The realization of a Lightweight RFID Middleware using NoSQL database.

Achraf Haibi[1], Khalid El Yassini[1], Kenza Oufaska[2]
[1]Moulay Ismail University, Faculty of Sciences Meknès, IA Laboratory
achraf.haibi@gmail.com , Khalid.ElYassini@gmail.com
[2]International University of Rabat, Faculty of Computer Science and Logistics, TICLab Laboratory
Kenza.Oufaska@uir.ac.ma

Abstract— *This paper presents various issues related to Radio Frequency Identification (RFID) technology. Given current concerns, we address the general concepts and realization & implementation. In the general concepts, we give different definitions, Radio Frequency Identification components, operating principle, architecture and advantages. The second part of the work presents the realization of a RFID Middleware designed for collecting and filtering large amounts of data from objects carrying the RFID tags.*

Index Terms— RFID, tag, middleware, reader, big data.

IJSER

1 Introduction

Radio Frequency Identification (RFID) is a form of Automatic Identification and Data Capture (AIDC) [2] that uses radio waves to automatically identify people or other objects [1]. This technology has recently seen growing interest from a wide range of industries such as retail, pharmaceutical, and logistics [5]. RFID technology can be used to track objects in a manner similar to using barcode based systems and Optical Character Recognition (OCR) systems [7], but RFID also brings additional benefits. RFID technology does not require line of sight readings, can read multiple tags simultaneously, and store large amounts of data in addition to the ID of the object tracked [3].

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 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 [3]. 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 [8].

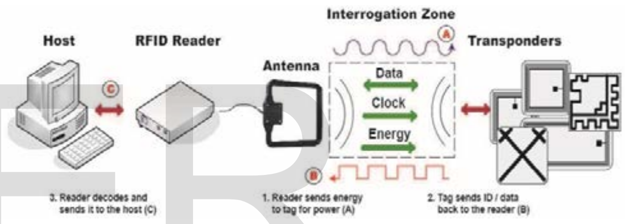

Host
RFID Reader
Antenna
Interrogation Zone
Transponders
A
B
C
Data
Clock
Energy
3. Reader decodes and sends it to the host (C)
1. Reader sends energy to tag for power (A)
2. Tag sends ID / data back to the reader (B)

Fig. 1. RFID System Coponents

2.1 RFID System Components

2.1.1 RFID Transponder/Tag

A RFID transponder, or tag, consists of a chip and an antenna [3]. 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 [2]. Read-only tags are much cheaper to produce and are used in most current applications. 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 [1].

2.1.2 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 that we can find their optimal deployment in the area [8], and they are differentiated by their storage capacity, processing capability, and the frequency they can read [3].

Fig. 2. Example of RFID Reader

2.1.3 RFID Middleware

Middleware, a combination of the English words middle and software [4], is a software interface between the RFID system and the information system of the company. The middleware manages the Automatic Identification and Data Capture (AIDC) technology equipment, receives the traceability events and transfers the formatted data to the information system [4].

Since the middleware is an interface between the readers and the information system, it plays a key role in the process of managing collected data flows from RFID readers.

The main functions of an RFID middleware are:

- It hides the entire hardware part from back-end applications
- It applies filtering: Any redundant, meaningless or useless information will be filtered.
- It is responsible for the raw data processing part before sending it to the relevant applications. It offers the possibility of the management of the readers.

2.2 Middleware Components

As shown in Fig. 3, An RFID middleware generally consists of four layers:

1. Reader Interface
2. Data Processor and Storage
3. Application Interface
4. Middleware Management

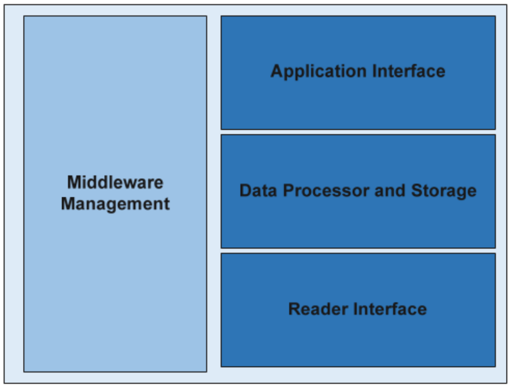


Fig. 3. Middleware architecture [1]

- ➢ Reader Interface

Represents the lowest layer, it is responsible for interactions of the middleware with the various hardware components of the RFID system. This layer maintains the device drivers for all the devices supported by the system.

