A Secure Middleware Architecture for Real-time Tracking Applications

Achraf Haibi[1,2], Kenza Oufaska[2], Khalid El Yassini[1], Mohammed Boulmalf[2]
[1] IA Laboratory, Faculty of Sciences, Moulay Ismail University, Meknes, Morocco
achraf.haibi@uir.ac.ma, khalid.elyassini@gmail.com
[2] TICLab Laboratory, College of Engineering & Architecture, International University of Rabat, Morocco
kenza.oufaska@uir.ac.ma, mohammed.boulmalf@uir.ac.ma

Abstract

RFID for Radio Frequency Identification represents an automated data-capture technology that evolved during the 90's 20th centuries (Qiwei 2007). It is an effective way to identify traceable objects and is widely applied in many domains (such as Healthcare Applications, SCM, Sensing Applications, Transportation, IoT Applications ...). The Internet of Things (IoT) is a global infrastructure for the information society based on a heterogeneous set of subsystems and technologies that need to be interconnected and interfaced in real-time with each other by following suitable standards for communication, integration, measurement, control and interoperability (Rouchdi et al. 2018). Of the interoperable solutions, the middleware is worth mentioning. In the IoT, RFID middleware is the software chain's brain that gives intelligence to the collected data from RFID-tagged objects, it represents a technical software layer that acts as an interface between the various IT applications and the RFID equipment network whose role is to manage the hardware network, deduplicating redundant information, aggregation, dissemination, and formatting of RFID data (Ajana et al. 2009). This work presents an RBAC and NoSQL based RFID software framework whose goal is to increase security by making the authentication process much more efficient. This new middleware architecture enables the monitoring, processing and storing of a large volume of incoming RFID events flow in real-time by combining RFID technology with MongoDB and provides enhanced interoperability through the option of presenting RFID data in JSON format.

Keywords
RFID, Middleware, IoT and Security.

1. Introduction
The acronym RFID represents a type of Automatic Identification and Data Capture (AIDC) (Ishikawa et al. 2003); in other words, it stands for all technologies that use radio waves for identification purposes. This technology utilizes radio waves to retrieve the data stored in equipment named RFID tags to automatically identify and detect target (items, people, or other objects) (Haibi et al. 2018). This technology has contributed to developing a wide range of domains, such as complex environments, access control, sensing applications, and other fields. RFID technology, compared with the barcode-based systems, brings additional benefits; it has the advantages of higher storage capacity, reading various labels simultaneously, and quick identification. The RFID system is made up of four components: RFID tags, RFID reader, business applications plus a software layer named RFID middleware.

For large-scale or complex RFID solutions, RFID middleware has a crucial role in interfacing business management modules with data from RFID tags. This work focuses on the software parts by proposing a new architecture of data acquisition and processing platform. The middleware will ensure the interface between the collected data (contained in the RFID tags memory) and the company management software exploiting these data; it ensures the extraction of RFID data from the readers, it also allows filtering the data, aggregating them and transmitting them after distillation to the enterprise information systems. From the scientific literature, we note that, in general, a limited number of research works have proposed traceability solutions designed for huge volumes of RFID events processing, acquisition and storage in real-time. Rouchdi et al. (2018) present a proposal for a three-tier RFID middleware, which is advantageous in terms of security but does not allow for the analysis & processing or storage of large RFID event streams. In (Sahni et al. 2018), authors suggested a Middleware architecture for structural health monitoring application; this architecture has multiple advantages, but it cannot be integrated into RFID-based tracking solutions because (like the previous architecture) it does not support a large data flow nor mobile RFID objects. Bouhouche et al. (2017) has proposed a middleware architecture designed for baggage tracking in airports. This middleware supports heterogeneous RFID equipment, but like other architectures, it does not address the problem of big data and does not take into consideration the security of RFID data.

This paper is organized as follows; section gives an overview of the RFID System. Section 3 details our proposed Middleware architecture components. Section 4 showcases two application domains. Section 5 concludes the paper.

