application Identifiers (AI) to distinguish data elements encoded within a single data carrier. The GLN can be associated with many AI's including: physical location, ship to location, and invoice to location.

A *SGLN (Serialized GLN)* represents only the physical location sub-type of GLN AI 414. The serial component is represented by the GLN Extension AI 254.

A *SSCC (Serial Shipping Container Code)* is used for logistic units: physical units established for transport and storage of products of any kind that need to be tracked and traced individually in a supply chain.

A *GRAI (Global Returnable Asset Identifier)* is used for returnable assets.

A *GIAI (Global Individual Asset Identifier)* is used for fixed assets.

A *GSRN (Global Service Relation Number)* is used for service relations by public or private service providers to track any entity's service requirements and needs over a continuing relationship.

A *GDTI (Global Document Type Identifier)* is used to identify a document by type and can uniquely identify it where required.

C RFID technology

This appendix contains a presentation of the working principles of radio frequency identification (RFID) technology. RFID is an automatic data capture technology that has great potential to improve business by tagging interesting physical objects and allowing them to be detected automatically. RFID readers collect data statements such as: "Object O was seen at time T and place L." and "Object O was aggregated into pallet P".

The crucial difference of RFID when compared to other radio-frequency technologies, like WLAN[1] and Bluetooth[2], is that the transponder relies on the reader for its power. Finkenzeller and Muller [2010] is a good starting point for an in-depth study of RFID technology. Landt [2005] provides an historical account of the technology's development.

C.1 Reader

RFID communication occurs between readers and transponders (tags), using a wireless power and data link. First the reader sends commands, then the tag responds.

When an RFID reader transmits energy, it activates all transponders in range and the responses are sent by all at the same time, causing signal collisions. Since the transponders do not hear the signals from other transponders, and they can only listen to the reader's signal, it is up to the reader to prevent collisions by following an anti-collision protocol like using predefined slots or assigning turns.

RFID is not a single technology but a suite of technologies. A major distinction in RFID is the choice of operating principle, between "inductive coupling in the electromagnetic near-field with load modulation at LF/HF" or "wave coupling in the electromagnetic far-field with back-scatter at UHF/MW". RFID using inductive coupling is only practical in the near-field[3]. whereas RFID using wave coupling typically uses the far-field[4].

Inductive coupling works on a electrical transformer principle. The transponder talks back to the reader using load modulation i.e. by interfering with the whole system energy. Inductive coupling is good for short read ranges – up to 1 meter – and has a greater ability to penetrate objects and to operate in metallic environments. The most common frequencies are in the ranges LF (Low Frequency - 30 to 300 kHz) to HF (High Frequency - 3 to 30 MHz).

[1] Wireless Local Area Network. IEEE standard 802.11.

[2] IEEE standard 802.15.1.

[3] The *near-field* is an energy storage field, where energy is preserved, and moves from the capacitor to the circuit.

[4] The *far-field* is an energy propagation field, where electromagnetic waves propagate and would go on forever, were it not for the absorption losses.

Wave coupling works on a radar principle i.e. the reader sends a signal and the transponder talks back to the reader using interference. Wave coupling allows much greater reading ranges – up to 100 meters – even though the readings are more erratic because of destructive wave interference. The most common frequencies are in the ranges UHF (Ultra-High Frequency - 300 to 3 000 MHz) and MW (Microwaves - 2.5 to 5.8 GHz).

RFID uses ISM (Industrial, Scientific, and Medical) radio bands. Radio communication services operating within these bands must accept harmful interference, which may be caused by other applications. The available ISM frequencies are different in Europe, Americas, and Asia. The choice of RFID readers and tags has to take these differences into account because there is a trade-off in manufacturing: tags designed for use in a particular geography will typically outperform a global tag, because tags either have a high performance at a narrow frequency band or lower performance in a wider frequency range.

C.2 Tag

There are three major categories of tags:

- *Passive* or *battery-less* – use only power provided by the RFID reader's signal;
- *Semi-passive* or *battery-assisted* – use a battery to boost response signal and/or collect sensor readings;
- *Active* or *battery-powered* – have more power available that allows for additional range, processing capabilities, and autonomy.

