frontiers | Frontiers in Mechanical Engineering

TYPE Original Research
PUBLISHED 09 October 2023
DOI 10.3389/fmech.2023.1242612

OPEN ACCESS

EDITED BY
Augustine O. Nwajana,
University of Greenwich, United Kingdom

REVIEWED BY
Fuqiang Zhang,
Chang'an University, China
Richard Otuka,
Nottingham Trent University,
United Kingdom
Ndidi Arinze,
Enugu State University of Science and Technology, Nigeria
Emenike Raymond Obi,
RaySoft AssetAnalytics, Canada

*CORRESPONDENCE
Achraf Haibi,
achraf.haibi@gmail.com

RECEIVED 19 June 2023
ACCEPTED 01 September 2023
PUBLISHED 09 October 2023

CITATION
Haibi A, Oufaska K, El Yassini K, Boulmalf M and Bouya M (2023), A new RFID Middleware architecture based on a hybrid security technique using data encryption and RBAC for modern real-time tracking applications.
Front. Mech. Eng 9:1242612.
doi: 10.3389/fmech.2023.1242612

A new RFID Middleware architecture based on a hybrid security technique using data encryption and RBAC for modern real-time tracking applications

Achraf Haibi[1,2,3]*, Kenza Oufaska[4], Khalid El Yassini[2], Mohammed Boulmalf[3] and Mohsine Bouya[4]

[1]Research, Development and Innovation Laboratory, Mundiapolis University, Casablanca, Morocco, [2]IA Laboratory, Faculty of Sciences, Moulay Ismail University, Meknes, Morocco, [3]TIC Laboratory, College of Engineering and Architecture, International University of Rabat, Rabat, Morocco, [4]LERMA Laboratory, College of Engineering and Architecture, International University of Rabat, Rabat, Morocco

Radio Frequency Identification (RFID) is a contactless technology that has developed over the 90s and 20th centuries. It employs electromagnetic or electrostatic coupling in the radio frequency part of the electromagnetic spectrum to uniquely identify traceable objects, and is widely used in various sectors (e.g., medical, Supply Chain Management, transportation, and IoT applications.). Through the supply of real-world monitoring and context information about things, the integration of this technology in such areas delivers various benefits in the future of ubiquitous computing. However, one of the primary challenges will be the capacity to manage data since RFID events have specific characteristics and requires special treatment, such as the large volume of data flow, inaccuracy, temporal and spatial data, are typical examples of RFID event data. The goal of this research is to first highlight the concerns and limitations of existing middleware architectures before introducing and implementing a new Middleware architecture to address the identified issues, specifically real-time processing of massive volumes of data coming from physical RFID infrastructure. This middleware combines role-based access control with an encryption algorithm to increase security, a NoSQL database for storing large amounts of data, complex event processing (CEP) to provide high-volume data stream processing, and improved interoperability via the Data Transformation Module. Finally, our architecture is evaluated and compared to several middleware architectures based on standard ISO/IEC 9126 metrics.

KEYWORDS

RFID technology, RFID middleware, RBAC, blowfish algorithm, CEP

1 Introduction

Communicating, identifying and detecting objects are natural and trivial activities for humans, but when it comes to computer systems, they are quite complex. RFID technology has the power to circumvent these difficulties by being able to give objects the ability to communicate their presence and their identity (Ropraz, 2008a). In other words, RFID makes it possible to connect things to the Internet, so that companies can track them and share data about them, which promotes the new concept of the Internet of Things. RFID technology uses radio waves to transmit and receive information (Kreowski et al., 2009) to enable the automatic identification of objects on which the tags are mounted in order to extract information for flexible, reliable and permanent real-time traceability (Ishikawa et al., 2003). The last few years have seen the rise of radio frequency identification or RFID technologies as manufacturers began to take a close look at the potential of radio frequency waves for identification purposes. The application areas have continued to diversify, and the technology can be found in many areas such as IoT applications, healthcare, transportation, Supply Chain Management (SCM), animals, and sensing systems (Aqeel-ur-Rehman et al., 2008; Venkatalakshmi et al., 2011; Marczewski et al., 2016; Bridge et al., 2019; Lin et al., 2022; Ahmad Kamal et al., 2023). The RFID system comprises four elements (Hu et al., 2008): an RFID reader, RFID tags, business applications, and a software layer known as "RFID middleware" (Ryu et al., 2011); the latter is the main component of the RFID architecture (Ajana et al., 2009a), it plays a vital role as it offers a range of functionalities that ensure efficient RFID data management. Designing RFID Middleware is an evolving research area; various approaches and designs have been suggested in the literature to implement RFID middleware architectures. According to the findings, the industry has adopted a standard RFID middleware architecture based on the EPCglobal Application-Level Events (ALE) standard. However, since RFID is more incorporated in complex applications, this specification methodology has demonstrated its incapacity to cope with the constraints of a range of application domains (Amaral et al., 2011). Among these constraints are the storage of massive RFID data, and also the semantic processing of RFID event data in real-time, as it provides neither dedicated big data tools nor efficient support for complex event processing (CEP), as for such modern applications the ability to react in a timely manner to the occurrence of real situations in the system environment has become a fundamental requirement (Liu et al., 2015).

In this work, we explored the problems related to the development of RFID middleware for new applications such as IoT applications by presenting an RFID framework that aims to increase the integrability of the RFID middleware system in modern applications, supported by CEP network concept for the monitoring and reporting of complex events spread over different sites. Besides the storage, processing and monitoring of large RFID data stream, the security of RFID data is one of the key goals of this work, as security is the main issue facing every user, because security is the major issue facing every user. To secure the transmission of data, cryptographic techniques can be employed. Of all cryptographic algorithms, the Blowfish algorithm is the best in terms of execution time, memory usage, throughput, power consumption, and security (Suresh and Neema, 2016) and therefore well suited for RFID applications. To increase security, we proposed a security module based on a combination of the Blowfish algorithm and the RBAC tool to ensure that only authorised users can access the decrypted data.

The remainder of this paper is organized as follows. First, we present an overview of RFID systems. Secondly, we survey RFID Middleware architectures, focusing on the different functionalities, characteristics, and standards contained in each architecture. Section 4 analyses current RFID middleware implementations and describes their main issues and shortcomings. Section 5 details our proposed Middleware architecture components, the implementation and the evaluation, and Section 6 concludes the paper.

2 RFID technology overview

2.1 General concept

Radio Frequency Identification (RFID) is an identification system (Ait Lhadj Lamin et al., 2021) developed to store and retrieve data remotely. It uses a small chip coupled to a miniature antenna (Rouchdi et al., 2018a), which together form markers called RFID "Tag" or RFID "Transponder". The chip stores in its memory (depending on the memory type) different data. The tag in question will be attached to all sorts of objects thus making it possible to identify them, follow their path and find out their characteristics. Then, to exploit the information contained in these tags, it is imperative to have the RFID reader which uses radio signals to communicate with the antenna of the tag and convert the radio waves into data that can be read by an RFID software called RFID Middleware (Burnell, 2008). It should be noted that in recent years, an increasing amount of research has been done with the aim of introducing a new solution called RFID chipless technology (Perret, 2017). In this technology, the tag is equipped with a planar encoder and sometimes an antenna to communicate with the reader (Herrojo et al., 2019).

2.2 Basics of RFID technology

2.2.1 Operating principle

In general RFID uses electromagnetic waves to transmit and receive information stored in a tag to or from a reader (Gabsi et al., 2022). The RFID operation system is given in Figure 1, which presents a passive or semi-passive RFID system that communicates using the backscattering principle. The RFID reader interrogates the tag and eventually supplies it with the energy it needs to operate via radio waves on a specified frequency. The tag converts a part of the received signal into energy and uses it to respond to the reader (in case of a passive or semi-passive tag). A dialogue is then established between the reader and the tag according to a well-defined communication protocol. The middleware then collects and processes the RFID data and is responsible for monitoring the RFID readers, filtering, formatting, aggregating, and storing the collected RFID events for further processing and exploitation by the applications concerned. The scanned data is transmitted to the user application for exploitation.

2.2.2 RFID system components

2.2.2.1 RFID tags

The tag consists of a substrate on which is deposited an antenna connected to the RFID chip (Bouazza et al., 2020). A tag provides a link between the electromagnetic wave coming from the reader and the energy transmitted to the chip. It also allows communication with the reader. The antenna is responsible for transmitting and receiving RF waves, enabling communication (Amin, 2013). Its

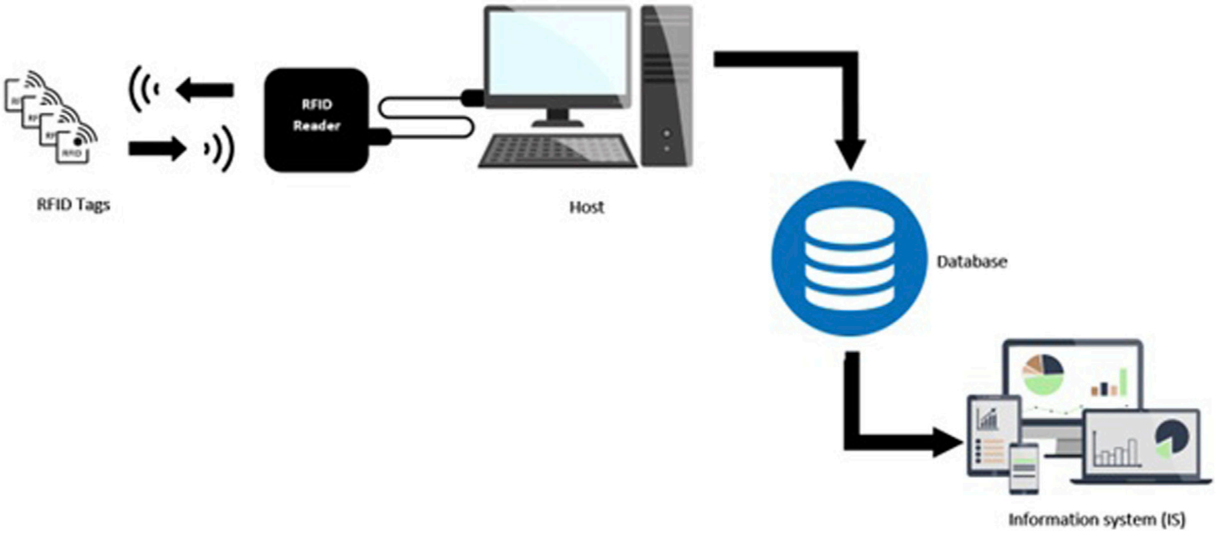


FIGURE 1
Rfid system components (Haibi et al., 2022a).

geometry depends on the type of coupling (near field, far-field) and the operating frequency. As mentioned earlier, there are many applications demanding the identification of objects remotely. Such applications are constrained by different reading ranges, object types, and the environment in which the tags are applied (Rance et al., 2017). This large range of applications and their particular constraints explain to some extent the broad range of RFID technologies that can be found.

The RFID tags are classified according to different criteria (Rance et al., 2017). A first criterion determining the price of the tag is the presence or absence of a power source within the tag. According to this criterion, there are three types of tags (Chen et al., 2010):

- Passive tags: As they do not have a power source, they use the energy propagated by the reader's radio signal and picked up by the tag's antenna (Nash, 2010). They use the principle of remote powering to recover the energy supplied by the reader during a communication. This type of tag offers the advantage of being inexpensive and absolutely maintenance-free.
- Active tags: have both an on-board power source and an RF transmitter (Pupunwiwat, 2012), allowing them to emit a signal autonomously to the RFID reader. They are therefore configured as bidirectional radio communication devices. The energy source can be in the form of a battery or from an alternative energy source (solar, heat, movement, etc.). The magnetic or electromagnetic field received from the reader is therefore not necessary to power the chip. This means that the field can be much smaller than the field required to operate a passive transponder. This condition considerably increases the communication range.
- Semi-passive RFID: Similar to passive tags, it uses a chip without an RF transmitter but with a power source. It does not use its battery to transmit the signals to the reader, this energy source is used to power other external components like sensors.

RFID tags are also classified according to the type of memory, in this context three types of transponders are distinguished (Boontrai et al., 2009):

- Read-Only: preprogramed passive tag, i.e., they are defined once and for all by the manufacturer, which makes it readable several times.
- Write Once, Read Many (WORM): WORM tags are delivered blank by the manufacturer. The user can then enter data once and it is always possible to read them as many times as required.
- Read-Write RFID transponders in this category are rewritable. The content of these transponders can be modified, deleted, and rewritten several times, with a lot of read and write access.

Another classification, besides the memory type, is communication frequency. The communication between the tag and the reader takes place on different frequency ranges. The different radiofrequency bands used in RFID are Low frequency (LF), High Frequency (HF), Ultra-High Frequency (UHF) and Microwave Frequency (Marrocco, 2008; Khaddar et al., 2011).

2.2.2.2 RFID reader

The RFID reader is an essential device for the use of the RFID system. It creates a reading area through radio waves, and the tags located in its magnetic field will be enabled to transfer its information. The communication between the RFID interrogator and the tag is possible thanks to RFID antennas integrated with each of the 2 components. In the case of passive RFID tags, the reader supplies the RFID tag with energy through radio waves so that the chip sends the information it contains, and the reader receives the tag's responses and transmits them to the middleware (Turcu, 2011).

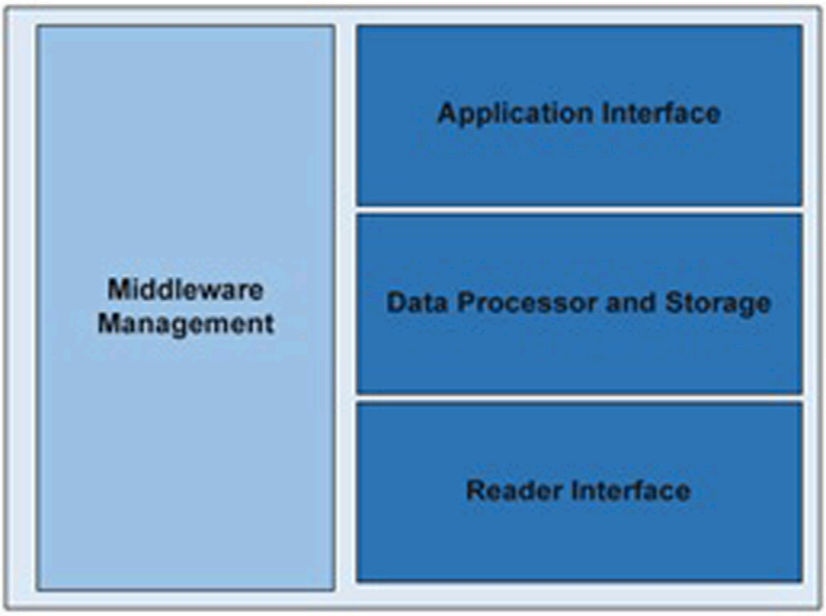


FIGURE 2
Middleware architecture (Rouchdi et al., 2018b).

RFID readers can take various forms depending on the use for which they will be intended (Ajana et al., 2011), the most used reader is the fixed reader, but it can also take the form of a handled reader (Venot, 2015).

2.2.2.3 RFID Middleware

The RFID middleware is a set of software applications that serve as a bridge between the hardware components (physical RFID infrastructure) and the software components of RFID systems (Fan and Wu, 2012). It manages the events resulting from the data capture equipment (the RFID readers). The middleware is responsible for the following main functions: data transmission, filtering, and the conversion of data formats between the RFID hardware and the IS (Haibi et al., 2019; Haibi et al., 2021a).

2.3 Middleware components

As presented in Figure 2, generally, the software tool called "RFID Middleware" comprises four software layers:

- Reader Interface–Middleware Management.
- Application Interface–Data Processor and Storage.

2.3.1 Reader interface

Sometimes in real cases of RFID technology adoption, we find different RFID products from different manufacturers, which poses the heterogeneity problem. From this arises the importance of the Reader Interface layer. It is the lowest in middleware architecture, oversees managing all the middleware interactions with the various components of the physical RFID infrastructure; it also provides drivers for various devices supported by the middleware (Amaral et al., 2009).

Therefore, the Reader Interface layer controls all hardware-related parameters such as Air Interface, Reader Protocol, and the host-side communication. It provides a uniform communication interface between RFID devices and the rest of the middleware layers, allowing applications to operate independently of RFID readers.

2.3.2 Data processor and storage

It is the layer responsible for many services that an RFID middleware is supposed to provide (Anouar Abdelhakim Boudhir et al., 2019); it processes the raw data flow sent by the RFID readers based on the reading events management (reading the data stored in the RFID tags). Given the data volume received from RFID tags, this requires a filtering & aggregation feature that is responsible for eliminating data that applications do not need in particular duplicates, in order to select only the useful data for applications, once the data has been filtered and aggregated, the data can be transmitted to the upper layer of the architecture (Amaral et al., 2009). In fact, rather than sending useless data over the network, applications receive only useable data.

2.3.3 Application interface

With the aim of providing client applications with access to the various services offered by the middleware, this layer ensures the interfacing of the latter with the back-end applications while ensuring the abstraction of the system. Application Interface provides the necessary resources to allow the applications to request these services as well as the desired data and let us not forget the operations' execution in one or more RFID readers.

2.3.4 Middleware management

Ensures the provision of information on all running processes. It provides the ability to:

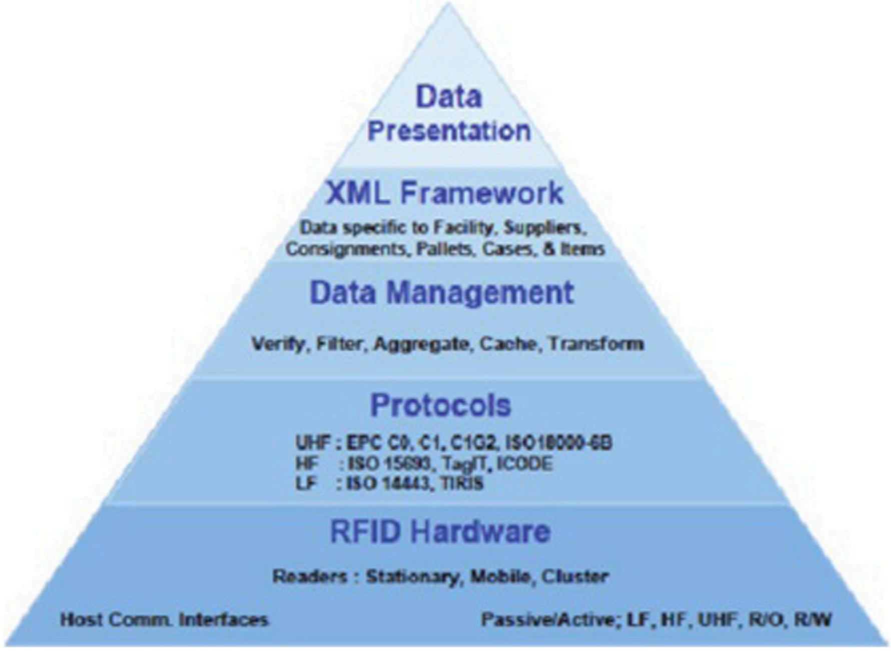

Data Presentation
XML Framework
Data specific to Facility, Suppliers, Consignments, Pallets, Cases, & Items
Data Management
Verify, Filter, Aggregate, Cache, Transform
Protocols
UHF : EPC C0, C1, C1G2, ISO18000-6B
HF : ISO 15693, Tag/T, ICODE
LF : ISO 14443, TIRIS
RFID Hardware
Readers : Stationary, Mobile, Cluster
Host Comm. Interfaces
Passive/Active; LF, HF, UHF, R/O, R/W

FIGURE 3
WinRFID architecture (Prabhu et al., 2006).

