



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Assessment of an open-source, standards-based RFID supply chain integration

Guilherme Pereira

Dissertation for Master's Degree in
Engenharia Informática e de Computadores

Júri

Presidente:	Doutor José Carlos Martins Delgado
Orientador:	Doutor José Manuel da Costa Alves Marques
Co-Orientador:	Eng. Miguel Filipe Leitão Pardal
Vogais:	Doutor Rui Manuel Rodrigues Rocha

October 2009

Acknowledgements

I would like to express my deep gratitude to my thesis coordinator, Prof. José Manuel da Costa Alves Marques of the Department of Computer Science and Engineering from Instituto Superior Técnico in Lisbon, Portugal and Link Consulting for his advice, encouragement, enduring patience and critic spirit that lead me to the conclusion of this work. He was never ceasing in his belief in me (though I was often doubting my own abilities), and always providing clear explanations when I was lost.

Very special thanks to the Eng. Miguel Pardal who graciously volunteered to act as my co-advisor. He was extremely helpful in providing the additional guidance and expertise I needed in order to complete this work, especially with experience in the matter and constant support and encouraging. Without him this work could not be completed.

I also wish to thank to my co-coordinator, Eng. José Pires of the Link Consulting and Mário Romano also from Link Consulting. Their suggestions, comments and additional guidance were invaluable to the completion of this work.

Additionally, I want to thank the professors from the Department of Computer Science and Engineering and staff for all their hard work and dedication, providing me the means to complete my degree and prepare for a career as a computer engineer.

And finally, I want to thank all my family and friends for putting up with me during the development of this work with continuing, support and no complaint. In special I want to my brother, for being my cop and forcing me to keep working on this thesis, and my girlfriend for all her loving support, I do not have the words to express all my feelings here, only that I love you, Barbora!

Abstract

In today's competitive market one of the ways organizations can gain market position is by improving their supply chain efficiency.

With the emergence of a globalized economy, most companies deal with efficiency problems in their supply chains. Radio Frequency Identification (RFID) might be the answer to solve some of them. A set of standards are emerging from a standards organization named EPCglobal Inc. whose primary goal is to facilitate the adoption of RFID technology that improve the supply chain efficiency, but also focusing on business information sharing across organizations (Business-to-Business).

This thesis work evaluates the standards from EPCglobal, the inherent problems of adoption, focusing in the integration of the business logic with Fosstrak, an open-source implementation of the standards.

The most relevant contributions of this work are: a framework that allows an easier implementations on the Fosstrak platform, and a prototype implementation that solved some of the problems that a company was dealing with, in a given case study.

Keywords:

- RFID
- EPCglobal Architecture Framework
- Supply Chain Management
- Capture Applications
- Business Logic Integration

Resumo

As organizações pretendem melhorar a eficiência das suas cadeias de valor como forma de se prepararem para as exigentes condições de mercado. Hoje em dia a maioria das empresas tem problemas com a falta de eficiência da tecnologia usada nas suas cadeias de fornecimento, e neste campo a Identificação por Radio Frequência (RFID), promete resolver alguns destes problemas. Um conjunto de standards estão a aparecer por parte da EPCglobal Inc, cujo objectivo é facilitar a adopção da tecnologia RFID e melhorar a eficiência das cadeias de fornecimento, com especial foco na partilha de informação RFID com contexto de negocio, entre múltiplas empresas.

Este trabalho avalia este conjunto de standards, os problemas na adopção dos mesmos, em particular focando na integração da lógica de negócio com os sistemas RFID, usando uma implementação em concreto destes standards chamada Fosstrak.

As contribuições mais relevantes desta dissertação é uma framework que permite facilitar a implementação da plataforma Fosstrak, ao simplificar a integração da lógica de negócio e por outro lado o protótipo por si próprio tal como os resultados obtidos ao resolver os problemas de um caso de estudo dado.

Palavras Chave:

- RFID
- EPCglobal Architecture Framework
- Cadeias de Fornecimento
- Aplicações de Captura(Capture Applications)
- Integração da Lógica de Negócio

Contents

Acknowledgements	ii
Abstract	iv
Resumo	vi
List of Figures	xii
List of Tables	1
1 Introduction	3
1.1 Motivation	3
1.1.1 RFID	3
1.1.2 EPCglobal Inc.	4
1.1.3 Fosstrak	4
1.1.4 Supply Chains and EPCglobal Standards Integration	5
1.1.5 Capture Applications	6
1.2 Research Question	6
1.3 Outline	6
2 Background	9
2.1 RFID Technology	9
2.1.1 Tag	9
2.1.2 Interrogator	10
2.1.3 Controller	10
2.1.4 RFID Frequencies	10
2.2 EPCglobal Standards	11
2.2.1 Electronic Product Code	12
2.3 EPCglobal Architecture Framework	13
2.3.1 Reader and Tag roles and interfaces	14
2.3.2 Application Level Events (ALE)	15
2.3.3 Electronic Product Code Information Services (EPCIS)	15

2.3.4	Object Name System (ONS) Root	16
2.3.5	Tag Data Translation (TDT)	16
2.4	EPCIS Interface Details	17
2.4.1	EPCIS Capturing Application	17
2.4.2	EPCIS Query Application	17
2.4.3	EPCIS Repository	18
2.4.4	EPCIS Events	18
2.4.5	EPCIS Events, ALE Reports and Capture Applications	20
2.5	Fosstrak	20
2.5.1	LLRP Commander	22
2.5.2	Filtering & Collection Middleware with ALE and LLRP Support	22
2.5.3	Fosstrak EPCIS	23
2.5.4	Tag Data Translation (TDT)	23
2.6	Summary	24
3	Problems	27
3.1	Fosstrak Assessment	27
3.2	EPCglobal and Fosstrak Business Integration	27
3.2.1	The need for a capture application framework	29
3.2.2	The need for simpler concepts	30
3.3	Conclusion	32
4	Proposed Framework for Capture Applications	33
4.1	Motivation	33
4.2	Capture Applications	34
4.3	Capture Application Hierarchies	34
4.3.1	Reading Situations	34
4.3.2	Capture Application Hierarchies	36
4.4	Framework	38
4.4.1	Concepts	39
4.4.2	Services	40
4.4.3	Business	42
4.4.4	Usage	42
4.4.5	Implemented Functionality	44
4.5	Framework and Capture Application Hierarchies	44
5	Case Study and Prototype	47
5.1	Case Study	47
5.1.1	Business information	47
5.1.2	Business Processes As-Is (Current Situation)	48
5.1.3	Business Processes To-Be (Future Situation)	49
5.2	Prototype Decisions	51
5.2.1	Business Decisions	51
5.2.2	Technical Decisions	52

5.2.3	Assumptions	53
5.2.4	New Business Description	54
5.3	Prototype Technical Specification	56
5.3.1	Components	57
5.3.2	Automation System	57
5.3.3	Warehouse Expedition Gate System	58
5.3.4	Vehicle	58
5.3.5	Tracking	59
5.3.6	Deployment	59
5.4	Prototype Use	62
5.4.1	Example	62
5.4.2	Interaction with Framework	63
5.5	Conclusion	64
6	Evaluation	65
6.1	EPCglobal	65
6.1.1	Limited Range of Solution	65
6.1.2	Centralized Storage of Information	66
6.1.3	Reader Support	66
6.2	Fosstrak	67
6.2.1	Fosstrak Evolution	67
6.2.2	Fosstrak Documentation	67
6.2.3	Fosstrak Compatible Readers	67
6.2.4	Fosstrak integration with supply chains	68
6.3	Hierarchies of Capture Application	68
6.4	Proposed Framework	69
6.4.1	Comparison with Fosstrak	69
6.4.2	Advantages	70
6.4.3	Main Problems	71
6.5	Prototype	71
6.5.1	General Comments	71
6.5.2	Requirements Assessment	72
6.5.3	Problems	73
7	Conclusions	75
7.1	Conclusion	75
7.1.1	Contributions	77
7.2	Future Work	77
	Bibliography	80

List of Figures

1.1	RFID Tag and Reader - (Source http://www.logsystems.hu)	4
1.2	The EPC Architecture Framework - source http://www.epcglobalinc.org/standards	5
1.3	A Capture Application	6
2.1	RFID Frequencies - (Source http://www.apax.com.hk)	11
2.2	The vision of EPCglobal - Source: ChainLink Research (www.clresearch.com)	13
2.3	The 96 bit schema of EPC - Source http://journalism.berkeley.edu	13
2.4	The EPC Architecture Framework - Source: http://www.epcglobalinc.org/	14
2.5	The role of ALE in the EPCglobal Network (http://www.epcglobalinc.org/)	16
2.6	The differences between Master Data and Event Data [10]	20
2.7	The capture application in EPCglobal Architecture Framework (source: BEA Systems www.oracle.com)	20
2.8	The Fosstrak platform (Version prior to release of LLRP Project - June 2009)	21
2.9	The Fosstrak Filtering and Collection Architecture - Source Fosstrak.org	22
2.10	The Fosstrak EPCIS System Architecture - Source: Fosstrak.org	24
2.11	The Tag Data Translation Overview - Source: Fosstrak.org	25
3.1	The capture applications problem, initially considered out of scope by Fosstrak	28
3.2	Example of a capture application	28
3.3	The requirements for different example capture applications: vehicle and palletizer	29
3.4	A vehicle capture application example	29
3.5	The internal data structure of an EPCISEvent	30
4.1	The missing Business Layer in Fosstrak	33
4.2	Readers along the supply chain in static locations - Source: http://www.ubbtec.com/	35
4.3	Detail in a conveyor reader - Source: http://www.mmh.com	35
4.4	RFID Mobile Station - Source: http://www.mmh.com	36
4.5	A Palletizer - Source: http://www.intelligrated.com	37
4.6	Classes of capture applications	37
4.7	The Class Diagram of the Concepts package (A "Capture" in the diagram corresponds to an Event)	39
4.8	The Class Diagram of the Services package	41

4.9	The Class Diagram of the Business package	42
4.10	The required effort of implementing a Capture Application following the framework	45
5.1	The current business steps	50
5.2	The relation between basic items in a class diagram	55
5.3	The new business states	56
5.4	Screenshot of the Tracking Interface	59
5.5	The prototype deployment diagram	60
5.6	A potential real-world deployment diagram	61
5.7	The UML sequence diagram of an interaction	64

List of Tables

2.1	The current implemented standards in Fosstrak	21
4.1	The current framework implementation status for the EPCIS Events	41
6.1	Comparison between the use of Framework or Fosstrak natively	69

Chapter 1

Introduction

1.1 Motivation

The Internet of Things is a vision of having trillions of things communicating with one another without human intervention. It is also called The Product Internet or T2T (Thing to Thing). EPCglobal has defined standards that provide the first step towards this ambitious vision.

Today, RFID is starting to be used in enterprise supply chain management to improve the efficiency of inventory tracking and management. However, growth and adoption in the enterprise supply chain market is limited because current commercial technology does not link the indoor tracking to the overall end-to-end supply chain visibility. The key ingredients to achieve long-term and sustainable RFID technology adoption are linked with fair cost-sharing mechanisms, rational motivations, and justified returns from RFID technology investments [42].

Supply chain parameters may be improved by 90% and retail sales increased; there is talk of “cradle-to-grave” tracking, plus items electronically telling the freezer to restock; the microwave oven to cook correctly; and finally, the recycler to separate the materials correctly: all without human intervention [6].

1.1.1 RFID

RFID (Radio Frequency Identification) is a technology intended to complement or replace traditional barcode technology to identify, track, and trace items automatically.

RFID uses radio waves technology to extract data stored in tags. This data can have multiple contexts of use, but also represent a unique identifier for an item.

Figure 1.1 shows an RFID tag and a Reader.

RFID Systems

RFID Systems can store all the movements of items across a supply chain, and adding the proper business semantics it may potentially be used by other systems to create value for organizations.

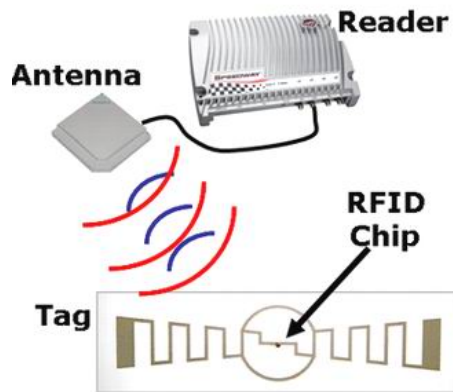


Figure 1.1: RFID Tag and Reader - (Source <http://www.logsystems.hu>)

RFID Systems claim to add intelligence and minimize human intervention in the item identification by using electronic tags. [24]

1.1.2 EPCglobal Inc.

EPCglobal is a joint venture between GS1 (formerly known as EAN International) and GS1 US (formerly the Uniform Code Council, Inc.). It is an organization set up to achieve world-wide adoption and standardization of Electronic Product Code (EPC) technology.

The main focus of the group currently is to create both a world-wide standard for RFID and the use of the Internet to share data via the EPCglobal Network.

EPCglobal Architecture Framework

The EPCglobal Architecture Framework also known as EPC Network is the proposed architecture to ease the development of RFID systems, by standardizing the technology. The EPC Network leverages the Internet to gain access to large amounts of associated information that can be shared between authorized users [47]. The main goals of the Architecture Framework are: [44]

- To simplify the exchange of information and physical objects between trading partners;
- To foster the existence of a competitive marketplace for system components and to encourage innovation.

In figure 1.2 the current version of the architecture framework is presented. The standards try to cover all the end-to-end solution scope of the RFID, from the data definition at the tag level to the sharing of the information among trading partners.

1.1.3 Fosstrak

According to Fosstrak's website: "Fosstrak (Free and Open Source Software for TRAcK and trace) is an open source RFID software platform that implements the EPC Network specifications. It is intended to support application developers and integrators by providing core software components

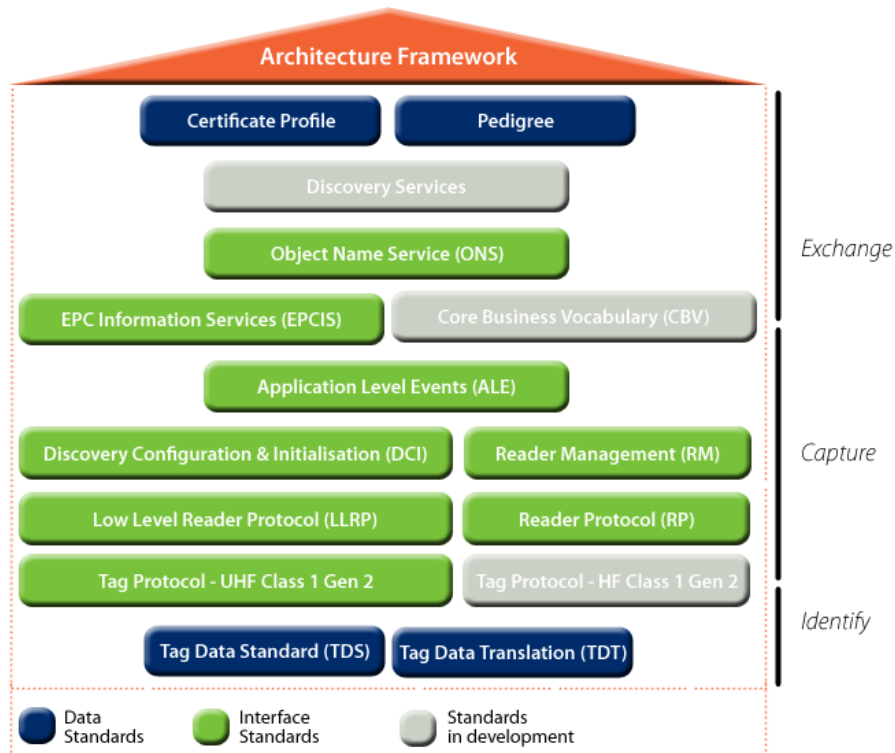


Figure 1.2: The EPC Architecture Framework - source <http://www.epcglobalinc.org/standards>

for track and trace applications”.

The Fosstrak implementation is based on the standardized interfaces published by the predominant RFID standardization body - EPCglobal Inc. The Fosstrak platform consists of four separate modules: the Fosstrak LLRP Commander, the Fosstrak Filtering & Collection Middleware with ALE and LLRP Support, the Fosstrak EPCIS Repository, and the Fosstrak Tag Data Translation (TDT) Library.

1.1.4 Supply Chains and EPCglobal Standards Integration

Supply Chain Management, spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption. RFID has potential in this area because reading individual items automatically may allow the full automation of data flows and processes, dispensing most human intervention.

EPCglobal standards with other enterprise systems may allow this automation to be a reality. The final goal of EPCglobal is enabling total visibility over supply chains, where any involved entity can access all the data about an individual item, since it is produced until the final consumer.

1.1.5 Capture Applications

Capture applications are one of the links of the business logic with the RFID Systems. At this level the RFID raw reads containing lists of EPC identifiers are given business context such as: business step, disposition, location, etc.; transforming them in meaningful information inside and across organizations.

These capture applications, one of the main focuses of this thesis, are needed in every implementation of the EPCglobal standards and still until recently (June 2009) not under Fosstrak's scope. Even today are not yet provided methods and concepts to facilitate the development of capture applications.

Figure 1.3 summarizes the role of the Capture Applications in the EPCglobal Network.

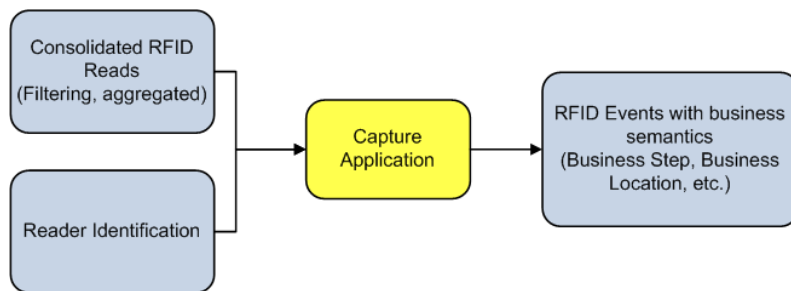


Figure 1.3: A Capture Application

1.2 Research Question

The main questions that this thesis attempts to answer are:

Is it possible to integrate and merge the Supply Chain Management of an organization using the Fosstrak implementation of the EPCglobal standards? What is needed to achieve this integration of RFID Systems with an organization's business logic? Can this integration be eased somehow?

1.3 Outline

This section presents the outline of the thesis work:

- 1 - Introduction** - The current chapter, it presents an introduction to the problem, the motivation and the question that the work attempts to answer.
- 2 - Background** - This chapter provides an introduction to the RFID technology, the EPCglobal Inc and its Architecture Framework and finally about Fosstrak.
- 3 - Problems** - Introduces the main problems that this thesis focuses. It mentions the problem that capture applications are not considered on scope by Fosstrak, and how it is important to have means to facilitate its development and implementation

- 4 - Proposed Framework for Capture Applications** - In this chapter it is proposed a framework for capture applications that introducing a set of concepts, methods and guidelines has the goal of simplify the task of development and integrating capture applications.
- 5 - Case Study and Prototype** - Presents a case study given by Link Consulting, with some requirements to be accomplished. It is pretended to assess Fosstrak platform and the developed framework.
- 6 - Evaluation** - Presents the results achieved by the thesis work, evaluating and criticizing them, as well as some comments to the EPCglobal solution and Fosstrak.
- 7 - Conclusion** - This chapter introduces some final comments, the contributions of this work and some future work ideas.

Chapter 2

Background

2.1 RFID Technology

RFID technology appeared in the beginning of the 20th century, with all the aspects of radio wave transmission being controlled [23] by advances in Physics. There are three basic elements in any RFID system [32]:

- A **tag** (also called a transponder), which compromises a semi-conductor chip, an antenna, and sometimes a battery;
- An **interrogator** (sometimes called a reader or a read/write device), which is comprised of an antenna, an RF (Radio Frequency) electronics module, and a control electronics module;
- A **controller** (sometimes called a host), which most often takes the form of a PC or a workstation running database and control (often called middleware) software.

Readers use radio wave technology to extract information from the tags, the antenna emits radio signals to activate and power the tag and then read and write data to it. When any RFID tag passes through the electromagnetic zone, it is powered by the reader's activation signal.

The tag then uses a technique called backscatter to reply to the interrogator. This does not involve a transmitter on the tag, but is a means of "reflecting" the carrier wave and putting a response signal into that reflection.

The reader finally decodes the response and the data is passed to the host computer for processing.

2.1.1 Tag

The basic role of an RFID tag is to store data and relay data to the interrogator. Tags can be classified based on the power source [32]:

- Active - RFID tags are said to be active if they contain an on-board power source, such as a battery. When the tag needs to relay data to the interrogator, it uses this source to draw the power for the transmission;

- Semi-passive or battery-assisted - These tags contain batteries that are temporarily activated to help powering the tag when data needs to be read;
- Passive - It contains no battery and uses the reader as the power source.

Another distinguishing factor between tags is their memory type. There are roughly two kinds: the read-only (RO) which contain a very limited amount of data, and read/write (RW) tags where data can be written and usually stores larger amounts of data.

2.1.2 Interrogator

The RFID Interrogator, usually called Reader, is the bridge between the tag itself and the controller. The basic roles are:

- Read and write data from and to the tag;
- Deliver the data to the controller;
- Power the tag in case it is passive.