2. RFID system components

RFID systems are comprised of four main components: a tag (also called a transponder) attached to the target to be identified, a reader (also called an interrogator) that is tasked with data extracting from tags, RFID middleware installed on a host computer, and business applications. The latter represents the software components that exploit and interpret the RFID data; each application has its own requirements and requests the desired information set. As Figure 1 shows, the signals sent from the reader‘s antennas constitute an interrogation space. When a label is in this area, it will be activated to communicate its data with the interrogator (Burnell 2008). Afterward, the collected data is processed by the middleware; this component monitors interrogators (readers), as well as aggregates, formats, filters, and stores the collected RFID data for subsequent use by concerned applications (Haibi et al. 2019).

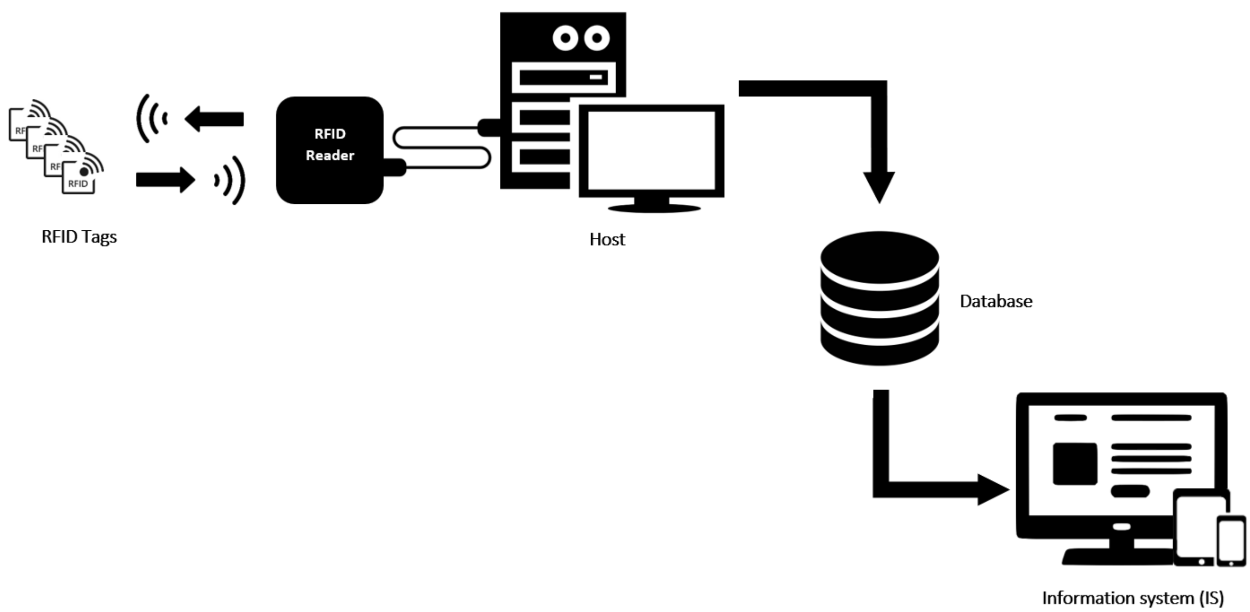

Figure 1. RFID System Components

2.1 RFID Tags

Also called transponders. According to powering, tags are available on the market in 3 versions: active tags, passive tags, and semi-passive tags. This component consists of two main elements a chip and an antenna (Ajana et al. 2009). In addition to a unique serial number, the chip may be capable of storing additional information depending on the tag's memory type, which can be Read Only tags (identifiers and data are saved on the chip by the manufacturer), Read/Write tags (are editable in terms of content and can be read several times on different readers), or (WORM) Write-Once and Read-Many (the content cannot be modified. but it is possible to read them several times on different RFID readers) (Venot 2015). Read-only tags are inexpensive to produce than other types of tags and are used in most current applications. The antenna allows the exchange of information from the chip to a reader.

2.2 RFID Reader

The reader, or interrogator, is a scanning device used to communicate with the RFID tags by transmitting a signal at a specific frequency to tags located in its reading field. These send back a signal. When the tags are "awakened" by the reader, a dialogue is established according to a predefined communication protocol, and the data is exchanged. As shown in Figure 2, the reader can take the form of a hand-held portable device or a fixed device. They differ in their storage capacity, processing capacity, and the frequency they can read (Burnell 2008).

Figure 2. Type of RFID readers (Linda and Samuel Fosso 2007).

