Cost Model for RFID-based Traceability Information Systems

Miguel L. Pardal and José Alves Marques
Department of Computer Science and Engineering
Instituto Superior Técnico, Technical University of Lisbon, Portugal
Email: miguel.pardal@ist.utl.pt, jose.marques@link.pt

***Abstract*—Radio Frequency Identification (RFID) is a technology that can be used to tag physical objects and to detect and identify them automatically along the supply chain.**

A RFID-based traceability information system uses the captured data to answer track, trace, and bill-of-materials queries. There are several published system proposals, but it is unclear how solutions for a given supply chain problem can be compared.

This paper presents an analytical model to compare traceability information systems based on the estimated cost of data capture and query processing.

I. Introduction

Radio Frequency identification (RFID) technology [1] can be used to automatically identify tagged physical objects and to collect attributes such as location and time. RFID data has the potential to greatly improve business processes as it can be used to answer traceability queries [2], like the following:

- **Track/Recall query**:
  What is the current location of the 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?

However, RFID data is distributed by nature because it is captured at disperse locations by different organizations. A traceability information system has to be designed to effectively collect the data required to answer track, trace, and BoM queries.

The goal of our current research effort is to quantitatively estimate the cost of traceability information systems. Our *assessment process* is the following:

- **Model a supply chain domain** - specifying industry, product classes, volumes, queries, historic data needs, etc;
- **Model candidate system solution** - specifying architecture components and message exchange patterns;
- **Compute results**;
- **Validate model** - using either simulations or measurements;
- **Further parametrize the model** - using domain-specific information.

This paper presents a *traceability information system cost model*. The model estimates total system cost from the cummulative processing and communication costs of data capture and query execution. We opted for analytical cost modeling because it allows a simple comparison of different approaches without requiring actual system implementations. Our modeling approach is based on previous work by Murthy and Robson [3] that abstracted out the characteristics of two IBM systems and built a simple query execution model to compare the performance.

In the next section we present the related work of standards and existing system proposals. Then, we describe the parameters, assumptions, and formulae of the cost model followed by example results. Finally, we draw conclusions and point future work directions.

II. Related work

A. Standards

The GS1 EPCglobal architecture [4] is a set of standards that define the building blocks - hardware, software, and data - for a supply chain traceability information system. At the core of the architecture is the Electronic Product Code (EPC) standard, that defines globally unique identifiers for items in the supply chain. The Application Level Events (ALE) [5] allows client applications to access filtered, consolidated, real-time RFID data. The EPC Information Services (EPCIS) [6] captures data coming from ALE and records events with additional business context. At the top of the EPC stack there is a placeholder for a Discovery Service (DS) standard. A DS is a facilitator service, linking data providers (EPCIS) to data consumers. After the providers are referenced, the consumers can contact them to obtain detailed data.

A DS is just one of the possible approaches to build a traceability information system. There is a market need for such systems as documented in an industry study [7] where more precise requirements were elicited.

B. Survey

There are already several published DS implementation proposals. Evdokimov et al. [8] recently evaluated four of them qualitatively, framing functional requirements and comparing their characteristics using a quality framework based on an ISO standard for software quality.

We surveyed over twenty published proposals for traceability systems and summarize the results in Figure 1. The classification criteria are data integration and centralization, as proposed by Do et al. [9]. The *data integration* criterion considers where data is physically stored. Data can be copied to specific locations - materialized integration - or referenced

- virtual integration. 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.

