# RFID APPLICATION TO AIRPORT LUGGAGE TRACKING AS a GREEN LOGISTICS APPROACH

[1,2]Yassir Rouchdi, [1]Achraf Haibi, [1]Khalid El Yassini
[1]IA Laboratory, Faculty of Sciences Meknes, Moulay Ismail University, Morocco
yassir.rouchdi@uir.ac.com
Achraf.haibi@gmail.com
Khalid.elyassini@gmail.com

[2]Mohammed Boulmalf and [2,1]Kenza Oufaska
[2]International University of Rabat, TIC Lab, Rabat, Morocco
Mohammed.boulmalf@uir.ac.ma
Kenza.oufaska@uir.ac.ma

*Abstract*— **the purpose of this study is to enhance RFID application benefits as a green Airport luggage tracking system. Firstly, by defining RFID architecture, components, functioning and middleware roles. Secondly, by discussing the implementation of Role-Based Access Control as a tool regulating access to RFID data, therefore making authentication methods more robust and flexible. To eventually presenting our UIR middleware solution and BAGTRAC application, allowing easier manipulation and real-time visualization of the luggage transportation process.**

Keywords— ***RFID, Middleware, Role-Based Access Control, Green Logistics.***

## I. INTRODUCTION

RFID technology has grown considerably in recent decades. The rapid advances of microelectronic transceivers have reduced the size and cost of HF and UHF RFID infrastructure, allowing longer and faster reading rates than ever before. RFID technology is now able to cope with new applications with greater mobility using a large number of components, allowing specific functionalities and general services and offering important advantages over other identification mechanisms [1].

The main objective of this study is to apply Radio Frequency Identification as a luggage tracking technology, the application have been done before, but part of this approach was to fix privacy and security issues related to RFID, by enhancing authentication protocols in existing solutions.

As to managing the tracking system, we built a middleware, but instead of building our architecture from scratch, its design is built according to already developed RFID standards, leading to a framework suitable for both RFID and WSN integration applications. Allowing adoption of RBAC model as a tool regulating access to data between 'Data and Event Management' and 'Application Abstraction' layers, leading to resolving accessibility and authorization problems occurring in anterior RFID middleware solutions.

The paper is presented as follows, first, an introduction to RFID technology, middleware definition, roles and existing examples. Secondly, a presentation of our middleware solution, followed by the definition of RBAC model, along with its mathematical formula and syntax. Then, we define used technologies, and present an RBAC authentication test example. Finally yet importantly, we talk about our Back-end application BagTrac, defining its architecture and capabilities, to closing the work with a general conclusion and perspectives.

## II. RFID TECHNOLOGY

### A. components

RFID systems are basically composed of three elements: a tag, 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 [7]. Unique identification or electronic data stored in RFID tags can be consisting of serial numbers, security codes, product codes and other specific data related to the tagged object. The available RFID tags in today's market could be classified with respect to different parameters. For example with respect to powering, tags may be passive, semi-passive, and active. In terms of access to memory, the tags may be read-only, read-write, Electrically Erasable Programmable Read-Only Memory, Static Random Access Memory, and Write-once read-many. Tags have also various sizes, shapes, and may be classified with respect to these geometrical parameters. The RFID reader is a device that transmits and receives data through radio waves using the connected antennas. RFID reader can read multiple tags simultaneously without line-of-sight requirement, even when tagged objects are embedded inside packaging, or even when the tag is embedded inside an object itself. RFID readers may be either fixed or handheld, and are now equipped with tag collision,

978-1-5386-4385-3/18/$31.00 ©2018 IEEE

reader collision prevention and tag-reader authentication techniques [8].

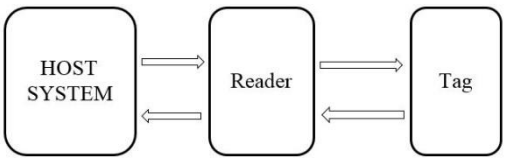



Figure 1. RFID Components

### B. Frequency Characteristics

Frequency refers to the size of the radio waves used to communicate between RFID systems components. RFID systems throughout the world operate in low frequency (LF), high frequency (HF) and ultra-high frequency (UHF) bands. Radio waves behave differently at each of these frequencies with advantages and disadvantages associated with using each frequency band. If an RFID system operates at a lower frequency, it has a shorter read range and slower data read rate, but increased capabilities for reading near or on metal or liquid surfaces. If a system operates at a higher frequency, it generally has faster data transfer rates and longer read ranges than lower frequency systems, but more sensitivity to radio wave interference caused by liquids and metals in the environment [13].

