RFID Middleware Design and Architecture

Mehdia Ajana El Khaddar[1], Mohammed Boulmalf[3], Hamid Harroud[2] and Mohammed Elkoutbi[1]
[1]SI2M Lab, ENSIAS
[2]WML Lab, Alakhawayn University in Ifrane
[3]Canadian University of Dubai
[1,2]Morocco
[3]UAE

1. Introduction

Radio Frequency Identification (RFID) is a form of *Automatic Identification and Data Capture* (AIDC) technique (Ishikawa et al., 2003). RFID is recently being used in a wide range of areas such as Supply Chain Management (SCM), health care, traffic monitoring, retail, and access control (Polniak, 2007). The ability to store large amounts of data and identify items which are not in the line of sight has given RFID technology an edge over other automatic identification approaches such as the barcode based systems (Ishikawa et al., 2003) and optical character recognition systems (OCR) (Phoenix Software International, 2006). As an example, RFID technology integration in SCM systems has resulted in the reduced losses and improved visibility in various stages of supply chaining (Sheng et al., 2008), reduced numbers of data entry errors, efficient inventory management, and lower human labor costs in distribution centers (Tutorial-Reports, 2007).
A binary code comprising a field of bars and gaps arranged in parallel configuration is used by the barcode based identification systems. The analysis of the reflected beam on the bar gaps, allows the numerical and alphanumerical interpretation of the barcode sequence made up of narrow and wide bars. The interpreted value obtained specifies a unique code that is used for object identification. The disadvantage of the barcode system is that the barcode needs to be aligned in order to be read by the laser scanner (Ishikawa et al., 2003). The OCR based systems consist of optical machine readers used to recognize alphanumeric codes which are placed on the objects to be uniquely identified. The drawbacks of this system consist of the cost of operation, and the complexity of the OCR readers (Phoenix Software International, 2006).
The RFID systems basically consist of three elements: a tag/transponder, a reader and a middleware deployed at a host computer. The *RFID tag* is a data carrier part of the RFID system which is placed on the objects to be uniquely identified. The *RFID reader* is a device that transmits and receives data through radio waves using the connected antennas. Its functions include powering the tag, and reading/writing data to the tag. As shown Fig. 1, the signals sent by the reader's antennas form an *interrogation zone* made up of an electromagnetic field. When a tag enters this zone, it gets activated to exchange data with the reader (Al-Mousawi, 2004). Later, the identification data read by the RFID reader is processed by the software system, known as the *RFID middleware*. The RFID middleware manages readers, as well as filters and formats the RFID raw tag data so that they can be

accessed by the various interested enterprise applications (Floerkemeier & Lampe, 2005). Hence, the middleware is a key component for managing the flow of information between tag readers and enterprise applications (Burnell, 2008).
Major advantages of using RFID as an auto-ID system are the following:

- RFID readers do not require a line of sight to access data from the RFID tags.
- RFID systems can read data over varied range from few centimeters to few hundred meters.
- RFID readers can interrogate, and make RFID tags readings much faster.
- RFID systems can read and write different sizes of data from / to the tag, based on the type of tag.
- RFID systems can read tags in harsh environments, without any human interference.

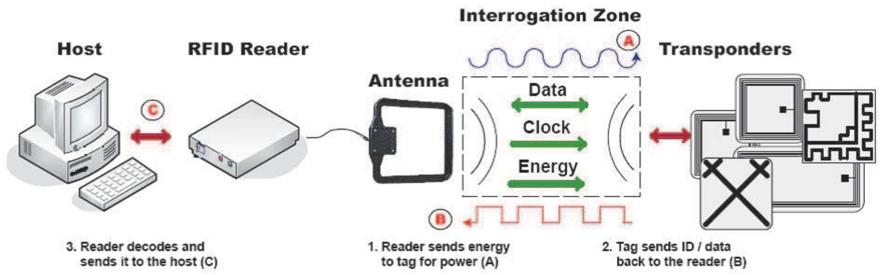
Interrogation Zone
Host
RFID Reader
Transponders
Antenna
Data
Clock
Energy
A
B
C
3. Reader decodes and sends it to the host (C)
1. Reader sends energy to tag for power (A)
2. Tag sends ID / data back to the reader (B)