- Remove, add, and modify RFID devices linked to the system.
- Edit parameters by back-end applications
- Enable and disable middleware supported functions.

Based on the above, it is concluded that typical middleware may need to support these different features: Hardware Abstraction, Duplicate removal & Filtering, Data Aggregation, Data Formatting, and Application Connector.

3 Literature review

3.1 Existing RFID middleware architectures

With the aim of addressing as much relevant research works as possible that discuss the middleware component, we conducted searches on a set of literature references respecting the detailed search strategy described in (Brereton et al., 2007; Haibi et al., 2022b). We considered the complete reading of the selected documents to ensure a complete and correct assessment.

3.1.1 WinRFID

It is an RFID middleware developed by the.NET Framework; as shown in Figure 3, it contains 5 layers which makes it among the multi-layer middleware architectures (Prabhu et al., 2006). This architecture components are listed below:

- **RFID Hardware:** It administrates Tags, Readers, and other sensors (hardware part). It is in charge of abstracting this system part in order to make easier the management and the addition of new components (Perret, 2017) (the input/output components), as it offers a unified interface to the physical RFID infrastructure devices.
- **Protocols layer:** It is responsible for abstracting the reader-tag protocols. It allows the middleware to support multiple communication protocols (e.g., ISO 15693, ISO 18000-6B, ICode, EPC Class 0, etc.), and depending on the reader used, this layer chooses the appropriate protocol when communicating with a tag.
- **Data Management:** Is responsible for processing RFID readers' data flows (removing duplicates, checking label readings, etc.).
- **XML Framework:** XML Framework layer processes the data from Data Management layer, formats it in XML data in order to make it presentable for later use by different enterprise applications.
- **Data Presentation:** It uses the data that comes from XML Framework to visualize the data according to end-users' needs (tables, graphs, etc.).

3.1.2 RF^{2}ID

It is an RFID middleware that has the following objectives: Reliability, Load Balancing, High Throughput, Scalability, and Data Organization. As shown in Figure 4, this middleware principle is based on these two notions: virtual reader abstraction and Vpath (Virtual Path) (Ahmed et al., 2007).

- **Virtual reader abstraction:** it manages a set of physical readers that are located in the same neighbourhood.
- **Vpath:** (comprised of a set of VRs) to capture the logical flow of information between Virtual readers as RFID tagged objects move through the environment.

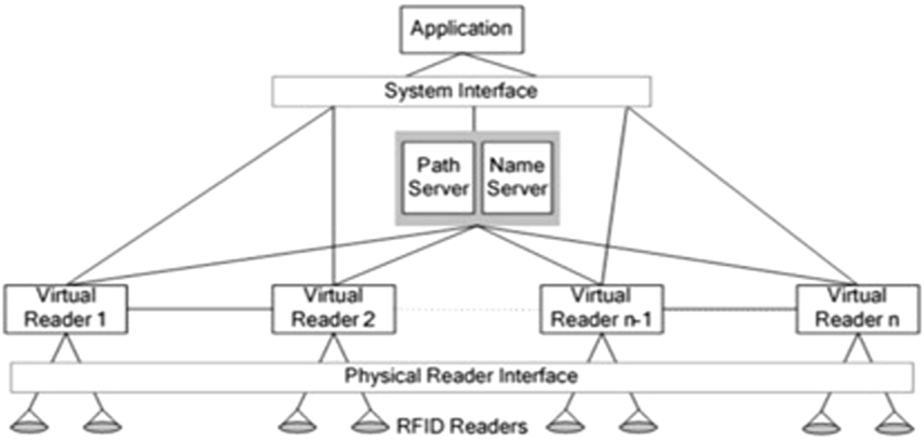


FIGURE 4
RF²ID architecture (Ahmed et al., 2007).

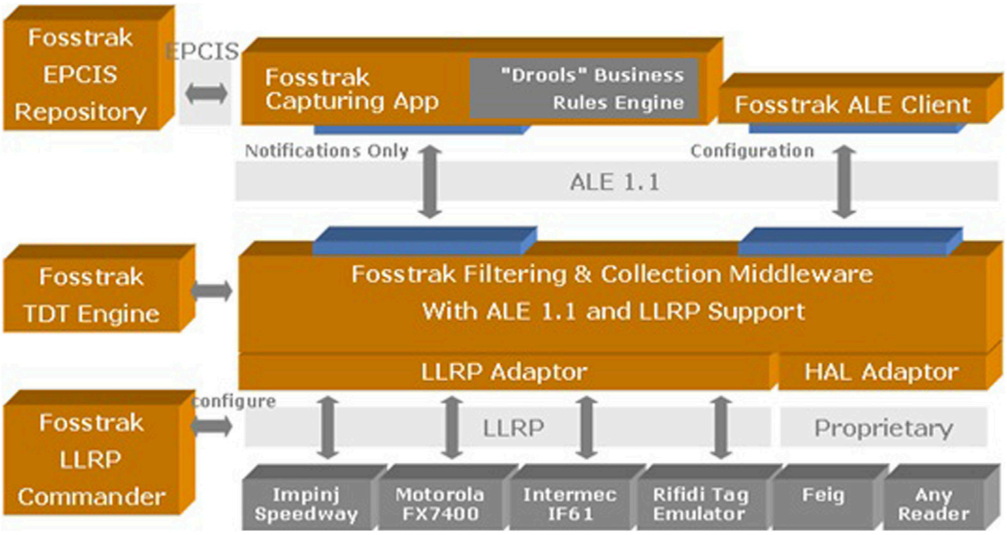


FIGURE 5
Fosstrak architecture (Floerkemeier et al., 2007).

3.1.3 Fosstrak

Fosstrak for Free and Open-Source Software for Trace and track (previously called Accada platform), is an open-source platform that implements the specifications of EPCglobal Inc. This RFID middleware is designed to meet the needs of tracking and tracing applications (Floerkemeier et al., 2007). As shown in Figure 5, this platform consists of three layers.

- **EPCIS (EPC Information Services):** it receives data from the middleware and transforms it into a more suitable format for enterprise/business applications.
- **Reader:** this module implements the EPCglobal specification. It provides several features, such as filtering and aggregation. It supports a large number of physical readers as well as a simulation mode for RFID readers.
- **Middleware:** this component allows applications to define a subscription in which they define the readers to be used, the type of data they are interested in, their formats. . .

3.1.4 AspireRfid middleware platform

ASPIRERFID is an RFID middleware compatible with EPCglobal standards and others; it is developed within the framework of a project named ASPIRE by ON2 and implements the NFC Forum and OSGi Alliance specifications, as well as several Java Community Process (JCP) specifications (Kefalakis et al., 2008). As can be seen from Figure 6, layers constituting this middleware are:

- Hardware abstraction layer
- Reader Core Proxy

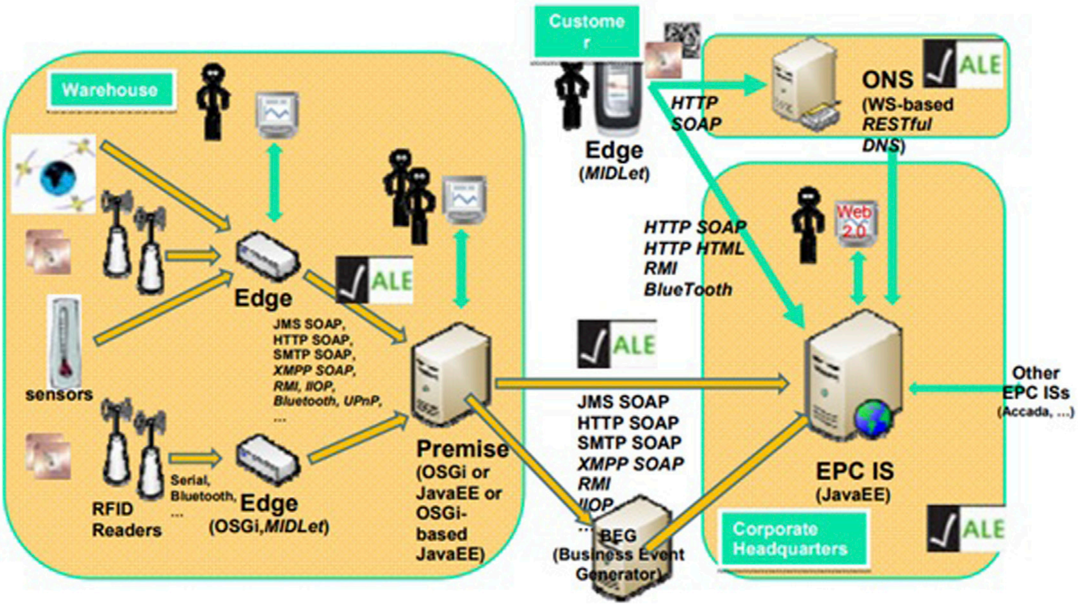


FIGURE 6
AspireRfid Middleware architecture (Kefalakis et al., 2008).

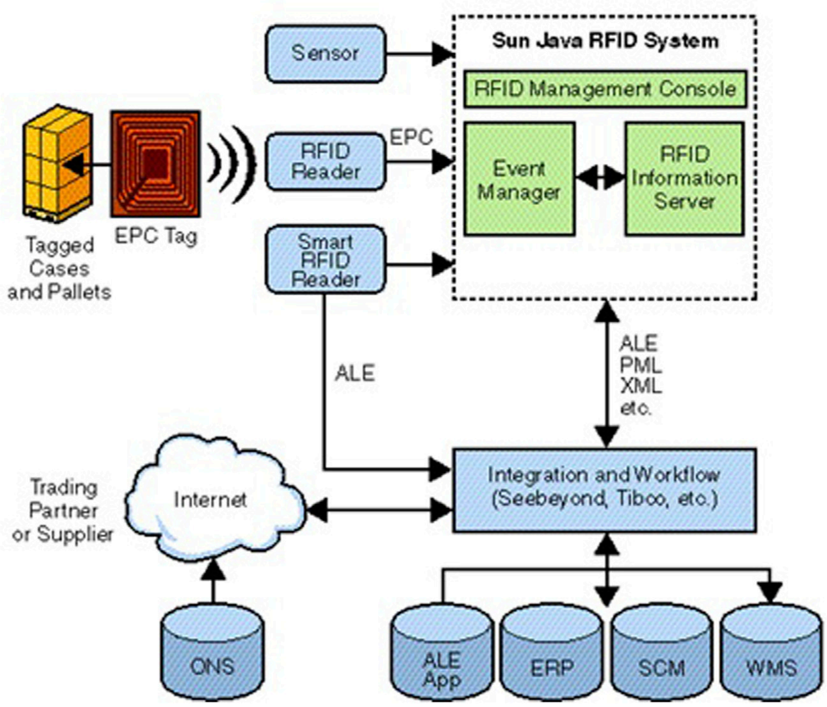


FIGURE 7
Sun java system architecture (Gupta and Srivastava, 2004).

- Filtering and Collecting
- Business Event Generator
- Integrated Development Environment

3.1.5 Sun java system

It is a Middleware platform (Java-based) designed by Sun Microsystems Inc. that supports EPCglobal standards in its

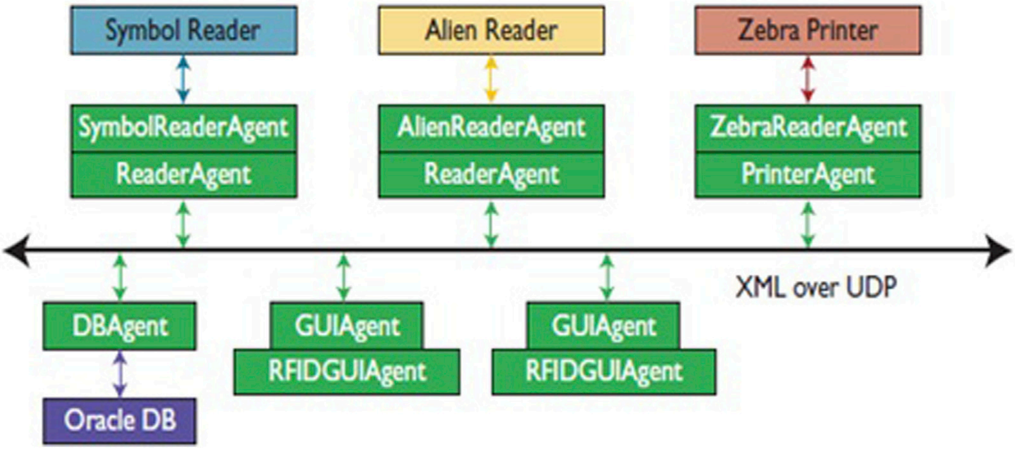


FIGURE 8
Tagcentric architecture (Hoag and Thompson, 2006).

design. As shown in Figure 7, its main components are the Event Manager and Information Server (Gupta and Srivastava, 2004):

- **Event Manager RFID:** it is based on Jini and its main objectives are to interface with the hardware part (tags or sensors and readers) to collect, filter, and transmit events to the RFID information system.
- **Information Server:** it is a J2EE application that serves as an interface for EPC data acquisition and query, used to transform the collected data to a high level of representation that is more suitable for enterprise applications. This component runs on the Sun Java System Application Server. It also interfaces with other information systems through the exchange of messages in XML.

3.1.6 Tagcentric

Is an open-source agent-based middleware developed by Arkansas University that collects RFID data and stores it in a database of choice (Oracle or MySQL). TagCentic allows managing heterogeneous RFID readers and RFID tag printers. This Middleware supports several popular RFID readers (Alien, Symbol, and ThingMagic) and supports simulated readers such as Rifidi (Hoag and Thompson, 2006). Figure 8 depicts the architecture of Tagcentric.

3.1.7 LIT middleware

LIT Middleware acronym for Logistics Information Technology Middleware is an RFID middleware based on both EPCIS and ALE layers in its implementation. The layer named ALE consists of the following four sub-layers (see Figure 9) (Kabir et al., 2008):

- **Application Abstraction Layer:** provides access to RFID data via ALE API.
- **State-based Execution Layer:** it is the core of LIT Middleware; it consists of different components, Thread Pool, controller, query manager, scheduler, and reader manager.
- **Continuous Query Layer:** it is the layer responsible for collecting, filtering, and removing duplicates.
- **Reader Abstraction Layer:** it provides a common interface for heterogeneous RFID devices.

EPCIS is composed of three sub-layers:

- Capturing Service Layer - Query Service Layer
- Repository Layer

3.1.8 Lightweight RFID middleware for WMS

This work represents lightweight middleware that supports data acquisition, processing, and download. The architecture structure is illustrated in Figure 10 (He et al., 2013).

This middleware architecture is based on the MySQL DBMS, which will provide the back-end applications with the required RFID data via SQL statements.

Lightweight RFID middleware architecture is composed of:

- Reader Connection Interface: this module consists of two sub-modules: Management of Reader RFID reader connection program.
- Data Processing Module
- Interface Connected to the Application Program

3.1.9 Lightweight-ALE-based embedded RFID middleware

It is a lightweight embedded RFID middleware architecture based on the ALE standard. It is characterized by an event-handling mechanism, and it provides a unified interface (Liu et al., 2009).

As shown in Figure 11 the middleware architecture consists of:

- **Device Manager:** Thanks to this module, the middleware supports heterogeneous RFID devices; to manage this equipment it provides a unified management interface.
- **Data buffer pool module:** It is the module responsible for the data interaction between the upper layers like ALE module and the application established on the device management module on the one hand, and Device Manager module on the

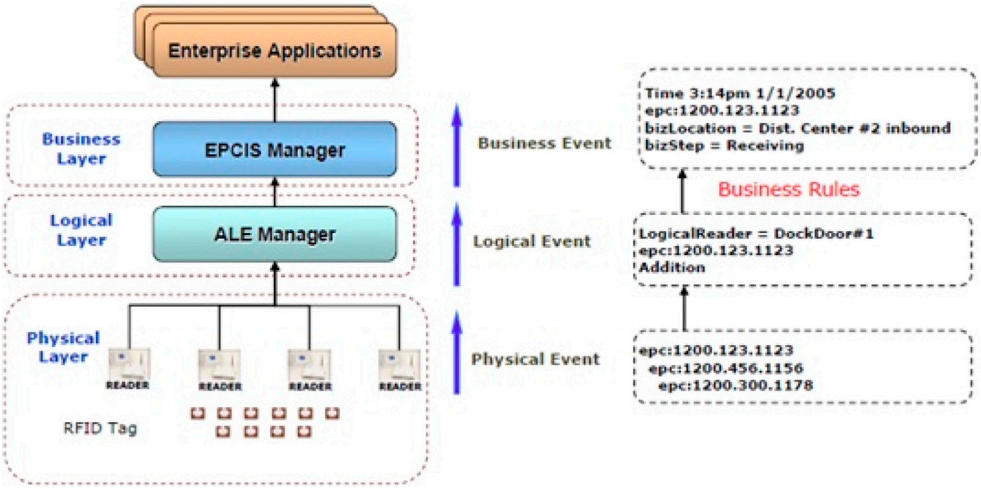


FIGURE 9
LIT Middleware architecture (Kabir et al., 2008).

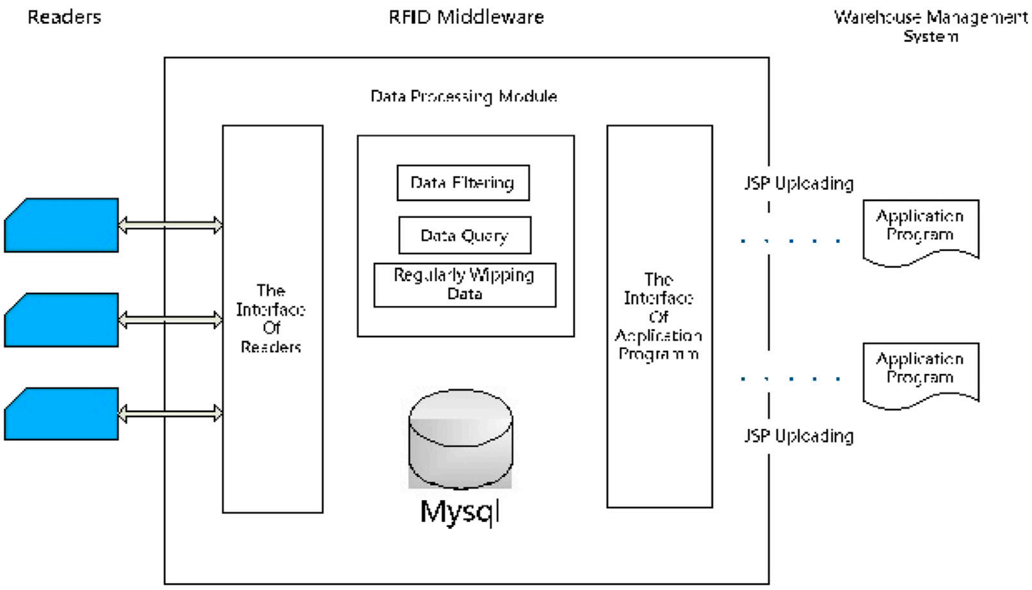


FIGURE 10
Structure of Lightweight RFID middleware (He et al., 2013).

other hand, which makes the module responsible for RFID data distribution.

