Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

RFID	Radio Frequency Identification
RF	Radio Frequency
LF	Low Frequency
HF	High Frequency
UHF	Ultra High Frequency
UWB	Ultra-wide band
IoT	Internet of Things
IT	Information Technology
UCC	Uniform Code Council
EAN	European Article Number
GDSN	Global Data Synchronisation Network
EPC	Electronic Product Code
XML	Extensible Markup Language
WSDL	Web Services Description Language
XSD	XML Schema Definition
EDI	Electronic Data Interchange
DoD	United States Department of Defense
LLRP	Low Level Reader Protocol
F&C	Filter and Collection
ALE	Application Level Events
C1G2	Generation 1 Class 2
MIT	Massachusetts Institute of Technology
IFF	Identify Friend or Foe
U.S.	United States of America
R&D	Research and Development
EM	Electromagnetic
EMF	Electromagnetic Field
EMW	Electromagnetic Wave
SCM	Supply Chain Management
NFC	Near-field communication
EIRP	Effective Isotropic Radiated Power
IC	Integrated Circuit
CW	Continuous Wave
FPGA	Field-programmable Gate Array
TDS	Tag Data Standard
TDT	Tag Data Translation
UUID	Universally Unique Identifier
C1G2	Class 1 Generation 2
WORM	Write Once Read Many
ID	Identifier

PC	Protocol Control bits
Gen2	Generation 2
TID	Tag Identification
CRC	Cyclic Redundancy Check
XPC	Extended Protocol Control
UMI	User Memory Indicator
TID	Tag identifier
XTID	Extended Tag identifier
MDID	Mask Designer Identifier
TMN	Tag Model Number
DSFID	Data Storage Format Identifier
URN	Universal Resource Name
URI	Uniform Resource Identifier
ONS	Object Name Service
GTIN	Global Trade Item Number
SGTIN	Serialised Global Trade Item Number
NID	Namespace Identifier
NSS	Namespace-specific String
SSCC	Serial Shipping Container Code
POS	Point of Sale
RM	Reader Management
DCI	Discovery, Configuration, and Initialization
EPCIS	EPC Information Services
CBV	Core Business Vocabulary
PLC	Programmable Logic Controllers
GPI	General Purpose Input
TCP	Transmission Control Protocol
IP	Internet Protocol
Spec	Specification
ROSpec	Reader Operation Spec
AISpec	Antenna Inventory Spec
SOAP	Simple Object Access Protocol)
API	Application Programming Interface
GLN	Global Location Number
SSCC	Serial Shipping Container Code
HTTP	Hypertext Transfer Protocol
LRSpec	Logical Reader Spec
ECSpec	Event Cycle Spec
ECReport	Event Cycle Report

CAPÍTULO 1

Introduction

1.1 Background and Motivation

The new industrial revolution is centered around data. Yet, the true extension of this reformation might not be fully understood.

From manufacturing to retail, information and communication technologies disrupted the understanding of what could be improved through the supply chain [1]. To grasp the advantages of RFID, we have to understand its place with other technologies and how it can empower the relationship between manufacturing, transportation, logistics and retail.

The introduction of information and communication technologies in the manufacturing process, has been the foundation of Industry 4.0, which some people refer as the fourth industrial revolution [2]. It expanded the cyber-physical systems [2] of the third industrial revolution and introduced Internet of Things (IoT), cloud and cognitive computing. The rise of wireless communications and the massification of data sensing in the manufacturing processes, allowed these technologies to enhance and optimize production processes, logistics operations and marketing strategies.

Adding IoT provided the network infrastructure for data transfer. Cloud computing enabled on-demand availability of Information Technology (IT) resources, fast deployment times and more efficient management strategies. Cognitive computing contributed with computer vision, signal processing and tools to analyze and process large amounts of data for operational optimizations and pattern recognition. These technologies support the modern digital end-to-end system of a business and its manufacturing processes.

UHF RFID improves the developments made in *Industry 4.0* by adding digital visibility of physical assets. But visibility inside a company is opaque to trading partners and vice versa. Visibility both inside and outside of the company is desirable. With EPC that is possible. Inside the company, physical objects can be digitalized, identified and wirelessly tracked. Outside, frontiers between company methods of operations and tools, can be abstracted in a

[1] the sequence of processes involved in the production and distribution of a commodity [1]

[2] a system in which a mechanism is controlled or monitored by computer-based algorithms

common set of interfaces and standards that allows visibility of products outside company bounds and share of information between trading partners. This achieves an end-to-end integration of the complete commercialization process, where data from billions of physical items can be shared through the internet, enabling businesses and consumers to identify, locate and engage each item.

The future of RFID seems promising. In 2016, 96% of apparel retailers had plans to deploy tags on their products [3]. The United States Department of Defense (DoD) requires that all contractors must use UHF RFID identification since 2005 [4].In 2014, Impinj, Intel, Google and Smartrac teamed up to form RAIN RFID, an organization whose mission is to promote the adoption of EPC UHF RFID. UHF RFID can greatly improve transport systems with cost reductions, smaller inventory, faster transportation and routing troubleshooting, lower insurance rates and greater efficiency [5].

Development and standardization of UHF RFID enables a symbiotic relation across the chain, connecting production, logistics, retail and client.

1.2 Scope

This dissertation intents to explore the implementation of smart shelve and software for retail inventory management systems following the global supply chain standards.

1.3 Outline

CAPÍTULO 2

Basic principles of RFID

2.1 Contextualization

The concept of identification using RF dates back to the late 1930s. By this time, the primitive biplanes made of wood and fabric used in Word War I, had evolved to all-metal monoplanes. They were capable of carrying heavy quantities of explosives and travel at hundreds of kilometers per hour, making the conventional method of visual identification of incoming aircrafts obsolete. To counteract this issue, nations invested in research and development that culminated in the development of the microwave radar. By the time of the World War II, both fronts were using radar technologies to detect approaching planes. Still, the problem of identifying allied aircrafts from enemie ones, remained [1]. The German aircraft force, *Luftwaffe*, observed that as pilots rolled their planes, it would change the radio signal reflected back. With this ingeniously simple maneuver, they were capable of discriminate allied aircrafts, being roughly the first RFID passive application. [6]

It was soon later that Watson-Watt, under a British project, developed the first active RFID application to be used in Identify Friend or Foe (IFF) systems. An active RF transmitter was attached to British planes, which on receiving radar signals from base stations, broadcasted a signal identifying the aircraft as allied. [7]

The technology kept advancing through the 1950s and 1960s, mainly in the academic field, but with private companies starting to commercialize anti-theft systems based on 1-bit transponders [2].

In the 1970s the first RFID patents were registered, namely, active tag with rewritable memory and a passive transponder for door lockers. It was during this time that the U.S. National Laboratory of Los Alamos was commissioned, by the Energy Department, to develop a tracking system for nuclear materials. The system revolved around transponders in trucks that would transmit information to readers at the gates of secure facilities. The Agricultural

[1] The attack at Pearl Harbor in 1941 was possible due to a mistaken assignment of incoming Japanese aircraft to an unrelated U.S. bomber flight.

[2] a simple inductive RC resonant circuit that detects transponders in field by changes in the reader coil voltage [8]

Department also requested Los Alamos for an animal tracking system, which was designed in the UHF band with passive transponders, marking the beginning of the UHF RFID technology. These systems were also transposed by the same engineers to automated toll payment systems in the private industry [7], [9], [10].

In the 1980s, the development of the personal computer and technology promises, fueled an upsurge in investment, with private industries wagering in RFID. By the 1990s, deployments of RFID systems had grown significantly. The necessity of compatibility and interaction between proprietary systems rose, being established the first industry standards.

The 2000s followed the same maturation process with slow adoption in the late part of the decade. Despite the interest presented by retail giants like Wal-Mart, and investment by the DoD, compromises created by the RFID industry did not justify the commitment to the technology. The high cost for investment, technical performance difficulties, conflicting standards, security issues and privacy concerns made the investment unappealing [11]. Companies had to strangle Research and Development (R&D) resources due to inexistent tools and complex implementations. Resources required to develop marketing and sales tools, which truly utilize the RFID infrastructure to increase revenue, had to be allocated to deploy RFID systems. Devise economic strategies for suppliers to transit to RFID also seemed to be a problem, since many ordinary companies don't have R&D resources at all to begin with [12]. Another big issue was the standardization of coding schemes in RF tags. The fight for dominance in the UHF RFID market led to multiple coding standards and protocols that made inter-operation among vendor and suppliers infeasible, if not impossible.

In the last decade, despite the advancements in the industry and adoption by big apparel retailers like Zara, Decathlon and Marks & Spencer [13], there are issues that inhibit wide-scale adoption of UHF RFID. These problems will be further discussed in chapter two of this dissertation.

2.2 RFID System

RFID is an identification technology that uses radio waves or electro-magnetic fields to automatically identify physical objects and collect data about them.

Generally, RFID solutions start with a radio device called tag or transponder. The tag is attached to the object that needs to be identified. When such a object is presented in front of an antenna connected to a suitable RFID reader, the tag transmits data to the reader via the reader antenna. The reader then forwards the information through a communication channel to a software application running on a computer.

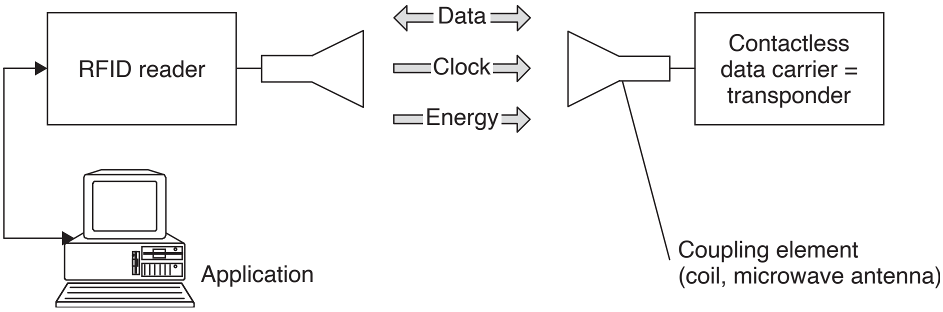


Figura 2.1: RFID system [14]

2.2.1 Components

An RFID system is a collection of components that implement an RFID solution [15]. Different references present different descriptions of RFID systems. Many specify where and which components should be implemented. The fact is, depending on the use case and requirements, these components can be implemented in different physical places or not implemented at all. With the current paradigm of IoT, machine learning and cloud computing, the discrimination of these components is even more difficult to establish. It is naive and intricate to predict what RFID system architecture will be like in the next few years.

The component description of RFID systems can be simplified to three physical blocks: *Tag* (or transponder) which is attached to the physical object to be identified, and is responsible for storing and transmitting appropriate data about the object; *Reader* and *Reader antenna* which interacts with transponders to read and write data; and *Back-end system* which encompasses all kinds of hardware and software components that are separated from the reader and support back-end infrastructures and business logic of a system.

From these, all components described in the literature can be specified and clearly added to one of the blocks. Other prevalent components used in RFID systems are:

Controller is an interface that allows an external entity to communicate with and control the reader's functions and devices connected to it.

Sensor, actuator and annunciator are optional components used for external input and output of the system.

Client Applications is a term for all hardware and software components that are separated from the reader device (e.g. middleware, point-of-sale terminals, etc.)

Communication infrastructure is the wired and/or wireless network, or serial connection infrastructure needed to connect the previously listed components

A run through the specificities of most of the components will be done in the following sections.

2.3 Electromagnetic Concepts

Before presenting the intrinsics around RFID system components, a basic expertise in EM concepts is paramount. Designing well performant systems requires a good understanding of electromagnetism fundaments and its peculiarities.

Like other wireless technologies, RFID uses Electromagnetic Fields (EMFs) or Electromagnetic Waves (EMWs) as interface to transmit information back and forward between reader and transponder. Engineers have to grasp how EMFs and EMWs behave within distance to the reader antenna, how materials in tagged objects can interfere with RF signals, how unaccounted poor RF environments can compromise deployment of RFID systems, to say a few.

An EMF is the phenomenon produced by moving electric charges. It can be described through Maxwell's equations and mathematical abstracted as a combination of an electric field and a magnetic field that interact with each other and their surroundings.

The behavior of the fields changes as the distance from the source increases and are usually defined as two main regions: *near-field* and *far-field*. These regions separate RFID technologies in two very concrete operational groups. In reality, boundaries are not precisely defined, since the regions change their behavior in a progressive manner. The rough discrimination between regions in RFID exist to separate mutual coupling and radiation based technologies.

2.3.1 *Near-field*

Near-field manifests from the electric and magnetic fields near the charges and current that directly produced them. It is the region where phenomena like EM induction and electrostatic occur. This field can be further split in two regions: reactive *near-field* and radiative *near-field*.

Is in the reactive region that *near-field* RFID technologies are defined for. It is the closest to the transmitting antenna and is characterized by non-radiative behaviors. In this region, if the energy is not absorbed by a receiver, self-capacitance and self-inductive effects cause the antenna to store energy very near its surface. When electrons from a nearby conductor are placed in this region, reactive *near-field* energy is transferred to them, resulting in an energy drain on the transmitter by a change in the impedance viewed by the reader [14], [16].

In the radiative *near-field*, i.e. Fresnel region, the back-coupling of the fields becomes out of phase with the antenna signal, and thus cannot efficiently return inductive or capacitive energy from antenna currents or charges. In this region conductive objects, such as metal structures, can behave as antennas by inductively receiving and then "re-radiating" some of the energy [17].

The power of the field differers between field components with the distance (d) from the antenna. The magnetic field strength is proportional to $1/d^3$ and the electric to $1/d^2$. The *near-field* components are quite powerful but usually only suited for close range RFID technologies due to the rapid fall-off with the distance [16].

In the context of RFID technologies, only the reactive zone is considered when referring to the *near-field* region. The radiative zone is *ineffective* and rather unpredictable and usually not accounted referring to RFID.

2.3.2 *Far-field*

The *far-field* region is the region where the EM field behaves as "normal" radiating field, composed of EM waves that propagate outwards, i.e. electromagnetic radiation.

EM waves are created as result of uniform vibrations between an electric field and a magnetic field. In other words, EM waves are composed of oscillating magnetic and electric fields. A change in one of the field components reflects an equal change of the other and one can not exist independently. These waves are detached from any feedback from the moving charges that produced it. Means that, after the waves leave the transmitter, they are completely independent of both transmitter and receiver, as opposed to the phenomena in the *near-field* region.

In this region the radiation amplitude decreases $1/d$ as the distance (d) from the reader antenna increases, being the suitable option for RFID technologies requiring high reading distances (e.g. UHF RFID).

2.3.3 Boundaries

The boundaries between these regions are characterized by locations where the activity of the associated field components are strongest. This does not mean that the other components aren't present, because they are. The transition between regions is progressive.

The *near* and *far fields* are roughly delimited by approximately one full wavelength of the RF wave emitted from a reader antenna. This can be more precisely defined taking in account the transmitting antenna characteristics.

For antennas whose size is comparable to one wavelength or bigger (used in UHF RFID), the *far-field* boundary is delimited by the Fraunhofer distance, radial from the antenna. The Fraunhofer distance is described by $2D^2/\lambda$, where D is the largest dimension of the radiator and λ the wavelength [16]. A representation of the borders between regions can be seen in figure 2.2.

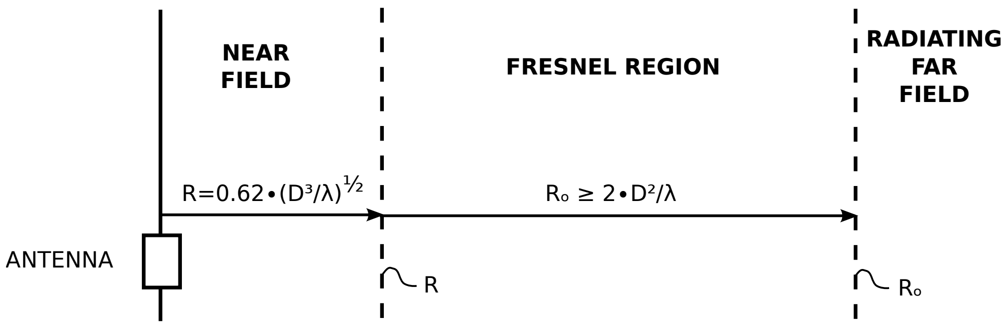


Figura 2.2: EM Field for antennas larger than the wavelength of the radiation it emits [18]

.