Fig. 1. RFID system components (Glasser et al., 2007)

RFID technology is becoming ubiquitous as RFID systems have recently undergone significant improvements. A variety of makes and models of RFID tags and readers, combined with decreasing RFID hardware prices, are making RFID deployment more attractive (Glasser et al., 2007). In the traditional applications of RFID such as access control, networking was not a concern and there was barely a need for a RFID middleware solution. However, in the novel application areas such as SCM, a number of RFID readers could be used to capture RFID data which need to be disseminated to a variety of enterprise applications. Hence, there is no longer a one-to-one relationship between reader and application (Floerkemeier et al., 2007).
The researchers in this area have reported a vast amount of research (e.g. (Burnell, 2008; Molnar & Wagner, 2004; Parliament Office of Science and Technology, 2004) about the benefits, possible misuses, ethical issues (e.g. privacy), and technical issues (Floerkemeier & Lampe, 2004) involved in the RFID technology. However, less significant attention has been paid to the issues involved in the RFID middleware that manages large deployments of readers producing high volumes of captured data, and encapsulates applications from the low level data by transforming them into more meaningful events (Burnell, 2008). Considering this void in the RFID middleware research, herewith, we discuss the design issues of RFID middleware, present our solution called FlexRFID which addresses the above aspects, and compare it to other middleware solutions. We analyze FlexRFID to the extent to which it addresses applications' needs, and allows an easy management of devices.

2. RFID system components

RFID systems are produced by many manufacturers and exist in countless variants. However, a RFID system consists mainly of three components; the transponder/tag, reader, and RFID middleware.

2.1 RFID transponder/tag

A RFID transponder, or tag, consists of a chip and an antenna. A chip can store a unique serial number or other information based on the tag‘s type of memory. The tag‘s type of memory can be read-only, read-write, or write-once and read-many (United States Government Accountability Office, 2005). *Read-only* tags are much cheaper to produce and are used in most current applications. *Read-write* tags are useful when information needs to be updated (Al-Mousawi, 2004). The antenna is used to transmit information from the chip to the reader, and the larger the antenna the longer the read range. The RFID tag can be either attached or embedded in an object to be identified, and can be scanned by mobile or stationary readers using radio waves (United States Government Accountability Office, 2005).
RFID tags exist in three different versions: passive tags, active tags, and semi-passive / semi-active tags.

2.1.1 Passive tags

Present the simplest version of RFID tags which do not contain their own power source, such as a battery, and cannot initiate communication with the reader. The passive tag derives its power from the energy waves transmitted by the reader and responds to the reader‘s radio frequency emissions, therefore the passive tag relies entirely on the reader as its power source. A passive tag should store, at a minimum, a unique identifier for the item tagged, and can be read from a range of about 10 to 20 feet under perfect conditions (United States Government Accountability Office, 2005). Passive tags have lower production costs, meaning that they can be applied to less expensive disposable goods (e.g. a bottle of shampoo).
The cost of passive tags varies based on the radio frequency used, amount of memory, and design of the antenna, and other tag requirements. Passive tags can operate at low, high, ultrahigh, or microwave frequency. The development of passive RFID tags has made wide scale use of them in many organizations. Examples of passive tag applications include mass transit passes, building access badges, and consumer products in the supply chain (United States Government Accountability Office, 2005).

2.1.2 Active tags

Unlike passive tags, active tags contain a power source and a transmitter, in addition to the antenna and chip, and send a continuous signal. These tags typically have read/write capabilities; tag data can be rewritten and/or modified. Active tags can initiate communication and communicate over longer distances up to 750 feet, depending on the battery power. Because these tags contain more hardware than passive RFID tags, they are more expensive and are reserved for costly items that are read over greater distances (United States Government Accountability Office, 2005). RFID manufacturers typically do not quote prices for active tags without first determining their storage type and quantity, and range.