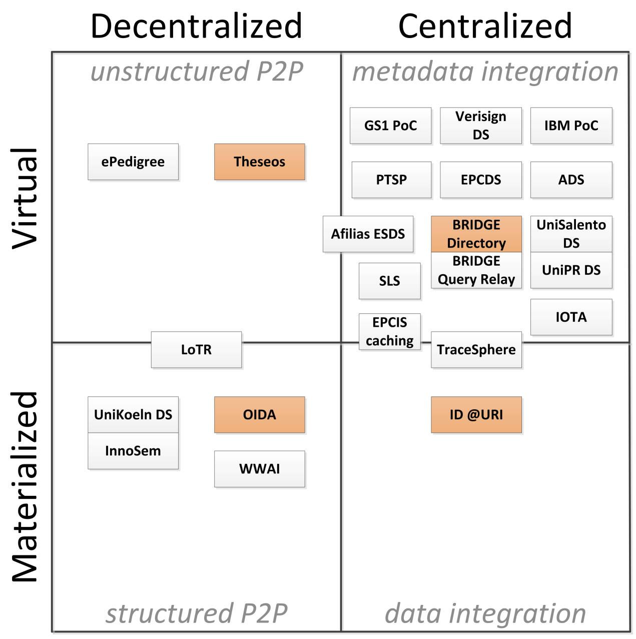


Fig. 1. Traceability information systems proposal classification.

Combining the criteria, we have four distinct approaches and their respective systems:

- **Metadata integration (MDI) approach** (virtual, centralized): GS1 PoC [10], Verisign DS [11], IBM PoC [12], PTSP [13], EPCISDS [14], EPCDS [15], ADS [16], Afilias ESDS [17], BRIDGE Directory [18], UniSalento DS [19], SLS [20], BRIDGE Query Relay [21], UniPR DS [22], EPCIS caching [23], TraceSphere [24], IOTA [25];
- **Data integration (DI) approach** (materialized, centralized): ID@URI [26];
- **Unstructured peer-to-peer (UP2P) approach** (virtual, decentralized): ePedigree [27], Theseos [28];
- **Structured peer-to-peer (SP2P) approach** (materialized, decentralized): LoTR [29], UniKoeln DS [30], OIDA [31], InnoSem [32], WWAI [9].

The MDI approach is clearly the most popular. We highlight one representative proposal in each quadrant of Figure 1: BRIDGE Directory, ID@URI, Theseos, and OIDA. *BRIDGE Directory* [18] relies on centralized services to store data provider links. It is closely aligned with the EPCglobal architecture. *ID@URI* [26] uses a product-agent architecture where all data concerning the item is forwarded to a central data store, managed by the product's manufacturer. *Theseos* [2] has several distributed data stores and queries are answered recursively. Each node can enforce its own data access policy. *OIDA* [31] relies on a Peer-to-Peer (P2P) network with a hashing algorithm for fully decentralized data placement in nodes. However, there are issues about response quality and timeliness, and query capabilities are limited to exact ID matching.

BRIDGE Directory, ID@URI, Theseos, and OIDA are all solid system designs. Each approach was modeled in an abstract representation, so that it could be compared effectively.

III. Cost model

In this section we present the developed cost model. We start by presenting the chain modeling, followed by the parameters, assumptions, and common formulae.

A. Chain modeling

We use directed acyclic graphs (DAGs) [33] to represent product flows and supply chains. A graph is defined by a set of vertices (V) and a set of edges (E), each connecting two vertices. A DAG is a graph with directed edges and without cycles. A DAG can be topologically sorted [33]. 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 0 is a begin-vertex. A vertex with out-degree 0 is a end-vertex. We consider item-defined DAGs and chain-defined DAGs.

1) Item DAG: 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. Each vertex has, at most, in-degree 1 and, at most, out-degree 1. There is a single begin-vertex and a single end-vertex. Figure 2 presents the item DAG for object A. Figure 3 presents the item DAG for object B.

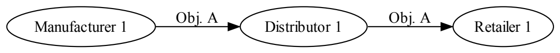


Fig. 2. Item-defined Directed-Acyclic Graph.

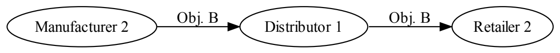


Fig. 3. Another item-defined Directed-Acyclic Graph.

2) Chain DAG: Each item flowing defines an item DAG of its own. At the end of the system's lifetime, we can define a chain DAG 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 4. It combines the item DAGs of objects A and B, presented earlier in Figures 2 and 3, respectively.