- ➢ Data Processor and Storage

It is the layer responsible for many services that an RFID middleware is supposed to provide, it process the raw data stream sent by rfid readers, it filters and regrows the information of the rfid tags.

This component also manages the data level events associated with the application.

- ➢ Application Interface

This layer is responsible for interfacing heterogeneous client applications with the middleware by giving them access to the different services provided by the middleware.

It provides the application with an API to communicate and configure the RFID middleware.

- ➢ 4. Middleware Management

Middleware Management: it provides information about all the processes running in the middleware. it allows to :

1. Add, remove, and modify the RFID readers connected to the system.
2. Change various settings by applications.
3. Enable and disable various functions supported by the middleware.

3 RELATRED WORK

3.1 WinRFID

WinRFID is a middleware developed using the Microsoft .NET Framework. WinRFID demonstrates the ease of integration of RFID technology into existing IT infrastructures [42]. It is presented as a scalable and extensible infrastructure (ease of integration with the applications of the company and those of its partners, thanks in particular to the numerous API16 that it has).

3.2 Fosstrak

Fosstrak is an open source platform (formerly known as the "Accada platform") that is designed to meet the needs of track & trace applications. This platform implements the specifications of EPCglobal Inc. (including the EPC Network standard) as three separate modules [9] [10]: a reader module, a middleware and the EPCIS service.

3.3 AspireRFID

AspireRFID is (Advanced Sensors and lightweight Programmable middleware for Innovative Rfid Enterprise applications) an open-source European project launched in the second half of 2008 by the OW221 consortium for the development and promotion of a reliable open source tool whose role is to facilitate deployment (with minimal cost) and management of RFID applications [11] [12].

3.4 SUN Java System RFID

SUN Java System RFID is an RFID platform developed by Sun Microsystems, which supports in its design the big standards

IJSER

accepted by the industrialists (like those developed by EPCglobal Inc.) [13]. It is designed to provide a high level of reliability and scalability of the EPC network by simplifying its integration into existing enterprise systems.

4 BTMIDDLEWARE

4.1 Introduction : why BTMiddleware ?

There are several Middleware each of them has its specificity, some Middleware use standard functions, others are classified among multi-layered middleware, our goal is to design and build a lightweight RFID Middleware called BTMiddleware, this Middleware differs from others in terms of the architecture and the number of data that can collect through the integration of the MongoDB NoSQL database that can underpin a real-time Big Data system.

4.2 BTMiddleware Architecture

BTMiddleware is a middleware which has been realized to meet the damands of the large volume requirements of data transmitted by objects bearing RFID tags.

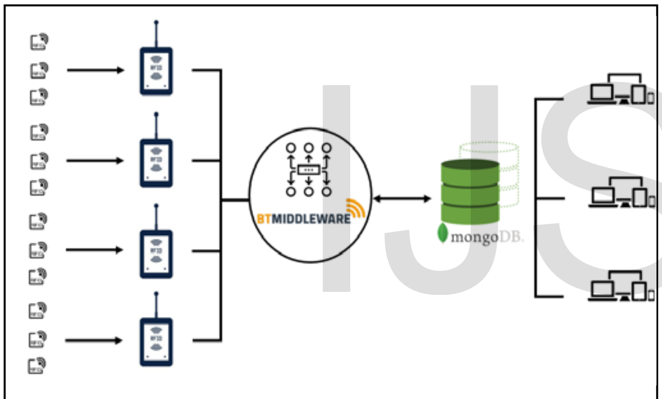


Fig. 4. BTMiddleware Architecture

The middleware is interacting with four external actors.

- With RFID readers: RFID readers will be able to open and close the connection with the middleware.