For small antennas, shorter than half of the wavelength of the emitting radiation, the *near-field* for RFID applications is usually upper limited by $\lambda/2\pi = 0.159\lambda$ [19]. Small antennas are used in Low Frequency (LF) and HF RFID technologies. For these technologies, which depend on the *near-field* for mutual coupling, the limit of the reactive region is the theoretical limit for the read distance. A delimitation of field regions for small antennas can be seen in figure 2.3.

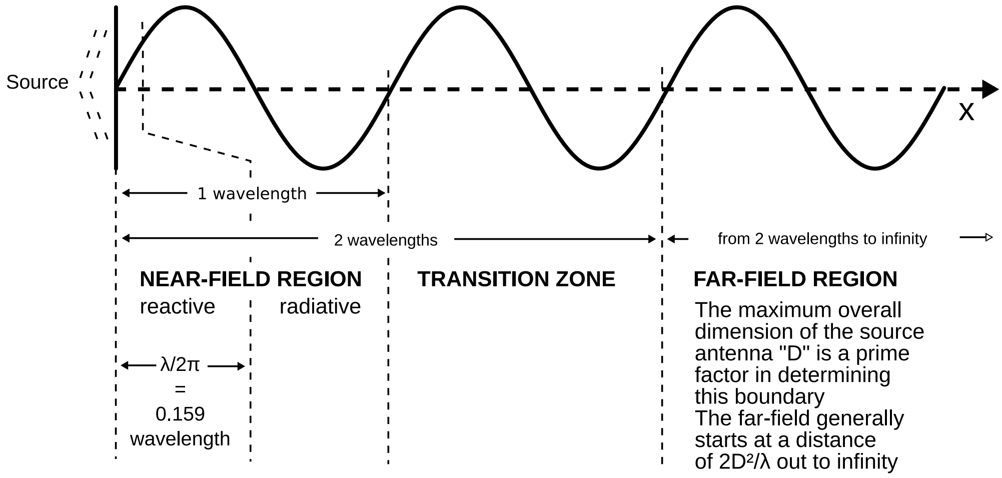


Figura 2.3: Fields for antennas equal to, or shorter than, one-half wavelength of the radiation they emit [20]

2.4 Tag

An RFID tag, also called transponder, is a device that can store and transmit data to a compatible reader in a contactless manner using radio waves or electro-magnetic fields.

Tag cost is the main factor in long term return on investment in RFID systems. It is one of the most important consideration when designing systems. After the initial investment in infrastructure, the expenses are mainly the acquisition of new tags. The reduction of cost per tag is the central focus of manufacturers and what allows the technology to be competitive in the world of Supply Chain Management (SCM).

At its simplest composition [3], a tag contains a microchip and an antenna, ilustrated on figure 2.4. Depending on the technology the tag architecture and operation varies. Tags are characterized by their microchip, power source, memory characteristics and operating frequency. The following subsections will discussed these in detail.

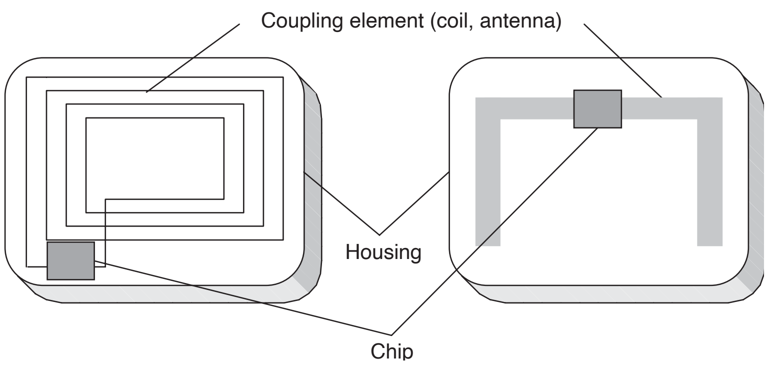


Figura 2.4: Components of a passive tag [14]

[3] an inductive RC resonant tag used in anti-theft systems is considered an RFID device, but is far from the requirements of modern RFID paradigms

2.4.1 Data Quantities

In the contextualization of RFID presented in section 2.1, we discussed the concept of identification and how it can be as simple as World War II IFF systems. An unidentified aircraft presents, upon radar lightening, its identification as being friend or foe. In computing language this can be abstracted as true or false, or 1 or 0. This kind of systems does not exchange more than a bit of information, the smallest unit of information, that can represent only two states. This type of transponders are called 1-bit transponders. Despite the limitations, they are still widely used in surveillance and and anti-theft systems. Apparel retailers and other goods retailers have been using them for almost 6 decades and are still well established nowadays. This dissertation requires the identification of more than 2 types of objects, so the following content will jump over technical explanations of this type of transponder [14].

The type of transponder discussed through out this dissertation are n-bit transponders which use an electronic microchip as the data carrying device. These transponders can transfer data in different methods and are currently used in a variety of standards all over the globe.

The following sections of this chapter will approach concepts that are required to understand systems using n-bit tags, specifically UHF RFID. For a deeper analysis of other RFID systems refer to *The RFID Handbook by Klaus Finkenzeller* [14].

2.4.2 Power Supply

Passive tags

Passive tags are characterized for not having an on-board power source. Instead they use the power emitted from the reader antenna to energize themselves and transmit the stored data to the reader. As such, tag-to-reader communication is always started by a reader, since it needs to energize the tag.

A few examples of state of the art passive UHF tag inlays can be seen in figure 2.5. The simple constitution makes these tags robust, capable of withstanding corrosive chemicals such as acid and high temperatures [4]. These are the type of tags that underwent most technology advancements in the last decade in order to meet performance, compatibility and cost expectations that make mass deployment feasible for companies.

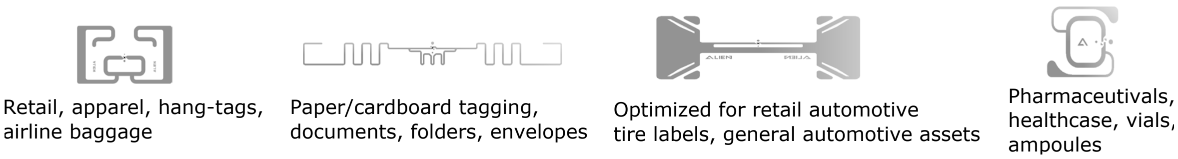


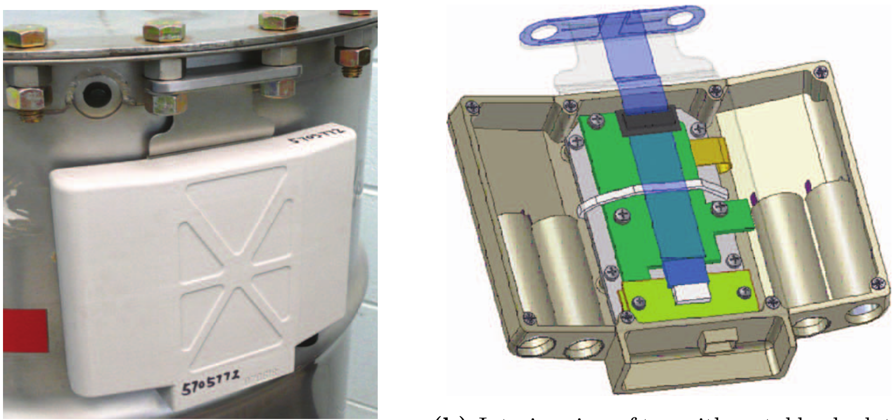
Figura 2.5: Selection of passive UHF tag inlays for distinct applications from *Alien Technology, LLC* 2020 Product Family [22]

[4]The new Impinj M730 and M750 UHF regular tags for item tagging sustains 206°C for 1 minute and can retain data at 125°C for 1 year [21, Tab. 18]

Active tags

Active tags have on-board power source and use it to transmit data to the reader. These type of tags do not need reader's emitted power for data transmission, therefore, in tag-to-reader communication, tags can either communicate first or be interrogated by a reader. Usually, active tags stay in a low-power state (i.e. sleep) in the absence of interrogation by the reader. When a reader wants to interact with a tag, issues a wake up command. The tag transits out the low-power state and resolves the interrogation. The RF environment generated by systems with these type of tags has generally much lower RF noise.

Since the presence of the reader is not necessary for data transmission, an active tag can broadcast its data, like a beacon, to its surrounding even in the absence of a reader. This is what is denominated as *transmitter*. Widely used for IoT applications as "wireless computers" to measure, process and transmit information about sensors, but out the scope of this dissertation.



(a) Tag mounted on Model 9975 drum

(b) Interior view of tag with metal back plate removed

Figura 2.6: U.S. Department of Energy prototype active tag with sensors for nuclear materials management working in $433.9MHz$ UHF ISO band [23]

Semi-passive tags

Semi-passive tags, also called *semi-active* or *battery assisted* tags, have on-board power source, but contrarily to active tags, it is only used for energizing the tag itself, thus, for transmitting its data, a semi-passive tag uses the reader's emitting power.

There are advantages of using these type of tags over passive ones. Semi-passive tags do not use the reader signal to excite itself, so it can be read from further distances compared to passive tags. Because no time is needed to energize a semi-passive tag, it can also be read much faster than a passive one, making them useful for applications were the tag is in the reading zone for a short period of time (e.g. tolls on highways). Finally, this type of tag might also offer better readability in RF-opaque and RF-absorbent materials.

Figura 2.7: *CAEN RFID* A927Z semi-passive UHF logger tag for temperature monitoring of sensitive products during transportation and storage [24]

2.4.3 Coupling

We discussed in section 2.3 how EMF behave throughout the space surrounding a transmitting antenna. Lets now understand how RFID technologies take advantage of Radio Frequency to establish a communication channel between reader and tag.

Coupling in RFID refers to the energy absorbed by a receiving antenna when a transmitting one is operating. It is a fundamental concept in RFID communications and affects several aspects of a system including range, frequency and cost. The coupling method and operating frequency are the defining parameters of technology categorization in RFID, presented in section 2.4.4 and used all over the world as a baseline to describe RFID systems.

There are three types of coupling techniques: inductive, capacitive and backscatter. This dissertation focuses on passive tag systems operating in the *remote-couple* (i.e. range from 1cm to 100cm) and *long-range* (i.e further than 100cm). Capacitive coupling is only exploited for data transmission in *close coupling systems* (i.e. range less than 1cm), therefore this type of coupling technique will not be discussed.

Inductive

Inductive-coupling systems, also known as magnetic-coupling, use the *near-field* magnetic component of the EMF to establish a mutual coupling between reader and tag.

The reader antenna coil generates a strong high frequency EM field. The variation of the magnetic flux excites the cross-section of the tag coil and induces a voltage. Considering the wavelength of the frequency range used ($<$ 135 kHz: 2400m, 13.58MHz: 22.1m) is several times greater than the distance between reader antenna and the transponder, we can treat it just like a simple inductive coupling system, just like a transformer.

This voltage is rectified and supplied to the tag circuitry. The circuitry is actually designed to resonate at the transmission frequency of the reader. The resonance will generate high current in the reader, which produces the required field strength necessary for operation.

To transmit data from tag to reader, inductive systems use load modulation. The microchip changes de load on its coil in relation with the digital data to transmit. When a transponder, resonant at the transmission frequency of the reader, is placed in the reactive *near-field* region, through mutual coupling, energy is drawn by the transponder, being detected by the reader as a voltage drop in the internal resistance of the reader through supply of current to the reader antenna. An illustration of an inductive coupled load modulation RFID system can be seen in figure 2.8.

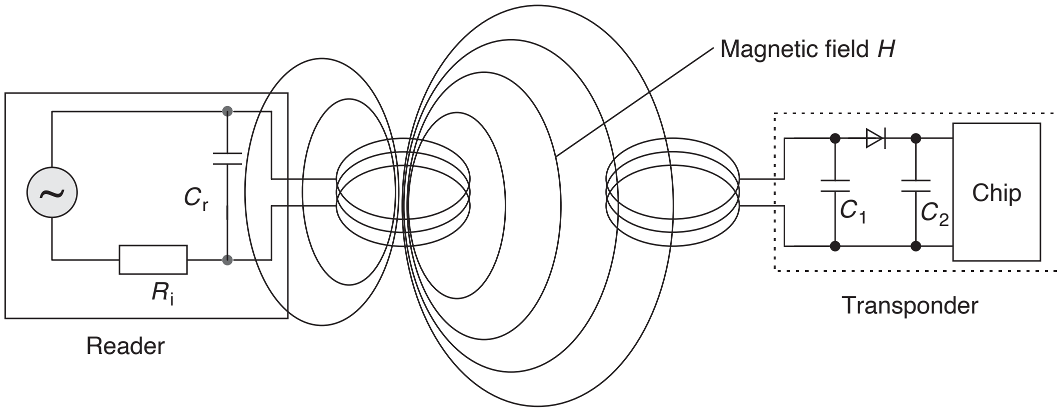


Figura 2.8: Inductive coupled RFID system overview [14]

This modulation method presents a fundamental problem. Due to weak coupling between antennas, there are voltage fluctuations at the antenna of the reader that are much higher than the fluctuations generated by the load modulation in the transceiver. This is a problem that usually appears in LF RFID systems. Detecting this slight voltages requires highly complicated circuitry which is undesirable. Modern systems (e.g. 13.58MHz HF RFID) use what is called: *load modulation with subcarrier*. It uses the side bands created around the operating frequency (f_T) for the transmission of data. Switching the load resistor at a high elementary frequency (f_s), generates two spectral lines at a distance of $\pm f_s$ which contains the modulated signal. An illustration of this can be seen in figure 2.9. The reader can demodulate and retrieve information that is carried in the sidebands of the two subcarrier sidebands, which are themselves created by the modulation of the subcarrier.

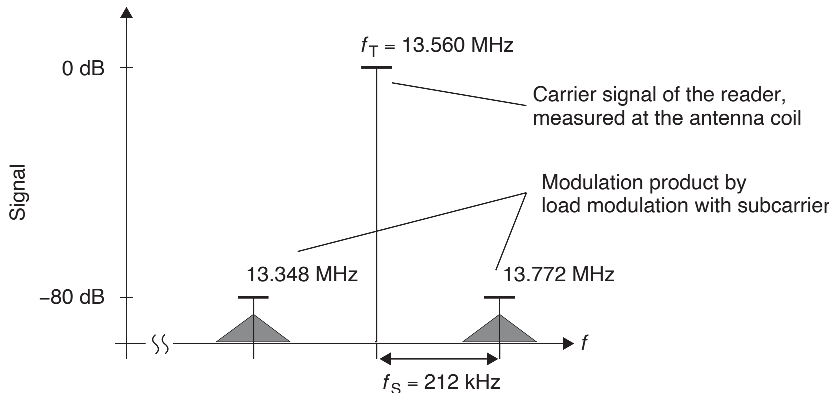


Figura 2.9: Load modulation with subcarriers of a typical HF RFID system [14]

Despite wide scale use and established global standardization, this type of coupling presents a few drawbacks. The reading distance is physically limited by the size of the reader antenna and the reactive *near-field* boundary discussed throughout section 2.3. The shrinkage of tag size is also technically challenging due to the power emissions regulations and the dependence with the variation of the magnetic flux through the cross-section of the tag antenna coil, that excites the tag circuitry. The power consumption is also much higher than the next coupling technique we will look at, making it unsuitable for *long-range systems*.

Backscatter

Backscatter coupling harvests energy from the EMWs, transmitted by the reader antenna, to power the tag circuitry. It is mainly used in *long-range systems* since it is the primarily coupling technique capable of operate in the *far-field* region. To transmit data back to the reader, the tag *reflects* back some of the power as a modulated signal through what is called *modulated reflection cross-section*.

Modulated reflection cross-section operated by the same fundamental principles as the first RFID system invented and presented in section 2.1. It is known from radar technologies that EMWs are reflected by objects with dimensions greater than around half the wavelength of the wave. The efficiency with which an object reflects EMW is described by its *reflection cross-section*. Objects that are in resonance with the wave front that hits them, as is the case for antennas tuned to the appropriate frequency, have a particularly large *reflection cross-section*.

The modulation of the RF signal is created by changing the impedance viewed from the tag antenna, in similar manner to inductive coupling systems. Depicted in figure 2.10, a load resistor, connected in parallel with the antenna, is switched on and off in time with the data stream to be transmitted. The power reflected back changes in amplitude with the changing of impedance of the tag. The reflected signal is picked by the reader antenna, decoupled using a *directional coupler*, and finally demodulated [14], [25].