A tag is composed by: integrated circuit (IC); antenna; connection between the IC and the antenna; and substrate on which the antenna resides. Optionally, a tag can be protected to endure rough environments. The manufacture process has the following steps:

1. Manufacture of the IC;
2. Manufacture of the antenna (the conductive element is shaped to a specific configuration);
3. Assembly of the IC to the antenna;
4. Conversion to package: the antenna with the IC is attached, first to substrate, and then to a package.

The tag cost is important for the adoption of RFID technology because it multiplies by the number of objects being tracked. Tag costs range from less than USD 0.1 to USD 10, and costs are expected to continue falling [ODIN, 2009].

D Supply chain model

This appendix details the supply chain model used for the cost assessment described in Chapter 3. The model uses *graphs* [Cormen et al., 2009] to represent the item paths and the supply chain. A graph is defined by a set of vertices (V) and a set of edges (E), each connecting two vertices. In the model, a vertex is a node in the supply chain, corresponding to a company, while an edge represents a connection between companies. The model is presented next, first without considering aggregation, and then considering it.

D.1 Item and chain graphs

Directed Acyclic Graphs (DAG) are used to represent item flows and supply chains. A DAG is a graph with directed edges, and without cycles. A DAG's in-degree is the number of incoming edges, and its out-degree is the number of outgoing edges. A vertex with in-degree of 0 is called a begin-vertex. A vertex with out-degree of 0 is called a end-vertex.

Item graph: An item flowing in a supply chain defines a DAG. The vertices represent companies, and the edges represent the item flow between companies. All vertices of an item DAG are connected by edges and each vertex has, at most, in-degree of 1 and, at most, out-degree of 1. There is a single begin-vertex and a single end-vertex. Figure D.1 presents the item DAG for an object A that goes from Manufacturer 1 to Distributor 1 and to Retailer 1. Figure D.2 presents the item DAG for an object B that starts at Manufacturer 2, goes to Distributor 1, and then to Retailer 2.

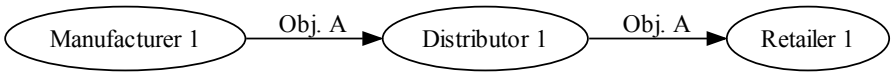

Figure D.1: Item-defined graph.

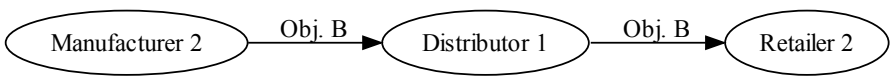

Figure D.2: Another item-defined graph.

Chain graph: Each item flowing in the supply chain defines an item DAG of its own. A chain DAG can be defined from the set of item DAGs. The vertices of the chain DAG are defined by the union of item DAG vertices. The edges of the chain DAG are defined by the union of item DAG edges. An example chain DAG is represented in Figure D.3. It combines the graphs of objects A and B, presented earlier in Figures D.1 and D.2, respectively.

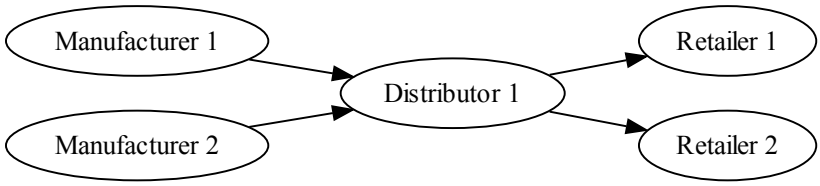

Figure D.3: Chain-defined graph.

The traceability queries can be formulated using the graph definitions:

- *Track query* – given a chain DAG and an item, find the vertex in the item DAG with the highest topological ordering i.e. find the vertex furthest ahead in the item DAG;
- *Trace query* – given a chain DAG and an item, recover the item DAG;

The *Bill-of-Materials (BoM)* query formulation requires aggregation.

D.2 Aggregation

An *aggregation* is a whole-part association between two physical objects: the aggregate and the component. The aggregation can be made for *transportation* purposes – the aggregate carries the component – or for *manufacturing* purposes – the component is assembled to the aggregate. The relationship $agg(a, c)$ is defined to mean that object a (the aggregate) aggregates object c (the component). The recursive relationship $aggr(a, c)$ is defined to mean that either $agg(a, c)$ holds or that there is another object $a2$ such that both $agg(a, a2)$ and $aggr(a2, c)$ hold.