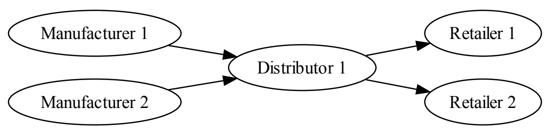


Fig. 4. Chain-defined Directed-Acyclic Graph.

3) Traceability query formulation: The traceability queries can be defined using the graph formulation.

- **Track/Recall query**:
  Find the item DAG vertex with the highest topological ordering (the vertex that is furthest ahead).
- **Trace/Pedigree query**:
  Recover the full item DAG.
- **Aggregation/Bill-of-Materials (BoM) query**:
  Retrieve the item DAGs defined by the item and by all of its parts.

B. Parameters

The cost model's parameters are presented in Table I.

Type	Name	Symbol	Unit
System	Bandwidth	β	bps
System	Processing speed	γ	bps
System	Seek time	θ	s
Application	Message size	μ	bit
Application	Item record size	δ	bit
Chain	Nodes	n	vertex
Chain	Avg. item records per node	r	item record
Chain	Avg. length	z	vertex
Product	Avg. BoM depth	b	level
Product	Avg. children per BoM level	c	node

TABLE I
COMMON PARAMETERS.

The parameters represent characteristics of the system, application, chain, and product. The *number of nodes in the chain* (n) is the number of vertices in the chain DAG. 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 DAG* (z) is the average of the length of all item DAGs defined over the lifetime of the system. 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 accummulated number of components is given by $\sum_{i=0}^{b} c^i$.

C. Assumptions

1) System-level assumptions
   a) System parameters are the same for every node.
   b) Messages and received item records can be processed in main memory.
   c) All data stores are append-only.
   d) The time cost of accessing the data store to retrieve a record is independent of store size and independent of record size.
   e) The time cost of storing a record can be ignored, because it can be done asynchronously, using otherwise idle time.
2) Application-level assumptions
   a) Application parameters are the same for every node.
   b) All messages have the same size.
   c) All item records have the same size.
   d) The mapping from Object ID to Product Class ID and from it to Manufacturer ID is well-known.
   e) The cost of locating the node where to issue a query is negligible.
3) Chain-level assumptions
   a) Companies store data records about the items moving through the chain.
   b) A supply chain can be represented by a Direct-Acyclic Graph (DAG). A vertex (node) in the graph represents a company. A directed edge in the graph represents a flow of items.
   c) Nodes in the DAG are topologically sorted [33] and each item moves in the chain from a begin-node towards an end-node.
4) Product-level assumptions
   a) A BoM for a product is represented as a tree. The root node is the product. The children nodes are components of the parent node.
   b) A product is never disassembled.

D. Cost formulae

The time cost is measured in seconds and is denoted by C.

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 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.

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

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

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

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

For a message exchange, k is the sum of item records in the request and response payloads.

IV. Metadata integration approach

In this approach, the system is centralized to integrate metadata about the location of data sources. The special nodes are called DS.

A. MDI-specific assumptions

1) There is a single, well-known DS Search engine.
2) There are multiple DS instances.
3) There is one EPCIS instance for each company.
4) The cost of DS Search is considered negligible because the result can be cached for the duration of the system's lifetime.
5) A DS publication is done only once for each item for each node.

B. Capture cost

When a company receives a product it creates r EPCIS records locally. A single publication is sent to the DS.

The time cost of an item's data capture ($C_{mdi\,capture}$) is a message exchange with an item record for each company where the product passes.

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

C. Track query cost

To answer a track query the DS Root (or a cache) is contacted to locate the suitable DS instance. A query is issued to the DS and the EPCIS location is returned in the response. The asker contacts the EPCIS with the most recent sighting of the object, as depicted in Figure 5.