2.1.3 Semi-passive tags

This type of tags is called also semi-active tags. Semi-passive tags do not initiate communication with the reader but contain batteries that allow the tag to perform other functions, such as monitoring environmental conditions and powering the tag‘s internal electronics. In order to conserve battery life, some semi-passive tags do not actively transmit a signal to the reader. Instead, they remain dormant until they receive a signal from the

reader. Semi-passive tags can be connected to sensors to store information for container security devices (United States Government Accountability Office, 2005). Semi-passive tags have the middle transmission range and cost (Vacca, 2009).
As a summary, passive tags are consequently much lighter than active tags, less expensive, and offer a virtually unlimited operational lifetime. The trade off is that they have shorter read ranges than active tags and require a higher-powered reader (Association for Automatic Identification and Mobility, n.d.). Table 1 shows a comparison among passive, semi-passive, and active tags.

	Passive Tags	**Semi-Passive Tags**	**Active Tags**
On board power supply	No (From Reader)	Yes (Internal Battery)	Yes (Internal Battery)
Transmission range	Short (up to 6.096 meters)	Medium (up to 30.48 meters)	Long (up to 228.6 m)
Communication pattern	Passive	Passive	Proactive
Cost	Cheap	Medium	Expensive
Type of memory	Mostly Read-Only	Read-Write	Read-Write
Life of tag	Up to 20 years	2 to 7 years	5 to 10 years

Table 1. Characteristics of passive, semi passive and active RFID tags (United States Government Accountability Office, 2005; Vacca, 2009)

2.1.4 RFID tags by type of memory

RFID Tags have various types of memory (United States Government Accountability Office, 2005):
Read-Only tags: have minimal storage capacity (typically less than 64 bits) and contain permanently programmed data that cannot be altered. These tags primarily contain item identification information and have been used in libraries and video rental stores.
Read-Write tags: in addition to storing data, they can allow the data to be updated when necessary. Consequently, they have larger memory capacity and are more expensive than read-only tags. These tags are typically used where data may need to be altered throughout a product‘s life cycle, such as in manufacturing or in supply chain management. Read-Write tags have three main procedures for managing and storing data:

- *EEPROM* (Electrically Erasable Programmable Read-Only Memory): is a type of non-volatile memory used to store small amounts of data that must be saved when power is removed. It is the most dominant procedure in many RFID systems, but has the disadvantages of high power consumption during the writing operation and a limited number of write cycles (Al-Mousawi, 2004).
- *FRAM* (Ferromagnetic Random Access Memory): its read power consumption is lower than the EEPROM by a factor of 100 and the writing time is 1000 times lower. Because of manufacturing problems, its widespread introduction onto the market was affected (Al-Mousawi, 2004).
- *SRAM* (Static Random Access Memory): SRAM are used for data storage in microwave system which facilitate very fast write cycles. The disadvantage of this procedure is that the data requires an uninterruptible power supply from an auxiliary battery (active transponder) (Al-Mousawi, 2004).

Write-Once, Read-Many tags: allow information to be stored once, but does not allow subsequent updates to the data. This tag provides the security features of a Read-Only tag while adding the additional functionality of Read-Write tags.

2.1.5 RFID tags operation frequencies

RFID tags operate in several frequency bands. Most of the used frequencies are those that are in the Industrial, Scientific or Medical (ISM) frequency ranges. RFID frequencies are divided into the following three basic ranges:

- *Low Frequency (LF)*: this range operates between 30 and 500 KHz. However 125-134 KHz is the most ordinary range used in animal tracking, car immobilizers, security access, asset tracking etc. LF tags are commonly used where there are liquids, electrical noise, or metals present and when a fast read rate is not required. Most of low frequency systems operate without the need of integrated battery in their tags, have short reading ranges, and are lower system costs (Al-Mousawi, 2004).
- *High Frequency (HF)*: this range operates between 10-15 MHz, but 13.56 MHz HF tags are the most commonly used, due mainly to the relatively wide adoption of smart cards based on RFID technology. The cost of the high frequency systems is inexpensive, but higher than the low frequency systems, they have longer read ranges and higher reading speeds than the LF systems. The HF systems are used in access control and smart cards (Al-Mousawi, 2004).
- *Ultra High Frequency (UHF) and Microwave Frequency*: Ultra High Frequency Systems operate between 400 and 1000 MHz and microwave frequencies between 2.4 and 2.5 GHz. These systems are the most expensive compared to the others. UHF tags are considered as being the most practical for item-level tracking as they offer a good balance between range (typically less than a few meters), a high reading speed, and the ability to read multiple tags. Unlike the other systems, line of sight is required for the communication between RFID reader and tags. UHF systems have a very long read range, and are used for such applications as railroad car tracking and automated toll collection. Microwave frequency band is also used by many other systems e.g. Bluetooth and Wi-Fi systems (Al-Mousawi, 2004).

2.2 RFID reader

A RFID Reader is a scanning device that reliably reads the tags and communicates the results to the middleware. A reader uses its own antennae to communicate with the tag by broadcasting radio waves to which all tags within range will respond. Readers can process multiple items at once, allowing for increased read processing times. They can be either mobile or stationary, and they are differentiated by their storage capacity, processing capability, and the frequency they can read (United States Government Accountability Office, 2005).
RFID reader consists of the following functional blocks:

2.2.1 HF interface

The master part of the reader which has these functions (Al-Mousawi, 2004):

- Supplying RFID transponders with power by generating high frequency power;
- Modulation of the signal to the transponder;
- Reception and demodulation of signals from the transponders.

2.2.2 Control unit

The slave part of the reader that performs the following functionalities (Al-Mousawi, 2004):

- Communication and execution of the application software's commands;
- Signal coding and decoding;
- Communication control with a transponder.

Some RFID readers have additional functionalities like *anti-collision algorithm*, *encryption* and *decryption* of transferred data, and *transponder-reader authentication* (Al-Mousawi, 2004).
Different designs of readers exist, because different applications have different requirements from each other. RFID readers are classified into three types (Al-Mousawi, 2004):

- *OEM readers*: Original Equipment Manufacturers readers are mostly used for data capture systems, access control systems, and robots.
- *Industrial use readers*: used in assembly and manufacturing plant.
- *Portable readers*: These readers are more mobile than the other readers, and supported with a LCD display and keypad. This kind of readers is used in animal identification, device control and asset management applications.

2.3 RFID middleware

The middleware refers broadly to software or devices that connect RFID readers and the data they collect, to enterprise information systems. RFID middleware helps making sense of RFID tag reads, applies filtering, formatting and logic to tag data captured by a reader, and provides this processed data to back-end applications (Burnell, 2008). RFID middleware serves in managing the flow of data between tag readers and enterprise applications, and is responsible for the quality, and therefore usability of the information. It provides readers connectivity, context-based filtering and routing, and enterprise / B2B integration. RFID middleware design and components will be discussed further in the next sections.
When designing a RFID middleware solution, the following issues need to be considered:

- **Multiple hardware support**: The middleware must provide a common interface to access different kinds of hardware offering different features.
- **Synchronization and scheduling**: There should be intelligent scheduling and synchronization among all the processes of the middleware. This minimizes the latency and improves the efficiency of the middleware.
- **Real-time handling of incoming data from the RFID readers**: The middleware should handle the huge amount of data captured by the connected readers in real time without read misses.
- **Interfacing with multiple applications**: The middleware should be capable of interacting with multiple applications simultaneously, by catering to all the requirements of the applications with minimal latency.
- **Device neutral interface to the applications**: The application developer should only use the generic set of interfaces provided by the middleware independently of the type of hardware connected to the system.
- **Scalability**: The middleware design must allow easy integration of new hardware and data processing features.