2.3 RFID Middleware

In general, middleware is a set of software layers providing the interface between several applications, whether hardware or software. For an RFID project, the middleware will ensure the interfacing between the RFID tags data (contained in the memory of the RFID tags as already mentioned) and the information system (or business application as named previously) of the company that is responsible for data exploitation. 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 (Rouchdi et al. 2018). The RFID middleware 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 filters RFID Data: 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.

Figure 3 shows the middleware position in an RFID system.

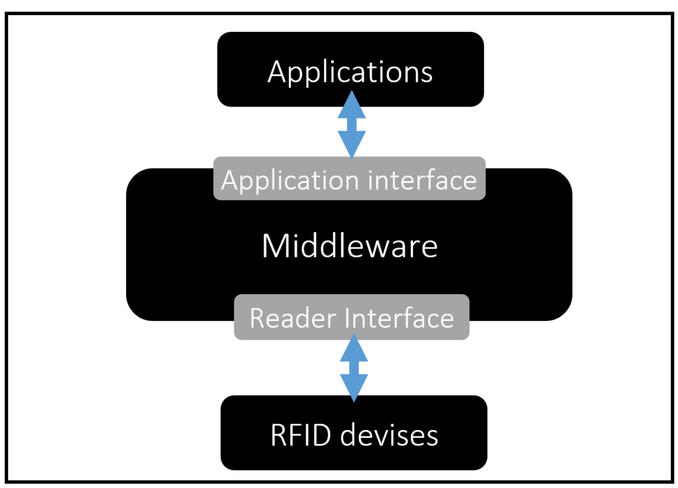

Figure 3. RFID Middleware position

3. The proposed Middleware architecture

Despite the hardware components and sensors evolution, the data acquisition, processing and storage solutions dedicated to this kind of equipment have not yet achieved significant contributions (Baruffa et al. 2020). In fact, despite the maximum optimization of relational DBMS, the limits are reached (Baruffa et al. 2020), especially in terms of performance regarding data reading and writing speed as well as the events' volume that can be generated. NoSQL databases are based on innovative systems allowing horizontal scalability and have a high level of flexibility thanks to a non-fixed schema data model (Baruffa et al. 2020), which makes them qualified to meet specific needs and ensure extreme scalability over very large datasets.

The large-scale implementation of RFID-based solutions creates a huge volume of data, taking as an example the case of a medium-sized retail chain adopting RFID technology, it can reach up to 300 million RFID reads a day (Dutta et al. 2007). And for a system operating with traditional IT solutions, it will be difficult to retrieve useful information from this huge volume of RFID events, which is where the importance of the middleware layer comes in.

An important set of advantages characterizes RFID, but like any technology, it presents multiple challenges, especially in the RFID data acquisition and processing part. (1) RFID networks generate a large stream of events continuously and redundantly. (2) The generated raw flow is not understandable by the IS, which requires to be formatted in semantic data. An RFID middleware layer that supports big data, as well as RFID device management, must be integrated into the system in order to process and transmit RFID data efficiently.