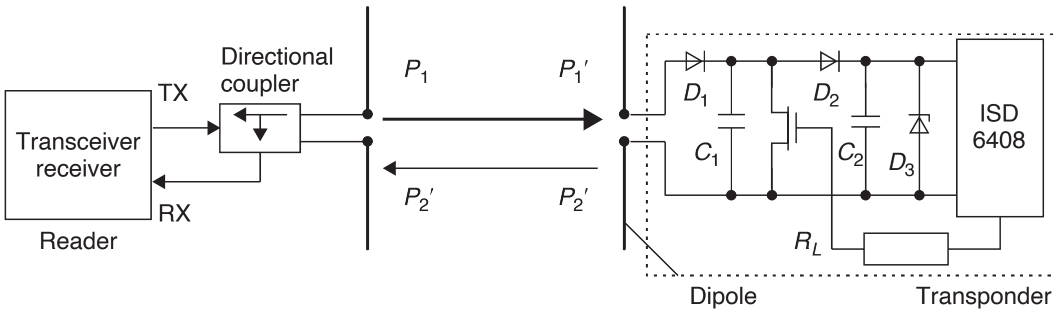


Figura 2.10: Passive backscatter RFID system [14]

2.4.4 RFID Technologies

When categorizing systems, the main aspect that defines RFID technologies is the operating frequency. In the previous sections we noticed a clear relation between operating frequency, through the wavelength, and system specifications like reading distance and data transfer speeds. So, it is no accident that this spontaneous categorization happened. RFID technologies tend to mature in a very clear path, being encompassed under use case specific standards with fixed operating frequencies carefully assigned by regulatory RF authorities in each country. It is indeed a useful classification method to summarize a set of specifications that usually need to be established a priori in RFID system design.

Being so, it is common in RFID slang to name systems by the frequency band they operate in. There are four preeminent frequency groups: LF, HF, UHF and Microwave. Ultra-wide band (UWB) RFID technologies will not be discussed, being in early phases of technological development. Each of the frequency groups presents characteristics inherent to their physical limitations that should be considered when choosing a technology.

Low Frequency (LF)

LF usually operates in the 125KHz to 134KHz frequency range, using passive tags through inductive coupling. Inherently, it has slow read speed and the reading range is limited to around 0.5m. Despite the drawbacks, they have good behavior in metals and liquids and are the most mature technology, probably having the largest installed based system. It is widely used in animal identification (ISO 11784/5 [26], [27], ISO 14223 [28]), access control, logistics and data collection [29], and automotive industry, in car ignition keys, to say a few.

High Frequency (HF)

HF operates mostly around the 13.54MHz frequency band. In same manner as LF, it uses inductive coupling. The higher operating frequency allows faster data read speeds while maintaining semi-decent behavior in metals and liquids. Widely used in smart cards [30], [31] for all kinds of applications and is the basis of Near-field communication (NFC) technology [32], [33]. NFC has been receiving a lot of attention. It is a global communication standard with a mature market (being roughly in any smartphone nowadays). This promoted big companies to invest in it, which began a wide scale manufacturing, resulting in the reduction of tag price ($0.35 - $10.00 per tag) and increase in system deployments.

Ultra High Frequency (UHF)

UHF operates in the band of 315MHz to 433MHz using active tags [34], and 860MHz to 956MHz using passive ones [35]. It is known for the promises it brings to SCM and logistics, being a cheap and affordable technology to wirelessly identify and track every item. It is mainly used in item tracking, but it can be adjusted for all kinds of applications. It uses backscatter coupling to transmit data between reader and tag. These frequencies present some drawbacks: poor behavior on liquid and metals is a big problem with passive tags; the allocation of the RF spectrum differs between countries being a huge problem for global deployments. This thematic will be further discussed throughout this dissertation.

Microwave

Microwave typically operates at 2.45GHz [5] [36], also called *Industry, Scientific, and Medical (ISM)* band, accepted globally. Uses backscatter coupling with passive, semi-passive and active tags. It performs very poorly in the presence of metals and liquids.

RF Materials Performance

RFID technologies, operating in the different frequencies, perform differently in the presence of distinct materials. Material can be: RF-lucent, in that they are translucent to

[5] 5.8GHz ISO standard was withdrawn

EMWs; RF-opaque, where EMF can not penetrate the material; and RF-absorbent, in that it absorbs part of the EMF energy. Concise information on RF properties in different materials is presented in table 2.1.

Tabela 2.1: RF properties in different materials [15]

Material	LF	HF	UHF	Microwave
Clothing	RF-lucent	RF-lucent	RF-lucent	RF-lucent
Dry wood	RF-lucent	RF-lucent	RF-lucent	RF-absorbent
Graphite	RF-lucent	RF-lucent	RF-opaque	RF-opaque
Liquids (some types)	RF-lucent	RF-lucent	RF-absorbent	RF-absorbent
Metals	RF-lucent	RF-lucent	RF-opaque	RF-opaque
Motor oil	RF-lucent	RF-lucent	RF-lucent	RF-lucent
Paper products	RF-lucent	RF-lucent	RF-lucent	RF-lucent
Plastics (some types)	RF-lucent	RF-lucent	RF-lucent	RF-lucent
Shampoo	RF-lucent	RF-lucent	RF-absorbent	RF-absorbent
Water	RF-lucent	RF-lucent	RF-absorbent	RF-absorbent
Wet wood	RF-lucent	RF-lucent	RF-absorbent	RF-absorbent

There is a big market around single purpose tags for materials with poor RF performance. An informed decision must be made when selecting tags for systems dealing with materials portraying these characteristics.

2.5 Reader

A reader is a device that can read from and write data to compatible RFID tags. It is responsible for sending and receiving RF signals to and from tags. If we strip down all the features modern readers offer, we can isolate three main components: RF front-end, processor, communication interface.

The RF front-end interfaces with the reader antenna. It is responsible for transmitting power and clock cycle via the antenna to tags in the reading zone, and demodulate RF signals received from tags to be further handled by the processor. Readers must adhere to RF modulation requirements and Effective Isotropic Radiated Power (EIRP) regulations if they want to be approved and commercialized. This fact promotes the implementation of the front-end by a dedicated transceiver microchip and complementary electronics, which is usually chosen. There are also efforts in implementing the RF front-end in Field-programmable Gate Arrays (FPGAs), which might change this paradigm in the next few years [37].

The processor is responsible for all logic necessary in the reader. It often comes with other complementary components like memory and dedicated Integrated Circuits (ICs) for specific tasks. Readers differ in architecture design between manufacturers, but the processor must at least implement the reader protocol to communicate with compatible tags. This includes decoding and error checking of the analog signals from the RF front-end, a controller interface to allow external entities to communicate with it, to transfer stored data, issue commands, accept commands and control the reader functions.

Communication interface is the component that provides communication instruction to a reader which allows external entities to interact with it. It usually comes inside microprocessors and microcontrollers ICs. It is usually seen in commercial readers as serial communication interface or ethernet.

Commercial readers come in a variety of types which are suited for different applications. The most common are stationary readers. Also called, *fixed* readers, they are what the name implies, used for making fixed reading zones. They also can be mounted on forklifts and inside of trucks. Other types are handheld readers, for a portable reading device and RFID printers which are used to print smart labels.

2.6 Reader Antenna

A reader communicates with tags through its antenna. Antennas are responsible for converting guided Continuous Waves (CWs) in radiating EMW and vice-versa. It will only be covered radiating antenna theory for it is what is necessary to follow this dissertation.

In handheld readers, the antenna is generally integrated into the device. In stationary readers is most commonly found as a separate device attached to an antenna port of the reader by means of a cable. There are different groups of antenna designs: antennas composed of a wire, which encompass the dipole, monopole and loop antennas; constituted of an opening, like the horn and cavity backed slot antennas ; and printed circuit antennas like path antennas, commonly used in UHF RFID.

Evaluating antennas for RFID systems can be done through the parameters specified by the manufacturer, a few rules of thumb and engineering sensibility. At UHF we can control the direction of waves and other parameters, and thus improve the readability of tags.

2.6.1 Radiation pattern

An important parameter in antenna selection is the radiation pattern. Radiation pattern is the mathematical description or representation of the radiation properties of the antenna as a function of space coordinates. The radiation pattern illustrates the regions where the antenna's energy is most effective. In RFID is usually called the footprint of the antenna and determines the read zone. Manufacturers usually provide the radiation patterns for RFID antennas in the datasheet.

Different antenna designs radiate with different patterns. These radiation patterns are classified in three main classes: isotropic, directional and omnidirectional. For UHF RFID, directional antennas are really useful since it is possible to maximize directivity, that is, how well the radiation emitted is concentrated in a single direction. This increases reading distances and defines boundaries in reading zones. UHF RFID directional antennas are commonly patch antennas, also called *microstrip* or *planar antennas.* They consist of rectangular metal foil or a plate mounted on a substrate such as Teflon or FR-4 (i.e. woven fiberglass cloth with an epoxy resin binder).

Is also important, when designing RFID systems, to have notion of the deformations that occur in the radiation patterns. *Multi-path* phenomena - the reflection of EMW in RF-opaque objects which causes waves to be scattered and arrive back at the reader in different paths and times - results in *phase* shifts leading to constructive and destructive interference. This creates protrusions and dead zones that will inevitably exist in most UHF RFID and microwave systems deployments.

2.6.2 Polarization

Antennas radiate EMW into surrounding space. Polarization is the direction of oscillation of the EMW and has a great importance in tag readability.

EMWs behave similarly to a wave of water. When a wave from a source, like a boat, reaches a buoy, the buoy moves mostly up and down. Analogously, EMW moves electrons in the plane perpendicular to the direction of propagation. The direction in which the field points, determines the polarization of the EMW. Unlike water waves, EMWs are not influenced by gravity, so the electric field can point in any direction in the plane perpendicular to the direction of propagation, and so have different polarizations [6].

The main types of polarization are known as linear and circular. Both types derive from the elliptical polarization, being linear and circular special cases of such.

Linear polarization

Linear polarized antenna emanate EMWs in a linear pattern, that is, emits EMW with only one field of energy.

These antennas have a narrower radiation beam with better directivity compared to circular polarized antennas. This results in longer reading distances and well-defined reading regions, which are essential to good RFID system design.

These advantages can sometimes fall short to the disadvantages. Linear antennas are sensitive to tag orientation with respect to its polarization direction. RFID tag antennas are usually dipoles constituted of narrow metal traces aligned in one direction. If the electric field is directed along the traces, it can push electrons back and forth inducing the voltage necessary to energize the tag IC. If the electric field is directed perpendicular to the trace axis, it moves electrons across the diameter of the trance, producing negligible current, insufficient to power the tag IC. Linear antennas are generally used in applications where tag orientation is fixed and predictable, or in conjunction with other complementary linear antennas.

Circular polarization

Circular polarized antennas solve, to certain degree, the orientation dependence problem with detriment of reading distance and a wider radiation beam. These antennas produce two energy fields that are equal in amplitude and magnitude, with a phase difference of 90° between them. This results in the electric field vector to be seen as helix like trace, a circular motion propagating through space.

Because the nature of circular polarized antennas, they are to a certain extent, unaffected by tag orientation. A circularly polarized EMW interacts with the linear antenna of a tag, tilted at any angle within the plane perpendicular to the axis of propagation, but in every case only half the transmitted power can be received. Good antenna design in tags can modestly improve energy harvesting performance, e.g. incorporate two dipole antennas on the tag directed orthogonally to one another or physically larger bow-tie antenna designs.

These antennas are widely used in passive UHF RFID, specially in RF environment with high degree of RF reflectance, like due to presence of metals.

In the next chapter, I will introduce GS1 and their EPCGlobal Architecture Framework, used in SCM and logistics across the world and their close realtion with UHF RFID.

CAPÍTULO 3

EPCglobal Architecture Framework

3.1 Context

Advances in UHF RFID gathered a lot of attention and investment in the beginning of the decade. The technology promised, for years, a disruption in the SCM which was never delivered.

Item level identification allows companies to capture product lifecycle information at remarkable levels of detail. RFID readers placed through out the supply chain can automatically capture information about tagged objects while they move from manufactured to consumer. An infrastructure that bridges the gap between the physical and the digital world, providing
would be essential.
real-time information about current supply chain operations. Furthermore, the instant share of information between intervening companies increases supply chain visibility, resulting in reduced uncertainty in operational and tactical supply chain planning. Stock levels could be precisely controlled and shared with trading partners, which in turn reduces inventory costs and optimizes intra-company operations [38], [39].

its
It was an utopia ahead of is time. The technology was not mature enough, the return on investment was not appealing and cloud computing had just started to get traction.

This did not stop the conceptualization and development of architectures capable of delivering the promises we were hoping for. The architecture that stood relevant through out these uncertain times is the *EPCglobal Architecture Framework*. It was created and maintained by GS1, a non profit organization tasked with developing global standards for business communications. The organization has the experience, resources and influence to make this utopia a reality. The architecture shows on paper an appealing concept of a network capable of doing amazing thing for the SCM without restricting businesses IT architectures.

3.1.1 Standardization Efforts

Standardization of UHF RFID for item level tagging and supply chain, by organizations like GS1, provided a common language to identify, capture and share supply chain data, ensuring important information is accessible, accurate and easy to understand [40].

that

USA

The first prominent adoption was by the DoD with a policy released on July 30th 2004. The policy stated that contracts issued for material delivery would require the use of UHF tags. The policy was later extended to all commodities and commodities pallets shipped to any DoD facility [4], [41].

In 2014, Impinj, Intel, Google and Smartrac, joined forces to create the RAIN RFID alliance after the ratification of GS1's UHF RFID Generation 2 version 2 standard in November of 2013. The alliance promotes the universal adoption of GS1's Gen2 UHF RFID technologies and cloud computing, where RFID-based data can be stored, managed and shared via the Internet [42]. The alliance fortified the adoption of GS1's standards and traced a common path for the ~~the~~ industry to progress [43].

On October 11th 2018 the European Commission published their positive implementation of the upper band for European countries [44]. It extended the power levels to $2W$ in the lower band and added the requested global band from 915MHz to 921MHz with power levels up to $4W$. This was the biggest effort by the European Commission to establish a global standardized frequency band for UHF RFID supply chain applications.

3.1.2 Current Problems

There are still RFID tags that do not conform with the international standards, often presenting proprietary formats and even encoding errors. These closed practices and struggle for market supremacy around UHF RFID creates a problematic situation that prevents conformity through the global logistics market.

Even in global standards, the adoption depends on the company and field of business. Usually one identification standard is already being used, and the migration cost for supporting multi-code integration can be high.

The information around global standards is also limited and hard to get through. It is divided in multiple specifications, identified with number notation and codification nomenclature. The ISO standards, in specific, are closed and have to be payed before even see ~~it's~~ contents. its These specifications are extensive and don't provide newcomers a good experience. Companies planning to implement UHF RFID systems following legitimate global standardization resort to consultants who have a deep knowledge on the standards complexity.

The closed mentality in the area slowed the industry progression. In comparison with the cloud and web industries, where experience and software is shared and open-sourced, UHF RFID tends to keep everything closed [45]. The existent freely available software is old, outdated and out of maintenance. Experience from real-world implementations is unavailable and it is not shared, making the industry prone to committing recurrent mistakes. This results in high investments in time and money on engineering resources that could be shared among industry leaders.

The positive implementation of the upper band frequency in Europe for global UHF RFID supply chain applications is also dependent on the acceptance by each European members. In particular, Germany and Netherlands are not accepting it [46]. The conflict with existing adopted bands in the countries makes a global homogeneous system a challenge that will need time to be established.

3.1.3 GS1 and EPCglobal

GS1 is a nonprofit organization dedicated to the development and implementation of standards for global supply chain solutions. The institution mission is to manage the GS1 System of Standards, create open, global, multi-sector standards fostering good business practices.

GS1 established itself in 2005 from the European Article Number (EAN) International, Uniform Code Council (UCC) and other local organizations from the United States [47]. The organization took under its umbrella the former EAN-UCC roles subsuming their technologies. From those technologies, it is worth mentioning: the barcode identification system (from EAN), Extensible Markup Language (XML) standards, Electronic Data Interchange (EDI) transaction sets and supply chain solutions [15].

The new GS1 organization then adopted much more ambitious projects, developing global standards and services for business communication. From those efforts resulted the network for the synchronization of master data Global Data Synchronisation Network (GDSN), the EPC integration for RFID, traceability and the upstream integration of the consumer goods industry suppliers and *EPCglobal Network*.

For RFID Technology to become viable in practice, an infrastructure must exist for processing and communicating EPC data. In meeting the goal of creating a common infrastructure, Massachusetts Institute of Technology (MIT) announced Auto-ID Release 1.0 in October 2003. At the same time, MIT entered into an exclusive licensing agreement with GS1. In turn, GS1 established a new division called EPCglobal to implement Release 1.0 and to conduct further development based on industry input. This put forth an initial set of standards that formed the infrastructure for EPC data. Later, Auto-ID Release 1.0 became the starting-point for the *EPCglobal Network* and Architecture Framework [48].