An *aggregated-item DAG* for object i has vertices for companies and the edges are defined by the flow between companies of the item i or of an aggregate a such that $aggr(a, i)$. All aggregated-item DAGs have vertices with out-degree of, at most, 1.

Figure D.4 presents the *transported-item DAG* for object A, an aggregated-item DAG where all vertices have an in-degree of, at most, 1. Object A was aggregated to object C during the flow from Distributor 1 to Distributor 2.

Figure D.5 presents the *assembled-item DAG* for object C, an aggregated-item DAG where all vertices have an in-degree greater or equal to 0, determined by the assembly. Objects A and B were aggregated on object C at the Manufacturer 1 node.

The traceability query formulations can now be extended to include aggregation:

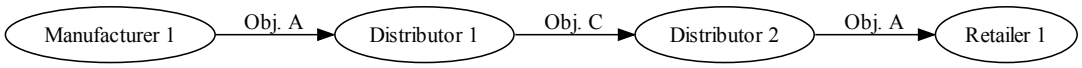

Figure D.4: Transported-item graph.

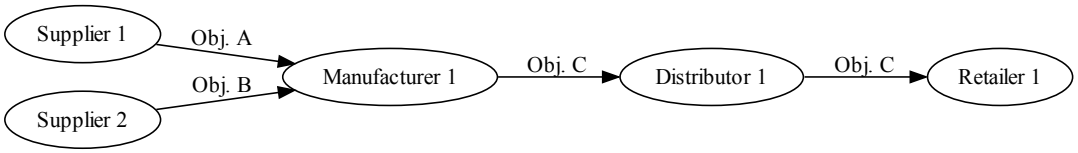

Figure D.5: Assembled-item graph.

- *Track query with aggregation* – given a chain DAG and an item, find the vertex in the transported-item DAG with the highest topological ordering;
- *Trace query with aggregation* – given a chain DAG and an item, retrieve the transported-item DAG defined by the item and by all of its containers;
- *Bill-of-Materials (BoM) query* – given a chain DAG and an item, retrieve the assembled-item DAG defined by the item and by all of its parts.

E Discovery service prototype

This appendix presents the Discovery Service (DS) prototype that was implemented early in the research effort to gain insight into the unspecified DS component of the EPC framework.

E.1 Specification

An EPC DS provides links to EPC IS repositories that have traceability data about specific physical objects. Each trading partner in the supply chain publishes link records to a DS and also specifies access control policies to restrict who has visibility of link information. The clients of the DS are the companies that want to access detailed data about the physical objects.

The DS prototype implementation followed a draft proposal by Traub and Sarma [2007] of a directory-based service with a push model, meaning that the trading partners pro-actively provide information to the DS. The DS prototype uses assertions to enable both discovery and access control, as illustrated in Figure E.1 where the DS answers a *WhoHasData(x)* request using the facts stated in assertions and stored in the directory. After the query is answered, the querying party knows which EPC IS instances to contact with *getEvents(x)* requests.

E.1.1 Assertions

Assertion are formal representations of logical conditions and, in the prototype, specify both discovery and authorization constraints. The model defines the following assertions:

- *Event(A, x, data)*: denotes an EPC IS event asserted by Participant A about EPC x;
- *HasData(A, x)*: denotes an assertion by Participant A that it possesses event data pertaining to EPC x;
- *WillShare(A, x, B)*: denotes an assertion by Participant A that it is willing to share data about EPC x with Participant B, without granting B any rights to share that data with others;
- *WillSharePropagate(A, x, B)*: denotes an assertion by Participant A that it is willing to share data about EPC x with Participant B, and with other participants that B may designate.

The *HasData* assertions enable discovery; *WillShare* assertions specify authorization constraints; and finally *WillSharePropagate* assertions allow delegation of authorization rights to other supply chain participants.

