

A Model-based Comparative Study of Traceability Systems

Karin Murthy, Christine Robson
IBM Almaden Research Center, San Jose, CA 95120
{klmurthy, crobson}@us.ibm.com

Abstract: Traceability, the ability to track parts and products, is a necessity for many enterprise applications. These include pilferage reduction, counterfeit prevention, and targeted recalls. In this paper, we share lessons learned from a comparative analysis of two state of the art traceability data management systems: a federated system for parts traceability; and a distributed system for distribution chain traceability. We have abstracted out the characteristics of the two systems and built simple query execution models to compare the performance of the federated and distributed approach to traceability. Using these abstract models we evaluate how different parameters such as supply chain layout and available infrastructure influence the costs incurred. Based on the comparison, we provide guidelines for businesses looking to buy or build a traceability system. This work is an important step towards a formal framework to compare disparate traceability systems, taking into account common industrial configurations.

Keywords: Traceability and tracking, supply and distribution chains, federated and distributed query processing, performance modeling

1 Introduction

Fueled by advances in technologies such as Radio Frequency Identification and new standards such as the Electronic Product Code, traceability is emerging as a key differentiator in many industries. For example, TraceSphere (Robson et al., 2007) was built for a large automobile manufacturer to facilitate parts traceability and recalls. Whereas Theseos (Cheung et al., 2007) was built with the pharmaceutical industry in mind to enable the generation of pedigree documents for products. Both systems support enterprise traceability, but their approaches to traceability are fundamentally different: TraceSphere is federated and places all authority with the party hosting the system; Theseos is distributed and assumes that control and data is distributed among all participants.

In this paper, we investigate the differences between the federated and distributed approach to traceability and share the lessons learned from a comparative analysis. An objective comparison of working traceability systems is difficult as differences in environment and industry, as well as differences in implementation, make it hard to accurately benchmark the capabilities and costs. Thus, we derive abstract models for query execution in the federated as well as the distributed approach to traceability. The derived models enable a comparison of the cost effectiveness of the two approaches independent of the system landscape the actual systems were deployed in.

We introduce formulae to measure the cost of executing typical traceability queries in the federated and the distributed approach, taking into account infrastructure parameters such as bandwidth and processing speed, as well as application parameters such as the width and length of the supply and distribution chains. We use the cost measures to evaluate various parameter combinations and show how these parameters influence the overall cost incurred with the two approaches.

Based on the comparison, we provide guidelines on choosing the appropriate type of traceability solution for specific needs. In addition to helping companies understand suitable traceability solutions for today, we also look at how future developments such as increased bandwidth might affect the cost in the federated and distributed approach to traceability. Our guidelines also cover factors beyond cost such as reliability and confidentiality of data. We use the lessons learned to suggest improvements for future traceability systems.

The remainder of this paper is organized as follows: Section 2 provides background on traceability and the industries we consider in this paper. Section 3 introduces abstract models for processing queries in the federated and the distributed approach to traceability. The results of the comparison and guidelines for choosing a traceability system are described in Section 4 and Section 5 respectively. Section 6 concludes the paper with a look at the future of traceability systems.

2 Background

In this section, we provide an overview of two key industries that employ traceability systems. We also briefly introduce the two solutions that form the basis for our comparison of federated and distributed traceability.

Automotive industry. The current trend in the automotive industry is towards outsourcing, which results in increasingly complex and dynamic supply chains. This trend has led to a situation where records of sub-component parts are distributed across a wide network of suppliers' databases, and the manufacturer does not "own" the records of some major components in the vehicle. In this situation, a traceability system enables the manufacturer to view the records of all parts in a vehicle, including complex outsourced components.

Traceability is mandatory for automotive manufacturers to enable effective recalls in case of defective component parts. Traceability is also necessitated worldwide by recycling regulations. For example, the Japanese end-of-life vehicle recycling law¹ requires all automotive manufacturers to conform to strict recycling guidelines. Such regulations have fueled the need for parts traceability of seemingly minor parts of a vehicle, which until recently were completely ignored in parts traceability systems.

A key characteristic of the automotive and other mass production industries is the power and authority of the manufacturer. Large manufacturers typically exert considerable influence on their supply and distribution (S&D) chains and often own significant stakes of primary suppliers and distributors.

TraceSphere (Robson et al., 2007) is a federated system designed for the automotive industry. The system is deployed by an automobile manufacturer and only the manufacturer can initiate queries. As vehicles are assembled, a list of all the parts is indexed by Vehicle Identification Number (VIN). Each supplier maintains its own data, the vehicle manufacturer maintains only an index where all data for a specific VIN is located. TraceSphere employs the IBM Information Integrator² to enable an automobile manufacturer to query its suppliers independently of their data format and schema.

Custom applications run on top of the TraceSphere index to enable special functions such as very fast selective recalls. The system is designed to index at least 1000 parts in a vehicle, drilling-down many tiers deep in the supply chain to index sub-component parts.