There are more complex controllers that can provide some extra functionality like tag authentication using encryption mechanisms or collision detection to allow reads and writes from multiple interrogators.

Usually interrogators are placed in strategic places, and because they do not require any line of sight with the tag, there is considerable freedom on their positioning. Places like: dock doors, along conveyor belts, and in doorways; are suitable to track moving objects through the supply chain.

2.1.3 Controller

RFID controllers are the component entitled to the “management” of the RFID system. They are used to integrate multiple RFID interrogators together and to process the information. The controller is usually a server running a database or application software. For example, the controller could use information gathered in the field by the interrogators to [32]:

- Keep inventory and alert suppliers when new inventory is needed, such as in a retail application;
- Track the movement of objects throughout a supply chain, and possibly even redirect them, such as on a conveyor belt in a manufacturing application.

2.1.4 RFID Frequencies

RFID is based on wireless communication, using radio waves, using part of the electromagnetic spectrum (frequencies from 30kHz to 30GHz) to communicate, like Wi-Fi and Bluetooth. However, they are designed for different purposes and therefore have different functionalities. The biggest

Band	LF Low frequency	HF High frequency	UHF Ultra high frequency	Microwave
Frequency	30–300kHz	3–30MHz	300 MHz–3GHz	2–30 GHz
Typical RFID Frequencies	125–134 kHz	13.56 MHz	433 MHz or 865 – 956MHz 2.45 GHz	2.45 GHz
Approximate read range	less than 0.5 metre	Up to 1.5 metres	433 MHz = up to 100 metres 865-956 MHz = 0.5 to 5 metres	Up to 10m
Typical data transfer rate	less than 1 kilobit per second (kbit/s)	Approximately 25 kbit/s	433–956 = 30 kbit/s 2.45 =100 kbit/s	Up to 100 kbit/s
Characteristics	Short-range, low data transfer rate, penetrates water but not metal.	Higher ranges, reasonable data rate (similar to GSM phone), penetrates water but not metal.	Long ranges, high data transfer rate, concurrent read of <100 items, cannot penetrate water or metals	Long range, high data transfer rate, cannot penetrate water or metal
Typical use	Animal ID Car immobiliser	Smart Labels Contact-less travel cards Access & Security	Specialist animal tracking Logistics	Moving vehicle toll

Figure 2.1: RFID Frequencies - (Source <http://www.apax.com.hk>)

difference between RFID and Wi-Fi and Bluetooth is that the first is usually passive or semi-passive, whereas the latter ones are active.

RFID works in unlicensed spectrum space, referred to as ISM (Industrial, Scientific and Medical) but the exact frequencies that constitute ISM vary depending on the regulations in different countries. These operating frequencies are considered to be organized into four main frequency bands. The figure 2.1 shows these different radio wave bands and the more common frequencies used for RFID systems.

2.2 EPCglobal Standards

EPCglobal promotes “an RFID layer on top of the Internet” that will make it possible for computers all over the world to identify uniquely tagged objects instantly. In 2003, EPCglobal was created through a joint venture between the Uniform Code Council (UCC), makers of the Universal Product Code (UPC) barcodes, and European Article Number (EAN). It is a nonprofit organization

entrusted by the RFID industry to support and establish standards for the EPC Network. Their goal is to promote the adoption of the EPC Network standard (section 2.3). The Auto-ID Labs conduct scientific research in support of the EPC Network.

Providing open standards for tags, readers, and middleware, EPCglobal enabled the creation of a standards based industry, where tags applied in one country can pass through many different organizations to their final destination and the identity of the object is properly understood and authenticated [45].

The RFID tag does not need to provide or store any information itself. Instead, the unique identifier of each tag serves as a pointer to an associated database entry. In a highly networked environment, a backend system can essentially provide unlimited record-keeping for an RFID tag. The RFID tag can be seen as a physical URL of sort, the data for the tag is kept within the company and partner information systems.

In the near term, RFID will serve as a supply chain management tool. It will replace manual processes for tracking supplies in warehouses and other business locations. As exemplified on figure 2.2 as a crate passes by, an RFID enabled portal on a loading dock can relay information to the backend system. This simplifies automated creation of shipping manifests and other data, whose generation currently involves manual labor. In principle, data generation by RFID systems means that information about, let's say, a crate of apples, can reach the destination even before the apples are loaded onto the truck. In this near-term view, RFID is a form of automation support for the supply chain management systems of today.

RFID tags have been used for decades mostly for special purpose, including proprietary tracking. However, in modern commerce almost everything needs to move fluidly across enterprise boundaries.

2.2.1 Electronic Product Code

The Electronic Product Code is the thread that links all data that flows within the EPCglobal Network [44].

The Electronic Product Code (EPC) is the proposal from EPCglobal to an eventually substitute barcodes in a near future. The EPC major difference to the (Universal Product Code) UPC in USA or the equivalent (European Article Number) EAN in Europe is assigning **individual serial numbers** to every item.

The Electronic Product Codes are uniquely assigned to physical objects, loads, locations, assets, and other entities which are needed to be tracked through the EPCglobal Network improving visibility of items.

The most used version of the EPC Schema is the 96 bit encoding. In this encoding the EPC number is a composition of several parts as shown in figure 2.3. In short, the triplet of: manager number, product number, and serial number; **uniquely** identifies an object.

The standard of 96 bits allows up to 268 million possible combinations of unique manufacturers and 16 million possible combinations for unique products for each manufacturer. For each item instance 36 bits are used which provides more than 68 billion possible unique items by product.

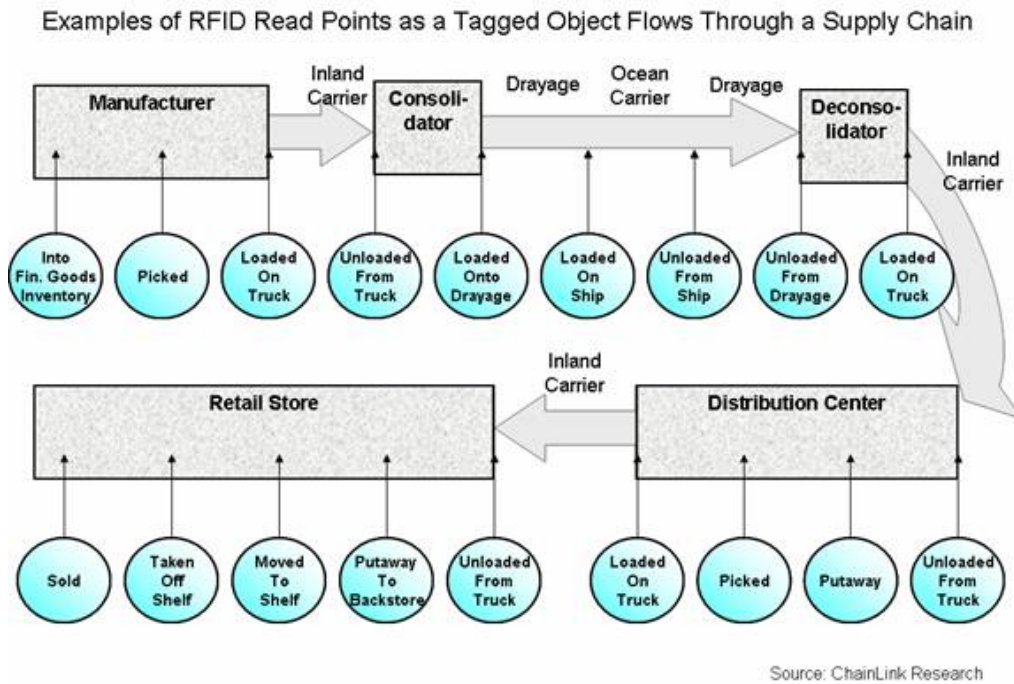


Figure 2.2: The vision of EPCglobal - Source: ChainLink Research (www.clresearch.com)

An EPC can also be represented as a URI (Uniform Resource Identifier) to enable the data exchange between software systems. This EPC URI is based on the EPC Tag Data Standards defined by EPCglobal [8].

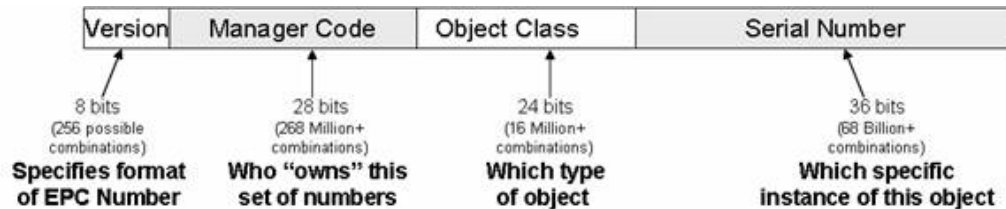


Figure 2.3: The 96 bit schema of EPC - Source <http://journalism.berkeley.edu>

2.3 EPCglobal Architecture Framework

The EPCglobal Architecture Framework also known as EPC Network is the proposed system to ease the development of RFID systems, by standardizing the technology. The EPC Network leverages the Internet to access large amounts of associated information that can be shared among authorized users [47]. The main goals of the Architecture Framework are the following: [44]

- To facilitate the exchange of information and physical objects between trading partners;
- To foster the existence of a competitive marketplace for system components and encourage innovation.

Figure 2.5 presents the current version of the EPCglobal Architecture Framework. The basic responsibilities of the roles and interfaces that compose the network are explained next. Some of these modules are still in the development stage and therefore they will not be mentioned.

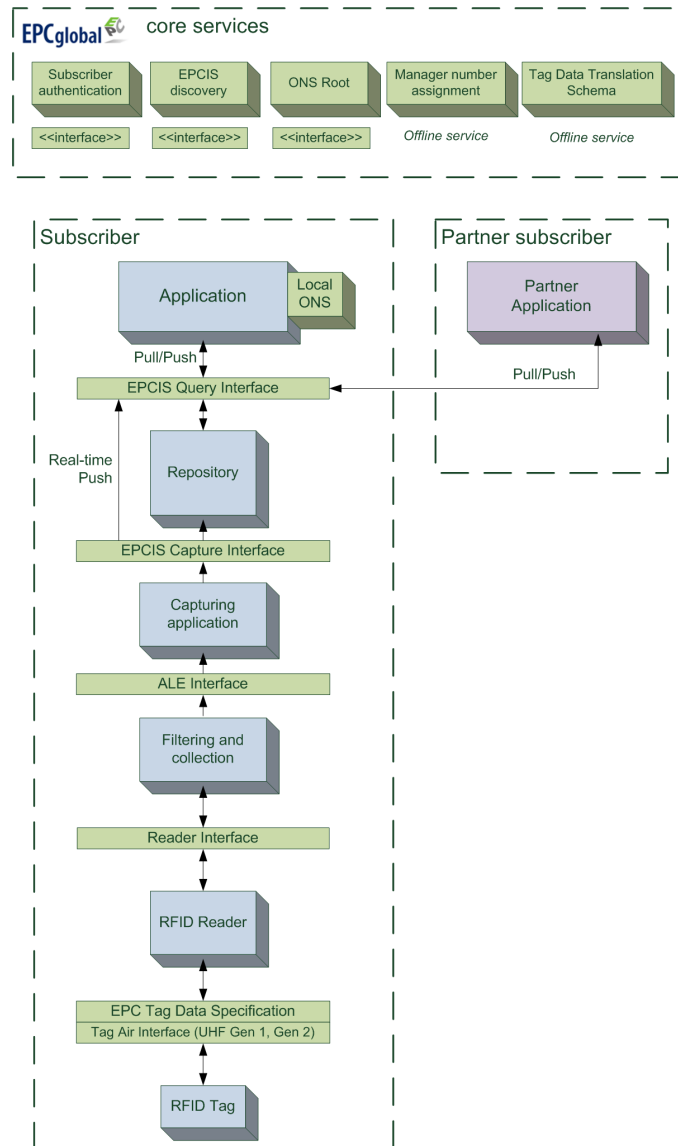


Figure 2.4: The EPC Architecture Framework - Source: <http://www.epcglobalinc.org/>

2.3.1 Reader and Tag roles and interfaces

These interfaces and roles are related to the reader functionalities such as management, monitoring and interaction between readers and tags using the over-the-air interfaces.

It is also defines the correct ways for reading, writing, and killing tags. It also establishes how the information should be physically stored on the tag.

- Reader Protocol (RP) 1.1 - This is the first Reader Interface standard developed by EPCglobal, and is now a ratified specification. It defines how the basic interaction (reads, writes and kills) is performed between applications and readers;
- Low-Level Reader Protocol (LLRP) 1.0 - This standard is the proposed evolution of the Reader Protocol. It provides more functionalities than the old version by enabling management and monitoring. The main advantage over the original Reader Protocol is the communication between two LLRP endpoints using a binary protocol, which is efficient and fast.
- Discovery Configuration & Initialization - It is a new standard recently released that compromises the auto discovery, configuration and auto initialization of readers.

2.3.2 Application Level Events (ALE)

The next grouping of interfaces and roles is the layer of the Application Level Events (ALE).

The role of the ALE interface within the EPCglobal Network Architecture is to provide independence between the infrastructure components that acquire the raw EPC data, the architectural components that filter and count that data, and the applications that use the data. This allows changes in one without requiring changes in the other, offering significant benefits to both the technology provider and the end user. The ALE interface described in the latest specification from EPCglobal achieves this independence through five means:

- It provides a means for clients to specify, in a high-level, declarative way, what data they are interested in or what operations they want performed;
- It provides a standardized format for reporting consolidated and filtered data resulting from operations;
- It abstracts the channels through which data carriers are accessed into a higher-level notion of “logical reader”, usually related to the “location”, hiding from clients the details of exactly what physical devices were used to interact with data;
- It abstracts the addressing of information stored on tags and other data carriers into a higher-level notion of named, typed “fields”. This allows hiding from clients the details of how a particular data element is encoded into a bit-level representation and stored at a particular address within a tag;
- It provides a security mechanism so administrators may choose which operations a given application may perform, as a policy that is decoupled from the application logic itself.

2.3.3 Electronic Product Code Information Services (EPCIS)

The last group, the Electronic Product Code Information Services (EPCIS) is an EPCglobal standard for sharing EPC related information between trading partners. EPCIS provides important

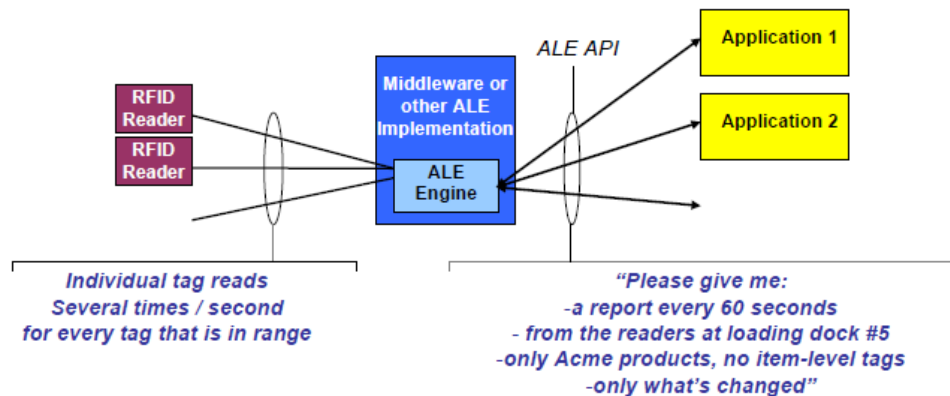


Figure 2.5: The role of ALE in the EPCglobal Network (<http://www.epcglobalinc.org/>)

new capabilities to improve efficiency, security, and visibility in the global supply chain, and complements lower level EPCglobal tag, reader, and middleware standards.

The EPCIS specification defines a data language for representing visibility information, namely events having four dimensions of “what”, “when”, “where”, and “why” [45].

2.3.4 Object Name System (ONS) Root

The Object Name System (ONS) is the global directory of EPC Information Services that are publicly available to query for product information.

This name system was developed having the following responsibilities in mind:

- Provide the initial point of contact for ONS lookups;
- In most cases, delegate the remainder of the lookup operation to a Local ONS operated by the EPC Manager for the requested EPC;
- Completely fulfill ONS requests in cases where there is no local ONS to delegate a lookup operation;
- Provide a lookup service for 64-bit Manager Index values as required by the EPC Tag Data Specification.

2.3.5 Tag Data Translation (TDT)

As seen before in section 2.2.1 an EPC identifier may be expressed in a number of representations or encodings, such as binary, tag-encoding URI, pure-identity URI, and legacy formats.

The objective of the Tag Data Translation (TDT) project is to provide flexible translation (encoding/decoding) between these different representations of an EPC. For example, TDT allows a binary string to be converted into a URI representation or vice versa. The coding schemes specified in EPCglobal Tag Data Standards v1.27 are supported in v1.0 of Tag Data Translation.

2.4 EPCIS Interface Details

Due to the importance of the EPCIS for this work it will be given extra attention in this section.

EPCIS is the standard at the top layer of the Architecture Framework, it differs from elements at the lower layers of the EPCglobal Architecture Framework in three key aspects:

- EPCIS deals explicitly with historical data. The lower layers of the stack deal exclusively with real-time processing of EPC data;
- EPCIS often deals not just with raw EPC data, but with observations that include meaning about the physical world and to specific business steps and business processes. The lower layers of the stack are purely observational in nature;
- EPCIS typically operates and exists in a more diverse IT environment than the lower levels of the EPCglobal Architecture because of a combination of factors including: the desire to share EPCIS data between organizations that are likely to have different solutions, the persistent nature of EPCIS data, and EPCIS being the natural point of entry into other enterprise systems.

The main roles and interfaces in the standard are described next.

2.4.1 EPCIS Capturing Application

The EPCIS Capturing Application exposes the capture interface. In its simplest form it is a simple “identity” that forwards the Application Level Events to stable storage, usually the EPCIS Repository. Middleware such as ALE push information to the EPCIS Capture service which is then responsible for pushing it to the EPCIS repository. The EPCIS Capture service may or may not acknowledge the reception of data and, in some cases, it may perform data transformation according to defined business rules, taking previous RFID events into consideration, when needed.

The main responsibilities are:

- Recognize the occurrence of EPC ALE events, and delivers these as EPCIS data;
- May coordinate multiple sources of data in the course of recognizing an individual EPCIS event;
- May control the carrying out of actions in the physical environment, including writing RFID tags and controlling other devices;
- Provide a path for communicating EPCIS events generated by EPCIS Capturing Applications to other components and systems.

2.4.2 EPCIS Query Application

EPCIS Query Client acts as the client to EPCIS Services and issues queries about EPC information. EPCIS Accessing Applications can operate internally within the organization, close to the EPCIS Server or can be completely external, accessing the EPCIS server remotely.

The main responsibilities are:

- Provide a means whereby an EPCIS Accessing Application can request EPCIS data from an EPCIS Repository or an EPCIS Capturing Application, and the means by which the result is returned;
- Provide a means for mutual authentication of the parties involved;
- Reflect the result of authorization decisions taken by the providing party, which may include denying a request made by the requesting party, or limiting the scope of data that is delivered in response.

2.4.3 EPCIS Repository

The EPCIS Repository is the place where the EPC Events are stored in a persistent way. The repository may use a database to store information. The stored information has EPCIS Events, which have a business meaning and context. The EPCIS Repository exposes the capture and query interfaces of the EPCIS.

The main responsibility is to record EPCIS-level events generated by one or more EPCIS Capturing Applications, and making them available for later querying by Accessing Applications.

2.4.4 EPCIS Events

The EPCIS standard data model define how the RFID data with business context must be stored, making it useful information for organizations. The EPCIS Events define WHAT (product), WHERE (location), WHEN (time), and WHY (business step and status) for granular product movements in the supply chain.

According to the situation, an EPCIS Event can consolidated in one of four defined ways:

Object Event An ObjectEvent represents a simple event that happened to some EPC tagged entities. It encapsulates information about the event's time, location, business step, and others;

Aggregation Event The AggregationEvent encapsulates information related to the physical aggregation of entities tagged by EPC codes. For example an aggregation event could be "At time t, these items were added to container c.";

Quantity Event The QuantityEvent describes an action for a quantity of entities belonging to a specific EPC class, but does not care about the EPCs of the entities themselves;

Transaction Event A TransactionEvent is very similar to an AggregationEvent by allowing aggregation of multiple items. The main difference is having one or more transaction id's, which allows a direct association with specific business transactions.

The following list details how the information is organized inside the EPCIS Events, what is the meaning of each name, as well as their definition according to the standard name under parenthesis.

- What

- EPC (epcList / childEPCList / parentEPCID) - can be a list (Object or Transaction Events) or parent/child (Aggregation or Transaction Events). It is possible to include any unique identity in the EPC field;
 - Business Transaction (bizTransaction) - includes a type (e.g.: Purchase Order, Invoice, etc.) and a number. By including the Business Transaction number in a business event, it is possible to relate EPCs to a Business Transaction.
- Where
 - Read Point (readPoint)- indicates the location where an event took place;
 - Business Location (bizLocation)- describes where the object is immediately after the event occurs.
- When
 - Event Time (eventTime) - states when an event took place;
 - Record Time (recordTime) - indicates when the event was received through the EPCIS Capture Interface. The event and record time can be different due to processing delay and clock synchronization.
- Why
 - Business Step (bizStep) - indicates what business operation was taking place at the time of the event - e.g.: Receiving, Picking, Loading, Shipping;
 - Disposition (disposition) - describes the status of the product immediately after the event occurs - e.g.: Saleable, In Progress, Non Saleable, Destroyed.