3. RFID middleware components

A RFID middleware is the interface that sits between the RFID hardware and RFID applications. It provides the following advantages:

- It hides the RFID hardware details from the applications;
- It handles and processes the raw RFID data before passing it as aggregated events to the applications;
- It provides an application level interface for managing RFID readers and querying the RFID data.

A layer of the RFID middleware incorporates all the device drivers of different hardware and exposes to the application standard interfaces to access this hardware. If the application was provided with all the device drivers of all connected readers, it will be a hard job to manage and interface each of the devices. The application developer will then need to understand all the hardware specific internals and operations. Also, the application, if provided with the huge amount of raw tag data reported by the readers, will find it very difficult to process the data in real time. A RFID middleware provides a standardized way of dealing with this flood of information, which processes the raw data and provides the application with clean and filtered data.

As shown in Fig. 2 a RFID middleware is generally composed of four major layers:

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

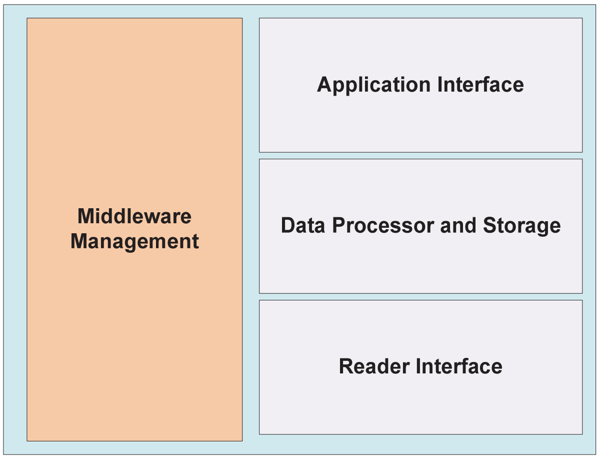

Fig. 2. RFID middleware components

3.1 Reader interface

The reader interface is the lowest layer of the RFID middleware which handles the interaction with the RFID hardware. It maintains the device drivers of all the devices supported by the system, and manages all the hardware related parameters like reader protocol, air interface, and host-side communication.

3.2 Data processor and storage

The data processor and storage layer is responsible for processing and storing the raw data coming from the readers. Examples of processing logic carried by this layer are data filtering, aggregation, and transformation. This layer also processes the data level events associated with a specific application.

3.3 Application interface

The application interface provides the application with an API to access, communicate, and configure the RFID middleware. It integrates the enterprise applications with the RFID middleware by translating the applications' requests to low level middleware commands.

3.4 Middleware management

The middleware management layer helps managing the configuration of the RFID middleware, and provides the following capabilities:

- Add, configure, and modify connected RFID readers;
- Modify application level parameters such as filters, and duplicate removal timing window;
- Add and remove services supported by the RFID middleware.

RFID readers are typically abstracted as a logical reader which is either a collection of several readers or a part of the reader. This grouping mechanism is used where there is a need to have a set of readers capturing data from a particular area such as a warehouse with many loading docks. The advantage of this is that the application can query a small number of logical readers rather than having to aggregate events from each of the individual readers.

There are two standardized interaction models used to define the communication between the middleware and the applications. An application can operate at *synchronous mode* when requesting services on demand or *asynchronous mode* when it registers for information to be sent to it when certain conditions are met. RFID middleware usually provide some kind of data filtering, because sometimes it might be required to report only certain type and value of the tag data to the application. The application needs to provide a set of defined patterns to the middleware. The middleware then allows only data that matches the pattern to be reported to the application. E.g. if an application needs to see only tag data that starts with a specific pattern such as "XYZ20", the filter can be set to this value by the application and communicated to the middleware (Al-Mousawi, 2004).

4. Examples of RFID middleware solutions

4.1 The savant middleware

There have been some proposals and research work involving middleware design and RFID data processing. The Auto-ID Center has developed a middleware component called Savant (Clark et al., 2003) that collects, accumulates, and processes Electronic Product Code (EPC) data obtained from several RF readers. It adjusts multiple readings of a tag, and performs tasks such as archiving data, and inventory control (Ishikawa et al., 2003).