Pharmaceutical industry. Counterfeit drugs are a big challenge for the pharmaceutical industry: Cockburn et al. (2005) report that an estimated 15% of all sold drugs are fake and Malykhina (2004) reports that the number of investigations by the US Food and Drug Administration (FDA) of counterfeit drugs has increased four-fold from 1997 to 2003 and is continuing to increase at a dramatic rate.

To combat counterfeiting, the FDA strongly recommends³ and some US states⁴ mandate the use of traceability technology to enable pedigree documents signed by each party in the distribution chain of a drug.

¹Japan METI Ministry of Economy, Trade and Industry: End-of-Life Vehicle Recycling Law, January 1st, 2005

²<http://www-306.ibm.com/software/data/integration/>

³http://www.fda.gov/oc/initiatives/counterfeit/report02_04.html

⁴Florida Statutes. Section 499.0121 and California Business and Professions Code. Section 4163.

However, generating the complete pedigree for a drug requires independent organizations to work together to enable life-cycle traceability. Many of these organizations are competing enterprises that want to protect their confidential information. In addition, pharmaceutical products are often sold and re-sold through many distributors, creating an extremely long distribution chain, which many companies would prefer to mask from their customers. Thus, a traceability solution for this environment should provide organizations with control over which information is shared with whom.

Theseos (Agrawal et al., 2006; Cheung et al., 2007) is a distributed system where any party in the S&D chain can initiate queries. Each party maintains their data locally and indexes only their data. To enable the execution of cross-organizational traceability queries, each party also maintains the information where it received products from and sent products to. Theseos uses these records to determine which companies in the S&D chain need to be consulted to answer a query. Theseos analyzes an incoming query, retrieves local results, rewrites the query if necessary, forwards it to neighboring nodes in the S&D chain, and aggregates results. The same process takes place at neighboring nodes. Local query execution is governed by confidentiality policies, allowing each party to decide on a query by query basis what data is shared with whom.

3 Modeling the Cost of Query Processing

We begin our analysis of the two traceability systems by establishing a common frame of reference for comparison. We abstract out the core elements of each system into simple query execution models. The *federated* approach is an abstraction of TraceSphere and the *distributed* approach is an abstraction of Theseos.

3.1 Abstract Models for Query Processing

We model an S&D chain as a tree. A node represents a company and the edges show the flow of items through the S&D chain (items move towards the root node). The root node represents the end-manufacturer in a supply chain or the end-retailer in a distribution chain. The depth of the tree denotes the length of the S&D chain. The branching factor of the tree determines the width of the S&D chain.

Companies store records about the items moving through the chain. We represent query execution to retrieve those records using a few simple graphical elements. Figure 1 provides a legend. The S&D chains are shown starting with the root node on the left side. Details of the S&D chains are omitted wherever they are not relevant to illustrate the query execution.

In the next three subsections we detail the kind of queries we evaluated: single-record lookup, pedigree, and bill-of-materials. This taxonomy of queries is not meant to be exhaustive, but it is representative of the common traceability queries in the industries described in Section 2.

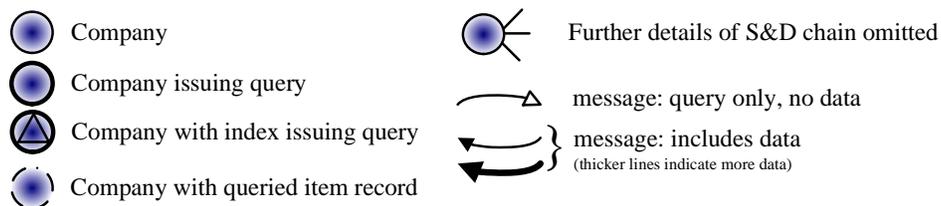


Figure 1. Legend for S&D chain model

3.1.1 Single-Record Lookup Query

A single-record lookup (SRL) query retrieves one specific record of an item. For example, an SRL query may retrieve an item's description from the manufacturer of the item. SRL queries form the basis for more complex queries.

In a federated system, the querying node sends the query directly to the company that owns the record, as shown in Figure 2(a). The query can be sent directly to the source of the information, because the party operating the federated system maintains a local index that indicates the location of the queried record.

In the distributed system, the query is propagated down the S&D chain to the node which owns the record, and then the record is returned along the same path. See Figure 2(b). Each node is responsible for maintaining information about the product flow, in order to pass queries on. Companies also need to propagate results back upstream.

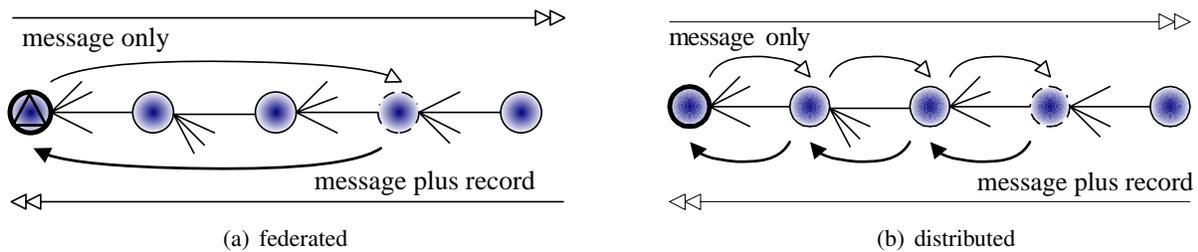


Figure 2. SRL query

3.1.2 Pedigree Query

A pedigree query is executed to obtain all records of an item in the S&D chain. For example, a pedigree query for a bottle of medicine returns the complete distribution history of the bottle. Pedigree queries are a common traceability query type in the pharmaceutical industry. In the automotive industry a pedigree query may be issued to retrieve the assembly history of a specific component part.

