rfrbNet A federated network for goods traceability

Carlos Perdigão
Department of Computer Science and Engineering
Instituto Superior Técnico
Technical University of Lisbon
Av. Rovisco Pais 1, 1049–001 Lisboa,
Portugal
Email: carlos.perdigao@ist.utl.pt

***Abstract*—This document presents the problem of object track and trace. Why is it so important and the state of the art of current RFID based solutions. The main objective is to present a solution that allows the execution of track and trace questions in a federation where ones participation mutates along time.**

Secondly a set of policies were defined for use inside federation to agilize their behavior in order to fasten the adaptation to changes in business processes.

This work is based on EPCGlobal's standards and framework, that define the structure of information, interfaces and naming conventions. Present solutions were also taken into consideration whose downsides this works final solution must overcome.

The solution will be validated through practical results that will be mapped against the European Union recommendations to track and trace systems.

I. Introduction

The world as we know it is quickly changing becoming more global then ever. Open borders, agile transportation, worldwide commerce and the competition between economies are some of reasons of this new reality. In order to adapt, some organizations are abandoning Make-to-Forecast production into Make-to-Order models [1]. This agile manufacturing model has an huge impact on supply chain management, where information sharing between all the participants is mandatory. To make the problem worse, new laws are legislated in order to control the distribution of goods such as ammunition [2] or medicaments [3]. In Europe, in case of medicaments, it is the owners responsibility to guarantee that individual products and its raw materials are tracked through its source, manufacturing, packing, storing transportation and delivery to the final facility where it will be used [4].

It's known that a product during it's life cycle, passes through a succession of states, places, conditions and organizations being transformed, destroyed aggregated or just moved. Registering these occurrences is the first step in any track and trace system.

In 2008 333 million tons of products were transported only in Portugal what demonstrates the size such a system might have in world wide supply chains. This makes impossible to manually control each product.

Nowadays there are two major types of track and trace systems. One based on pedigree records that move along with the object and another is a track and trace system mostly controlled by one of the stakeholders.

The objective is to create a network where peers actively participate in track and trace processes by tracking an sharing information. Any participant will be allowed to query the network for information related to a certain object, wherever it is in the supply chain. This queries should only envolve peers with acquaintance of the object.

This work's goal is to propose an alternative arquitecture, based on present systems, that solve the following requirements:

- Uniform the information in all the systems;
- Set Track and Trace primitives;
- Set Track and Trace algorithms;
- Set authentication and authorizations systems;
- Grant the systems' scalability;
- Mitigate abusive uses;
- Set different level details according to ones role in the supply chain;
- Set permissions on data access.

Additionally, the solution proposed should take in consideration the roadmap for the Internet of Things [5]:

- Allow information sharing between organizations;
- Set and modify access permissions;
- The flow of messages must replicate the movements of the object in the supply chain;
- Keep operational even if some systems crash;
- Access information directly on the systems instead of keeping local copies;

- Support a business language to set permissions on access, information detail, etc;
- Being able to process large volume of data.

II. Related Work

This section will present the state of the art in track and trace systems. It will begin with an introduction to present traceability systems. After that a presentation of new technologies that will be the basis to the track and trace system proposed.

A. Present traceability systems

1) Traceability using Pedigrees: Pedigrees are a "certified record that contains information about products, transactions, destinations and signatures" [6]. Their purpose was to maintain an historic record of the product's entire life cycle in order to identify possible anomalies. To build such an historic record, many dimensions must be registered, such as temperature, pressure, humidity, location, etc.

These systems show some flaws as pedigree records tend to have larger dimensions and are harder to maintain. Usually the pedigree record moves along with the objects and only the organization holding the object can access this information. Information is protected using cyphers and is impossible to adjust the exposure of information according to who's accessing it.

2) Traceability in proprietary systems: The second type of present solutions are traceability systems based in proprietary systems. This type of systems is widely used in automotive industries where the car maker determines the structure of the system and imposes their use to other smaller companies [7]. In this type of systems there are two main roles one that produces information and feeds the system, and another that consumes all the information produced. The first role is mainly attributed to the smaller companies that feed the system and cannot access any information. The second role is adopted by the car maker as it imposes the system to other companies and consumes all the information on his track and trace processes.

This model of systems is impractical in long term considering the characteristics of present supply chains. The track and trace system should be agile, changing processes, operations and flows of information, adapting itself to constant changes in supply chains.