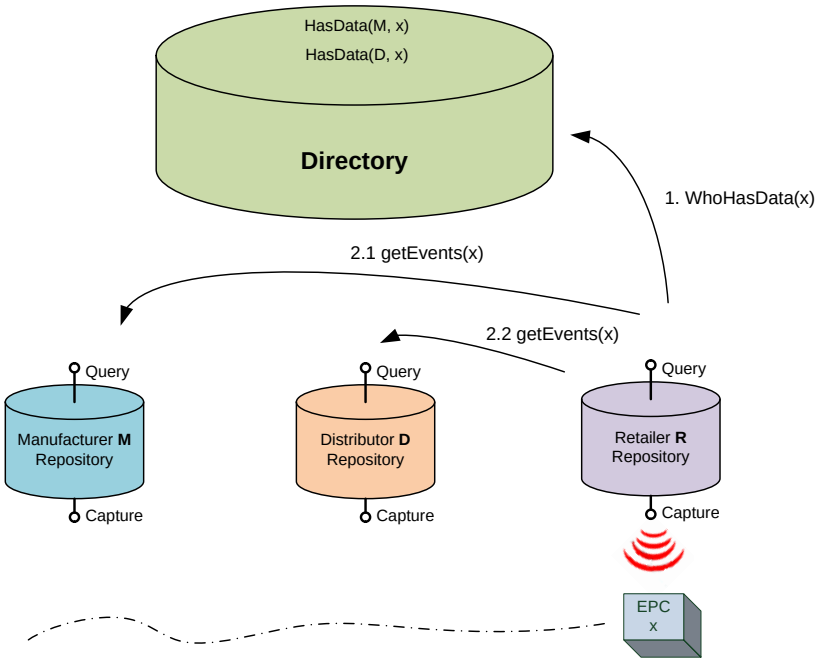

Figure E.1: Data discovery "driven" by assertions in a supply chain.

E.2 Implementation

The implementation of the DS – including the assertion repository and the assertion inference engine – was developed using the Java programming platform[1] and the MySQL relational database[2], both widely used technologies. For simplification, there was a single DS instance, and there was no message-level security, so there was no actual protection of confidentiality and integrity of the exchanged data, as would be required in a production implementation.

The development process followed standard practices and built a layered architecture [Fowler et al., 2002], to separate the implementation of different concerns: domain representation, persistence, and service invocation.

The prototype successfully showed that the functional requirements of DS are not very complex to implement and that assertions could be a practical approach for specifying authorization constraints.

[1] http://www.java.com
[2] http://www.mysql.com

F Externalized security

This appendix describes an externalized security architecture and the XACML authorization standard in detail. The goal of the externalized security architecture is to unify the security management across applications so that business rules can be applied consistently and changed dynamically. Externalized security encompasses user management, authentication, authorization, logging and auditing.

F.1 Standards

The core of security for enterprise applications is that company users have to be authenticated and data access must be authorized. Standards play an important role. To start, XKMS can be used for cryptographic key management [Verma, 2004].

XML-Signature [Eastlake et al., 2002] and XML-Encryption [Eastlake et al., 2002] allow XML data signature and cipher, respectively, and both can be applied selectively to parts of messages or to external contents. XAdES (XML Advanced Electronic Signatures) [Cruellas et al., 2003] extend XML signatures to allow non-repudiation and long-term storage, using secure time-stamping services.

Schläger et al. [2006] derived a pattern system able to express a generic attribute-based authentication and authorization infrastructure using standards like SAML and XACML. Hebig et al. [2009] further demonstrated how these standards can work together in practice. The authentication can be achieved with identity providers [Baier et al., 2013] and the exchange of SAML assertions [Cantor et al., 2005]. The authorization can be achieved with XACML [Parducci et al., 2013] that is an XML vocabulary to represent authorization policies and requests. XACML can avoid rules hard-coded in applications and improve the consistency of policy enforcement. XACML is discussed in detail in the next Section.

F.2 eXtensible Access Control Markup Language

XACML is a standard proposed by OASIS [Parducci et al., 2013] that represents authorization policies and requests. The standard follows a processing model and defines a policy format.

F.2.1 Processing model

The processing model that XACML assumes is defined by RFC 2753 [Yavatkar et al., 2000] and 2904 [Vollbrecht et al., 2000], and defines the following structural elements:

- PAP – Policy Administration Point;
- PEP – Policy Enforcement Point;
- PDP – Policy Decision Point; and
- PIP – Policy Information Point.

