International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

UIR- Middleware

Yassir Rouchdi[1], Khalid El Yassini[2], Kenza Oufaska[3]

[1, 2]IA Laboratory, Faculty of Sciences Meknes, Moulay Ismail University, Meknes, Morocco

[3]TIC Laboratory, International University of Rabat, Rabat, Morocco

Abstract: ***This paper presents Radio frequency identification components, functioning and Middleware's role. It discusses ESN middleware architecture and explains its security and privacy issues, including a discussion about resolving these problems by applying Role based access Control model as an authentication tool regulating back-end application's access to data. Moreover, it presents the proposed architecture of our three layers middleware 'UIR-', Explaining how Complex event processing can handle RIFD and WSN data, shows RBAC rules application and gives details on the implementation process.***

Keywords: RFID; WSN; Middleware; CEP; RBAC

1. Introduction

The internet of things holds the promise to offer advanced connectivity of devices, networks, and services that goes beyond machine-to-machine communications and to cover a wide range of protocols, domains, and applications. The interconnection of these embedded devices is expected to marshal in automation in nearly all fields, while also empowering advanced applications and elaborating to areas such as smart cities. In order to do so, IoT assembles both wireless and wired technologies into the same network, using Low-power wide-area networking (LPWAN) for long-range wireless connections, HALOW and LTE-advanced for medium-range, and Radio Frequency identification between many others (Bluetooth Low Energy, NFC, WIFI …) for short-range communication. One of the focuses of scientists nowadays is resolving the issues occurring during the use of combined technologies resulting in unexpected complex events.

WSN and RFID are both very efficient and reliable technologies but aren't any exception of the rule, their combination means dealing with RFID imperfect privacy and security and every WSN complex event issue, during this paper we will be trying to resolve some of the problems , therefore making both RFID and WSN more secure and stable. This paper presents Radio frequency identification components, functioning and Middleware's role. It discusses ESN middleware architecture and explains its security and privacy issues, including a discussion about resolving these problems by applying Role based access Control model as an authentication tool regulating back-end application's access to data. Moreover, it presents the proposed architecture of our three layers middleware 'UIR-', Explaining how Complex event processing can handle RIFD and WSN data, shows RBAC rules application and gives details on the implementation process.

RFID Components, Functioning and Middleware
RFID stands for Radio Frequency Identification. Its importance and efficiency are expressed by the vast amount of medical, military and commercial applications using this approach Worldwide [1]. Billions of the RFID systems are operated in transportation (automotive vehicle identification, automatic toll system, electronic license plate, electronic manifest, vehicle routing, vehicle performance monitoring), banking (electronic checkbook, electronic credit card), security (personnel identification, automatic gates, surveillance) and medical (identification, patient history) [2].

1.1. RFID 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 [1].

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, reader collision prevention and tag-reader authentication techniques [2, 3, 4]. Fig 1 illustrates RFID components.

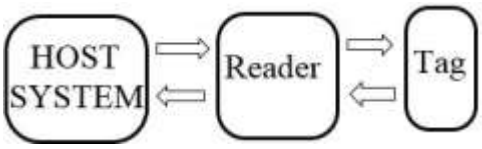

Figure 1: RFID Basic Components

1.2 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 [6].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.

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) [8, 9].

1.3 Middleware's basic functions

The three primary functions of 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)
- 'Feeding applications', with relevant information based on the information collected from devices after properly performing the appropriate conversions and formatting [7].

1.4 Middleware Architecture

The usual architecture of an RFID middleware is presented in figure 2:

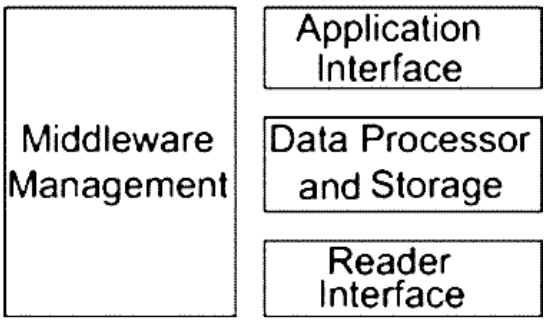

Figure 2: Basic RFID Middleware Architecture

Data Processor and Storage: The data processor is responsible for the management and processing of raw data from the readers. This component is also responsible for storing the raw tag data, so that it can be processed. Part of the important processing logic performed here is the Filtering and Grouping of RFID Data. This component also manages the level event data associated with the application. By way of example, when all applications request data captures in the same time interval, processing of the time stamp is performed by this component and the data is then transmitted to the applications.

Application Interface: The application interface component manages the interface of the middleware in dependence on those of the applications. It provides the application with an API (application programming interface) to communicate and evoke the RFID middleware. It accepts application requests and translates them down to the underlying components of the middleware. This component is responsible for integrating enterprise applications with RFID middleware.

