Implementing EPCIS services in BizTalk RFID

Pedro Daniel Parreira Ferreira
Instituto Superior Técnico
Universidade Nova de Lisboa
pedro.ferreira@ist.utl.pt

***Abstract*— Object transportation processes are becoming more complex with more players and involving a wide variety of information systems. The RFID technology has the potential to improve these processes, and therefore became an effort for their standardization led by EPCglobal. The EPCIS is one of the EPCglobal architecture layers that provides services for information sharing.**

The problem addressed in this work is the need to implement the EPCIS standard for sharing information contained in RFID tags between business partners that have different and heterogeneous information systems. In this sense, we identified the need for an implementation that facilitates compliance with the EPCIS standard for RFID middleware already in place. To demonstrate the functionality and usefulness of the developed solution, an extension to the BizTalk RFID middleware was successfully carried out, providing information to other independent and heterogeneous systems, according to the EPCIS standard. These developments included the preparation of web services, a repository for storing events and interfaces that meet the EPCIS standards of the EPCglobal architecture.

***Keywords* — RFID, BizTalk RFID, middleware, EPCglobal, EPCIS**

I. Introduction

Over time, supply chain processes are becoming more complex with more players. An object, from its origin to its final destination will pass through several players, each with their tracking and monitoring systems.

RFID (Radio Frequency Identification) [1] has been imposing on this environment and has been used more often in supply chains [2]. Despite the cost associated with used equipment and labels, this technology has advantages over bar code, making the process more simple and effective, notably [1]: The possibility of storing more information; Improved security, impossibility to read without proper readers and also

data can be encrypted; Automatic reading of the label, without human intervention; No need line of sight to be read off; Physically distributed information, the information may not be stored in a central system and may be stored on a label attached to the object itself, making possible to read at any place, anytime.

As the number of people interested in this RFID technology has increased, it became necessary to create rules to streamline the operation. The international organization EPCglobal, together with leading companies, has developed standards for the communication of information, thus creating the standard Electronic Product Code (EPC) which has become the standardization prevalent in the community RFID [3][4].

EPCglobal has created the EPC network [5] which was designed and implemented aiming to standardize the sharing of information on objects throughout all Internet network. This network was built taking into account technologies and web standards internationally recognized.

The component EPCIS (EPC Information Services) [6] arises in this context as one of the layers of the EPC network, which is responsible for providing services for information sharing.

The fast development of RFID applications, coupled with the advantages that this technology brings, led to an increase of the number of companies interested in participating in the development of systems and devices of this technology. Faced with various RFID applications and with various communication protocols, companies question themselves on how to make the existing information systems binding to the RFID readers. The essence of the matter is the interface between the information system and the RFID hardware, RFID middleware. This middleware does an abstraction of the hardware, presenting an interface for high level applications.

Thus, the EPCIS layer allows the compatibility of different RFID middleware through a set of interfaces and standardized services.

II. Problems and Objective

In a typical architecture of an RFID system, data are captured from the tags by readers, collected, filtered and stored in the database, and from these data, events with business information are generated. The business evolution leads to, over time, various partnerships with other companies, requiring that independent information systems communicate between them in order to achieve effective coordination. For communication between systems, established rules that define interfaces and communication protocols must be followed.

We will work in the scenario where an RFID system already exists but there are no interfaces that comply with EPCIS.

As an objective for this work, we propose the development of a solution to facilitate the adoption of EPCglobal standards for information systems, concretely the EPCIS. To demonstrate the functionality and utility of the implementation, we use Microsoft Biztalk RFID middleware and extend it to provide information to other independent and heterogeneous systems, according to the standards.

To ensure that the solution can be used in a real and business environment, validation and integration tests, with other existing solutions that meet the same standards, will be conducted.

To achieve these objectives we need to understand the existing needs, verify the developed work in this area that can be used and, finally, study the standards and recommendations of the community with more experience and knowledge in the area, so we can integrate our work in the best ideas and solutions developed.