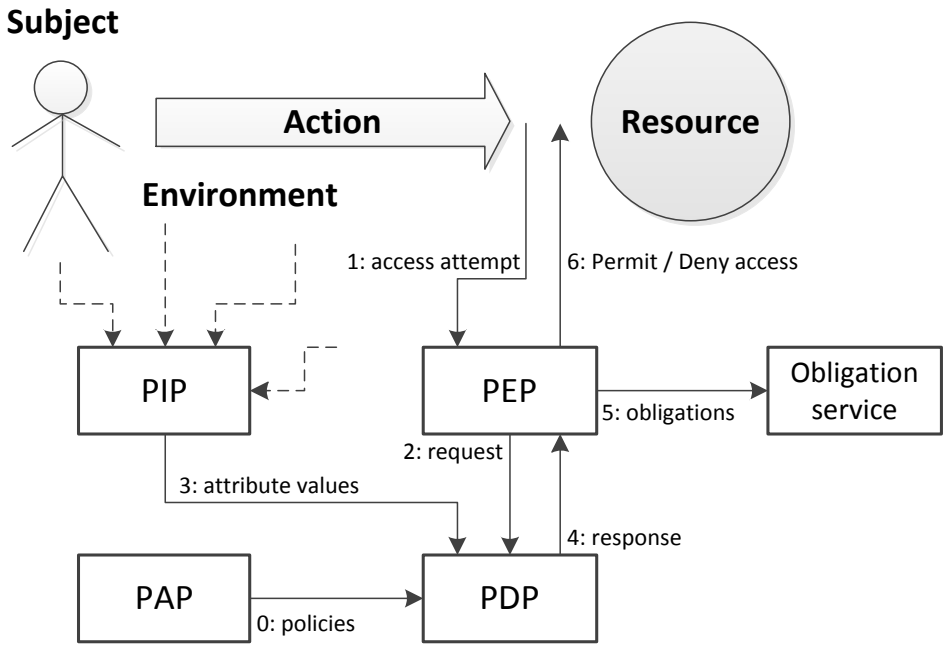

Figure F.1: XACML request processing.

The PAP is used to author and manage policies, and they are loaded before authorization requests can be accepted in step 0 of Figure F.1. An access attempt is intercepted by PEP in step 1 and an access request is sent to the PDP in step 2. The PIP provides attribute values, if necessary in step 3 and the PDP makes a decision in step 4. Any implied obligations are serviced in step 5 and the action is permitted or denied in step 6.

A XACML authorization request contains attributes about:

- *Subjects* – entities requesting access, with one or more attributes;
- *Resources* – data or service or component, with a single attribute;
- *Actions* – type of access requested, restricted to a single action, with one or more attributes;
- *Environment* – other attributes, like the current date and time.

F.2.2 Policy format

The XACML document structure is represented in Figure F.2. The document can contain a single policy or (nested) policy sets. A policy consists of a set of obligations, a target, a set of rules, and a rule-combining algorithm.