3.2 Overview

EPCglobal Architecture Framework is a collection of interrelated hardware, software, and data standards that interoperate with shared network services [49]. These services are referred to as *EPCglobal Network*, a computer network used to share EPC data between trading partners. They are operated by GS1, its delegates and others to provide automatic, real-time identification and data sharing of items both within and outside of a company [15].

The framework defines information systems, interface standards and data models. This approach frees the market of IT systems to create custom business solutions. Manufacturing can have their custom business logic closed, and expose production state information to the clients through the *EPCglobal Network* with the EPCglobal interface standards.

The existence of these standards promote not only the global adoption of EPC, but also though the exchange of information between business partners. Even ~~doe~~ the network was design ed primarily for RFID EPC data sharing, the network does not exclusively run~~s~~ on RFID data carriers. The Network can also be fed EPC data through data carriers like 1D and 2D barcodes.

The interoperability with the barcode was one of the most important considerations during the planning of the network [50].

3.2.1 Activities

The architecture defines three core activities, all of which have a group of standards and guidelines within the *EPCglobal Architecture Framework*: *Physical Object Exchange*, *Infrastructure for Data Capture* and *Data Sharing*, which I will be briefly brainstorm bellow. These activities are helpful in understanding the organization and scope of the framework but should not be interpreted as extremely rigid [49].

Physical Object Exchange

Identify individual physical objects - products, cases, loads, assets, return items, among others - so they can be tracked individually. Entities in the supply chain exchange physical objects identified with EPCs. Exchange operations can be such as shipping, receiving goods, and so on. For many End Users, the physical objects are trade goods, but this could not be the case. There are many other use cases, like library or asset management applications [51] that differ from the supply chain trade goods model, but still involve unique identification and tagging of objects. The architecture must be designed to ensure that when one *End User* delivers a physical object to another end user, the latter will be able to determine the EPC of the physical object and interpret it properly [49].

Infrastructure for Data Capture

Capture data about the movement of physical assets to create supply chain visibility. In order to gather EPC data, each *End User* carries out operations within its environment. Those can be: the creation of EPCs for new objects, follow the movements of objects by sensing their EPCs; and gather that information into ~~systems of recor~~d within the organization [49].
recording systems

Data Sharing

Exchange data with IT applications and trading partners, to turn visibility into information and action. *End Users* benefit from the *EPCglobal Architecture Framework* by sharing data with each other, increasing the visibility they have regarding movement of physical objects through the supply chain. The *EPCglobal Architecture Framework* standards provide a means for *End Users* to share data about EPCs within defined user groups or the general public.

3.2.2 Brief

can be difficult to analyse and use.

The *EPCGlobal Architecture Framework* ~~can be taxing to get to know~~. It is separated in multiple specifications and shows inconsistencies between documents, making the introduction to the framework fairly hard. Throughout this section I will introduce the baseline implementation of the *EPCGlobal Architecture*, introducing each intervening member and their role within the framework. This brief searches to provide a technical contextualization of the architecture for the next sections, where it will be thoroughly explained.

The *Architecture Framework* provides detailed specifications for three technologies developed by EPCGlobal: the Electronic Product Code (EPC), components of the *EPCGlobal Network™* and standards for RFID readers and tags.

The EPC is a unique ID used to identify an object. This EPC can be stored in different physical formats, including in RFID tags. These EPC enabled RFID tags can be attached to objects, and inspected throughout the supply chain by a network connected by services and readers. The simplest implementation of the *EPCGlobal Framework* can be structured like shown in figure 3.1. I will start~~ing~~ on the bottom and go up in the architecture structure.

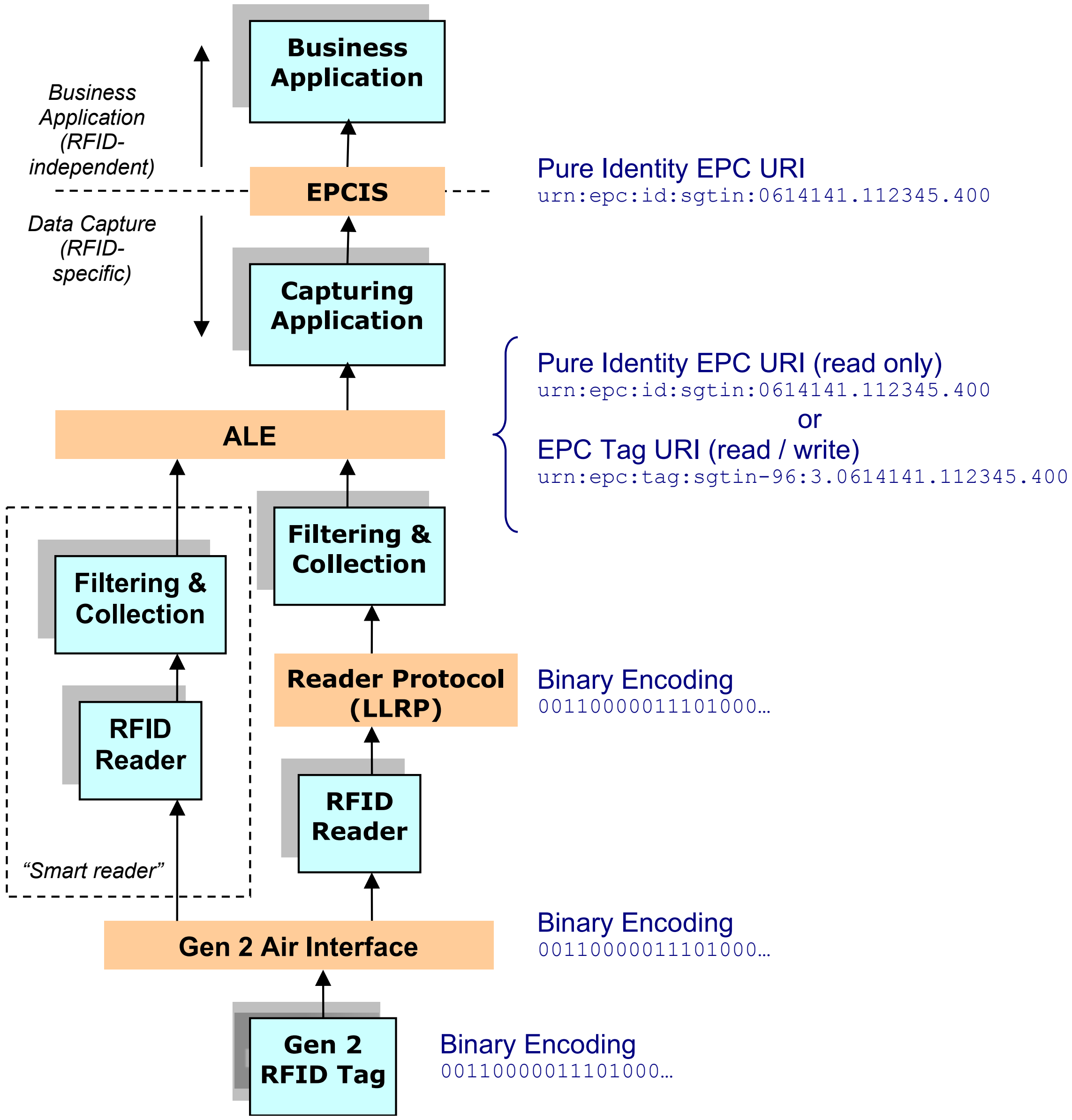


Figura 3.1: *EPCGlobal Architecture Framework* baseline example with EPC representations used at each level [52]

The Gen2 RFID Tag is the latest generation of the UHF tag standard design by EPCGlobal. Gen2 tags communicate with readers through the Gen2 Air Interface. The physical memory map of Gen2 tags is standardized in the Tag Data Standard (TDS) [52], in conjunction with the binary encoding of EPCs. The RF communication link, signaling and commands between Gen2 tags and readers is defined in the Gen2 UHF RFID Standard [53].

Readers communicate with clients through the LLRP standard [54]. Gen2 readers can generate large amounts of network traffic and data, which needs to be processed. For that, *EPCGlobal Architecture Framework* defines Filter and Collection (F&C) middleware, to translate binary EPC into human readable format, and filter/aggregate data. F&C can be implemented in the same physical device as the reader, being called a “smart reader”.

Middlewares then send the processed data to Capture Applications, where EPC data is contextualized with business logic. Middlewares send this data through an ALE interface. The ALE interface allows Capture applications to register and subscribe to their interested patterns? EPC patters and request strategies in ~~an~~ high-level declarative way. The interface is defined in its two documents [55] and [56].

Capture Applications send the contextualized data to Business Applications or data repositories. This data is sent following strict rules to ensure data is created and shared in a conformable and standardized way. The *EPCGlobal Architecture Framework* defines the EPCIS [57] and CBV [58] Standards, containing data models, vocabulary and communication interfaces required to create and share EPC related data with trading partners.

More standards and information paradigms exist in the *EPCGlobal Architecture Framework*, namely in the *EPCGlobal Network™* group of standards. The standards under the authority of *EPCGlobal Architecture Framework* can be seen in table 3.1. Relevant standards and congruous paradigms not in the practical scope of this dissertation: Reader Management (RM) standard to manage readers (not strongly adopted by the community); Discovery, Configuration, and Initialization (DCI); Tag Data Translation (TDT) for validation and translation of EPC identifiers; Object Name Service (ONS) a protocol to discover information about a product and related services from the EPC; and Global Data Synchronisation Network (GDSN), the network of interoperable data pools to share master data with their trading partners in a standard method.

Activity	**Standard**	**Status**
Object Exchange	*UHF Gen 2 Tag Air Interface*	Ratified Jul 2018 v2.1.0 [53]
	HF Class 1 Tag Air Interface	Ratified Sep 2011 v2.0.3 [59]
Data Capture Infrastructure	*EPC Tag Data Standard (TDS)*	Ratified Nov 2019 v1.13 [52]
	Low Level Reader Protocol (LLRP)	Ratified Oct 2010 v1.1 [54]
	Reader Management (RM)	Ratified May 2007 v1.0.1 [60]
	Discovery, Configuration, and Initialization (DCI) for Reader Operations	Ratified Jun 2009 v1.0 [61]
	Tag Data Translation (TDT)	Ratified Oct 2011 v.1.6 [62]
	Application Level Events (TDS)	Ratified March 2009 v1.1.1 [55], [56]
	EPCIS Capture Interface	EPCIS Ratified
Data Sharing	*EPCIS Data Standard* *Core Business Vocabulary (CBV)*	EPCIS Ratified Sep 2016 v1.2 [63] CBV Ratified Oct 2017 v1.2.2 [58]
	EPCIS Query Interface	EPCIS Ratified Sep 2016 v1.2 [63]
	Pedigree Standard	Ratified Jan 2007 v1.0 [64]
	EPCglobal Certificate Profile	Ratified Jun 2010 v2.0 [65]
	Object Name Service (ONS)	Ratified Jan 2013 v2.0.1 [66]
	Global Data Synchronisation Network (GDSN)	Ratified Nov 2020 v3.1.14 [67]
	Lightweigh Verificationt Messaging Standard	Ratified Jul 2019 v1.1 [68]
	GS1 EDI [69]: XML standards	Ratified Nov 2019 v3.4.1 [70]
	Discovery Services	In Development

Tabela 3.1: Standards within the EPCglobal Architecture Framework

3.3 Generation 2 UHF Tag

Gen2 RFID tags are passive RFID tags that conform to the EPC Class-1 Generation-2 UHF RFID Standard for communications in the 860 MHz to 960 MHz frequency band, the ISO/IEC 18000-6 standard (Type C), or related standards currently under development.

The EPC Generation 2 Air Interface Protocol defines the physical and logical requirements for RFID readers and passive tags, operating in the UHF band, to communicate with each others [53]. In the context of this dissertation, we are particularly interested in Class 1 UHF tags. Class 1 Generation 2 (C1G2) tags are characterized for operating in the UHF or HF [59] bands, in Write Once Read Many (WORM) type RFID systems like SCM and logistics. They are cheap, robust, support cryptographic authentication for anti-counterfeiting, functions for traceability protection and most important, are conformable with the ISO/IEC 18000-6 Type C air protocol, conjugating the two most prominent standards in UHF conformable communication standard of operation.

It is important to address the C1G2 and ISO/IEC 18000-6 conformability. Despite the interrogation commands and logical memory map being the same, the standards differ in the data encoding. GS1 and ISO have different formats and encoding rules to represent Identifiers (IDs). Systems that want to support both encoding types have to implement interoperability between them [71]. In this dissertation we focus on the *EPCGlobal Architecture Framework*. ISO encoding will not be covered.

3.3.1 Memory

Figure 3.2 depicts the logical memory layout in EPC Gen2 tags. It has four banks of non-volatile memory: *Reserved Memory*, *EPC Memory*, *TID Memory* and *User Memory*. Banks can be accessed by multiples of 16 bits words.

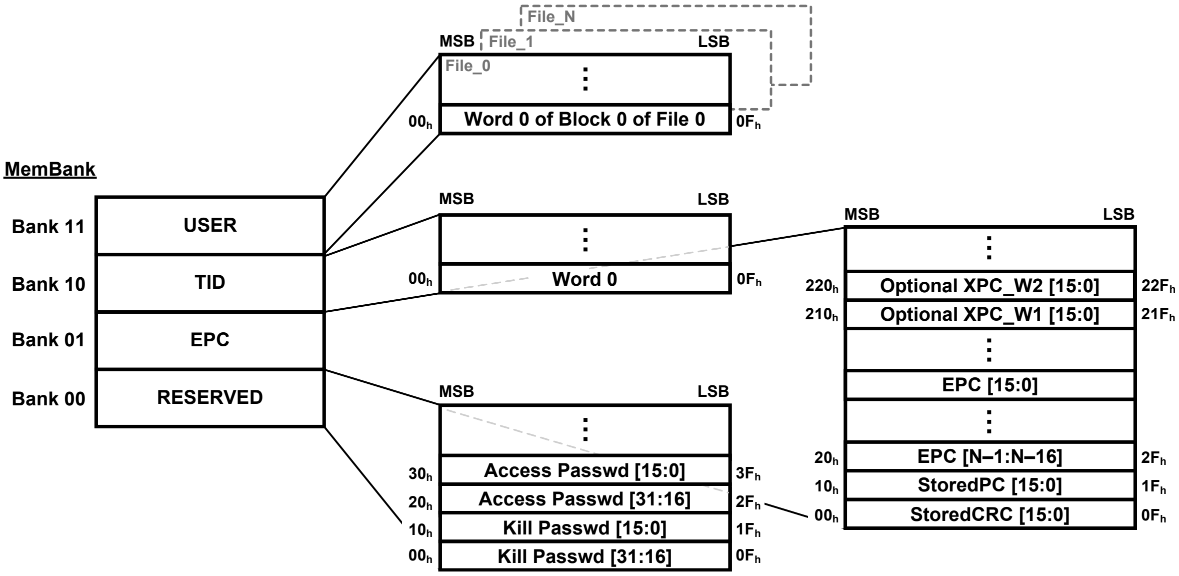


Figura 3.2: Logical memory map of EPCGlobal Gen2 UHF tags [53]

Reserved Memory

Reserved Memory (Bank 00) holds the Access and Kill passwords, if implemented by the manufacturer of the Tag. The *Kill password* is a 32-bit value used in the *Kill Command* to render a tag non-responsive thereafter. The *Kill Command* only executes if the password has been set to a value different from the default all-zero password. The *Access password* is a 32-bit value which allows a tag to transit~~ion~~ in to the Secure state. In the Secure state all Access commands can be executed, including writing to locked blocks. The default password is all zeros and must be changed if access protection is desirable [53], [72].

EPC Memory

The EPC Memory (Bank 01) contains a 16-bit Cyclic Redundancy Check (CRC) for error correction, a 16-bit PC and an EPC.

EPC binary encoding will be further discussed in section 3.4.4, it is the ID in which the *EPCGlobal Framework* revolves around to identify every item in the world. The EPC is by design extensible, being that, depending on the application requirements, it can be between 96 bits and the maximum EPC size supported by the tag (some tags can have EPC memory up to 496 bits). Important to refer, tag cost rises with the amount of EPC bits. Companies planning on implementing EPC must devise a serialization plan where a estimate of serial

Application	MSB															LSB
	10_h	11_h	12_h	13_h	14_h	15_h	16_h	17_h	18_h	19_h	$1A_h$	$1B_h$	$1C_h$	$1D_h$	$1E_h$	$1F_h$
GS1 EPCglobal	**L4**	**L3**	**L2**	**L1**	**L0**	**UMI**	**XI**	**T=**0	RFU							
Non-GS1 EPCglobal	**L4**	**L3**	**L2**	**L1**	**L0**	**UMI**	**XI**	**T=**1	AFI as defined in ISO/IEC 15961							

Tabela 3.2: PC assignments from EPC UHF Gen2 Air Interface Protocol [53]

numbers and unique products is done. This will influence the requirements in EPC memory and consequently tag price. The encoding of EPC is not defined in the Gen2 air protocol standard, but in the *EPC Tag Data Standard* which will be further discussed with detail ~~on~~ in section 3.4.