- **Lightweight ALE module:** It is the module that retains the important key functions of the standard ALE, making it the responsible module for RFID event management. Via the generic unified interfaces of the event processing the application users define the criteria for grouping and filtering of RFID data, plus the reporting model.
- **Middleware configuration management module:** All middleware configuration is performed by this module, through it the user can configure the parameters of the data buffer pool, RFID equipment, as well as the use of the Mobile proxy module and Lightweight ALE module.
- **Mobile proxy module:** This module ensures network communication and provides reconnection functions in the event of disconnection.

3.1.10 RFID middleware with database

This middleware, which is based on a simple architecture, uses almost less than 20% of the functions provided by the EPC system. It has a low-cost compact system compared to ALE structure for convenient use by small and medium-sized companies (Chen et al.,

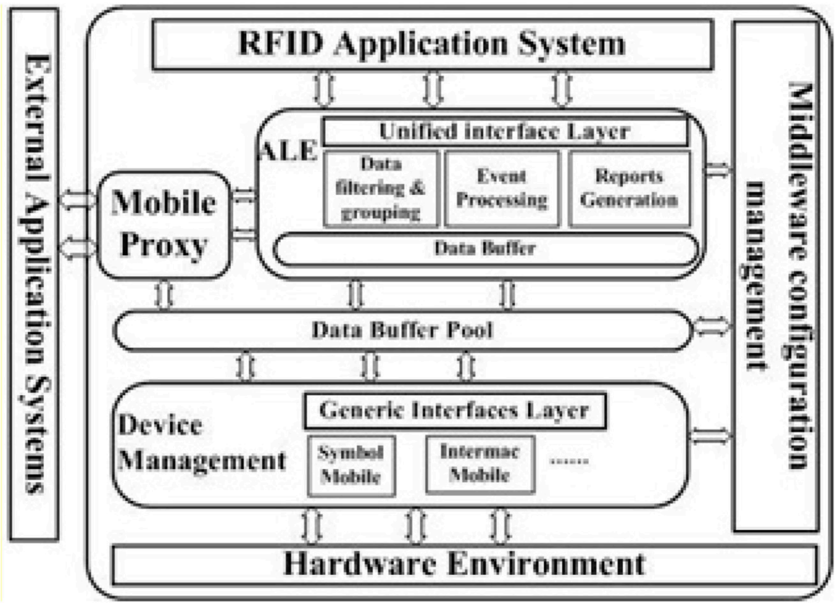


FIGURE 11
The lightweight-ALE-based Embedded RFID Middleware architecture (Liu et al., 2009).

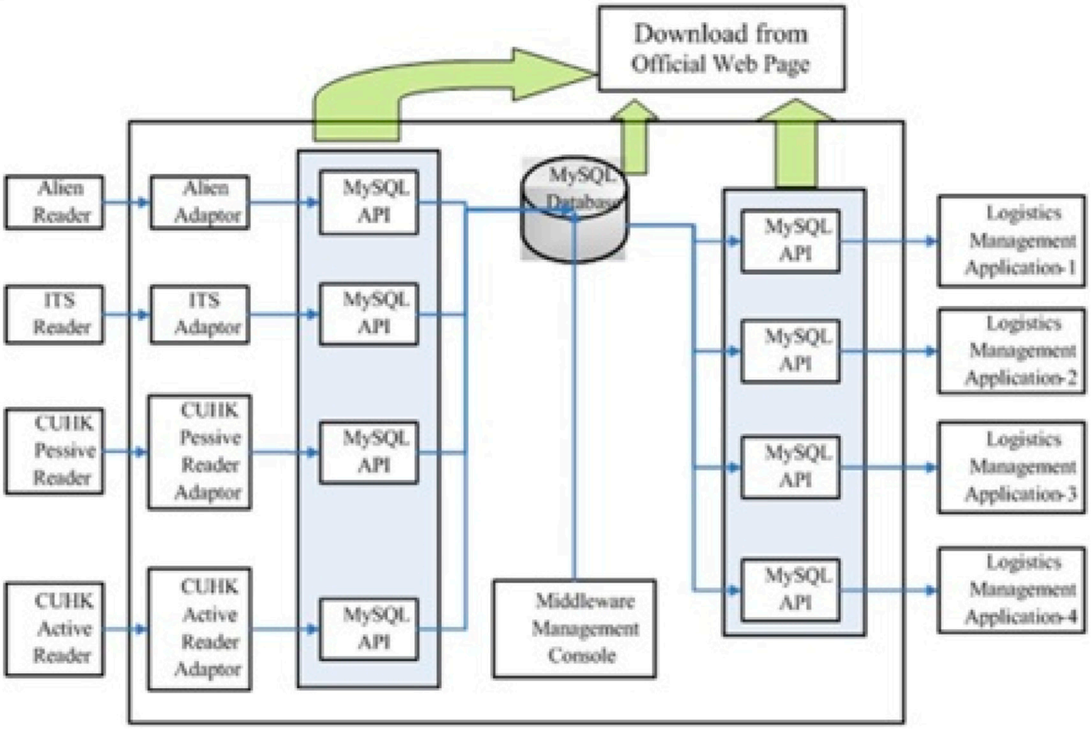

Download from Official Web Page
Alien Reader
Alien Adaptor
ITS Reader
ITS Adaptor
CUHK Pessive Reader
CUHK Pessive Reader Adaptor
CUHK Active Reader
CUHK Active Reader Adaptor
MySQL API
MySQL API
MySQL API
MySQL API
MySQL Database
Middleware Management Console
MySQL API
MySQL API
MySQL API
MySQL API
Logistics Management Application-1
Logistics Management Application-2
Logistics Management Application-3
Logistics Management Application-4

FIGURE 12
The Middleware architecture (Chen et al., 2017).

2017). This middleware architecture (Figure 12) has different aspects from the EPC system middleware, the differences are listed below:

- **Supported formats:** the proposed architecture supports different ID formats, plus the 2 formats ISO18000-6C (EPC C1G2) and ISO180006B, it also supports active tags; knowing

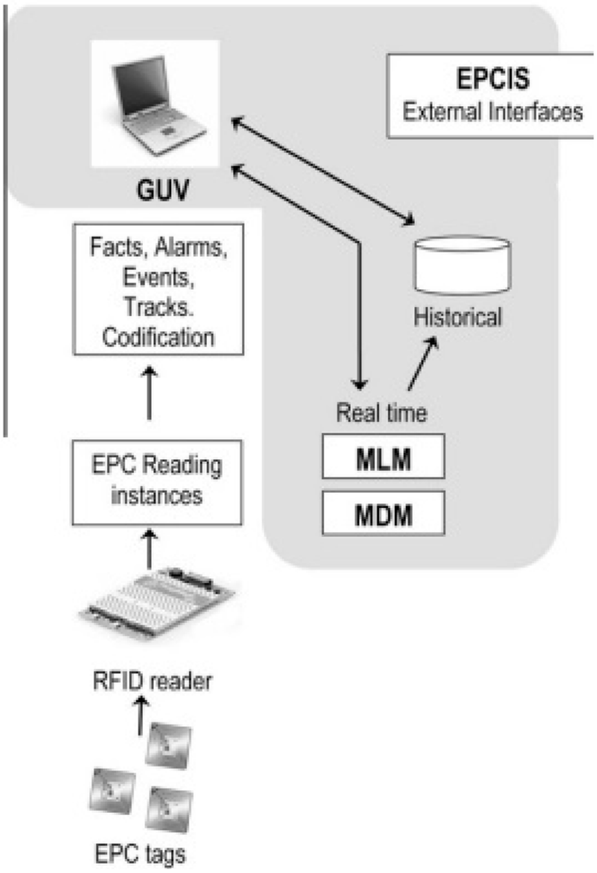


FIGURE 13
Architecture of DEPCAS (Cardiel et al., 2012).

that EPC Middlewares only support IDs of tags in GID-96 format.

- **Interface:** it aims to connect the reader with the application program based on interfaces given by MySQL.
- **EPCIS-related:** EPCIS cannot be used for this architecture because registration is not needed. The program which is responsible for the database queries in order to get production information.
- **Applicable extension:** System administrators in SMEs can program this middleware, as it is characterized by clear structure, which makes it extensible with less maintenance and convenient.
- **Filtering:** The filtering process here is easier than the EPC middleware because it does not require registration, and the data collected is very detailed and a database lookup table can perform the filtering.

3.1.11 DEPCAS middleware

Figure 13 depicts the overall DEPCAS architecture. The scheme proposed here is based on the architecture of modern SCADA (Supervisory Control and Data Acquisition) systems (Cardiel et al., 2012). The key issues of DEPCAS are as follows:

(1) Hiding heterogeneous RFID deployment systems with a homogeneous layer approach.
(2) Producing RFID processed data.
(3) Providing management capabilities in the RFID middleware environment.
(4) Translating business (async or sync) needs to RFID systems.

This Middleware system is organised into four sub-systems:

- Graphical User Viewer.
- Middleware Logic Manager.
- Information system exchange (EPCIS).
- Middleware Device Manager.

3.1.12 SafeRFID

Is an RFID middleware is programmed in Java and is based on the LLRP protocol to manage RFID readers and retrieve the data (Kheddam et al., 2013). The strength of this architecture is fault tolerance through two mechanisms, namely the online diagnostic algorithm based on statistical analysis of RFID events that guarantees to detect defective elements on the physical RFID infrastructure. And the second mechanism which is a verification process based on an extended finite state machine of LLRP.

As shown in Figure 14, SafeRFID is composed of three layers:

- **Data Processing Layer:** It is the layer in charge of carrying out most of the functionality provided by this middleware, as data aggregation and filtering, etc., which makes it the most essential layer in this proposed architecture.
- **Hardware Abstraction Layer:** It is the layer that provides the link between the physical RFID infrastructure and the application operations.
- **Application Abstraction Layer:** It offers an interface for back-end applications to reach out to the different middleware features.

3.1.13 MedRFID

Represents an RFID middleware that includes, in addition to the standard features that an RFID middleware should offer, areas of innovation especially related to mobility and manufacturer's autonomy.

The architecture of the MedRFID Middleware is shown in Figure 15. It is composed of 7 layers:

- **Collect:** This is the module that oversees the collected RFID data, writing on tags, managing readers, unifying the data format regardless of the reader type.
- **Mobile Application:** It is responsible for mobile type readers, as the previous layer; it unifies the data format of RFID tags regardless of the type of reader.
- **Administration:** Manages all matters related to the middleware settings, i.e., he is responsible for all changes in the parameters of the antennas, readers, and sensors.
- **Tag Process:** This layer is made up of 3 other sub-layers: Tags Format, Tags Filter, and Tags Aggregation. It is responsible for the RFID tag marking process (Filtering, eliminating duplicates, time stamping, contextualization, and/or formatting).
- **Connectors:** This is the layer that delivers the collected RFID data to back-end applications.

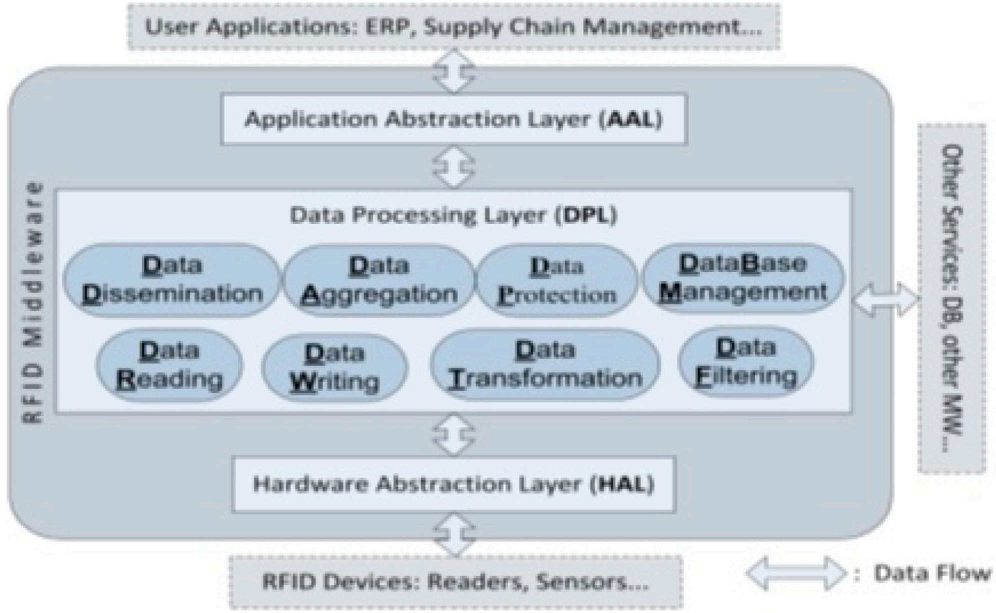

User Applications: ERP, Supply Chain Management...
Application Abstraction Layer (AAL)
RFID Middleware
Data Processing Layer (DPL)
Data Dissemination
Data Aggregation
Data Protection
DataBase Management
Data Reading
Data Writing
Data Transformation
Data Filtering
Hardware Abstraction Layer (HAL)
Other Services: DB, other MW...
RFID Devices: Readers, Sensors...
: Data Flow

FIGURE 14
Architecture de SafRFID-MW (Kheddam et al., 2013).

- **The User Interface:** It represents all the GUIs. It consists of three panels: Administration, Reads, and Writes.
- **Client's IT System:** In fact, it is not a part of the middleware, but it can be connected to it. It represents an application that can communicate with the data collected and stored by the Middleware.

3.1.14 Flex RFID

FlexRFID is middleware that is part of the Multilayer Middleware family. It consists of four layers: DAL, BEDPL, BRL, and AAL (Ajana et al., 2009b). This is illustrated in Figure 16.

- **DAL/Device abstraction layer:** allows interfacing with the Hardware; thanks to it, the Middleware can support a heterogeneous network of sensors and devices based on these three modules: Device Management and Monitoring Module, Data Source Abstraction Module, Device Abstraction Module.
- **BEDPL/Business Event and Data Processing Layer:** it represents the bridge between the DAL and the AAL, and provides the main functions of the Middleware, which are data dissemination, data aggregation, data transformation, data filtering, removal of duplicates, data replacement, data writing and privacy management.
- **BRL/Business Rules Layer:** this layer allows access or restriction to the Middleware's services and data. It contains the rules allocated to each client application to maintain order, consistency, security, or other ways of making a service successful.
- **AAL/Application Abstraction Layer:** offers a software abstraction that gives it the ability to act as an interface with business applications, and it collects all the requests from these applications.

3.1.15 The WebSphere RFID middleware

IBM WebSphere RFID is a middleware solution that spans the three domains Edge, Premises, and Business Process Integration Domain. It allows the interconnection of RFID equipment with business information systems. This is illustrated in Figure 17. It consists of three components (IBM Corporation, 2009):

- **Premises Server (PS):** is the centrepiece of IBM's solution. It is a J2EE application, considered as an intermediary between the physical world and the world of information technology, which uses the "Business Integration Domain".
- **Device Infrastructure (DI):** DI is a set of licensed technologies provided to manufacturers of programmable RFID equipment such as smart readers. It is an OSGi platform (Open Services Gateway initiative), allowing the customization of the RFID solution for specific needs.
- **Business Integrated Server (BIS):** BIS specifies how IBM's RFID solution connects the RFID platform, namely "Premises Server" and "edge controllers" to the company's information system. It offers a set of services that allow the business to integrate web applications with existing applications.

3.1.16 BTMiddlawre

BTMiddlawre refers to a lightweight RFID middleware architecture. The implementation of its architecture (Figure 18) follows the database approach, and the difference between this architecture and other database-based architectures is that it is DBMS NoSQL-based, which enables it to collect and process Big RFID Data (Haibi et al., 2018).

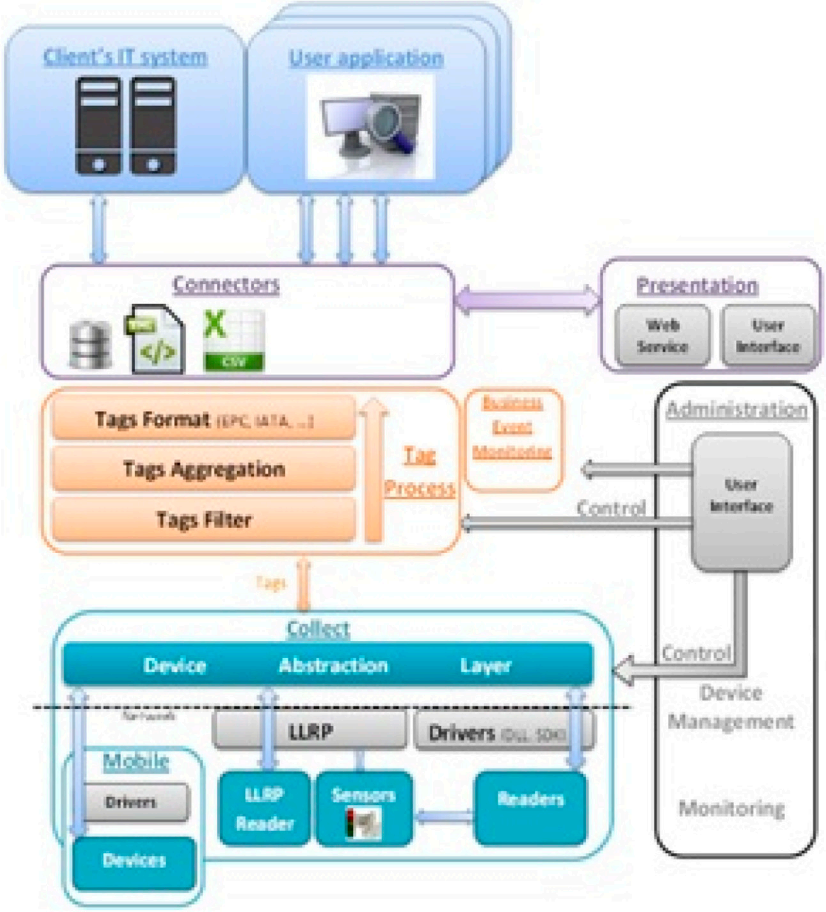


FIGURE 15
MedRFID architecture (Bouhouche et al., 2017).

3.1.17 RFID security Middleware model based on ECDSA

The major difference that characterizes this architecture is that the implementation layer is divided into four different functional modules as shown in Figure 19 (Qiyue and Ping, 2017).

This middleware architecture allows access only to authorised users by verifying their identities. Implementation layer consists of 4 modules:

- **Anti-virus module:** This algorithm-based digital signature module is adopted so that it will be applied when detecting the malicious access as well as the information carrying virus.
- **Digital signature module:** In order for this module to guarantee the authenticity of the information source it authenticates the signed information.
- **Encryption or decryption module:** In order to improve data security, this middleware uses this module to encrypt and decrypt information.
- **Intrusion detection module:** This module has the capability to detect malicious access as well as information carrying virus.