B. Internet of Things

The concept Internet of Things (IoT) is a recent one and refers to the connection between real world objects and their representation in the virtual one. The IoT can be described as an global dynamic network infrastructure with auto-configuration capacities, based on standards and communication protocols where physical and virtual things have identities, physical attributes, virtual personalities, use smart interfaces and are perfectly integrated in the information network" [8].

According IoT vision physical objects become elements in a virtual network being uniquely identified by EPC codes.

C. RFID Systems

Most track and trace systems currently use code bars to identify the product. This technology has many downsides when comparing with modern identification systems such as RFID identification. For example it needs visual contact with the object, needs extra codes to identify each item in the lot, cannot do multiple identifications at the same time, requires manual intervention in detections [9] [10]. Traditional rfid systems are composed by three elements [11]:

- Tag - Physical identifier placed on the product. It's activated by the presence of an electromagnetical field;
- Reader - Has a double function, it creates the magnetic field needed to activate the tag, and reads the information emitted by the tags;
- Software/Hardware infrastructure - The infrastructure where the events are registered and available for querying.

1) EPCglobal: The EPCglobal Network is a group of organizations united with the goal of promoting the use of rfid systems in object identification. It's their goal to set open standards, that might be used in industrial context and this way promote information sharing across the supply chain [12].

EPCGlobal has created standards for every phase involved in object identification, from the data inserted in the tag, the readers properties, interfaces, etc.

In this work the EPC (Electronic Product Code) the EPCIS (EPC Information System) and the Discovery Services take particular interest.

2) EPC: The EPC is used to identify objects up to the unit. A single EPC can identify the lot, producer, category and unit of a single product. This capabilities made experts consider this to be the end of bar codes [13] in a near future. An EPC code is divided in 4 sections and it is possible to identify up to [14]:

- 268 millions producers
- 16 millions object classes by producer
- 6.8 x 10^9 objects per class

TABLE I
EPCIS EVENTS

Event type	Purpose
EPCISEvent	Generic Class for all events
ObjectEvent	Represent observation events that happened in one or more entities
AggregationEvent	Represent aggregation events evolving two or more entities
QuantityEvent	Represent events related with a group of entities identified by a single EPC
TransactionEvent	Represent evento where one or more entities are linked with business transactions

A particular interesting characteristic of EPC is that it can identify both objects, containers, pallets, assembly events, readers, business operations...

3) EPCIS: The EPCIS is a standard for Information Services of the EPC. It's main goal is to define methods to query and provide information. This goal is achieved by answering four questions, "what?", "where?", "when?" and "how?" [15]. There are four types of EPCIS events with whom it's possible to define any objects' status.

The EPCIS has two interfaces to interact with the system. A capture interface to feed information to the repository and a query interface to query the repository for events.

4) Discovery Services: The EPCGlobal Discovery Services Standard is still under development. Although the work is in progress there has been a progress on Discovery Services functionalities. A discovery service main purpose is to allow the discovery of all the information related to a certain object. In the EPCGloval framework, a discovery service will identify every EPCIS that has information about a specific EPC Code.

Using Discovery Services simplifies processes of information sharing by offering a service that links sparse information as it is generated along the supply chain [16].

D. Traceability

Traceability refers to the capability to reconstruct an object's life cycle. This capability is widely used in supply chains where the involved parties pretend to track an object, trace it or create the bill of materials.

1) Track: A product's track is probably one of the more frequent queries in supply chain management. A track query is the capability to identify the last known location of an object. It's frequently used in web portals when clients want to know where a parcel was last seen. The result of a track query should contain the local and timestamp of detection.

2) Trace: Opposing to a track query, the trace query registers an object's entire historic of detections. The result should be a sorted list of locations and timestamps of detections.

A track query can be seen as a simplification of a trace as it's result is the same as the last position in a trace result.

Because of this similarity this work will only consider trace queries as they are more complex than the other.

3) Bill of Materials: An object's Bill-of-materials is in it simplest form a list of raw materials, components, parts and quantities needed to produce a single product [17].

