

A Middleware Architecture for RFID-enabled traceability of air baggage

T. Bouhouche¹, A. Raghib², B. Abou El Majd², M. Bouya³, M. Boulmalf¹

¹ ITC Lab., International University of Rabat, Morocco

² LIMSAD Lab., Hassan II university of Casablanca, Morocco

³ REMA Lab., International University of Rabat, Morocco

Abstract. 1980 marked the start of a boom in radiofrequency identification (RFID) technology, initially associated with a growing need for traceability. In view of the technological progress and lower costs, RFID's area of application became much broader and, today, multiple business sectors take advantage of this technology. However, in order to achieve the maximum benefits of RFID technology, the data collected should be delivered in the best conditions to the whole applications that have need of its exploitation. For that, a dedicated middleware solution is required to ensure the collection of RFID data and their integration in information systems. The issues and key points of this integration as the description of the RFID technology will be summarized in the present paper, with a new middleware architecture. We focus mainly on components and the design of our middleware MedRFID, solution developed in our Lab, and which integrates mobility and provides extensibility, scalability, abstraction, ease of deployment and compatibility with IATA standards and EPCglobal standards. Moreover, we have developed an application (FindLuggage) allowing a real time tracking of luggage in the airport, based on the proposed middleware MedRFID.

1 Introduction

RADIO FREQUENCY IDENTIFICATION, or RFID, is a generic term for technologies that use radio waves to automatically identify people or objects [16]. This technology has recently seen growing interest from a wide range of industries such as retail, pharmaceutical, and logistics [18-21]. In these domains, RFID technology holds the promise eliminating many existing business problems by bridging the economically costly gap between the virtual world of IT systems and the real world of products and logistical units. Common benefits include more efficient material handling processes, elimination of manual inventory counts, and the automatic detection of empty shelves and expired products in retail stores. RFID technology has a number of advantages over other identification technologies. It does not require line of-sight alignment, multiple tags can be identified almost simultaneously, and the tags do not destroy the integrity or aesthetics of the original object. The location of tagged objects can thus be monitored automatically and continuously.

In traditional RFID applications, such as access control, there was a little need for an RFID middleware because the RFID readers were not networked and the RFID data were only consumed by a single application. The appearance of the first RFID middleware on the market enables the integration of RFID technology in various fields (Air, Logistic, etc). Nevertheless, the fact remains that they allow the identification of several technical

issues regarding this integration. In novel application domains, such as supply chain management and logistics, there is no longer a 1-to-1 relationship between reader and application instance. However, in these domains, many readers distributed across factories, warehouses, and distribution centers capture RFID data that need to be disseminated to a variety of applications. This introduces the need for an RFID infrastructure that hides proprietary reader device interfaces, provides configuration and system management of reader devices, and filters and aggregates the captured RFID data.

This paper has been organized in the following way: The first part is comprised of a presentation of the RFID systems. A description of the proposed MedRFID and the FindLuggage application are presented in the second part. The second and final part concludes the paper.

2 RFID Systems

The RFID systems basically consist of two or three elements: a tag/transponder and a reader for a Simplified RFID system, or a tag/transponder a reader and a middleware deployed at a host computer. The RFID tag is a data carrier part of the RFID system, which is placed on the objects to be uniquely identified. The RFID reader is a device that transmits and receives data through radio waves using the connected antennas. Its functions include powering the tag, and reading/writing data to the tag. As shown in Fig. 1, the signals sent by the reader's antennas

form an interrogation zone made up of an electromagnetic field. When a tag enters this zone, it gets activated to exchange data with the reader [2]. Later, the identification data read by the RFID reader is processed by the software system, known as the RFID middleware. The RFID middleware manages readers, as well as filters and formats the RFID raw tag data so that they can be accessed by the various interested enterprise applications [11]. Hence, the middleware is a key component for managing the flow of information between tag readers and enterprise applications [3].

Major advantages of using RFID as an auto-ID system are the following:

- RFID readers do not require a line of sight to access data from the RFID tags.
- RFID systems can read data over varied range from few centimeters to few hundred meters.
- RFID readers can interrogate, and make RFID tags readings much faster.
- RFID systems can read and write different sizes of data from reader / to the tag, based on the type of tag.
- RFID systems can read tags in harsh environments, without any human interference.