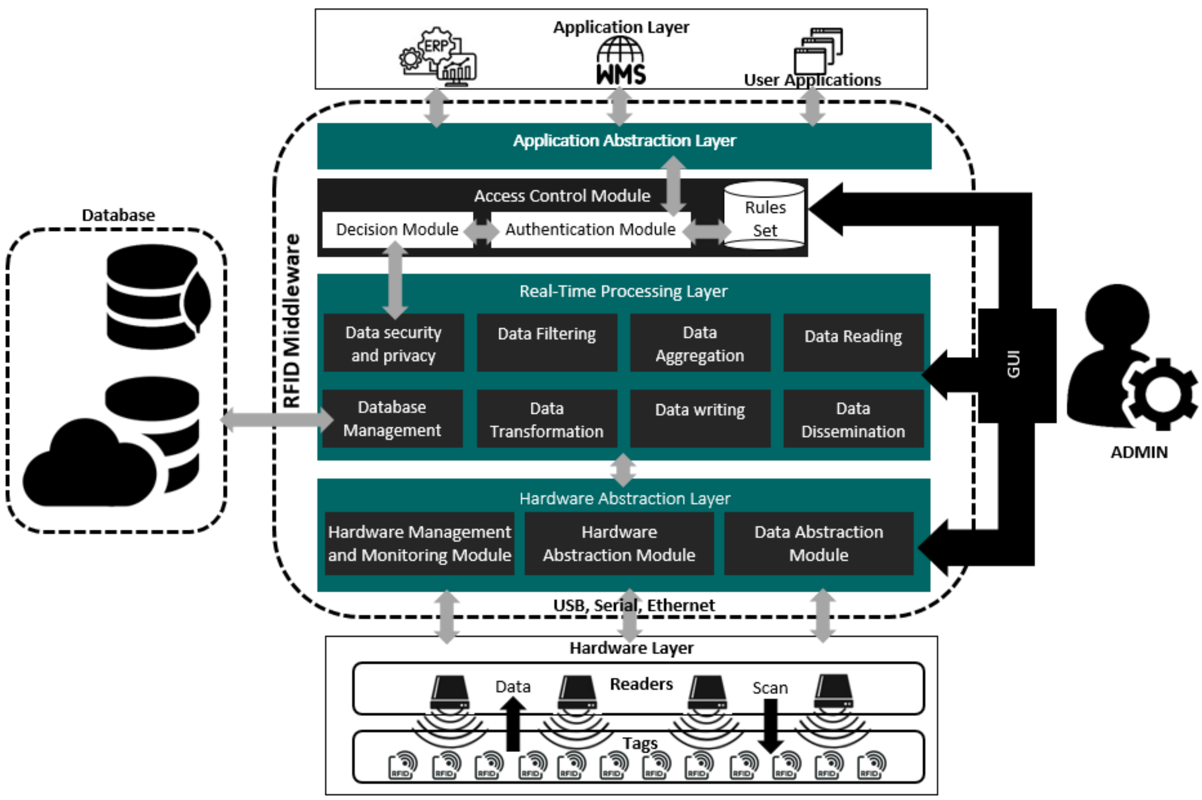

Figure 4. Middleware Architecture Components

The proposed architecture is illustrated in Figure 4. This architecture offers various advantages by taking into account several aspects. (1) Big data storage thanks to the NoSQL database named MongoDB, (2) processing and analysis of RFID data streams in real-time by applying the CEP approach, (3) Offering high interoperability by providing data in JSON format, (4) increasing the RFID data security by controlling the access with RBAC model. Therefore, regarding the problems presented above, the proposed architecture presents a solution for all these issues. The four layers constituting our architecture are:

A. Application layer
This layer provides functions and services to companies and IS users. The applications can take different forms: ERP, WMS, User applications (Mobile application, website, and desktop application).

B. Hardware Abstraction Layer

In order to ensure communication between RFID readers and software, EPCglobal announced the ratification of the Low-Level Reader Protocol standard in 2007. In fact, we find many readers of different brands, with some that follow this standard and others who do not consider it. The abstraction of RFID devices offers a common communication interface with heterogeneous RFID readers regardless of the manufacturer, the communication interface (Ethernet, USB, RS232) or even the frequency range (UHF, HF or LF). This layer is composed of 3 sub-layers:

- **Hardware abstraction module**

The communication between the RFID devices and the middleware is ensured via a common interface for all devices through this module. This sub-layer offers the system administrator the right to configure the RFID readers' settings and provides the functions of Start/Stop, Activate/Deactivate and Read/Write.

- **Data abstraction Module**

RFID data is shared with the upper layer (RTPL) via DAM in a common format regardless of the source. This module handles middleware queries such as RFID data, inventory results, ACK...

- **Hardware Management and Monitoring Module**

Thanks to this sub-layer, the user will choose between a wide range of reader brands according to his preferences and allow him to change the hardware without impacting the software solution. It makes the architecture more flexible since it calls the RFID reader driver libraries dynamically.