One of the problems related with BoM is the lack of standards for its representation. A BoM to be considered a complete bom, must have the following items [18]:

- Component identification
- Producer identification
- Quantity
- Description
- Sub Components - Hierarchies Identification

E. rfrbNet - federated network for good's traceability

rfrbNet was a project developed by Link Consulting with the objective to study an application of a traceability network for portuguese small/medium companies. Considering the target of the project, it acquires certain characteristics such as:

- Low entry costs;
- Different contributions in products traceability;
- Small size at first with potencial for fast growth.

This work contributed to the beginning of rfrbNet with the definition of traceability queries, entities, algorithms, and initial structure of the network.

III. ARCHITECTURE

This section presents the architecture of the solution. The first steps in the definition of the architecture were based on Agrawal's work on distributed queries for traceability [19]. Although simular this work differs from Agrawal's in some important aspects:

- An organization is obliged to participate on every research;
- All organizations have the same access to information;
- Query is propagated instead of locally solved;
- The organization requesting the information might not be the final user of it.

These differences let to a redefinition of elements.

A. Traceability

The definition of entities, algorithms and finally the architecture followed a top down approach started at a conceptual level and followed by a logical level. At the end a technologic level has defined where technologies were selected to map with elements defined in the previous levels.

1) Conceptual level: The conceptual level was defined to represent a simplification of the problem. Here all the need entities were identified.

Entities are the basic elements in the universe of the problem that represent both physical and abstract objects. Three types of entities were defined:

- Objects
- Organizations
- Events

Objects represent all physical objects that move along the chain. In order to distinguish one from the others each object must have an unique identifier.

Organizations are abstract entities that observe and move objects. They control a well defined set of information, the events.

Events are basic information entities shared among organizations. They represent a specific interaction with objects that took place at a specific time and space.

Figure III-A1 resumes all entities defined for this work.

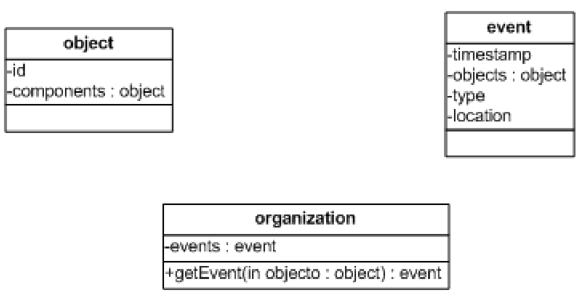


Fig. 1. Conceptual level entities

B. Logic level

Methods and interfaces were defined at the logic level, both implemented by entities. In the conceptual level an hierarchy has defined between the entities: organizations observe objects and from this action results an event.

To make this succession possible some methods must be defined and shared by entities.

An organization must be able to observe and identify both the object and the responsible for it. After this an organization has to share the events related to a certain object. Table II resumes the identified methods with the respective arguments and returns.

Method	Arguments	Return
getId	Object	Identifier
getResponsavel	Identifier	Organization
getEventos	Identifier	Event list

TABLE II
METHODS DEFINED AT LOGICAL LEVEL

1) Algorithms: Once the basic methods have been defined it's time to define both BoM and Trace's algorithms.

a) Bill-of-materials: Bill-of-Materials refers, as stated before, to a detailed description of a product's components. This description may be implemented using a tree where the root is the object, and each leaf a component.

The construction of this tree considers all the EPCIS aggregation events as these are the ones that change an object state (Figure III-B1a).

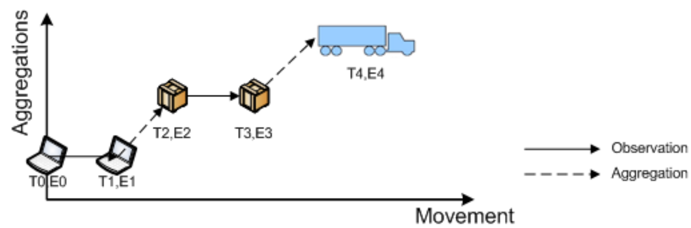


Fig. 2. Events considered in the construction of a BoM

The first step to create a BoM is to identify the producer of the object. The producer is then queried to retrieve all the related information. The analysis of this information adds components to an objects' BoM. These objects might be unknown to the client so this process must be repeated to each component until no new information is obtained.

A BoM's algorithm should be:

1) Identify the object
2) Identify the object's producer
3) Get object's related events from producer
4) Build BoM
5) Repeat 1 to 4
6) Aggregate all temporary BoMs

Figure III-B1a represents the flow of operations in a conceptual diagram.