3.1.18 CUHK RFID system

CUHK RFID System is an RFID middleware layer that complies with the EPCglobal standard specifications. It includes an interface that follows the Application-Level Events standard providing read/write functions on the RFID tags and offers IS applications access to the RFID network. The connection with the RFID readers can be done either by IP networks or RS-232 adapters. This middleware allows configuration, control, and monitoring of the RFID equipment network. Figure 20 presents CUHK RFID System architecture. The key Middleware system elements are (Mak et al., 2007):

- **ALE interface module:** It represents a standard API that gives back-end applications the ability to access RFID data. The desired RFID event and data formatting type can be chosen and specified by the applications, and they can also specify a notification channel for the middleware to report in each event cycle.
- **ALE Engine:** It is the most important element of the architecture, providing all the middleware functionalities (data collection, aggregation, filtering, and dissemination according to the application's requirements). It uses a buffer for the temporal storage of all the collected data. According to the application requirements, it filters the RFID data and generates reports at each event cycle.

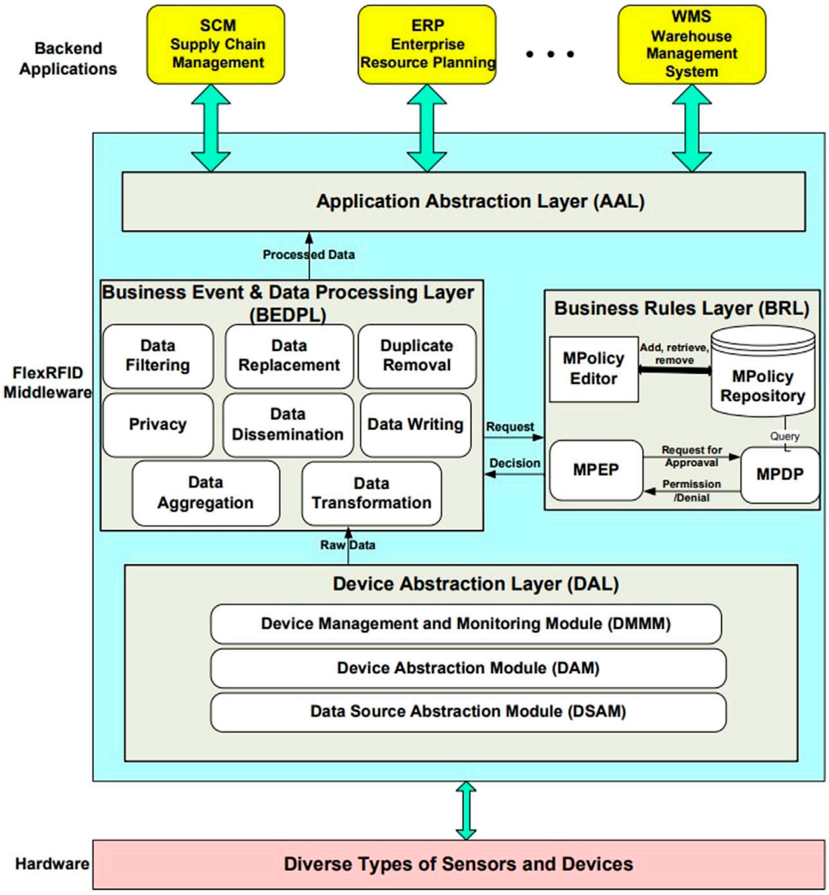

Backend Applications
SCM Supply Chain Management
ERP Enterprise Resource Planning
WMS Warehouse Management System
Application Abstraction Layer (AAL)
Processed Data
FlexRFID Middleware
Business Event & Data Processing Layer (BEDPL)
Data Filtering
Data Replacement
Duplicate Removal
Privacy
Data Dissemination
Data Writing
Data Aggregation
Data Transformation
Business Rules Layer (BRL)
MPolicy Editor
Add, retrieve, remove
MPolicy Repository
Query
Request
Decision
MPEP
Request for Approval
Permission /Denial
MPDP
Raw Data
Device Abstraction Layer (DAL)
Device Management and Monitoring Module (DMMM)
Device Abstraction Module (DAM)
Data Source Abstraction Module (DSAM)
Hardware
Diverse Types of Sensors and Devices

FIGURE 16
Flex RFID architecture (Ajana et al., 2009b).

- **Reader emulator:** This module is an RFID reader emulator, which allows to test an RFID equipment network without the need to dispose of real RFID transponders. The generation of RFID events by the emulator can be done according to the user's specifications or in a random manner.
- **Management console:** It is a web-based GUI that facilitates the interaction between the administrator and the middleware. Via this interface the administrator configures the RFID readers (real or logical), controls the network status and manages the back-end applications.
- **Tag Viewer:** Visualizes RFID events generated through a program using the ALE API. The module's source code is available as a reference to RFID system developers. Application users specify the events to be visualized, such as adding or deleting events, as well as the current RFID tag data. They also have the right to limit a range of tag IDs to be displayed.
- **Device adaptors:** This component allows the middleware to cover a variety of RFID devices from different manufacturers, such as barcode scanners or RFID readers, which ensures the heterogeneity option. Device adaptors ensure the interaction between the hardware components and the ALE engine core.

3.1.19 IoT middleware for intelligent industrial parks

This middleware architecture is mainly based on these 3 layers (Zhang et al., 2020):

1) Service-scheduling layer.
2) Device driver layer.
3) Application business layer.

It provides an application abstraction based on a unified data interface. And for scheduling service, this middleware concentrates on the management of the entire middleware platform, the planning and management of application services, the planning and management of devices and the log system.

3.1.20 UIR middleware

This RFID middleware is organized as a three-tier architecture, with a physical RFID infrastructure combined with WSN, RFID middleware (UIR-Middleware), and back-end applications. To ensure hardware abstraction, this architecture proposes the HAL layer which in turn consists of the following 3 sub-layers AAL, EDML, and HAL the abbreviations for Application Abstraction Layer, Event and Data Management Layer, and Hardware Abstraction Layer respectively (Rouchdi et al., 2018c).

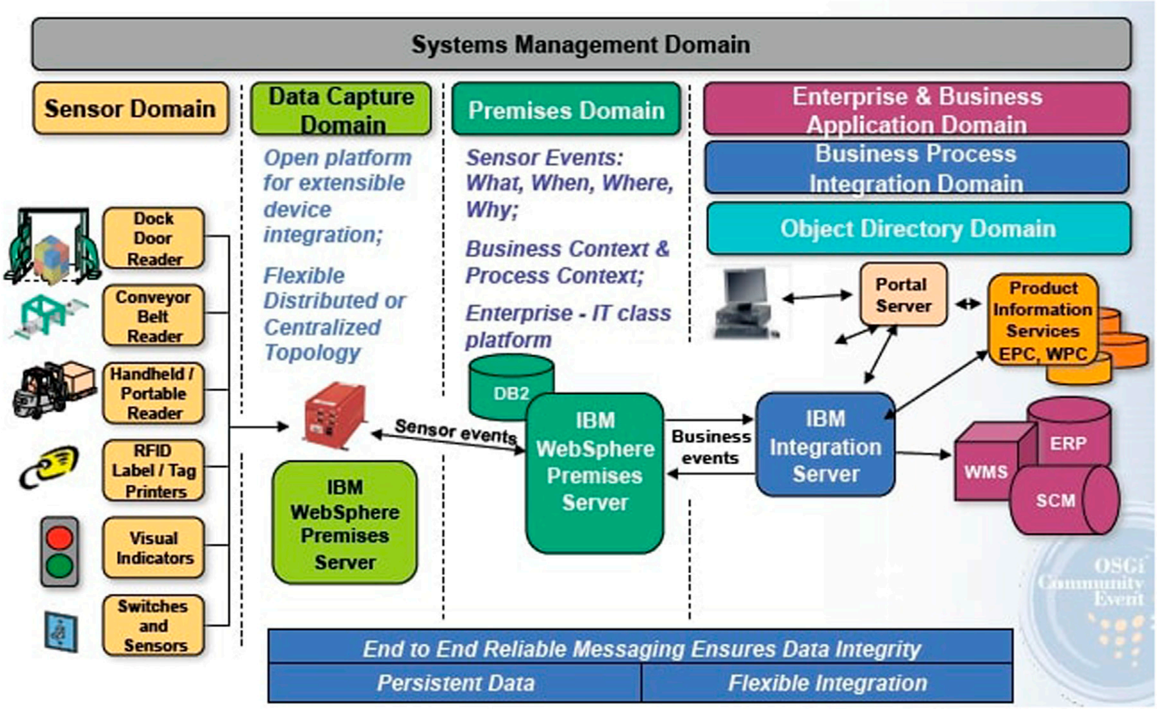

Systems Management Domain
Sensor Domain
Data Capture Domain
Premises Domain
Enterprise & Business Application Domain
Business Process Integration Domain
Object Directory Domain
Open platform for extensible device integration;
Flexible Distributed or Centralized Topology
Sensor Events: What, When, Where, Why;
Business Context & Process Context;
Enterprise - IT class platform
Dock Door Reader
Conveyor Belt Reader
Handheld / Portable Reader
RFID Label / Tag Printers
Visual Indicators
Switches and Sensors
Sensor events
DB2
IBM WebSphere Premises Server
IBM WebSphere Premises Server
Business events
IBM Integration Server
Portal Server
Product Information Services EPC, WPC
ERP
WMS
SCM
OSGi Community Event
End to End Reliable Messaging Ensures Data Integrity
Persistent Data
Flexible Integration

FIGURE 17
WebSphere RFID architecture (IBM Corporation, 2009).

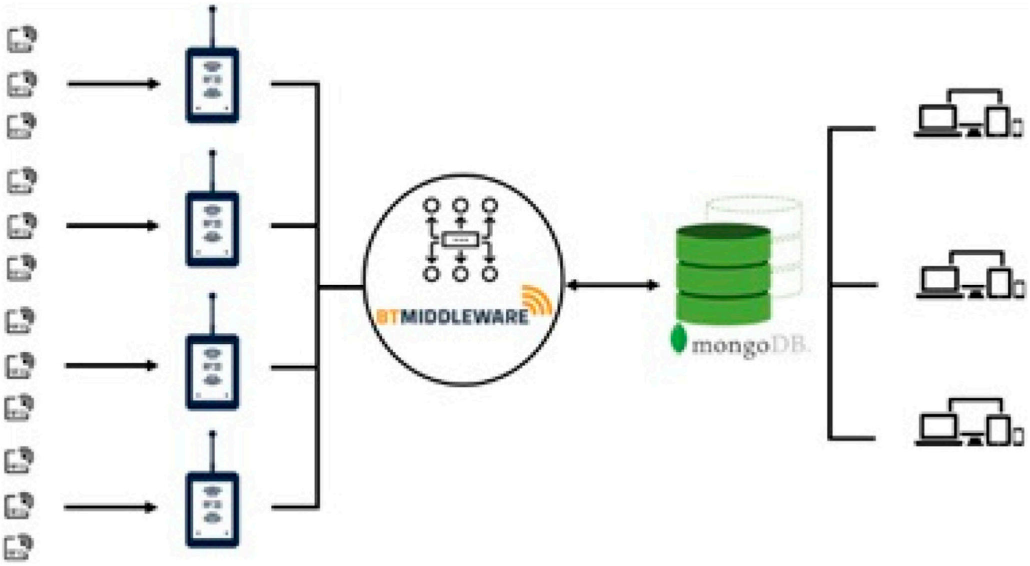


FIGURE 18
BTMiddleware architecture (Haibi et al., 2018).

3.2 Analysis

3.2.1 Classification of RFID middleware

This section classifies the different RFID middleware according to various functionalities. First, by following the study (Ahmed et al., 2011), the middleware solutions studied are grouped according to their design approaches, as follows:

- **Generic RFID Middleware:** This type of middleware is dedicated to processing large amounts of data and focuses

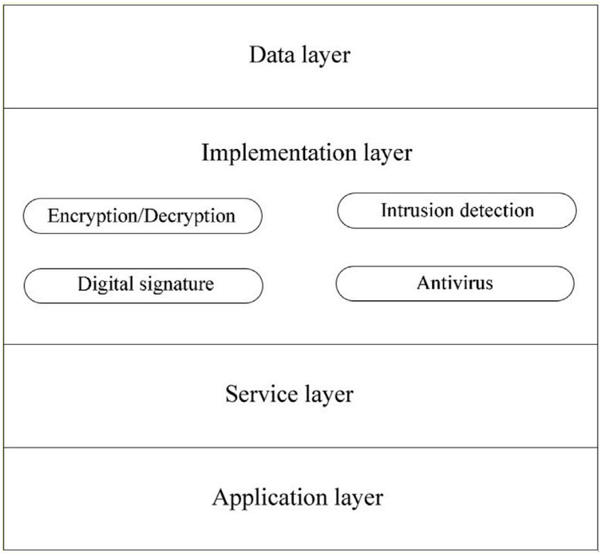


FIGURE 19
Architecture of RFID Middleware model based on ECDSA (Qiyue and Ping, 2017).

on scalability, data organization, large-scale data management, and high throughput.

- **Event-Based RFID Middleware:** This kind of middleware processes data in events form. This category can have better control of RFID data.
- **Dynamic Resource Management RFID Middleware:** Labelled objects network with RFID tags can generate a huge data flow, which can overload the system for a particular period. This problem requires the middleware to have specialized algorithms to handle this load. This class of middleware solutions is characterized by the ability to handle this type of unexpected load.
- **Special Purpose RFID Middleware:** In some applications, we speak of indoor RFID installations which require the protection of the human user's privacy, this middleware type includes architectures characterized by considering confidentiality as a major concern.
- **Commercial RFID Middleware:** This category contains middleware solutions offered by companies, which makes it difficult to assess them due to lack of information.

Table 1 groups the 20 architectures studied in this manuscript according to the classification presented in (He et al., 2013). By applying the taxonomy presented in (Ahmed et al., 2011) to all the examined studies, we observe that current Middleware architectures consider the constraint of regular data collection and processing with different mechanisms, but none of them takes into account users' privacy issue, which requires the proposal and implementation of a new well-secured middleware architecture to be perfect in terms of RFID data security. We can conclude that all the presented architectures are oriented to the industrial domains concerns. This result indicates that researchers must focus on proposing robust security architectures to protect human data. Among the potential proposals is the coupling of encryption solutions with access rights management solutions at the same time to increase data security.

About twenty research works have been studied, 17 middleware architectures between them are implemented, and two studies present architectures without implementation. This gives 85% of middlewares are implemented (Figure 21A). This result shows that the majority of RFID middleware goes beyond the modelling phase through to the implementation phase. Since the RFID middleware market is dominated by three trends (Abad et al., 2012a; Abad et al., 2012b), giants' software vendors, Specialized companies, Research Results (Takes origin in research centres and universities), the first two types are supposed to implement the architectures for commercial purposes. A set of middleware in the third category also goes into the implementation phase because most of these research studies are done as part of research projects requiring the implementation phase. This is why we find a majority of middleware architectures with implementation.

Figure 21B shows that the majority of middleware architectures examined are not applied to a well-defined application domain, and

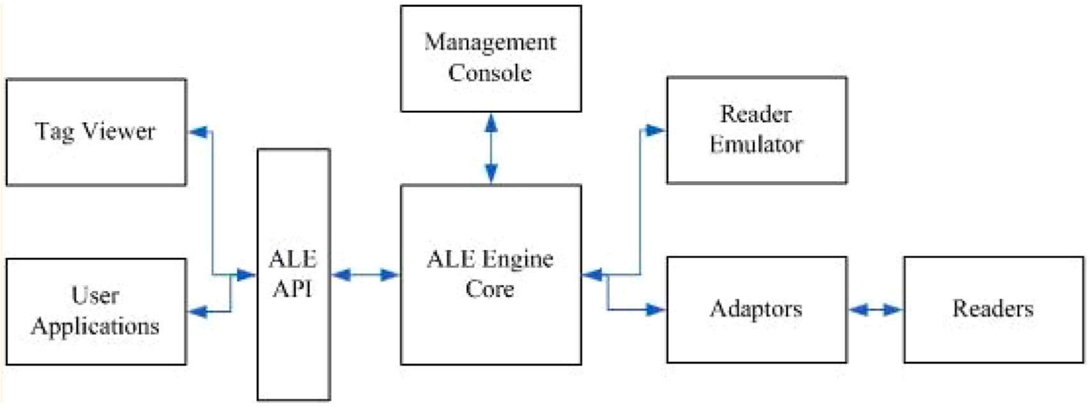


FIGURE 20
CUHK RFID system 1.0 architecture (Mak et al., 2007).

TABLE 1 Classification of RFID middleware systems.

Classification	Properties	Middlewares
Generic RFID Middleware	Scalability, Data management, Item Tracking applications	Fosstrak, WinRFID, RF2ID, AspireRFID, FlexRFID, LIT Middleware, TagCentric, SafeRFID, MedRFID, ECDSA Middleware, DEPCAS, RFID middleware with database, CUHK RFID
Event Based RFID Middleware	Treats RFID data as Complex events	IoT middleware for intelligent industrial parks, UIR Middleware, BTMiddleware, Lightweight-ALE-Based Embedded RFID Middleware
Dynamic Resource Management RFID Middleware	Uses mechanism to Handle unexpected large amount of data	RF2ID, BTMiddleware
Special Purpose RFID Middleware	Consider item location applications	-
Commercial RFID Middleware	Developed by various Commercial vendors	IBM Webspher, Sun java system RFID.