The Savant has a set of *Processing Modules* or *Services* which may be combined to meet the user's application's needs. This modular structure allows innovation to be promoted by independent groups of people, which helps avoiding the creation of a single monolithic specification that attempts to satisfy all needs for everybody (Clark et al., 2003). Fig. 3 shows the three key elements of the Savant middleware architecture: *Event Management System* (EMS), *Real-Time in-Memory Data Structure/ Real-Time in-Memory Event Database* (RIED) and *Task Management System* (TMS) (Ishikawa et al., 2003).

The EMS provides a JAVA API for different types of RF readers and it serves to collect tag read events. The EMS allows adapters to be written for various types of readers, collecting

EPC data from readers in a standard format, allowing filters to be written to smooth or clean EPC data, allowing various loggers to be written, and buffering events to enable loggers, filters and adapters to operate without blocking each other.
The EMS is composed of the following elements (Auto-ID Center, n.d.):

- *Reader Interface*: Allows readers and adapters to communicate events detected by the Auto-ID readers
- *Reader Adapters*: Communicate with readers to capture EPC events
- *Event Loggers (Event Consumers):* Allow for varied processing of events; store the information in the database, store events in a memory data structure, and broadcast the events to remote servers
- *Event Queues (Event Forwarders):* Handle multiple reader event loggers with synchronous implementations

The RIED is an in-memory database that can be used to store event information by Edge Savants. It provides the same interface as a database, but offers much better performance. The RIED should be a high-performance in-memory and a multi-versioned database.
The TMS manages tasks, just as the operating system manages processes, and provides an interface for task management. Task examples include data gathering, remote task scheduling, personnel alerts, and remote upload. The TMS should be a platform-independent system requiring little memory processing power, should automatically upgrade the tasks it executes, and should present a well-defined, interoperable external interface to schedule, monitor, and remove tasks. Tasks should also be written in a platform-independent language using a simple well-defined SDK.

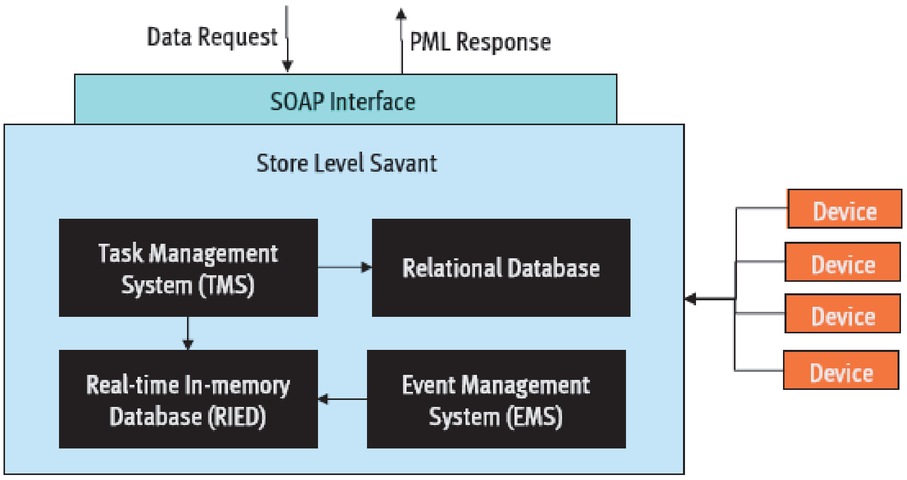

Fig. 3. Savant middleware key components

4.2 WinRFID middleware

WinRFID (Prabhu et al., 2005a) developed at the University of California Los Angeles (UCLA), is another middleware architecture that uses web services and enables rapid RFID applications development. It is a multi-layered middleware that consists of five main layers shown in Fig. 4.