III. Related Work

A. EPCglobal

The recommended architecture by EPCglobal organization is a standard of the RFID technology, and arose from the need for commercialization of this technology by several companies in the technology and distribution market [7]. This organization has established a set of global standards of the GS1[1] system, which combines RFID technology with the infrastructure of communication networks and the EPC [8]. Moreover, it even developed a framework aiming to identify and locate, in an immediate and automatic way, one object on its journey along the supply chain, establishing a set of services that can be used by business partners, to seek and access large amounts of information from each EPC. This information is only shared among authorized users.

1) EPCIS

The EPCIS is a component of the architecture recommended by EPCglobal, whose goal is to allow that applications and RFID systems share EPC data. Thus, it is possible that the participants in the EPC network, will acquire a common view of EPC objects within a context of relevant business.

As illustrated in Figure 1, the component EPCIS is located on top of the EPCglobal architecture framework, above the level of raw data readings of the EPC (hardware) and also above the level of consolidated and filtered data (middleware).

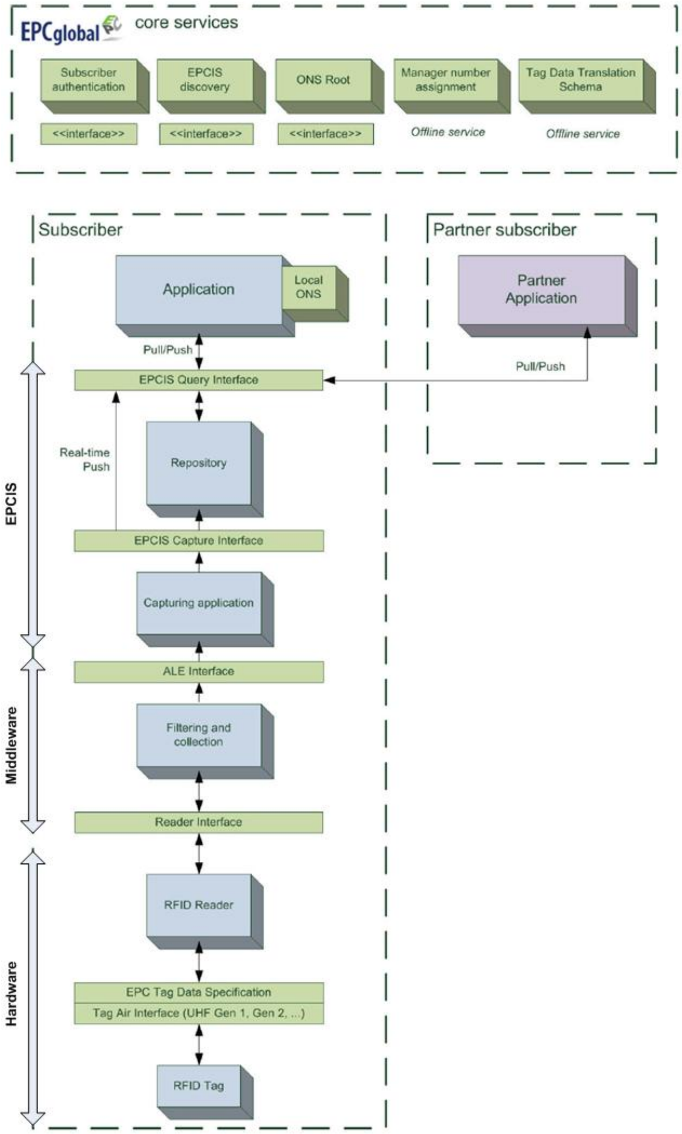

Fig. 1 – Components of the EPCglobal architecture

This component is responsible for capturing EPC events coming from the lower layers (ALE middleware) and makes them available for queries and subscriptions that may be made by business partners. This component uses data from the lower layer and adds the business context coming from other sources such as ERP and WMS, and save this result in the repository , in order to be able to serve other information processing tasks.