Middleware Management: The middleware management component helps with the management of the RFID middleware conspiracy. It provides information about all the processes running in the middleware. The middleware management provides the administrator with the following features:

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

2. Related Work

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, CEP technology is used in several RFID middleware systems. Event processing language was used to define complex events. In this paper we will be applying complex event processing to combine unions and intersections of both RFID and WSN simple events, defined as complex events.

Concerning the WSN part, several collectors of sensor data and sharing architecture already distinguished [6-8]. Global Network of sensors (GSN) middleware which is the database Capture of virtual sensors and powerful query tools makes access to the heterogeneous wireless sensor knots easier. Hi-fi architecture [9] is a hierarchical architecture for processing distributed RFID data and network of sensors. It includes many components, such as data receiver, data stream processor, data sender, resource manager, query listener, etc. This article propose a new approach. Instead of building our architecture from scratch, UIR middleware design is built according to already developed RFID standards. it leads to a framework suitable for both RFID and WSN integration applications.

3. UIR Middleware

3.1. UIR Architecture

We propose to develop an RFID middleware called UIR, bearing in mind the design problems discussed in the second chapter. Our system is organized as a three-tiered

architecture, with back-end applications, middleware (UIR-) and both RFID and WSN 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). The next figure (Fig. 3) illustrates UIR-RFID architecture.

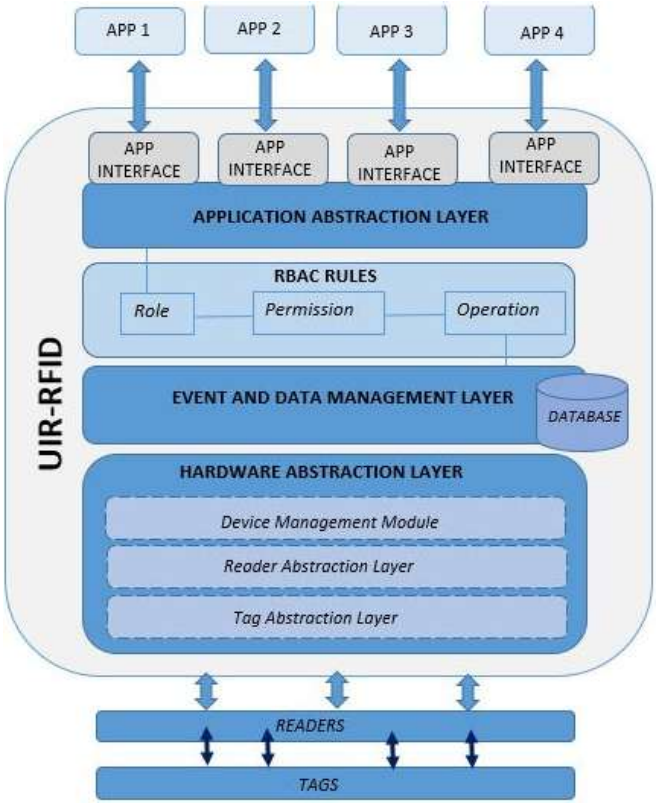

Figure 3: UIR- Middleware Architecture

3.2. 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).

3.3. 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. Later, simple events are grouped together to form complex events. Complex events provide more meaningful reporting and improve automation of the system [15].

3.4. Simple and Complex Events in ESN middleware

In the ESN middleware, two types of event objects exists, RFID event object and sensor event object. the RFID event object is defined as a tuple (ID, L, T) where ID represents the EPC code of the RFID tag, L for Location and T for time. The sensor event object, which is more complex than RFID, is defined hierarchically. The first layer is still a tuple (ID, L, T, D), where ID is the identification of a Sensor node, L Location, T Time and D is the sensor data (Temperature, weight …). The ID includes both the reader ID and the ID of the sensor node. To achieve unique IDs identification, the reader's EPC codes, and the sensor node's can be used as identifiers. Whereas in the second layer of event object in D, a sensing type tuple such as (humidity, temperature, pressure) can be included [16].

A simple event is the RFID event or sensor with constraint.

For example, the RFID event (S1) in an application level (localization constraint) is a simple RFID event. S1 = (ID, L {L = "Test Location"}, T) And the sensor event (S2) per example, with a temperature higher than 20 degree is also a simple sensor event. S2 = (ID, L, T, D {Value of D. temperature> 20}). A complex event is a combination of simple events or complex events with the following set:
AND (∧): E1∧ E2 represents two events, where E1 and E2 occur together.

OR (∨): E1∨ E2 means E1 or E2 occurs.
NOT (!) :! E1 means that E1 does not occur.
SEQ (→): E1 → E2 means that E1 is followed by E2.
Relative peripheral (Rp): Rp (E1, E2, E3) means that E2 occurs between E1 and E3, perhaps several times.