The *physical layer* deals with the hardware consisting of readers, tags and other sensors. This layer abstracts the hardware elements; readers, tags and host I/O interfaces. This abstraction allows extending the middleware capabilities in the advent of introduction of new RFID technology (Prabhu et al., 2005).

The *protocol layer:* The ability to support multiple tag protocols and add new ones is becoming imperative in middleware designs. The protocol layer of the WinRFID middleware allows abstracting the reader-tag protocols. It wraps the command syntax and semantics of a variety of published protocols such as ISO 15693, ISO 14443, ISO 18000–6 A/B, ICode, EPC Class 0 and EPC Class 1. It also deals with protocol specifics such as byte-based, block or even page reading and writing, structure and length of the command frames, partitioning of the tag memory space, checksums, etc (Prabhu et al., 2005 b).

The *data processing layer* deals with processing data streams generated by the network of readers. It includes processing rules that deal with problems due to tag density, read/write distance, orientation of tags and material of item that introduce inconsistencies in reading or writing such as multiple reads of the same tag, some tags not being read, erroneous reads, etc. All of these discrepancies are processed as exceptions and a variety of altering systems are available for resolution such as emails, messages, and user defined triggers (Prabhu et al., 2005 b).

The *XML framework layer* formats the cleaned tag data in a variety of ways to a higher level XML based representation. The purpose of this layer is to provide data in a suitable format to the application layer for decision making (Prabhu et al., 2005 b).

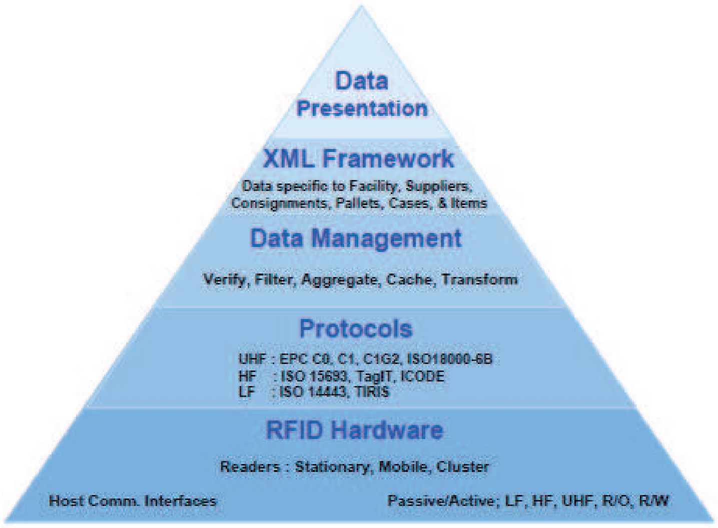
Data Presentation
XML Framework
Data specific to Facility, Suppliers, Consignments, Pallets, Cases, & Items
Data Management
Verify, Filter, Aggregate, Cache, Transform
Protocols
UHF : EPC C0, C1, C1G2, ISO18000-6B
HF : ISO 15693, TagIT, ICODE
LF : ISO 14443, TIRIS
RFID Hardware
Readers : Stationary,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

Fig. 4. WinRFID middleware multi-layered architecture (Prabhu et al., 2005 b)

The *data presentation layer* presents the data as per the requirements of end-users or different applications requirements. It facilitates data visualization for decision making. This layer supports two components the portal and the database connector. The portal provides the users with an interface to subscribe to the information of interest. For the database connector, currently the middleware can populate SQL Server and Oracle

RDBMS. The databases get populated in an asynchronous fashion in a trickle mode – a process with least priority so as to avoid the edge hosts getting locked up (Prabhu et al., 2005 b).
WinRFID exploits the .Net framework's runtime plug-in feature to support the addition of new readers, protocols, and data transformation rules with minimum disruption of the existing infrastructure (Prabhu et al., 2005 a).