In the federated system, a separate query is issued in parallel to each company which owns relevant information about the item under investigation. See Figure 3(a). All item records are returned directly to the query originator.

The distributed system executes a pedigree query similar to an SRL query, by propagating the query down the S&D chain to find all sources with records that pertain to the item in question. See Figure 3(b). The pedigree query differs from the SRL query in that each node appends their relevant item records to the response as the query response propagates back upstream to the querying node.

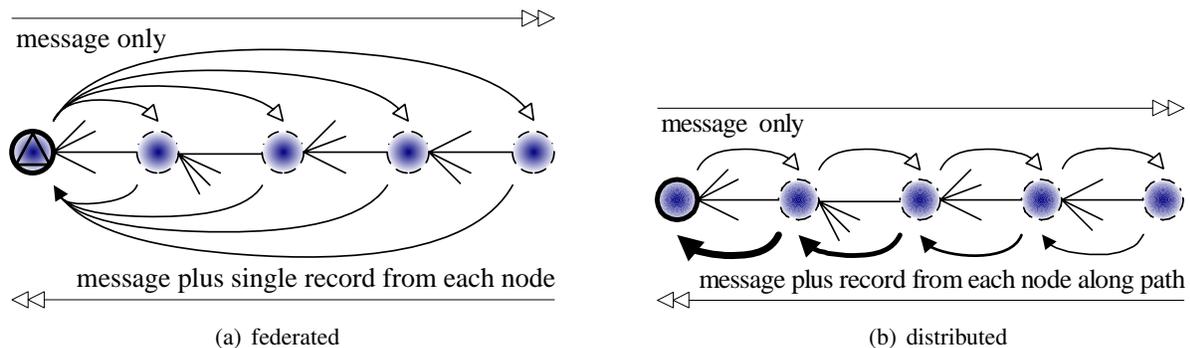


Figure 3. Pedigree query

3.1.3 Bill-of-Materials Query

A bill-of-materials (BOM) query returns all containment-related information about an item. For example, a BOM query is used to find every part used in a car. BOM queries are a common traceability query type in the automotive industry.

A federated system executes a BOM query by directly querying everyone who has ever handled the product or any part in the product. See Figure 4(a). In the distributed system, the query executes “organically” through the S&D chain. The company issuing the query forwards the query messages to all of its suppliers, who then forward the query on to each of their suppliers. When a company has no one else to ask, it returns its records back along the query execution path. As the reply propagates towards the source of the query, each company appends its records to the response, and forwards it. Figure 4(b) illustrates this execution.

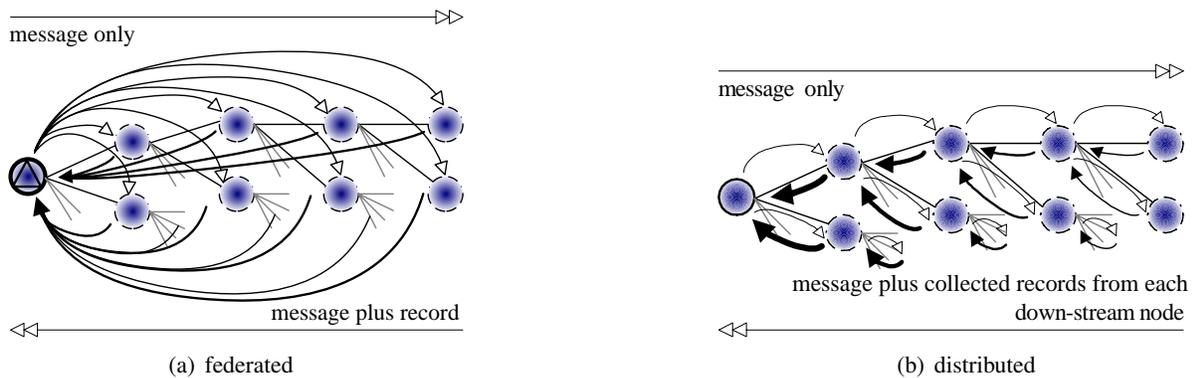


Figure 4. BOM query

3.2 Measuring Cost of Query Processing

We use the abstract models introduced in Section 3.1 to evaluate the cost of query execution in federated and distributed systems. We measure cost as the *aggregated processing time* of a query, which is the sum of the following costs incurred by each company in the S&D chain: message passing and processing, and database lookups. For the federated system, we amortize the cost of building index tables and distribute a percentage of that cost to each query over the lifetime of the system. The effect of measuring cost in this way is two-fold: First, all costs assume long-term investments, since the overhead cost of a federated system only pays off over time. Second, cost is measured for the entire S&D chain, not for one particular company, since the execution cost for each company involved in a distributed query response is added up.