C. Real Time Processing Layer

The RTPL layer represents the core of the architecture; it contains all the functionality and services of the Middleware. Its operating principle is as follows; the RTPL will not process requests from an application if it does not hold the necessary authorizations from the ACM. Regarding its relationship with the lower layers, the RTPL receives the raw RFID event stream, and its modules will be in charge of filtering, formatting, storing and sharing the data with the concerned applications. The RTPL is based on the CEP approach in order to make the Middleware capable of tracking, analysing and processing data as and when an RFID event occurs. CEP, which is defined as any computation/processing that performs operations on complex events (Oufaska et al. 2018), is an advanced event processing based approach that collects and combines data from different relevant sources to discover events and patterns that can drive actions. CEP is considered a critical component for RFID solutions to ensure real-time processing and analysis of a large volume of RFID data. The components of RTLP are as follows:

- **Data Dissemination**

The collected RFID data is essential for the backend applications; this module is responsible for sharing the formatted RFID data with all interested applications.

- **Data Aggregation**

As mentioned before, the company's IS is composed of a set of applications, each of which is interested in a part of the RFID data. DA is responsible for grouping data according to typical rules, such as grouping data about an object or grouping data from the same RFID reader...

- **Data Security and Privacy**

The Data Security and Privacy module communicates with the ACM in order to check the list of granted authorizations each time an application requests information from the Middleware.

- **Database Management**

The high volume of data characterizes the new application sectors of RFID technology, and in these sectors, it is necessary not only to manage very large amounts of data but data that can also have a very large size, knowing that relational DBMS show their limit with very high data flows (Jose and Abraham 2017). To optimize the Big Data processing time, the best solution is NoSQL systems (Chauhan 2019). This sub-layer manages the MongoDB database, providing functions such as storing, deleting and sharing data; in other words, it plays the role of database manager.

- **Data Reading**

Data reading is one of the basic functions that an RFID middleware should perform; that is why it is found on any architecture. So, like all Middlewares, this module is responsible for reading the flow of RFID elements.

- **Data Transformation**

The raw RFID stream is not understandable by the backend applications until the formatting phase, and that is the role of the Data Transformation Module, which is in charge of receiving the raw RFID data and transforming it into business events so that applications can exploit it.

According to the scientific literature, all RFID middleware transforms data into XML format, and this is quite normal since, for many years, XML was the only choice for data sharing and transfer. Despite the advantages offered by this format, it is not suitable for large datasets, is more difficult to read for humans, is much more verbose and lacks similarity to the data model used in modern programming languages (Breje et al. 2018), which makes it less and less prevalent in new systems. JSON and XML are the two most commonly used formats for data sharing. However, JSON format performed well among languages in contrast to XML (Vanura and Kriz 2018) and offers several advantages over XML, e.g. it is easy to parse by machines and independent of the languages that use it, processing of data in JSON format is simpler than XML (Due to a complex structure, XML takes more time to process than JSON). Today JSON is generally preferable to XML and is quickly replaces it (Lanthaler and Gütl 2012), which motivated us to add the possibility of formatting also in JSON format. The added value of our architecture is that it allows the user to choose between the two formats, XML or JSON, according to the specifications of the company's IS.

- **Data Filtering**

As mentioned before, the RFID data flow is voluminous, and it is necessary to filter it in order to keep only useful information; it is the role of this sub-layer to eliminate all kinds of undesirable data and in particular, to eliminate duplicates.

- **Data Writing**

As mentioned in section 2.1, on the market, there are Read / Write type tags; that is why the middleware integrates this sub-layer to ensure the function of writing on the tag's chip.

D. Application Abstraction Layer

It provides a common interface for applications with the hardware to hide the complexity of the system. Through this interface, the applications access the services and functionalities offered by the middleware if they have the necessary authorizations.

E. Physical layer

Represents the infrastructure of the deployed RFID equipment (readers, tags...).

F. Access Control Module

In order to ensure security within companies and organizations, this layer makes it possible to assign access authorizations to middleware services only to users who actually need them for their activity. Sensitive data is thus protected from unauthorized access and unwanted changes. The principle of this layer is based on role-based access control (RBAC). The role-based access control implementation process starts with assigning a role to each user; each role has a different level of access to the middleware services, and this layer will automatically restrict their movement through the enterprise network. Integrating RBAC into the system optimizes operational efficiency, protects data from leakage or theft, reduces administration and IT support (Rouchdi et al. 2019).

Its operating principle lies first on authentication to certify the user's identity, and subsequently, according to the permissions assigned to that user, the AC module shares the decision. If the user does not have the necessary authorizations, his requests will be rejected; if the user is authorized, this module will determine the resources to which he will have access and the operations he can perform.