The standard also defines Extended Protocol Control (XPC) words, XPC_W1 and XPC_W2 respectively. These words are optional and usually not implemented by tag manufacturers. In fact, the tags used in this dissertation - with *Impinj's Monza* chips - are widely used in all kinds of tag applications around the globe, and don't implement XPC words. These words contain bit indicators for settings like hazardous materials and intractability to say a few.

The PC word contains metadata used to interpret the EPC. A more detailed representation of the PC can be seen in table 3.2. From $10_h - 14_h$, 4 bits with the length of the EPC/ID in words, at 15_h a toggle bit called User Memory Indicator (UMI) which indicates if *User Bank* 11 has any data in it, and at 16_h a toggle bit to indicate if the tag has extended PC. To distinguish ~~an~~ GS1 from an ISO standard ID, the most reasonable way is to look at bit 17_h, which contains a toggle bit to indicate whether the ID is GS1 or non-GS1 family. From bit 18_h to $1F_h$ is reserved for future use under the *EPCGlobal* specification. Under some circumstances GS1 EPCglobal may permit other standard body or organization to use one or more of these RFU values for standardization purposes. Also to note, on ISO 18000-6, these bits are used for ID's supplementary meta data called Application Family Identifier.

TID Memory

TID Memory (Bank 10) holds chip manufacture~~d~~r information and tag capability indicators. This memory bank is permalocked at the time of manufacture, in that it can not be changed.

The TID Memory bank contains two fields - Mask Designer Identifier (MDID) and Tag Model Number (TMN) - which are commonly associated together and called TID. MDID encodes the tag chip manufacturer ID which is assign ed by GS1 as a unique identifier. The TMN is attributed by the manufacturer of the chip and describes the chip identification number.

~~In~~ the TDS, where the encoding of TID is specified, ~~it~~ is often referred ~~has~~ as *Short* Tag Identification, because the TID can be extended.

The TID extension is called Extended Tag identifier (XTID) and is intended to provide more information about the capabilities of tags. It adds support for serialization and information about key features implemented by the tag [52]. Serialization is unique number generated by the tag manufacturer and can be used to uniquely identify one tag from another. This

MONZA 6 TID MEMORY BANK																
Word	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
50_h-$5F_h$	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1
40_h-$4F_h$	1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	1
30_h-$3F_h$	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1
20_h-$2F_h$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10_h-$1F_h$	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0
00_h-$0F_h$	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0

MEMORY MAP LEGEND				
Segment	Location	Bits	Binary	Value
ISO / IEC 15963 Class Identifier	00_h-07_h	8	11100010	GS1 EPCglobal Class 1 Gen 2
XTID Indicator (X bit)	08_h	1	1	Indicates the presence of an extended TID (XTID)
Security Indicator (S bit)	09_h	1	0	Does not implement *Authenticate* or *Challenge* commands
File Indicator (F bit)	$0A_h$	1	0	Does not implement the *FileOpen* command
Mask Designer Identifier (MDID)	$0B_h$-13_h	9*	000000001	Impinj
Tag Model Number (TMN)	14_h-$1F_h$	12	000101110000	Tag model number (Monza R6-P)
EPC Tag Data Standard Header	20_h-$2F_h$	16	0010000000000000	Supports extended TID (XTID) – 48-bit SN
Wafer Mask Revision	30_h-32_h	3	000	Indicates the Mask Revision for the tag
Integra™ TID Parity	33_h	1	1	Bit is set to guarantee bits 30:5F have even parity
Reserved for Future Use	50_h-52_h	3	000	
Monza Series ID	53_h-54_h	2	01	Supports Series 0 – Series 3
Monza Series Cycle Counter	34_h	1	0	Series rollover indicator
Serial Number		38	00011011111111001100 111010110001000100 1	30037989513 (decimal)

**The GS1 currently defines the MDID as the 9 bit value from $0B_h$ to 13_h in the Tag Data Standard (TDS). A previous TDS definition included the X, S and F bits in the MDID. For applications using a 12-bit MDID, the value would be 100000000001_b or 801_h.*

Figura 3.3: TID Memory Bank of Monza R6-P Series used in this dissertation [73]

identifies the tag itself, rather than the item it is applied to. XTID implementations are common among tag manufactures. So common in fact, people started referring to the MDID, TMN and serialization combination ~~has~~ *TID number*. The XTID is specially useful in cases where the EPC is not serialized or invalid. An example of a TID memory bank with XTID can be seen in figure 3.3.

User Memory

The User Memory Bank provides a variable size memory to store additional data related to the tagged object. It is frequently used to save information like temperature, maintenance logs, expiration dates and other type of data. The bank implementation is optional and must be indicated in the UMI bit in the PC Word.

To ensure compatibility with other protocols, the first eight bits of the bank shall contain a Data Storage Format Identifier (DSFID) as specified in ISO15962 [74]. This dissertation does not make use of the User Memory Bank, whereby it will not be covered. For further reference, GS1 presents a solution for encoding SCM data in the User Memory Bank in the TDS [52].

The access to the bank is made in blocks, through the Gen2 Air Protocol. The arrangement of the data in the Bank is important, as it can impact reading speeds. GS1 provides a user memory encoder, a software which converts application data, into suitable encoded data ready to be stored in the User Memory Bank of Gen2 RFID tags [75].

3.4 Electronic Product Code (EPC)

EPC is a universal identifier used to identify every physical object anywhere in the world. Differently from common Universally Unique Identifier (UUID) identifiers, EPC has a set of collective terms for the identification code, standardized by GS1 [76]. These terms convey context about the physical object in the encoding of the EPC itself.

3.4.1 EPC in EPCGlobal Architecture Framework

EPCs are presented throughout the EPCGlobal Framework in various levels of abstraction. From low-level binary encoded, in Gen2 RFID tags, to text based URIs in business level applications. The Framework presents seven representations [1] for a single EPC. In general, only three are primarily used: *Pure Identity*, *Tag URI* and *Binary Encoding.* Referring back to figure 3.1, we observe that the three representations are used in different contexts throughout the framework.

The *Pure Identity EPC URI* format is, as the name suggests, represented in Uniform Resource Identifier (URI) format. GS1 uses the Universal Resource Name (URN) scheme with the `epc` Namespace Identifier (NID) registered for the EPCglobal's EPC and related standards [77]. It is the most platform agnostic representation of an EPC, offering human readability and compatibility between heterogeneous systems.

For components like middlewares, requiring more information about the EPC memory bank of Gen2 RFID tags, the TDS provides a *Tag URI* scheme. This representation maintains the URI representation, changing the Namespace-specific String (NSS) from `id` to `tag`, and adding *control information* used to guide the process of data capture of RFID tags. This scheme preserves information regarding the EPC Memory Bank in the URN namespaces that are usually disregarded in business applications but necessary in middleware operations. In

[1] binary, tag-encoding URI, pure-identity URI, legacy, legacy AI, element string and ONS hostname

other words, the *EPC Tag URI* is a text equivalent of the entire EPC memory bank contents. Examples of booth URI representations can be seen in figure 3.4.

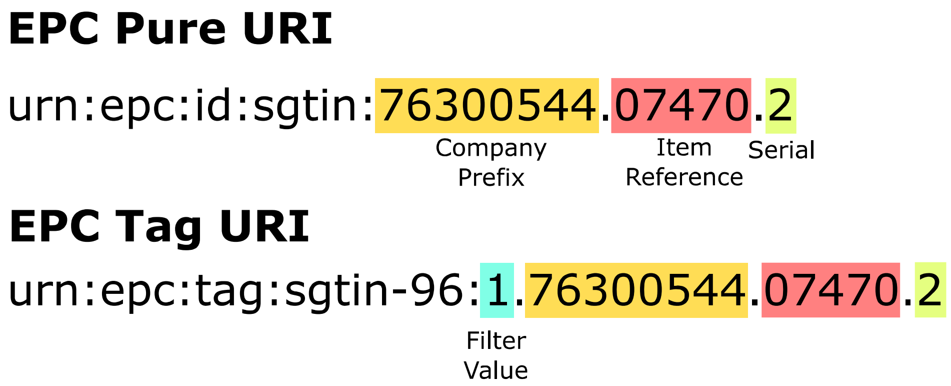


Figura 3.4: *Pure Identity* and *Tag URIs* EPC representation adapted from [78]

The Binary Encoding of EPC contains a compressed encoding of the EPC and additional *control information* in a compact binary form, like showed on figure 3.5. A deep analysis of the binary EPC encoding will be presented in section 3.4.4.

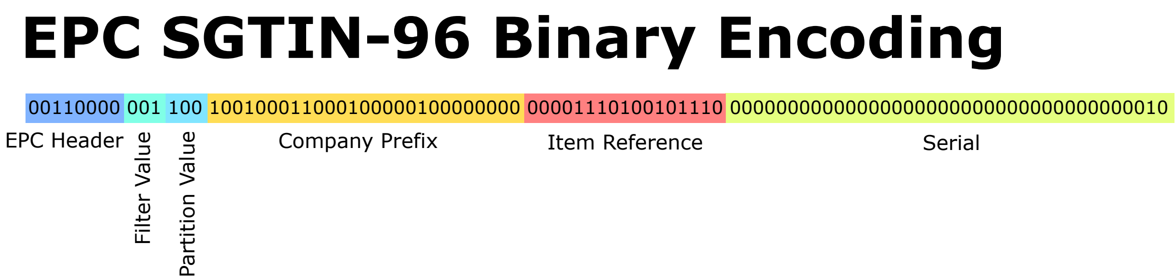


Figura 3.5: EPC in binary encoding representation adapted from [78]

3.4.2 Relationship between EPCs and GS1 keys

Before going into the encoding of EPC in Gen2 RFID tags, lets understand the underlying concepts of EPC and its schemes.

Previously, I mentioned that the EPC was designed to identify every physical object in the world. The *EPCGlobal Framework* uses this concept to its very extent. A physical object in a SCM can be a broad term that does not provides much information regarding the object itself. To contextualize the objects, the architecture defines GS1 keys and corresponding EPC schemes. Each GS1 key denotes a class or grouping of physical objects. These classes encompass trade items, locations, assets, logistic units, transport groupings to say a few. GS1 keys add valuable information to SCM operations by allowing intervening entities in the supply chain and logistics, to retrieve class context information regarding the tagged objects.

For each GS1 key there is a corresponding EPC scheme, including encoding specifications for both EPC representation: URI and a binary encoding, for use in RFID tags. Each EPC scheme and corresponding GS1 key can be seen in Appendix A.

3.4.3 SGTIN

In the scope of this dissertation, the SGTIN scheme ~~might~~ is the most important encoding scheme to look at. The SGTIN encodes a GTIN plus a unique product or serial number.

Figura 3.6: Relation and interoperability between SGTIN and barcode's GTIN

The GTIN is used by companies to identify trade items [79], [80]. GS1 defines trade items as products or services that are priced, ordered or invoiced at any point in the supply chain. GTIN can be used to identify types of products at any packaging level (e.g., consumer unit, inner pack, case, pallet [2]).

There are four GTIN formats: GTIN-8, GTIN-12, GTIN-13 and GTIN-14. In RFID applications and IT applications, it is used the generalized 14-digit GTIN format. Gen2 RFID SGTIN encoding scheme is specified for GTIN-14. Other GTIN formats are mainly used in barcodes in point-of-sale applications: in the U.S. is commonly used a GTIN-12 with UPC barcodes for single products and GTIN-14 with ITF-14 barcodes for product grouping. In contrast, in the rest of the world is commonly used GTIN-8 with EAN-8 barcodes for single products and GTIN-13 with EAN-13 for single products packaging configurations [82], [83].

A GTIN is composed of three field: *Company Prefix*, *Item Reference* and check digit. All these fields are encoded in the same manner in the different N-digit GTIN formats. A GTIN-13 can be converted in a GTIN-14 by adding a leading zero, and other GTINs formats follow the same logic.

GS1 *Company Prefix* is an identifier licensed by GS1 to identify a company globally. Nowadays, the registration of such is essential to companies wishing to sell products in big retail stores and digital marketplaces like Ebay and Amazon [84]). A *Company Prefix* does not uniquely identify a manufacturer. Companies can register and own more than one *Company Prefix* [49], and in certain circumstances *Company Prefixes* can change. When

[2] even if a GTIN can be used to identify pallets, the use of Serial Shipping Container Code (SSCC) is preferable [81]

licensing a *Company Prefix*, there is an assessment and plan of unique product items a company shall produce. Depending on that, GS1 attributes a prefix with a length adjusted to company requirements. Companies requiring high quantities of unique item are given a shorter Company Prefix, accommodating more digits for item identification. Also worth mentioning, the attribution of *Company Prefixes* is made by one of the GS1 branches. Each branch has a prefix which uses to assign *Company Prefixes* (e.g. GS1 Portugal has 560, GS1 Schweiz, Suisse and Svizzera have 760-769) [85].

Item References are assign ed by the managing entity of the product. The item reference must be concatenated with the *Company Prefix* to calculate the Check Digit, forming a GTIN. From a GTIN, an SGTIN can be commissioned [3], allowing to uniquely identify a product within its product grouping. Important to clarify, when a GTIN is stored in RFID tags, as in SGTIN coding schemes, the Check Digit has no purpose, so it is dropped. The Check Digit exists for error checking in barcodes. When converting barcode GTIN to EPC SGTIN encoding scheme, the Check digit must be dropped.

3.4.4 Binary Encoding

VOU AQUI!

Lets now understand how EPCs are stored in Gen2 RFID tags. We will focus on SGTIN encoding scheme, but the encoding method is analogous to other encoding schemes.

The binary encoding of an EPC consists of a fixed length header followed by a series of fields inherent to the encoding scheme like showed on SGTIN-96 Coding table 3.3. For instance, lets take the *Tag URI* `urn:epc:tag:sgtin-96:1.76300544.07470.2` has example. This *Tag URI* presents a SGTIN-96 encoding scheme, *Filter value* of “1”, *Company Prefix* “76300544”, *Item Reference* “7470” and serial “2”.

Following table 3.4, showing the *Tag URI* encoding, we first need to address the EPC header value. A snippet of the EPC header values can be seen in table 3.5. We observe that an SGTIN with 96-bit coding scheme encodes a 8-bit header with binary value of “00110000”.

Next, we need to encode the additional *information* included in the EPC Memory Bank: *Filter* and *Partition*.

The *Filter* encodes the packing level. An EPC SGTIN can be used to identify different levels of item packaging sharing the same GTIN. Differently from barcodes, which only encode the GTIN, an EPC with *Filter* field allows a coherent GTIN across all item packages. The *Filter* value allows RFID readers to select or deselect tags in the *EPC UHF Gen2 Air Interface Protocol* corresponding to certain filter levels. This makes it easier to read desired tags in an environment where there may be other tags present (e.g. logistics companies only wanting to read and track Unit Loads like pallets). Referencing table 3.6, with a filter value of “1”, it encodes a POS Trade Item with binary encoding of “001”.