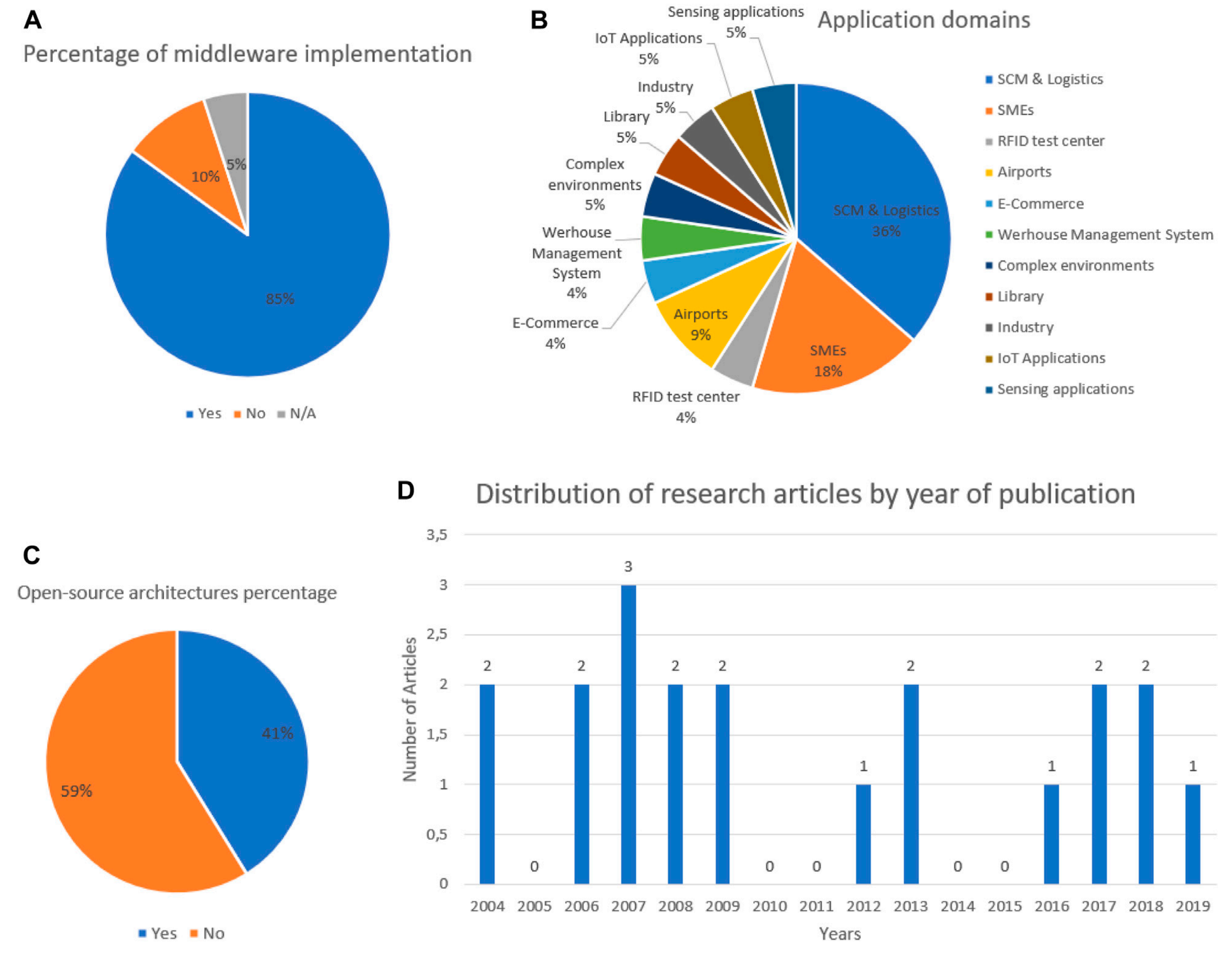


FIGURE 21
(A) Percentage of middleware implementation. **(B)** Middleware application areas. **(C)** Open-source architectures percentage **(D)** Distribution of research articles by year of publication.

it also shows which areas are attracting the most attention from researchers and companies, namely (SCM & Logistics) and SMEs, with respectively 36% 18% of all evaluated studies, followed by the aeronautical sector. In recent years, we also note that new application areas have seen the integration of RFID technology, namely, IoT and sensing applications. The first RFID middleware appeared, used mainly in cases where data needs to be shared in more than one location at a time and a variety of business applications requiring access to data collected by RFID readers (Al-Jaroodi et al., 2009), such as logistics and SCM fields. This

TABLE 2 Open issues in existing middleware architectures.

Challenges	Details
Reliability	As mentioned earlier, there is a growing trend towards the adoption of this technology, particularly in critical areas where reliability is vital. Reliability means that no disturbance is tolerated in the RFID network. To address reliability issues, fault-tolerant algorithms need to be included in middleware architectures, but according to the literature, this aspect is not yet mature. It should be noted that middleware based on such algorithms will provide functionalities such as self-recovery and self-reaction, guaranteeing continuity of services offered to users
Autonomy	The infrastructure of the distributed RFID network is typically composed of a high number of RFID devices, and the task of controlling this physical part is often difficult. While existing architectures do not offer autonomy and self-adaptability, future middleware architectures will have to integrate this functionality so that they are aware of any possible alteration of the RFID network infrastructure conditions, thus facilitating the monitoring and management of RFID devices
Interoperability	Backend applications do not have the ability to interpret the data without going through the formatting of raw RFID data Haibi et al., (2021b). The literature on current middleware shows that the majority of architectures rely on XML for sharing and exchanging RFID data Breje et al., (2018), but this format is limited because it is generally more verbose than its alternatives (i.e., it contains more characters), which is why it is more explicit for a human Vanura and Kriz, (2018). Among the recommended alternatives is the JSON format, which is easier to share and retrieve data, moreover it is lighter than XML and saves resources. Future middleware proposals will need to format and present RFID data while respecting the requirements of business applications to provide improved semantic interoperability
Security	There is no doubt that data security is important, in general, but few architectures have well-defined security policies in place to deal with cybercriminals, or even to stay ahead of them, furthermore the relationship between the module or the security layer is not clear and precise in these architectures. Therefore, future middleware architectures will have to invest more in this crucial functionality, following the scientific advances in computer security
Storage	The current literature shows that existing architectures are based on RDBMS for RFID data management, but with Big Data, RDBMS have shown their limits very quickly, on the one hand to the high volume of data, and on the other hand to the diversity of data types Baruffa et al., (2020). This makes it increasingly difficult to manage RFID data that arrives in increasingly varied forms and that is produced more and more rapidly in the case of novel RFID applications. It is therefore to meet these new requirements of scalability, availability and storage distribution in future middleware proposals that the so-called "NoSQL" DBMS Baruffa et al., (2019) which is able to manage a large volume of structured, semi-structured and unstructured data and which offers fast performance and horizontal scalability Ganapathi and Shanmugapriya, (2019)
Heterogeneity	One of the problems raised in the literature is the hardware support in heterogeneous physical RFID infrastructures, which is why mechanisms and protocols (such as LLRP.) need to be implemented and integrated in middleware architectures in order to guarantee a better hardware abstraction
Real-time RFID analysis	Collecting, processing, exploiting, visualising, extracting information from big RFID data in real time to make decisions, is detected among the challenges of RFID applications Chung and Berhe, (2021). And from the literature review it was found that none of the existing architectures address this aspect. This is why future middleware architectures will have to integrate real-time stream processing tools, which allow the immediate analysis of data sent in a continuous flow. This tool will allow to detect very quickly patterns, correlations between the incoming RFID data flow and historical data

type of areas requires multiple readers to spread across factories, warehouses, and distribution centres. The integration of RFID technology into SCM & Logistics systems has reduced waste and enhanced visibility at multiple steps of the supply chain (Amin, 2013; Sheng et al., 2008) through its capacity to automatically track each article across the chain securely and in real-time. RFID also contributes to reducing data entry errors, efficiently manages inventory, and optimizes the flow of goods (Ropraz, 2008b). This mainly justifies why a large number of SCM & Logistics companies are investing in RFID.

Among examined works in our study, we notice that the majority of RFID Middleware implementations are not Open-Source with a percentage of 59% (Figure 21C). This reflects that a significant number of middleware implementations can only be modified by their original authors, limiting the possibility that this middleware will be improved and developed as more people are unable to work for its improvement.

Figure 21D illustrates the trend in publications over time. As we can see, it is clear that the serious effort to address this research field only began in 2007. We also note a decrease in the number of works published between 2013 and 2016, followed by an increase between 2016 and 2019, reflecting those earlier studies till 2016 have shortcomings and no longer keeps pace with technological development, which has forced researchers to present and propose new improvements in middleware architectures. Since there is a huge amount of research work has been paid to RFID hardware components, this result shows that there is a gap between the number of research work contributing to RFID tags' hardware design & performance, and those dealing with RFID data acquisition, processing, and management solutions. Given that RFID is moving into critical areas characterized by massive amounts data, where the concept of real time is very important. It should be highlighted that the traditional data platforms and techniques are less effective in these kinds of fields (Haibi et al., 2022a; Oussous et al., 2017).

3.2.2 Challenges

Based on the study of RFID technology and in particular the Middleware components, it is noted that this field is very active, and researchers continue to contribute to it. The main limitations and issues identified in the current architectures are listed in Table 2.

4 Proposed middleware architecture (UMIUR Middleware)

4.1 Preliminaries

In this work, we have explored the gaps in RFID middleware architectures for new applications such as IoT applications. By analysing different constraints associated with this kind of applications, a large part of them is related to the processing and storage of large amounts of RFID data in real-time, as well as to security and interoperability. This wide variety of applications and their specific constraints has prompted us to propose a new architecture that, based on a set of technologies, fills this set of gaps by ensuring.

4.1.1 Interoperability

The volume of RFID data is constantly growing in modern applications. This (big data) reality means that data transformation is more important than ever for back-end applications, as they do not have the capacity to understand and process raw RFID data without it being transformed so that they can make the right decisions. From the existing literature on RFID middleware architectures, most middleware architectures use XML as an exchange format. Nevertheless, this can be understood given that, by far, XML has been the single option to share and transfer of data for a long time. Although this exchange format has its benefits, it is not adequate for big RFID data sets, is significantly more verbose, relatively more difficult to read and interpret, and has a much more complex syntax that does not directly correspond with data structures in today's coding languages (Breje et al., 2018), it is becoming unsuitable for use in modern systems. Over the last decade, the debate between XML *versus* JSON has been one of the most prominent topics in developer circles. However, based on the literature comparing these two formats, we find that JSON has several advantages over XML (Vanura and Kriz, 2018). For instance, data processing with JSON is easier than with XML, and JSON is independent of the languages that use it and easy for machines to parse, allowing JSON to rapidly replace XML in last few years (Lanthaler and Gütl, 2012). This set of advantages allowed us to consider adding the functionality of formatting raw RFID data in JSON format as well.

4.1.2 Big data storage

The term Big Data is used when traditional data management tools are not able to store or process such massive data sets (Khemiri et al., 2022), and this is the case with RFID data in modern applications. Big Data is often defined based on three concepts called the 3 Vs. (Kouanou et al., 2018; Erevelles et al., 2016):

- **Volume:** the data volume is already large and is constantly increasing. This can generate storage and analysis difficulties -which is one of the characteristics of RFID data-. For example, the use of RFID technology as a traceability system by a medium-sized retail chain will produce 300 million RFID scans a day (Boontrai et al., 2009) and getting the relevant information out of this vast data stream will not be easy.
- **Variety:** the data come from different sources, disciplines, formats. This can lead to difficulties in understanding and integration; and this is the case for RFID applications, as RFID data often comes from heterogeneous data sources.
- **Velocity:** large volumes of data are collected very frequently (this can be in real-time). This can generate difficulties in data processing.

Thus, the development of autonomous objects equipped with RFID tags produces large amounts of data that challenge many traditional approaches in Information Systems. The latter have to deal with huge amounts of disparate RFID data, i.e., highly variable, structured or unstructured, sometimes imperfect (Moniruzzaman and Hossain, 2013). A set of data characterized by these notions must therefore be managed using dedicated Big Data tools. Today only the NoSQL systems can manage the huge volumes of big data which is generated very fast (Jose and Abraham, 2017a). The term NoSQL covers database systems that are not relational. These databases offer an alternative to traditional databases for processing large volumes of data. They do not present the properties of a relational database [ACID properties (atomicity, consistency, isolation and durability)] and are not, in general, interrogable with SQL language. They have been designed to manage large volumes of data (for this they have the "horizontal scalability" properties, because the data can be distributed over several databases) and unstructured data (their data model is more flexible than that of RDBMSs (Relational Database Management Systems).

NoSQL databases can be classified into four categories: key-value databases (Dynamo, Reddis, Voldemort, etc.), document-type databases (MongoDB, CouchDb, etc.), column-type databases (Hbase, Cassandra, BigTable …) and graph type databases (Neo4J, InfoGrid …) (Xiang et al., 2016).

For our architecture, we chose the MongoDB database (belongs to the document-oriented model databases) for its multiple advantages and functionalities. MongoDB is a system that stores data in BJSON (binary JSON) (Zhang et al., 2014), it is optimized for loading from JSON files (which makes it perfect for our Middleware that uses this type of data formatting), that's why we have coupled for our Middleware the JSON format for data presentation and the MongoDB database for storing large amounts of data. Table 3 shows some important features of MongoDB (Jose and Abraham, 2017a).

4.1.3 Real time RFID data processing

In a general context, modern RFID applications require continuous processing of streaming data detected at different locations, at different times and at different rates, in order to achieve added value in their business and service areas. Real-time analysis is now a central concern for companies (Khaddar et al., 2011). It is an essential practice to significantly increase turnover but also to remain competitive. The science that examines raw data in real-time with the aim of drawing conclusions–without delay-is called real-time analysis. Analytical tools are adopted to empower organizations and businesses to make better decisions, and data analysis is considered among the most essential tasks behind success in several fields of services and business. Some examples of these areas include the RFID technology application areas where they rely on timely and rapid analysis based on available data to make quality decisions. Recently, CEP has proven to work as a basic tool for processing RFID data in real time to identify all circumstances on demand and act instantly (Aftab et al., 2018), as it was designed specifically to solve issues related to real-time event processing in distributed systems (Amaral et al., 2011). The CEP is characterized by scalability, efficiency, speed, robustness, and heterogeneity. It inputs an endless and infinite stream of events from different sources to facilitate real-time data management and detects a massive number of events with low latency. Furthermore, CEP can be used for a wide range of applications (Elkhoukhi et al., 2022),

TABLE 3 MongoDB features.

Feature	Explanation
Rich Query language	Among the large number of features offered by RDBMS available in MongoDB: easy aggregation, dynamic queries, secondary indexes, sorting and rich updates. It also offers adaptability and scalability Jose and Abraham, (2017a)
Flexibility	MongoDB stores data in document format using JSON. It is schema less and maps to local programming language types Jose and Abraham, (2017a)
Sharding	This allows linear scaling of the cluster. It is made possible by adding more machines. Because of sharding, the efficiency can be maintained even if there is an unexpected increment of load in the web
High availability	MongoDB supports the creation of replicas. This is the grouping of servers that maintain the same data set.
Ease of Use	As mentioned before, MongoDB is a freely available document database which is easy to install. In addition, its use, maintenance, and configuration are also very simple Jose and Abraham, (2017a)
High Performance	It allows faster processing of queries; this is enabled by supporting embedded documents and indexing. It promotes speed by reducing I/O actions on database systems Jose and Abraham, (2017a)
Support for Multiple Storage Engines	It uses the Wired Tiger storage engine which has several storage engines Jose and Abraham, (2017a). It also supports the Pluggable Storage Engine API which allows a third party to develop a storage engine for MongoDB.

from simple surveillance to very complex applications such as fraud detection and algorithmic trading (Tawsif et al., 2018). Thus, CEP can handle massive unpredictable RFID data, which is generated by RFID networks from multiple sources (Aftab et al., 2018).

4.1.4 Security

With the rapid developments in the RFID technology field, there has been a growing trend toward the adoption of this technology and even in critical sectors. This raised the important security issue of how to control and prevent unauthorized access to RFID data.

- RBAC

One approach to protect the privacy of stored data is to use access controls. Many access control models have been proposed over the years in the literature (Gouglidis et al., 2012). In this context, there are two well-known models, RBAC and ABAC, and according to (Aftab et al., 2018) the RBAC is most trustworthy instead of ABAC. Role-based access control (RBAC) is a well-known access control model that can help simplify security management, especially in large-scale systems. Since its first formalisation in the 1990s, RBAC has been widely used in many systems to provide users with flexible controls over access to their data. The RBAC model was extended and updated in 1996 (Zhou et al., 2015), and the RBAC standard was proposed in 2000 (Gouglidis et al., 2012). This model makes it possible to establish, within a company's information system, an efficient access control to the services and applications and of this IS. It is mainly based on the definition of roles to be attributed resources and users. RBAC has become the most widely used management model because it can be easily applied to different structures. Each role is a simple collection of permissions and users are granted permissions only through the roles to which they are assigned.

When defining an RBAC model, the following conventions are useful.

- U: users
- R: roles
- P: permissions
- S: sessions
- OPS: Operations
- OBS: Objects
- UA: User Assignment
- PA: Permission assignment

The role is the core of the RBAC model and is seen as an intermediate entity between users and permissions as it groups a set of privileges and then assigns them to users according to their role (Dana and Sèdes, 2009/3; Al Kukhun and Sèdes, 2012).

As shown in Figure 22A, role assignment in the RBAC model follows a mutual relationship where a User (person, computer process, machine, etc.) can play multiple roles in a single session and a role can be assigned to multiple users.

$$UA \subseteq U \, x \, R$$

Assigning a role will grant multiple permissions to the user and a permission can be assigned to multiple roles.

$$PA \subseteq R \, x \, PRMS$$

The nature of a permission describes the type of operations (OPS) (e.g., read, write, update, etc.) allowed on OBS objects (data resources: documents, computer processes, machines, etc.) (Al Kukhun and Sèdes, 2012). The relationship between these objects and the assigned operations is also mutual; an operation can be allowed on several objects and an object can be assigned different permissions (Breje et al., 2018).

$$PRM \subseteq OPS \, x \, OBS$$

- **Blowfish**

As mentioned before, in the current era of RFID applications, with terabytes of data generated daily, securing information is a challenge. Cryptography is a process of making information unintelligible to an unauthorised person, thus providing confidentiality to genuine users which can make the RFID data environment more secure (Patil et al., 2016). In general, there are two main families of cryptographic encryption algorithms:

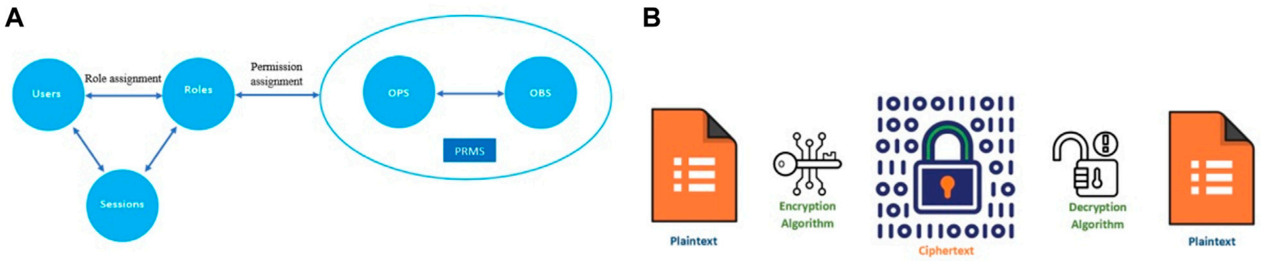


FIGURE 22
(A) RBAC mechanism (Tawsif et al., 2018). **(B)** Blowfish algorithm process.

Symmetric or private key algorithms and Asymmetric or public key algorithms (Latif, 2020).

4.1.4.1 Symmetric key algorithms

A symmetric key algorithm consists of using the same key (a secret key) to encrypt plaintext and decrypt ciphertext data.

Principle:

- Symmetric encryption involves applying an operation (algorithm) to the data to be encrypted using the private key, in order to make it unintelligible.
- Exchange and the private key between entities, so that it can be used in the decryption process.

4.1.4.2 Asymmetric encryption (or public key encryption)

Consists in using a public key for encryption and a private key for decryption (Gerla and Reiher, 2015).

Principle:

- The users choose a random key that only they know (this is the private key) (Kumar et al., 2020).
- They each automatically deduce an algorithm (this is the public key) (Kumar et al., 2020).
- The users exchange this public key through an unsecured channel (Kumar et al., 2020).