Table III resumes the methods defined and maps them with the methods defined in the logical level.

b) Track and Trace: Track e Trace queries refer to the capability to locate a product in time and space. The ideal structure to represent this historic is a sorted list, ordered by chronological order.

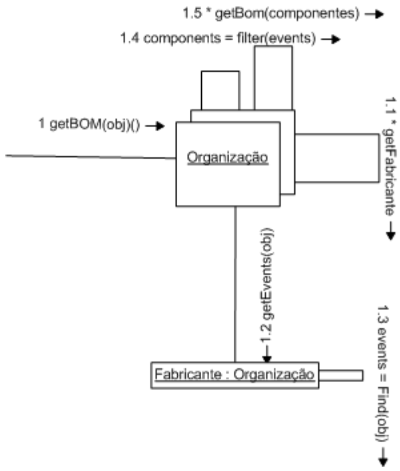


Fig. 3. Conceptual diagram of BoM algorithm

Method	Logic level	Arguments	Return
getBOM	getId	Object	BoM
getFabricante	getResponsavel	Identifier	Organization
getEventos	getEventos	Identifier	Event List
getBOM	getId	Component	BoM

TABLE III
MAPPING BETWEEN METHODS DEFINED IN LOGICAL AND CONCEPTUAL LEVELS

The construction of this list obliges to an analysis of both aggregations an observations. Observations allow to identify time and space were an object was observed. Aggregation allow to identify changes in the object structure such as transportation inside containers. Figure III-B1b represents both events and how they are used in a Track or Trace query.

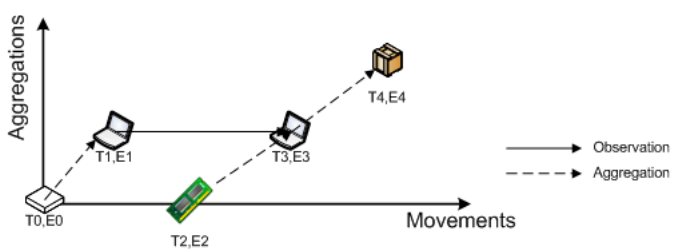


Fig. 4. Events considered while building a Track or Trace

Opposing to a BoM query, this events are scattered along the entire supply chain, so all the organizations that interacted with the object or it's containers, must be contacted.

Trace's algorithm will be:

1) Identify the object
2) Identify all the organizations that interacted with the object

Method	Logic level	Arguments	Return
getTrace	getId	Object	Trace
getHandlers	getResponsavel	Identifier	Organization
getEventos	getEventos	Identifier	Event list
getTrace	getId	Container	Trace

TABLE IV
MAPPING BETWEEN LOGIC AND CONCEPTUAL LEVEL METHODS

3) Get from each organization all the related events
4) Build object's trace
5) Repeat steps 1 to 4 for each container
6) Aggregate all temporary traces

Figure III-B1b presents the conceptual diagram of a trace query identifying all the operations.

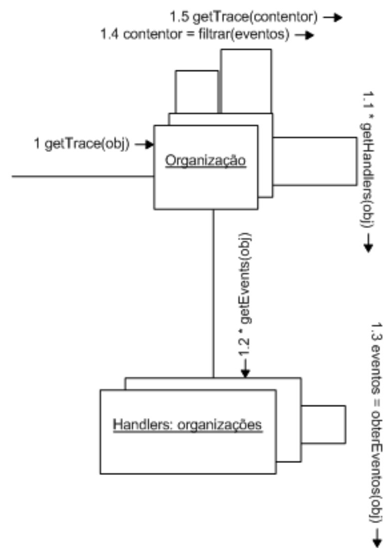


Fig. 5. Trace's conceptual diagram

Table IV resumes the operations described and maps them with methods defined at conceptual level.

C. Technological Level

The last phase in the definition of the architecture is the selection of a set of technologies to implement elements identified in previous levels.

The first entity defined was an object. An object will be represented by an EPC code an it's identification will be realized using rfid tags. EPC codes will identify both objects and locations in a hierarchical manner. This means that a shelf located inside a Warehouse in a certain city will be identified by A:WarehouseB:Shelf22. This way everything is uniquely identified inside and outside an organization border.