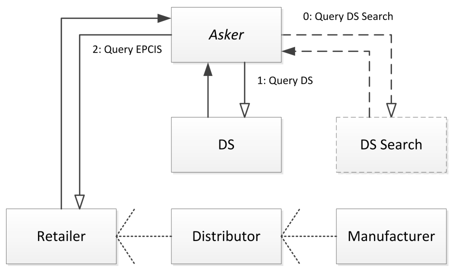


Fig. 5. Metadata integration approach track query cost.

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 EPCIS.

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

D. Trace query cost

A trace query is issued to the DS and a list of EPCIS locations are returned. The asker contacts each EPCIS 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 EPCIS.

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

E. BoM query cost

A BoM query starts with a DS query to track the first observation of the product. Then, the asker contacts the EPCIS for all aggregation records. The BoM query continues recursively for each component. 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 EPCIS aggregation queries.

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

V. Data integration approach

In this approach the system is centralized to integrate all data.

A. Capture cost

The capture records are sent to the Manufacturer for storage. No data is kept at the other nodes. The time cost of an item's data capture ($C_{di\,capture}$) is a message exchange with an item record for each company where the product passes outside of the Manufacturer.

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

B. Track query cost

The track query is sent directly to the Manufacturer, as shown in Figure 6.

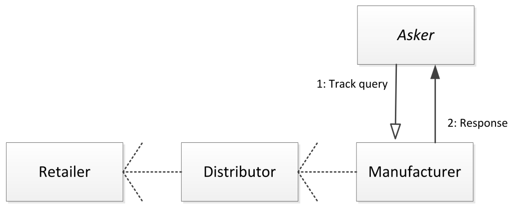


Fig. 6. Data integration track query.

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

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

C. Trace query cost

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

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

D. 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.

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

VI. Unstructured P2P approach

In this approach the system is decentralized and data is distributed across the capture nodes.

A. UP2P-specific assumptions

1) The cost of determining the address of the next node is a single lookup.

B. Capture cost

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

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

C. Track query cost

The track query processing is represented in Figure 7.

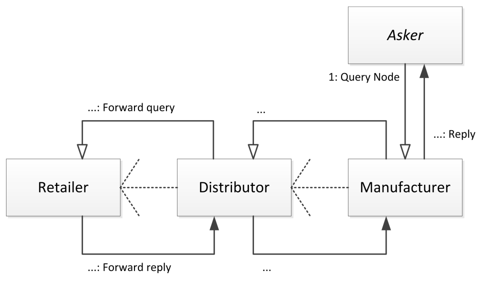


Fig. 7. Unstructured P2P track query.

The time cost of a track query is ($C_{up2p\,track}$) 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.

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

D. Trace query cost

The time cost of a trace query is ($C_{up2p\,trace}$) the sum of the cost of propagating the query to the node where the item is with the cost of propagating back the response with the accumulated trace records.

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

E. 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.

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

$$\sum_{i=0}^{b} \Big(c^i \cdot C_M\big(\sum_{j=i}^{b} c^{(b-j)}\big)\Big)$$

VII. Structured P2P approach

In this approach the system is decentralized and data is distributed according to a hashing algorithm to form a Distributed Hash Table (DHT) [34].

A. SP2P-specific assumptions

1) There is one DHT node for each chain node.
2) The hashing algorithm distributes the data evenly across the DHT nodes, using a unique item identifier.
3) The number of message hops to put or get a value is the logarithm of the number of DHT nodes [34].
4) The cost of determining the address of the next hop is a single lookup.
5) The nodes join the DHT once during the system's lifetime and never leave.
6) The cost of joining the DHT is negligible.
7) The item records are all kept on the same DHT node, indexed by the unique item identifier.

B. Capture cost

When a company receives a product, it contacts a known DHT node to forward the data record. The hash algorithm determines the destination node that is reached through a series of forwarding messages (hops). The time cost of data capture ($C_{sp2p\,capture}$) 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.

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

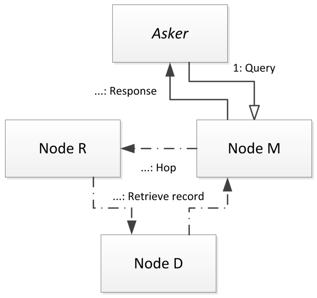


Fig. 8. Structured P2P track query.