Table 1 summarizes the key parameters that influence the cost of executing a query. Transmission cost is the number of seconds spent in transferring data. Processing cost is the number of seconds spent processing messages and item records. We measure both bandwidth and processing speed in kilobits per second (Kbps). The cost of accessing the database to retrieve a result record is approximated by the disk seek time and measured in seconds. The size of messages and item records is measured in kilobits (Kbits).

A simple way to describe the layout of an S&D chain is to measure its length and the average number of children per node. A more detailed description takes into account the length of the S&D chain and the exact number of nodes at each level of the S&D chain. For SRL and pedigree queries the average depth influences the cost of query processing.

Note, that we are not attempting to simulate actual system behavior under various circumstances. This would require many more parameters than listed in Table 1. However, building cost formulae using these key parameters provides enough flexibility to compare the cost of query processing in the two kinds of systems at a high-level. Thus, when calculating the cost of processing a query we make the following

Table 1. System, application, and S&D chain parameters

	Name	Symbol	Unit
<i>System Parameters</i>	Bandwidth	β	Kbps
	Processing speed	γ	Kbps
	Disk seek time	θ	sec
<i>Application Parameters</i>	Message size	μ	Kbits
	Item record size	δ	Kbits
	Index record size	λ	Kbits
<i>S&D Chain Parameters</i>	Length of S&D chain	z	
	Avg. # children per node	a	
	# of nodes per level	$B = [b_1, \dots, b_z]$	
	Avg. depth of look-up	n	

simplifying assumptions: system and application parameters are the same for every node; on average, each node contributes one item record to a query answer; all messages and all item records have the same size; messages and received item records can be processed in main memory; the cost of accessing the database to retrieve a record is independent of the record size; and the cost of accessing the local index in the federated model and the cost of determining the address of the next node in the distributed model is negligible.

In order to enable easy summation of all costs we aggregate the cost over the number of communications that take place. There are two types of communications: sending and receiving of the query message (see left side of Figure 5) and sending and receiving of the reply message (see right side of Figure 5). The cost for sending and receiving a query message is the sum of the following costs: message processing by sending node ($\frac{\mu}{\gamma}$), message transfer ($\frac{\mu}{\beta}$), message processing by receiving node ($\frac{\mu}{\gamma}$), and database look-up by receiving node (θ). The cost for sending and receiving a reply message (which may include k additional result records) is the sum of the following costs: message processing by sending node ($\frac{\mu}{\gamma} + \frac{k*\delta}{\gamma}$), message transfer ($\frac{\mu}{\beta} + \frac{k*\delta}{\beta}$), and message processing by receiving node ($\frac{\mu}{\gamma} + \frac{k*\delta}{\gamma}$).

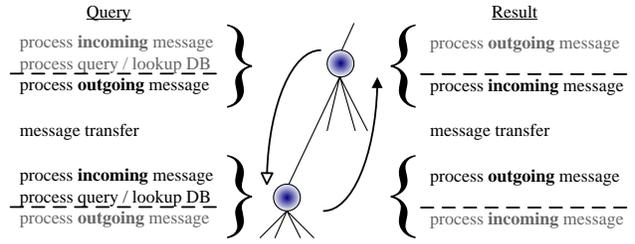


Figure 5. Costs per communication

Cost of SRL query in federated system. The cost of an SRL query in a federated system is composed of exactly two communications: one query message is sent and received (that is, $2\frac{\mu}{\gamma} + \frac{\mu}{\beta} + \theta$) and one reply message including one item record is sent and retrieved (that is, $2\frac{\mu+\delta}{\gamma} + \frac{\mu+\delta}{\beta}$). Summing up the costs for the two communications we derive the following formula to compute the cost of an SRL query:

$$C_{federated}^{SRL} = 4\frac{\mu}{\gamma} + 2\frac{\mu}{\beta} + 2\frac{\delta}{\gamma} + \frac{\delta}{\beta} + \theta$$

Cost of SRL query in distributed system. In the distributed system the cost depends on the depth of the record look-up n (that is, the number of parties involved in routing the query). The cost for each party is the same as in the federated system.

$$C_{distributed}^{SRL} = n \cdot C_{federated}^{SRL}$$

Cost of pedigree query in federated system. In the federated system the cost of a pedigree query is determined by the length of the pedigree n (that is, how many parties contribute an item record to the query

answer).

$$C_{federated}^{Pedigree} = n \cdot C_{federated}^{SRL}$$

Cost of pedigree query in distributed system. In the distributed system additional costs accumulate for passing results back along the chain.

$$C_{distributed}^{Pedigree} = n(4\frac{\mu}{\gamma} + 2\frac{\mu}{\beta} + \sum_{i=1}^z (2\frac{\delta}{\gamma} + \frac{\delta}{\beta})) + \theta$$

Cost of BOM query in federated system. In the federated approach, a BOM query can be viewed as a sum of SRL queries. We distinguish between two cases: the average number of children per node is specified by a or the number of nodes for each level of the S&D chain is specified by a vector B .