## III. RFID MIDDLEWARE

Radio Frequency Identification (RFID) technology holds the promise to automatically and inexpensively track items as they move through the supply chain. The proliferation of RFID tags and readers will require dedicated middleware solutions that manage readers and process the vast amount of captured data [5]. The efficiency of an RFID application depends on the precision of its hardware components, and the reliability of its middleware which is the computer software that provides services to software applications beyond those available from the operating system [6].

Middleware makes it easier for software developers to perform communication and input/output, so they can focus on the specific purpose of their application. Middleware includes Web servers, application servers, content management systems, and similar tools that support application development and delivery. It is especially integral to information technology based on Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web services, SOA, Web 2.0 infrastructure, and Lightweight Directory Access Protocol (LDAP) [14, 15].

### A. Middleware's basic functions

The three primary functions of an RFID middleware can be broadly classified as device integration (that is, connecting to devices, communicating with them in their prescribed protocols and interpreting the data). Filtering (the elimination of duplicate or junk data, which can result from a variety of sources, for example: the same tag being read continuously or spikes or phantom reads caused by interference) and feeding applications with relevant information based on the information collected from devices after properly performing the appropriate conversions and formatting [7].

Even though most RFID Middlewares share the same clear basic functions, every middleware has an architecture of its own, which is a direct result to the absence of an architecture standardization.

### B. Security and privacy issues

With the adoption of RFID technology, a variety of security and privacy risks need to be addressed by both organizations and individuals. RFID tags are considered "dumb" devices, in that they can only listen and respond, no matter who sends the request signal. This brings up risks of unauthorized access and modification of tag data [16]. In other words, unprotected tags may be vulnerable to eavesdropping, traffic analysis, spoofing or denial of service attacks.

- Eavesdropping (or Skimming): Radio signals transmitted from the tag, and the reader, can be detected several meters away by other radio receivers. It is possible therefore for an unauthorized user to gain access to the data contained in RFID tags if legitimate transmissions are not properly protected. Any person who has their own RFID reader may interrogate tags lacking adequate access controls, and eavesdrop on tag contents. Traffic Analysis: Even if tag data is protected, it is possible to use traffic analysis tools to track predictable tag responses over time. Correlating and analyzing the data could build a picture of movement, social interactions and financial transactions. Abuse of the traffic analysis would have a direct impact on privacy [16, 17].
- Denial of Service Attack: the problems surrounding security and trust are greatly increased when large volumes of internal RFID data are shared among business partners. A denial of service attack on RFID infrastructure could happen if a large batch of tags has been corrupted. For example, an attacker can use the "kill" command, implemented in RFID tags, to make the tags permanently inoperative if they gain password access to the tags. In addition, an attacker could use an illegal high power radio frequency (RF) transmitter in an attempt to jam frequencies used by the RFID system, bringing the whole system to a halt [16, 17].
- Personal Privacy as RFID is increasingly being used in the retailing and manufacturing sectors, the widespread item-level RFID tagging of products such as clothing and electronics raises public concerns regarding personal privacy. People are concerned about how their data is being used, whether they are subject to more direct marketing, or whether they can be physically tracked by RFID chips. If personal identities can be linked to a unique RFID tag, individuals could be profiled and tracked without their knowledge or consent [16, 17].

### C. Related Work & Contribution

In the RFID domain, Savant middleware is a successful implementation of the EPC network. Currently, many of the

large IT companies already offer commercial RFID software, such as SUN EPC Network and IBM WebSphere RFID Premises Server. More recently, Complex Event Processing technology was used in several RFID middleware systems, specifying that event-processing language have been adopted to define complex events. In this paper, we will be applying CEP to define unions and intersections of both RFID and WSN simple events, resulting as complex events [1, 3].

To clarify the contribution of this paper, we state that first, it proposes a new approach, instead of building our architecture from scratch, UIR middleware design is built according to already developed RFID standards, leading to a framework suitable for diverse applications. Secondly, it declares the adoption of RBAC model as a tool regulating access to data between Data & Event Management, and Application Abstraction layers, resolving accessibility and authorization problems occurring in anterior RFID middleware. Finally, the application proposed is just a basic example to test our middleware - which is still under development and improvement - and does not express the wide range of applications our middleware could handle, hence its adaptability aspect.