The EPCIS component consists of different layers, the Capture Interface Repository and Query Interface (Control and Callback).

B. Fosstrak

As mentioned before the EPCglobal community developed a

[1] http://www.gs1.org/

set of standards[2], such as ALE, RP, LLRP, DTT and EPCIS. The Fosstrak is an open-source system, available for free use for tracking and monitoring objects that implement those standards.

In this context it is noteworthy that this solution aims to provide key components for applications in the EPC network, train users of the EPC network, leading to adoption and promoting the use of EPCglobal standards in education and research and to facilitate prototyping.

The Fosstrak project (initially called by Accad) was started by a group of distributed systems and the Auto-ID Lab from ETH University in Zurich, targeting developers, new users of EPC, system integrators and university and industrial research groups [9].

According to Pereira [10], who did a study on the possible use of Fosstrak in a supply chain, the Fosstrak is prepared only for prototyping, since there are occasional flaws for small production systems and presents a difficult integration with rules business logic, when this is applied in a real environment.

C. RFID Middleware

By analyzing the available middlewares on the market, we note that, at the moment, there are several systems that are owned by companies in the area of RFID that have built their own architecture similar to that recommended by the organization EPCglobal [11]. Alguns exemplos são o Siemens RFID middleware[3], o Sun Java System RFID Software[4], o Oracle Sensor EdgeServer[5], o IBM WebSphere RFID Premises Server[6] e o BizTalk RFID[7]. Although these systems are very efficient regarding to the processing of RFID data, do not fully consider the EPCIS specification that is responsible for sharing information, which would be very useful for the development of different components by various organizations, so that they are easily integrated [12].

1) Biztalk RFID

Do not confuse BizTalk RFID with BizTalk Server, which will be described later. BizTalk RFID is an RFID middleware while the BizTalk Server is an integration server.

The Microsoft BizTalk RFID middleware [13] facilitates the installation and configuration of RFID systems, providing a uniform way to discover, communicate and manipulate RFID devices in a Microsoft Windows environment [14].

The success of Microsoft's approach stems from the possibility of adding new layers of software technology, allowing that all types of devices may be incorporated in a plug-and-play way [14].

This infrastructure works with a line of business applications such as ERP, WMS and other more specialized software, allowing some flexibility [14].

BizTalk RFID, with its extensible architecture and security features, offers a platform that facilitates the development of a wide variety of RFID solutions. Its architecture was designed so that it is possible an easy transition from a low to a high volume of RFID installations. With the DSPI interface, new devices are added easily.

a) Comparison between BizTalk RFID and EPCglobal architecture

EPCglobal has established a software certification program to ensure that systems adopt a set of rules established by it.

BizTalk RFID fits into the architecture recommended by EPCglobal, playing the role of layer Filtering & Collection and Capture Application Layer. BizTalk RFID is responsible for filtering and storing the information coming from readers after the reading of RFID tags (Filtering & Collection) and also contains a simplified version of Capture Application, which converts the EPC events in EPCIS events and sends them to the Capture Interface.

2) BizTalk Server

The Business Process Management (BPM) server, Microsoft BizTalk Server, is a product that enables medium and large organizations to automate their business processes.

BizTalk gives a central point of integration between systems that do not share communication protocols, providing a wide range of features, including most importantly: communication services and processing of XML messages, the orchestration services that automate the various processing steps (workflow), the ability to interact with external systems via proper adapters, the possibility of integration with Web Services [15].

The communication between BizTalk and other systems is done by exchanging messages. The BizTalk messaging engine receives messages from other systems, identifies the layout, extracts the message elements, applying forwarding rules (routing), and delivers messages to the destination. The processing applied to each message is defined by one or more orchestrations, and during this process is used a database called the MessageBox.

BizTalk Server provides some features, the automation of business processes, the modeling of business processes, Business to Business (B2B) communication, enterprise application integration (EAI) and messaging broker.