The *Partition* value encodes a pair of variable-length numeric fields referent to the *Company Prefix* and *Item Reference* memory partition. Previously we mentioned that *Company Prefixes* can vary in length depending on company requirements for unique *Item References*. In barcodes the *Company Prefix* and *Item Reference* are encoded together in a GTIN value. In

[3] “Commisioning” is the technical term used when creating new EPCs for objects

Scheme	SGTIN-96					
URI Template	`urn:epc:tag:sgtin-96:`*F.C.I.S*					
Total Bits	96					
Logical Segment	EPC Header	Filter	Partition	GS1 Company Prefix (*)	Indicator (**) / Item Reference	Serial
Logical Segment Bit Count	8	3	3	20-40	24-4	38
Coding Segment	EPC Header	Filter	GTIN			Serial
URI portion		*F*	*C.I*			*S*
Coding Segment Bit Count	8	3	47			38
Bit Position	$b_{95}b_{94}...b_{88}$	$b_{87}b_{86}b_{85}$	$b_{84}b_{83}...b_{38}$			$b_{37}b_{36}...b_{0}$
Coding Method	00110000	Integer	Partition			Integer

Tabela 3.3: Coding Table of SGTIN-96 [52]

Field	**Value (dec)**	**Value (bin)**	**Length (bits)**	**Notes**
EPC Header	48	00110000	8	SGTIN-96 encoding
Filter Value	1	001	3	Point of Sale (POS) item
Partition Value	4	100	3	Company Prefix has 8 digits
Company Prefix	76300544	1001000110001000 00100000000	27	
Item Reference	7470	00001110100101110	17	Length depends on the partition MSB zero fill
Serial	2	0000000000000000 0000000000000000000010	38	Serial is of fixed length. Can be extended with bigger EPC Memory Bank

Tabela 3.4: SGTIN-96 binary encoding example of `urn:epc:tag:sgtin-96:1.76300544.07470.2` *Tag URI* retrieved from the Platform developed in this dissertation

Header Value (binary)	Header Value (hexadecimal)	Encoding Length (bits)	Coding Scheme
...			
0010 1100	2C	96	GDTI-96
0010 1101	2D	96	GSRN-96
0010 1110	2E	96	GSRNP
0010 1111	2F	96	USDoD-96
0011 0000	30	96	SGTIN-96
0011 0001	31	96	SSCC-96
0011 0010	32	96	SGLN-96
0011 0011	33	96	GRAI-96
0011 0100	34	96	GIAI-96
0011 0101	35	96	GID-96
0011 0110	36	198	SGTIN-198
0011 0111	37	170	GRAI-170
0011 1000	38	202	GIAI-202
0011 1001	39	195	SGLN-195
0011 1010	3A	113	GDTI-113 (DEPRECATED as of TDS 1.9)
0011 1011	3B	Variable	ADI-var
0011 1100	3C	96	CPI-96
0011 1101	3D	Variable	CPI-var
0011 1110	3E	174	GDTI-174
0011 1111	3F	96	SGCN-96
0100 0000	40	110	ITIP-110
0100 0001	41	212	ITIP-212
...			

Tabela 3.5: EPC headers snippet adapted from TDS [52]

Gen2 tags, although there is fixed size memory shared between *Company Prefix* and *Item Reference*, there is also the *Partition* value, which specifies the distribution of that partition between booth fields. In the case of the *Tag URI* example, *Company prefix* has 27 bits (8 digits) and *Item reference* has 17 bits (5 digits). Referring to table 3.7 with the pair of variable-lengths, the *Partition Value* is “4” encoded has “100”.

The *Company Prefix*, *Item Reference* and *serial* are encoded converting from decimal to binary and add leading zeros to fill all bits in each Logical Segment.

Type	Filter Value	Binary Value
All Others	0	000
Point of Sale (POS) Trade Item	1	001
Full Case for Transport	2	010
Reserved	3	011
Inner Pack Trade Item Grouping for Handling	4	100
Reserved	5	101
Unit Load	6	110
Unit inside Trade Item or component inside a product not intended for individual sale	7	111

Tabela 3.6: SGTIN Filter Value Table [52]

Partition Value (*P*)	GS1 Company Prefix		Indicator/Pad Digit and Item Reference	
	Bits (*M*)	**Digits (*L*)**	**Bits (N)**	**Digits**
0	40	12	4	1
1	37	11	7	2
2	34	10	10	3
3	30	9	14	4
4	27	8	17	5
5	24	7	20	6
6	20	6	24	7

Tabela 3.7: SGTIN Partition Table [52]

3.5 Low Level Reader Protocol (LLRP)

The Low Level Reader Protocol (LLRP) is specification for the network interface between RFID Readers and Client applications [86]. LLRP is not exclusive to the Gen2 Air Protocol. It was designed to support multiple RFID air protocols, including future versions of Gen2 Air Protocol.

3.5.1 Design Requirements

In some RFID systems, there is a requirement for explicitly tune RFID air protocols and the ability to control Readers that implement RFID air protocol communications [54]. LLRP provides tuning features and commands for that purpose.

Devices intended to operate and communicate with RFID Readers can vary from software applications, to middleware on local hardware, Programmable Logic Controllerss (PLCs) and even cloud services. LLRP is "simple" enough to be implemented in all kinds of computer architectures.

LLRP was design without requiring real-time interaction between the application software and Reader, and but with time-critical tasks at heart. The Reader application software passes operational rules to the Reader in non-real time. The Reader then triggers and runs those operation rules to achieve time-critical requirements. The triggers can come from the applications directly, from timers, General Purpose Input (GPI) hardware, or any other trigger defined by the Reader. This declarative operation method allows Readers to achieve peak performance without constraints caused by network or host latency.

3.5.2 Connection Details

LLRP is a binary protocol which runs over the TCP/IP internet transport protocols. It is an asymmetric protocol where the LLRP client send commands to the Reader.

The protocol supports both reader and client initiated connections. By default, LLRP clients connect to TCP port 5084. In reader initiated connections, the Reader will actively try to establish a connection with the host application. LLRP does not specify the behavior delivering data when a connection if broken. The reader used in this dissertation and others I've seen will continue to collect tag data and optionally deliver upon resumption of the connection.

Readers only allow one LLRP connection at any time. The Transmission Control Protocol (TCP) connection between reader and client stays open until the client closes it or connection drops.

3.5.3 Operation

The primary function of the LLRP interface is to allow a client to finely tune the Gen2 Tag Air Standard parameters, and command the reader to perform an inventory and otherwise access tags for read, write, lock, and kill.

To meet these requirements, LLRP defines Specs, containing descriptions of "what, how and when" the Reader should perform certain operations. Specs are run when triggered. The trigger range from boundary trigger information defined in the Spec itself, GPIs, or by immediate triggers from applications for a more imperative behavior. Specs are sent inside of messages, which are the unit of communication between client and Reader. LLRP contains forty basic messages which range from commands, responses, events and a `CUSTOM_MESSAGE` for vender extensions. A list and description of every message can be found in Appendix B.

In figure 3.7 we observe an example of binary encoding of the `RO_ACCESS_REPORT` message. Inside the message there is a Spec, and inside a Spec, data is sent as *parameters* and *fields*. *Fields* are individual data elements with a known basic format. *Parameters* are named data elements that contain other parameters and/or fields, much like structures in programming languages. Inside the `RO_ACCESS_REPORT` message, in figure 3.7, it can be observed the memory allocated for a list of a parameter called `TagReportData`. `TagReportData` is then encoded following figure 3.8, which inside encodes an `EPCData` or `EPC-96` parameter, where the binary encoded EPC is allocated. When constructing LLRP messages, parameters can be optional, but field must be presented and within their valid range. Some tooling software provides user friendly features like inferring Reader capabilities and default values for unspecified fields.

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="3">Rsvd</td><td colspan="3">Ver</td><td colspan="10">Message Type = 61</td><td colspan="16">Message Length [31:16]</td></tr>
<tr><td colspan="16">Message Length [15:0]</td><td colspan="16">Message ID[31:16]</td></tr>
<tr><td colspan="16">Message ID[15:0]</td><td colspan="16"></td></tr>
<tr><td colspan="32">TagReportData Parameter (0-n)</td></tr>
<tr><td colspan="32">RFSurveyReportReportData Parameter (0-n)</td></tr>
<tr><td colspan="32">Custom Parameter (0-n)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figura 3.7: `RO_ACCESS_REPORT` message binary encoding [54]. This message is sent by the Reader and contains inventory and access operations results

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="6">Reserved</td><td colspan="10">Type = 240</td><td colspan="16">Length</td></tr>
<tr><td colspan="32">EPCDataParameter [See notes below]</td></tr>
<tr><td colspan="32">ROSpecID Parameter (0-1)</td></tr>
<tr><td colspan="32">SpecIndex Parameter (0-1)</td></tr>
<tr><td colspan="32">InventoryParameterSpecID Parameter (0-1)</td></tr>
<tr><td colspan="32">AntennaID Parameter (0-1)</td></tr>
<tr><td colspan="32">PeakRSSI Parameter (0-1)</td></tr>
<tr><td colspan="32">ChannelIndex Parameter (0-1)</td></tr>
<tr><td colspan="32">FirstSeenTimestampUTC Parameter (0-1)</td></tr>
<tr><td colspan="32">FirstSeenTimestampUptime Parameter (0-1)</td></tr>
<tr><td colspan="32">LastSeenTimestampUTC Parameter (0-1)</td></tr>
<tr><td colspan="32">LastSeenTimestampUptime Parameter (0-1)</td></tr>
<tr><td colspan="32">TagSeenCount Parameter (0-1)</td></tr>
<tr><td colspan="32">AirProtocolTagDataParameter (0-n)[See Notes below]</td></tr>
<tr><td colspan="32">AccessSpecID Parameter (0-1)</td></tr>
<tr><td colspan="32">OpSpecResultParameter (0-n) [See notes below]</td></tr>
<tr><td colspan="32">Custom Parameter (0-n)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figura 3.8: `TagReportData` parameter binary encoding [54]. `TagReportData` is generated per tag and contains a mandatory parameter `EPCData` with encoding

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="6">Reserved</td><td colspan="10">Type = 241</td><td colspan="16">Length</td></tr>
<tr><td colspan="16">EPCLengthBits</td><td colspan="16"></td></tr>
<tr><td colspan="32">EPC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figura 3.9: `EPCData` parameter binary encoding [54]. `EPCData` shall be used for encoding a non-96 bit EPC, whereas the `EPC-96` Parameter on figure 3.10 is be used for encoding a 96-bit EPC

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td colspan="7">Type=13</td><td colspan="24">EPC[95:72]</td></tr>
<tr><td colspan="32">EPC[71:40]</td></tr>
<tr><td colspan="32">EPC[39:8]</td></tr>
<tr><td colspan="8">EPC[7:0]</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Figura 3.10: `EPC-96` parameter binary encoding [54]

3.5.4 Specification (Spec)

There are two main Specs defined in the LLRP Standard: *Reader Operation Specs (ROSpecs)* and *Access Specs*. Most of other Specs and parameters are enclosed under one of the two. I will briefly discuss relevant knowledge required to understand the Specs context. Detailed information on necessary fields will be presented later in opportune moments throughout this dissertation.

Reader Operation Spec (ROSpec)

ROSpecs control the operation of the Reader. They describe the inventory operations the Reader has to perform. A Reader only supports one ROSpec at a time. An example of a ROSpec is presented inside the *ADD_ROSPEC* message seen in Code 1. XML is used in the context of many LLRP clients to provide human readability of the binary protocol, and also used in this dissertation for the same reason. In the *ADD_ROSPEC* XML message we observe one ROSpec and constituting parameters and fields. ROSpecs contain the following elements:

`ROSpecID` is an ID set by the client to uniquely identify the Spec in subsequent commands.

`Priority` should be set to “0” [86]. By default, all ROSpecs should have the same priority. High priority ROSpecs can preempt an active lower priority one and start execution as soon as the start condition occurs.

`CurrentState` describes the current state of the ROSpec - *disabled*, *inactive*, *active*. When adding and ROSpec the state must be set to *disable*.

`ROBoundarySpec` is a parameter containing a description of the start and stop conditions for the Spec. These conditions can be, for the start trigger:

- `None/Null` - ROSpec will not start unless indicated by the client application through an *START_ROSPEC* message;
- `Immediate` - the ROSpec will start immediately after enabled, and will continuously restart itself until disabled;
- `Periodic` - when enabled, the ROSpec will be triggered periodically;
- `GPI` - the enabled ROSpec will start when the GPI event enters a certain state.

For the stop triggers:

- `None/Null` - the ROSpec only stops when stopped by the client through the *STOP_ROSPEC*;
- `Duration` - ROSpec stops when it has been active for a specified duration;
- `GPI` - ROSpec will stop when the GPI event enters a certain state.

`Antenna Inventory Spec (AISpec)` contains the settings for the Reader’s antennas and Gen2 Air protocol options. An ROSpec can have one or more AISpecs which are executed when the ROSpec runs.

`ROReportSpec` is optional and describes “when” reports should be forward to the client and “what” the report contains. If not defined, the Reader will use the current or default settings.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ADD_ROSPEC xmlns="http://www.llrp.org/ltk/schema/core/encoding/xml/1.0" Version="1" MessageID="0">
  <ROSpec>
    <ROSpecID>666</ROSpecID>
    <Priority>0</Priority>
    <CurrentState>Disabled</CurrentState>
    <ROBoundarySpec>
      <ROSpecStartTrigger>
        <ROSpecStartTriggerType>Periodic</ROSpecStartTriggerType>
        <PeriodicTriggerValue>
          <Offset>0</Offset>
          <Period>5000</Period>
        </PeriodicTriggerValue>
      </ROSpecStartTrigger>
      <ROSpecStopTrigger>
        <ROSpecStopTriggerType>Duration</ROSpecStopTriggerType>
        <DurationTriggerValue>2000</DurationTriggerValue>
      </ROSpecStopTrigger>
    </ROBoundarySpec>
    <AISpec>
      <AntennaIDs>0</AntennaIDs>
      <AISpecStopTrigger>
        <AISpecStopTriggerType>Null</AISpecStopTriggerType>
        <DurationTrigger>0</DurationTrigger>
      </AISpecStopTrigger>
      <InventoryParameterSpec>
        <InventoryParameterSpecID>1</InventoryParameterSpecID>
        <ProtocolID>EPCGlobalClass1Gen2</ProtocolID>
      </InventoryParameterSpec>
    </AISpec>
    <ROReportSpec>
      <ROReportTrigger>Upon_N_Tags_Or_End_Of_ROSpec</ROReportTrigger>
      <N>0</N>
      <TagReportContentSelector>
        <EnableROSpecID>0</EnableROSpecID>
        <EnableSpecIndex>0</EnableSpecIndex>
        <EnableInventoryParameterSpecID>0</EnableInventoryParameterSpecID>
        <EnableAntennaID>1</EnableAntennaID>
        <EnableChannelIndex>0</EnableChannelIndex>
        <EnablePeakRSSI>0</EnablePeakRSSI>
        <EnableFirstSeenTimestamp>0</EnableFirstSeenTimestamp>
        <EnableLastSeenTimestamp>0</EnableLastSeenTimestamp>
        <EnableTagSeenCount>0</EnableTagSeenCount>
        <EnableAccessSpecID>1</EnableAccessSpecID>
      </TagReportContentSelector>
    </ROReportSpec>
  </ROSpec>
</ADD_ROSPEC>
```

Código 1: Example of *ADD_ROSPEC* message in XML representation

3.5.5 Application Flow

A typical LLRP application flow can be seen in figure 3.11. Upon establishing a TCP connection, a Client add and enables an ROSpec, containing information about the inventory procedures. The Client can also register a AISpec for Gen2 commands, like writing EPCs. The Reader, upon trigger, executes the Specs, performing EPC inventory and tag operations, which are reported back to the Client.

Figura 3.11: Example of LLRP Application Flow [86]

Access Specification

An *Access Spec* handles the extended tag operations of Gen2: read, write, lock and kill. It was design to be extensible, in order to support multiple air protocols and their respective commands. These will not be directly used in the practical context of this dissertation, so no further description will be added.

3.6 Filter and Collection (F&C)

F&C is a specification for middlewares operating between readers and client applications, to receive, filter, translate and deliver data to client applications.

Gen2 RFID readers can generate large amounts of network traffic. The amount of requests generated, containing reports with EPC raw data, can bottleneck most networks, being unsuited for internet data exchanges. Often, client applications are hosted in different networks and even different physical places. With the current paradigm of cloud computing, the interest in processing data on data centers only seems to be growing. Furthermost, raw EPC data is not "high-level" compared to the paradigm of most business logic in client applications. Client applications, without any middleware, have to process and contextualize huge amounts of data, overburdening services with workloads that are not in their scope of work.

3.6.1 Responsibilities

In the standard specification, EPCGlobal defines responsibilities and interfaces for F&C middlewares. Companies are free to extended these responsibilities and add features in their own implementations.

A F&C middlewares must be able to receive raw tag reads from one or multiple RFID readers through an LLRP client interface. These raw tags, in binary encoding, must be decoded and translated in to URI representations, used in client applications.

Tag reads must all be processed to reduce the EPC data volume: this includes filtering, by ignoring EPCs according to subscription patterns; aggregate EPCs over time intervals, eliminating duplicate reads within that interval; summarize and count EPCs in specific object classes; and differential analysis, by reporting which EPCs have been added or removed.

F&C must be able to invoke Gen2 commands on the reader, required by applications to write, lock, kill, and otherwise operate upon tags. F&C should also map readers by *logical reader names* defined in the configurations commands and/or file.

Lastly, F&C middleware must expose an ALE interface to provide means for client applications to request EPC data, reader information, control information and configure all F&C aspects described previously.

3.7 Application Level Events (ALE)

The ALE interface role is to provide independence between the infrastructure components and the applications that use the data. Business application have access to the lower layers of the infrastructure, like readers and middlewares, without configuring hardware or make low level logical indicators to select data. Succinctly, the interface allows client applications to register and subscribe to their interested EPC patters and request strategies in an high-level declarative way.

ALE uses a Web Services Description Language (WSDL), an XML based interface used to describe functionality offered by web services, to define, configure and request reports from middlwares and smart readers. The XML schema is defined in a XML Schema Definition (XSD) file. The XSD file expresses a set of rules to which XML documents must conform in order to be considered valid. The interface also defines Simple Object Access Protocol) (SOAP)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:LRSpec xmlns:ns2="urn:epcglobal:ale:wsdl:1" xmlns:ns3="urn:epcglobal:ale:xsd:1">
    <isComposite>false</isComposite>
    <readers />
    <properties>
        <property>
            <name>ReaderType</name>
            <value>org.fosstrak.ale.server.readers.llrp.LLRPAdaptor</value>
        </property>
        <property>
            <name>Description</name>
            <value>Warehouse reader for inventory management at Aveiro boutique store</value>
        </property>
        <property>
            <name>PhysicalReaderName</name>
            <value>ImpinjSpeedwayShelve1</value>
        </property>
        <property>
            <name>ip</name>
            <value>169.254.1.1</value>
        </property>
        <property>
            <name>port</name>
            <value>5084</value>
        </property>
        <property>
            <name>clientInitiated</name>
            <value>true</value>
        </property>
    </properties>
</ns3:LRSpec>
```

Código 2: Example of *LRSpec* used to register a single Reader named *ImpinjSpeedwayShelve1* with Internet Protocol (IP) 169.254.1.1

bindings on top of the WSDL for the callback interface for reading and writing Application Programming Interfaces (APIs).