## IV. ROLE-BASED ACCESS CONTROL (RBAC)

In order to resolve the security and privacy issues and prevent the RFID Tag data, we decided to use the role based access control regulation method, so that only authorized users get access to specific data.

RBAC is a method of regulating access to computer or network resources based on the roles of individual users within a network [18]. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file. Roles are defined according to authority and responsibility within the Network [19]. To clarify the notions presented in the previous section, we give a simple formal description, in terms of sets and relations, of role based access control. No particular implementation mechanism is implied. For each subject, the active role is the one that the subject is currently using:

AR(s: subject) = {the active role for subject s}.

Each subject may be authorized to perform one or more roles:

RA(s: subject) = {authorized roles for subject s}.

Each role may be authorized to perform one or more transactions:

TA(r: role) = {transactions authorized for role r}.

Subjects may execute transactions.

The predicate exec(s,t) is true if subject ‘s’ can execute transaction ‘t’ at the current time, otherwise it is false:

Exec(s: subject, t: tran) = true if subject s can execute transaction t.

### *A. RBAC primary rules*

- Role assignment: A subject can exercise a permission only if the subject has selected or been assigned a role.
  ∀s:subject,t :tran (, exec(s,t) ⇒ AR(s) ≠O/ ).
- Role authorization: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
  ∀s: subject (, AR(s) ⊆ RA(s)).
- Permission authorization: A subject can exercise a permission only if the permission is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can exercise only permissions for which they are authorized.
  ∀s: subject,t : tran (, exec(s,t) ⇒ t ∈TA(AR(s))).

### *B. RBAC security Implementation*

A properly administered RBAC system enables users to carry out a broad range of authorized operations, and provides great flexibility and breadth of application. System administrators can control access at a level of abstraction that is natural to the way that enterprises typically conduct business. This is achieved by statically and dynamically regulating users, actions through the establishment and definition of roles, role hierarchies, relationships, and constraints [19] [20].

In our case, security issues related to data access occur when backend end applications require information they are unauthorized to get. Where comes the necessity of applying RBAC. The implementation of an RBAC model in middleware security is not as simple as it seems, findings indicate that many well known middleware technologies under study fall short of supporting RBAC. Custom extensions are necessary in order for implementations compliant with each middleware to support RBAC required or optional components. Some of the limitations preventing support of RBAC are due to the middleware’s architectural design decisions; however, fundamental limitations exist due to the impracticality of some aspects of the RBAC standard itself [9, 10].

### *C. RBAC implemented Syntax*

Assignment of authorizations

```
public boolean autorisation(){
if ((ReaderID==1 && ReaderIPAddr =="192.168.1.3") ||
(ReaderID==2 && ReaderIPAddr =="192.168.1.4"){
        permission=true;
        role1=true;
        role2=true;
        role3=true;}
else{
        permission=false;
        role1=false;
        role2=false;
        role3=false;}
return permission;
                }
```

Affectation des authorisations

```
if (permission==true){
        if (role1==true){fonction 1}
        if (role2==true){fonction 2}
```

```
        if (role3==true){fonction 3}
else
        { System.out.println("Permission denied");}
                                }
```

## V. UIR MIDDLEWARE

### A. UIR Architecture

We propose to develop an RFID middleware called UIR, bearing in mind the design problems discussed in the second section. Our system is organized as a three-tiered architecture, with back-end applications (BagTrac), middleware (UIR) and RFID hardware.

UIR middleware offers a design that provides the application with a neutral device protocol and an independent platform interface. It integrates three hardware abstraction layer (HAL), event and data management layer (EDML) and Application Abstraction Layer (AAL).

### B. Hardware Abstraction Layer

HAL is the lowest layer of (UIR-) and is responsible for interaction with the hardware. It allows access to devices and tags in an independent manner of their various characteristics through layers of tag abstraction and reader.

The reader abstraction provides a common interface for accessing hardware devices with different characteristics such as protocols (ISO 14443, EPC Gen2, ISO 15693), UHF (HF) and host side interface Interface (RS232, USB, Ethernet).

The abstraction of the reader exposes simple functions such as opening, closing, reading, writing, etc. To accomplish complex operations of the readers.

The abstraction of readers and tags in UIR- make it extensible to support various tags, readers and sensors.