3) rfrbNet

Upon completion of this thesis, Link Consulting was in the

[2] *http://www.gs1.org/gsmp/kc/epcglobal*
[3] *http://www.automation.siemens.com/rfid/en/competence-in-rfid.html*
[4] *http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/sjsrfid/RFID.html*.
[5] *http://download.oracle.com/docs/cd/B14099_19/wireless.1012/b13819/rfid.htm*
[6] *http://www.redbooks.ibm.com/abstracts/sg247147.html*.
[7] *http://www.microsoft.com/biztalk/en/us/rfid.aspx*

process of developing a distributed platform with the goal of tracing assets equipped with RFID tags in open chains, allowing an easy membership and interconnection between the members of the supply chain.

This project called rfrbNet (Integrated Network of Asset Tracking) respects the following challenges:

- Entity Federation - Entities involved in the transport and control of assets should be able to register in the network and provide their monitoring services to other entities that can use them whenever it wishes to locate an asset, or know the course of it.

- Scalability - Nowadays, the tracking of assets throughout the involved organizations in the supply chain is something that often happens to the lot, pallet or other gathering unit. The use of the item level raises a problem of scale and volume of data that suggests distributed and decentralized solutions as a way to distribute the large amount of generated information.

- Privacy - It is important to consider mechanisms to ensure the minimum security and privacy requests on tag's information, while ensuring that the information provided in response to several surveys by the users along the route tags takes into account each permissions.

- Security - Along the supply chain, the tags have to cross several security domains, both at physical and logical level, so the proper adjustment to the various policies is crucial.

The main idea of this platform is to allow that the different participants of the supply chain have the server on its installations, using the platform just to share information with other participants, or have their server hosted in EPCIS platform.

The project purpose is to have lower implementation costs by scale economy effect and reuse of the structure, allow greater flexibility in the federation to the network, increase the capacity to integrate smaller players who currently don't have the capacity to enter into the traceability channels by RFID and improve the speed of assembly the traceability process.

The relevance of rfrbNet for our work is the opportunity to do some integration test that ensure communication with other systems that meet the EPCglobal recommendations.

IV. Solution

The developed solution takes into account the architecture shown in Figure 2. The following diagram illustrates the work done and the decisions taken for each of the components of the architecture, which will be explained and justified in this chapter.

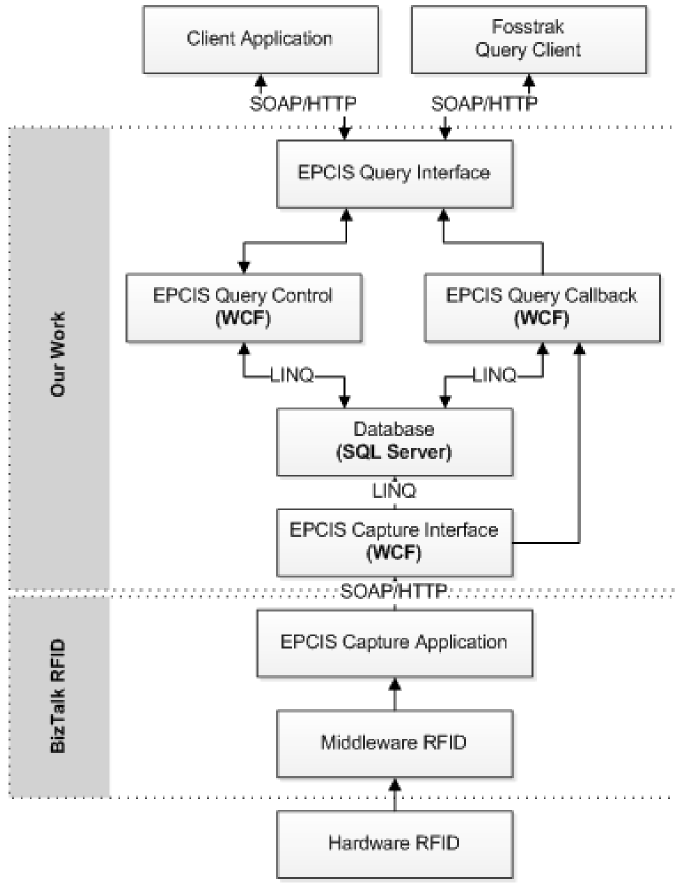

Fig. 2 – Diagram of the solution architecture

A. Repository

This layer contains the repository of EPCIS events and the subscriptions. Creating this database we made sure that a structure in which the loading of sample data Fosstrak solution was as simple as possible.