Event Data and Master Data

The EPCIS Events according to the standards deals with two kinds of data: event data and master data. Event data arises in the course of carrying out business processes, and is captured through the EPCIS Capture Interface and available for query through the EPCIS Query Interfaces. Master data is additional data that provides the necessary context for interpreting the event data (for example the product type). It is available for query through the EPCIS Query Control Interface, but the means by which master data enters the system is not specified in the EPCIS 1.0 specification [10].

Still regarding the event data and master data, the EPCIS Standard does not define any standard vocabulary for the exchange of information. Instead it leaves the definition to trading partners. This may lead to problems if one company for example considers a shipment as the following bizStep: "urn:epcglobal:epcis:bizstep:fmcg:shipped" and the second one represents as "urn:epcglobal:epcis:bizstep:fmcg:shipment". This may pose problems later when sharing the data.

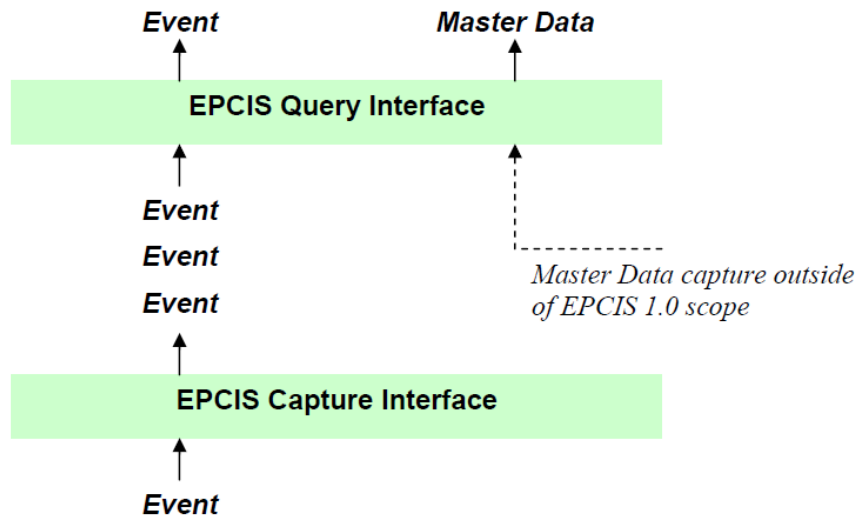


Figure 2.6: The differences between Master Data and Event Data [10]

2.4.5 EPCIS Events, ALE Reports and Capture Applications

An ALE Report is the result of the Filtering and Collection middleware and it contains the result of the consolidation of the raw reads without business context. The capture application is entitled to transform this information into ALE Reports into EPCIS Events adding the business context.

Figure 2.7 presents an example of the capture application and its role in the EPC Network.

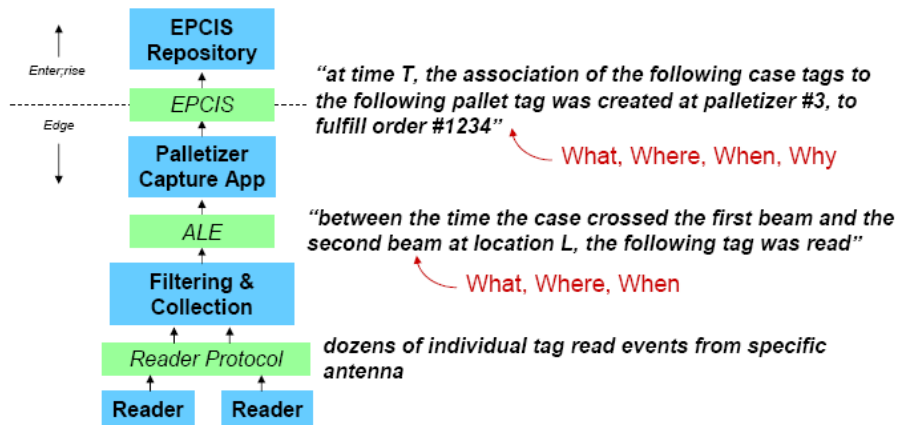


Figure 2.7: The capture application in EPCglobal Architecture Framework (source: BEA Systems www.oracle.com)

2.5 Fosstrak

"Fosstrak is an open source RFID prototyping platform that implements the EPC Network specifications. It is intended to foster the rapid prototyping of RFID applications" It was initiated

by Christian Floerkemeier, Matthias Lampe, and Christof Roduner of the Distributed Systems Group at ETH Zurich and the Auto-ID Lab at ETH Zurich/University of St. Gallen [33].

The Fosstrak platform consists of four separate modules also considered projects:

- Fosstrak LLRP Commander;
- Fosstrak Tag Data Translation (TDT) Library;
- Fosstrak Filtering & Collection Middleware with ALE and LLRP Support;
- Fosstrak EPCIS Repository;

Table 2.1 presents the standards and interfaces of the EPCglobal Architecture Framework currently implemented by Fosstrak.

EPC Standard	Implemented	Project
Tag Data Standard	No (out of scope)	-
Tag Protocol	No (out of scope)	-
Tag Data Translation (TDT)	Yes	Tag Data Translation (TDT) Library
RFID Interface (RP)	Deprecated	Reader (Deprecated Project)
RFID Interface (LLRP)	Yes	Fosstrak LLRP Commander
Reader Management Interface	Deprecated	Reader (Deprecated Project)
Filtering and Collection (ALE 1.1)	Yes	Filtering & Collection Middleware
EPCIS Capture Interface	Yes	EPCIS Repository
EPCIS Query Interface	Yes	EPCIS Repository
EPCIS Repository	Yes	EPCIS Repository
Object Name Service (ONS)	No (out of scope)	-

Table 2.1: The current implemented standards in Fosstrak

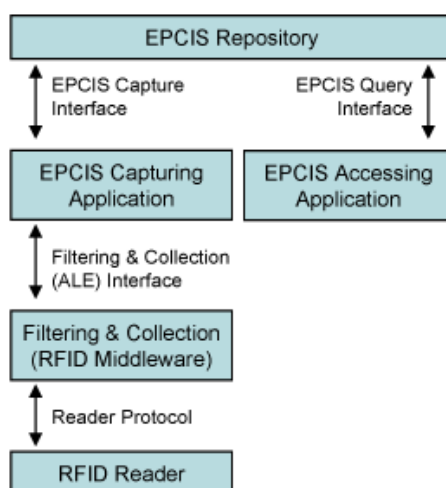


Figure 2.8: The Fosstrak platform (Version prior to release of LLRP Project - June 2009)

The following sections further detail the development of each of the Fosstrak projects. Note that this section despite addresses the structure of the last version of Fosstrak (June 2009), the prototype development was performed using a version of March 2008.

2.5.1 LLRP Commander

LLRP stands for "low-level reader protocol" and specifies a protocol for the control of RFID readers (see section 2.3.1).

The main functionalities provided by this module are:

- Interact with a reader that implements the EPCglobal Low-Level Reader Protocol Version, by allowing not only the reads and writes to tags but also all configuration of all the related parameters;
- Communicate with a reader via the EPCglobal LLRP from within a Java application using the Fosstrak Reader RP Proxy.

The Fosstrak LLRP Commander thus has the potential to become a universal framework for reader clients, eliminating the need to have a different reader client for each reader type.

2.5.2 Filtering & Collection Middleware with ALE and LLRP Support

The objective of the Filtering and Collection project is to implement the corresponding role in the EPC Network and to develop the appropriate tools that facilitate communication with filtering and collection instances.

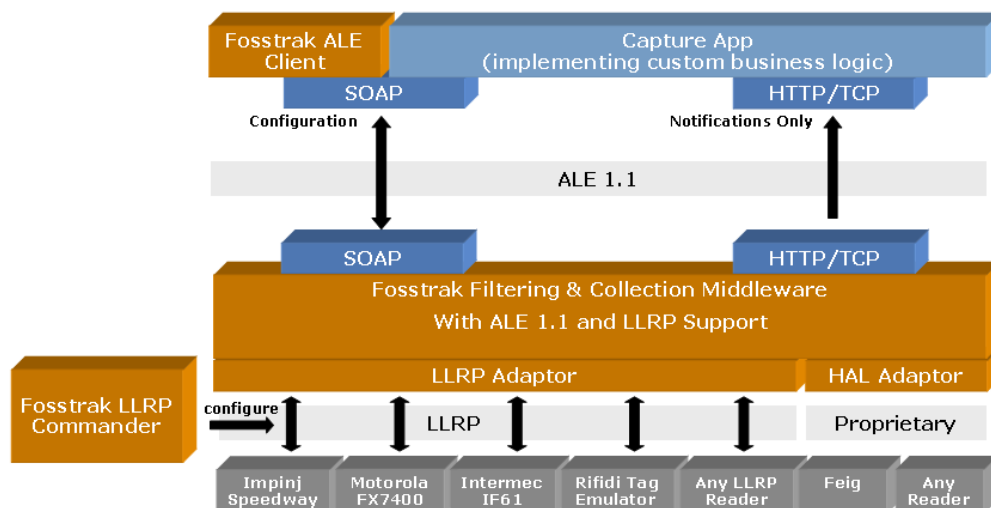


Figure 2.9: The Fosstrak Filtering and Collection Architecture - Source Fosstrak.org

This project allows the following activities:

- Communicate with RFID readers (the Fosstrak ALE Middleware uses LLRP);
- Filter and aggregate and consolidate raw RFID data from multiple physical readers into one logical reader;

- Deploy an RFID middleware that implements the Filtering and Collection role and supports the LLRP and ALE specification.

In Fosstrak the Filtering and Collection Project comprises three separate modules:

- The Filtering and Collection implementation itself;
- A standalone client to configure filtering and collection servers;
- A web client to configure filtering and collection servers.

All three implement the ALE Specification defined by EPCglobal. The middleware itself communicates with the readers via the LLRP standard. The Fosstrak filtering and collection middleware represents a single interface to the potentially large number of readers that composes a RFID system deployment. This allows applications to define a subscription, which is then used to configure the corresponding reader devices using LLRP. Once the readers capture relevant tag data, the readers notify the middleware, which combines the data arriving from different readers in a report that is sent according to a predetermined schedule to the subscribing applications.

Since the middleware receives data from multiple readers, it provides additional aggregation functionality. Redundant read events from different readers observing the same location can be omitted. It can also span multiple readers to avoid failures in the reading process due to the object's nature.

2.5.3 Fosstrak EPCIS

The modules implement the EPC Information Services (EPCIS) specification defined by EPCglobal and its members. The major responsibilities of EPCIS were stated in the section 2.4.

The Fosstrak EPCIS implementation comprises three separate modules:

- An EPCIS Repository implementation;
- An interactive EPCIS Capture Client;
- An interactive EPCIS Query Client.

The figure 2.5.3 gives an overview of the Fosstrak EPCIS Implementation.

The Fosstrak EPCIS implementation is a certified and complete implementation of the latest EPCIS standard specification (Version 1.0.1 of September 21, 2007).

2.5.4 Tag Data Translation (TDT)

As seen before in section 2.3.5 provide flexible translation (encoding/decoding) between these different representations of an EPC. Figure 2.5.4 presents an overview of the Fosstrak implementation of the TDT standard.

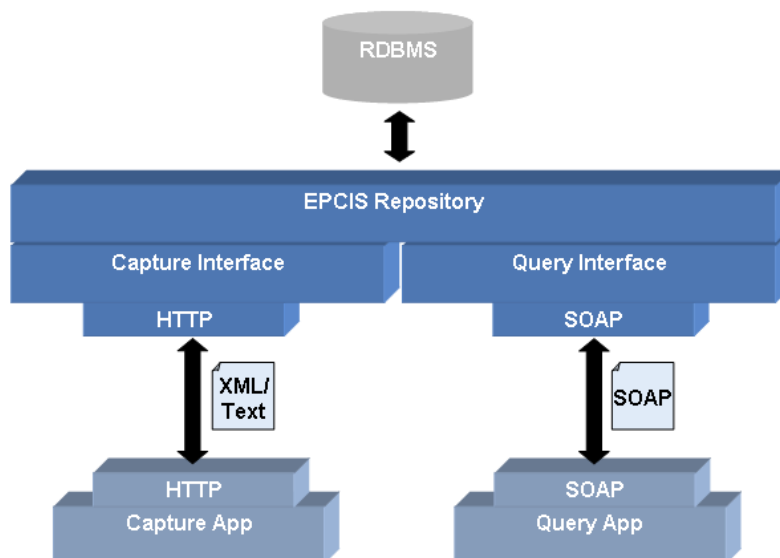


Figure 2.10: The Fosstrak EPCIS System Architecture - Source: Fosstrak.org

2.6 Summary

This chapter provided an introduction to the key terms and concepts required to follow the next chapters.

It was firstly introduced the RFID technology. Then it was given an overview on the EPCglobal Inc, and the EPCglobal Architecture Framework, where the main modules were introduced and described in a high level overview.

Then the focus was moved into the EPC Information Services (EPCIS), with special highlights on the **capture applications**. Capture Applications are responsible for processing simple RFID reads into EPCIS Events with meaningful business context. This thesis will focus mostly on this component.

Finally it was presented Fosstrak, the way its development is being processed, with a brief description and overview for every module and their relationship with the EPCglobal Architecture Framework.

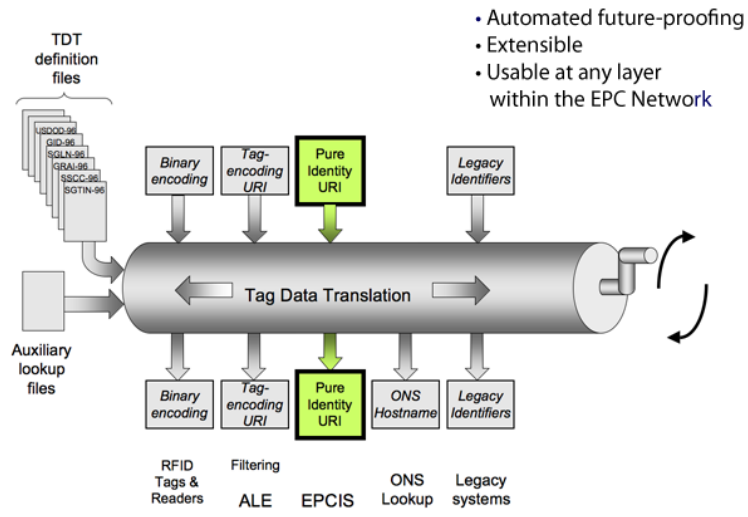


Figure 2.11: The Tag Data Translation Overview - Source: Fosstrak.org

Chapter 3

Problems

The acceptance of RFID realizes the need for a good “integration fabric” that allows data to seamlessly flow from the devices (tags) through the readers to the RFID middleware systems, and be utilized by the existing or new applications to trigger meaningful transactions [37].

The main questions this work aims to answer are: Is it possible to integrate and merge the Supply Chain Management of an organization using the Fosstrak implementation of the EPCglobal standards; What is needed to achieve this integration of RFID Systems with an organization’s business logic; And finally if this integration be eased somehow.

3.1 Fosstrak Assessment

As stated above one of the initial goals is perform an assessment of the Fosstrak platform, focusing in the integration of the RFID technology with the organization’s business details and particularities.

Questions such as what is the maturity state of the EPCglobal standards, the current stage of development of the platform, the scope of the covered platform, the expected problems during an implementation phase, or even what is the required effort and steps to implement a prototype solution should be answered.

3.2 EPCglobal and Fosstrak Business Integration

According to the standards and as explained in section 2.4.5 the main integration point of any EPCglobal based solution with the business processes are the **capture applications**.

Capture applications receive as input **ALE Reports** and produce as output **EPCIS Events** (see figure 3.1).

ALE Reports are transformed in EPCIS Events with some effort. The main role of a capture application is to incorporate the situation and context in which events are detected and to give

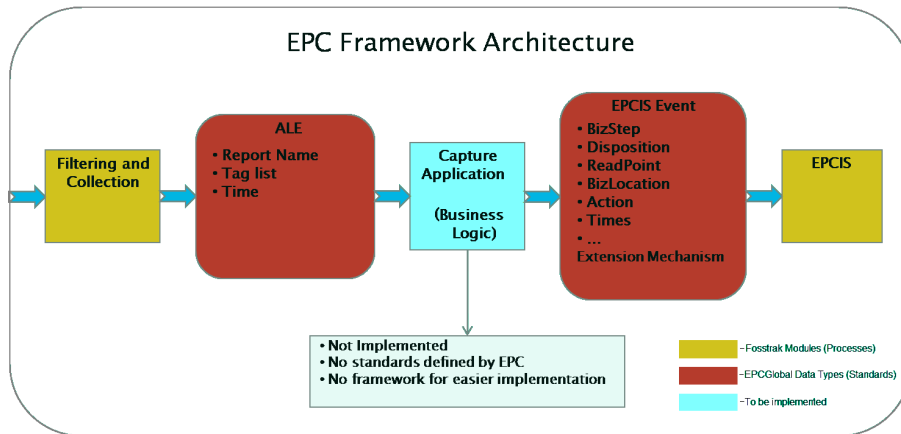


Figure 3.1: The capture applications problem, initially considered out of scope by Fosstrak

them the additional business semantics.

Figure 3.2 shows the importance of capture applications on the EPC Network. The example palletizer capture application added the business semantics and established the association with an order number by interpreting the read situation and context.

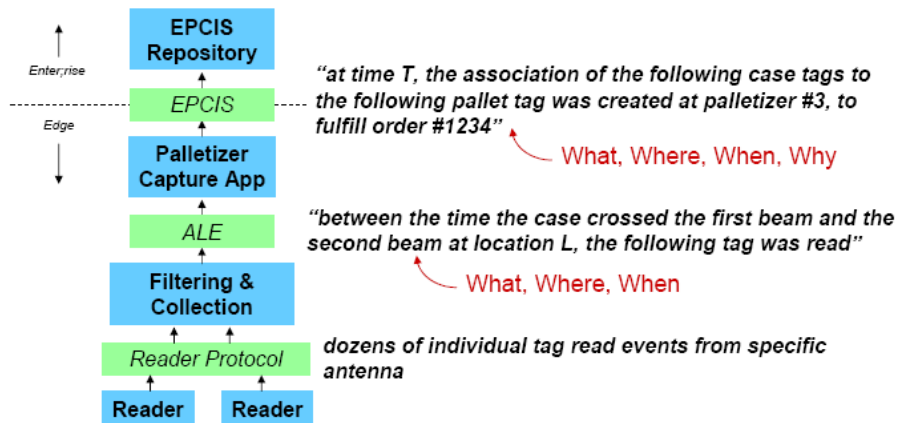


Figure 3.2: Example of a capture application

Fosstrak initially (during the prototype development) considered capture applications out of its scope, therefore did not provide this module, or any means or concepts to facilitate or speed up the integration of its platform with the organization's business processes.

In the latest release of the platform some finally work was done on this way, and there is provided a simple version of a Capture Application that is able to transform ALE Reports into EPCIS Events, by the use of "ECReportsHandler". Although still not enough concepts and methods are provided, and these handlers still must deal with concepts such as EPCIS, which are far from the real world layer, where the traditional SCM systems operate.

The next two sections describe in more detail two of the major problems that developers face

when employing a Fosstrak solution.

3.2.1 The need for a capture application framework

The main differences from one capture application to another are the business rules and particularities of the situations and contexts that must be interpreted to correctly process an ALE Report into an EPCIS Event. Figure 3.3 shows an example of a vehicle's capture application that may require manual input, potential GPS (Global Positioning System) integration and an alert mechanism. The palletizer example requires integration with an ERP and a picking machine.

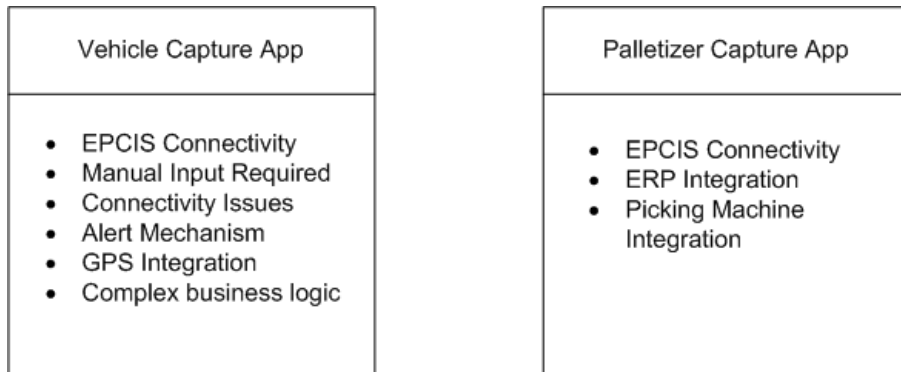


Figure 3.3: The requirements for different example capture applications: vehicle and palletizer

The usual task of interpreting the read situation and context can be illustrated by the figure 3.4. The vehicle's capture application receives the item read with the EPC X123 (1) and needs to acknowledge the business situation and context. To do so, it inquires the EPCIS Repository (2), and knowing the event is a "shipment" (3), concludes based on the implemented business rules that the current event must be a "delivery" (4). The last step is capturing this information into the EPCIS Repository (5).