The task of choosing the best algorithm to integrate in our architecture was not simple due to the number of existing encryption algorithms. In order to choose we relied on (Nazeh Abdul Wahid et al., 2018), which implemented and analysed in detail the costs and performance of the commonly used cryptographic algorithms DES, 3DES, AES, RSA and blowfish (Patil et al., 2016). (Lanthaler and Gütl, 2012) showed in an overall performance analysis in contrast to theoretical comparisons. The paper states that each of the encryption techniques has its own strengths and weaknesses, which showed that of all the cryptographic algorithms, Blowfish algorithm is the best in terms of execution time, memory usage, throughput, power consumption, security (Suresh and Neema, 2016) and thus blowfish algorithm is well suited for RFID applications. This algorithm was designed by Bruce Schneier in 1993 as an alternative to existing algorithms. It can take a key length ranging from 32 bits to 448 bits. Since its conception, it has been extensively analysed and is now considered to be a robust encryption algorithm. It is also present in many solutions (OpenVPN). Among its benefits, it consumes the least time among DES, 3DES, AES and RSA and in Blowfish, the memory required for implementation is the smallest (Lanthaler and Gütl, 2012; Patil et al., 2016). Therefore, since time and memory are major factors four our case, Blowfish is the best option. Figure 22B outlines the Blowfish algorithm process.

4.2 UMIUR middleware description

To highlight the contribution of this article, we point out that, firstly, it suggests a new approach which fills a set of known gaps in existing RFID middleware architectures, The proposed design of our architecture represents a framework suitable for a variety of applications and is based on the already developed RFID standards. Secondly, it declares the coupling of the RBAC model and the Blowfish encryption algorithm as a tool regulating the access and display of decrypted data between the AAL and DPL layers, solving the security problem occurring in previous RFID middleware. And, to tackle the problem of processing a very large number of extremely complex queries, our architecture is based on the paradigm of real-time event processing based on the CEP. Moreover, in order to improve interoperability, the proposed middleware provides RFID data formatting in JSON which has become a popular alternative to XML for its various advantages. Finally, the proposed architecture is also based on NoSQL technology to ensure the storage of large volumes of RFID data. Figure 23 shows the different layers and sublayers of our Middleware architecture.

4.2.1 Application layer

This layer is responsible for providing the various services and functions to the end users of the IS. This layer can be composed of one or many backend applications, such as WMS, ERP, user applications (website, mobile application).

4.2.2 Hardware abstraction layer (HAL)

HAL is in charge of interfacing with the physical RFID infrastructure. It represents the abstraction of the physical parts of the system and supports different devices (read-only, read/write, passive, active tags.) working with multiple frequency bands (HF, LF or UHF). The device abstraction is done in a different way in RF^{2}ID,

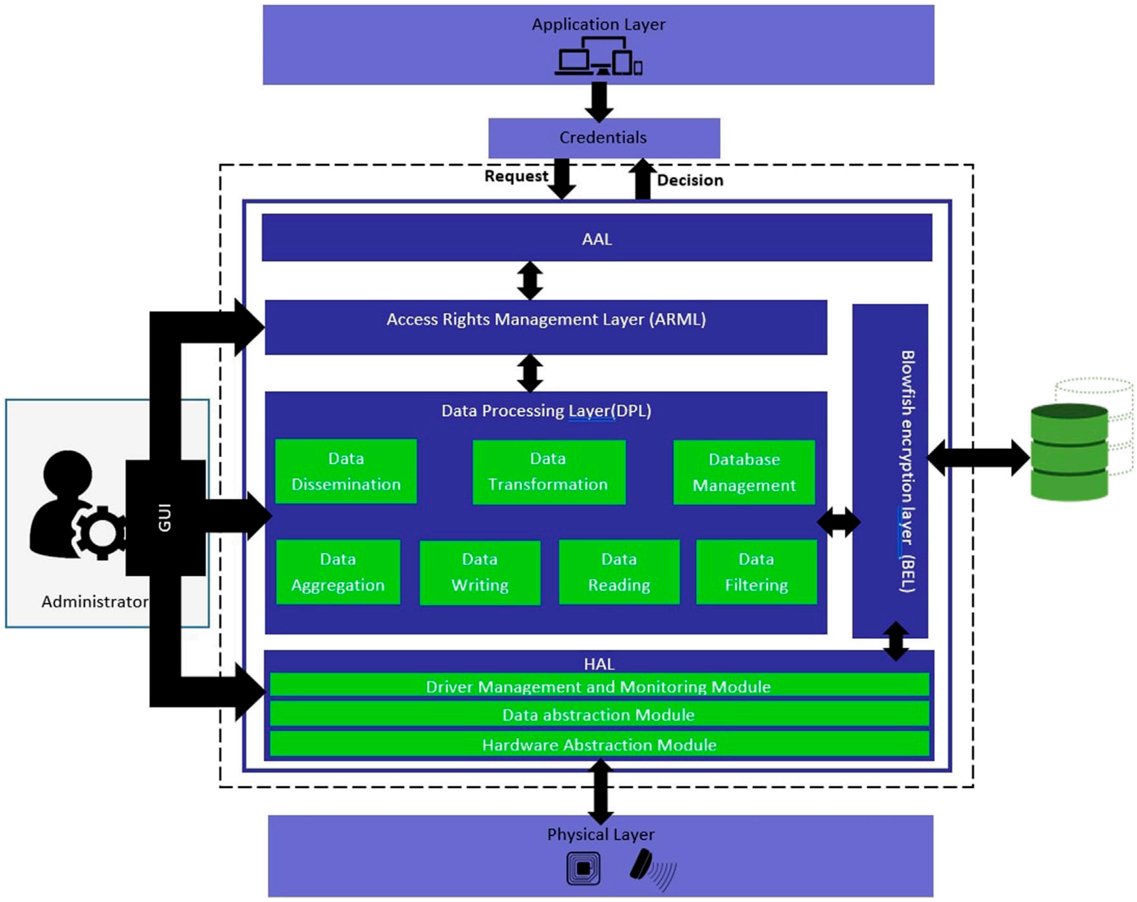


FIGURE 23
UMIUR Middleware architecture.

it is based on the notion of virtual readers so that each virtual reader manages a set of physical readers in the same neighbourhood. The diversity of RFID devices constituting the Physical Layer in our case makes the RFID network heterogeneous, and this requires the integration of an abstraction layer of RFID devices into the middleware that interacts with the physical RFID network. Such an approach allows companies to adopt RFID solutions without managing the low-level programming. Furthermore, this approach improves the flexibility of the middleware in terms of hardware selection, as it provides a unified communication gateway to the physical RFID infrastructure, i.e., heterogeneous RFID devices, regardless of the employed frequency range (HF, UHF or LF), or the adopted communication protocol (e.g. LLRP or other), the manufacturer or the communication interface. So, in our case, the Hardware Abstraction Layer will ensure the interfacing with the hardware independently of its characteristics. The HAL is structured in three different sub-layers.

- Hardware Abstraction Module

As mentioned before, there are many readers of different brands and of course with different characteristics and protocols. Thanks to the functionalities offered by this module, our middleware architecture will be able to communicate with a heterogeneous RFID physical RFID infrastructure no matter which communication protocol is employed (USB, Ethernet or RS232), the brand, or the frequency range, because this module acts as a unified interface of communication with the physical RFID infrastructure. In addition, via HAM, all basic RFID reader parameters are under the control of the system administrator, this set of parameters includes Stop/Start, Activation/Deactivation, and Read/Write.

- Device Management and Monitoring Module

This module offers the user the possibility to select from a broad range of RFID devices depending on their requirements and allows him to replace equipment directly without influencing the insurance software solution. Thanks to DMMM the RFID reader driver libraries are called dynamically which makes the architecture more flexible.

- Data abstraction Module

Still with the aim of guaranteeing a unified interfacing with heterogeneous RFID devices, this module ensures that the RFID data will be communicated with the RTDP layer in a common format regardless of the characteristics of the hardware RFID components, in addition DAM handles middleware requests such as ACKs and inventory results.

4.2.3 Data Processing layer

Equivalent to the "data management layer" for WinRFID, it fills a set of inconsistencies (read errors, unread tag, tag read several times), defines management rules for checking the data read, aggregation and redundant data filtering.

Gathered at the layer named Middleware for the Fosstrak middleware, which is in charge of filtering, aggregating and distributing data, this layer also allows applications to define a subscription in which each application defines the readers to be used, the type of data it is interested in, their formats, etc.

For the AspireRFID middleware this layer is called the Hardware Abstraction Layer, it provides a hardware abstraction to unify the way the middleware interacts with readers from different manufacturers, and which use different communication protocols.

It is the "RFID Event Manager" layer in the Sun Java System middleware, its main purpose is to interface with readers, collect EPC events, filter redundant data and feed important EPC events to the RFID information server or other ERP application. This layer is based on Jini technology.

The Data Processing Layer (DPL) layer for our case, which lies between AAL and HAL, is the core of the Middleware. Because any kind of middleware service or functionality is provided through it. The lower HAL layer communicates with the RTPL by providing it with the raw collected RFID stream. Subsequently the different modules of the RTPL layer will be responsible of storing, formatting, filtering, and the exchange of formatted RFID data with the appropriate back-end applications. The CEP module is required to apply complex queries to several data streams simultaneously to detect specified conditions (events), thus triggering appropriate actions in real time. This enables the middleware to track, analyse and process large RFID data streams when an event is detected. This layer consists of.

- Data Dissemination

Normally, client applications have to exploit the RFID data. DD module is implemented in order to transmit this data to any interested application. When a back-end application requests RFID information and if it has the required authorisations, this module ensures the transfer of the data it is interested in.

- Data Aggregation

The DA component is tasked with grouping RFID data according to specific criteria so that each back-end application receives only the data relevant to it, and to reduce the increased granularity. For instance, "grouping data coming from the same RFID reader".

- Database Management

DM is responsible for all database organization e.g., deletion, storing and sharing data. As discussed in Section 5.1, among the things that characterize new areas of application of RFID technology is the flow of RFID data, which is really very important, and in order to ensure the storage of this huge volume of RFID data our approach relies on the SQL solution for big data storage, and more specifically the MongoDB database.

- Data Reading

For our approach the Data Reading component represents the same thing for most proposed middleware architectures, it is in charge of collecting data from the RFID tags.

- Data Transformation

The raw data has no meaning for the applications, that's why "Data transformation" in our architecture transforms it into business events. For instance, in the scenario of deploying RFID technology in the baggage handling process, a bag's location can be expressed as: the bag was in this location at this hour. According to the literature, it has been observed that middleware architectures are mainly based on the representation of data in the XML form, certainly, this format offers a number of significant benefits in terms of data presentation and exchange, but it has a set of limitations which has led us to think of other data presentation formats to ensure better interoperability. As mentioned in (Lanthaler and Gütl, 2012), today the lighter approach of JSON has proven to be popular and is rapidly replacing XML, which led us to consider including the option to format raw RFID data in JSON as well, according to the need of the IS applications.

- Data Filter

The RFID data stream generated is always huge in the case of a network of mobile RFID objects. The task of extracting useful data from this vast selection requires data filtering mechanisms. So, it is the role of the module DF to eliminate the undesired data, especially the duplicates.

- Data Writing

As mentioned in the "RFID System Components" section, we find read/write tags, which enable the insertion of information on the RFID tag chips. This proposed middleware architecture provides this functionality by calling the DW module.

- CEP Module

As mentioned in the introduction, RFID enhances the presence of data that evolves regularly over time and needs to be processed continuously, due to the massive presence of RFID tags. Such exponential data growth causes the 3V in Big Data nature with extreme data streaming speed. This poses significant challenges to overcome the problems of Big Data analysis and real-time decision-making issues. To this end, it is realistic to consider the representation of data as streams and to use processing models corresponding to this information. To tackle the problem, our architecture is based on the paradigm of real-time complex event processing via the CEP Module which solves the Big Data velocity

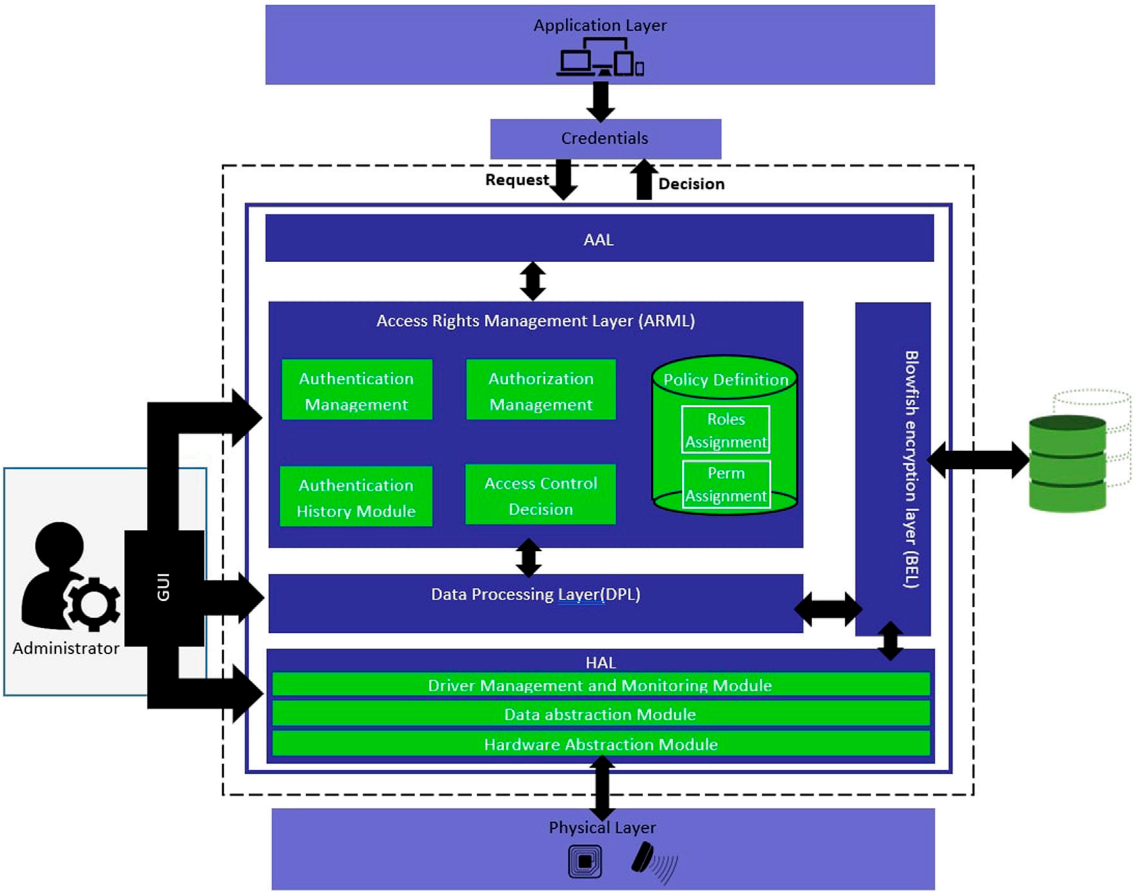

Application Layer
Credentials
Request
Decision
AAL
Access Rights Management Layer (ARML)
Authentication Management
Authorization Management
Policy Definition
Roles Assignment
Perm Assignment
Authentication History Module
Access Control Decision
Blowfish encryption layer (BEL)
Administrator
GUI
Data Processing Layer(DPL)
HAL
Driver Management and Monitoring Module
Data abstraction Module
Hardware Abstraction Module
Physical Layer

FIGURE 24
ARML Layer architecture.

problem. CEP Module will be responsible for tracking RFID data streams from multiple sources to analyse events in real time in order to respond as quickly as possible, providing organisations with greater situational awareness and business agility.

4.2.4 Application abstraction layer

Equivalent to the data presentation layer in the WINRFD architecture, which provides data and functionality to users. In order to hide the complexity of the RFID system, this module aims to provide a common interface for the application layer to give them the option to access the different functionalities of the middleware if they are granted the necessary authorisations.

4.2.5 Access rights management layer (ARML)

One of the aims of this work is to secure RFID data, as security is the major problem facing every user. This prompted us to include the Access Rights Management Layer (ARML) in our proposed middleware design. Figure 24 depicts the ARML architecture, which controls all connection requests emerging from the application layer, ensuring that no requests from applications without the necessary authorisations from ARML will be processed. This layer based on the RBAC module limits the system against unauthorized access, as there are a number of restrictions to access RFID data with each user. ARML will therefore help to regularize access to RFID data so that only authorized users to access, will have the right to use the decrypted RFID data. Its operating principle is firstly based on authentication to certify the identity of the user, and then, depending on the authorizations granted to this user, the Access Control Decision module (ACD) shares the decision. In the case that it has appropriate authorisation, ACD will identify the resources he can access and the operations he can perform; if the user is not authorized, their requests will be denied.

- Authentication Management

Each time the Middleware receives an authentication request, it delegates it to the Authentication Management Module, which is in charge of analysing the credentials provided by the user during the Login phase and then saves the authentication results to the Authentication History module.

- Authorization Management

After communication with the Decision Module, it determines the permissions granted to an authenticated user.

- AC Decision Module

This is the element that evaluates incoming requests ("who" wants to do "what" on "what") against registered rules ("who" can do "what" on "what"), the Allow/Deny decision will depend on the roles and permissions assigned to users.

- Policy Definition

Policy Definition contains statements that collect roles and permissions to express what can and cannot be accessed. Policies in ARML can grant or deny actions on the middleware. In this layer, roles are used to associate users with permissions on resources. Users are assigned roles, and permissions are assigned to roles rather than to individual users; only users who have been granted role membership can access the permissions associated with the roles and can therefore access resources. The ARML consists of:

- Permissions Assignment: The administrator is the only actor who can assign permissions to roles. Permissions represent statements describing the behaviour, i.e., the set of operations that can be performed on the middleware.
- Roles Assignment: The administrator is the only actor who can assign roles to users, a role contains one or more permissions, roles are assigned to users and, therefore, users will have the right to access permissions assigned to roles.

- Authentication History Module

This module records historical information about the use of the middleware by storing all successful and failed authentications.

4.2.6 Blowfish encryption layer (BEL)

This layer is responsible for data encryption, upon arrival of RFID events, BEL encrypts all RFID tag data collected by the reader based on the Blowfish encryption algorithm.

4.3 UMIURMiddleware implementation

4.3.1 Architecture

RFID has been used in many sectors for many years, but due to the evolution of RFID equipment, its application has broadened in recent years to include critical areas. As these kinds of domains face a variety of new challenges, especially in gathering, processing, analysing and securing big RFID data in real-time, that is why our middleware must support interoperability by ensuring the dissemination of RFID data according to the format requested by the IS, while supporting the heterogeneity of RFID devices, the security of RFID data, and the processing and storage of real-time RFID big data.

Aviation is one of these areas; today airlines face many challenges that have an impact on customer retention, and therefore on profitability. SITA (Société Internationale de Télécommunications Aéronautiques) reports that both the number of travellers and the volume of luggage are constantly rising. International Air Transport Association (IATA) and SITA estimates, lost baggage costs airlines worldwide $2.1 billion annually! And according to traveller surveys, baggage disputes and delays are the major causes of traveller dissatisfaction, and baggage reclaim is the main stressor for travellers. The aviation industry is one of the areas that has great opportunity to profit from RFID and IoT. To provide a better and secure system for the aviation industry and travellers, we have proposed a design for an RFID-based baggage tracking system. This project aims to build a baggage tracking system for Moroccan airports. To our knowledge, this is the first innovative academic research study on this topic in Morocco. The implementation of this system will provide an efficient tool for Moroccan airports to recover lost luggage. Figure 25A shows the scenario architecture.