C. 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 hops. Figure 8 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.

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

D. 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.

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

E. 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.

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

VIII. Results

The cost model was used to compute estimates and generate plots for two industry supply chains [3]: Automobile (Figure 9) and Pharmaceutical (Figure 10). The vertical axes show the time cost, and the horizontal axes show the number of items.

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.

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.

The most expensive query is the BoM. It is always at least three orders of magnitude more costly than data capture and the track and trace queries. The peer-to-peer approaches are more expensive for all kinds of query. 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 other solution concerns, such as security. The UP2P has the best possible capture cost: zero. It has a similar performance to DI and MDI for the BoM query, but is significantly costlier for track and trace. However, it allows complete data ownership control, which is a strong requirement for some companies [7]. The SP2P approach is always costlier than DI and MDI, and is only slightly better than UP2P on track and trace. However, because of the use of DHT technology, it has the most potential to scale.

IX. Conclusion

This paper presented a cost model developed to quantitatively compare traceability system architectures. We did a survey of publications and used a classification to summarize them into four main approaches. Each approach was then modeled using cost formulae. The absolute cost values produced by the model cannot be easily translated to actual system cost but they allow us to compare approaches. The specific supply chain domain being addressed is relevant to the choice of system architecture because the chain layout and the product features change the parameters that, in turn, significantly affect the estimated costs.

The *cost* measure is important to realize if the system is worth being built. However, there are other solution concerns that are left unaddressed. *Data ownership protection* is very important, as an organization can benefit from sharing data with trusted business partners but can also be harmed if data is exposed to competitors. Data providers and consumers have to be authenticated and the data access must be authorized. The required *performance* level is demanding, because it will have to provide timely responses to most queries. The *availability* level will have to be high, because critical business processes will rely on it. Finally, the system should be *scalable* to handle increasing numbers of data and users without suffering a noticeable loss of performance or increase in administrative complexity. A complete assessment of a traceability solution has to account for all of these concerns.

A. Future work

We will extend the cost model to include storage cost, and to compute partial costs for different kinds of nodes. We will use real-world supply chains to further validate the model. We already have some data from Retail and Air Transportation scenarios. It is important to have a settled business context to properly interpret the model results.

The model does not consider trust and other concerns, so we will have to complement the cost model with additional assessment tools. We will then measure the impact of adding *domain-specific rules* to the traceability system to try to improve the system by leveraging recurring supply chain data access patterns due to physical (time-space) and business realities (documents and processes).