G. Admin GUI

It is the user interface that allows communication with the Middleware; after authentication, the Admin through it can configure the RFID network (Power, Port number, IP addresses, Communication protocols, filtering rules ...), database configuration, assign access rights to roles and then assign roles to users, choose data formatting type, and real-time visualization of RFID data.

Our proposed data acquisition and processing platform offers a solution to many issues addressed previously, such as the ability to store huge RFID data. It also supports heterogeneous hardware on the HAL that gives a common interface for a set of RFID devices with different properties. In addition to that, it integrates the CEP algorithm for processing data streams to enable real-time data analysis. In addition, on the RTPL layer, the DM module provides NoSQL database integration solving the big data storage problem while increasing security and interoperability by integrating RBAC and data formatting by JSON format, respectively.

4. Applications

RFID is found in many sectors, but thanks to the hardware evolution, the application field of this technology has expanded towards critical areas in recent years, such as the medical and aeronautical sectors where the notion of real time is vital. The evolution of RFID tags capable of storing more and more data has enabled the growth of Big Data. And this requires IT tools capable of managing, processing and storing this type of data, taking into account all of its constraints. Such domains present a set of challenges, particularly in terms of data collection and processing: (1) RFID events stream has no meaning for the IS, which imposes that they must be transformed and aggregated into semantic data that can be understood by the IS (Haibi et al. 2018), (2) The RFID data flow is often continuous, in high volume and redundant. To have good conditions for RFID data intended for processing and transmission, the middleware layer, which manages both readers and a large number of RFID events in real time, must be integrated into the system (Rouchdi et al. 2018).

Regardless of the application, the deployment principle remains more or less the same; the object to be identified is labelled with an RFID tag, the readers communicate the position and other information with the Middleware layer in real time. The latter collects, filters and transforms the raw RFID data into a format understandable by applications, and transmits only the necessary information to the concerned applications.

4.1. Industrial sector

The industrial sector is constantly modernizing in recent years, especially with the emergence of the connected factory and "Smart Manufacturing". To produce more efficiently and reduce errors, the traceability of components and products is a major concern for manufacturers. With RFID technology, the accuracy of traceability is significantly improved throughout the production chain. Industries such as energy and automotive require an even greater degree of precision and increasingly rely on RFID solutions.

To cover the entire chain with RFID technology, firstly, readers with different shapes are needed: Portal Readers, Fixed Readers, and Mobile Readers, which requires a software system implementation capable of processing a heterogeneous equipment network in real-time. Secondly, a UHF RFID Tags capable of communicating its information with these readers even if they are placed in metal. Thanks to our middleware architecture, the RFID system deployment will be compatible with smart industry applications. It allows:

- Interfacing with heterogeneous RFID readers.
- Multitude of objects data collection, filtering, and Distribution.
- Provision of transparent data access for applications.

4.2. Airport

Airlines today face many challenges that impact customer retention, and therefore profitability. Among these challenges is the ever-increasing number of travellers (SITA reported an increase of 1.2 billion between 2007 and 2016, reaching 3.7 billion people) (Haibi et al. 2018), which causes a significant increase in the number of baggage. The problem of lost baggage costs airlines billions of dollars each year (Haibi et al. 2019), and the cost of losing baggage, the time required to identify and locate it are today considered unacceptable by travellers and give rise to major frustrations in terms of customer service. Greater efficiency in airline services requires, in particular, the adoption of modern technological solutions for their entire service offering. RFID technology enables this improvement in one of the most critical areas of their operations: baggage handling, to minimize the amount of lost baggage, with the goal of increasing customer satisfaction.

The development of IoT systems using RFID technology (in which tagged bags are the "objects" of the network) provides real-time visibility into luggage location, with valuable verification steps performed at different stages of the journey without any human intervention. This requires the development of a traceability system characterized by security and robustness in terms of analysing the large data streams that can be collected in real-time and the ability to receive and transfer formatted traceability data in the most common formats.

Streamlining baggage handling with RFID technology by adopting a system that operates according to the middleware architecture specifications discussed above will bring several baggage benefits to Moroccan airlines and airports:

- Automation of procedures
- Reduction in the number of misplaced luggage, with associated costs.