One of the ways in which airlines can become more efficient is to adopt and leverage technological advances in their overall offering. RFID has demonstrated strong benefits in many areas, so it can contribute to the improvement of the baggage handling process, with the aim of solving the problem of lost baggage or at least minimising the number of lost bags, which can be reflected in customer satisfaction. The implementation of an RFID-based IoT system (in which labelled suitcases are considered the "objects" of RFID infrastructure) will offer real-time monitoring of baggage location, with accurate checking stages carried out at different phases of the travel process and without manual handling. Streamlining luggage handling using RFID technology by implementing a system that performs following our proposed middleware architecture specifications outlined previously, will benefit Moroccan airports and airlines in several ways, and by exploiting an approach such as the one described in this paper, airports will benefit from the widespread connectivity and availability of RFID data, which will be integrated into the IS. Figure 25B show some of the interfaces of our middleware, namely: Authentication interface, Dashboard, Reader configuration, Encrypted RFID data.

4.3.2 Implementation

As shown in Figure 26A, first, the Blowfish middleware layer helps encrypt RFID data as it arrives. Using the ARML layer, the middleware limits the system against unauthorized access, as there are a number of restrictions to access RFID data.

To test the functioning of our middleware we have developed an application dedicated to the airport staff. Figures 26B, C show some graphical interfaces of the application. Figure 26C illustrates that the middleware prevents the user Achraf from viewing the location of the luggage.

On arrival at the airport, the traveller is first required to go through the check in phase, in which the details of each traveller are recorded in the database with the Electronic Product Code (EPC) assigned to him. Each step in the airport is equipped with an RFID reader as shown in the Figure 27A, so each step is represented by an RFID reader. The RFID middleware is the manager of this RFID network since it is connected with this set of readers, and it receives the events generated by these 5 readers each time a bag or suitcase passes through the interrogation zone of a reader. Each event contains baggage information: EPC code, suitcase location and the time. In this way, the deployed system tracks the position of the RFID-tagged suitcases throughout their journey. And in order to detect possible problems related to the baggage handling process in real time, we consider that the maximum time for a "trip" on the conveyor is N seconds, and

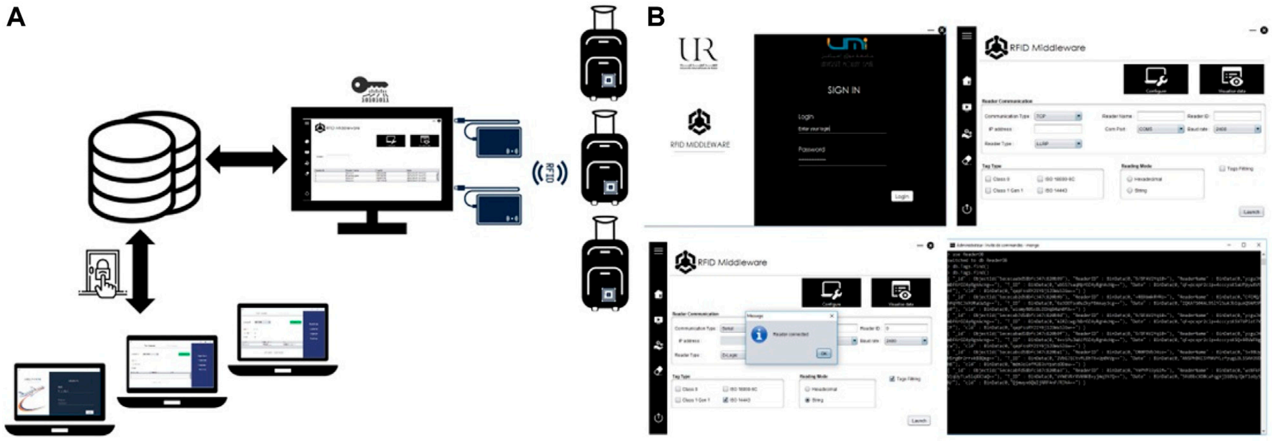
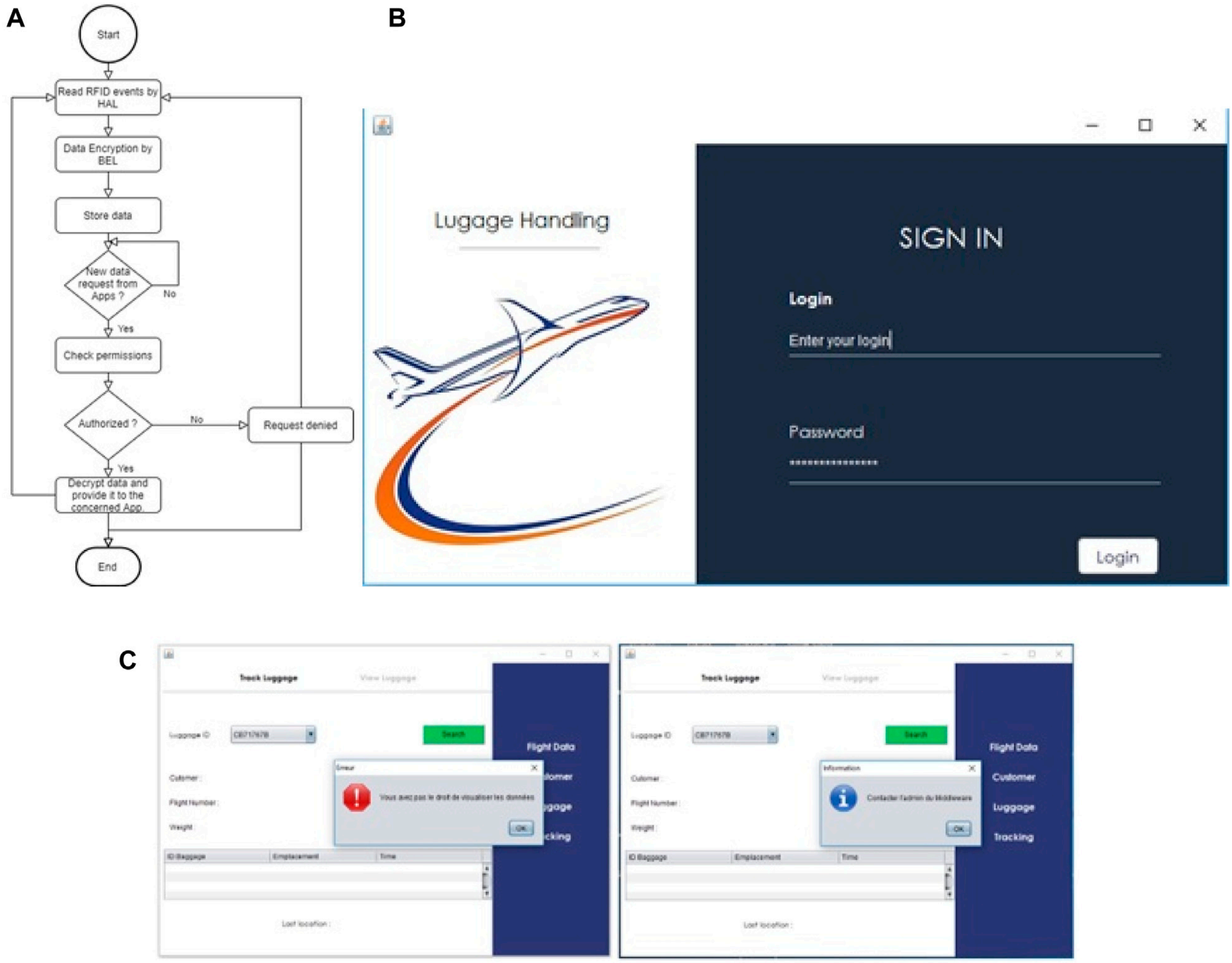


FIGURE 25
(A) Project architecture. **(B)** Middleware interfaces.



Vous avez pas le droit de visualiser les données

Contacter l'admin du Middleware

FIGURE 26
(A) Security Process of UMIUIR Middleware. **(B)** Login page of the test application. **(C)** Limitation of access to resources.

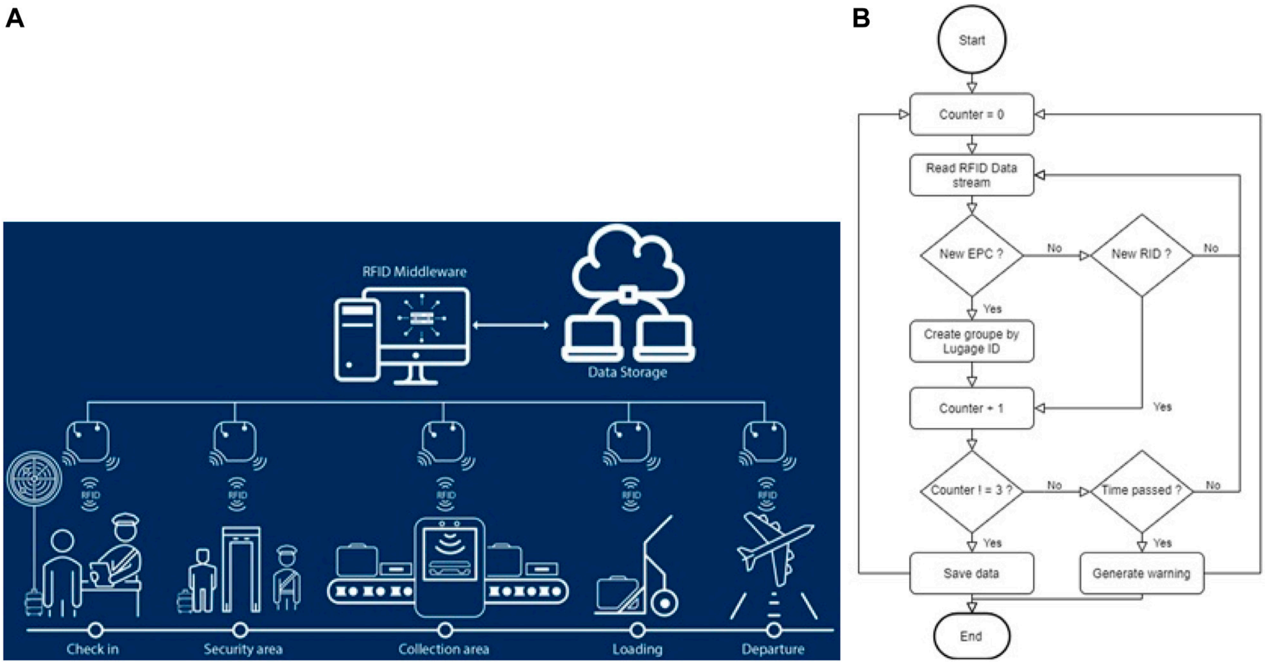


FIGURE 27
(A) Luggage handling system. **(B)** CEP Module algorithm.

after this time the baggage should be considered missing. The method therefore consists in verifying that the same baggage has passed through the five readers within the expected time interval and therefore that there are five events detected for the same baggage in the last n seconds.

We employed three readers for our implementation test, and the event processing rules are implemented using a query language, called EPL (Event Processing Language), which allows to query the content of events. The flowchart (Figure 27B) shows the pseudo code behind the operation of the CEP module which verifies that the same baggage has passed through all three readers within the expected time interval, if so, it registers the state of the baggage and continue listening to events. Otherwise, it triggers an event to notify airport staff that a suitcase is lost and continue to listen for upcoming events.

This scenario represents only an application to test and validate the concept of our proposed middleware architecture and does not reflect the full scope of possible applications that our proposed middleware could address, hence its scalability aspect.

4.4 Evaluation and comparisons

Among the most widely adopted evaluation methods for middleware is the use of application examples to quantitatively evaluate the performance of the system based on the number of lines of code (Haibi et al., 2018), but this method has certain limitations (Chung and Berhe, 2021). The most efficient method is to evaluate the middleware according to certain required functionalities.

4.4.1 Application evaluation

The ISO/IEC 9126 standard, via the portability metric, offers two evaluation factors called scalability and heterogeneous system support. The first of these is used to assess the durability of stable status in increasing of application, i.e., the number of requests coming from applications and the number of applications that can be connected to the middleware. The second factor is used to assess the middleware abstraction level to ensure communication with different backend applications.

We tested our middleware with a prototype baggage tracking application and an access control application, and also identified middleware integration scenarios in the SCM and healthcare domains by integrating it with a Hospital Information System (HIS). According to the authors of (Baruffa et al., 2020), the most important alternative factor of portability metric is heterogeneous system support; this led us to develop a generic class via the AAL to be integrated into the applications that will use our middleware.

4.4.2 Hardware evaluation

The ISO/IEC 9126 standard, again via the portability metric, allows middleware to be assessed in terms of scalability and capacity to support heterogeneous devices. Here, the term scalability means the sustainability of the steady state of the middleware as the number of RFID devices increases. Typically, manufacturers provide RFID equipment with a set of brand-specific APIs to ensure interaction with the middleware. In our proposed architecture, the HAL provides a communication interface with the heterogeneous RFID equipment network. The LLRP protocol is implemented to support readers that use this protocol, in addition to the APIs provided by manufacturers. To test our middleware, we used the readers available in our laboratory, which were D-Logic, RC522, and EM4100.

4.4.3 Context evaluation

In this assessment, metrics were assigned to the environmental contexts in which the middleware was implemented. The context can take different forms, such as the mobility of the objects to be tracked, location, time, and the user's activities. Context assessment can be done by applying scenarios and verifying that the application is context-sensitive. Real-time baggage tracking can play the role of a scenario that will assess the context sensitivity of the middleware. In the baggage traceability scenario using RFID technology, each suitcase is tagged with an RFID tag to guarantee real-time tracking. Using the CEP implementation, in the event of a problem (e.g., lost baggage), the middleware broadcasts context-dependent information, and informs only the specific users that are affected.

4.4.4 Interoperability evaluation

The functionality metric of the ISO/IEC 9126 standard was used to assess the interoperability of our middleware. Interoperability was identified as the most important alternative factor of functionality (Baruffa et al., 2020). Until formatting is applied, back-end applications cannot process the raw RFID events stream. In our proposed architecture the task of receiving and transforming the raw RFID data into business events exploitable by applications is guaranteed by the data transformation module. Currently, XML and JSON are the two most frequently used data exchange formats. However, JSON is generally preferred over XML, and has quickly come to replace it (Chung and Berhe, 2021), which motivated us to add the option to use the JSON format. The added value of our architecture is that it allows the user to choose between XML or JSON depending on the specifications of the company's IS.

In addition to these essential functionalities, our proposed architecture also supports other services that are specific to the newer areas of application of RFID technology. For example, this kind of application is characterised by a high volume of data, and in some sectors, it is not only necessary to manage very large quantities of data but also data with a very large size, although the relational DBMS are limited for use with very high data flows (Baruffa et al., 2019). Our database management module enables the storage of big data due to the integration of the NoSQL MongoDB database. There is also the real-time analysis service, which extracts relevant real-time content from huge raw RFID data stream via the CEP module, which captures information from real-time data streams.

4.4.5 Security and privacy assessment

The security and privacy assessment are intended to evaluate the security and privacy of the RFID middleware, and how it protects sensitive application data where necessary through the use of policies. This is achieved by generating a scenario in which access to decrypted data is restricted to specific parts and by testing how the middleware handles this access control policy. The example of data security that we used in our scenario is illustrated in Figure 26A. This policy maps an airport business rule that says that only the admin can access decrypted traveller information after the authentication phase. This policy restricts access to this information.

5 Conclusion

The RFID business sector has expanded considerably, and the number of labelled objects could be increased, which creates major challenges. Thus, it is necessary to integrate a Middleware system between the information system and the hardware part to ensure the collection of RFID data. This paper gives the main functions of several existing RFID Middleware which have already been implemented and tested to conclude the advantages of each of this presented middleware, in order to present the current state-of-the-art and to serve as a guideline to researchers for future research in the field of RFID middleware. It is found that a set of research studies address the RFID middleware layer. But as analysed no study covers the full set of challenges as functionalities to meet the modern applications requirements. However, it is necessary to propose a generic middleware architecture that takes into account all the challenges and limitations presented in this paper so that it can to be a solution to be integrated into the new application areas. However, special attention must be paid to respect for privacy as the use of RFID can by nature have a significant impact on privacy. When RFID technology is associated with individuals, the issue of privacy protection is paramount and therefore must be taken into consideration. In this context, this work offers RFID middleware that combines encryption and role-based access control to increase the security of RFID data. In addition, it takes into account the interoperability, storage and processing of large RFID data in real time.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

Conceptualization, AH and KO; methodology, AH, KE, and MBf; software, AH and KO; validation, KE, MBf, KO, and MBa; data curation, AH, KO, KE, and MBf; writing—original draft preparation, AH; writing—review and editing, AH, KE, and MBf; visualization, AH and KO; resources, MBf and MBa; supervision, KE, MBf, KO, and MBa; project administration, MBa; funding acquisition, MBa. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abad, I., Cerrada, C., Cerrada, J. A., Heradio, R., and Valero, E. (2012b). Managing RFID sensors networks with a general purpose RFID middleware. *Sensors* 12, 7719–7737. doi:10.3390/s120607719

Abad, M. I., Cerrada, C., Cerrada, J. A., Heradio, R., and Valero, E. (2012a). Managing RFID sensors networks with a general purpose RFID middleware. *Sensors (Basel).* 12 (6), 7719–7737. doi:10.3390/s120607719

Aftab, M. u., Qin, Z., Zakria, H., Ali, S., Pirah, H., and Khan, J., "The evaluation and comparative analysis of role based access control and attribute based access control model," 2018 *15th international computer conference on wavelet active media technology and information processing* Germany, (ICCWAMTIP, 2018, pp. 35–39. doi:10.1109/ICCWAMTIP.2018.8632578

Ahmad Kamal, U. M., Nayan, N. A., Jaafar, R., and Ismail, S. N. A. Medical equipment tracking technologies in healthcare: A review. *Front. Robotics AI*, 10, 87, 2023.

Anouar Abdelhakim Boudhir (2019). "Innovations in smart cities applications edition 2," in *The proceedings of the third international conference on smart city applications*. Editors Ahmed, K., Mohamed, Ben, and Ali, Younes (Germany: Springer).

Ahmed, N., Kumar, R., French, R. S., and Ramachandran, U. (2007). *RF2ID: A reliable middleware framework for RFID deployment*. Germany: IEEE International Parallel and Distributed Processing Symposium.

Ahmed, N., and Ramachandran, U. (2011). "RFID middleware systems: A comparative analysis," in *Unique radio innovation for the 21st century*. Editors D. Ranasinghe, Q. Sheng, and S. Zeadally (Berlin, Heidelberg: Springer).

Ajana, M. E., Boulmalf, M., Harroud, H., and Hamam, H. (2009a). 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, October 12-14. IEEE.