As the Fosstrak solution database is in mySQL while our solution is in SQL Server, to load the sample data was needed to make a conversion from the SQL script. To do this we used the conversion tool "Microsoft SQL Server Migration Assistant 2005 for MySQL" and the conversion was straightforward.

This step was very important to compare the results of our solution with the results of the Fosstrak solution.

The database was created using the tool SQL Server and we used LINQ (Language-Integrated Query) to do the queries.

B. Capture Interface

For this layer we developed a WCF service that was created according to the recommendations of EPCglobal. This layer receives an EPCIS event, sends it to the Query Callback layer and stores it in the repository, using the LINQ technology. The table of the database where the events are stored depends on the type of the captured event. For example, if is received an event of type object, the event will be recorded in the table called event_objectEvents.

We avoid the bond to a specific Capture Application, with this interface you can replace the Capture Application without making any changes, since it meets the standards.

C. Query Control

This layer consists of a service created with the help of WCF technology and this service has the purpose to allow queries (Poll) and allow queries subscriptions (Subscribe).

When we make a query, this service search for events in the repository, assuming the parameters submitted by the user. If it finds results, sends them to the upper layer, the Query Interface, in order to be delivered to the user that requested it.

The queries that can be made on the database are not free, we defined a set of parameters for the queries. The results are sent to the user through a SOAP message which will always have the same structure, differing in the number of returned events.

D. Query Callback

This layer, like the previous one, consists of a service created with the help of technology WCF and this service has the purpose to finding registered subscriptions and deliver the results to the respective receiver.

This mechanism has a thread created every minute (a minute interval is configured by default) that checks for subscriptions. When is detected a subscription is created a new thread (a thread for each found subscription) to perform the subscribed query and to send the results to the address specified by the user. With this thread mechanism we allowing the existence of parallelism in the delivery of results, preventing a sequential delivery of results and avoiding there are users who receive results much later than others.

The system knows that the user did not received certain events because whenever a search is performed, the date of the last execution is stored , so the system only send the events that occurred after that date.

As already mentioned it’s possible to use a triggered subscription instead of a schedule. When an event is received, it is checked whether there is any subscription for that event. If so, the event is sent to the address specified in the subscription. This alternative allows the user to receive near real time events. All these events are sent directly from Capture Application layer and are stored in the database.

E. Query Interface

Since it is intended that our solution would follow the standards recommended by EPCglobal and Fosstrak presents a solution that meets these standards, we chose to use the EPCIS service description that is provided by Fosstrak.

We used the tool WSCF.blue to generate the interface from the WSDL Fosstrak and we found some problems at this stage of development.

The WSDL used was complex, contained in imports of XSD files and the tool could not identify their location. For identify the XSD files they also need to be added to the project, in addition to the WSDL.

The WSCF.blue uses the tool “xsd”, which is part of .NET framework, and during the development was found an error interpreting complex data types, in this case, the multidimensional arrays.

The error was reported to Microsoft, which confirmed the existence of error and that was not yet part of their plans to address them to the tool. To solve the problem a workaround that fixed the problem was suggested, described in the following figure. The workaround was to introduce an attribute that, although not used in the solution, enables the use of ArrayOfString type.

```
<xsd:complexType name="ArrayOfString">
    <xsd:sequence>
      <xsd:element name="string"
type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
    <!-- workaround -->
    <xsd:attribute name=" OptionalAttribute "
type="xsd:string" />
</xsd:complexType>
```

Fig. 3 – Workaround to solve XSD file import problem.