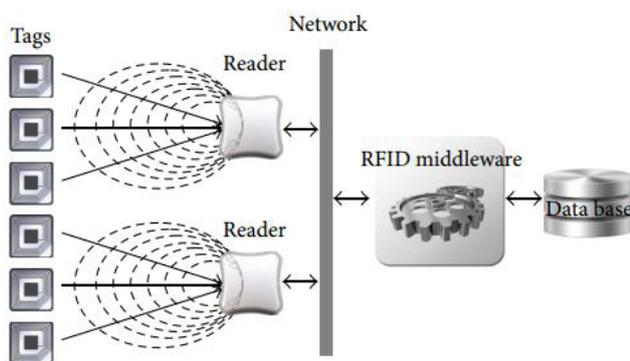


Figure 1. RFID System components.

Indeed, since the introduction of RFID technology in global solutions, a recurring issue is the integration of RFID in a transparent and efficient manner. As a result, there is a strong mobilization on the issue of research and development in mainly three areas:

➤ *Objects Identification:*

For a better identification of object, we need to focus the development on a standardization of RFID tags and a good transcription codes on RFID tags (Algorithm).

➤ *Communication between objects and middleware:*

The start point for this part is a good choice of useful RFID technologies, after a development of communication protocols between readers is necessary, following the middleware standardization can help us to done a good follow up with the objects, then the normalization of protocols shall needed for the better lopping of the communication steps.

➤ *The information processing:*

Decoding the RFID information collected (standard, algorithm) is important for structuring collected RFID information to make it usable directly by any application

of the information system, which need a real Time treatment volume (architecture allows scaling mass treatment of RFID data collected).

A. RFID System Components:

RFID systems are produced by many manufacturers and exist in countless variants. However, a RFID system consists mainly of three components; the transponder/tag, reader, and RFID middleware.

1) *RFID Transponder/Tag:*

A RFID transponder, or tag, consists of a chip and an antenna [22]. A chip can store a unique serial number or other information based on the tag's type of memory. The tag's type of memory can be read-only, read-write, or write-once and read-many [11]. Read-only tags are much cheaper to produce and are used in most current applications. Read-write tags are useful when information needs to be updated [2]. The antenna is used to transmit information from the chip to the reader, and the larger the antenna the longer the read range. The RFID tag can be either attached or embedded in an object to be identified, and can be scanned by mobile or stationary using radio waves [11, 14-17].

2) *RFID Reader:*

A RFID Reader is a scanning device that reliably reads the tags and communicates the results to the middleware. A reader uses its own antennae to communicate with the tag by broadcasting radio waves to which all tags within range will respond. Readers can process multiple items at once, allowing for increased read processing times. They can be either mobile or stationary, so that their optimal employment in the area can be found [12-13], and they are differentiated by their storage capacity, processing capability, and the frequency they can read [11].

Different designs of readers exist, because different applications have different requirements. RFID readers are classified into three types [2]:

- **OEM readers:** Original Equipment Manufacturers readers are mostly used for data capture systems, access control systems, and robots.
- **Industrial use readers:** used in assembly and manufacturing plant.
- **Portable readers:** These readers are more mobile than the other readers, and supported with a LCD display and keypad. This kind of readers is used in animal identification, device control and asset management applications.

3) *RFID Middleware :*

The middleware refers broadly to software or devices that connect RFID readers and the data they collect to enterprise information systems. RFID middleware helps make sense of RFID tag reads, applies filtering, formatting and logic to tag data captured by a reader, and provides this processed data to back-end applications [3]. RFID middleware serves in managing the flow of data between tag readers and enterprise applications, and is responsible for the quality, and therefore usability of the information. It provides readers with connectivity, context-based filtering and routing, and enterprise/B2B integration. RFID middleware design and components will be discussed further in the next sections.

When designing a RFID middleware solution, the following issues need to be considered:

- **Multiple hardware support:** The middleware must provide a common interface to access different kinds of hardware offering different features.
- **Synchronization and scheduling:** There should be intelligent scheduling and synchronization among all the processes of the middleware. This minimizes the latency and improves the efficiency of the middleware.
- **Real-time handling of incoming data from the RFID readers:** The middleware should handle the huge amount of data captured by the connected readers in real time without read misses.
- **Interfacing with multiple applications:** The middleware should be capable of interacting with multiple applications simultaneously, by catering to all the requirements of the applications with minimal latency.
- **Device neutral interface to the applications:** The application developer should only use the generic set of interfaces provided by the middleware independently of the type of hardware connected to the system.
- **Scalability:** The middleware design must allow easy integration of new hardware and data processing features.

B. RFID Middleware Components:

A RFID middleware is the interface that sits between the RFID hardware and RFID applications. It provides the following advantages:

- It hides the RFID hardware details from the applications.
- It handles and processes the raw RFID data before passing it as aggregated events to the applications.
- It provides an application level interface for managing RFID readers and querying the RFID data.