3.7.1 Specifications and Reports

To configure and subscribe to interested EPC patterns, client applications must first register *logical readers* using a *Logical Reader Spec (LRSpec)*. An example can be seen in Code 2. *LRSpecs* are basically readers registration forms.

To subscribe to EPC data, clients register *Event Cycle Specs (ECSpecs)*. ECSpecs take inspiration from the same ideas as in LLRP, by describing event boundaries, filtering and subscription patterns, aggregation rules and differential analysis report options in a truly high-level form. One example can be seen in Code 3. The example subscribes to EPC data from the logical reader *Reader_Shelve1*. The middleware should send back to the client two types of reports every 5 seconds: one reporting all EPC tag-URIs added to the reading zone of the reader; and other reporting all EPC tag-URIs removed from the reading zone and a count of total removed tags. A complex *ECSpec* with subscription patterns will be presented in next chapters, in the practical implementation sections of this dissertation.

Subsequently, client have to indicate where the *ECSpecs* reports should be delivered, by

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ECSpec xmlns:ns2="urn:epcglobal:ale:xsd:1">
    <logicalReaders>
        <logicalReader>Reader_Shelve1</logicalReader>
    </logicalReaders>
    <boundarySpec>
        <repeatPeriod unit="MS">5000</repeatPeriod>
        <duration unit="MS">3000</duration>
        <stableSetInterval unit="MS">0</stableSetInterval>
    </boundarySpec>
    <reportSpecs>
        <reportSpec reportName="additions">
            <reportSet set="ADDITIONS" />
            <output includeTag="true" />
        </reportSpec>
        <reportSpec reportName="deletions">
            <reportSet set="DELETIONS" />
            <output includeTag="true" />
        </reportSpec>
    </reportSpecs>
</ns2:ECSpec>
```

Código 3: *ECSpec* example used in early stages of the practical work of this dissertation

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ECReports terminationCondition="DURATION" totalMilliseconds="3003"
↪   ALEID="ETHZ-ALE-1001980346" date="2020-07-15T15:44:20.651Z"
↪   specName="ec_warehouse_shelve1" xmlns:ns2="urn:epcglobal:ale:xsd:1">
    <reports>
        <report reportName="deletions">
            <group>
                <groupList>
                    <member>
                        <tag>urn:epc:tag:sgtin-96:3.0037000.030241.1041970</tag>
                        <tag>urn:epc:tag:sgtin-96:3.0037000.030241.1041971</tag>
                        <tag>urn:epc:tag:sgtin-96:3.0037000.030241.1041972</tag>
                    </member>
                </groupList>
                <groupCount>
                    <count>3</count>
                </groupCount>
            </group>
        </report>
    </reports>
</ns2:ECReports>
```

Código 4: Example of *ECReport* generated by the ECSpec's *detetions* Report Spec in Code 3, where a three tags were removed from the reading zone

indicating a `NotificationURI`. These reports are named *Event Cycle Reports (ECReports)* and an example can be seen in Code 4.

3.8 Capture Application

In the *EPCGlobal Framework*, the Capture Application role serves to contextualize EPC data from *ECReports* into EPCIS data, and send it to an EPCIS repository to be stored.

Capture Applications requirements are not further specified by the *EPCGlobal Framework*.

From what I have encountered, generic capture applications implement dynamic run-time rule engines, to specify business logic in an high-level context (e.g. Java Drools). Many integrate interfaces for other device-generated data such as bar codes, matrix data (e.g. QR Code), human input and data gathered from other software systems. Business logic in capture applications can be complex: retail capture applications can implement anti-thief mechanism, predict out-of-stock situations, infer clients shopping experience and gather data for machine learning services (e.g. to infer client interests) [87].

3.9 EPC Information Services (EPCIS)

EPCIS enables enterprises to create and share EPC related data with trading partners. The standard was conceived as part of a broader effort to enhance collaboration between trading partners, to create visibility of physical and digital objects within relevant business contexts [88].

3.9.1 Data Model and Core Business Vocabulary (CBV)

To ensure conformity in data shared across systems, the EPCIS standard defines a data model. This allows disparate application to operate under coherent data structures capable of being providing visibility in relevant business operations.

EPCIS structures the real-world business information as a sequence of individual business processes. Business processes can vary. From packaging product, packing into a shipping container, shipping, receiving, and so forth. Each completion of one of these business process is modeled by an EPCIS Event.

An EPCIS Event is the basic unit of data in the EPCIS data model. It provides a detailed picture of a business processes by organizing its contents into four dimensions: *What*, *When*, *Where* and *Why*.

The *What* dimension identifies the physical and digital objects involved in the event. It uses the EPC TDS schemes, seen previously, to do so. On section 3.4, we talked about EPC and how it encodes information about the physical object it refers to. We focused on SGTIN for item level identification, but different GS1 Keys exit to identify a multitude of SCM assets: like pallets, batches and lots, returnable assets, to say a few (see Appendix A).

When dimension records when the event took place. It contains three elements: `Event Time` contains the date and time at with the event took place; `Event Time Zone Offset` indicates the time zone at the place and time of the event; and `Record Time` which saves the date and time when the EPCIS event was recorded into an EPCIS repository.

Where dimension, as it implies, captures where the event physically took place and/or where are the objects following the event. The dimension allows for two location types: `Read Point` and `Business Location`. The `Read Point` is the location where the event took place. The `Business Location` is the location where the objects reside after the event.

Locations are preferably identified by a GLN. A GLN is a GS1 Identification Key used to identify physical location like factories, businesses and facilities [81]. GLNs are often

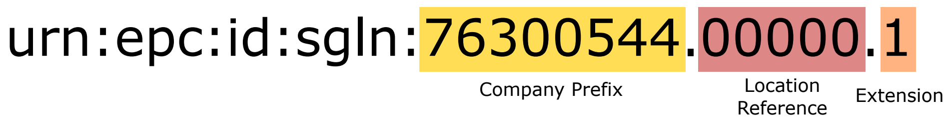


Figura 3.12: Example of GLN with extension (SGLN) in URI format. In the practical context of this dissertation, identifies the improvised shelve reading point in a Nespresso boutique store

accompanied by an extension used to identify internal physical locations within a location identified by the GLN. An example of a GLN can be seen in figure 3.12.

The *Why* dimension describes the context of the business process in which the event took place. This dimension includes a multitude of parameters, which are included in the EPCIS Event as a combination of such [4]:

- **Business Step** identifies what was taking place from a business context (e.g. `packing`, `shipping`, `inspecting`, `retail_selling`).
- Disposition identifies the business condition ensuing the event of the object (e.g. `recalled`, `retail_sold`, `in_transit`, `stolen`).
- **Business Transaction List** identifies business transactions relevant to an event. It is described by a pair of identifiers: `Business transaction type` which identifies the type of transaction; and `Business transaction ID`, and ID that identifies a specific transaction.
- **Source** and **Destination** are used to provide additional business context when the event is part of a business transfer of ownership, responsibility or custody.

The possible values of these parameters are standardized under the CBV Standard, which operates in association with the EPCIS to provide common and standardized business vocabulary [58]. The CBV, much like most of *EPCGlobal Framework*, uses a URI syntax has parameter values. An example of a populated EPCIS Event can be seen in table 3.8.

[4]Some of parameters change between EPCIS and CBV versions due to reformulations in both standards

Dim	Data Element	Design Choice (Section 4.6)	Actual EPCIS Event Contents
	Event Type	Object Event	
	Action	OBSERVE	
What	**EPC List**	A list containing one element: the SSCC of the pallet (instance-level identification)	`urn:epc:id:sscc:0614141.0123456789`
When	**Event Time**	The date and time at which the pallet is shipped	`2014-03-15T10:11:12Z`
	Event Time Zone Offset	The time zone offset in effect where the pallet was shipped	`-05:00`
Where	**Read Point**	Shipping dock #2 of building 10	`urn:epc:id:sgln:0614141.11111.2`
	Business Location	(omitted)	(omitted)
Why	**Business Step**	Shipping (from CBV)	
	Disposition	In Transit (from CBV)	
	Business Transaction List	A list containing two business transaction references: the Retailer's purchase order and the Manufacturer's invoice.	*Type* `urn:epcglobal:cbv:btt:po` `urn:epcglobal:cbv:bt:5012345678900:1234` *Type* `urn:epcglobal:cbv:btt:inv`
	Source List	A list containing one source of type "owning party," indicating the Manufacturer as the owning party at the source	*Type* `urn:epcglobal:cbv:sdt:owning_party` `urn:epc:id:sgln:0614141.11111.0`
	Destination List	A list containing one source of type "owning party," indicating the Retailer as the intended owning party at the destination	*Type* `urn:epcglobal:cbv:sdt:owning_party` `urn:epc:id:sgln:5012345.67890.0`

Tabela 3.8: EPCIS Event Information Content example from the business process of shipping a pallet [88]

3.9.2 Event types

To allow for more flexibility and variations in the structure of the *What* dimension, EPCIS defines a few variations of the basic EPCIS Event:

- **ObjectEvent** is the most simple commonly used. It represents an event that happened to one of more objects (e.g. shipping a pallet using a SSCC).
- **AggregationEvent** represents an event that happened to physically aggregated objects (e.g. aggregating cases in to a pallet, or removing cases from a pallet).
- **TrasformationEvent** represents an event in which the input objects are fully or partially consumed to produce output objects (e.g. processing raw materials in a product to be commercialized).
- **TransactionEvent** represents an event in which one or more objects become associated or disassociated with one or more business transactions (e.g. linking a pallet and cases to a commercial invoice)

```xml
<epcis:EPCISDocument xmlns:epcis="urn:epcglobal:epcis:xsd:1">
    <EPCISBody>
        <EventList>
            <ObjectEvent>
                <eventTime>2015-07-15T10:00:00.000-05:00</eventTime>
                <eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
                <epcList>
                    <epc>urn:epc:id:sgtin:0614141.012345.101</epc>
                    <epc>urn:epc:id:sgtin:0614141.012345.102</epc>
                    <epc>urn:epc:id:sgtin:0614141.012345.103</epc>
                </epcList>
                <action>OBSERVE</action>
                <bizStep>urn:epcglobal:cbv:bizstep:transporting</bizStep>
                <disposition>urn:epcglobal:cbv:disp:in_transit</disposition>
                <readPoint>
                    <id>geo:41.6725,-86.255278</id>
                </readPoint>
                <example:TemperatureC>15</example:TemperatureC>
                <example:RelativeHumidity>80</example:RelativeHumidity>
            </ObjectEvent>
        </EventList>
    </EPCISBody>
</epcis:EPCISDocument>
```

Código 5: Example of an EPCIS Report sent to a EPCIS capture interface. EPCIS Reports can be extended with User/Vendor Extensions. In this example we see a `TemperatureC` and `RelativeHumidity` vendor extensions

3.9.3 Capture and Query Interfaces

To complement the data model, EPCIS also standardizes two interfaces: a *Capture interface* that allows client to send EPCIS Events, to be saved, typically in a persistent repository of EPCIS data, called EPCIS repository; and a *Query interface* from which EPCIS data may be requested by applications and trading partners.

The data shared between clients and these interfaces is expressed in XML. EPCIS provides XML bindings for the data models described previously. The *Capture interface* can be used with either message queue or Hypertext Transfer Protocol (HTTP). In Code 5 can be seen an XML EPCIS Event example sent to a capture application.

EPCIS *Query interface* is defined as a web service and can be interacted with by the SOAP transport mechanism using the following available operations:

- `poll`: queries for EPCIS events matching specific criteria.
- `subscribe`: allows client to register a subscription for events matching specific criteria, which are delivered asynchronously to the client.
- `unsubscribe`: removes a registered subscription.
- `getQueryNames`: returns a list with the supported types of queries supported by the service.
- `getSubscriptionIDs`: returns a list of active subscriptions.
- `getStandardVersion`: return the EPCIS version supported by the service.