Another problem that arose in this layer was that in requests made by the client application Fosstrak, the SOAP messages contain the field "soap: action " without a value. On the server was required to be an action and could not exist different methods with the same action. To circumvent this problem we had to change the dispatcher service, by creating a "custom dispatcher. " The dispacthers are responsible for translating messages received in calls to the application method, and sending the results back to the caller of the service. With this amendment, the dispatch has to be done through the method name instead of the action name.

F. Security

In order to consider a secure system, this should include auditing, authentication, authorization and integrity, and all these requirements are covered by the WCF technology features [16].

WCF provides access to these features by setting bindings and behaviors of services. The bindings define the way of security, credentials and other security settings, while behaviors define the way of customers credentials are authenticated and authorized and also the credentials of the service.

In this solution has not been set any of these security features, but we did want to emphasize that, with this solution, is possible to configure any security mechanism supported by WCF.

V. Evaluation

A. Use Case

The Figure 4 illustrates a scenario of using an EPCIS system in the supply chain[17]. Depicts the stages that a product goes through from its production to its delivery to the customer. In each of these stages there is an RFID reader that detects an object with a tag, an event is generated and is sent to EPCIS Capture Interface.

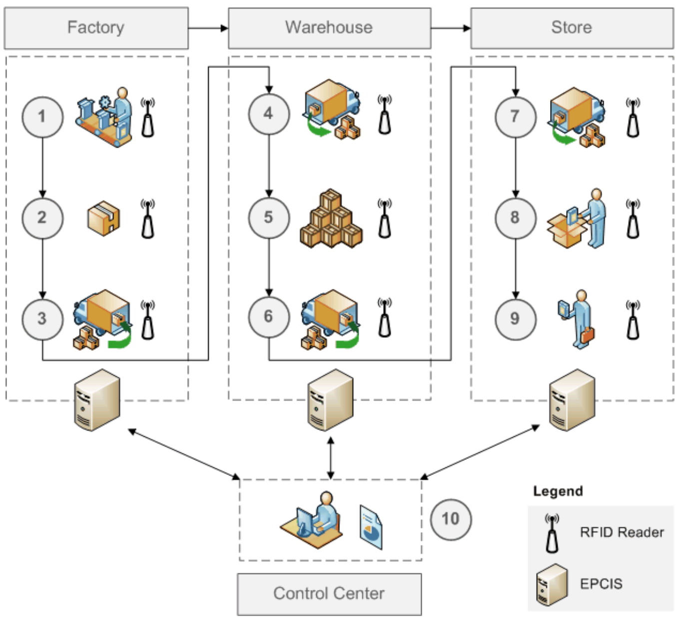

Fig. 4 - Exemplification of an use case.

1. Product manufacture and tag assembly
2. Palletizing of manufactured objects
3. Pallet loading onto the transportation vehicle
4. Unloading the pallet from the transportation vehicle
5. Inventory
6. Pallet loading onto the transportation vehicle
7. Unloading the pallet from the transportation vehicle
8. Depalletization of the objects manufactured
9. Product sale
10. Query information

B. Integration with Fosstrak

To test the integration of the developed solution with other solutions that comply with the EPCIS standards set by EPCglobal, we simulated the send and query of information on two EPCIS systems with different technologies but with the same data in the database.

To make sure that the data are the same in different systems, we replicate the data from the Fosstrak database to our database.

The same tests were executed at two interfaces, Capture and Query Control, using the Fosstrak client applications that have some real scenario examples. We wanted it to confirm if the behavior of both systems is identical.

C. Integration with rfrbNet

As the issues …are solved

As the issues related to provisioning, discovery and policy actors are solved, the rfrbNet network keeps its primitive traceability thanks to the Capture and Query interfaces standardized by EPCglobal.