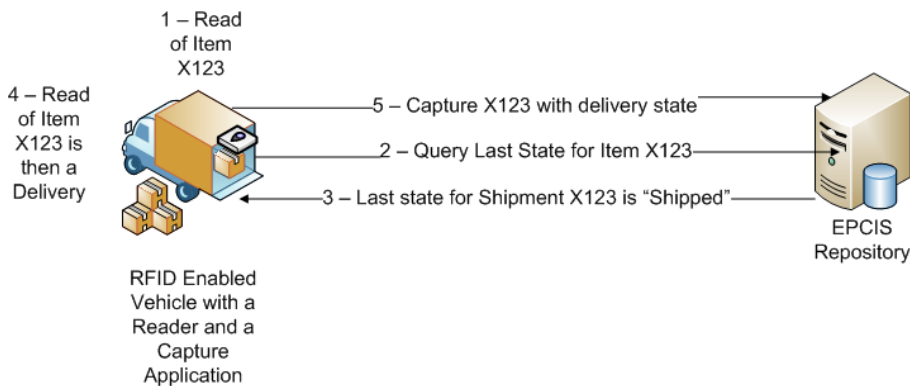


Figure 3.4: A vehicle capture application example

From an organization's perspective, the perfect Capture Application would be one where analysts would specify the business rules and the required interactions in a standardized format, relating: readers, rules, locations, and systems. By loading these rules on the capture application, it would be able to interpret and perform the business logic automatically.

Unfortunately for companies and developers, neither the EPCglobal standards define such a format, nor is Fosstrak leading this way. Being required to deploy different capture applications inside an organization, a developer is, for the moment, encouraged to create mechanism to simplify his task and reduce the development effort.

To illustrate the need for a framework, we show code provided by Fosstrak of how to start the development capture application, at the time of development of this thesis. The listing 3.1 presents an example how to parse an ALE event and create a simple EPCIS Event (an ObjectEvent in this case) with “hardcoded” business rules [39].

As seen from the code provided by Fosstrak, transforming the ALE Reports into EPCIS Events is not easy, but the complexity is increased by the process of adding the logic to interpret the read situation and context.

3.2.2 The need for simpler concepts

Supply Chain Management (SCM) applications usually apply and use concepts at the “real world level” such as: items, packages, localizations; to perform and apply the business logic. Fosstrak provides only concepts from another abstraction level like EPCIS Events or ALE Reports (see details in section 2.3).

One of the first noticeable things in Fosstrak was the missing concept of “item” or “package” that could be used in the capture applications to simplify the application of the business logic. This level of concepts directly related with the capture applications is not yet provided by Fosstrak.

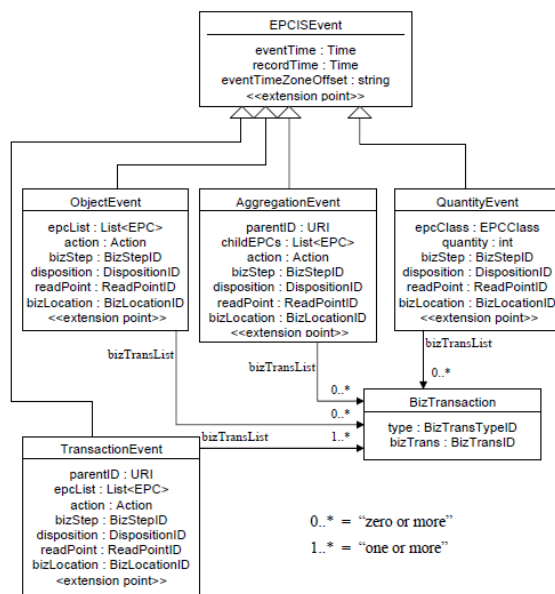


Figure 3.5: The internal data structure of an EPCISEvent

As shown in figure 3.5, applications that deal directly with EPCIS Events need to know about

Listing 3.1: The proposed example by Fosstrak to a capture application for capturing an Object Event

```

List<ECReport> theReports = reports.getReports().getReport();  //(Read
    reports from ALE)
List<EPC> epcs = new LinkedList<EPC>();
if (theReports != null) {
    for (ECReport report : theReports) {
        (...)  //Report Reading into "epcs" list
    }
}

(...)  //Handling of no reports read

ObjectEventType objEvent = new ObjectEventType();  // create the epcis event

(...)  //Setting and prepare Callendar and timezone details

EPCListType epcList = new EPCListType();
 // add the epcs from the Report to the EPCIS Event
for (EPC epc : epcs) {
    org.Fosstrak.epcis.model.EPC nepc=new org.Fosstrak.epcis.model.EPC();
    nepc.setValue(epc.getValue());
    epcList.getEpc().add(nepc);
}
objEvent.setEpcList(epcList);

 // Hardcoded business logic
objEvent.setAction(ActionType.ADD);  // set action
objEvent.setBizStep("urn:Fosstrak:demo:bizstep:testing");  // set bizStep
objEvent.setDisposition("urn:Fosstrak:demo:disp:testing");  //disposition
ReadPointType readPoint = new ReadPointType();  // set readPoint
readPoint.setId("urn:Fosstrak:demo:rp:1.1");
objEvent.setReadPoint(readPoint);
BusinessLocationType bizLocation = new BusinessLocationType();  //bizLocation
bizLocation.setId("urn:Fosstrak:demo:loc:1.1");
objEvent.setBizLocation(bizLocation);

 //create the EPCISDocument
EPCISDocumentType epcisDoc = new EPCISDocumentType();
EPCISBodyType epcisBody = new EPCISBodyType();
EventListType eventList = new EventListType();
 //Adding the Object Event
eventList.getObjectEventOrAggregationEventOrQuantityEvent().add(objEvent);
epcisBody.setEventList(eventList);
epcisDoc.setEPCISBody(epcisBody);
epcisDoc.setSchemaVersion(new BigDecimal("1.0"));
epcisDoc.setCreationDate(now);

 //Capturing the EPCIS Event
int httpResponseCode = client.capture(epcisDoc);
if (httpResponseCode != 200) {
    System.out.println("The_event_could_NOT_be_captured!");
}

```

four different kinds of event, and be prepared handle possible extensions. Also the relevant data is encapsulated under several levels, which makes it hard to access and extract.

This may happen because EPCIS Event have been designed taking more in consideration the storing, the facility to share of information across multiple partners or the flexibility to support many scenarios rather than the easy and fast interpretation of the stored information. The issue regarding the EPCIS Master Data (see section 2.4.4) that despite being covered by the EPCIS standards, the standard does not define yet means to capture this information into the EPCIS Repository, forcing solutions to store Master Data in other locations than the EPCIS Repository may also confirm this problem.

3.3 Conclusion

The effort of deploying capture applications inside an organization makes critical the use of a framework to accelerate and simplify this task.

The provided abstraction level coming by the available Fosstrak structures is far away from the usually used by SCM applications what makes harder applying business logic. Also the methods provided by Fosstrak do not provide the right ways that allow simplifying the development of capture application. A framework may be a solution to ease this task.

Proposed Framework for Capture Applications

This chapter presents implementation guidelines and the proposed framework to simplify the development and implementation of capture applications for supply chain management.

4.1 Motivation

When first exploring Fosstrak and connecting the provided modules, it was noticeable that transforming ALE Events into EPCIS Events was difficult, as no capture application was provided. Figure 4.1 summarizes the problem (see section 3.2).

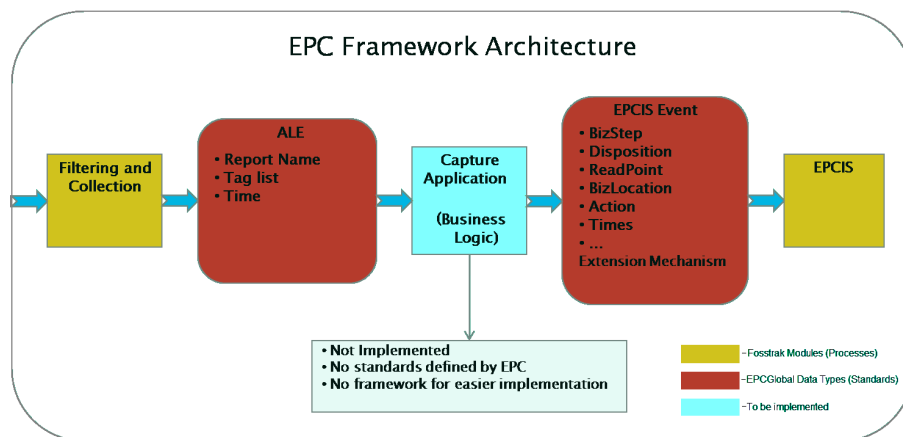


Figure 4.1: The missing Business Layer in Fosstrak

It was then required to develop a set of 'wrappers' during the prototype development. Later it evolved to a more structured definition that evolved into a framework, presented in the next sections.

4.2 Capture Applications

Starting from figure 4.1 and using a “black box” analogy for capture applications with well-defined input and output, we can distinguish two types of capture applications:

Stateless Capture Application The interpretation of capture context and its configuration parameters are static and independent from external systems. These static definitions are enough for transforming ALE Reports into EPCIS Events.

Stateful Capture Application It requires information about the previous capture(s) and/or orchestration with another system(s) to properly interpret the capture context and transform ALE Reports into EPCIS Events.

The required type of capture application depends on the nature of the business steps that the associated readers must process. Complex uses of readers will mostly likely require stateful capture applications as already shown on figure 3.4.

This dissertation focuses mostly on Stateful Capture Applications, since they are more useful in real world cases than the Stateless ones, and allow taking out more benefits from RFID technology.

4.3 Capture Application Hierarchies

Capture applications transform ALE Reports into EPCIS Events incorporating and interpreting the read situation and context to give the business semantics to the EPCIS Events.

It is possible to extract common characteristics and interactions of the capture application placed in the strategic points along supply chain and stereotype them.

The capture application class hierarchy provides: a high-level understanding of the different possibilities for capturing RFID data, generic implementations that handle the most common problems, and implementation guidelines.

4.3.1 Reading Situations

The next sections present the most common places to track and trace RFID items in a logistics supply chain, grouping them according to their characteristics.

Conveyors, Gates and Portal Tracking

The most generic use of the RFID reader along a supply chain is in gates, portals and conveyors. Figure 4.2 exemplifies the situation.

These readers are usually fixed and deal with items and package along the supply chain with the main responsibility of assuring their correct pathway. There are usually one or more associated readers, each of them with one or more antennas, and the read cycle is started when the product is first read and terminates implicitly when the item gets out of range from an antenna.

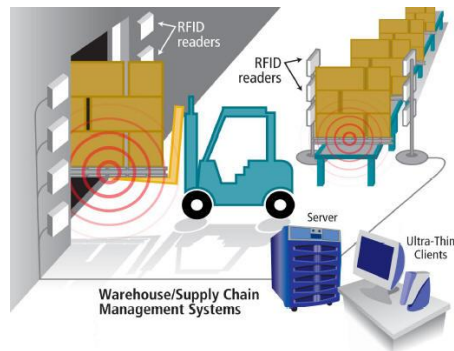


Figure 4.2: Readers along the supply chain in static locations - Source: <http://www.ubbtec.com/>

Usually these fixed readers require only the EPCIS Repository as a source of information to correctly apply the business logic. Also as they usually do not have user interfaces an external log system is required to log all the errors that occur.

Figure 4.3 shows more detail about these kinds of readers and their implementation in a supply chain conveyor.

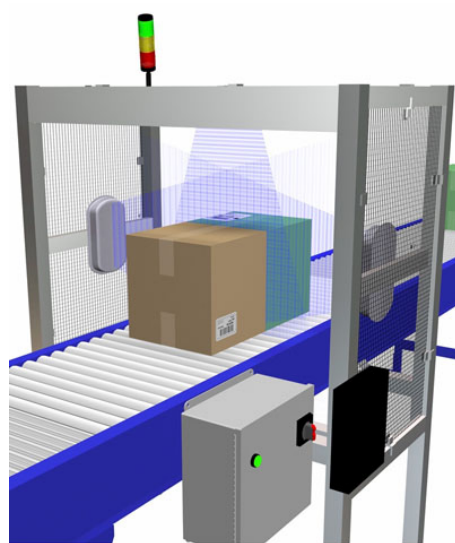


Figure 4.3: Detail in a conveyor reader - Source: <http://www.mmh.com>

Small extensions may be required according to the needs of the location. For example the reader at the warehouse docking gate may include an accurate Advanced Shipping Notice Handling System, or the conveyor reader an interaction with a robot to properly redirect items, etc.

Mobile Readers

The main responsibility of a mobile reader is to perform business logic in a non-fixed location, such as delivery and recovering of items, inventory, points of sales, etc. Figure 4.4 provides an example of a mobile station reader.



Figure 4.4: RFID Mobile Station - Source: <http://www.mmh.com>

The mobile station, usually equipped with one or more readers, is able to deal with potentially complex logic. Multiple business steps can be processed by a mobile station which may involve data orchestration between multiple systems.

Due to the nature of these readers, interfaces for human input and feedback are usually required. Location systems may also be required to properly interpret the situation.

Finally these kinds of readers and their capture applications may need to deal with offline access conditions. This may involve offline synchronization and other procedures to solve the problem.

Palletizer

A palletizer is responsible for the picking of items and organizing them in boxes ready to be distributed. When employing RFID technology in a supply chain, the proper integration between orders, clients and packages into the RFID systems is crucial. The correct stage to perform it is at the palletizer.

The palletizer may have or not an associated RFID reader, although it must be aware of RFID tags (via external system) to properly capture this information. It requires integration with the ERP, receiving order information and capturing this information on the EPCIS Repository. The connection with the palletizer/picking robot may be needed or not, depending on the implementation.

4.3.2 Capture Application Hierarchies

When employing a solution based on the EPC Network, for each deployed reader it is required an associated capture application as explained in section 2.4.1.

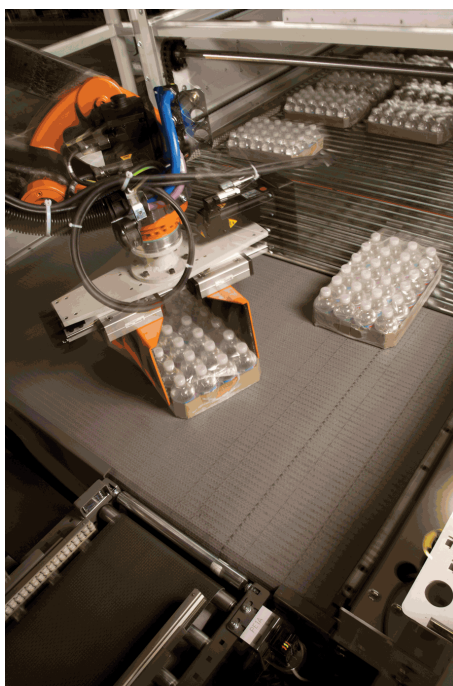


Figure 4.5: A Palletizer - Source: <http://www.intelligrated.com>

Figure 4.6 proposes a hierarchy, that starting from a “Generic Capture Application” with the basic and common methods, is extended by other subclasses suitable for the above described situations.

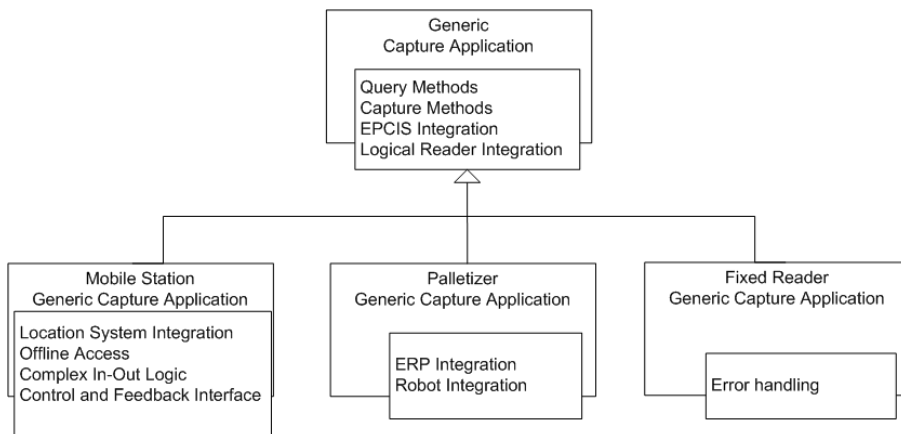


Figure 4.6: Classes of capture applications

The next section gives more detail about each of the proposed classes.

Generic Capture Application

A generic capture application is considered the simplest case of stateful capture applications. Its idea is to be a starting point for extensions for particular cases. Its basic functions are just to pro-

cess ALE Reports into EPCIS Events using only the EPCIS Repository as a source of information.

The generic capture application has the following characteristics:

- Connection with a logical reader that may be implemented by one or more physical readers;
- Connection with an EPCIS Repository and methods for querying about items;
- Link with an EPCIS Repository and methods for Capturing Events.

Subclasses of Capture Applications

Starting from the locations described above, the following capture applications subclasses are proposed:

Fixed Reader This is a generic reader located on a portal, gate or conveyor in the supply chain. Its main goal is to control items on its pathway along the supply chain. It adds a connection with a log system (which may be implemented via EPCIS Repository) to the generic capture application;

Mobile Station This resembles a mobile station that is moving on the supply chain (for example a vehicle, or an inventory station) usually responsible for the delivery, recovering or inventory items. It extends the generic capture application requiring connection with a location system, interfaces for feedback and input, and complex logic.

Palletizer This is the generic palletizer of a supply chain. It is the initial point of entry for orders. It requires connection with the ERP System, and potentially integration with a picking robot.

4.4 Framework

The framework proposed here, and given the provided case study that will be discussed on next chapter, has been developed for the logistics and transportation scenario. It mainly addresses the requirements of this particular business, although it can be presumed that it can be useful to many other environments with small adjustments.

The proposed framework has the following main packages:

Concepts (Notions) New concepts based on Fosstrak concepts, with simpler data models adapted to perform business logic. Support some methods for automatic conversion between them;

Services The services provided by the framework. It includes a Query Client and Capture Client that provide easier methods of interrogating and capturing data in the simplified model provided by concepts. Also provides a thread that will listen asynchronously and in a given port for ALE reports and automatically process them based on the implemented business rules;

Business Is an interface available with a set of interfaces that must be implemented by the application using the framework. There is also available some methods that realize simple business logic by default. These rules must be defined under the business layer (package) of the capture applications.

In the next sections the resulting packages will be detailed.

4.4.1 Concepts

This package, called Notions (noções) in the developed code, provides a simpler data model for both the EPCIS Event and ALE Report from Fosstrak. Figure 4.7 presents the Class Diagram of this package.

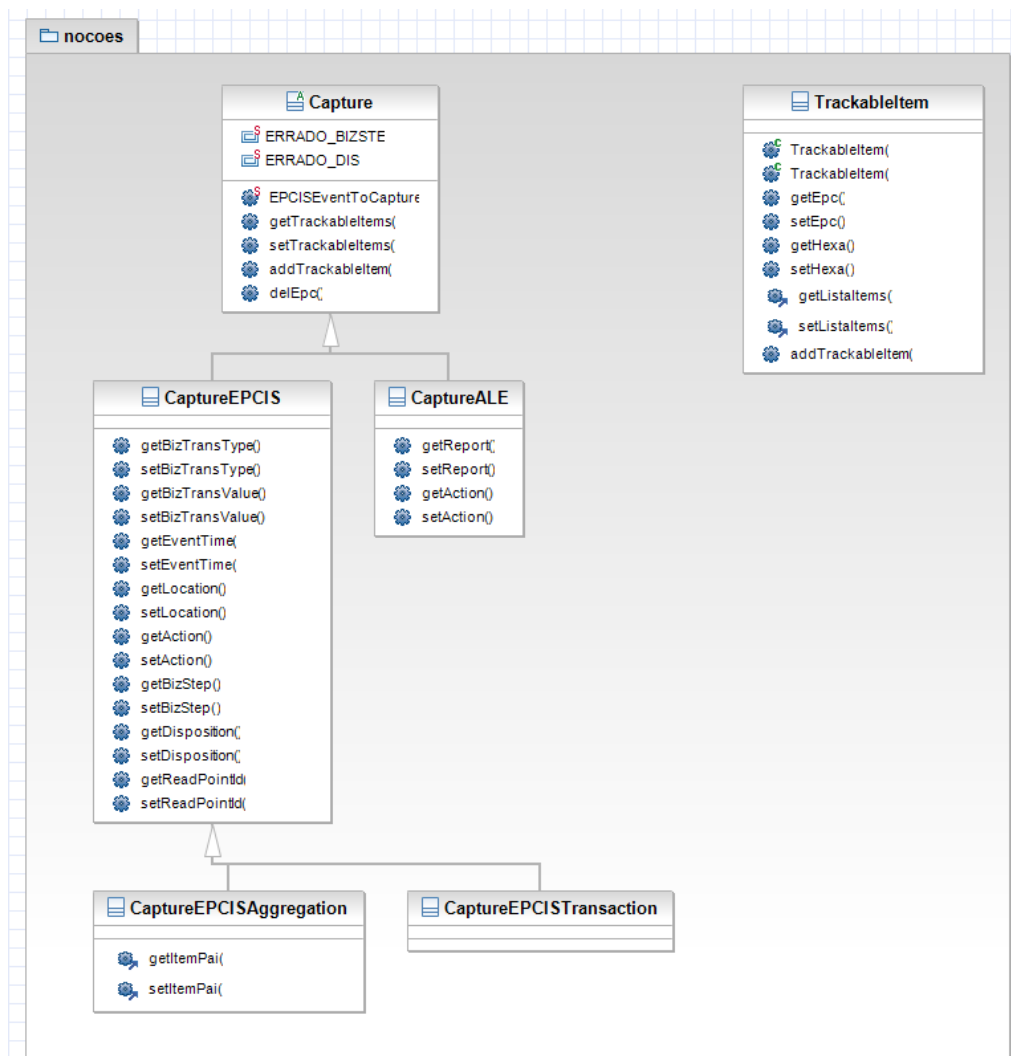


Figure 4.7: The Class Diagram of the Concepts package (A "Capture" in the diagram corresponds to an Event)

These simplified concepts try to use as much as possible the basic Java data types (e.g. String

and int) and keep the meaningful data at a higher level. Also the use of inheritance between them is an advantage as we can deal with both ALE Reports and EPCIS events in the same way, if for example we just need to access the EPC List.