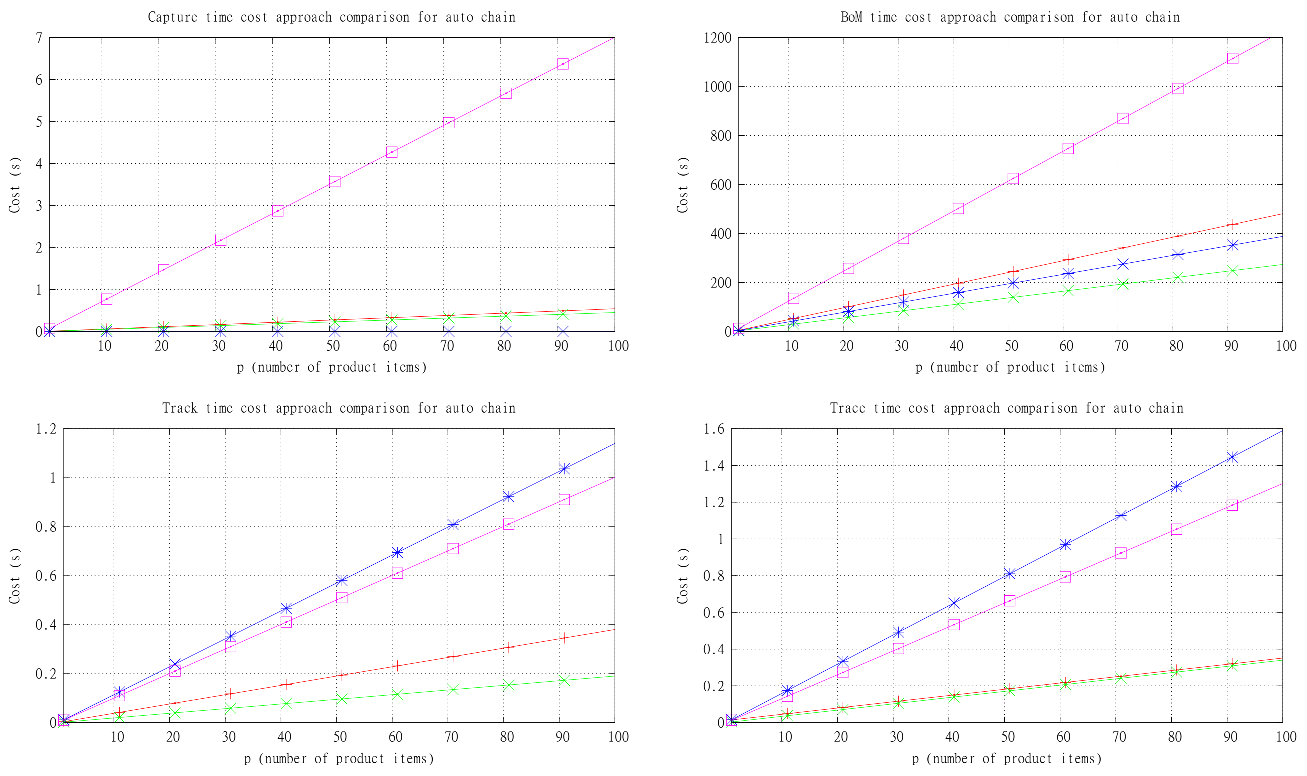
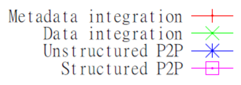


Fig. 9. Cost for Capture, BoM, Track, and Trace for a generic Auto supply chain.