$$C_{federated}^{BOM} = \begin{cases} \sum_{i=1}^z a^i \cdot C_{federated}^{SRL} \\ \sum_{i=1}^z B[i] \cdot C_{federated}^{SRL} \end{cases}$$

Cost of BOM query in distributed system. In a distributed system additional costs accumulate for passing results back along the chain.

$$C_{distributed}^{BOM} = \begin{cases} \sum_{i=1}^z a^i (4\frac{\mu}{\gamma} + 2\frac{\mu}{\beta} + \theta) + \sum_{k=i}^z a^{(z-i)} (2\frac{\delta}{\gamma} + \frac{\delta}{\beta}) \\ \sum_{i=1}^z B[i] (4\frac{\mu}{\gamma} + 2\frac{\mu}{\beta} + \theta + \sum_{k=i}^z B[z-i] (2\frac{\delta}{\gamma} + \frac{\delta}{\beta})) \end{cases}$$

Cost of building an index. For the comparison, we add a percentage of the cost to build an index to each of the three formulae for the federated system.

$$C_{per-record}^{Index} = 4\frac{\mu}{\gamma} + 2\frac{\mu}{\beta} + \theta$$

4 Comparison

Using the cost formulae described in Section 3.2, we now compare the performance of federated and distributed traceability systems. We begin with a description of the parameters used for the comparison.

We use today's numbers for high-speed internet and off-the-shelf data systems to bind the system parameters. We use Mbps bandwidth, GHz processors (which we loosely equate to having Gbps processing speed), and 10ms seek time on disk drives. The application parameters are bound as follows: item records as well as messages (including addressing, query, and other communication overhead) are 100 Kbits in size; index records are 96 bits in size. A 96-bit EPC code is sufficient to uniquely identify every item in an S&D chain (EPCglobal, 2006).

Table 2 summarizes the five representative S&D chains we picked for comparison. The first three S&D chains are typical for the pharmaceutical industry where drugs change hands up to 20 times from the manufacturer until they reach a pharmacy or hospital. The fourth S&D chain is a typical automotive supply chain with six tiers of suppliers and an average of three different supplier for each part. To cover a broad range of S&D chain configurations, we also consider a short and very broad S&D chain.

Table 2. S&D chain layouts

Name	z	a	# nodes
Very long, single-source	20	1	20
Very long, narrow	20	≈ 1.5	$\approx 4,000$
Long, narrow	12	2	$\approx 4,000$
Short, broad	6	3	≈ 700
Short, very broad	5	10	100,000

We compare the cost of federated and distributed traceability systems for the three query types introduced in Section 3.1. Due to space constraints, Figure 6 only shows the most representative charts: BOM queries for different S&D chains; pedigree queries for different depths; pedigree and BOM queries for different bandwidths. We vary the percentage of queries executed per item record over the lifetime of the system and measure the respective cost. We use a logarithmic scale for the percentage of queries per item record to cover a broad range in a single chart. Given the greater importance of comparing the relative costs of the systems over the absolute costs, we also scale the costs within each scenario to present results for different scenarios together in one chart.

Figure 6(a) shows the cost of BOM queries for different S&D chain layouts. Federated traceability systems begin with a high fixed cost (the indexing overhead), and then incur a small cost for each query executed over the lifetime of the system. The cost for a distributed system grows linear with the number of executed queries. The curve looks exponential only because it is plotted along a logarithmic scale.

For each of the four S&D chain layouts there is an *inflection point* below which a distributed system is less expensive, and above which a federated system is less expensive. In the case of the Long, narrow S&D chain with 4,000 companies, the inflection point is at 1.5%. That is, the federated approach is cheaper if more than 1.5% of the item records are queried over the lifetime of the system.

The analogous pedigree evaluation is done for different depths of a record in the S&D chain. As differently shaped S&D chains have no influence on pedigree queries (see Figure 3), we only vary the number of companies that are involved in building the pedigree. We evaluate depths of 2, 5, 10, and 20 companies. The results shown in Figure 6(b) are based on the Very long, narrow S&D chain with 4,000 companies. For queries which execute five levels deep in the S&D chain, the inflection point is at 110%. That is, if on average pedigrees are only generated once for each item then a distributed system is cheaper.

The ratio of bandwidth to processing and disk seek time is key to the cost incurred by each system. Figure 6(c) and Figure 6(d) show the effect of increased bandwidth on BOM and pedigree queries respectively. The results shown are based on the Short, broad S&D chain. We discuss these two charts in Section 5.

5 Guidelines

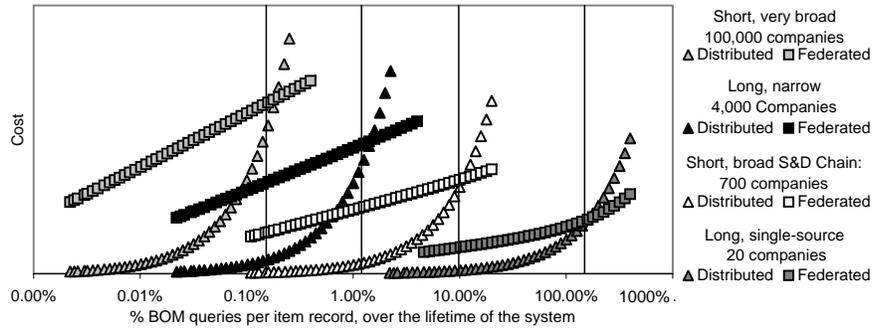
In this section we provide guidelines for choosing the most cost-effective and appropriate traceability system. We start out with an executive summary and then provide details of individual aspects.