The device management module in HAL is responsible for the dynamic loading and unloading of the reader libraries depending on the use of device hardware. This allows the system to be light because only the required libraries are loaded. This layer contains the devices for various operations, as specified by the upper layers. It is also responsible for monitoring and reporting the status of the device. Some of the functions provided by the HAL to access RFID hardware are as follows:

• The Device-opening: function is responsible for opening a connection with the device. The connection parameters are provided as an argument to this function. When a successful connection is made to the reader, a response is returned by this function. This response is then used as a reference to access the device in subsequent calls.

• The Device-reading function: reads data from the internal reader. The read parameters such as the protocol to be read by the reader, the size of the data to be read, are specified as arguments of this function. The function responds successfully if valid data is present in the reader if not with an error code.

• The Device-Writing function writes data to the Tag. Arguments Specified with this function, the unique ID partially or totally, which triggers the data to be written to the tag. The function responds successfully if the data is written to the Tag or returns an error code (for example, when the tag is not identified only).[1, 3]

### C. Event and Data Management Layer (EDML)

EDML handles various reader-level operations, such as reading tags and informing readers of disconnected notions such as device failure, write failure, and so on. The layer acts as a conduit between the hardware abstraction layer (HAL) and the application abstraction layer (AAL). It accepts commands from AAL, processes them and therefore issues commands to HAL. Similarly, the responses are brought from the HAL, processed and transmitted to the LAA by this layer.

The EDML is the kernel of this middleware. It filters out uninteresting data, formats the remaining useful data and builds complex events according to real-time specifications. The event specification analyzer interprets and transforms event specifications into four processes steps: filtering, grouping, aggregating, and complex construction of events. The volume of event data is very important in the NSE middleware system. The filter selects only those events in the upper layer, thus reducing the reports data dramatically. In the ratio to the upper layer, event data are separated in several groups for a clear demonstration. The aggregation provides statistical information event data. . By aggregating, the volume of the declared data may be reduced again.

### D. Application Abstraction Layer (AAL)

The Application Abstraction Layer (AAL) provides various applications with an independent interface to RFID hardware. The interface is designed as an API by which Applications use UIR-RFID services. All operations at the application level such as reading, writing, etc. Are interpreted and translated into the lower layers of UIR- by the AAL. In order to restrain unauthorized back-end application from getting acces to Data, we used Role-Based access control method of regulating access to guarantee data protection from unauthorized back-end applications.

The next figure (Fig. 2) illustrates UIR-RFID architecture.

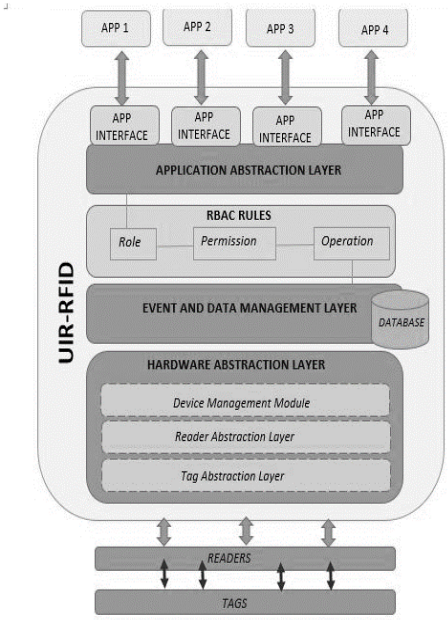



Figure 2. UIR Middleware Architecture

### *E. UIR-RFID Implementation*

For the UIR-RFID implementation, we propose the use of Microsoft Visual Studio .Net 2010 as Framework and development tool. The reasons for this choice are the powerful utilities for Application Development that this framework provides. The code to use to develop the Project is C Sharp (C #). We propose the use of graphical user interface features provided by the .Net Framework and Microsoft Visio 2013 to develop conceptual models and middleware architecture. For the purpose of data management and storage, we offer Microsoft SQL Server 2008, It is a cohesive set of tools, utilities and interfaces collaborating to provide excellent data management. The database schema generated by this DBMS provides a comprehensive view of the data and its relationships. To view the database, retrieve, modify, delete, and store data, we propose the use of the SQL language (Structured Query Language) [1, 20].

## VI. USED TECHNOLOGIES AND TEST

### *A. Eclipse*

Eclipse IDE is a free, integrated development environment (the Eclipse term also refers to the corresponding IBM-initiated project) that is extensible, universal, and versatile, potentially enabling the creation of development projects that implement any programming language. Eclipse IDE is mainly written in Java (using the SWT graphical library, IBM), and this language, thanks to specific libraries, is also used to write extensions.