A layer of the RFID middleware incorporates all the device drivers of different hardware and exposes the application standard interfaces to access this hardware. If the application was provided with all the device drivers of all connected readers, it will be a hard job to manage and interface each of the devices. The application developer will then need to understand all the hardware-specific internals and operations. Also, the application, if provided with the huge amount of raw tag data reported by the readers, will find it very difficult to process the data in real time. A RFID middleware provides a standardized way of dealing with this flood of information, which processes the raw data and provides the application with clean and filtered data.

RFID middleware is generally composed of four major layers:

➤ *Reader Interface:*

The reader interface is the lowest layer of the RFID middleware which handles the interaction with the RFID hardware. It maintains the device drivers of all the

devices supported by the system, and manages all the hardware related parameters like reader protocol, air interface, and host-side communication.

➤ *Data Process or and Storage:*

The data processor and storage layer is responsible for processing and storing the raw data coming from the readers. Examples of processing logic carried by this layer are data filtering, aggregation, and transformation. This layer also processes the data level events associated with a specific application.

➤ *Application Interface:*

The application interface provides the application with an API to access, communicate, and configure the RFID middleware. It integrates the enterprise applications with the RFID middleware by translating the applications' requests to low level middleware commands.

➤ *Middleware Management:*

The middleware management layer helps managing the configuration of the RFID middleware, and provides the following capabilities:

- Add, configure, and modify connected RFID readers.
- Modify application level parameters such as filters, and duplicate removal-timing window.
- Add and remove services supported by the RFID middleware.

RFID readers are typically abstracted as a logical reader which is either a collection of several readers or a part of the reader. This grouping mechanism is used where there is a need to have a set of readers capturing data from a particular area such as a warehouse with many loading docks. The advantage of this is that the application can query a small number of logical readers rather than having to aggregate events from each of the individual readers.

There are two standardized interaction models used to define the communication between the middleware and the applications. An application can operate at synchronous mode when requesting services on demand or asynchronous mode when it registers for information to be sent to it when certain conditions are met. RFID middleware usually provide some kind of data filtering, because sometimes it might be required to report only certain type and value of the tag data to the application. The application must provide a set of defined patterns to the middleware. The middleware then allows only data that matches the pattern to be reported to the application. For example, if an application must only see tag data that starts with a specific pattern such as "XYZ20", the filter can be set to this value by the application and communicated to the middleware [2].

3 MEDRFID

3.1 Introduction: Why MedRFID?

Because some existing RFID middleware use a standard functionalities, our concern is to design and implement a specific middleware platform called "MedRFID, which is

distinguished by its innovative aspects in terms of architecture including mobility to enable the collection of information from mobile RFID readers, operating aspects of the data in real time with the ability to generate a business rules engine and especially by integrating communication protocol RFID readers disregarding the dependency to the initial manufacturer protocol. The implementation of this innovative architecture will give real flexibility for users of the middleware and above all a means of real-time processing a large volume of data that emanate from heterogeneous environments while ensuring ease of integration of these data and information and their best use by business applications systems.

3.2 MedRFID: Design and Architecture

RFID middleware is the core in the implementation of RFID technology in companies that have large amount of data to be processed and, above all, to share with the various components of the value chain and with its partners.

Thus, the successful integration of RFID within companies is conditioned by the successful design and implementation of RFID middleware. That is why our research team chose to design and implement a RFID middleware which integrates the latest innovations in the field.

MedRFID is a middleware that has been designed to meet the overall functionality that must contain a conventional RFID middleware, but it also contains areas of innovation mainly around the mobility and independence of manufacturers. It is an innovative device management and event processing platform at the edge of the enterprise. It is designed to provide a scalable, extensible platform for deployment, and management of rich RFID and sensor solutions.

As shown in Fig. 2, the MedRFID architecture is composed of seven major layers

2.1) Collect:

The collect module is the heart of the middleware. It's responsible of all collected tag's data, and it's a key instrument of the connection with the readers and the software intelligence. Moreover, it's used to control the readers, add data to the tags and format them in a single format regardless the used reader. To control the readers, this module has embedded an LLRP driver but also each driver needed for RFID readers, which don't support LLRP. These drivers are often provided by RFID manufacturer in different programming languages.

Each tag is formatted in an abstract layer in order to have the same format for all the tags whether it come from a mobile device, a LLRP or a specific reader (manufacturer driver).

2.1.1) LLRP:

LLRP (Low Level Reader Protocol) is a standard that facilitate the communication between the software and the readers. All the RFID readers speak the same language, so they can all understand the software whichever their types. This allows to the customers the choice of its hardware with the software. If they want to change the hardware, it could be without affecting the software solution. The collect process is as follow:

The software sends to the readers a RO Spec (Reader Operation spec). The RFID readers use these specs when they are enabled and when the start condition is reached. Then, the reader sends to the software a report of each tag or a list of them.