- With the database: distance applications will have the possibility to consult the data recorded in the database by middleware.

It interact also with the user: this user is the main actor of the system; it comes into play in all the parameters. It has the ability to update the data enable and add the readers.

Fig.5 shows filtering working process of BTMiddleware.

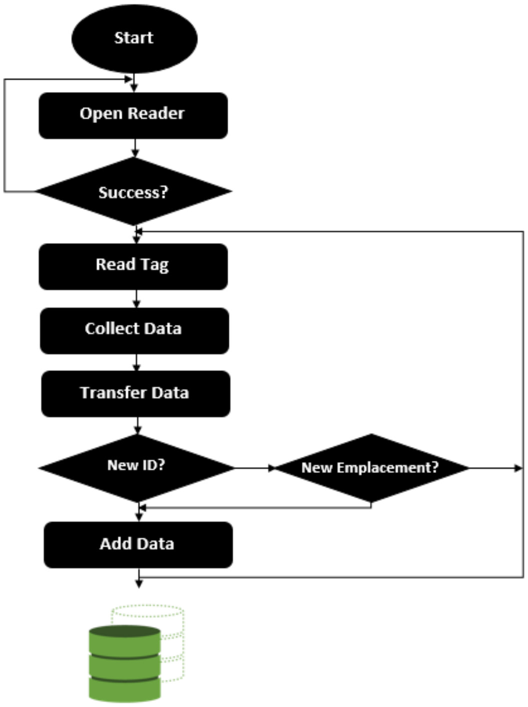


Fig. 5. The working process

4.2 BTMiddleware implementation

For the implementation part, we have used NetBeans IDE as a development environment for the different tools that offers, regarding the programming language, we chose the Java language, and for the database part, we used the MongoDB NoSQL database.

4.2.1 Add a reader

The user is responsible for completing the following fields for each reader

- ReaderID: the reader ID
- ReaderName: represents the name of the reader
- ReaderIPAddr: The IP address of the reader.

When the user finished entering the information, he must click on the "add" button to add the reader.

IJSER

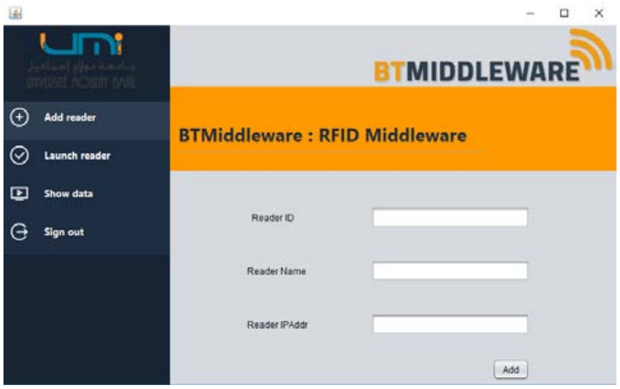
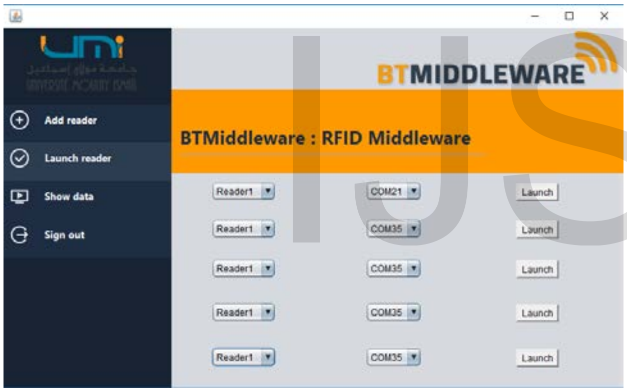


Fig. 6. Add reader

4.2.2 Launch reader

To "launch" a reader, press the "Start" button corresponding to the name of the reader and the number of the port to which the reader is connected. In our example, we will press the "Launch" button associated with "reader1".