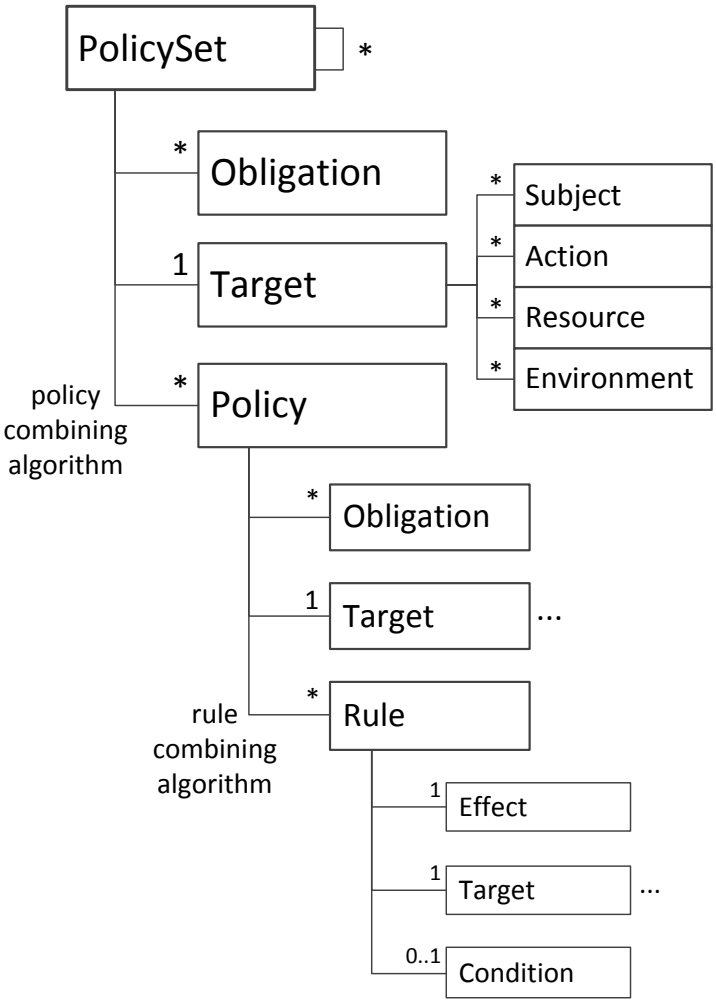

Figure F.2: XACML policy structure.

Obligations are actions that must be executed when a request is processed and are typically used to write audit logs. The *target* defines a simplified set of conditions that determine if the policy is relevant for the request and provides policy index keys.

Rules are conditions that evaluate to 'Permit', 'Deny', 'NotApplicable' (when no target matches), or 'Indeterminate' (when internal errors occur). A policy rule is composed of: effect, target, and conditions. The rule *target* again determines if the rule is relevant for the request. The rule *conditions* are statements about attributes with arbitrary nesting of functions[1] that, upon evaluation, return either 'True', 'False', or 'Indeterminate'. If a condition evaluates to 'Indeterminate', the rule returns to 'Indeterminate'. If a condition evaluates to 'False', the rule returns 'NotApplicable'. If a condition evaluates to 'True', rule returns the value of effect. The rule effect is the intended consequence of the rule – 'Permit' or 'Deny' – when the condition returns 'True'.

Finally, *policy/rule combining algorithms* are responsible for reconciling conflicts between policies/rules and to arrive at one outcome per policy per request using logical conjunction, disjunction or other algorithms.

F.2.3 Implementation survey

There are several XACML libraries available, both open-source and commercial. The open-source implementations were surveyed in 2012 and the findings are presented in Table F.1. Butler et al. [2010] evaluated *correctness* by comparing responses from different XACML implementations. Turkmen and Crispo [2008] compared *performance* of the 'policy loading' and 'request evaluation' operations of XACML implementations, and reported loading issues with more than one hundred policies. Liu et al. [2010] discussed performance optimization techniques and showed that several improvements are possible.

F.2.4 Policy translation survey

Policy translation approaches are presented by several authors. The policy is defined first using a higher level representation and is then translated to XACML for standard representation and enforcement. Karjoth et al. [2008] converted a vendor-specific policy format to XACML that included ACLs. Alm and Illig [2010] translated complex policies such as 'Role-Based Access Control' and 'Separation of Duty'. Brucker et al. [2012] demonstrated how role-based access control, separation of duty, and binding of duty requirements can be specified in SecureBPMN and then automatically translated into XACML policies and enforced by one or more generated PEPs.

[1] XACML predefines the functions that are available for use. Custom functions can be provided by the implementation but at the cost of breaking compatibility in the interpretation of the policy.

Impl.	Version	Last update	Sponsors	Comments
sunxacml	1.2	Jul 2004	Sun Microsystems (currently Oracle)	Reference implementation but has not been updated since 2004.
HERAS-AF	1.0.0-M2	Sep 2010	U. Applied Sciences, Rapperswil, Switzerland	Well documented but only has in-memory policy repository.
enterprise-java-xacml	r258	Jan 2009	Zian Wang	Best performance reported by Turkmen and Crispo [2008] but insufficient documentation available.
PicketBox	3.0.0.Final	Feb 2011	JBoss, Red Hat	Again, insufficient documentation available.
xEngine	beta 0.2	Aug 2010	Michigan St. U., North Carolina St. U.	Best performance reported by Liu et al. [2010] but insufficient documentation available.

Table F.1: Open-source XACML implementations.

G Linked data

This Appendix provides a summary of Linked Data technology. Linked Data is the underlying distributed data model of the Semantic Web [Allemang and Hendler, 2011] and it can be used to represent data access policies, as discussed in Chapter 5.

The typical Linked Data application architecture, depicted in Figure G.1, extends a database application architecture. The persistent RDF store merges the information and applies models to infer new data and validate conditions. Data input converters transform data from other formats – web pages, spreadsheets, tables, databases – to RDF. The query engine is implemented using SPARQL. The application interface uses the contents of an RDF store in interactions with end-users.