Initially, the network was built on the rfrbNet Fosstrak platform, which, as already mentioned, is an RFID middleware platform open source repository which has also been certified by EPCglobal[8]. Since the capture layer of the rfrbNet network is based on BizTalk RFID middleware, our work will present an alternative to the EPCIS interfaces Fosstrak.

By testing our solution with the rfrbNet system, we wanted to validate that our solution don't have errors and use a business system that already has a set of tests. The idea was to run the set of tests using the EPCIS services of Fosstrak in the rfrbNet system, then replace these services by our solution and verify that the results are the same.

The replacement of the implementation of the EPCIS interfaces Fosstrak by our solution produced a positive result, since we obtained the same results after replacement.

D. Performance Tests

To test the performance of our solution there was a set of tests performed at both systems, our solution and Fosstrak.

These tests consisted in populate databases of both systems with a large number of records (equal for both systems) and measuring their query response time.

Number of returned events	Average Response Time (ms)	
	Fosstrak	Our Solution
1	60	50
10	108	162
100	276	414
1000	1657	2488
10000	21015	31554

Table 1 – Performance tests results

[8] *http://www.gs1.org/epcglobal/certification/sw_cert*

Our solution just has lower response time for a single record. For more records, it presents a lower performance, on average, it takes 1.5 times the Fosstrak's response time. We propose, as future work, to identify the problem and optimizing system performance.

VI. CONCLUSION

As an objective for this work, we proposed to develop a solution that would facilitate the adoption of EPCglobal standards for information systems, concretely the EPCIS.

So we decided, initially, to develop EPCIS services that communicate with Biztalk RFID, allowing him to provide EPC data to other systems. Although developments have been made to this middleware, the solution may be used by any middleware that complies with EPCglobal standards.

We complied with the recommendations of the EPCglobal developing a solution by layers that has the advantage of having standard interfaces independent from implementation, and extensible to be used in any business.

We developed a solution that can be installed on any system that meets the standards. This solution has been valued by the company Link Consulting because it can be used in future projects that have the need to implement the EPCIS service, guaranteeing the possibility of a future replacement for more robust implementations (eg IBM and Oracle). This solution, when compared with a solution built on BizTalk Server, has the advantage of not having licensing, thus allowing the possibility of the implementation of small projects without high cost.

The existence of a layered architecture facilitates the replacement of more advanced layers that conform to EPCglobal, allowing a natural evolution and scalability of the solution.

To meet our primary objective, several steps were necessary, including the study of EPCglobal standards and the RFID middleware analysis. We highlight the importance of an academic solution, Fosstrak, which allowed the comparison of results. The whole mechanism behind the services is not simple, there are publish/subscribe mechanisms, verification of subscriptions has forced the creation of a timer that makes periodic checks.

The compatibility with the defined interfaces involved a "contract-first" approach in the integration of services via SOAP. The contracts specified by EPCglobal don't define the field "action", which is mandatory in WCF to determine the operation. The solution was to change the dispatcher service, creating a "custom dispatcher." With this change the dispatch has to be done through the method name instead of the action name.

The evaluation has fulfilled our expectations, with the exception of performance tests, and results of the tests were always the correct results. We considered that the use cases identified resemble real situations in business reality.

The results of performance tests left us surprised, giving response times 1.5 times higher than the response times of the Fosstrak solution. Despite showing a lower performance, we believe that the solution returning 333 records per second (average) are acceptable, considering that these records correspond to the most important moments of the object movements in a supply chain.

We guarantee that the EPCglobal standards are met by comparing our solution with another that already meets them and we proved that our solution can communicate with other systems of different technologies.

With the solution that was developed and the results of our tests, we believe that this work is an important step in building a solution that can be applied in a real environment.

In terms of implementation and use of the work, we consider that it can be used for other related work that needs to implement the EPCIS layer, and also for companies that want to provide information to their business partners.

VII. FUTURE WORK

Although all the objectives we set have been completed, throughout the development some new ideas have arisen that can enrich the work.