Ajana, M. E., Harroud, H., Boulmalf, M., and Elkoutbi, M. (2011). FlexRFID middleware in the supply chain: strategic values and challenges. *Int. J. Mob. Comput. Multimedia Commun. (IJMCMC)* 3 (2), 19–32. doi:10.4018/jmcmc.2011040102

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

Al Kukhun, D., and Sèdes, F. (2012). Security and ambient systems: A study on the evolution of access management in pervasive information systems. *Comput. Sci. Ambient Intell.*, 135–146. doi:10.1002/9781118580974.ch8

Al-Jaroodi, J., Aziz, J., and Mohamed, N., "Middleware for RFID systems: an overview," 2009 33rd Annual IEEE International Computer Software and Applications Conference, 20-24 July 2009, USA, IEEE, 2009, pp. 154–159. doi:10.1109/COMPSAC.2009.129

Ait Lhadj Lamin, S., Raghib, A., and Abou El Majd, B., "Deployment of RFID readers using a robustness multi-objective approach," 2021 Third International Conference on Transportation and Smart Technologies 22 June, 2021, China, (TST, 2021, pp. 90–95. doi:10.1109/TST52996.2021.00022

Amaral, L. A., Hessel, F. P., and Corrȇa, J. C., "Cooperative CEP-based RFID framework: A notification approach for sharing complex business events among organizations," 2011 IEEE International Conference on RFID, October, 5, 2023, USA, IEEE, \, pp. 215–222. doi:10.1109/RFID.2011.5764624

Amaral, L. A., Hessel, F. P., Bezerra, E. A., Corrêa, J. C., Longhi, O. B., and Dias, T. F. O. (2009). An adaptative framework architecture for RFID applications. *33rd Annu. IEEE Softw. Eng. Workshop* 2009, 15–24. doi:10.1109/SEW.2009.9

Amin, Y. (2013). *Printable green RFID antennas for embedded sensors*. China: Doctoral dissertation, KTH Royal Institute of Technology.

Aqeel-ur-Rehman, F., Abbasi, A. Z., and Shaikh, Z. A. (2008).Building A smart university using RFID technology, International Conference on Computer Science and Software Engineering, August 12-13, 2023, USA. IEEE.

Baruffa, G., Femminella, M., Pergolesi, M., and Reali, G. (2020). Comparison of MongoDB and Cassandra databases for spectrum monitoring as-a-service. *IEEE Trans. Netw. Serv. Manag.* 17 (1), 346–360. doi:10.1109/tnsm.2019.2942475

Baruffa, G., Femminella, M., Pergolesi, M., and Reali, G. (2019). *Comparison of MongoDB and Cassandra databases for supporting open-source platforms tailored to spectrum monitoring as-a-Service"*. Germany: IEEE Transactions on Network and Service Management.

Boontrai, D., Jingwangsa, T., and Cherntanomwong, P. (2009).Indoor localization technique using passive RFID tags, 9th International Symposium on Communications and Information Technology, 06 December 2018, New York. IEEE, 922–926.

Bouazza, H., Lazaro, A., Bouya, M., and Hadjoudja, A. (2020). A planar dual-band UHF RFID tag for metallic items. *Radioengineering* 29 (3), 504–511. doi:10.13164/re.2020.0504

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

Breje, A. R., Gyorödi, R., Gyorödi, C., Zmaranda, D., and Pecherle, G. (2018). Comparative study of data sending methods for XML and JSON models. *Int. J. Adv. Comput. Sci. Appl.* 9. doi:10.14569/ijacsa.2018.091229

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., and Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* 80, 571–583. doi:10.1016/j.jss.2006.07.009

Bridge, E. S., Wilhelm, J., Pandit, M. M., Moreno, A., Curry, C. M., Pearson, T. D., et al. (2019). An arduino-based RFID platform for animal research. *Front. Ecol. Evol.* 7, 257. doi:10.3389/fevo.2019.00257

Burnell, H. (2008). "What is RFID middleware and where is it needed?," in *RFID update*.

Cardiel, I. A., Gil, R. H., Somolinos, C. C., and Somolinos, J. C. (2012). A SCADA oriented middleware for RFID technology. *Expert Syst. Appl.* 39 (12), 11115–11124. doi:10.1016/j.eswa.2012.03.045

Chen, B., Mak, A., Lin, F., Yuan, B., Liu, W., and Wang, H. (2017). Implementation of radio frequency identification middleware with database. *Trans. Inst. Meas. Control* 39 (4), 455–465. doi:10.1177/0142331216684550

Chen, M., Gonzalez, S., Leung, V., Zhang, Q., and Li, M. (2010). A 2G-RFID-based e-healthcare system. *IEEE Wirel. Commun.* 17 (1), 37–43. doi:10.1109/mwc.2010.5416348

Chung, Y., and Berhe, T. H. (2021). Long-range UHF RFID tag for automotive license plate. *Sensors*, 54. doi:10.3390/s21072521

Dana, A. L. k., and Sèdes, F. (2009). La mise en œuvre d'un modèle de contrôle d'accès adapté aux systèmes pervasifs. Application aux équipes mobiles gériatriques. *Doc. numérique* 12, 59–78. doi:10.3166/dn.12.3.59-78 Available at : https://www.cairn.info/revue-document-numerique-2009-3-page-59.htm.

Elkhoukhi, H., Bakhouya, M., El Ouadghiri, D., and Hanifi, M. (2022). Using stream data processing for real-time occupancy detection in smart buildings. *Sensors* 22, 2371. doi:10.3390/s22062371

Erevelles, S., Fukawa, N., and Swayne, L. (2016). Big Data consumer analytics and the transformation of marketing. *J. Bus. Res.* 69 (2), 897–904. doi:10.1016/j.jbusres.2015.07.001

Fan, Y.-H., and Wu, J.-O. (2012).Middleware software for embedded systems, 26th International Conference on Advanced Information Networking and Applications Workshops, 26-29 March 2012, China. IEEE.

Floerkemeier, C., Roduner, C., and Lampe, M. (2007). RFID application development with the Accada middleware platform. *IEEE Syst. J.* 1, 82–94. doi:10.1109/jsyst.2007.909778

Ganapathi, Padmavathi, and Shanmugapriya, D. (Editors) (2019). *Handbook of research on machine and deep learning applications for cyber security* (USA: IGI Global).

Gabsi, E. S., Kortli, Y., Beroulle, V., Kieffer, Y., and Belgacem, H., "Adoption of a secure ECC-based RFID authentication protocol," 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 28-30 May 2022, Germany. IEEE, 2022, pp. 69–74. doi:10.1109/SETIT54465.2022.9875855

Gerla, M., and Reiher, P. (2015). Securing the future autonomous vehicle: A cyber-physical systems approach. *Secur. Cyber-Physical Syst.*, 197–220. doi:10.1201/B19311-12

Gouglidis, Antonios, and Mavridis, Ioannis (2012). "Grid access control models and architectures," in *Computational and data grids: Principles, applications and design*. Editor N. Preve (IGI Global), 217–234. doi:10.4018/978-1-61350-113-9.ch008

Gupta, A., and Srivastava, M. (2004). *Developing auto-ID solutions using Sun java system RFID software*.

Haibi, A., Bouazza, H., Bouya, M., El Yassini, K., Oufaska, K., Boulmalf, M., et al. (2021b). A new compact metal mountable dual-band UHF RFID tag antenna with an adapted middleware for Transport and SCM fields. *Int. J. Commun. Antenna Propag. (IRECAP)* 11 (2), 106–117. doi:10.15866/irecap.v11i2.20048

Haibi, A., El Yassini, K., and Oufaska, K. (2018).Suitcase traceability system via RFID and NoSQL database, ACM Proceedings of the 3rd International Conference on Smart City Applications, 10 October 2018, Germany. IEEE.

Haibi, A., Oufaska, K., Bouya, M., El Yassini, K., and Boulmalf, M., "Research gaps and trends in radio frequency identification: scoping review," 2022b *Microwave mediterranean symposium* China, (MMS, pp. 1–6. doi:10.1109/MMS55062.2022.9825532

Haibi, A., Oufaska, K., El Yassini, K., and Boulmalf, M. (2021a). A secure middleware architecture for real-time tracking applications, Proceedings of the 4th International Conference on Industrial Engineering and Operations Management, Ball, January 7 2023. IEOM Society, 1230–1239.

Haibi, A., Oufaska, K., and El Yassini, K. (2019). “Tracking luggage system in aerial Transport via RFID technology,” in *Innovations in smart cities applications edition 2. SCA 2018, lecture notes in intelligent transportation and infrastructure* (Germany: Springer), 259–306.

Haibi, A., Oufaska, K., Yassini, K. E., Boulmalf, M., and Bouya, M. (2022a). Systematic mapping study on RFID technology. *IEEE Access* 10, 6363–6380. doi:10.1109/ACCESS.2022.3140475

He, H., Xu, H. Y., and Zhang, Z. H. (2013). Design of lightweight RFID middleware for warehouse management system. *Adv. Mater. Res.* Vols. 706–708, 729–732. doi:10.4028/www.scientific.net/AMR.706-708.729

Herrojo, C., Paredes, F., Mata-Contreras, J., and MartínChipless-Rfid, F. (2019). Chipless-RFID: A review and recent developments. *Sensors* 19, 3385. doi:10.3390/s19153385

Hoag, J. E., and Thompson, C. W. (2006). *Architecting RFID middleware*. China: IEEE INTERNET COMPUTING.

Hu, W., Ye, W., Huang, Y., and Zhang, S. (2008).Complex event processing in RFID middleware: A three layer perspective, Third International Conference on Convergence and Hybrid Information Technology, 11-13 Nov. 2008, Germany. IEEE.

IBM Corporation (2009). *IBM corporation*. IBM WebSphere Sensor Events.

Ishikawa, T., Yumoto, Y., Kurata, M., Endo, M., Kinoshita, S., Hoshino, F., et al. (2003). 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 Univ.*

Jose, B., and Abraham, S., "Exploring the merits of nosql: A study based on mongodb," 2017a International Conference on Networks & Advances in Computational Technologies (NetACT), 20-22 July 2017, USA, NetACT, pp. 266–271. doi:10.1109/NETACT.2017.8076778

Kabir, A., Hong, B., Ryu, W., and Ahn, S. (2008). “LIT middleware: design and implementation of RFID middleware based on the EPC network architecture,” in *Dynamics in logistics*. Editors H. J. Kreowski, B. Scholz-Reiter, and H. D. Haasis (Berlin, Heidelberg: Springer).

Kefalakis, N., Leontiadis, N., Soldatos, J., Gama, K., and Donsez, D. (2008). IEEE. Supply chain management and NFC picking demonstrations using the AspireRfid middleware platform Proceedings of the ACM/IFIP/USENIX International Middleware Conference Companion on Middleware ’08 Companion01 December 2008New York

Khaddar, M. A. E., Boulmalf, M., Harroud, H., and Elkoutbi, M. (2011). “RFID middleware design and architecture,” in *Designing and deploying RFID applications* (London, United Kingdom: IntechOpen). Available: https://www.intechopen.com/chapters/18099.

Kheddam, R., Aktouf, O., and Parissis, I. (2013). SafeRFID-MW: A RFID middleware with runtime fault diagnosis. *J. Commun. Softw. Syst.* 9 (1), 57. doi:10.24138/jcomss.v9i1.158

Khemiri, N., and Sidhom, S. "*From Human and Social Indexing to Automatic Indexing in the Era of Big Data and Open Data.*" (2022): 153–164.

Kouanou, A. T., Tchiotsop, D., Kengne, R., Zephirin, D. T., Adele Armele, N. M., and Tchinda, R. (2018). An optimal big data workflow for biomedical image analysis. *Inf. Med. Unlocked* 11, 68–74. doi:10.1016/j.imu.2018.05.001

Kreowski, H. J., Scholz-Reiter, B., and Thoben, K. D. (2009). “Dynamics in logistics,” in *Second international conference* (Canada: Ldic).

Kumar, P. R., Wan, A. T., and Suhaili, W. S. H. (2020). Exploring data security and privacy issues in internet of things based on five-layer architecture. *Int. J. Commun. Netw. Inf. Secur.* 12 (1), 108–121. doi:10.17762/ijcnis.v12i1.4345

Lanthaler, M., and Gütl, C. (2012). *On using JSON-LD to create evolvable RESTful services*. China: Proceedings of the Third International Workshop on RESTful Design - WS-REST, 53.

Latif, I H. (2020). Time evaluation of different cryptography algorithms using labview IOP Conference Series: materials Science and Engineering. *IOP Publ.* 745 (1), 012039. doi:10.1088/1757-899x/745/1/012039

Lin, S., Shi, Q., and Zhou, N. (2022). Construction of a traceability system for food industry chain safety information based on internet of things technology. *Front. Public Health* 10, 857039. doi:10.3389/fpubh.2022.857039

Liu, F., Lin, Y., Ruan, Y., and Yu, H. (2009). Lightweight-ALE-based embedded RFID middleware. 5th International Conference on Wireless Communications, June 30-July 02, Beijing, Networking and Mobile Computing, 1–4.

Liu, G., Zhu, W., Saunders, C., Gao, F., and Yu, Y. (2015). Real-time complex event processing and analytics for smart grid. *Procedia Comput. Sci.* 61, 113–119. doi:10.1016/j.procs.2015.09.169

Lubna, L., Hameed, H., Ansari, S., Zahid, A., Sharif, A., Abbas, H. T., et al. (2022). Radio frequency sensing and its innovative applications in diverse sectors: A comprehensive study. *Front. Comms. Net.* 3, 1010228. doi:10.3389/frcmn.2022.1010228

Mak, A., Lam, A., and Qu, D. M. (2007). *CUHK RFID middleware", system design document*.

Marczewski, T., Ma, Y., and Sun, W. (2016). Evaluation of RFID tags to permanently mark trees in natural populations. *Front. Plant Sci.* 7, 1342. doi:10.3389/fpls.2016.01342

Marrocco, G. (2008). The art of UHF RFID antenna design: impedance-matching and size-reduction techniques. *IEEE Antennas Propag. Mag.* 50 (1), 66–79. doi:10.1109/MAP.2008.4494504

Moniruzzaman, A. B., and Hossain, S. A. (2013). *NoSQL database: New era of databases for big data analytics - classification, characteristics and comparison*.

Nash, T. A. (2010). *RFID technology and its impact on the supply chain*.

Nazeh Abdul Wahid, M. D., Ali, A., Esparham, B., and Marwan, M. D. (2018). A comparison of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish for guessing attacks prevention. *J. Comp. Sci. Appl. Inf. Technol.* 3 (2), 1–7. doi:10.15226/2474-9257/3/2/00132

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., and Belfkih, S. (2017). Big data technologies: A survey. *J. King Saud Univ. - Comput. Inf. Sci.* doi:10.1016/j.jksuci.2017.06.001

Patil, P., Narayankar, P., Narayan, D. G., and Meena, S. M. (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish. *Procedia Comput. Sci.* 78, 617–624. doi:10.1016/j.procs.2016.02.108

Perret, E. *Technologie RFID sans puce. La Revue de l’Electricité et de l’Electronique, Société de l’Électricité, de l’Électronique et des Technologies de l’Information et de la Communication*, 2017.

Prabhu, B. S., Su, X., Ramamurthy, H., Chu, C.-C., and Gadh, R. (2006). *WinRFID: A middleware for the enablement of radiofrequency identification (RFID)-Based applications", mobile, wireless, and sensor networks*, 313–336.

Pupunwiwat, P. (2012). *Tag anti-collision resolution for improved quality of RFID data streams*. China: Griffith University.

Qiyue, W., and Ping, Z. (2017).Design and application of RFID security middleware model based on elliptic curve digital signature, Proceedings of the 2nd International Forum on Management, Education and Information Technology Application, 18 June 2020, New York. IEEE.

Rance, O., Perret, E., Siragusa, R., and Lemaitre-Auger, P. (2017). *1–Automatic identification technology. RCS synthesis for chipless RFID*. Amsterdam, Netherlands: Elsevier, 51–85.

Ropraz, F. (2008b). *Group electronic business course using RFID for supply chain management project work*.

Ropraz, F. (2008a). *Using RFID for supply chain management*. Switzerland: University of Freiburg Schweiz.

Rouchdi, Y., El Yassini, K., and Oufaska, K. (2018b). “Complex event processing and role-based access control implementation in ESN middleware,” in *Innovations in smart cities and applications* (Germany: Springer), 37, 966–975.

Rouchdi, Y., El Yassini, K., and Oufaska, K. (2018c). UIR-Middleware. *Int. J. Sci. Res. (IJSR)* 7 (2), 1492–1496. doi:10.21275/23111708

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 21-27 Oct 2018, USA, IEEE, pp. 642–649. (2018a).

Rouchdi, Y., Oufaska, K., Haibi, A., El Yassini, K., and Boulmalf, M. (2019). Role-based access control in BagTrac application", international journal of knowledge engineering and soft data paradigms. *Int. J. Knowl. Eng. Soft Data Paradigms* 6 (3/4), 196–206. doi:10.1504/ijkesdp.2019.10025586

Ryu, W., Kwon, J., and Hong, B. (2011). A simulation network model to evaluate RFID middlewares. *Int. J. Softw. Eng. Knowl. Eng.* 21 (06), 779–801. doi:10.1142/s0218194011005517

Sheng, Q., Li, X., and Zeadally, S. (2008). Enabling next-generation RFID applications: solutions and challenges. *IEEE Comput.* 41 (9), 21–28. doi:10.1109/mc.2008.386

Suresh, M., and Neema, M. (2016). Hardware implementation of blowfish algorithm for the secure data transmission in internet of things. *Procedia Technol.* 25, 248–255. doi:10.1016/j.protcy.2016.08.104

Tawsif, K., Hossen, J., Emerson Raja, J., Jesmeen, M. Z. H., and Arif, E. M. H. "A review on complex event processing systems for big data", 2018 Fourth International

Conference on Information Retrieval and Knowledge Management 26-28 March 2018, USA, (CAMP, 2018.

Turcu, Cristina (Editor) (2011). *Designing and deploying RFID applications* (BoD–Books on Demand).

Vanura, J., and Kriz, P. (2018). *Perfomance evaluation of java, JavaScript and PHP serialization libraries for XML, JSON and binary formats.* China: Lecture Notes in Computer Science.

Venkatalakshmi, B., Renold, A. P., and Packiam, R. S. L. (2011). Smart RFID care [SRC] for pervasive health care system. IEEE 3rd International Conference on Communication Software and Networks, 27-29 May 2011, China, IEEE.

Venot, E. (2015). *Middleware RFID: Traçabilité et objets connectés.*

Xiang, L., Huang, J., Shao, X., and Wang, D. A mongodb-based management of planar spatial data with a flattened R-tree." *ISPRS Int. J. Geo-Information* 5. (2016): 119, doi:10.3390/ijgi5070119

Zhang, X., Song, W., and Liu, L., "An implementation approach to store GIS spatial data on NoSQL database," 2014 22nd International Conference on Geoinformatics, 25-27 June 2014, China, IEEE, pp. 1–5. doi:10.1109/GEOINFORMATICS.2014.6950846

Zhang, L., Yuan, H., Chang, S.-H., and Lam, A. (2020). Research on the overall architecture of Internet of Things middleware for intelligent industrial parks. *Int. J. Adv. Manuf. Technol.* 107, 1081–1089. doi:10.1007/s00170-019-04310-z

Zhou, L., Varadharaj an, V., and Hitchens, M. (2015). Trust enhanced cryptographic role-based access control for secure cloud data storage. *IEEE Trans. Inf. Forensics Secur.* 10 (11), 2381–2395. doi:10.1109/TIFS.2015.2455952