2.1.2) Custom Drivers:

Some readers do not support LLRP, but these readers may be interesting to support. Indeed, some of them are really good in RFID performance. Fortunately, readers manufacturer provide most often the driver in different languages to communicate with the reader. Languages mostly supported are C, C# and JAVA. So, to implement the driver for these readers, we have to develop a separate module which is an interface between the driver provided by manufacturer and the command provided by the middleware.

2.1.3) Mobile Readers:

Some of the mobile readers fully support LLRP and they can be used like fixed readers. For others, there are also drivers provided by manufacturers for control like the fixed RFID readers. These drivers can be embedded in a mobile app and transmit data to the middleware backend when it is connected to the cellular network.

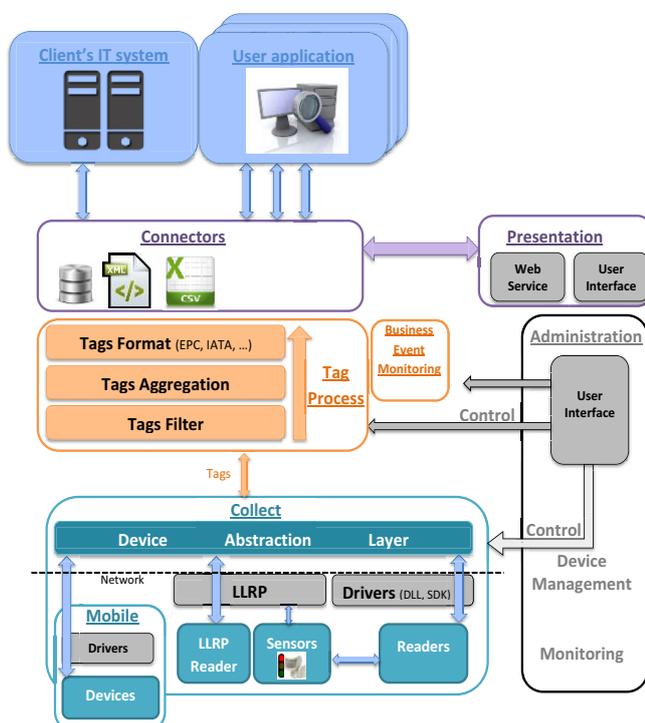


Figure 2. MedRFID Architecture

2.1.4) Sensors:

The middleware should also support Sensors. There are two types of sensors: IN Sensors and OUT Sensors.

IN Sensors are detectors, radars and are used to trigger the collection of readers. For example, when a tracked object is coming, IN Sensors detect it and trigger the start of the collection. With these sensors, it is possible to read only when there is something to track and facilitate the process (hardware + software).

OUT Sensors are used by users on the ground. It can be for example lights or alarms. They are used to secure the RFID process, for example, a green light may be switched on when a tag is read to show to the user that it can continue. Or a red light can be switched on if there is a problem with the tag read or if an access is forbidden to this user.

2.2) Mobile Application:

The software should include a mobile application. Mobile RFID readers allow RFID collect stationary objects. The reader is embedded in a mobile device with powerful RFID antennas and can communicate with the backend over cellular network.

Tags are stored directly in database if a connection is established with the backend or the mobile application stock tags read in the local memory if not. Like the collect module, tags are formatted in the same way in an abstract layer.

For the backend, reader device is transparent whether it is from a fixed reader or a mobile device. A mobile application is useful for huge objects to track and it is not possible to read them with a fixed reader.

2.3) Administration:

The administration module concerns the settings of the middleware. RFID Readers, antennas and sensors settings can be modified such as IP Address, Power, and numbers of antennas... In addition, each reader's RO Specs can be altered: Start/Stop trigger, numbers of tag reports...

Tag process preferences can also be modified, for example: Duration of the duplicate elimination process, decoding process.... Finally, Business rules can be added, edited or removed. Reader supervision will also be embedded in this module.

The supervision is a module which consists of logging each event about a reader. When something does not work properly, the middleware should send an alert (via mail, messages, SMS ...) to explain where the problem is.

Problems can be about the network, the reader, or its antennas. Ideally, the middleware should explain what type of alert it is and what the problem is. This is possible by doing multiple tests: For example, if there is no tag reads for X hours but the ping to the reader is OK, we may think it's a problem in the reader or its antennas (or both) but not a network issue.

2.4) Tag Process:

Each tag will go through a tag process that can have multiple operations such as filtering, eliminating duplicates, time stamping, contextualization and/or formatting.