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <epcisq:Poll xmlns:epcisq="urn:epcglobal:epcis-query:xsd:1">
            <queryName>SimpleEventQuery</queryName>
            <params>
                <param>
                    <name>eventType</name>
                    <value>
                        <string>ObjectEvent</string>
                    </value>
                </param>
                <param>
                    <name>EQ_bizLocation</name>
                    <value>
                        <string>urn:epc:id:sgln:76300544.00000.1</string>
                    </value>
                </param>
            </params>
        </epcisq:Poll>
    </soap:Body>
</soap:Envelope>
```

Código 6: Example of EPCIS Query requesting all `ObjectEvents` from the Business Location `urn:epc:id:sgln:76300544.00000.1`

- `getVendorVersion`: returns a vendor string identifying any non-standard extensions supported by the service (e.g. `TemperatureC` and `RelativeHumidity` vendor extension in Code 5)

A simple `poll` operation can be seen in Code 6. Inside the SOAP envelope we can observe a `poll` operation requesting every `ObjectEvent` collected by the reading point `urn:epc:id:sgln:76300544.00000.1`.

In the next chapter I will present the state of the art of RFID smart shelves, purpose hardware and research.

APÊNDICE A

Information References

A EPC Schemes and Corresponding GS1 keys

EPC Scheme	Tag Encodings	Corresponding GS1 key	Typical use
sgtin	sgtin-96 sgtin-198	GTIN key (plus added serial number)	Trade item
sscc	sscc-96	SSCC	Pallet load or other logistics unit load
sgln	sgln-96 sgln-195	GLN of physical location (with or without additional extension)	Location
grai	grai-96 grai-170	GRAI (serial number mandatory)	Returnable/reusable asset
giai	giai-96 giai-202	GIAI	Fixed asset
gsrn	gsrn-96	GSRN – Recipient	Hospital admission or club membership
gsrnp	gsrnp-96	GSRN for service provider	Medical caregiver or loyalty club
gdti	gdti-96 *gdti-113* (DEPRECATED) gdti-174	GDTI (serial number mandatory)	Document

Tabela A.1: EPC Schemes and Corresponding GS1 keys Part 1 [52]

EPC Scheme	Tag Encodings	Corresponding GS1 key	Typical use
cpi	cpi-96 cpi-var	[none]	Technical industries (e.g. automotive) - components and parts
sgcn	sgcn-96	GCN (serial number mandatory)	Coupon
ginc	[none]	GINC	Logical grouping of goods intended for transport as a whole, assigned by a freight forwarder
gsin	[none]	GSIN	Logical grouping of logistic units travelling under one despatch advice and/or bill of lading
itip	itip-110 itip-212	(8006) + (21)	One of multiple pieces comprising, and subordinate to, a whole (which is, in turn, identified by an SGTIN or the combination of AIs 01 + 21).
upui	[none]	GTIN + TPX	Pack identification to combat illicit trade
pgln	[none]	Party GLN	Identification of economic operator; identification of owning party or possessing party in the Chain of Custody (CoC) / Chain of Ownership (CoO)
gid	gid-96	[none]	Unspecified
usdod	usdod-96	[none]	US Dept of Defense supply chain
adi	adi-var	[none]	Aerospace and defense – aircraft and other parts and items
bic	[none]	[none]	Intermodal shipping containers
imovn	[none]	[none]	Vessel identificaton

Tabela A.2: EPC Schemes and Corresponding GS1 keys Part 2 [52]

Message Name	Resp	Client/ Reader	Description
ADD_ACCESSSPEC	Y	C	Adds an AccessSpec to the reader
ADD_ROSPEC	Y	C	Adds an ROSpec to the Reader
CLIENT_REQUEST_OP	Y	R	Not supported by Impinj LTK
CLOSE_CONNECTION	Y	R	Reader generates this message before a client initiated connection closure
CUSTOM_MESSAGE	N/A	R or C	A custom message wrapper defined by LLRP to hold all custom message communication between the client and reader
DELETE_ACCESSSPEC	Y	C	Deletes an *AccessSpec* from the Reader
DELETE_ROSPEC	Y	C	Deletes an *ROSpec* from the Reader
DISABLE_ACCESSSPEC	Y	C	Disables an *AccessSpec* on the Reader
DISABLE_ROSPEC	Y	C	Disables an *ROSpec* on the Reader
ENABLE_ACCESSSPEC	Y	C	Enables an *AccessSpec* on the Reader

Tabela A.3: LLRP Messages (except for responses) Part 1 [86]

Message Name	Resp	Client/ Reader	Description
ENABLE_EVENTS_-AND_REPORTS	N	C	A message generated by the client to enable the RO_ACCESS_REPORT and READER_EVENT_NOTIFICATION. Only required by client when using the *HoldEventsAndReports* feature of LLRP.
ENABLE_ROSPEC	Y	C	Enables an *ROSpec* on the Reader
ERROR_MESSAGE	N	R	A message generated by the Reader when it is unable to properly decode and respond to a client message.
GET_ACCESSSPECS	Y	C	Gets the Reader's currently configured *AccessSpecs*
GET_READER_-CAPABILITIES	Y	C	Gets the Readers capabilities
GET_READER_-CONFIG	Y	C	Gets the Readers configuration
GET_REPORT	N	C	A message sent by the client to trigger a report to be generated from the Reader. This is in addition to any report triggers configured in the *ROSpec* or *AccessSpec*.
GET_ROSPECS	Y	C	Gets the Reader's currently configured *ROSpecs*
KEEP_ALIVE	Y	R	Reader periodically generates this message (when configured by the client). The response to this message generated by the client is called a KEEP_ALIVE_ACK (as opposed to the normal response nomenclature of LLRP)
READER_EVENT_-NOTIFICATION	N	R	A message generated by the reader to post asynchronous reader events (as opposed to tag events) to the client.
RO_ACCESS_REPORT	A	R	The report containing tag inventory and access data
SET_READER_CONFIG	Y	C	Sets the Reader configuration
START_ROSPEC	Y	C	Starts (activate) and *ROSpec* on the Reader
STOP_ROSPEC	Y	C	Stops (deactivates) an *ROSpec* on the Reader

Tabela A.4: LLRP Messages (except for responses) Part 2 [86]

Referências

[1] *Oxford Languages and Google - English* -, es-ES, https://languages.oup.com/google-dictionary-en/.

[2] B. Marr, *What Is Industry 4.0? Here's A Super Easy Explanation For Anyone*, en, https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/.

[3] B. C. Hardgrave e J. Patton, «2016 State of RFID Adoption Among U.S. Apparel Retailers,» en, p. 7,

[4] *DOD Releases Final RFID Policy - 2004-08-09 - Page 1 - RFID Journal*, https://www.rfidjournal.com/articles/view?1080.

[5] V. Oana, Barna, C. Maria, I. Surugiu, R. Petrescu e V. Alexandrescu, «RFID Technology in Containers Multimodal Transport,» *Supply Chain Management Journal*, vol. 4, jan. de 2013.

[6] D. M. Dobkin, *The RF in RFID, Second Edition: UHF RFID in Practice*, Second. USA: Newnes, 2012, ISBN: 0-12-394583-6.

[7] *The History of RFID Technology | RFID JOURNAL*, https://www.rfidjournal.com/the-history-of-rfid-technology.

[8] André Ventura da Cruz Marnoto Zúquete e N. Borges de Carvalho, *Identificação Por RFID, 2017-2018, Slides*, English, Universidade de Aveiro, 2018.

[9] J. Landt, «The History of RFID,» *IEEE Potentials*, vol. 24, n.º 4, pp. 8–11, 2005.

[10] H. Casier, M. Steyaert e A. Roermund, van, eds., *Analog Circuit Design : Robust Design, Sigma Delta Converters, RFID*, English. Germany: Springer, 2011, ISBN: 978-94-007-0390-2. DOI: 10.1007/978-94-007-0391-9.

[11] *RFID Adoption Stalls: Executive Summary | Computer Economics – for IT Metrics, Ratios, Benchmarks, and Research Advisories for IT Management*, https://www.computereconomics.com/article.cfm?id=1203.

[12] S. Gaudin, *Some Suppliers Gain from Failed Wal-Mart RFID Edict*, en, https://www.cio.com/article/2436434/some-suppliers-gain-from-failed-wal-mart-rfid-edict.html, abr. de 2008.

[13] *RFID in Retail Apparel*, en-US, https://gaorfid.com/rfid-in-retail-apparel/.

[14] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Second. Wiley Publishing, 2003, ISBN: 0-470-84402-7.

[15] S. Lahiri, *RFID Sourcebook*. IBM Press, 2005, ISBN: 0-13-185137-3.

[16] C. A. Balanis, *Antenna Theory: Analysis and Design*. Wiley-Interscience, 2005.

[17] *Electromagnetic Radiation: Field Memo | Occupational Safety and Health Administration*, https://www.osha.gov/radiofrequency-and-microwave-radiation/electromagnetic-field-memo.

[18] Zerodamage e Peter R. Franchi et al., *Far and Near Fields (Vectorized)*, en, mar. de 1991.

[19] P. Nikitin, K. Rao e S. Lazar, «An Overview of near Field UHF RFID,» sér. IEEE Int Conf RFID, abr. de 2007, pp. 167–174. DOI: 10.1109/RFID.2007.346165.

[20] *Safety and Health Topics | Radiofrequency and Microwave Radiation - Electromagnetic Radiation: Field Memo | Occupational Safety and Health Administration*, https://www.osha.gov/SLTC/radiofrequencyradiation/electromagnetic_fieldmemo/electromagnetic.html.

[21] *Impinj M730 & M750 Product Brief / Datasheet*, en-US, http://support.impinj.com/hc/en-us/articles/360010797539.

[22] Alien, *Alien Product Family Overview (2020 06 29)*, jun. de 2020.

[23] H. Tsai, K. Chen, Y. Liu, J. Norair, S. Bellamy e J. Shuler, «Applying RFID Technology in Nuclear Materials Management,» *Packaging, Transport, Storage and Security of Radioactive Material*, vol. 19, mar. de 2008. DOI: 10.1179/174651008X279000.

[24] Caen, *Caen A927Z Temp Logger Data Sheet 2018*.

[25] *RFID Coupling Techniques: Backscatter Capacitive Inductive » Electronics Notes*, https://www.electronics-notes.com/articles/connectivity/rfid-radio-frequency-identification/coupling-techniques-capacitive-inductive-backscatter.php.

[26] ISO, *ISO 11784:1996*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/02/58/25881.html.

[27] ——, *ISO 11785:1996*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/01/99/19982.html.

[28] ——, *ISO 14223-3:2018*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/96/59602.html.

[29] ——, *ISO/IEC 18000-2:2009*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/61/46146.

[30] ——, *ISO/IEC 15693-2:2019*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/36/73601.

[31] *MIFARE*, https://www.mifare.net/en/.

[32] ——, *ISO/IEC 14443-4:2018*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/35/73599.

[33] ——, *ISO/IEC 18000-3:2010*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/34/53424.

[34] ——, *ISO/IEC 18000-7:2014*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/73/57336.

[35] ——, *ISO/IEC 18000-6:2010*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/61/46149.

[36] ——, *ISO/IEC 18000-4:2015*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/25/62539.

[37] Hugo Manuel Oliveira de Miranda, «Sistemas RFID UHF,» pt-PT, MSc Dissertation, Universidade de Aveiro, Aveiro, 2015.

[38] M. Lorenz, J. Mu?ller, M.-P. Schapranow, P. D. A. Zeier e H. Plattner, «Discovery Services in the EPC Network,» em, jun. de 2011, ISBN: 978-953-307-265-4. DOI: 10.5772/16658.

[39] D. Simchi-Levi, P. Kaminsky e E. Simchi-Levi, *Cadeias de Suprimentos Projeto e Gestao*. Bookman, 2003, ISBN: 978-85-363-0119-8.

[40] Anonymous, *Standards | GS1*, en, https://www.gs1.org/standards, Text, dez. de 2014.

[41] *DoD_Suppliers_Passive_RFID_Info_Guide_v15update.Pdf*, https://www.acq.osd.mil/log/SCI/.AIT.html/DoD_Suppliers_

[42] *What Is RAIN RFID Technology? | IMPINJ*, https://www.impinj.com/about-rfid/about-rfid.

[43] *Technology Companies Create RAIN to Promote EPC UHF RFID Adoption - 2014-04-09 - Page 1 - RFID Journal*, https://www.rfidjournal.com/articles/view?11665.

[44] *En_302208v030101p.Pdf*, https://www.etsi.org/deliver/etsi_en/302200_302299/302208/03.01.01_60/en_302208v030101p.pd

[45] *What Could Slow RFID Adoption? | RFID JOURNAL*, https://www.rfidjournal.com/what-could-slow-rfid-adoption.

[46] *EU Upper Band Country List*, en-US, http://support.impinj.com/hc/en-us/articles/360006823440.

[47] «Publication: LEBENSMITTEL ZEITUNG LZ/NET NEWS,» en, p. 2,

[48] *Global RFID: The Value of the EPCglobal Network for Supply Chain Management | Request PDF*, en, https://www.researchgate.net/publication/273130979_Global_RFID_The_Value_of_the_EPCglobal_Network_for_Su

[49] *GS1 EPCglobal Architecture Framework v1.6 April 2014*, https://www.gs1.org/sites/default/files/docs/epc/architecture_1_ framework-20140414.pdf.

[50] *RFID_Barcode_Interoperability_Guidelines.Pdf*, https://www.gs1.org/docs/barcodes/RFID_Barcode_Interoperability_

[51] Dong-Ying Li, Shun-Dao Xie, R.-J. Chen e H.-Z. Tan, «Design of Internet of Things System for Library Materials Management Using UHF RFID,» em *2016 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, set. de 2016, pp. 44–48. DOI: 10.1109/RFID-TA.2016.7750755.

[52] *EPC Tag Data Standard (TDS) v1.13*, https://www.gs1.org/sites/default/files/docs/epc/GS1_EPC_TDS_i1_13.pdf.

[53] *UHF Gen2 Tag Air Interface v2.1.0*, https://www.gs1.org/sites/default/files/docs/epc/gs1-epc-gen2v2-uhf-airinterface_i21_r_2018-09-04.pdf.

[54] *Low Level Reader Protocol (LLRP) v1.1*, https://www.gs1.org/sites/default/files/docs/epc/llrp_1_1-standard-20101013.pdf.

[55] *Application Level Events (ALE) Specification v1.1.1 Part 1: Core*, https://www.gs1.org/sites/default/files/docs/epc/ale_1_ standard-core-20090313.pdf.

[56] *Application Level Events (ALE) Specification v1.1.1 Part 2: XML and SOAP Bindings*, https://www.gs1.org/sites/default/files/docs/epc/ale_1_1_1-standard-XMLandSOAPbindings-20090313.pdf.

[57] «EPC Information Services (EPCIS) Version 1.0.1 Specification,» en, p. 146,

[58] *Core Business Vocabulary (CBV) Standard v1.2.2*, https://www.gs1.org/sites/default/files/docs/epc/CBV-Standard-1-2-2-r-2017-10-12.pdf.

[59] *HF Class 1 Tag Air Interface v2.0.3*, https://www.gs1.org/sites/default/files/docs/epc/epcglobal_hf_2_0_3-standard-20110905r3.pdf.

[60] *Reader Management (RM) v1.0.1*, https://www.gs1.org/sites/default/files/docs/epc/rm_1_0_1-standard-20070531.pdf.

[61] *Discovery, Configuration, and Initialization (DCI) v1.0*, https://www.gs1.org/sites/default/files/docs/epc/dci_1_0-standard-20090610.pdf.

[62] E. Tdt, T. WGs e M. Harrison, *Tag Data Translation (TDT) v1.6*, en, https://www.gs1.org/sites/default/files/docs/epc/tdt_1_6_RatifiedStd-20111012-i2.pdf.

[63] *Information Services (EPCIS) Standard v1.2*, https://www.gs1.org/sites/default/files/docs/epc/EPCIS-Standard-1.2-r-2016-09-29.pdf.

[64] *Pedigree Standard v1.0 Jan 2007*, https://www.gs1.org/sites/default/files/docs/epc/pedigree_1_0-standard-20070105.pdf.

[65] *EPCglobal Certificate Profile Specification 2.0*, en, https://www.gs1.org/sites/default/files/docs/cert/cert_2_0-standard-20100610.pdf.

[66] K. Dean, *GS1 Object Name Service (ONS) v2.0.1*, en.

[67] joe.horwood, *GDSN Standards Maintenance Release v3.1.14*, en, https://www.gs1.org/standards/gdsn, Text.

[68] david.buckley, *GS1 Lightweight Verification Messaging*, en, https://www.gs1.org/verification-messaging, Text, jan. de 2019.

[69] Anonymous, *GS1 Electronic Data Interchange (EDI) Standards*, en, https://www.gs1.org/standards/edi, Text, dez. de 2014.

[70] david.buckley, *GS1 XML Standards 3.4.1*, en, https://www.gs1.org/standards/gs1-xml/3-4-1, Text, nov. de 2019.

[71] I. Mizutani e J. Mitsugi, «A Multicode and Portable RFID Tag Events Emulator for RFID Information System,» em *Proceedings of the 6th International Conference on the Internet of Things*, sér. IoT'16, New York, NY, USA: Association for Computing Machinery, 2016, pp. 187–188, ISBN: 978-1-4503-4814-0. DOI: 10.1145/2991561.2998470.

[72] *RFID EPC Gen2 Memory Bank Layout Including TID, User*, en-US.

[73] *TID Memory Maps for Monza Self-Serialization*, en-US, https://support.impinj.com/hc/en-us/articles/203444983.

[74] ISO, *ISO/IEC 15962:2013*, en, https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/04/34/43459.htm

[75] marco.santos.diamond, *FAQ's*, en, https://www.gs1.org/services/epc-encoder/faqs, Text, abr. de 2020.

[76] *GS1 General Specifications*, https://www.gs1.org/sites/default/files/docs/barcodes/GS1_General_Specifications.pdf.

[77] M. Mealling <michael@refactored-networks.com>, *A Uniform Resource Name Namespace for the EPCglobal Electronic Product Code (EPC) and Related Standards*, en, https://tools.ietf.org/html/rfc5134.

[78] *SGTIN INFO*, en-US.

[79] *GS1 GTIN Executive Summary*, https://www.gs1.org/docs/idkeys/GS1_GTIN_Executive_Summary.pdf.

[80] *GS1 GTIN Management Standard*, https://www.gs1.org/sites/default/files/docs/barcodes/GS1_GTIN_Management_Standa

[81] *GS1 Keys: Implementation Guidelines*, https://www.gs1.org/docs/tl/T_L_Keys_Implementation_Guideline.pdf.

[82] *Barcode_GS1_General_Specifications.Pdf*, https://www.gs1.org/sites/default/files/docs/barcodes/GS1_General_Specificatio

[83] *GS1 Barcode Chart*, https://www.gs1us.org/DesktopModules/Bring2mind/DMX/Download.aspx?Command=Core_Download

[84] *Listing Requirements: Product IDs (GTINs) - Amazon Seller Central*, https://sellercentral.amazon.com/gp/help/external/2003 US&ref=mpbc_200202190_cont_200317470.

[85] Anonymous, *GS1 Company Prefix - ID Keys | GS1*, en, https://www.gs1.org/standards/id-keys/company-prefix, Text, dez. de 2014.

[86] *Impinj LTK Programmers Guide*.

[87] *RFID in Retail: The Key Benefits for Retailers and Customers*, en-US, https://comparesoft.com/assets-tracking-software/retail/rfid-in-retail/.

[88] *EPCIS Guidelines*, https://www.gs1.org/docs/epc/EPCIS_Guideline.pdf.