The specificity of Eclipse IDE comes from the fact that its architecture is developed around the notion of plug-in (in accordance with the OSGi standard): all the functionalities of this software workshop are developed as a plug-in.

Many commercial software packages are based on this free software, such as IBM Lotus Notes 8, IBM Symphony or Websphere Studio Application Developer.Units.

A. 1. Supported programming languages

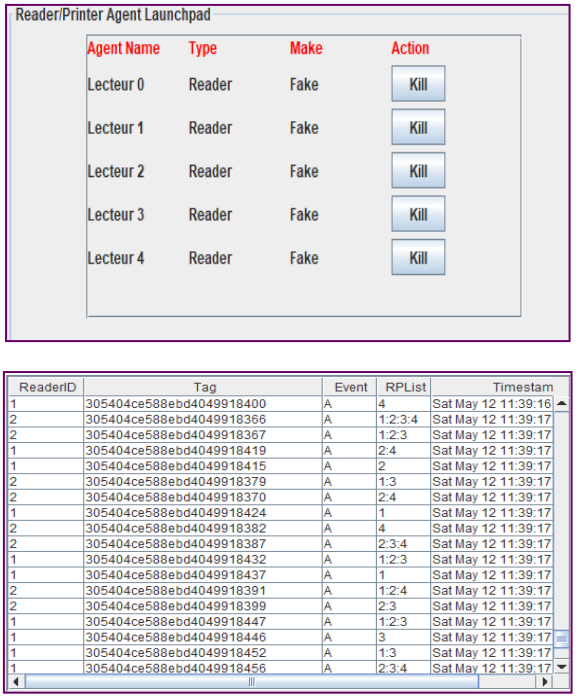
Numerous languages are already supported (most thanks to the addition of plug-ins), among which: Java, RPG for system I, C #, C ++, C, Objective Caml, Python, Perl, Ruby, COBOL, Pascal, PHP, Javascript, XML, HTML, XUL, SQL, ActionScript, Coldfusion.

### *B. Abstract Application Test*

TagCentric: is an open source RFID middleware that controls heterogeneous RFID devices and collects RFID-related data into a user-specified database. Its cost (free!) And its simplicity make it ideal for small businesses, RFID test centers and universities.

The goal of this project is to extend TagCentric by developing and integrating the RBAC model.

For our test, we add and activate five readers, and using the RBAC model that we added only the "authorized" readers begin to send data -For our example the readers 1 and 2 are allowed-.




| ReaderID | Tag | Event | RPList | Timestam |
|---|---|---|---|---|
| 1 | 305404ce588ebd4049918400 | A | 4 | Sat May 12 11:39:16 |
| 2 | 305404ce588ebd4049918366 | A | 1:2:3:4 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918367 | A | 1:2:3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918419 | A | 2:4 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918415 | A | 2 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918379 | A | 1:3 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918370 | A | 2:4 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918424 | A | 1 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918382 | A | 4 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918387 | A | 2:3:4 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918432 | A | 1:2:3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918437 | A | 1 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918391 | A | 1:2:4 | Sat May 12 11:39:17 |
| 2 | 305404ce588ebd4049918399 | A | 2:3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918447 | A | 1:2:3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918446 | A | 3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918452 | A | 1:3 | Sat May 12 11:39:17 |
| 1 | 305404ce588ebd4049918456 | A | 2:3:4 | Sat May 12 11:39:17 |

Fig. 3. launch of readers and Test

## VII. BAGTRAC APPLICATION

### *A. Introduction: Why BagTrac?*

The purpose of this embedded application is to collect information transmitted by an RFID reader, and send them to a database in real time to process and transmit them to an application installed in a mobile terminal, working with the

Android operating system, to allow a tracking of the position of luggage at airports.

*B. BagTrac: Architecture*

First, an RFID Tag is fixed on a bag or a case, by using wireless communication, the reader disposed on the path of the bag, follows its position throughout the transportation process, by sending the retrieved id of the tag to the Arduino board.

Next, it establishes a connection with the database via its USB port through a Java application, to store the location of the bag.

Finally, the mobile application communicates with the database to display to the user the location of his suitcase. The position is then collected, centralized in real time and transmitted to the corresponding user. [Fig.4].