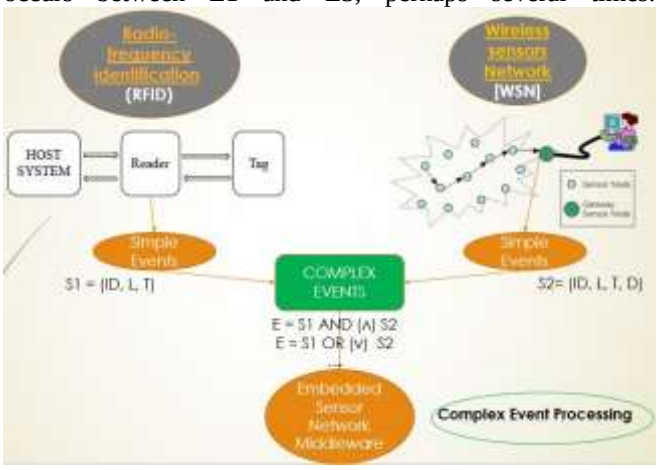

Figure 4: Illustration of CEP

3.5. 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 [19].

3.6. Role-Based access control

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.

3.7. 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))).

4. Application Example

A smart medicament transportation application, can illustrate the use of our ESN middleware. Medicaments put into small boxes are tagged and localized by an Alien reader installed in the transportation vehicle. Meanwhile, a sensor node equipped with a microcontroller, a transceiver, a temperature sensor and a humidity sensor senses the temperature and humidity of the Medicaments environment. Sensor nodes in different Vehicles compose a WSN and transfer data to reader using. As an example, one of the complex events in the above application is generated when there is Medicaments on the truck and temperature is above 13 degrees. For this complex event, the corresponding event rules are as follows: S1 = (ID{ID= 'MedsID}, L{L="reader A"}, T) S2 = (ID, L{L="Track A"}, T, D{D.temperature.value>13}) C1 = S1 ∧ S2 When this complex event is generated, EPCIS gets a report, and action would be token.

5. Conclusion

Our proposed middleware (UIR-) architecture offered a solution to many issues discussed in earlier chapters. Resolving the Multiple Hardware Support issue on The reader abstraction layer that provides a common interface for accessing RFID hardware devices with different characteristics. Resolving Synchronization and Scheduling issues in The EDML, that manages data flow between the other layer handling various reader-level operations, Servicing Multiple Applications and offering a Device Neutral Interface to the applications on The Application Abstraction Layer (AAL), that provides various applications with an independent interface to RFID hardware. We were also successful 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.
Moreover, we explained the use of CEP technology in the ESN middleware, the integration architecture between RFID and WSN, The events of RFID, WSN and their interactions were also analyzed. By adopting CEP technology. We built a middleware system which has the functions of filtering, grouping and aggregation of real-time event data. And

Regulating Access to data by using The RBAC Mechanism that makes sure only authorized users (applications) access the needed data depending on the permissions allowed and the role assigned.

References

[1] M. Ajana, M. Boulmalf, H. Harroud, H. Hamam , "A Policy Based Event Management Middleware for Implementing RFID Applications", IEEE International Conference on Wireless and Mobile Computing, 2009
[2] United States Government Accountability Office, "INFORMATON SECURITY Radio Frequency Identification Technology in the Federal Government," United States Government Accountability Office, May 2005. [Online].
[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, D. Donsez, "Middleware Building Blocks for Architecting RFID Systems", Mobile Lightweight Wireless Systems, Vol 13, pp 325-336. 2009
[6] X. Su, C. Chu, B. S. Prabhu, 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. and Kuhn, D.R. "The NIST Model for RoleBased Access Control: Toward a Unified Standard", 5th ACM Workshop Role-Based Access Contro, pp: 47–63, 2000
[13] D.F. Kuhn, D.R. Sandhu, "RBAC Standard Rationale: comments on a Critique of the ANSI Standard on Role-Based Access Control", IEEE Press. 5 (6), pp: 51–53 , 2007
[14] M. Zuluaga, J. Montanez, J. van Hoof, "Green Logistics – Global Practices and their Implementation in Emerging Markets", pp: 2-3, Colombia, 2011
[15] R. Sandhu, Edward J. Coynek , Hal, L. Feinsteink and Charles E. Youman "Role-Based Access Control Models" , IEEE Computer, Vol 29 (2), pp: 38-47, 1996,
[16] R. Sandhu "Role-Based Access Control (RBAC)", CS 6393 Lecture 3, 2016
[17] T. Zhang, Y. Ouyang, 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, E. Coyne, "ABAC and RBAC: Scalable, Flexible, and Auditable Access Management", IT Professional, vol. 15 (4), pp. 14-16, 2013
[19] J. Caiyuan, Shen, Aodong,Y.Wenxue, "The RBAC System Based on Role Risk and User Trust", International Journal of Computer and Communication Engineering, 2016