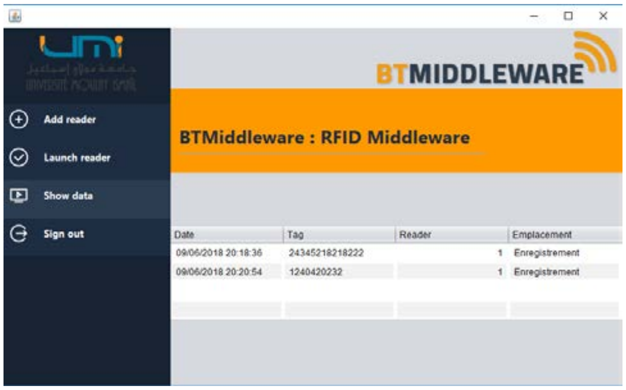
Reader1	COM21	Launch
Reader1	COM35	Launch
Reader1	COM35	Launch
Reader1	COM35	Launch
Reader1	COM35	Launch

Fig. 7. Launch reader

IJSER

4.2.3 Show Data

To view the data collected by readers in real time, an interface offered for this.



Date	Tag	Reader	Emplacement
09/06/2018 20:18:36	24345218218222	1	Enregistrement
09/06/2018 20:20:54	1240420232	1	Enregistrement

Fig. 8. Show data

4 Application Example

Every year so many people travel through airports, of course with their luggage. As shown in the Fig.9, bar code requires a line of sight for the reading of the labels, so a baggage tracking system with bar code technology has a number of limits, taking as an example, a label can be hidden and then the suitcase will be unidentifiable.

The implementation of a luggage traceability system via rfid technology, can reduce the number of lost luggage, and it will result a decrease in the anxiety of travelers

To do this, it is necessary to place on each bag or suitcase a RFID tag, the interest of the BTMiddleware will be the collection of information transmitted by RFID readers in real time through all the phases of the baggage tracking process at an airport, namely: Registration of luggage, Check and scan luggage, Storage of luggage, Sorting the luggage, Withdrawal of luggage, Loading and unloading planes, transfer of baggage between terminals.

5 Conclusion

This document presents RFID technology and its various components. It also describes Middleware implementation and its architecture.

In terms of future work, we intend to develop a complete system for luggage traceability at airports.

References

[1] Yassir Rouchdi , Khalid El Yassini and Kenza 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] T. Ishikawa, Y. Yumoto, M. Kurata, M. Endo, S. Kinoshita, F. Hoshino, S. Yagi and M. Nomachi, Applying Auto-ID to the Japanese Publication Business to Deliver Advanced Supply Chain Management, Innovative Retail Applications, and Convenient and Safe Reader Services, Auto-ID Center, Keio University, (2003).

[3] J. Burnell, What Is RFID Middleware and Where Is It Needed?, In: RFID Update, (2008).

[4] E. Venot, Middleware RFID : traçabilité et objets connectés, Editions T.I, (2015).

[5] 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, (2011), pp. 362-370.

[6] Aqeel-ur-Rehman, Abu Zafar Abbasi, Zubair A. Shaikh, “Building A Smart University using RFID Technology”, International Conference on Computer Science and Software Engineering 2008.

[7] M. E. Ajana, M. Boulmalf, H. Harroud, and H. Hamam "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.

[8] Yassir Rouchdi , Khalid El Yassini and Kenza 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.

[9] C. Floerkemeier, C. Roduner and M. Lampe, "RFID Application Development With the Accada Middleware Platform," IEEE Systems Journal, pp. 1-13, 2007.

[10] C. Floerkemeier, M. Lampe and C. Roduner, "Facilitating RFID Development with the Accada Prototyping Platform," in Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops. PerCom Workshops '07., Zurich, Switzerland, 2007.

[11] J. Soldatos, "AspireRFID Can Lower Deployment Costs," 16 Mars 2009. Available: http://www.rfidjournal.com/article/view/4661.

[12] ASPIRE Project, "ASPIRE - The EU funded project that brings RFID to SMEs," 2009. Available: http://www.fp7- aspire.eu.

[13] A. Gupta and M. Srivastava, "Developing Auto-ID Solutions using Sun Java System RFID Software," October 2004. Available: http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/sjsrfid/RFID.html.

IJSER