The second entity defined was the organization. This entity is an abstract entity whose representation in the

Conceptual Level	EPCIS Event	EPCIS Observation	EPCIS Aggregation
Event Type	action(Observe)	Mandatory	-
	action(Add)	-	Mandatory
	action(Delete)	-	Mandatory
Objects	epcList	Mandatory	-
	parentId	-	Mandatory
	childEPC	-	Mandatory
Place	bizLocation	Mandatory	Mandatory
Time	eventTime	Mandatory	Mandatory

TABLE V
ATTRIBUTE MAPPING BETWEEN CONCEPTUAL LEVEL AND EPCIS EVENTS

architecture will be a set of modules. Each organization will control and manage a well defined set of events.

Finally the third entity identified was the event. Events as defined could be one of several types, would register both time and space and finally would have a list of related objects. Table V present a mapping between events and their representation in EPCIS events [20].

Other attributes weren't considered as they don't have an active role in a products traceability and therefore are considered optional.

1) EPCIS + Discovery Services: With all the elements defined the definition of the first federated architecture has straight forward. Many elements from the EPCIS standard were used and just a few elements were custom made. Among them can be found the client application and the discovery service.

The architecture uses Discovery Services to solve the traceability queries. It was considered that the discovery services were under control of the federation and that each organization was responsible to feed them with information during the transportation of goods.

The Discovery Services implement the methods getHandlers and getFabricantes previously defined.

Figure III-C1 presents the architecture with all the existing elements identified.

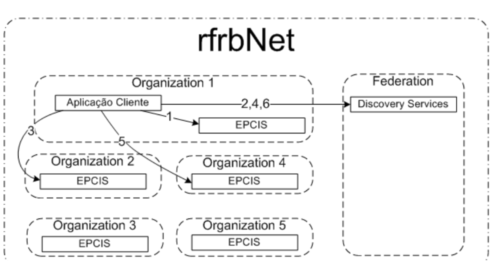


Fig. 6. EPCIS + Discovery Services Architecture

IV. POLICIES

Until now basic functionalities were added to the architecture. Right now it is possible to execute different queries on the supply chain. The next step is to turn the architecture into a dynamic system, were responses to queries change according to the client performing the query. To achieve this dynamic behavior and to simplify the representation and management of the federation, all the organizations were organized into groups and roles. A group would identify all the elements that participate on a specific supply chain and the role would identify it's functional role in the supply chain. Four functional roles were identified:

- Producer
- Distributor
- Store
- Maintenance

With these four roles it's possible to identify all active parties in traceability and at the same time a simplification of the problem is achieved.

Once the groups and roles have been defined the next step is to use policies to add a dynamic behavior to the federation.

Analyzing a supply chain it is common to identify a group of organizations that today compete on a certain resource but in a near future must collaborate on the production of a different product. The network must be able to easily adapt to these situations. In order to achieve that adaptation the concepts of policy and rules must be introduced. IBM classifies policies as being a demonstration of intent or guidance and rules as a specific implementation of a policy [21].

This work considers operational policies that should define for each group/role, a sharing rule. This rule should determine the level of detail for the information returned for the client.

Each pair of elements in the federation should agree on a rule for sharing. In case of omission the rule for the group should be applied, after this the rule for the role, and finally if none exist, a standard rule for organizations that don't belong to the federation. After agreeing on all the rules to apply, each organization should define it's policies and store them at a public repository on the federation.

As stated before, each rule should define a level of detail for a specific type of information. This work only considered information envolved in the tracebility process such as:

- Space
- Time
- Event type

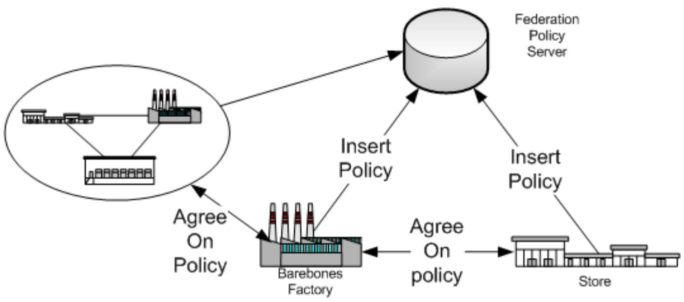


Fig. 7. Policy agreement process

TABLE VI
EXPECTED RESULTS AFTER APPLYING FILTERS

Element	Original	Level 1	Level 3
Space	A:CityA:StorageB :Shelf22	A:CityA:StorageB	A
Time	2011-01-15 09:31:15	2011-01-15 09:31:00	2011-01-15

- Object type

Table VI exemplifies the expected result after applying filters to data existing in EPCIS.