5.1 Executive Summary

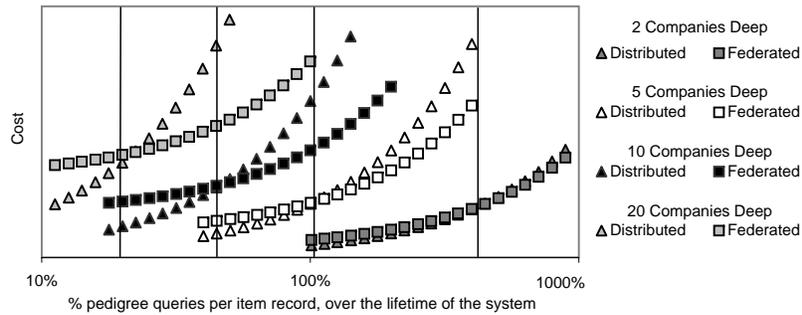
In general, a distributed system has less direct cost to a company than a federated system. There is less management overhead and less storage requirement, and most computation and communication overhead can be shared with partners. However, a distributed system requires cooperative companies. In a federated system, queries can be managed very efficiently and fast response times can be guaranteed.

Distributed systems are more cost-effective than federated systems when there are few queries compared to the number of item records. If data is frequently queried, a federated system is cheaper. A distributed system performs well on pedigree queries, such as those that would be asked in selective recall applications or quality control investigations. It performs less well on exhaustive BOM queries. Faster bandwidth dramatically reduces the cost of distributed systems, whereas faster processors and disks dramatically reduce the cost of federated systems. A distributed system is far more expensive for large item records.

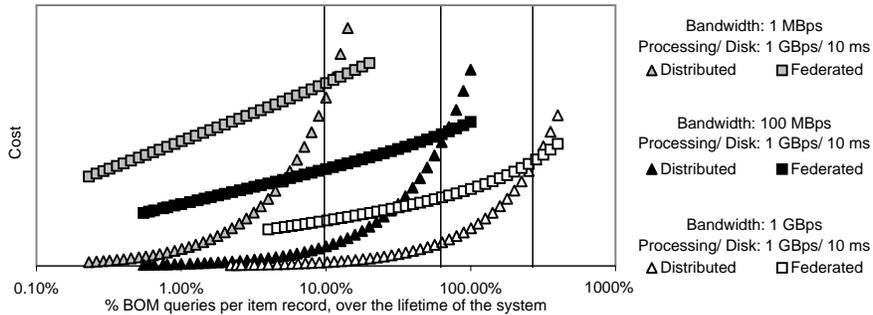
Contractual issues over information sharing, particularly those which prevent the sharing of data unless absolutely necessary, present a barrier to federated systems. On the other hand, the challenge of securing an agreement to universally adopt a standardized system is a barrier to distributed systems. Distributed systems have greater capacity for preserving confidentiality and autonomy, whereas federated systems can provide better reliability.



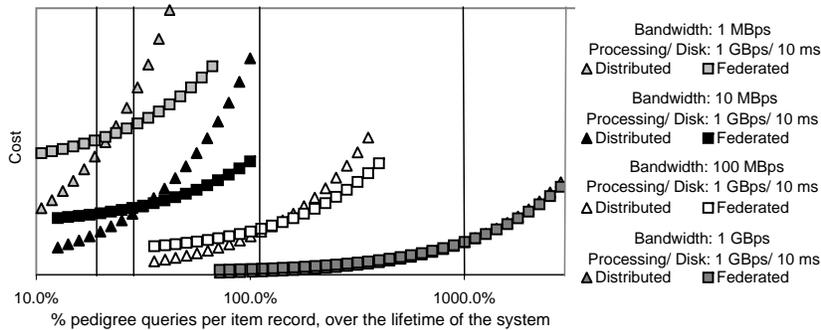
(a) BOM queries for different S&D chains



(b) Pedigree queries for different depth



(c) BOM queries for different bandwidth



(d) Pedigree queries for different bandwidth

Figure 6. Relative cost of queries in federated and distributed system (costs are relative within each scenario; the x-axis is in a logarithmic scale)