The tag process is composed of three layers:

➤ Tags Filter:

The Tags filter is the layer which removes tags that we don't want to read. Each tag has a Unique ID and a User Memory where you can write personal data. This memory is often used to filter tags because every tag that we are interested in contains this value in its User Memory.

➤ Tags Aggregation:

The Tags aggregation is the mapping between the Tag Number (UID) and the business data we have in a database. The tag number is just an ID but it's often (always) linked to personal or business data. For example, the box with the RFID N°XXX (UID) contains three of YYY items (Business Data).

➤ Tags Format:

In some case, some tags are encoding following a strict format. So, the number contains data. There are multiple formats for RFID Tags. The middleware should include encoding and decoding algorithm in order to get back the information.

2.4.1) Duplicate Elimination:

The duplicate elimination is very important in an RFID application. It's often the first layer in the tag process in a middleware.

RFID readers can read more than 100 tags per second. Indeed, the same tag is read more than once but only one read is important during this second. So the duplicate elimination process permits to keep only one read of a tag during a customizable duration. With the duplicate elimination filter, the middleware eases the database from storing duplicate data. So, instead of 300 reads/writes in 3 seconds, there are only 3. Each tag will then be evaluated in a filter layer and then a business rules engine to redirect it if needed.

2.4.2) Filtering:

A filter defines rules that allows or stops tags during the tag process. Sometimes, multiple actors in the same location use RFID tags. In order to only keep the interesting tags, filter rules allows removing tags that we don't need to read. Filtering can be on specific serial numbers of tags, or in business number of a tag. Some tags are encoding following different norms. These norms such as IATA RP1740c or GRAI-96 encode business data in the tag. For example, IATA RP1740c encodes data about flights, so in the tag number is encoded: the company prefix, the date time of the flight, source and destination airports etc... By decoding the tag, it is possible to filter tags that have the company prefix for AIR France or other companies.

If a tag match the filter value, then the tag will be saved and continue through the tag process. Else, it is deleted

from the tag process. With the filtering, only interesting tags are collected and saved in the database.

2.4.3) Business Rules Engine:

Business rules are rules that match some tags or dates and practice different services according the tag. This allows for example to send data to a client for all the tags of its client but hide him the rest of the reads. It is possible to edit and add new rules to the engine. Then, rules are affected to the tags read events through the engine.

2.5) Connectors:

Tag reads, once being treated in the tag process and by the business rule engine, will then be transmitted via connectors such as Database connector, Files or Queues. The connectors allow external application to get the data provided by the middleware. External applications may be presentation software in order to show statistics, tag reads, etc. It can also be formatted files such as comma-separated files or web services in order to send the data to another software in order to automatically fulfill stocks etc.

2.6) The User Interface:

This is the view of the middleware. It is composed of three tabs: Reads, Writes and Administration.

The UI of the middleware shows tag reads in real time and permits the configuration of RFID readers / application.

The "Reads" view provides information about reads, EPC, date time, reader, antennas and ensures that the tag has been written into the database.

The user interface can also be developed by using the web service API.

The "Administration" view represents all the settings alterable for the middleware.

Reader settings: It comprises all the settings for each reader. How a read is triggered, how many antennas are connected to the reader, how long is the duplicate elimination.

The middleware settings represent all the settings alterable for the middleware itself: Connect or disconnect readers to the middleware, the SQL connection settings, and paths to different log files.

2.7) Client's IT System:

Each client of the middleware can be connected to it with the connectors and add data to its information system.

The middleware provides different connectors to transmit data to other information system. So, it is possible for external clients to develop a little interface in order to add the RFID data to their existing application or develop a new application by using these connectors.

The client's IT system is not a part of the middleware but we have to think it may be connected to it.

4 FIND LUGGAGE APPLICATION

With tens of millions of passengers traveling through the major European cities each year, through hundreds of check-in desks, luggage is extremely important for these airports. Traditional moving-belt bar code scanners require 'line of sight' in order to correctly read labels, but these could be hidden as the luggage moves on conveyor belts, rendering the luggage unidentifiable. This could slow the process down, as manual intervention is required to obtain the information and locate the luggage, and could directly contribute to misplaced or lost items.

In the airline industry, RFID considered as one of the best technologies that help the sector to remain competitive and innovative and to face the demands of increasingly cost and security, and increased traffic. This, to accommodate the needs of travelers but also to be operational.

Travelers are always afraid of luggage loss or theft. Now, RFID helps airports and airlines to limit this kind of risk, and meet the growing demands in the field of luggage handling (speed, security, cost control, etc.). As for industries, they benefit from changes in the direction of lower costs, faster processes, and improved communication.