- RFID data security.
- Better interoperability thanks to the two formats, XML and JSON.
- Ability to process a large amount of RFID events.
- Ability to store a large amount of data.
- Multiple readers Integration, allowing real-time baggage tracking.
- Reduce labour costs.
- Reduction in the number of lost bags.
- Providing a global view of baggage location.

5. Conclusion

In this document, we propose an RFID Middleware architecture. The proposed middleware differentiates from existing middleware architectures in that it can store any type of RFID data, integrates data stream processing to allow real-time data analysis, supports massive data, and stores data in JSON format for better interoperability (Breje et al. 2018). It also describes two application areas for RFID technology. Such a platform allows automated and real-time tracking, reduced management time, real time location, reduction of the workforce.

Our middleware architecture is in its first stage of development. At this phase, we have outlined the features and services that will be used in our future Middleware. It was found in the initial assessment study that our architecture can solve problems such as security, storage, large amounts of data processing, and heterogeneity. For future work, we plan to complete the implementation of our Middleware. This will allow us to evaluate it using a real deployment case to quantify the entire proposed system's performance.

References

Qiwei, L., Research and Design on Radio Frequency Identification Reader, *International Workshop on Anti-Counterfeiting, Security and Identification (ASID)*, pp. 356-359, 2007.

Rouchdi, Y., Haibi, A., El Yassini, K., Boulmalf, M. and Oufaska, K., RFID Application to Airport Luggage Tracking as a Green Logistics Approach, *IEEE 5th International Congress on Information Science and Technology (CiSt)*, 2018.

Ajana, M. E., Boulmalf, M., Harroud, H., and Hamam, H., 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, Marrakesh, Morocco*, 2009.

Ishikawa, T., Yumoto, Y., Kurata, M., Endo, M., Kinoshita, S., Hoshino, F., Yagi, S. and Nomachi, M., 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.

Linda, C. and Samuel Fosso, W., An Inside Look at RFID Technology, *Journal of Technology Management & Innovation,* vol. 2, no. 1, 2007.

Haibi, A., El Yassini, K. and Oufaska, K., Suitcase traceability system via RFID and NoSQL database, *ACM Proceedings of the 3rd International Conference on Smart City Applications*, 2018.

Rouchdi, Y., El Yassini, K. and Oufaska, K., UIR-Middleware, *International Journal of Science and Research (IJSR)*, vol. 7, no. 2, pp. 1492-1496, 2018.

Sahni, Y., Cao, J., and Liu X., MidSHM: A Middleware for WSN-based SHM Application using Service-Oriented Architecture", *Future Generation Computer Systems*, pp. 263-274, 2018.

Bouhouche, T., Raghib, A., Abou El Majd, B., Bouya, M., and Boulmalf, M., A Middleware Architecture for RFID-enabled traceability of air baggage, *MATEC Web of Conferences*, 2017.

Burnell, J., What Is RFID Middleware and Where Is It Needed?, In: *RFID Update*, 2008.

Haibi, A., Oufaska, K., and El Yassini, K., Tracking Luggage System in Aerial Transport via RFID Technology. *In: Ben Ahmed M., Boudhir A., Younes A. (eds), Innovations in Smart Cities Applications Edition 2. SCA 2018*. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham, 2019.

Ajana, M. E., Harroud, H., Boulmalf, M., and Hamam, H., FlexRFID: A flexible middleware for RFID applications development, *IFIP International Conference on Wireless and Optical Communications Networks, Cairo*, pp. 1-5, 2009.

Venot E., Middleware RFID : traçabilité et objets connectés, *Editions T.I*, 2015.

Rouchdi, Y., El Yassini, K. and Oufaska, K., Complex Event Processing and Role-Based Access Control Implementation in ESN Middleware, *Innovations in Smart Cities and Applications*, pp. 966–975, 2018.

Baruffa, G., Femminella, M., Pergolesi, M., and Reali, G., Comparison of MongoDB and Cassandra Databases for Spectrum Monitoring As-a-Service, *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 346-360, 2020.

Baruffa, G., Femminella, M., Pergolesi, M., and Reali, G., "Comparison of MongoDB and Cassandra Databases for supporting Open-Source Platforms tailored to Spectrum Monitoring as-a-Service", IEEE Transactions on Network and Service Management, 2019.

