

Theseos: A Query Engine for Traceability across Sovereign, Distributed RFID Databases

Alvin Cheung, Karin Kailing, Stefan Schönauer
IBM Almaden Research Center
650 Harry Road
San Jose, CA
{alvin, kkailin, sschoena}@us.ibm.com

1 Introduction

The ability to trace the history of individual products, especially their movement through supply and distribution chains, is key to many solutions such as targeted recalls and counterfeit detection. In most traceability applications a number of independent organizations have to work together (see Figure 1 for an illustration).

EPCglobal [4] has proposed an architecture for a network of RFID databases where each database provides a standardized query interface. That architecture facilitates simple retrieval of traceability data from individual repositories, but it does not support complex traceability queries or cross-organizational query processing.

Theseos [1] provides traceability applications with the ability to execute complex traceability queries that may span multiple RFID databases. Theseos has the following capabilities and benefits:

1. Theseos supports a subset of SQL (including aggregation, joins across databases, and recursion) that is sufficient to express common traceability query types such as pedigree, recall, and bill-of-material queries.
2. Theseos hides data distribution across multiple RFID databases from the traceability application.
3. Theseos allows organizations to selectively share traceability data (a common need for most enterprises).
4. Traceability applications may be thin applications (for example, web applications).

When a traceability application sends a query to its local Theseos instance, the following steps are performed: First, the query is translated to a local query that is executed on the local database. Policy enforcement

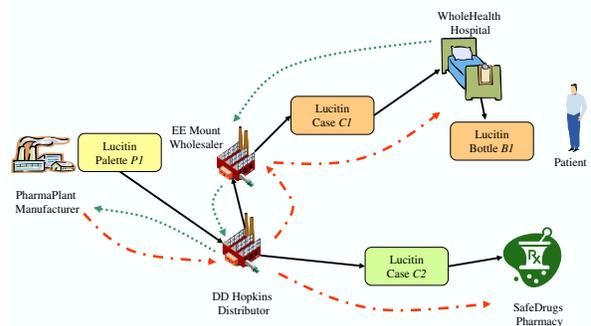


Figure 1. A Pharmaceutical Supply Chain

[5] ensures that in this step only data compliant with the organization’s confidentiality policy is retrieved.

Based on the results retrieved locally, the initial query is further analyzed and, depending on the outcome, rewritten and enhanced by local results. The rewritten query is then forwarded to other RFID databases in the network. The results retrieved from the network are added to the local results. If necessary, post-processing is performed on the results.

Common types of queries that can be answered with Theseos are: Pedigree queries that reconstruct the complete history of an object; recall queries that detect the current location of objects; bill-of-material queries that return information about all component parts of an object. The latter two queries rely on the fact that information about packing and unpacking and assembly of products is recorded.

2 Implementation and Setup

For our implementation we chose the relational model. We implemented Theseos in Java 1.4 as a web service using Apache AXIS 1.2 [2]. For query analy-

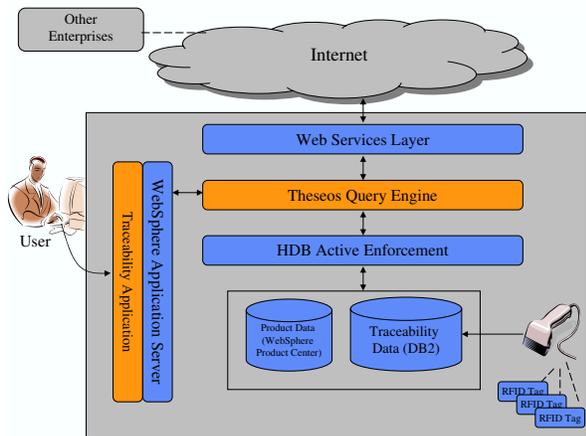


Figure 2. Software Stack at one Enterprise

sis and rewriting we used the QGM model [6]. We also developed a traceability application on top of Theseos using Apache Struts 1.2.7 [3].

For the demo we produced RFID event data (representing reads from smart doors, smart shelves and handheld readers) and other product data for a supply chain with *five* independent enterprises (as depicted in Figure 1). We simulated, pallets of drugs moving from a manufacturer to two distributors where the pallets are broken down into cases and passed on to two retailers where individual pill bottles are dispensed.

The system running at each of the five parties in the demo consists of a DB2 UDB V8.2 instance, WebSphere Application Server, and HDB Active Enforcement [5]. The Theseos query engine and the traceability application are installed at each party. Figure 2 shows the complete stack of software at one party.

3 Scenarios

Figure 1 illustrates our two demo scenarios. The dotted lines show the pedigree generation and the dashed line the targeted recall.

Pedigree Generation. At WholeHealth hospital a patient is treated with a drug called Lucitin. The patient does not show the expected signs of recovery. WholeHealth starts an investigation of pill bottle *B1* (the bottle that contained the pills administered to the patient). WholeHealth uses its traceability application to request a pedigree for *B1*.

Theseos first translates the pedigree query into a local query and retrieves local results. It then analyzes the initial query and detects that the queried information is not completely available locally. It finds out from its local database that *B1* was part of case *C1* and that *C1*

was received from EE Mount, a wholesaler. So, Theseos rewrites the query to ask for all information about *C1* and forwards the query to EE Mount (EE Mount may not know anything about *B1* itself).

The same steps of analyzing, rewriting and query forwarding take place at each party that handled *B1* or one of its containers, until the query reaches PharmaPlant, the drug manufacturer. Each party in the query chain returns only results based on the locally installed confidentiality policy. Once Theseos at WholeHealth has received the results from all other Theseos query engines, it unites them with the local results and returns them to the pedigree application.

Using the resulting pedigree, WholeHealth confirms that *B1* indeed came from the manufacturer stated on the pill bottle. However, the pedigree results also reveal that *B1* may have been tampered.

Targeted Recall. The manufacturer PharmaPlant needs to recall all pill bottles sent out on pallet *P1*. PharmaPlant uses its traceability application to retrieve the current location of all the items on *P1*. The local Theseos instance detects that the pallet was sent to the distributor DD Hopkins and forwards the query to that distributor. The Theseos instance at DD Hopkins detects that *P1* was split into two cases *C1* and *C2*, rewrites the query appropriately, and forwards it to the wholesaler EE Mount and the pharmacy SafeDrugs. EE Mount forwards the query to WholeHealth hospital.

As soon as PharmaPlant receives the result that the bottles on pallet *P1* are now located at WholeHealth and SafeDrugs, it issues a recall notice to only those two parties.

References

- [1] R. Agrawal, A. Cheung, K. Kailing, and S. Schoenauer. Towards Traceability across Sovereign, Distributed RFID Databases. In *Proc. of the 10th Int. Database Engineering & Applications Symposium (IDEAS '06)*, 2006.
- [2] Apache Axis. <http://ws.apache.org/axis/>.
- [3] Apache Struts. <http://struts.apache.org/>.
- [4] EPCglobal. <http://www.epcglobalinc.org/>.
- [5] K. LeFevre, R. Agrawal, Ercegovac, R. Ramakrishnan, Y. Xu, and D. DeWitt. Limiting Disclosure in Hippocratic Databases. In *VLDB '04: Proc. of the 30th Int. Conf. on Very Large Data Bases*, pages 108–119, 2004.
- [6] H. Pirahesh, J. M. Hellerstein, and W. Hasan. Extensible/Rulebased Query Rewrite Optimization in Starburst. In *ACM SIGMOD Conference on Management of Data*, pages 39–48, 1992.