The actors that interact directly with the system studied are:

- Holders of RFID tags: The luggage.
- RFID Readers: responsible for receive data, activation of RFID tags in each area, and communicate data to the middleware (MedRFID).
- MedRFID: collecting and sorting the data transmitted by the readers, to share with relevant the application or store them in a database.
- Employee: who may be responsible or single control officers who watch over the path of luggage to aircraft in good conditions, otherwise make alerts or even answer questions dedicated to specific luggage.

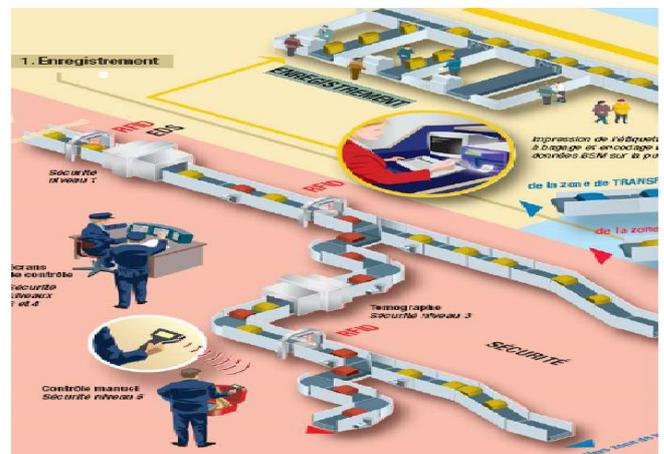


Figure 3. Example of platform tracking luggage in the airport.

Our study consists of the management and reporting of the processing of luggage immediately following

registration (check-in). The RFID tag is already encoded and contains luggage's data, printed and stored on it. Then, by using MedRFID, we will track the luggage at any time, and any different points in passages (the inputs / outputs of luggage sorting areas, security & control, loading / unloading, etc.) as shown for example in figure 3, which allows an automated traceability, reliable and in real-time for a good geo-location.

To properly manage followed, we offer an application that will be the RFID interface solution installed and designed for both the environment and luggage, in order to track, locate luggage in real time, and ensure safe movement of flights, as well as avoiding wastage of time lost in inventories.

When a passenger makes the check-in step, an RFID tag is attached on his luggage containing specific information (ticket number, destination, etc.). Exploitation of this tag allows identifying his luggage at various traffic areas along its presence in the airport.

After the passage of luggage in an area, the RFID reader passes to the tag an enough energy to its activation, with the aim to read its information or to add new data, by interacting in both cases with the MedRFID.

After receiving the luggage data, MedRFID is responsible for centralizing, sorting and filtering it for the needs of business applications, by providing understandable and reliable data to the other applications or database. Therefore, the user can easily locates a specific luggage, checks the safety, sorts, and even monitors it in a very small time. Our application deals with actual and real requirements by identifying the objectives, priorities, rules of management and key processes of the organization as shown in figure 4.

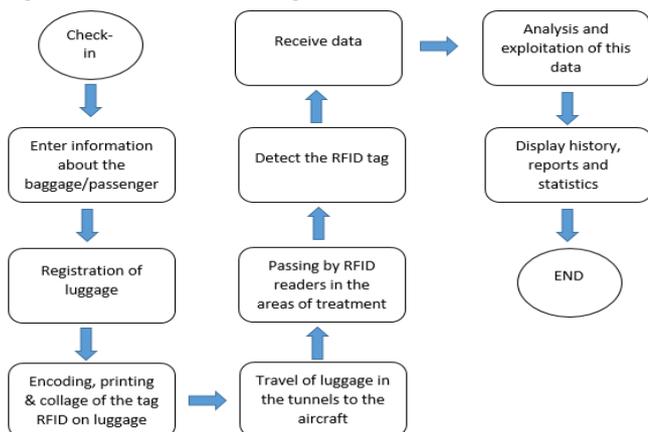


Figure 4. Luggage functional process.

The use of RFID traceability optimizes all stages of the processing airport, from check-in to loading bunkers namely: Registration luggage, security checks, sorting, reconciliation and loading. To overcome these problems and automate tracking of luggage, we decided to design and implement a web application: FindLuggage, based on the RFID solution and using the middleware MedRFID for tracking and monitoring a luggage in real time, and reducing the costs.



Figure 5. Communication between MedRFID and FindLuggage.

As shown in figure 5, the use of this service requires the implementation of a RFID reader at different points of traceability, and the placement of each piece on an RFID tag; this is followed by the use of MedRFID luggage tracking to collect information on these different points of traceability at the airport and save them in a database. Latterly, the application plays a role in integrating these data and providing information about the luggage. RFID readers can be deployed at both the check-in and the luggage collection stages, which enable to read the attached tags as it left the airport or, prior to collection by passengers. After the deployment of RFID readers, the middleware MedRFID collects traceability information on luggage at all tracking points within the airport and makes them available to the application via a database. The application FindLuggage operates and integrates these data and then uses them to be able to indicate the pathway of each luggage through the airport. It allows to find a misplaced luggage or give more details on the passage of a bag at a given collection point.