Dutta K., Ramamritham K., Karthik B., Laddhad K., Real-Time Event Handling in an RFID Middleware System. *In: Bhalla S. (eds) Databases in Networked Information Systems*. DNIS. Lecture Notes in Computer Science, vol 4777. Springer, Berlin, Heidelberg, 2007.

Qadir, S., and Siddiqi, M., Performance evaluation of a secure Low Level Reader Protocol(LLRP) connection, 2009.

Oufaska, K., Rouchdi, Y., Boulmalf, M., and El Yassini, K., RFID application to airport luggage tracking as a green logistics technology, *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2018.

Jose, B. and Abraham, S., Exploring the merits of nosql: A study based on mongodb, *International Conference on Networks & Advances in Computational Technologies (NetACT)*, 2017.

Chauhan, A., A Review on Various Aspects of MongoDbDatabases. *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 05, 2019.

Breje, A. R., Gyorödi, R., Gyorödi, C., Zmaranda, D. and Pecherle, G., Comparative Study of Data Sending Methods for XML and JSON Models, *International Journal of Advanced Computer Science and Applications*, 2018.

Vanura, J., and Kriz, P., Perfomance Evaluation of Java, JavaScript and PHP Serialization Libraries for XML, JSON and Binary Formats. *Lecture Notes in Computer Science*, 2018.

Lanthaler, M. and Gütl, C., On using JSON-LD to create evolvable RESTful services, *Proceedings of the Third International Workshop on RESTful Design - WS-REST*, 2012.

Rouchdi, Y., Oufaska, K., Haibi, A., El Yassini, K. and Boulmalf, M., Role-based access control in BagTrac application, *International Journal of Knowledge Engineering and Soft Data Paradigms*, 2019.

Biographies

Achraf HAIBI received his Bachelor's degree in Computer Networks and Telecommunications from Polydisciplinary Faculty of Errachidia - Moulay Ismail University, Morocco in 2014 and subsequently he obtained his Master's degree in Computer Networks and Embedded Systems from Faculty of Sciences of Meknes in 2017, Moulay Ismail University, Morocco. He is currently a PhD student in Computer Science at Moulay Ismail University (UMI) in collaboration with the International University of Rabat (UIR). His main research interests are Radio Frequency Identification, Real-time systems, Big Data technologies and Internet of Things.

Kenza OUFASKA is an assistant professor at the International University of Rabat - Morocco and attached to « Ecole Supérieure d'Informatique et du Numérique » and member of TICLab. She obtained a PhD in Mathematics (Operations Research and Logistics option) at Moulay Ismail Meknes University. Her research work focuses on multi-objective mathematical programming applied to the fields of intelligent transport, intelligent mobility, optimization of supply chains, green and sustainable logistics and humanitarian logistics.

Khalid EL YASSINI holds a PhD and MSc in Mathematics (Operations Research) from University of Sherbrooke - Canada (2000 and 1994). Previously, he had a degree in Applied Mathematics (Statistics) at Abdelmalek Essaadi University - Morocco (1991). During his academic career, he has taught in several Canadian institutions such as University of Sherbrooke, Univesité du Québec à Rimouski or St-Boniface University at Winnipeg; and in various Moroccan institutions such as Moulay Ismail University, International University of Rabat, ENSAK, ESTM, ENSAM or Royal Military Academy. He has taught various courses in mathematics, computer science, operations research and logistics. His current research activities focus on mathematical programming, artificial intelligence, information systems, security in computer networks, telecommunications, intelligent systems, and logistics engineering with applications in hospital field or medication domain. Currently, he is a Professor at the Faculty of Sciences - Moulay Ismail University, Morocco. Its main current research activities focus on mathematical programming (linear and multiobjective optimization), artificial intelligence, information systems, security in networks, telecommunications, intelligent systems (smart cities and smart vehicles), logistics engineering, and green logistics with applications in hospital field or medication domain.

Mohammed BOULMALF is a Professor and Dean at School of Computer Science and Digital Engineering in the International University of Rabat, Morocco. He served on technical program committee for several international and national conferences. In addition, he has been a reviewer for several international journals and conferences. He is author/co-author many articles in fields of wireless networking and communications, wireless sensor networks, mobile computing, RFID technologies and network security.