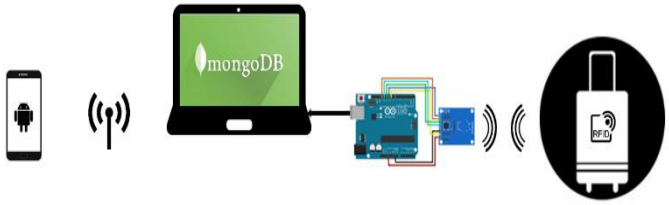



Fig. 4. Project architecture

*C. Android application*

This application is dedicated to users to provide the location of luggage to each customer, it offers different interfaces.

C.1. Start interface

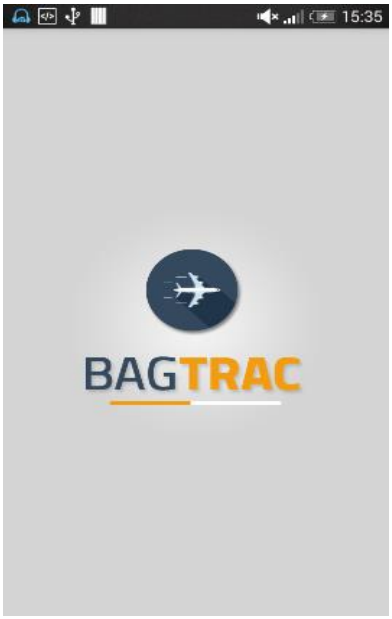



Fig 5. Start interface

C.2. Login space


Fig 6. Login space

In case of entering an incorrect identifier or password, the identification interface generates an error message inviting the user to retype the identifiers [Fig.7] [Fig.8].


Fig. 7. Incorrect identifier


Fig. 8. Incorrect password

C.3. Client Space

After the authentication, the application makes a connection with the database to retrieve the name of the corresponding user and the location of his luggage if it was found, [Fig.9], otherwise a message will appear to warn him that there is no checked baggage [Fig.10].


Fig. 9. Luggage location


Fig. 10. No luggage

## VIII. Green Logistics

Green logistics describes all attempts to measure and minimize the ecological impact of logistics activities. This includes all activities of the forward and reverse flows of products, information and services between the point of origin and the point of consumption. It is the aim to create a sustainable company value using a balance of economic and environmental efficiency. Green logistics have its origin in the mid-1980s and was a concept to characterize logistics systems and approaches that use advanced technology and equipment to minimize environmental damage during operations.

### *A. RFID application Impact on environment*

RFID has shown its efficiency when applied to airports luggage tracking systems. However, we also see an environmental dimension to this effort. Predicting that in helping to tracking passenger's luggage more accurately, RFID will improve sustainability by reducing unnecessary luggage truck deliveries, as well as passenger's trips to the airport for lost items, and therefore reducing air pollution in the area. Moreover, indirect impacts of RFID application can be more significant if analyzed properly. Lost luggage or baggage arrival delay, means more waiting time for passengers, therefore a delay for all their planned activities ( postponed meetings, multiple transportations, canceled duties… ), and its impact on time, place and energy consummation, touching not only the concerned person, but all the little network around him. On a bigger scale network, these situations deteriorative impacts on the environment can be more remarkable.

## IX. Conclusion:

Our proposed middleware UIR architecture offered a solution to many issues discussed in earlier Sections, starting by resolving the multiple hardware support issue on the reader abstraction layer that provides a common interface for accessing RFID hardware devices, with different characteristics. It also optimized synchronization and scheduling in the middleware on the EDML, which manages data flow between the other layers, handling various reader-level operations, servicing multiple applications and offering a device neutral interface to the applications. In addition to that, the Application Abstraction Layer (AAL) provides various applications with an independent interface to RFID hardware, resolving scalability problems on the Hardware Abstraction Layer that allows access to devices and tags in an independent manner of their various characteristics, through layers of tag and reader abstraction. While regulating access by using RBAC Mechanism to make sure only authorized users (applications) access the needed data, depending on the permissions allowed and the role assigned. The environmental impact was cited in a brief qualitative manner, but further work will be invested in a way to get concrete quantitative results and data.

In order to cover all aspects of the luggage-tracking scenario, our RFID Tracking system contains three major parts, RFID hardware, insuring the localization of bags, UIR RFID middleware, which is responsible for collecting, filtering and aggregating Data, and finally, back-end application BagTrac, that allows real-time visualization of the bag transportation process.