TrackableItem Any generic physical object that is moving along the supply chain may be considered a TrackableItem. The closest comparison in the Fosstrak platform is the EPC. The advantage of the TrackableItem over EPC class is the capacity of aggregation. A TrackableItem has a unique EPC, but can contain other TrackableItems inside itself. For example this allows an abstraction for containers not available on the ALE Level, and introduce an **Item** notion inexistent on Fosstrak and a "**container**" concept that is potentially easier to deal than the provided Aggregation Event (see section 2.4.4) by providing aggregation of items at this level.

Capture It is an abstract class that will be implemented by a "real" event, that can either be a CaptureEPCIS (EPCIS Event) or a CaptureALE (ALE Report). It contains a list of Trackable Items, and a static method for transforming the EPCISEvent from Fosstrak structure into the framework simplified data model.

CaptureEPCIS It is one of the main classes that motivated the construction of this framework. Itself this class represents a generic EPCIS Event with a simpler data structure, and at the same time an ObjectEvent (see section 2.4.4) when capturing (via Framework) this class into the EPCIS. The choice for considering the ObjectEvent as the parent is that it has all the basic data that the AggregationEvents or TransactionEvents have.

CaptureALE This class represents a simplified data structure for an ALE Report. It has the report name and a list of TrackableItems.

CaptureEPCISTransaction and CaptureEPCISAggregation These classes are simplifying, respectively, the EPCIS TransactionEvent and the EPCIS AggregationEvent (see section 2.4.4). They contain the needed fields to support the structures defined by the standards.

The framework currently is only supporting the data standards that were need during developing the case study. As seen on the previous description, it only covers the ObjectEvent, TransactionEvent and AggregationEvent. For the moment it also does not cover the full EPCIS Event standard (see section 2.4.4 for more details). The following table shows what is implemented from the full standard:

From the ALE Report point of view, all of the specification fields are covered by the simplified data structure of the CaptureALE, using strings and integers. The exception is the read time that is being neglected at the moment (the time of processing at the capture application is used instead).

4.4.2 Services

This package provides the methods to interact with both the EPCIS and the Filtering and Collection system. It was created to automate the processing of business rules using the developed simplified concepts. It can be described by the following class diagram:

Data Structure	Implemented	Notes
eventTime	Partially	Using Calendar. The value provided by the ALE is being neglected. It is used the current time at the Capture Client when the ALE Report is processed
recordTime	Yes	Using Calendar
epcList/epcChildList	Yes	Using a list of Strings
action	Yes	Using a String
bizStep	Yes	Using a String
disposition	Yes	Using a String
readPoint	Yes	Using a String
bizLocation	Yes	Using a String
parentEPCID	Yes	Using a String
bizTrans	Partially	Currently only supporting one Business Transaction per event (Standard does not limit it)
Extensions mechanism	None	

Table 4.1: The current framework implementation status for the EPCIS Events

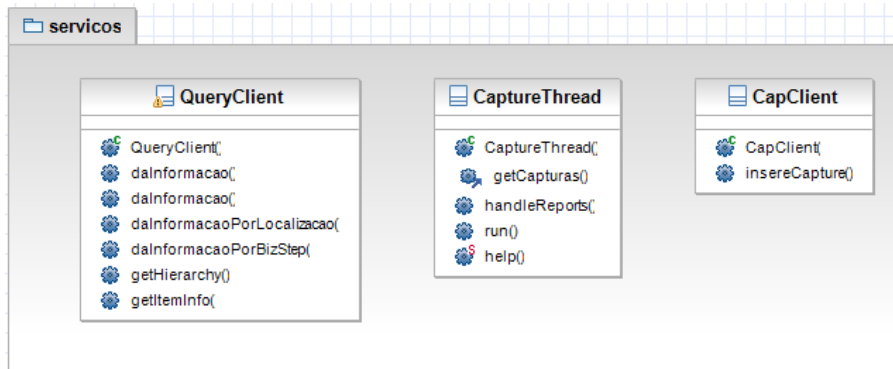


Figure 4.8: The Class Diagram of the Services package

As shown in figure 4.8, the available classes are the following:

QueryClient This class contains the methods that inquire the EPCIS Repository in a simplified way, for example inquiring directly over locations, business steps, etc. The advantage is that it is possible to use only a string for inquiring the EPCIS for all the events that happened in a location, for example. The result of this query will be a CaptureEPCIS (or CaptureEPCISAggregation or Transaction), in the framework simplified data structure for an easier application of any logic. In this class there are also methods that allow extracting a hierarchy of an item that was stored using an AggregationEvent, and getting it as a TrackableItem.

CapClient This is the class that will do the bridge between the Application and the EPCIS Repository using the framework, being responsible for the captures, by passing the framework concepts into the complex data structures (EPCISEvent) provided by Fosstrak and storing them at the EPCIS. There is a unique method that, based on the CaptureEPCIS, will perform the correct storing in the EPCIS (as ObjectEvent, AggregationEvent or TransactionEvent).

CaptureThread This is the provided thread that will run in asynchronous mode and capture the

reports as they come from the Filtering and Collection. This functionality and interaction will be described later in the prototype chapter (5.4.2).

4.4.3 Business

This package is responsible for the business logic. It contains an interface and an abstract class that must be built within the business rules. The class diagram is presented in the next figure (4.9). Some of the names are being displayed on the Portuguese language, in accordance with the source code.



Figure 4.9: The Class Diagram of the Business package

This package includes only one interface and one abstract class:

IBusinessManager (IGestorNegocio) Developers must implement this interface to define the business logic. Any capture application using the framework must set up both the handle-Capture (`tratacaptura`), function that will receive a List of CaptureALE and has to return a list of CaptureEPCIS according to the business rules. The other method is the capture (`captura`) that is responsible for the capture in the EPCIS by calling the CapClient from the service package. The operation of applying rules and store was divided in two to be able to introduce other business logic, like store the event in another system, or make a call to the ERP before capturing into the EPCIS.

ABusinessManager (AGestorNegocio) Is an implementation of the IGestorNegocio that already has some predefined functions and constructors that make implementing business rules easier, and allows the implementation of basic rules just by filling in some fields. In future it is planned that this class can read business rules from XML files and automatically apply the read business logic.

The listing 4.1 presents an example of an implementation of the handleCapture at the business layer.

4.4.4 Usage

It is not easy to define the interaction of a framework without a real application using it. Therefore this interaction will only be provided later in the section 5.4.2, using the developed prototype for a case study to test both the framework and the prototype itself.

Listing 4.1: An example of a handleCaptura (tratacaptura)

```

public boolean verificaEntrega(TrackableItem item)
{
    if(getEpcisQuery() != null){
        List<CaptureEPCIS> results = getEpcisQuery().daInformacao(
            item.getEpc());
        (...) // Deal with empty
        //Get last EPCIS Event
        CaptureEPCIS captura = results.get(results.size() - 1);
        if(captura.getBizStep().compareToIgnoreCase("lastCorrectStep")
            == 0){
            return true;
        }
        else{
            return false;
        }
    }
}

public List<CaptureEPCIS> trataCaptura(CaptureALE captura){
    List<CaptureEPCIS> capturas = new LinkedList<CaptureEPCIS>();
    CaptureEPCIS entregas = new CaptureEPCIS();
    CaptureEPCIS incorrectos = new CaptureEPCIS();

    for (TrackableItem item : captura.getTrackableItems()) {
        if(verificaEntrega(item)){ //Correct Item
            entregas.addTrackableItem(item);
        }
        else{
            incorrectos.addTrackableItem(item);
        }
    }
    //Set details for EPCIS Events from Database
    entregas.setBizStep("someBizStep");
    entregas.setDisposition("someDisposition");
    entregas.setReadPointId("someReadpoint");
    entregas.setLocation("someLocation");

    (...) // Same for incorrectos

    capturas.add(entregas);
    capturas.add(incorrectos);

    return capturas;
}

```

4.4.5 Implemented Functionality

The framework developed, from the beginning had in mind simplifying the required effort of implementing the business logic at the capture applications. In summary, the framework provides:

- A concept called `TrackableItem` that is comparable to the item or packages of the real world, which can contain other items, and is not implemented in Fosstrak;
- A set of Capture concepts, which can inherit the EPCIS Events or the Reports ALE, in a simple data structure, which is potentially easier to interpret, manipulate and apply business logic over it, by raising the abstraction level;
- An asynchronous mechanism that automates the capture of ALE Reports into an EPCIS Event using the defined business logic;
- A class “`QueryEPCIS`” that allows easy querying of the EPCIS Repository and returning the results in simplified data structures;
- A class “`CaptureEPCIS`” that performs the complex capture process that using the simplified data structures captures it as EPCIS Event in the repository;
- Allows to easily get the entire story about one item, as well as its hierarchy (parent or child items in case of `AggregationEvent`), under the simplifying concepts by calling a single method. This functionality is not implemented in Fosstrak;
- An interface “`IBusinessManager`” (`IGestorNegocio`) describing clearly the methods that are required to the business logic.

4.5 Framework and Capture Application Hierarchies

With the framework and the set of capture applications subclasses introduced, this section clarifies where the implementation effort must be spent when following the implementation guidelines, having a better overview of the entire scope of the problem before jumping into the case implementation.

Figure 4.10 presents the implementation effort when developing a capture application starting from the framework. As the caption suggests, blue packages require more effort to configure/implement the business requirements, while the white ones can be considered almost static and used in most implementations without changes.

For example the business package is strongly dependent on the requirements therefore the configuration/implementation of this package will be required. From the other side, the concepts package can be considered generic taking into consideration the logistics and transportation scenario, and no effort should be required. Finally the services package may or may not need to be extended, depending on the situation where the reader will be implemented.

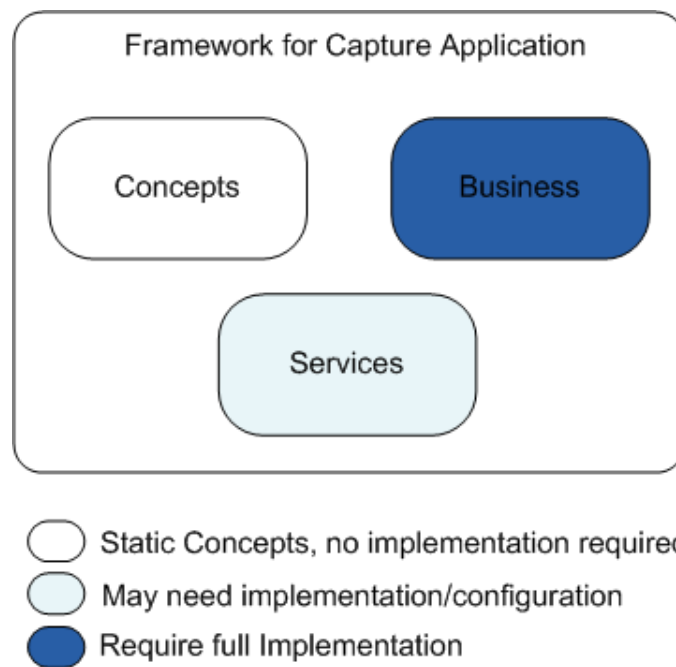


Figure 4.10: The required effort of implementing a Capture Application following the framework

Chapter 5

Case Study and Prototype

5.1 Case Study

To test and assess the Fosstrak platform, as well as the developed framework, a case study was provided by Link Consulting (www.link.pt).

The provided case study is related with pharmaceutical distribution and logistics, a business that involves millions of Euros and can be easily distinguished from other more traditional logistics by particular requirements related with the nature of the items being transported. Any mistake in the supply chain may result in health injuries.

The company called FarmaDistr (fictitious name) had some problems on its logistics processes and required a solution to control the costs, improve the visibility, and optimize its supply chain. The part of the business that is considered in this case study is just the outbound distribution of the items to its clients, not the whole supply chain where FarmaDistr may be involved.

5.1.1 Business information

FarmaDistr distributes pharmaceutical items, which is a logistics business with special cares.

FarmaDistr delivers medicine to the final client inside special boxes called containers (banheiras) that should not be open after the picking performed at the warehouse and until the final destination. Once containers are delivered, usually they are left at client facilities and potentially returned on a next delivery. The need to track the containers back is one of the motivations for the RFID system.

The clients of the FarmaDistr are mainly Pharmacies but there may also be other entities like: government, hospitals, private clinics, or any other that needs big quantities of drugs. The main suppliers are the laboratories, or any other company in the middle that imports the drugs at national level. It is considered that FarmaDistr already has a picking automation system that is in charge of carrying out the fulfillment of the containers. This automation is already integrated

with the ERP.

The system can be considered in a closed loop for the containers (banheiras). Banheiras must be returned and FarmaDistr is aware of their location all the time, which means that the items are all the time inside the boundaries of its supply chain.

Regarding the medicine, it may be considered in an open loop since it involves interaction with other entities. The medicine comes from the suppliers and is delivered to the FarmaDistr clients, therefore the product moves in and out of the company's boundaries.

There are still two more concepts that are important to mention. Both of them "virtual" as they only exist on information systems. The first is the order number that came from an Enterprise Resource Planning (ERP) system. The information about the client, delivery destination, and the ordered items exists in this system. The other important notion is the package that is considered the aggregation of all the containers that were ordered at the same time and therefore have a common order number. Each package is linked uniquely to an order number.

In summary, the business concepts are the following:

Medicine This is the trading item of the company, what clients are paying for and are interested in. The safe transport of the items to the clients is crucial;

Containers Boxes where medicines are delivered to the final client. Each container can have multiple medicines inside. Every container is left at the client and returned to the warehouse in a later delivery;

Package It is a virtual notion that does not exist physically. It is considered a group of containers that have been ordered under the same order number;

Order number Created by the Order Entry System or the ERP, which will be kept and associated with a single package.

The used ERP system can be considered a proprietary system and is going to be ignored. It is assumed that the system is able to incorporate on its pipeline communication with other web services over the network.

FarmaDistr may also have multiple warehouses distributed over the country, but for the case study and for simplification reasons it is going to be assumed as a single warehouse.

5.1.2 Business Processes As-Is (Current Situation)

Currently the logistics technology and processes of FarmaDistr are based on manual optical barcode readers. The tracking currently employed is performed only at the container level, for technical and organizational limitations. Each container has a unique barcode that is manually read in three key locations: at the vehicle when starting or ending a delivery in the warehouse, at a client's location when delivering or returning items, and at the warehouse in the picking automation system.

The circuit is started with the client's order, which is inserted through the order entry system that will pass the order to picking automation. This one will then fulfill the order, creating the needed packages and making the association in the systems between containers, packages, medicines and order numbers. For example it will be inserted that container X with medicine 123 and 134 and Y with medicine 981 and 982 belong to the package 32 and is the order number 1234 and has to be delivered at the Pharmacy Z.

From this point onward, items are going to be read when getting inside the vehicle and at client destinations, using the optical hand reader. All the containers must be read, when delivered to and returned from the client.

Although FarmaDistr's whole logistic process could be described with more **business steps**, the following ones are the considered in the actual tracking system that the prototype covers:

1. **Picked** - The step of picking (pick the items from shelves into the containers), it is consolidated when the containers are related to the packages and order numbers, and the physical package is ready to be sent;
2. **Sent** - This step occurs when the containers are carried into the vehicle, and read by the onboard optical reader;
3. **Delivered** - It occurs when containers are delivered at the clients. It means a read at a client's location;
4. **Returned** - The business step associated with the return of containers from the clients.

For example, the normal cycle of the containers across the supply chain using the described business steps can be seen on figure 5.1.

Main problems in supply chain

The main problems that FarmaDistr is facing are:

- Too many items are being lost on the supply chain. Currently used technology cannot do anything to prevent this;
- The process of delivery is slow as it involves reading multiple boxes using manual hand readers;
- There is the risk that counterfeit medicine is inserted into FarmaDistr boundaries;
- The visibility of the items on the supply chain is poor and needs to be improved.

5.1.3 Business Processes To-Be (Future Situation)

FarmaDistr wants to follow other companies that successfully implemented RFID on their logistics processes, and adjusted themselves to the market's needs. FarmaDistr expects that by employing RFID technology along its supply chain it can gain market advantage as well correct some of its current supply chain management issues.

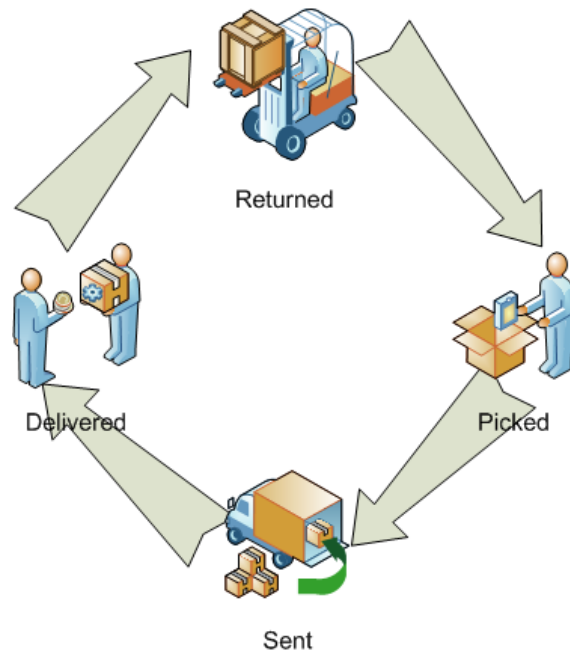


Figure 5.1: The current business steps

Other companies, like Throttleman [28], which is a good example in Portugal, has valorized itself after a successful adoption of a RFID system. Success cases like this one, convinced FarmaDistr's management to move towards RFID.

"Our RFID (...) system has provided a number of important benefits including a reduction of 60 percent in storage space in our distribution center, faster shipments to our stores that translate into a gain of four selling days, and a significant optimization of our back-end processes. In addition, we now have the ability to order by store rather than by family item, and this further simplifies our operations." by Miguel Maya, Associate General Director of Throttleman [28].

By adopting RFID technology, FarmaDistr has the following main goals:

- Track the items automatically in main business locations, to improve the visibility;
- Ease and speed the processes of delivering and returning items;
- Ease and expedite the process of delivering and returning the items;
- Keep a history of every item in its way along the supply chain;
- Fight the counterfeit of medicines;
- Decrease the number of lost items on their path across the supply chain;
- Expedite the process of receiving inbound items;
- Increase as much as possible the efficiency of the system, keeping the costs under control.

5.2 Prototype Decisions

This section describes the decisions that were made to solve the business problems of FarmaDistr and provides the technical details of the prototype.

5.2.1 Business Decisions

Most of the choices described next were decided together with Link Consulting.

Frequency

What RFID frequency/technology should be adopted in this particular project? The general choices on this issue are described on section 2.1.4. In this track and trace project, we opted for UHF frequency together with EPC Class1 Gen2 tags. It is important to add, that as reviewed on section 2.1.4, this frequency is commonly accepted from the market experience to be the more adequate to the tracking of items in a supply chain of “mid-priced” items due to its main advantages: low cost (i.e. cheap tags), range, and no need for batteries.

RFID Reader Locations

What are the places that need to be RFID-enabled to ensure the best visibility over the supply chain and assure an adequate tracking.

The warehouse “expedition gate” should be one of these points. Every item going in and out of the warehouse passes through this point, making it strategic.

A little more complicated decision was the location for the reader that confirms the delivery at the final client (delivered in the old business steps).

Two choices emerged for this problem: the first one would be enabling the vehicles with door mounted RFID readers and use the “in” and “out” complex semantic of the items to realize the business logic, by being aware of the last state of each item. The second choice would to enable the pharmacies with reader and use only these locations as delivery point confirmation using a simpler logic.

After a deep analysis of this question, it was decided to use the vehicle as the location for the reader as it provided a better level of tracking with more steps being tracked using only the onboard reader. Also this solution needs fewer types of equipment because there are more clients than vehicles, which potentially makes this a cheaper solution.

It was decided for simplification’s sake to use only two locations. Given the case and to better fulfill the requirements, it was decided to use the **warehouse gate** and **readers inside vehicles**.

Level of tagging

It was important to settle the level of tracking that was going to be adopted. This question is usually closely related to the nature of the tagged items.

For this particular case study it is needed to choose one from the two available choices: the pallet-level and the item-level.