To ensure that the computing resources of the system are managed in a more efficient way, we suggest using a thread pooling mechanism. Very basically, the mechanism of thread pooling is a process of creating and managing threads so whenever there is a new task, the uses an existing thread instead of creating a new one.

To simulate the whole process from reading an RFID tag to the sharing of information with other systems, it was necessary to use a Capture Application. This application is one of the layers of architecture recommended by EPCglobal and has the responsibility of converting an event in RFID EPCIS event. This conversion is performed based on some business rules, that we did not have for this work. So, we used this application with a simple conversion events, without adding business context. Usually these business rules are provided by other systems and we propose as future work to change the Capture Application for receiving information from other business systems or third parties and converts the RFID events in EPCIS events with more useful information.

We also suggest, as future work, the ONS layer from EPCglobal architecture. The ONS is a network service that allows you to search the addresses of EPC repositories through the EPC. Specifically, the ONS provides a means to search for an address of an EPCIS service provided by the organization that was in charge of the object with the EPC in question. In our solution we assume that all shared repositories addresses are known to all business partners.

The Discovery Service mechanism that finds all EPCIS repositories that may have information about a particular EPC, is referred in the recommendations of the EPCglobal but the specification was not developed. We recognize that this mechanism add value to EPCIS solutions, is particularly useful when you want to do a query to a EPCIS service but do not know the address.

.

REFERENCES

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

[2] S. Lewis, "A basic introduction to RFID technology and its use in the supply chain," *Laran RFID*, 2004.

[3] G. Capone, D. Costlow, W. Grenoble, and R. Novack, "The RFID-Enabled Warehouse," *SAP University Thought Leadership Supply Chain Paper, Penn State*, 2004.

[4] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID Application Development With the Accada Middleware Platform," *IEEE Systems Journal, VOL. 1, NO. 2, December 2007*.

[5] K. Leong, M. Ng, and D. Engels, "EPC Network Architecture," *White Paper Series / Edition 1, Auto-ID Labs*, 2005.

[6] "EPC Information Services (EPCIS) Version 1 . 0 . 1 Specification," *GS1 EPCglobal*, 2007.

[7] C. Du and S. Han, "Integrating EPCglobal Network with Web Services," *IEEE Computer Society*, 2009.

[8] K. Traub, F. Armenio, H. Barthel, P. Dietrich, J. Duker, C. Floerkemeier, J. Garrett, and M. Harrison, "The EPCglobal Architecture Framework," *GS1 EPCglobal*, 2010.

[9] C. Floerkemeier, M. Lampe, and C. Roduner, "Facilitating RFID Development with the Accada Prototyping Platform," *IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007.

[10] G. Pereira, "Dissertação de Mestrado: Assessment of an open-source, standards-based RFID supply chain integration," *DEI, IST*, 2009.

[11] W. Yanyan, Z. Xiaofeng, W. Yaohua, and X. Peipei, "The research of RFID middleware's data management model," *IEEE International Conference on Automation and Logistics*, Sep. 2008.

[12] T. Nguyen, Y. Lee, R. Huq, B. Jeong, and S. Lee, "A Data Model for EPC Information Services," *Department of Computer Engineering, KyungHee University, Korea*, 2007.

[13] M. Beckner, M. Simms, and R. Venkatesh, *Pro RFID in BizTalk Server 2009*, Apress, 2009.

[14] "Implementing EPC Information Services with Microsoft BizTalk Server 2006 R2," *http://msdn.microsoft.com/en-us/library/cc721650.aspx*.

[15] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*, Springer Verlag, 2004.

[16] J.D. Meier, C. Farre, J. Taylor, P. Bansode, S. Gregersen, M. Sundararajan, and R. Boucher, "Improving Web Services Security - Scenarios and Implementation Guidance for WCF," *patterns & practices by Microsoft*, 2008.

[17] A. Ilic, E.T.H. Zurich, T. Andersen, and F. Michahelles, "EPCIS-based Supply Chain Visualization Tool," *Auto-ID Labs White Paper*, 2009.