Vol / Bagages / Détails

Vol: 1 Avion: Vol1
 CASABLANCA ==> TEXAS
 Compagnie: China Southern
 Date Vol: 20/06/2013 12:45:00

ID_Tag	Date Detection	Tache	Emplacement	N° PassePort
B00001	20/06/2013 12:01:00	Enregistrement	E PT 1	P1
B00001	20/06/2013 12:06:00	Sécurité	S PT 1	P1
B00001	20/06/2013 12:09:00	Tri	T PT 1	P1
B00001	20/06/2013 12:13:00	Tri	T PT 1	P1
B00001	20/06/2013 12:26:00	Tri	T PT 2	P1
B00001	20/06/2013 12:28:00	Tri	T PT 3	P1
B00001	20/06/2013 12:31:00	Réconciliation	R PT 1	P1
B00001	20/06/2013 12:35:00	Avion	Chargement PT	P1
B00004	20/06/2013 12:20:00	Enregistrement	E PT 2	P4
B00004	20/06/2013 12:22:00	Sécurité	S PT 1	P4

1 2 3

Figure 6. MedRFID application : FindLuggage.

Traçabilité / Vol

ID_VOL	ID_AVION	DATE VOL	EMBARQUEMENT	DEBARQUEMENT	COMPAGNIE	Total Bagages	Bagages Chargés	Plus d'info
1	Vol1	20/06/2013 12:45:00	CASABLANCA	TEXAS	China Southern	4	3	Description
2	Vol2	20/06/2013 14:30:00	CASABLANCA	BORDEAU	Air France	2	0	Description

Figure 7. MedRFID application : FindLuggage.

As show in figures 6 and 7, the FindLuggage application facilitates access to information for the employee according to its type (lead or manager). He has the possibility to track the complete path of all flight's luggage or a specific one, using the flight number or the identifier of a / set of RFID tag. This possibility is also allowed to the passengers in order to have a vision on their luggage, but with very limited access to the consultation option. MedRFID's FindLuggage application has a several attractive features:

- It improves traceability due to better-read rates.
- An easy integration of many readers, allowing a real time tracking of luggage.
- Reducing labor costs.
- Reducing the number of lost or delayed luggage.
- Reducing the luggage handling time.
- Larger encoding capabilities: Passenger/ Luggage data traveling with luggage.
- Improves the comfort and safety of people in transit, but also to simplify their movement within airports.
- To have a global vision on luggage present in a specific area and manage it in real time.
- Generate reports and statistics of usage history.
- Filtering, aggregation and Fusion data from multiple readers in real time, Detect anomalies.
- Analysis and research related data in external databases.
- Etc.

These advantages make the application more robust by having two major concepts applied for the traceability of luggage. First, it provides a flexible and convenient structure for monitoring and management of luggage. Second, it can be adapted to the needs of airlines and passengers for an organization more personalized in the airport.

5 Conclusion

During the last decade, this wide spectrum of applications that have been observed is nothing but the result of the emergence of research and development, which were conducted in different areas such as logistics, air, and military to field security and personal services. It falls into the so-called Internet objects or connected objects. Connected objects and communicating could provide opportunities for the industry in the years to come, and a new middleware architecture become strategic regarding a massive rise of "unintelligent" internet of objects is expected by 2020. This work response to this need with a vision of connecting functions placed in objects.

An estimated 50 billion items could be tagged by 2020. This ability of communication will develop virtual reality devices. Thus research and innovation in the field of data integration collected by technologies that deal with

communication between objects is still needed to support this development and is a major factor for the success of these technologies and their relevance.

This paper introduces RFID technology and the role of RFID middleware in the enterprise information systems. It also describes the design of our middleware MedRFID and its architecture. With such an innovative architecture, MedRFID provides extensibility, scalability, abstraction and ease of deployment like the FindLuggage application demonstrates. To be compatible with different areas including logistics and airports, it integrates respectively EPCglobal standards and IATA standards making it easily deployed in heterogeneous sites.