In order to apply and force the execution of filters a new module should be added to present architecture. This module should replace the EPCIS as the target for the client application and expose the same interface. This would allow to add the new module with no impact on the previous architecture. The only difference would be the endpoint returned by the discovery services that should point to the filter instead of the EPCIS.

The filter module should contact the policy repository to get the active policy for the client. After that it should query the EPCIS to get all the original data the client would access in the previous architecture, and the apply all the filters on this set of information.

The figure 8 presents the final architecture for the traceability federation.

V. IMPLEMENTATION

After the definition of the architecture the solution was implemented in order to test and validate the requirements defined for this system. To agilize the implementation process, many open source projects were adapted. Figure V resumes the technologies used.

The following section will present the custom modules developed in particular the filter and policies modules.

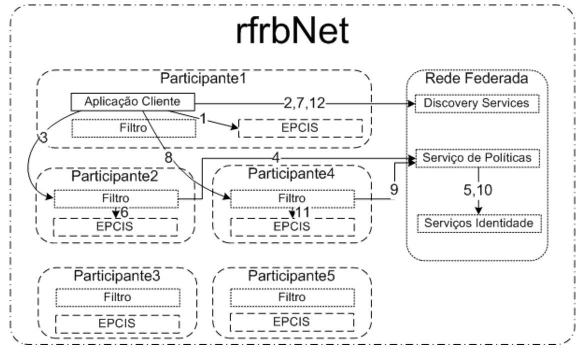
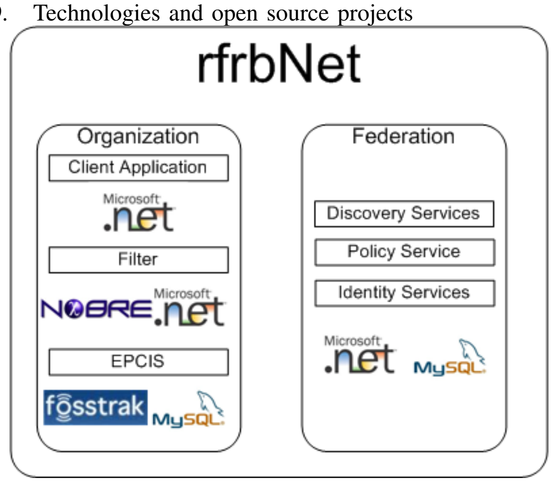


Fig. 8. Final Architecture

Fig. 9. Technologies and open source projects



A. Filter

The EPCIS filter was implemented using the .Net framework to expose the web-services, and to connect to the policy server in order to get the active policy. This component exposed the same interface has an EPCIS and used an inference engine (NxBre) to calculate the policy result. The inference engine received 5 arguments: the group and role of the client, group and role of the provider and finally the active policy.

The active policy was an xml file divided in 3 sections:

- Variable definitions;
- Clients group and role identification;
- Result of the evaluated policy.

These sections are exemplified in the following code excerpts.

Policy - Variable definitions:

```
<xBusinessRules xsi:noNamespaceSchemaLocation=
    "xBusinessRules.xsd" xmlns:xsi="http://www
    .w3.org/2001/XMLSchema-instance">
<Integer id="components" value="1"/>
...
<ObjectLookup id="Producer" objectId="group"
    member="producer"/>
...
```

```
<ObjectLookup id="MyRole" objectId="myself"
    member="RoleId"/>
```

Policy - group and role identification:

```
<Logic>
  <If>
 <And>
      <Equals leftId="Group" rightId="MyGroup"
          />
    </And>
    <Do>
      <Logic>
        <If>
          <And>
            <Equals leftId="Role" rightId="
                MyRole"/>
          </And>
...
```

Policy - Result evaluation:

```
<Evaluate id="PolicyResult">
  <Parameter name="TagType" valueId="
      componentscontainers"/>
  <Parameter name="EventType" valueId="
      aggregationsobservations"/>
  <Parameter name="Detail" value="-1"/>
</Evaluate>
```

VI. Evaluation

The execution and architecture's behavior was evaluated on a functional manner, an tested using a prototype. From the functional requirements defined one couldn't be tested and the remainder were successfully validated.