Item-level Every item or just the lower level items are tagged. In this case only the medicine will be tagged, and the container level is optional. This would enable much more possibilities, like the anticounterfeiting or theft problems. It can be troubling and expensive in cost and labor force to adopt if the items are not already RFID tagged by their suppliers;

Pallet-level Using this method only the pallet items would be tagged. In this case it would mean that only the containers across the supply chain are RFID enabled. The main advantage is to keep the processes simpler and more similar to what is implemented. As an issue to this solution is using this tracking most of the main objectives of FarmaDistr are hardly achieved.

Item-level tagging was required to achieve almost every point from the list of objectives of FarmaDistr. The choice of the level of tagging was for **item-level tagging**, where all items including packages are tagged.

Global Positioning System (GPS) Integration

In matters of integration of the prototype with other external systems, a GPS localization system was developed that was needed to equip the vehicle with the reader to make it aware of its location. Given a coordinate, the system will provide the name of the nearby pharmacy.

This functionality was developed not only as a real need but also to prove that we can integrate other systems to enrich the information read by the RFID readers orchestrating this and other sources to provide a better quality of tracking.

5.2.2 Technical Decisions

This section describes the main technical decisions that were made in the prototype.

Capture Applications

Most of the effort of implementing the prototype was in developing the respective “capture applications”. There are some choices regarding their implementation. The first choice is related with the use of Stateless or Stateful capture applications (see section 4.2). The second refers to the centralization of the system: should be use a central capture application that would process the ALE Reports coming from every reader across the organization, or multiple capture applications (distributed system) each of them responsible for its directly associated readers (for example a capture application for the vehicle, other the warehouse gate, etc.)

The final decisions was for the **Stateful Capture applications** that are closely related with the real case implementations and the traditional ERP methods.

It was also decided that the capture applications will be distributed along the locations, instead of centralized capture application. This allows rules simplifications, an easier development, and

more freedom at the deployment time. It may be important to mention that all these capture applications can be running in the same central computer as if they were just one.

EPCIS Values

As seen on section 2.4.4, the EPCglobal does not define the standard values for the EPCIS data fields, for example how a shipment can be designated. For these reasons, it was needed to create some values to be used, each of them with business meaning, and each of them representing corresponding business logic.

To solve this, the examples provided by Fosstrak were adapted to the prototype's needs and used.

Reader Simulation

The readers supported by Fosstrak during the development time were limited, and because of its costs and availability, Link was not able to provide me with such equipments. One reader supporting UHF was available, but without support to the EPCglobal Reader Protocol or Low Level Reader Protocol. The effort required to connect them with the Fosstrak due to involve low-level programming was out of the scope of the thesis.

To be able to continue with this prototype it was decided to use the Fosstrak reader simulator with some minor changes to adapt it to the medicine scenario.

5.2.3 Assumptions

To be able to develop the prototype with the technical and business decisions taken above it was still needed to make some assumptions. These assumptions were needed to simplify and do not get into detail for things that are out of the scope of this thesis.

Tagged Medicine Under this thesis it is assumed that all the medicines will come either tagged from the manufacturer or will be manually tagged by FarmaDistr, or any other third-party company; therefore all distributed medicine are tagged;

Connectivity We are assuming there is connectivity in every point of the tracking system. The critical point is the vehicles that need to be provided with a GSM (Global System for Mobile communications) connection or similar to have a full-time connection with the central systems.

Vehicle reader There are some commercial available solutions of RFID door mounted readers for vehicles, like for example the IV7 Vehicle Mount RFID Reader from Intermec [40]. Their correct operation is assumed, and that they can clearly read only those items that are getting in or out of the vehicle with an acceptable low rate read failure.

EPCIS Master Data As seen on section 2.4.4, the EPCIS interface for capturing Master EPCIS data is not yet defined by the standards. The data related to objects descriptions (Master Data) will be stored in a database which runs side by side with the EPCIS Repository. The EPCIS Repository will be only used for storage of event data.

5.2.4 New Business Description

In the following section we present the process changes, the options and the new interactions with the system.

Business Concepts

The four kinds of items from the previous business model will be kept. This allows keeping new processes closer to previous one, making the implementation at operational level simpler.

The following list presents the changes that are required to be made to the business concepts:

Medicine Every medicine requires to be EPC Tagged in the new system;

Container Must be RFID enabled as containing a unique EPC identifier. The association between medicines and containers will be stored in the system;

Package It is a virtual notion as it does not exist physically. It is a group of containers that have been ordered at the same time with the same delivery location. Each package will be associated with a unique EPC created at the system and inserted into the EPCIS of the prototype solution;

Order number This number will be associated in the EPCIS with the unique EPC identifier created for the package.

Under the framework, at the capture applications, these items will be considered and treated as TrackableItems (see section 4.4.1), with special focus the Medicine and Containers, instead of only EPC's as Fosstrak natively supports. The relation between these objects using a class diagram can be represented as the following:

There are also a few extra business concepts that will need to be RFID identified. Each location and vehicle and client must have his own identifier to be uniquely distinguishable from others. The following list shows the items that are uniquely distinguished in the proposed RFID system:

Clients/Locations Each client needs to be associated with a unique EPC identifier. It is assumed one location by client;

Warehouses Each warehouse has a single EPC identifier;

Vehicles The same for vehicles, each one can be distinguished from the others by its unique EPC identifier.

New business logic

To implement the final solution the current business processes had to be analyzed and reengineered. The current accounted business steps as told in section 5.1.2 were not enough to settle a proper visibility across the supply chain using RFID as intended. The proposed solution, starting from the existing processes and steps, created some new business steps that were needed to enrich the supply chain.

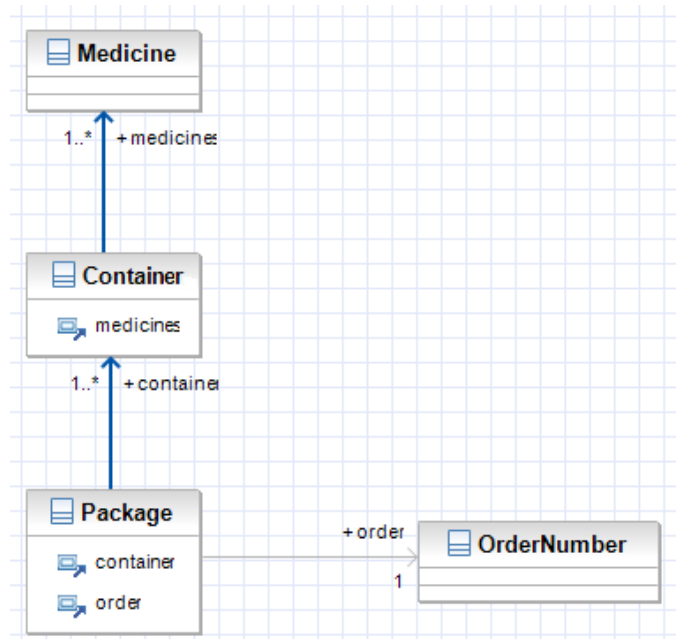


Figure 5.2: The relation between basic items in a class diagram

After some considerations it was decided to use the following business steps, in the natural order of the supply chain:

Picked The same state than in the previous process;

Shipping This new business step is related to the item arriving at the expedition bay. When an item is read at this zone, its state is set to shipping. In this state items are considered to be waiting to get into the vehicle, to start the shipping process;

Shipped This business step is related to the existing one;

Delivering Another new step. It occurs, when an object is taken out of the vehicle in an intermediate state before final delivery;

Delivered It is the same step as the one existing in the previous model;

Returning This new step accounts the state when objects are in transit inside the vehicle being returned to warehouse.

Recovering This step is related to returning items to the expedition bay, and is set when the items are read out from the vehicle at a warehouse location, and are waiting for the final recovery;

Recovered It is related with the old business step of Returned.

More intermediate steps could be created to increase the visibility of the system with the addition of extra readers and capture applications to the solution. Although for simplification it was decided to proceed with the ones mentioned above.

The **error** state was also created to represent what happens when an item goes out of the correct order of processing. At the prototype depending on the occurring error, only some of the error states can be corrected and return to the correct order.

The next figure shows the natural order of states and how they succeed one another, and under which conditions. The supply chain has been summarized in a finite state machine which provides a better understanding, and allows an easier implementation later on.

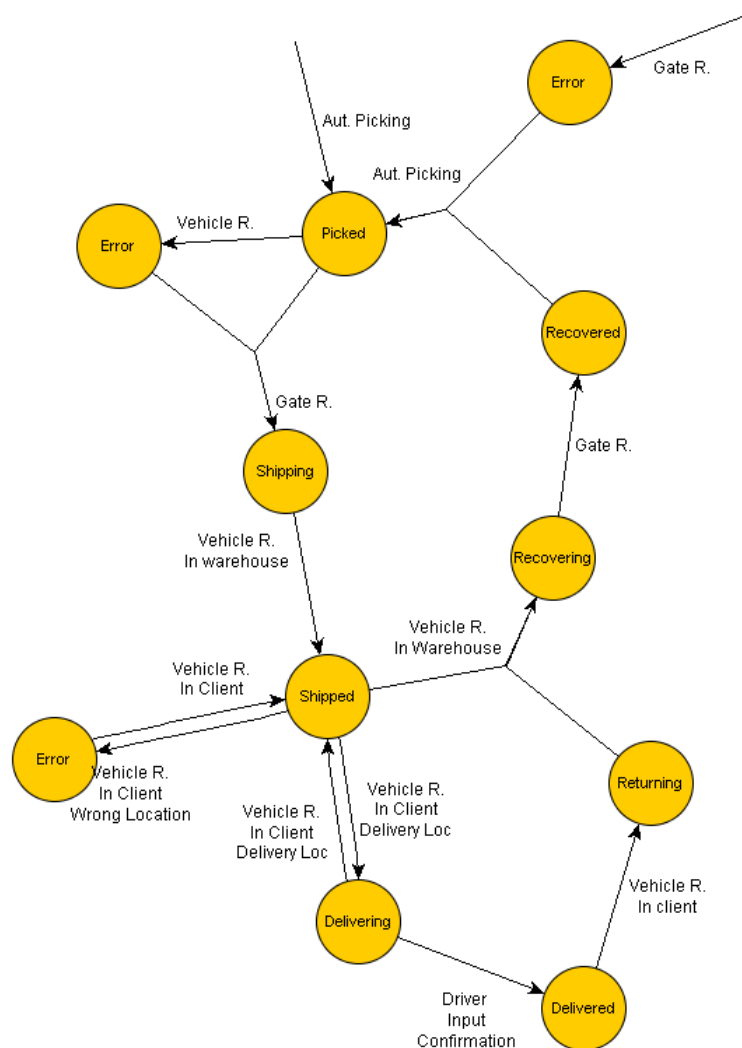


Figure 5.3: The new business states

5.3 Prototype Technical Specification

This section describes the technical specification of the developed prototype.

5.3.1 Components

With the requirements clear and the decisions taken, it was possible to understand what modules were needed to provide the desired functionality.

The prototype development was done in 4 different modules:

Automation Responsible for assembling the order entry from the ERP or Entry Order System (manual input for now), and then command the automation robot to create the needed packages;

Warehouse expedition gate Performs the business logic for the Warehouse reader;

Vehicle This is the most complex component and where the business logic for the vehicle is;

Tracking Responsible for inquiring the EPCIS and showing the tracking data to the company and its partners.

Each one of these modules has been developed using the proposed framework to smooth the development process. In the next sections more detail is presented for the capture applications. The rest of the RFID system was done using the modules provided by Fosstrak.

The modules were developed using Java servlets [30] technology, which can be easily deployed as a container in any compatible environment like Tomcat. This choice was made mostly because of the need of an interface that could be easily prototyped, as well as the simplicity of the Servlet programming model.

The implementation has taken in consideration the guidelines presented on section 4.3, decided according to the business/capture situation. Although taking in consideration these guidelines, all the development was done having as basis the code proposed for the generic capture application 4.3.2.

5.3.2 Automation System

This system follows the implementation guidelines for a palletizer (see section 4.3.2) since they have similar requirements.

It is considered the entry point for the orders created by the ERP system. Once an order is entered through the interface or the web service (not implemented, must be manually inputted via frontend), the picking machine is ordered to perform the physical picking of the items, associating the medicine to the containers.

It is also responsible for inserting the items at the EPCIS in case they are not yet in this system. It reads data from EPCIS for items that have previous events, and Master Data database for new items that do not have any previous EPCIS event.

The current implemented order entry provides a web interface, where it is possible to associate each free container with the medicine in stock, as well as the destination and the order number.

From this point that package is in the EPCIS system, and ready to keep his trail in the supply chain.

This system is responsible for the following business steps (without error states):

- (Not Present) -> Picked
- Returned -> Picked

5.3.3 Warehouse Expedition Gate System

This module follows the implementation guideline suggested at 4.3.2 to the fixed reader, due to its requirements of tracking a key location where no special rules or systems are involved.

It has a logical reader (see section 2.3.2) associated, and is responsible for inserting a new state into the EPCIS to improve visibility over the supply chain. It also sends warnings if “lost” items are passing through its range. For example, a not “picked” item passing at this point will send a warning.

This system is responsible for the following business steps (without error states):

- Picked -> Shipping
- Returning -> Recovered

5.3.4 Vehicle

This is the system that is potentially running at the vehicles, and is responsible not only to perform tracking of items out of the warehouse but also to assure their correct delivery. It follows the guidelines presented on the section 4.3.2 for the mobile station.

It has a web interface where the current status of deliveries can be seen: wrong delivered items, and to be returned items that have been shipped in and out of the vehicle.

This system also needs a logical reader associated, which will mostly be a vehicle mounted door reader.

This system is responsible for the following business steps (without error states):

- Shipping -> Shipped
- Shipped -> Delivering
- Shipped -> Recovering
- Delivering -> Shipped
- Delivering -> Delivered (with user input)
- Delivered -> Returning
- Returning -> Recovering

5.3.5 Tracking

This module allows extracting information from the EPCIS Repository and presents it in a graphical way, easily readable by humans. Since it does not capture EPCIS Events into the Repository and having no readers associated, the Tracking Module is a special case. Its implementation started from the generic capture application proposed at 4.3.2.

It inquires both EPCIS and Master Data database to collect event history across the supply chain and item information, presenting in the data, with the item description, the historical of an item and special focus to the last event recorded.

In figure 5.4 we inquired the system about the state of item "container26" that is a container.

Item Tracked	Banheira26
Tipo Item	banheira
EPC	urn:epc:id:sgtin:00037000.30241.33554426
Ultimo tracking	2008-08-19 21:48:44
Ultima localizacao	Armazem3
Ultimo bizStep	urn:accada:demo:bizstep:error

Histórico :

Data	EPC	BizStep	Estado	Localizacao	ReadPoint
2008-08-19 21:33:00	Banheira26	urn:accada:demo:bizstep:picked	urn:accada:demo:disp:readyToDepartue	Armazem3	urn:accada:demo:rp:automato1:leit0r1
2008-08-19 21:33:00	Xanax3, Xanax2, Xanax1	urn:accada:demo:bizstep:picked	urn:accada:demo:disp:readyToDepartue	Armazem3	urn:accada:demo:rp:automato1:leit0r1
2008-08-19 21:35:44	Xanax1, Xanax2, Xanax3, Banheira26	urn:accada:demo:bizstep:shipping	urn:accada:demo:disp:waitingForVehicle	Porta 2 - Armazem 3	urn:accada:demo:rp:armazem3:porta2:leit0r1
2008-08-19 21:37:51	Banheira26	urn:accada:demo:bizstep:shipped	urn:accada:demo:disp:InDelivery	Armazem3	urn:accada:demo:rp:vehicle3:leit0r1
2008-08-19 21:39:21	Xanax1, Xanax2, Xanax3, Banheira26	urn:accada:demo:bizstep:delivering	urn:accada:demo:disp:outofvehicle	Farmacia Santos	urn:accada:demo:rp:vehicle3:leit0r1
2008-08-19 21:41:51	Xanax1, Xanax2, Xanax3, Banheira26	urn:accada:demo:bizstep:Delivered	urn:accada:demo:disp:InClient	Farmacia Santos	urn:accada:demo:rp:vehicle3:leit0r1
2008-08-19 21:46:20	Banheira26	urn:accada:demo:bizstep:returning	urn:accada:demo:disp:invehicle	Farmacia Santos	urn:accada:demo:rp:vehicle3:leit0r1
2008-08-19 21:48:20	Banheira26	urn:accada:demo:bizstep:error	urn:accada:demo:disp:perdido	Porta 2 - Armazem 3	urn:accada:demo:rp:armazem3:porta2:leit0r1
2008-08-19 21:48:44	Banheira26	urn:accada:demo:bizstep:error	urn:accada:demo:disp:perdido	Armazem3	urn:accada:demo:rp:vehicle3:leit0r1

Figure 5.4: Screenshot of the Tracking Interface

This system currently allows us to inquire the EPCIS using the following properties:

System items Using one of the existing items on the Master Data database. Using this method there is no need to manually enter the EPC;

EPC Manually input the EPC of an item that exists on the system;

Localization Inquiring all the events that occurred in a given localization, that can be a client, the warehouse gate, etc;

Business Step To search all the events that have the given Business Step;

Order Number Get the items that are related to a given ERP order number.

5.3.6 Deployment

The deployment is the assembly or transformation from a packaged form to an operational working state. We present the development and prototype implementation as well as a possible deployment

model in a real world.

Development Deployment

During the development of this work and for simplifying issues it has used a system with:

- Two Apache Tomcat Servlets containers, one for the Fosstrak web services (Filtering and Collection and EPCIS), and one for development environment of the modules;
- Two readers, using the Fosstrak reader simulator, one configured for the Vehicle, and one for the Warehouse gate;
- One MySQL server running two databases, one for the EPCIS schema and one for item description (Master Data).

All these modules were running on the same machine. Figure 5.5 presents the development deployment diagram.

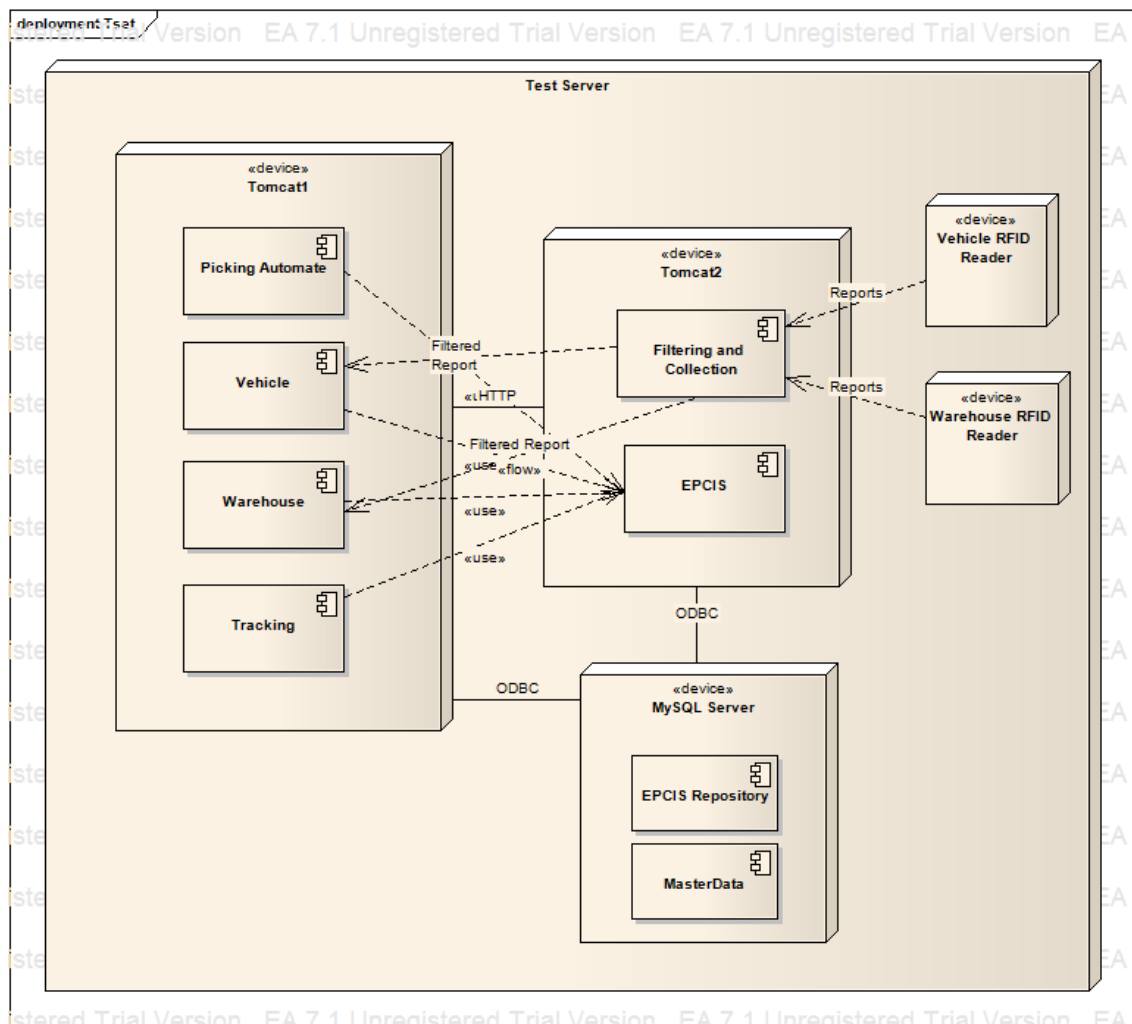


Figure 5.5: The prototype deployment diagram

Potential Deployment

A real-world solution for the deployment would use multiple servers, as depicted in the following deployment diagram:

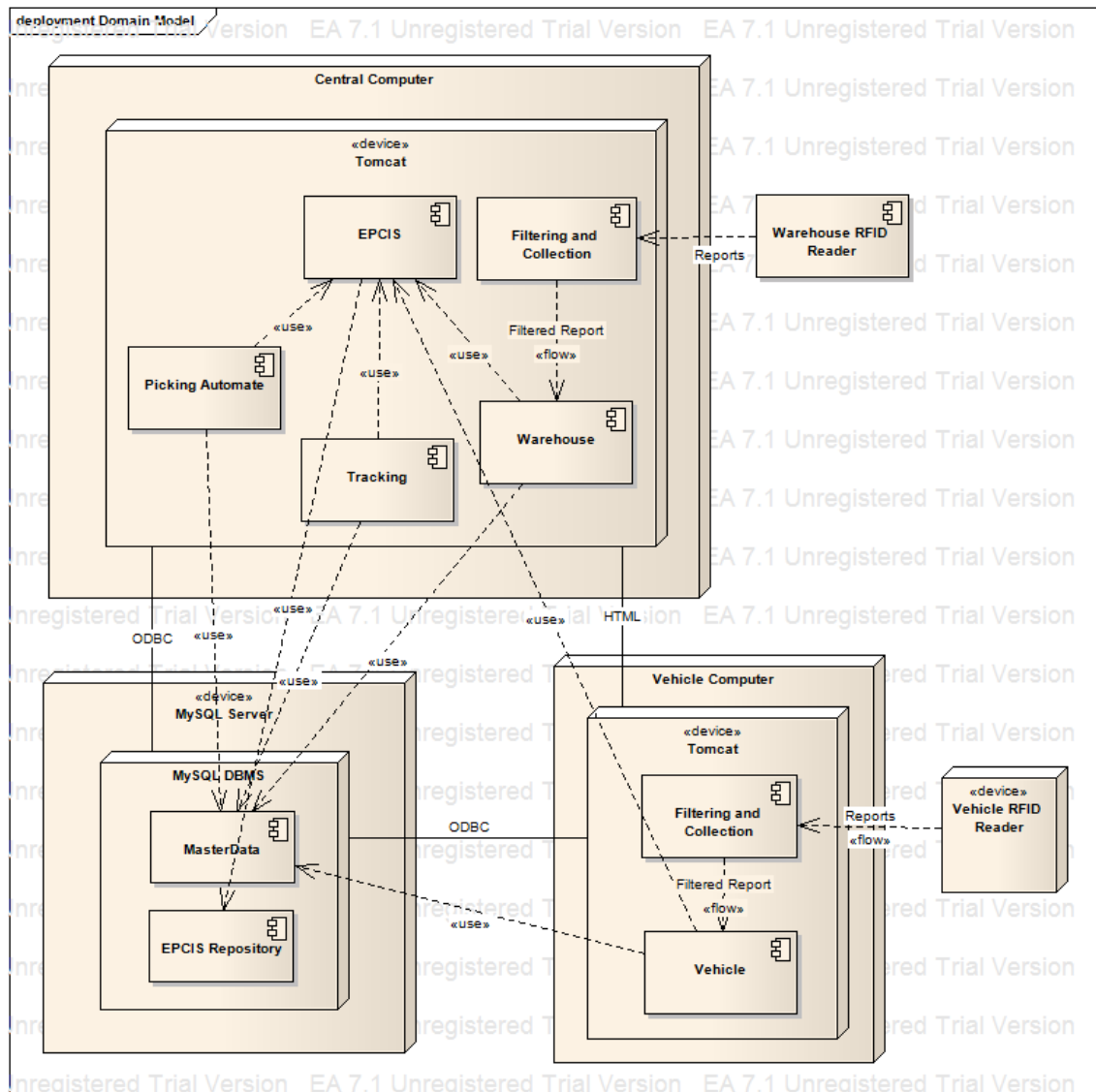


Figure 5.6: A potential real-world deployment diagram

In this case the vehicle would have its own Tomcat, with less need for communication with central services, just for EPCIS and product description queries. The DBMS is also isolated from the Servlet container server, which would result well, especially because the need of EPCIS for the business logic.

Any other solution with the same components could be viable. With the cost and reliability of mobile internet solutions being acceptable, there are multiple choices for the deployment.

If this connectivity is still a problem, a solution without need for the wireless connection is also possible. The solution looks like the one presented above, but with a snapshot of the EPCIS

(containing the data needed for the daily activity, for example) at the vehicle system. Using a solution like SQLAnywhere [41] the data could be synchronized hourly or daily for example.

5.4 Prototype Use

This section describes the interaction of the framework with an application. In this case we are going to use the Warehouse Gate system. It is also going to be present the comparison of framework against the direct use of Fosstrak.

5.4.1 Example

In this section it is going to be shown an example of interaction between all the systems, simulating a real world use. This is also the example that has been used to demonstrate the platform.

For this we are going to consider that we want to deliver two orders:

Order 1234 - Pharmacy Santos	
Container26	Xanax1
	Xanax2
	Xanax3

Order 4321 - Pharmacy Avenida	
Container31	Xanax8
	Xanax9

In this delivery the Container26 has to be returned before returning the vehicle.

To perform a cycle of the items across the system, and test our components, the following steps can describe a normal interaction with the system:

1. To input the above order we have to start by the automation system entering the order in the web interface, doing the association between the container and the medicine, as well as the order numbers. After this step the items will exist in the EPCIS in the **picked** state.
2. In the next step the packages must pass the warehouse expedition gate to get into the vehicle. To do this we must pass the items by the corresponding reader. The system that the reader is connected with, will receive the information and register the packages with the matching business step: **shipping**
3. From the warehouse, packages will be placed into the vehicle. To perform this we must use this vehicle system web, and in the web interface set the coordinates to one of the Warehouses. After this we pass the items in the associated reader, and checking the interface to confirm the items are being stowed correctly into the truck. To test this system we can forget one item in first pass, and confirm that this item will be in red as warning that it is missing. After confirming, we can pass all items and confirm that none is missing in the interface. All the items will be set as **shipped** in EPCIS.

4. After the packages being inside the vehicle, we start the delivery process. We need to start by setting the coordinates to the first client, the Pharmacy XYZ, then by mistake we will take the two packages (the correct and the wrong) out of the truck, by passing them by the reader. We can then confirm that the correct items will be shown correctly delivered in the interface, and all the items of the other package as wrongly delivered. At this point all the items will be inserted with the **delivering** state into the EPCIS. To correct the situation we must pass the incorrect delivered items again into the truck by passing them again in the reader. The corrected items will be set as shipped once again.
5. The next step is confirming the delivery, the driver should confirm the delivery in the interface. From this point every item outside the truck and with the delivering state will be set as **delivered**.
6. We can move to the next pharmacy to finish the delivery of another package. After setting the correct coordinates, we will once more forget one item, by passing all the items that were in the vehicle except one. It can be easily confirmed at vehicle system interface that the forgotten item is missing. We can assume the missing medicine item was not wanted by the client by some reason and must be returned.
7. Once more the driver must confirm the delivery, and the accepted items will be set as delivered
8. Now we move again to the first delivered pharmacy, by setting again at vehicle interface that coordinates, and move the container into the truck, by passing at the reader. This container will be set to the **returning** business step in the EPCIS Repository.
9. Now to conclude the returning, we must set to vehicle to the Warehouse location, and take every item outside by passing them at the associated reader. Every item at this point will be set as **recovering**.
10. The last step is done at the warehouse expedition gate system. We must pass the items by the reader, and returning items will get the business step of **recovered**. From now every returned item can be used again for new orders. It can be confirmed at the automation System, that the returned container and medicine are again available for a new delivery.

5.4.2 Interaction with Framework

In the following diagram is possible to see the interaction between the Warehouse gate system developed for the prototype, and the Framework. The business logic of this system, is all implemented in the class BusinessLogic, by implementing the interface provided by the framework.

The UML Sequence diagram is shown in figure 5.7.

The diagram function can be summarized in the following steps:

1. The Warehouse Gate System, will create the thread, supported by the framework that will run asynchronously as well as implement the business rules by implementing framework interface, or the abstract class;

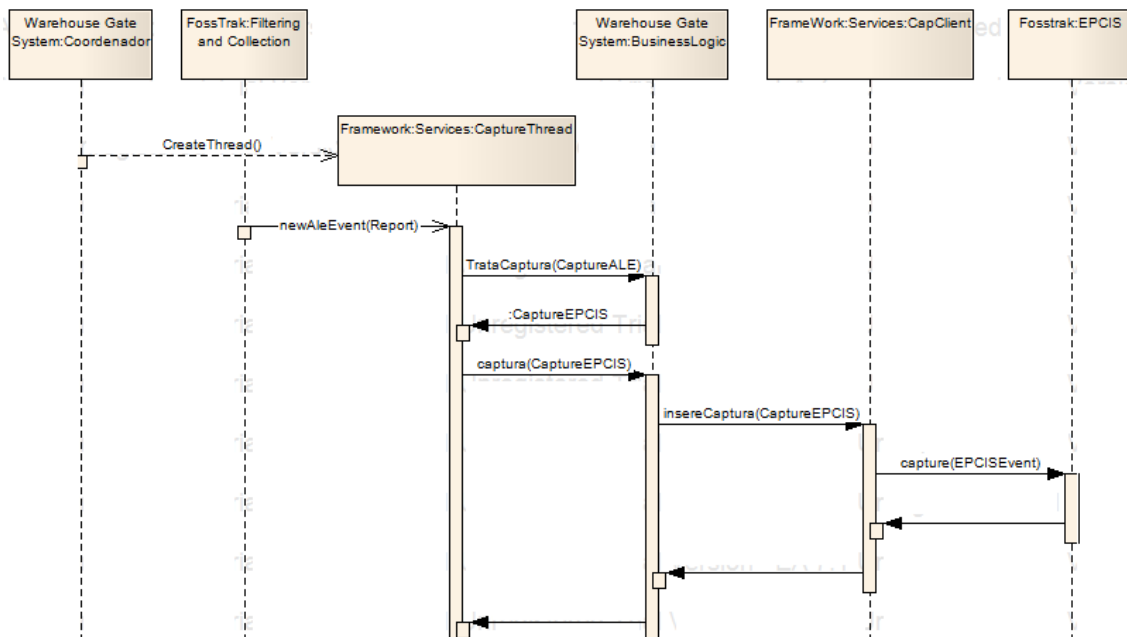


Figure 5.7: The UML sequence diagram of an interaction

2. After the creation, the thread will be running, and waiting for the ALE Reports in the given port. When a report income from the filtering and collection module, the thread will transform the ALE Report into a CaptureALE;
3. The same running thread, will then call the function trataCaptura (handleCapture) implemented in the business logic class;
4. This class, as seen before, is responsible for transforming a list of CaptureALE into a list of CaptureEPCIS, and of returning them to the thread;
5. The thread, if configured to save promptly the events, will call the function captura (capture) once again from the business logic;
6. This invoked class will call insereCaptura(insertCapture), from the CapClient class of the framework, that will do the bridge between the data models, and insert the proper Capture into the EPCIS Repository calling this service.

As seen the sequence diagram demonstrated, the framework was a most value for the prototype development.

5.5 Conclusion

This chapter described the developed prototype and the needed business and technical decisions, assumptions needed to be taken before jump into the prototype. It introduced then the business description before (as-is) and what the company pretended after RFID implementation (to-be). Finally the prototype specification was presented and to conclude an example of usage of the prototype as well as its interaction with the framework.

Chapter 6

Evaluation

This chapter evaluates and comments the main subjects of this thesis, from the EPCglobal Standards, to the proposed framework, including Fosstrak, the hierarchies of capture applications, and the prototype developed to implement the case study.

6.1 EPCglobal

RFID technologies require a huge effort in terms of standardization. RFID standards are a major issue to secure the high investments in RFID technology on different levels. [17]

6.1.1 Limited Range of Solution

The architecture proposed by the EPCglobal Umbrella, pretends to cover the full scope of the supply chain's RFID systems. Although from one side this can be positive when we think on standardizing a solution, it can also pose some new problems.

For example the proposed standards are limited, where the ISO standards for air interface cover all the frequency range, EPC operates only within the UHF between 860-930 MHz with one standard for 13.56MHz. [17]

Another example can be given with the reader protocol, which the architecture pretends to standardize. The readers to be fully compatible and directly used by the Filtering and Collection module need to comply with the EPCglobal standards.

Despite EPCglobal's attempts to standardize the RFID, the proposed architecture framework narrows the solution scope of the final system when comparing with the technology currently available. If this is positive or not for the evolution of RFID Systems, only time will tell.

6.1.2 Centralized Storage of Information

One of the most criticized problems of the architecture proposed by EPCglobal Inc. is the data storage problem. According to the section 2.4, it is argued that information must reside not on the items (tags) themselves, but instead on a central storage of information designated EPCIS Repository.

EPCglobal created security policies on the EPC Information Services, which allows setting up access permissions to the EPCIS Repositories. Still two options appeared regarding the location of the information. Companies have either the chance to get limited access of information to each other EPCIS Repository or to save it under a centralized EPCIS.

Both solutions have cons and pros. If information would be stored across multiple EPCIS, for example, with manufacturing information on a first, selling information on a second EPCIS and shipping information on a third, where trading partners would have shared access to all the information. Would this spread of information be efficient? Would it be viable? While in some cases there are only a few involved in the supply chain, in other cases it may be dozens of companies.

The second solution is much more dependent from the ONS standard, and it looks like that EPCglobal wants to move this way. With the ONS (see section 2.3.4) standard, each EPC would be linked with only one central EPCIS Repository. This would lead to the creation of centralized storages of information, managed by third-party entities or even organizations created in cooperation between trading partners. Is this solution more viable? Are companies willing to store information about their own business outside their boundaries? What information would organizations have access?

This problem is closely related with the discovery services, a subject being researched by the Eng. Miguel Pardal on his PhD thesis. He proposes a solution based on a blackboard approach, which will help solving it. This topic will be discussed in the upcoming years.

6.1.3 Reader Support

The reader support provided by the EPCglobal standards should be considered as an implementation constraint. The standards define that communication with the readers must be performed via the defined standards (either the LLRP or RP see section 2.3.1), excluding direct communication with any other standards (ISO Standards for example) from the architecture.

Letting the manufacturers decide if they readers will be natively compliant with EPCglobal may look risky, especially if the architecture is still imposing itself in the market. In early stages it posed some problems when getting compatible readers for prototyping and implementations.

6.2 Fosstrak

6.2.1 Fosstrak Evolution

Since this thesis work has been started, some changes occurred to the Fosstrak platform. The most remarkable was the name change, until recently Fosstrak was named Accada, but because of a trademark dispute it was adopted the new name.

There were two major releases during this period. The first around March of 2008 (used for the prototype implementation) included major bug fixes, which made the platform jump a step ahead at the functionality level allowing the practical work of this thesis to be performed. The second one in June 2009, introduced the LLRP, deprecating the old Reader Protocol and allowing a better interaction with a larger number of readers, as well as a simple capture application engine.

The first versions of Fosstrak that I worked with were impossible to keep running stable for long periods because of huge memory leaks, among other major problems. Given time and a couple of releases this was fixed, making the platform stable enough for prototyping.

Currently and according to my knowledge of information systems and RFID, the overall platform can be considered as mature enough for small prototyping attempts. For large-scale implementations I wouldn't recommend the solution without at least extensive prior testing.

6.2.2 Fosstrak Documentation

The provided documentation by Fosstrak of its modules by the time of the development of this thesis was poor. For example, the documentation on how to create a capture application didn't exist. Recent releases have greatly improved the documentation. Now the module documentation can be considered adequate. However, there still is no documentation about how to link the different components to get a working platform. A "module connection user guide" would be a plus from Fosstrak.

6.2.3 Fosstrak Compatible Readers

Another problem that could be noticed in first generations of the platform was the list of supported readers.

The list of Readers supported by custom wrappers in the first generation of Fosstrak was only two readers (details in the list below), both of them expensive and hard to deal without vendor information. On these first releases any EPC Reader Protocol compliant Reader could also be used, but if for the moment these readers are far limited on the market, during the first generation of the Fosstrak releases they were almost nonexistent.

The list of supported readers by the first generation of Fosstrak via wrappers:

- Impinj Speedway
- FEIG ELECTRONICS OBID i-scan ID ISC.MR100/101 (COM) or OBID i-scan ID ISC.LRU1000 (TCP/IP)

The compatibility with other readers (not supporting RP or LLRP) could be done by developing a wrapper using the Fosstrak Hardware Abstraction Layer (HAL). This is a tough task, involving low-level programming and good knowledge of the equipments.

The compatibility list of Fosstrak can be considered quite poor, but it is expected to change soon. With the latest release of the LLRP commander and the capacity of the platform to interact with these readers, the list of compatible devices with Fosstrak may soon be expanded.

6.2.4 Fosstrak integration with supply chains

Another possible fault in the Fosstrak itself, which according to the developers was designed focusing on the tracking and tracing items, is the missing integration of the platform with the business rules and logic. This is the main problem (see 3.2) covered by this thesis and what motivated the development of the proposed framework.

Until recently a good example was the missing capture application module or concepts for simplifying the development and integration task. This situation happened in the past because Fosstrak considered capture applications out of its scope. With a recent release (June 2009) the situation changed, by Fosstrak releasing a first version of a capture application engine. Still this engine does not provided the abstraction level usually used in the SCM applications (item's and packages), what still make complex the capturing of the business logic over structures like EPCIS or ALE Reports.

By the time this thesis was started, Fosstrak only provided a poor example about how to create a simple capture application [39].

The framework was developed to allow the reuse of a generic capture application in multiple contexts, simplifying the integration task. Also the development of a set of concepts closer to “real world” objects easier to assimilate and perform business logic was required.

This may suggest a not so good communication or interaction between development team and the “real world”, since this is something that must be developed in every implementation. More cooperation between these two entities may be needed, to fulfill the organizations requirements without need for ad hoc solutions.

The framework proposed at this work is a contribution for this missing connection.

6.3 Hierarchies of Capture Application

The proposed Subclasses Capture Applications that stereotyped the most common situations where RFID readers are employed along a supply chain is also a contribution of this work.

The proposed objectives (see section 4.3) of providing a high-level understanding of the different possibilities for capturing RFID data and their particular requirements, together with a set

of implementation guidelines, were achieved. The hierarchies of capture applications was taken in consideration in the prototype development, allowing the developer to be aware of the systems' interactions and expected problems that need to be dealt with in the characterized situations.

The last objective of providing generic implementations that handle the most common problems for the described subclasses of capture applications, was not completed. Only the proposed generic capture application 4.3.2 has been fully developed and can be considered a generic implementation and a starting point to develop capture applications. The other subclasses of capture application to the specific situations have been considered, and will be suggested as future work.

On the prototype all the modules, were developed from a generic capture application 4.3.2 that could be used in any reading situation. For the reasons above mentioned it was used directly an implementation focused on a particular situation.

6.4 Proposed Framework

This section evaluates the proposed Framework and presents some of the results.

6.4.1 Comparison with Fosstrak

Some results are provided comparing the use of the Fosstrak provided methods and structures directly against the ones provided by the Framework.

To prove the advantage of using the framework it is possible to compare the required line of code for different capture application functionalities either using the Fosstrak provided structures and methods or using the framework's. More details on the implemented functionality can be found in section 4.4.5.

Functionality	Using Fosstrak natively	Using Framework
Creating an EPCIS Object Event	37 Lines	7 Lines
Inserting a Capture in EPCIS Repository	4 Lines	1 Line
Parsing an ALE Report to get relevant data	28 Lines	0 - Automatic translated into the simplified data structure
Querying the EPCIS about a certain EPC	20 Lines	1 Line
Getting in a single structure all the hierarchy of a package or item	Requires Complex programming	1 line

Table 6.1: Comparison between the use of Framework or Fosstrak natively

In the Fosstrak website [39] there is an example, presented on section 3.2.1, for constructing a basic capture application. Note that basic and hardcoded business logic is being performed in both cases. The code provided in the listing 6.1 performs the same functionality than the presented on the section 3.2.1 but using the framework.

Listing 6.1: Executing a basic capture

```
CaptureEPCIS shipping = new CaptureEPCIS();
//(Read reports into trackable items - Done by framework capture thread)
for (TrackableItem item : captura.getTrackableItems()) {

    shipping.addTrackableItem(item);
}

//Setting the properties
shipping.setBizStep("urn:accada:demo:bizstep:shipping");
shipping.setDisposition("urn:accada:demo:disp:waitingForVehicle");
shipping.setReadPointId("urn:accada:demo:rp:warehouse3:gate1:leitOr1");
shipping.setLocation("Armazem3");

getEpcisCapture().insereCapture(shipped);
```

Using the framework, in the simplest case of a capture application, it is only needed to develop two main methods, the `handleCapture` and the `capture`. Also in this case these methods can deal directly with concepts closer to “real world objects” (`TrackableItem` for example, see section 4.4.1), where is easier to apply business rules than using the ones provided by Fosstrak. This is potentially a major advantage, not only regarding the abstraction level but also by making the code easier to maintain and develop.