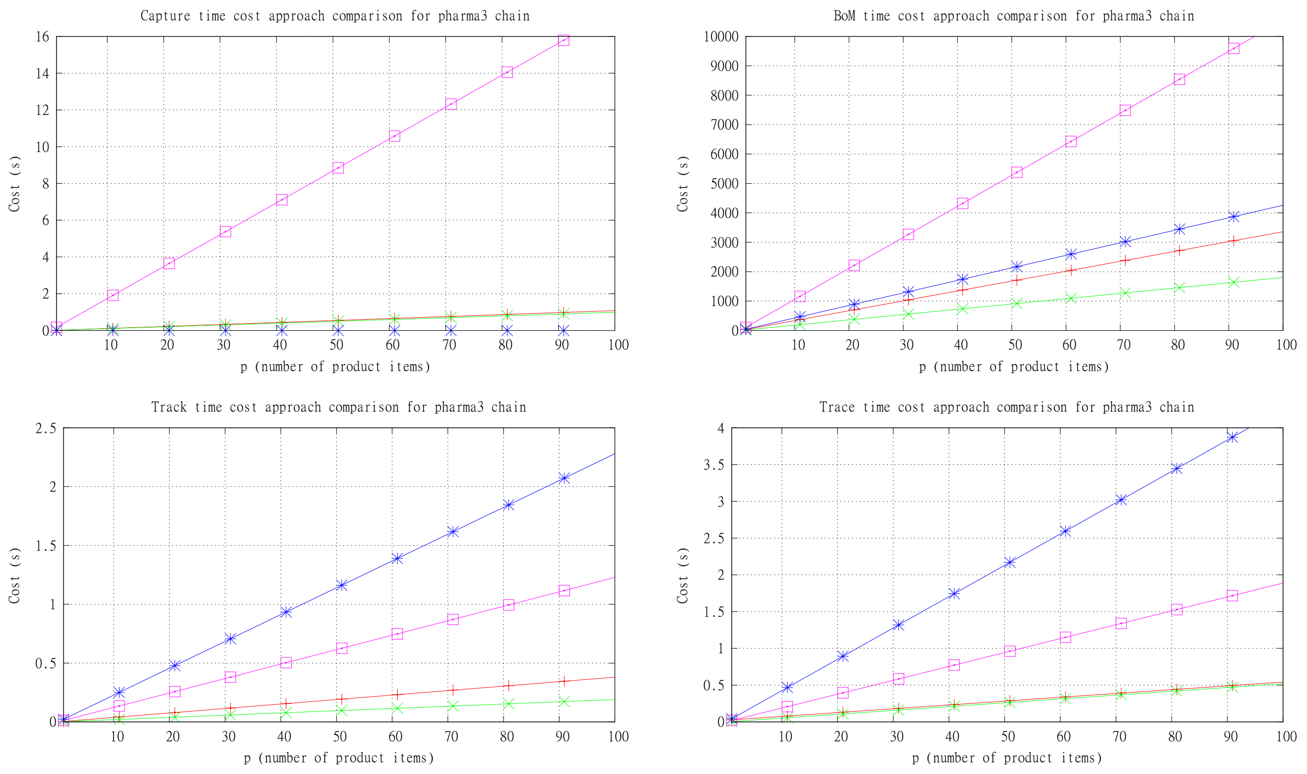


Fig. 10. Cost for Capture, BoM, Track, and Trace for a generic Pharma supply chain.

Acknowledgment

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

The authors also acknowledge Prof. Sanjay Sarma and Dr. Christian Floerkemeier for their early reviews of this work.

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

References

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

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

[3] K. Murthy and C. Robson, "A model-based comparative study of traceability systems," in *Proceedings of the International Conference on Information Systems, Logistics and Supply Chain (ILS)*, 05 2008, madison, Wisconsin.

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

[5] EPCglobal, *Application Level Events (ALE) Specification 1.1.1*, GS1 Std., 03 2009. [Online]. Available: http://www.epcglobalinc.org/standards/ale

[6] ——, *EPC Information Services (EPCIS) 1.0.1 Specification*, GS1 Std., 09 2007. [Online]. Available: http://www.epcglobalinc.org/standards/epcis

[7] BRIDGE, "Requirements document of serial level lookup service for various industries," University of Cambridge and AT4 wireless and BT Research and SAP Research and ETH Zurich and GS1 UK, Tech. Rep., 08 2007.

[8] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann, "Comparison of discovery service architectures for the internet of things," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2010, pp. 237–244.

[9] H.-H. Do, J. Anke, and G. Hackenbroich, "Architecture evaluation for distributed Auto-ID systems," in *Proc. 17th International Workshop on Database and Expert Systems Applications (DEXA)*, 2006, pp. 30–34.

[10] P. Tearnen, "EPCglobal US proof of concept," EPCglobal US conference, 09 2005.

[11] Verisign, "The EPCglobal network: Enhancing the supply chain," Verisign, Tech. Rep., 2008. [Online]. Available: http://www.verisign.com/static/DEV044095.pdf

[12] S. Beier, T. Grandison, K. Kailing, and R. Rantzau, "Discovery services - enabling RFID traceability in EPCglobal networks," in *International Conference on Management of Data (COMAD)*, 2006.

[13] Y. Cao, D. Wang, and H. Sheng, "PTSP: a lightweight EPCDS platform to deploy traceable services between supply-chain applications," in *1st Annual RFID Eurasia*, 2007, pp. 1–5.

[14] G. Lee, J. Shin, D. Park, and H. Kwon, "Discovery architecture for the tracing of products in the epcglobal network," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 2, 2008, pp. 553–558.

[15] J. Worapot, Y. Li, and A.-I. Somjit, "Design and implement of the EPC discovery services with confidentiality for multiple data owners," in *IEEE International RFID Technology and Applications Conference (RFID-TA)*, 2010, pp. 19–25.