Due to a lack of real track and trace data, and the inoperability of the simulator it was impossible to test the architecture using different types of scenes, configurations and load. This way the system's scalability was not tested.

The remainder requirements were validated due to the design of the architecture or the functional tests realized with small amounts of data.

VII. Conclusions

Product traceability is an important process in supply chain management that many organizations need to automatize. Present systems lack in the capability to add dynamic behavior to systems in order to quickly adapt to changes in the supply chain. This paper presented a solution of a federated traceability network based on current standards. The main contribution of this work is the definition of filters that used above current EPCIS and whose behavior is determined by the active policy in the moment. The solution proposed was successfully tested in a controlled environment and most functional requirements were achieved.

References

[1] J. Gattorna, *Dynamic Supply Chain Alignment: A New Business Model for Peak Performance in Enterprise Supply Chains Across All Geographies, Chapter 4*. Gower Pub Co, 2009.

[2] C. Europeia, " Directiva 2008/43/CE da Comisso, de 4 de Abril de 2008, que cria, nos termos da Directiva 93/15/CEE do Conselho, um sistema para a identificao e rastreabilidade dos explosivos para utilizao civil." 2008.

[3] ——, "Relatrio da Comisso ao Conselho e ao Parlamento Europeu sobre a aplicao do Regulamento (CE) n.1830/2003 do Parlamento Europeu e do Conselho relativo rastreabilidade e rotulagem de organismos geneticamente modificados e rastreabilidade dos gneros alimentcios e alimentos para animais produzidos a partir de organismos geneticamente modificados e que altera a Directiva 2001/18/CE," 2003.

[4] P. Europeu, " Commission Directive 2009/120/EC of amending Directive 2001/83/EC on the Community code relating to medicinal products for human use as regards advanced therapy medicinal products." 2009.

[5] C. O. T. E. COMMUNITIES, "Internet of Things An action plan for Europe, Communication from the commission to the European parliament, the council, the European economic and social committee and the committee of the regions, Brussels," 2009.

[6] EPCGlobal, " Pedigree Ratified Standard, 3 Version 1.0," 5 Janeiro 2007.

[7] C. R. Karin Murthy, "A Model-based Comparative Study of Traceability Systems," IBM, Tech. Rep., 2008.

[8] A. de Saint-Exupery, "Internet of things Strategic research Roadmap," 2009.

[9] G. P. Gareth R.T. White, Georgina Gardiner and A. A. Razak, "A comparison of barcoding and rfid technologies in practice," *Journal of Information, Information Technology and Organizations*, pp. 119–132, 2007.

[10] H. Clampitt, "RFID vs Bar Codes," 2005.

[11] B. P. L Zhekun, R Gadh, "Applications of RFID technology and smart parts in manufacturing - Proceedings of DETC," in *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2004.

[12] D. L. B. Edmund W. Schuster, Stuart J. Allen, *Global RFID: the value of the EPCglobal Network for supply chain management, Part I Chapter 3*. Springer, 2007.

[13] K. M. Luke McCathie, "Is it the end of barcodes in supply chain management?" in *Collaborative Electronic Commerce Technology and Research Conference LatAm*, University of Talca, Chile, 2005.

[14] D. L. Brockc, " White Paper, The Electronic Product Code (EPC), A Naming Scheme for Physical Objects," Massachussets Institute of Technology, Tech. Rep., 2001.

[15] EPCGlobal, "EPC Information Services (EPCIS) version 1.0.1 Specification," 2007.

[16] S. Beier, T. Grandison, K. Kailing, and R. Rantzau, "Discovery services-enabling rfid traceability in epcglobal networks." in *COMAD'06*, 2006, pp. 214–217.

[17] N. R. Reid, R. Dan; Sanders, "Operations Management," 2002.

[18] N. E. M. I. Inc., "In Search of the Perfect Bill of Materials (BoM)," 2002.

[19] 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.

[20] F. Thiesse, C. Floerkemeier, M. Harrison, F. Michahelles, and C. Roduner, "Technology, standards, and real-world deployments of the EPC network," *IEEE Internet Computing*, vol. 13, no. 2, pp. 36–43, 2009.

[21] M. Hondo, J. Boyer, and A. Ritchie, “Policies and rules improving business agility: Support for business agility; demonstrating the value that can be derived from using policy and rules,” *IBM Developer Works*, March 2010.