References

1. Ajana, M. E., Boulmalf, M., Harroud, H. & Hamam, H. (2009). A Policy Based Event Management Middleware for Implementing RFID Applications, Proceedings of WiMOB 2009 5th International Conference on Wireless and Mobile Computing, Networking and Communications, ISBN 978-0-7695-3841 -9, Marrakesh, Morocco, October 12-14, 2009.
2. Al-Mousawi, H. (2004). Performance and Reliability of Radio Frequency Identification (RFID).
3. Burnell, J. (2008). What Is RFID Middleware and Where Is It Needed?, In: RFID Update.
4. Floerkemeier, C., Roduner, C. & Lampe, M. (2007). RFID Application Development with the Accada middleware Platform. IEEE Systems Journal, Vol.1 No.2, pp. 82-94, ISSN 1932- 8184.
5. Floerkemeier, C. & Lampe, M. (2005). RFID Middleware Design: Addressing Application Requirements and RFID Constraints, Proceedings of SOC'2005 Smart Objects Conference, pp. 219-224, ISBN 1 -59593-304-2, Grenoble, France, October, 2005.
6. Glasser, D. J., Goodman, K. W. & Einspruch, N. G. (2007). Chips, Tags and Scanners: Ethical Challenges for Radio Frequency Identification. Ethics and Information Technology, Vol.9, No.2, pp. 101 -109, ISSN 1388-1957.
7. Molnar, D. & Wagner, D. (2004). Privacy and Security in Library RFID: Issues, Practices, and Architectures, Proceedings of ACM CCS 2004 11th Conference on Computer and Communication Security, ISBN 1 -58113- 961 -6, Washington, DC, USA, October, 2004.
8. Prabhu, B. S., Su, X., Ramamurthy, H., Chu, C. & Gadh, R. (2005 a). WinRFID: A Middleware for the Enablement of Radio Frequency Identification (RFID) Based Applications, In: Wireless Internet for the Mobile Enterprise Consortium (WINMEC).
9. Prabhu, B. S., Su, X., Ramamurthy, H., Chu, P., Qiu, C. & Gadh, R. (2005 b). WinRFID: Middleware for Distributed RFID Infrastructure, In: Wireless Internet for the Mobile Enterprise Consortium (WINMEC).
10. Sheng, Q. Z., Li, X. & Zeadally, S. (2008). Enabling Next-Generation RFID Applications: Solutions and

- Challenges. IEEE Computer, Vol.41, No.9, pp. 21 - 28, ISSN 0018-9162.
11. United States Government Accountability Office (2005). Information Security Radio Frequency Identification Technology in the Federal Government.
 12. A.Raghib, B.Abou El Majd (2014).”Multi-level optimal deployment of RFID readers by using particle swarm optimization”, Proceedings of META'14 5th International Conference on Metaheuristics and Nature Inspired Computing, Marrakesh, Morocco, October 27-31, 2014.
 13. Indrajit Bhattacharya, Uttam Kumar Roy, “Optimal Placement of Readers in an RFID Network Using Particle Swarm Optimization,” International Journal of Computer Networks & Communications– IJCNS, vol.2, no.6, November 2010.
 14. E. Jaselskis, T.El-Misalami, Implementing radio frequency identification in the construction process, Journal of Construction Engineering and Management (ASCE) 129 (6) (2003) 680–688.
 15. S. Chae, T. Yoshida, Application of RFID Technology to prevention of collision accident with heavy equipment, Journal of Automation in Construction 19 (2010) 368–374.
 16. C. Legner, F. Thiesse, RFID-Based maintenance at Frankfurt airport, IEEE Pervasive Computing 5 (1) (2006) 34–39.
 17. R. Want. An introduction to RFID technology. IEEE Pervasive Computing, 5(1):25–33, Jan.-March 2006.
 18. G. M. Gaukler, “Item-level RFID in a retail supply chain with stock-out based substitution,” IEEE Transactions on Industrial Informatics, vol. 7, no. 2, pp. 362-370, 2011.
 19. T.-M. Choi, “Coordination and risk analysis of VMI supply chains with RFID technology,” IEEE Transactions on Industrial Informatics, vol. 7, no. 3, pp. 497-504, 2011.
 20. J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. J. Gonzalez Castano, and J. Alcaraz, “Multiframe maximum-likelihood tag estimation for RFID anticollision protocols,” IEEE Transactions on Industrial Informatics, vol. 7, no. 3, pp. 487-496, 2011.
 21. Y.-H. Chen, S.-J. Horng, R.-S. Run, J.-L. Lai, R.-J. Chen, W.-C. Chen, Y. Pan, and T. Takao, “A novel anti-collision algorithm in RFID systems for identifying passive tags,” IEEE Transactions on Industrial Informatics, vol. 6, no. 1, pp.105-121, 2010.
 22. Roy W., Daniel M.R., “Ubiquitous Electronic Tagging,” IEEE Distributed Systems Online, Vol. 1, No 2, Jan. 2004.