With the release of a first version of a capture application engine by Fosstrak itself in June 2009 with a major bug fix that made it workable in August 2009, it makes sense to perform a proper evaluation of both engines to compare common points and the approaches to solve the same problem, enumerating their advantages and disadvantages. Unfortunately due to the late time Fosstrak solution was finally released, it was not possible to perform this assessment in time. It is left to future work.

6.4.2 Advantages

The potential advantages of using the proposed framework when developing any capture application are:

- Allows a faster “prototyping” when implementing the Fosstrak platform simplifying most of the required effort of developing capture applications, especially in track and trace environments. With small effort the framework can be easily reused for other contexts;
- The final developer, who will implement a real case, may focus only on the application context, responsible for processing the business situation. This allows reducing the costs of future implementations;
- Covers one requirement gap existing in Fosstrak;
- Provides a set of concepts closer to the real world objects where is easier to apply business logic than over the ones provided in an EPCIS structure;

- Allows an easier integration with external services, by making clear where the interaction should be performed. The language used (Java) also favors this integration. In the prototype we give the GPS location system example.

It can be concluded that the framework allows simplifying the integration of the platform within an organization's supply chain.

6.4.3 Main Problems

There are some problems on the framework that must be noticed. First that not every component of the EPC Standards or of the ALE is implemented. For example, the QuantityEvent is missing as it was not needed, or the read time of an ALE Report is being neglected for simplification reasons. Also some other improvements can be done to make it more functional.

This will be further described in the future work section.

6.5 Prototype

Given the initial goal of assessing a Fosstrak based solution, the problems found during analysis and development phase, it is imperative to evaluate the final solution.

6.5.1 General Comments

The prototype was developed with the main objective of performing an assessment of the Fosstrak as well as testing the proposed framework within a case scenario.

The prototype, despite these main goals, also showed some of the potential advantages of integrating the supply chain management with RFID technology. To be able to extract more conclusions on this point further assessment must be performed with conditions closer to the real world where a validation could be performed (RFID Readers, Real requirements, environment problems, etc).

To easily incorporate and interpret the read situation into the Capture Application the business process was described in a finite-state-machine (see section 5.2.4, which could be easily reflected as rules in the prototype for the new RFID System. Representing the business as a state machine can be considered a valuable contribution from this thesis.

The developed stateful capture applications for the prototype use the EPCIS Repository to in "real-time" inquire the repository about the status of the read item. This together with the coordination with other systems enables the capture application to interpret the read situation, and incorporate the business context in the EPCIS Event.

The business requirements also stated that a cost-efficient solution was needed where the number of readers should be minimized. The approach mentioned above allowed reducing the number of required readers. A good example is the vehicles that mounted with only one reader can perform multiple business steps interpreting the situation. This is also one of the main achievements of

the developed prototype.

The integration of the developed prototype with a location system demonstrated also the possibility of orchestrating data from multiple sources by consolidating information when making a decision. Other sources of data may be integrated, with the effort depending on the available interfaces to the data.

The use of the framework allowed a simpler development process with much less effort. The separation of a layer for the business rules allowed raising the abstraction level.

The concept of `TrackableItem` (see section 4.4.1, with a direct connotation with real world objects) was a great advantage to the development of the prototype. The aggregation notion introduced in the `TrackableItem`, together with the methods of extracting `AggregationEvents` directly into `TrackableItem` allowed having the idea of "package" simplifying the creation of the business rules.

6.5.2 Requirements Assessment

In this section the initial goals and requirements referred on section 5.1.3, are assessed.

Enable real-time tracking of items along the supply chain, improving visibility It can be seen in the system at real-time as items moves through the supply chain in the RFID enabled points. To improve the visibility more enabled points can be added. This requirement was covered;

All the RFID tagged items should be tracked in the key points of the supply chain The items are being tracked in the implemented locations with the right business context. Not every "key point" of the business was covered or RFID enabled for simplifying. This can be extended by adding more readers;

Ease and speed the processes of delivering and returning items The process of delivery is potentially more efficient as there is no more need for the manual handling of optical readers. The system automatically recognizes lost items and incomplete packages as they pass through the reader range. This requirement is potentially covered, although can just be proven with a real world implementation.

Keep a history of every item on its way along the supply chain The EPCIS events produced at each module, after being recorded are able to provide the history of items. This requirement was covered;

Fight the counterfeit of medicines It can be prevented new RFID tagged medicine from entering the supply chain through `FarmaDistr`, if they are not inserted in the Master Data database. On the current system this situation will generate errors messages. Untagged medicine could still be easily inserted as it will not be noticed by the readers. This requirement was not entirely covered, as just warnings about out of the order items are being issued, but the basis for making it possible is already implemented;

Decrease the number of lost items on their path across the supply chain The number of lost items upon implementing the prototype would likely decrease. Even if items get

lost, it is possible to trace at each point it happened, and take measures to prevent future situations. This requirement is potentially covered;

Speed the process of receiving inbound items This goal has not been covered by the prototype. A new system can be developed that is able read the incoming new items and register them as necessary in the RFID systems;

Increase as possible the efficiency of the system, controlling the costs The use of a finite-state-machine to describe the business process, and its implementation on the capture applications enabled to perform multiple business steps, allowed an efficient cost-benefit solution.

Most of the goals were accomplished by the prototype, although for a real validation, a more concrete implementation of the system with real readers should be attempted. Due to the limitations of getting such readers it was not possible to be done.

6.5.3 Problems

During developing the prototype some problems were found. In the following list they will be described in detail.

- The standards proposed by EPCglobal are not yet finished. One of the main problems was the Master Data (section 2.4.4). To EPCglobal this data should be stored at EPCIS Repository, which schema is focused in the EPCIS events, and without providing methods for capturing Master Data. The only way to do this properly would be using the EPCIS extension mechanism that is not completely free of bugs in Fosstrak. The solution found was to use an external database to work as a Master Data database;
- The resulting work could be more appealing if a real case business was chosen, where a small but real “implementation” of the proposed system could be attempted to prove the simulation results;
- To make the work more reliable, the prototype should be connected with real readers. For the reasons mentioned in section 5.2.2 that was not possible, making the final result not so rich. For example, with real readers the system could be benchmarked regarding the read and missing rates. These practical problems may make the whole solution unviable.

Conclusions

This final chapter presents the main contributions from this work and identifies future work opportunities.

7.1 Conclusion

To better understand the context of some affirmations it may be important to clarify that the time difference between this thesis was finished and the practical work was performed is around 1 year. During this year the development of Fosstrak continued, with a couple of releases including a major one, which may lower the value of some conclusions and contributions.

EPCglobal and their proposed Architecture Framework propose standardizing the RFID in an end-to-end scope solution in the supply chains environments. Their approach at first glance may look positive, a global solution for RFID that enables companies easily to capture, use and exchange RFID data in a standard way. But as seen on section 6.1.1, its standards definitions consider only a part of the available RFID technology and solutions currently available, which may pose later problems and limit somehow the development. Only time will tell if the standards will achieve their goals.

However and despite being a still evolving standard, EPCglobal Architecture Framework is gaining adoption. On the market there are already available the first readers supporting the new reader standards LLRP. For the moment and according to the EPCglobal web site not so much case studies are available to the public, or even articles about the real world implementations. This may show that organizations may want to keep this information in private, or they are still waiting to see the future trend before jumping into implementations.

About the Fosstrak platform and its maturity level, it is possible to comment that a lot changed since the start of this dissertation (September 2007). On its early versions it was hard to keep the provided modules connected and running for more than 10 minutes without memory leaks and crashes, making any real attempt of prototyping useless. Finally, a couple of releases later, things changed and based on the results of this thesis we can assume that the platform is ready

for prototyping. For the moment small production systems may be considered with the proper planning and testing.

The prototype showed a working system, still with some occasional crashes and failures, but some of these issues have already been fixed in later releases.

The most important comment that can be addressed from this thesis results is the difficult integration of Fosstrak with the business logic and rules when employing the technology on the “real world”. Only recently (June 2009) Capture Applications (see section 2.4.1) are covered by the Fosstrak implementation, although not providing a trivial way to process the business logic with concepts like items and packages.

The developed and proposed framework on this thesis covers this point, by easing the integration of the business logic into and EPC Network in particular using Fosstrak, although it should be easily connected with any other standard compliant implementation of EPCglobal.

Regarding the case study, it can be added that most of the requirements were fulfilled, but a real case implementation would be required to fully validate the potentially advantages of integrating RFID with the supply chain management. One of the main achievements was the representation of the business logic in a finite state machine, from where the prototype implementation could be planned.

The developed prototype also helped keep the overall cost of the entire solution viable. The reuse of readers coupled with stateful capture applications enabled to perform multiple business steps by interpreting the situation and context of the capture which can be considered one of the most valuable contributions of this thesis.

The proposed hierarchies of capture application allowed having a high-level understanding of the different read situations and their particular requirements, together with the suggested interactions that have been used as a guideline for the prototype implementation.

The objective of providing generic implementations that handle the most common problems of the complemented capture situations, despite considered, was not achieved. It is proposed as future work, and their development can use this work as a starting point.

Finally the proposed framework, which emerged with the main objective of simplifying the development of capture applications and the integration the business logic into the EPC Network. Providing methods and “wrappers”, which deal with EPCIS and ALE interfaces, and a set of concepts closer to the objects of the real world, allows implementing and integrating the platform inside organizations in an easier way, saving costs and effort of implementations.

With the release of a first version of a capture application engine by Fosstrak itself, it would be imperative to perform a proper evaluation of both engines to compare common points and the approaches to solve the same problem. Unfortunately due to the late time Fosstrak solution was finally released, it invalidated this comparison.

Although framework implements only the data structures of EPC Standards and functionality that were needed during the development of the prototype, as the results showed, the advantages of using the framework are clear (see section 6.4.1).

The results from this thesis work provide a working platform that may be used in the future to finally reach the objective initially proposed for this thesis work, which had more focus on the exchange of information between trading partners.

7.1.1 Contributions

In the following list are the main contributions of this thesis:

- A brief description of Fosstrak software: what it is, how its modules are organized, and which EPC Network standards it implements;
- A Fosstrak assessment in a pharmaceutical supply chain track-and-trace scenario. The prototype development allowed the study of the platform, as well as the usual problems when putting together all its components. A working prototype with all the modules linked with a capture application was achieved;
- A capture application class hierarchy that provides a high-level understanding of the different possibilities for capturing RFID data, as well as implementation guidelines for a specific subclass;
- A framework to facilitate the development of capture applications that convert Filtering and Collection reports into EPCIS Events, providing simplified EPCIS and ALE data structures, and a simplified method for capture and query. The framework was used to apply business rules in the track-and-trace scenario;
- An overview of the generic supply chain management issues that a company faces when using the traditional technology of optical readers and in RFID technology adoption. The prototype implementation provided some indications about how the EPCglobal standards and the Fosstrak implementation can benefit a supply chain.

7.2 Future Work

After the development of this thesis, the following work is proposed:

Regarding the resulting prototype there are still a few items that stayed in the TO-DO list:

- Perform an implementation "on the ground" of the developed prototype connecting it with RFID Readers and by dealing with the related problems like misreads, read rates, etc.
- Continue this thesis work with the scope suggested by the original name of this thesis. This task should reuse the developed prototype and the framework, and integrate multiple entities and organizations sharing RFID data along a supply chain;
- In the prototype, at the Automate System, where the order entry to the RFID enabled supply chain is being done, integrating with the ERP system to automate the flow of information inside organization, should be carried out. When an order arrives through the ERP it would be followed automatically to the Automate RFID system to be picked;

- Create a mechanism of data mining and data management that starting from the recorded data on the EPCIS would realize some sort of “Data Mining”, allowing patterns to be discovered. If all the tracking data is present at the EPCIS, it should be processed to take advantage of that information, and extract not explicit correlations from it;
- Use the developed prototype to create a mechanism of controlling the expiry date of the medicines. In the business logic at every step, it should implement a method to verify the status of every drug. If this expiry date has passed or it is almost over, some alarm should trigger the removal of that item. This problem is present in every supply chain, not only in medicines, but the hazards of a perished medicine are obvious;
- Use the subscribe mechanism provided by EPCIS to subscribe the “Error” states to a system that would take measures of high level to solve the situation, as well to prevent future ones. This new system would receive all the lost and wrongly read items, from all the systems writing to the EPCIS, and inform the closest personnel to the error place to get it solved. This pro-activeness provided by a system would be a major advantage to prevent potential errors;
- Use the developed system to integrate serious anti-counterfeit measures. This is one of the main concerns of the pharmaceutical industry. This can be integrated in the system, at the capture application level.

About the framework and the hierarchy of capture applications:

- Complete the framework of supported EPCIS structures. For the moment it only supports the EPCIS Events (see section 2.4.4) needed by the prototype. The work of completing is important for reuse of the framework in any other project;
- Despite the simplicity of implementing business rules in the framework, it is still needed to write down the business rules in the middle of the code. This can be simplified in tracking systems like the warehouse expedition gate (see section 5.3.3), where creating the business rules could be defined by a standard. The standard would define the proper previous state, the proper next state and store this information, for example, in XML format, or in a database;

The capture application, using this new functionality provided by framework, when starting would read or load this configuration, and auto configure itself, without any need for programming. In this case for every simple tracking system, it would be enough one generic capture application with a configuration for each profile;

- Centralize the solution. The provided framework is optimized for one capture application for every local system. A main improvement would be a central capture application organization-wide, where all the business logic could be centralized;
- Perform an assessment of the proposed framework against the capture application engine recently released by Fosstrak. Compare approaches, enumerating advantages and disadvantages of both.

- Complete the proposed objective of developing generic implementation of capture application for the suggested hierarchies / subclasses, allowing even easier future implementations. Using these generic implementations only configuration would be needed when “implementing” capture applications.

Bibliography

- [1] Zygmunt Mierdorf Antony Burgmans. Global commerce initiative EPC roadmap. 2003.
- [2] C. Atock. Where's my stuff? 24-27. *Manufacturing Engineer, IEEE SYSTEMS JOURNAL*, 2003.
- [3] Beth Bacheldor. Sam's club tells suppliers to tag or pay. *RFID Journal*, 2008. <http://www.rfidjournal.com/article/articleview/3845/1/1/>.
- [4] Harvey Bale. Pharmaceutical counterfeiting: Issues, trends, measurement. *WIPO/OECD Workshop*, 2006.
- [5] Indranil Bose and Raktim Pal. Auto-ID: managing anything, anywhere, anytime in the supply chain. *Commun. ACM*, 48(8):100–106, 2005. <http://doi.acm.org/10.1145/1076211.1076212>.
- [6] Raghu Das. Free idtechex white paper RFID explained. 2005.
- [7] Kin Seong Leong; Mun Leng Ng; Daniel W. Engels. EPC network architecture. *Auto-ID Labs*, page 7, 2004.
- [8] EPCglobal. EPC tag data standards v1.1, 2005.
- [9] EPCglobal. EPCglobal reader protocol (RP) standard v. 1.1, 2006.
- [10] EPCglobal. EPC information services (EPCIS) version 1.0.1 specification, 2007.
- [11] Thomas Erl. The principles of service-orientation. *SOA World Magazine*, page 21, 2006.
- [12] J. McQuivey; M. Feehan. RFID could enhance pharmaceutical marketing. *RFID Journal*, 2005.
- [13] Mark Ferelli. *Mass storage delivers the FedEx tracking system*, volume 6. West World Productions, Inc., Beverly Hills, CA, USA, 2002.
- [14] K. Finkenzeller. *RFID Handbook*. 1999.
- [15] M Fowler. Patterns of enterprise application architecture, 2003.

- [16] Greg Goth. RFID: Not quite prime time, but dawdle at your own risk. *IEEE DISTRIBUTED SYSTEMS ONLINE*, 2005.
- [17] Martina Gerst; Raluca Bunduchi; Ian Graham. Current issues in RFID standardisation. *RFID standardisation Workshop*, 2005.
- [18] Matthew Herper. Tiny chips could combat counterfeit pills. *Forbes*, 2004.
- [19] IBM. *Sun Java System RFID Software Introduction*. SUN Microsystems, <http://docs.sun.com/source/819-1697/sw-overview.html>.
- [20] Service Interfaces, Patrik Ajalin, Unai Briongos, Ari Kesäniemi, and Outi Tarvainen. SAP R/3 integration to RosettaNet processes using web service interfaces. In *Introducing BEA WebLogic Integration 8.1*, pages 57–61, 2004.
- [21] John Karolefski. Preparing for the EPC: new tool enables companies to analyze and plan for implementation of the electronic product code. *Food Logistics*, 2005.
- [22] M. Kärkkäinen. Increasing efficiency in the supply chain for short shelf life goods using rfid tagging. *International Journal of Retail & Distribution Management*, pages 529–536., 31/10/2003.
- [23] Jeremy Landt. Shrouds of time the history of RFID. *AIM Global - The global trade association for automatic identification*, 2001.
- [24] Zaheeruddin Asif; Munir Mandviwalla. *Integrating the supply chain with RFID: A technical and Business Analysis*. 2005.
- [25] Panayiotis Mavrommatis Marios Assiotis. The EPC global network: A formal specification of EPCIS. <http://mavrommatis.googlepages.com/report.pdf>, May 2006.
- [26] J. Maselli. Privacy group focuses on RFID. *RFID Journal*, 2003. <http://rfidjournal.com/article/articleview/547/1/1/>.
- [27] Rob van Kranenburg Matt Ward and Gaynor Backhouse. *RFID: Frequency, standards, adoption and innovation*. JISC Technology and Standards Watch, May 2006.
- [28] Miguel Maya. Sybase customer success story - throttleman. 2008. http://www.sybase.com/files/Success_Stories/Sybase_iAnywhere_Throttleman_SS.pdf.
- [29] Kendra Mayfield. Radio ID tags: Beyond bar codes. <http://www.wired.com>, 2002. www.wired.com/news/print/0,1294,52343,00.html.
- [30] Sun Microsystems. Jsr-000315 javatm servlet 3.0. <http://jcp.org/aboutJava/communityprocess/pfd/jsr315/index.html>.
- [31] CERP Cluster of European RFID Projects. Working paper on future rfid research needs. Technical report, CERP - Cluster of European RFID Projects, 2007. <http://www.rfid-in-action.eu/cerp/cerp-working-paper/working-paper-on-future-rfid-research-needs>.

- [32] V. Daniel Hunt; Albert Puglia; Mike Puglia. *RFID: A Guide to Radio Frequency Identification*. John Wiley & Sons, 2007.
- [33] Christian Floerkemeier; Christof Roduner; and Matthias Lampe. RFID application development with the accada middleware platform. *IEEE SYSTEMS JOURNAL*, page 13, 2007. <http://www.vs.inf.ethz.ch/publ/papers/floerkem-rfidap-2007.pdf>.
- [34] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, chapter RFID Systems and Security and Privacy Implications, pages 454–470. Springer Berlin / Heidelberg, 2002. citeseer.ist.psu.edu/sarma02rfid.html.
- [35] SCOR and Supply Chain Council. Supply-chain operations reference-model overview version 7.0. *Supply-Chain Council*, 2005. <http://www.supply-chain.org/site/scor7booklet.jsp>.
- [36] Tony Seideman. Bar codes sweep the world. *Forbes*, 1993.
- [37] Sandhya Sule; Sapan Shah. Integration in rfid - a patni white paper. 2004.
- [38] Nitin Singh, Kee hung Lai, and T.C.E. Cheng. Intra-organizational perspectives on IT-enabled supply chains. *Commun. ACM*, 50(1):59–65, 2007. <http://doi.acm.org/10.1145/1188913.1188918>.
- [39] Fosstrak Web Site. Fosstrak capturing application - developers guide. <http://www.Fosstrak.org/capturingapp/docs/dev-guide.html>.
- [40] Intermec Web Site. IV7 vehicle mount RFID reader. http://www.intermec.com/products/rfid2_iv7/index.aspx.
- [41] Sybase. Sql anywhere. <http://www.sybase.com/products/databasemanagement/sqlanywhere>.
- [42] Adi Tedjasaputra. Sustainable growth of RFID supply chain markets. *RFID Asia*, 2007. <http://www.rfid-asia.info/2007/07/sustainable-growth-of-rfid-supply-chain.htm>.
- [43] Dale R. Thompson. RFID technical tutorial. *Journal of Computing Sciences in Colleges*, pages 8 – 9, 2006.
- [44] Traub. The epcglobal architecture framework, 2007.
- [45] Craig Asher; Gena Morgan; Richard Swan;Ken Traub. EPCIS (electronic product code information service) frequently asked questions. Technical report, EPCglobal Inc., 2007. http://www.EPCglobalinc.org/standards/epcis/epcis_1_0-faq-20070427.pdf.
- [46] EAN; UCC. EAN international and the uniform code council, note to editors,.
- [47] Samuel Fosso Wamba, Louis A. Lefebvre, and Elisabeth Lefebvre. Enabling intelligent B-to-B ecommerce supply chain management using RFID and the EPC network: a case study in the retail industry. In *ICEC '06: Proceedings of the 8th international conference on Electronic commerce*, pages 281–288, New York, NY, USA, 2006. ACM.

- [48] Seongjin Kim; Mikyeong Moon; Seonghun Kim; Sunmee Yu; Keunhyuk Yeom. RFID business aware framework for business process in the EPC network. *Fifth International Conference on Software Engineering Research, Management and Applications*, page 8, 2007.