As for future and actual work, we intend to enhance our middleware architecture in order to offer a solution to many RFID related issues. Starting by resolving the multiple hardware support issue on the reader abstraction layer, and also, optimizing synchronization and scheduling in the middleware EDML, which manages data flow between the other layers, handling various reader level operations, servicing multiple applications and offering a device neutral interface to the applications.

## References


[1] Y. Rouchdi , K. El Yassini and K. Oufaska, "Complex Event Processing and Role-Based Access Control Implementation in ESN Middleware", Innovations in Smart Cities and Applications, LNNS 37, Springer, pp. 966–975, 2018

[2] United States Government Accountability Office, "Information Security Radio Frequency Identification Technology in the Federal Government," United States Government Accountability Office, 2005

[3] Q. Sheng, X. Li, and S. Zeadally, "Enabling Next-Generation RFID Applications: Solutions and Challenges", IEEE Computer, Vol. 41 (9), 2008

[4] H. Al-Mousawi, "Performance and reliability of Radio Frequency Identification (RFID)", in Agder University College, 2004

[5] N. Kefalakis, N. Leontiadis, J. Soldatos and D. Donsez, "Middleware Building Blocks for Architecting RFID Systems", Mobile Lightweight Wireless Systems, Vol. 13 (9), pp. 325-336, 2009

[6] X. Su, C. Chu, B.S. Prabhu and Rajit Gadh, "On the creation of Automatic Identification and Data Capture infrastructure via RFID and other technologies" Taylor & Francis Group, 2007

[7] M C. Bornhövd, T. Lin, S. Haller and J. Schaper, "Integrating Automatic Data Acquisition with Business Processes - Experiences with SAP's Auto-ID Infrastructure". In Proceedings of the 30st international conference on very large data bases (VLDB). Toronto, pp. 1182–1188, 2004

[8] S. Bell , "RFID Technology and Applications", Cambridge University Press, pp. 6–8, London, 2011

[9] M. Catherine O'Connor, "Rfid is the Key to Car Clubs Success" , RFID JOURNAL, 2011

[10] R. Russell, "Manufacturing Execution Systems: Moving to the next level", Pharmaceutical Technology, pp. 38–50 , 2004

[11] M. Darwish, "Analysis of ANSI RBAC Support in Commercial Middleware", PhD Thesis, University of British Columbia, Vancouver, Canada, 2009

[12] R. Ferraiolo, D.F. Kuhn and D.R. Sandhu "The NIST Model for Role-Based Access Control: Toward a Unified Standard", 5th ACM Workshop Role-Based Access Control, pp. 47–63, 2000

[13] W. Weixin, S.Jongwoo & K. Daeyoung. "Complex event processing in EPC sensor network middleware for both RFID and WSN", 2008

[14] M. Zuluaga, J. Montanez and J. van Hoof, "Green Logistics – Global Practices and their Implementation in Emerging Markets", pp. 2–3, Colombia, 2011

[15] T. Bouhouche, M. A. El Khaddar, M. Boulmalf, M. Bouya and M. Elkoutbi, "A new middleware architecture for the integration of RFID technology into information systems," International Conference on Multimedia Computing and Systems (ICMCS), pp. 1025-1030, Marrakech, 2014

[16] Y. Rouchdi , K. El Yassini and K. Oufaska, "Resolving Security and Privacy Issues in Radio Frequency Identification Middleware", International Journal of Innovative Science, Engineering & Technology (IJISET), Vol. 5(2), pp. 2348-7968, 2018

[17] T. Zhang, Y. Ouyang and Y. He, “Traceable Air Baggage Handling System Based on RFID Tags in the Airport”, School of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, China, Journal of Theoretical and Applied Electronic Commerce Research, Vol. 3 (1) , pp. 106–115, 2008

[18] R. Weil and E. Coyne, "ABAC and RBAC: Scalable, Flexible, and Auditable Access Management", IT Professional, vol. 15 (4), pp. 14-16, 2013

[19] J. Caiyuan, S. Aodong and Y. Wenxue, “The RBAC System Based on Role Risk and User Trust”, International Journal of Computer and Communication Engineering, 2016

[20] Y. Rouchdi , K. El Yassini and K. Oufaska, “UIR- Middleware”, International Journal of Science and Research (IJSR), Vol. 7(2), pp. 1492-1496, 2018