[16] J. Muller, J. Oberst, S. Wehrmeyer, J. Witt, A. Zeier, and H. Plattner, "An aggregating discovery service for the EPCglobal network," in *43rd Hawaii International System Sciences Conference (HICSS)*, 2010, pp. 1–9.

[17] M. Young, *Extensible Supply-chain Discovery Service Concepts*, IEFT, Internet Engineering Task Force (IETF) Std., 08 2008. [Online]. Available: http://tools.ietf.org/html/draft-young-esds-concepts-04

[18] J. Cantero, M. Guijarro., G. Arrebola, E. Garcia, J. Banos, M. Harrison, and T. Kelepouris, "Traceability applications based on discovery services," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 09 2008, pp. 1332–1337.

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

[20] E. Polytarchos, S. Eliakis, D. Bochtis, and K. Pramatari, "Evaluating discovery services architectures in the context of the internet of things," in *Unique Radio Innovation for the 21st Century*, D. C. C. Ranasinghe, Q. Z. Z. Sheng, and S. Zeadally, Eds. Springer Berlin Heidelberg, 2010, pp. 203–227. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03462-6_10

[21] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse, "Discovery service design in the EPCglobal network, towards full supply chain visibility," *Internet of Things*, pp. 19–34, 2008.

[22] A. Rizzi, P. Simonazzi, and R. Vitulli, "Design, implementation and In-Field testing of an original discovery service for EPC network infrastructure," 2009.

[23] S. Song, T.-K. Shim, and J.-H. Park, "Proxy based EPC track&trace service," in *IEEE International Conference on e-Business Engineering (ICEBE)*, 10 2006, pp. 528–531.

[24] C. Robson, Y. Watanabe, and M. Numao, "Parts traceability for manufacturers," in *IEEE 23rd International Conference on Data Engineering (ICDE)*, 2007, pp. 1212–1221.

[25] A. Laurence, J. L. Moulec, J. Madelaine, and I. Bedini, "Experiments of discovery services interconnection," in *International Workshop on RFID Technology (IWRT)*, 2010. [Online]. Available: http://wings-project.fr/uploads/PmWiki/DS_interconnection.pdf

[26] K. Framling, T. Ala-Risku, M. Karkkainen, and J. Holmstrom, "Design patterns for managing product life cycle information," *Communications of the ACM*, vol. 50, no. 6, pp. 75–79, 2007. [Online]. Available: http://dialog.hut.fi/publications/CommACM2007_KF_forWeb.pdf

[27] D. Huang, M. Verma, A. Ramachandran, and Z. Zhou, "A distributed ePedigree architecture," in *IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, 2007, pp. 220–230.

[28] A. Cheung, K. Kailing, and S. Schonauer, "Theseos: A query engine for traceability across sovereign, distributed RFID databases," in *23rd IEEE International Conference on Data Engineering (ICDE)*, April 2007, pp. 1495–1496.

[29] S. Wakayama, Y. Doi, S. Ozaki, and A. Inoue, "Cost-effective product traceability system based on widely distributed databases," *Journal of Communications*, vol. 2, no. 2, pp. 45–52, 03 2007.

[30] N. Schoenemann, K. Fischbach, and A. Manteuffel, "Flexible semantic services to facilitate innovative and dynamic ubiquitous supply chain networks," in *2nd International Conference on Computer Science and its Applications (CSA)*, 2009, pp. 1 –5.

[31] B. Fabian, "Implementing secure P2P-ONS," in *IEEE International Conference on Communications (ICC)*, 2009, pp. 1–5.

[32] N. Schoenemann, K. Fischbach, and D. Schoder, "P2p architecture for ubiquitous supply chain systems," in *17th European Conference on Information Systems (ECIS)*, 2009.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2009.

[34] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Communications of the ACM*, vol. 46, pp. 43–48, 02 2003. [Online]. Available: http://doi.acm.org/10.1145/606272.606299