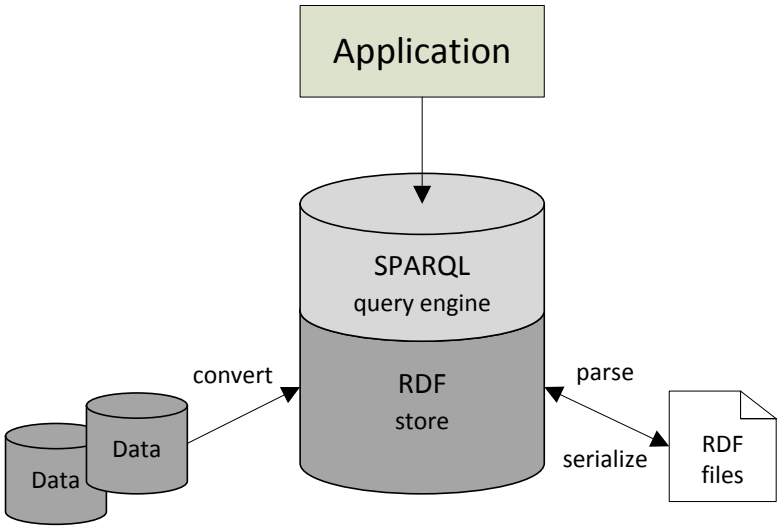

Figure G.1: Linked Data application architecture, adapted from [Allemang and Hendler, 2011].

G.1 The Semantic Web

The Semantic Web is intended to be an *organized* worldwide system where information flows in a smooth and orderly way. However, the **AAA** slogan – **A**nyone can say **A**nything about **A**ny topic – is intrinsic to the design of the Semantic Web. This means that, in practice, there can be many contradictions and inconsistencies in the data that make it hard to use effectively. As a result, different sources, organizations, and styles of information need to co-exist, and semantic modeling tools are required to build models that make data usable and useful in this context.

G.1.1 RDF

At the core of the distributed data model is the Resource Description Framework (RDF) that is a universal data model that represents data structures as triples [Manola and Miller, 2004]. Each triple – subject, predicate, and object – can be represented in graph format. For instance, the triple ':company0 cta:publishes :record0' is represented in Figure G.2.

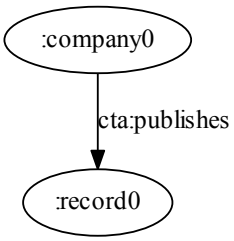

Figure G.2: RDF triple represented as a graph.

G.1.2 SPARQL

The SPARQL[1] is a query language whose purpose partly resembles the Structured Query Language (SQL) widely used in relational databases [Harris and Seaborne, 2012]. SPARQL is dedicated to finding matches for RDF statements that may contain variables rather than extracting specific values from table records.

G.2 Linked data for security survey

Linked Data technologies – RDF, OWL and SPARQL – provide open-ended data representation and querying. The OWL versus SPARQL inference approaches contrast declarative with procedural knowledge representation [Brachman and Levesque, 2004]. Declarative knowledge is "knowing that" and, in this case, is expressed by OWL statements. Procedural knowledge is "knowing how" and, in this case, is expressed by SPARQL procedures.

There is significant work where Linked Data technologies are used for security purposes. Cadenhead et al. [2012] describes a general-purpose, scalable RDF policy engine that includes support for a diverse set of security policies. The policy engine was evaluated as being highly available and scalable. Finin et al. [2008] show different ways to support the NIST standard Role-Based Access Control (RBAC) model in Web Ontology Language (OWL) and then discuss how the OWL constructions can be extended to model attribute-based RBAC or more generally attribute-based access control. Ferrini and Bertino [2009] start from XACML that does not natively support RBAC and introduce XACML+OWL, a framework that integrates OWL ontologies and XACML policies for supporting RBAC. It decouples the design by modeling the role hierarchy and the constraints with an OWL ontology and the authorization policies with XACML. Papakonstantinou et al. [2012] use 'quadruples' to encode access labels for RDF triples, representing information such as time, trust, and provenance. The authors make use of the SPARQL language to determine the triples that define these labels.

[1]SPARQL is a recursive acronym that stands for SPARQL Protocol and RDF Query Language.