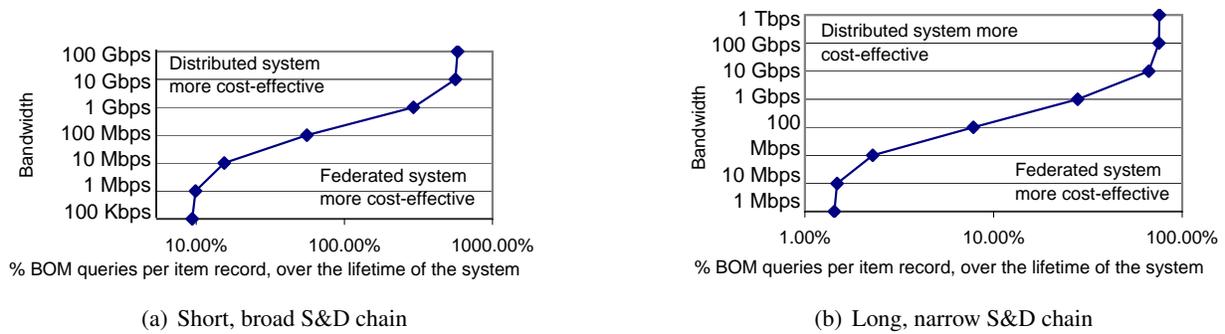


Figure 7. Cost inflection curves

5.2 Cost Considerations

When looking into a distributed versus a federated traceability system, a few key considerations determine which approach is more cost effective: the available infrastructure; the layout of the S&D chain; frequency and type of queries; and the item record size. In this section, we discuss the effect of those characteristics on the cost for query processing.

Effect of available infrastructure. Figure 6(c) shows the same automotive supply chain BOM query example, for different communication bandwidths. If the bandwidth available to the system increases to 1 Gbps, while the processing speed stays at 1 GHz, then a distributed system becomes much more cost-effective. In this scenario, each item record would need to be queried three times before the overhead of the federated system pays off. If processing speed and disk seek time scale with bandwidth, there is no effect on the relative costs of the distributed or the federated system.

Figure 7 shows the inflection points for the Short, broad S&D chain and the Long, narrow S&D chain for different bandwidth. If the bandwidth and expected query frequency fall above the line drawn between these inflection points, a distributed system is the more cost-effective option. Below the line, the federated system is the cheaper approach.⁵

Effect of layout of S&D chain. The size and layout of an S&D chain is crucial to determining how much infrastructure is required to run a traceability system. In general, the larger the S&D chain, the more cost effective a federated system is in comparison with a distributed system. See the different scenarios in Figure 6(a) and compare the two scenarios in Figure 7. Overall, the total size of an S&D chain is more important than its width and length.

Effect of frequency and type of queries. The overhead for a federated system begins to pay off once queries are executed on a significant portion of the item records. For example, for the Short, broad S&D Chain in Figure 6(a): With today's infrastructure, if BOM queries are executed on more than 10% of the item records, the overhead pays off and the federated system is cheaper.

Pedigree queries are relatively cheaper for distributed systems, as compared to BOM queries, since the cost of passing data does not compound to the same degree as in a BOM query. This can be seen by comparing Figure 6(a) and Figure 6(b).⁶

⁵At the high-bandwidth end of the graphs, the cost of processing data completely dominates the system. Message passing is effectively free, but processing the messages and item records is very expensive. A distributed system spends a large amount of time on processing item records as results are passed back along the S&D chain. At the low-bandwidth end of the graphs, the cost of sending data completely dominates the system. For a federated system, the high fixed cost of sending messages to index the entire space of item records is the most important factor.

⁶Note, that the layout of the Long, narrow S&D chain in Figure 6(b) varies slightly from the layout of the Very long, narrow S&D chain used in Figure 6(b). However, both S&D chains have the same total size of approximately 4,000 companies.

Effect of item record size. In a federated system, item records are transmitted directly from the record owner to the query originator. See Figure 4(a). In the distributed system, item records may be transmitted multiple times, and the amount of data being passed grows exponentially as the response returns to the query originator. See Figure 4(b). Thus, with today’s infrastructure, if item records are huge, on the order of 1 Gbit, a BOM query on just 1 in 10,000 items would justify the overhead expense of a federated system.

5.3 Considerations beyond Cost

The cost of query processing is not the only factor to consider when choosing a traceability system.

Contractual issues. The federated system was modeled with the understanding that an automotive manufacturer has near-absolute control over its supply chain. For other, less tightly controlled industries, such as low-end consumer electronics, building a federated system can be difficult. Suppliers in highly competitive environments are less likely to allow a manufacturer to index their product records, fearing that competitors may get access to their data.

The distributed system was designed for environments where manufacturers can exert little influence on their distributors. In this case, it may be nearly impossible (and costly) to secure the contracts necessary to preemptively access distribution data from every distributor. Thus, even in high-volume query situations, where a federated system would be the cheaper alternative, a distributed system may be the only alternative.

Confidentiality of data. Distributed systems have a far greater capacity for protecting confidentiality of data than federated systems, since there is no “big brother” company that keeps track of the supply and distribution relationships between the various parties. In a distributed system, each company independently keeps a record of the companies with which it does business. It is free to choose which information it reveals in response to a query.

Data integrity & counterfeit detection. Neither of the two described approaches can detect all counterfeit attacks. For example, a pallet of drugs may be replaced by counterfeit products, and the authentic drugs sold in the black market. In some cases, the federated system may be more able to detect counterfeiting. For example, a company receives a pallet of authentic medication, duplicates the records and labels, and distributes two pallets of medication, both with identical records, to two different downstream distributors. One of the two pallets contains counterfeit medication. A federated system may detect the counterfeit as the two downstream distributors report back with the same pallet code. However, if the traceability system is distributed the counterfeiter could possibly forge query results to prevent detection.

Reliability. A distributed system is inherently more prone to down-time problems. Consider the execution of a pedigree query. For a federated system, if a company fails to respond, only its item record is missing from the response. In a distributed system, if any company along the query path fails to respond, all records from subsequent downstream companies are unreachable. In the worst case, if a direct supplier or distributor is unresponsive, the query does not return any results at all.

Storage requirements. We compare the required storage for a representative automotive scenario. For our comparison, we assume a 30 million life-time unit production, which is 10 years of production, at three million vehicles a

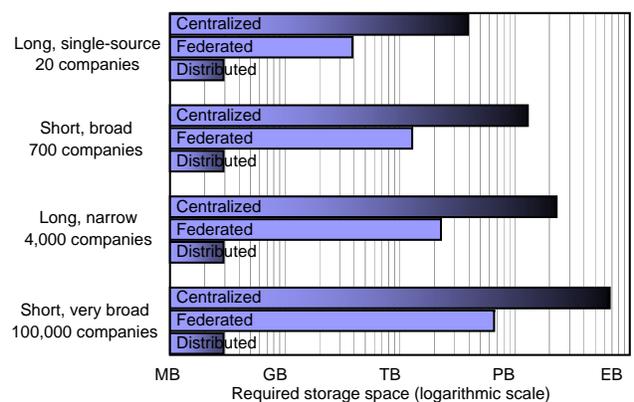


Figure 8. Storage space

year.⁷ On average, each company in the supply chain contributes one part to a car.

Figure 8 shows the required storage for four kinds of S&D chains. We include the costs for a centralized data warehouse, where all data is cached locally, to show that this is not a viable alternative. The figure shows that the disk space needed to maintain all data locally is effectively three orders of magnitude higher than the disk space required for indexing locally. The cost for a distributed system is the same regardless of the size of the S&D chain, as no additional data or indices are stored locally.

6 Summary and Conclusion

We examined two state of the art traceability data management systems, abstracted general characteristics from their implementations, and thus created a platform for comparison. Based on the developed models and cost measures, we compared the performance of the federated and distributed approach to traceability. We leveraged this information to provide guidelines for enterprises, pointing out the parameters in choosing a traceability solution that is appropriate for individual needs.

To conclude, we present three possible directions to improve the two described approaches to traceability. Federated and distributed traceability can be combined to overcome the respective drawbacks. For example, a large manufacturer may employ a federated traceability system for tracking parts from dedicated suppliers, or for tracking parts with safety and quality concerns. A distributed system could be used as an ad-hoc way to gather more comprehensive information about less important parts in the final product.

A significant source of expense in the distributed approach is the cost of propagating query results back along the path the query took. To avoid this inefficiency, results could be sent back directly to the query issuer. To ensure accuracy of the results, a signature authenticating the item record could be propagated back along the path the query took. This is a cost-effective alternative in environments where the identity of all companies can be revealed to the query issuer.

Instead of one company building an in-house index, the index can be shared to distribute the cost for building and maintaining the index. However, calling a central index adds additional message passing costs to each query and introduces a single point of failure. Also, sharing the index requires agreement among all participants. EPCglobal⁸ is currently leading standardization efforts to build a traceability network with a central index (Beier et al., 2006).

References

- Agrawal, R., Cheung, A., Kailing, K., and Schoenauer, S. (2006). Towards Traceability across Sovereign, Distributed RFID Databases. In *Proc. of the 10th Int. Database Engineering & Applications Symposium*.
- Beier, S., Grandison, T., Kailing, K., and Rantzau, R. (2006). Discovery Services Enabling RFID Traceability in EPCglobal Networks (Demo). In *Proc. of the 13th Int. Conf. on Management of Data*.
- Cheung, A., Kailing, K., and Schoenauer, S. (2007). Theseos: A Query Engine for Traceability across Sovereign, Distributed RFID Databases (Demo). In *Proc. of the 23rd Int. Conf. on Data Engineering*.
- Cockburn, R., Newton, P. N., Agyarko, E. K., Akunyili, D., and White, N. J. (2005). The Global Threat of Counterfeit Drugs: Why Industry and Governments Must Communicate the Dangers. *PLOS Med*, 2(4):302–308.
- EPCglobal (2006). EPCglobal Tag Data Standards Version 1.3. *Ratified Specification*.
- Malykhina, E. (2004). Drugmaker Ships RFID Tags With OxyContin. *Intelligent Enterprise*.
- Robson, C., Watanabe, Y., and Numao, M. (2007). Parts Traceability for Manufacturers. In *Proc. of the 23rd Int. Conf. on Data Engineering (ICDE '07)*.

⁷<http://www.audiworld.com/news/05/090505/>
<http://world.honda.com/automobile/report/2006/>

⁸<http://www.epcglobalinc.org/>