4.3 The WebSphere RFID middleware

The WebSphere RFID middleware solution, designed by IBM, consists of three main components as shown in Fig. 5: *RFID devices*, *WebSphere Premises Server*, and *Websphere Business Integration Server* (IBM Corporation, 2009).
The IBM WebSphere is a sensor enabled product that allows sensor data aggregation and analysis, deriving insights from sensor data and integrating those insights with the SOA business processes. The software provides the use of intelligent business rules that manage complex event identification and processing (IBM Corporation, 2009).
This solution expands device services allowing a single platform to support multiple sensor types, and supports workflow tooling for sensor data integration with business processes (IBM Corporation, 2009). Therefore, it delivers new and enhanced capabilities to create a robust, flexible, and scalable platform for capturing new business value from sensor data.

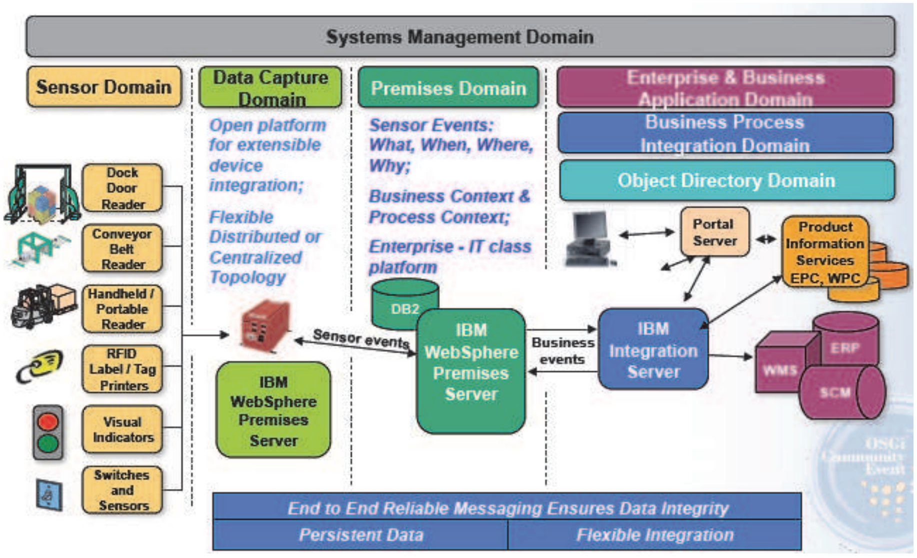
Systems Management Domain
Sensor Domain
Data Capture Domain
Premises Domain
Enterprise & Business Application Domain
Business Process Integration Domain
Object Directory Domain
Open platform for extensible device integration;
Flexible Distributed or Centralized Topology
Sensor Events: What, When, Where, 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
WMS
ERP
SCM
End to End Reliable Messaging Ensures Data Integrity
Persistent Data
Flexible Integration

Fig. 5. The IBM sensor and actuator solutions framework (Eisma, 2008)

4.4 The Sun JAVA RFID system

Sun Java System RFID software is a Java based commercial middleware platform provided by Sun. It is a critical RFID infrastructure component that allows a safe, secure, and efficient data and device integration from the edge of the enterprise into enterprise application systems. It has a dynamic, service provisioning architecture that enables scaling from small pilots to large deployments with high data volume (Sun Microsystems, 2006 b).

The Java System RFID Software supports a variety of new and existing standards, such as EPC, ISO, Gen 2, passive and active tags and devices, read/write tags, and commercial and government standards. It is a part of the Java Enterprise System (JES) and has four components as shown in Fig. 6: the *RFID Event Manager*, the *RFID Management Console*, the *RFID Information Server*, and a *Software Development Kit* (SDK). The RFID Event Manager is a Jini-based event management system that facilitates the capture, filtering, and eventual storage of events generated by RFID readers. The RFID Management Console provides a browser based management interface, which allows configuration of various attributes and parameters of the middleware. The RFID Information Server is responsible for storing and querying the EPC related data, it also manages inter Enterprise handling of the data. The SDK provides a development platform to build custom applications (Sun Microsystems, 2006 a).
The Sun middleware exposes to the application, the hardware as logical readers. These logical readers may be a collection of one or more physical readers that the application can select and apply the various processing parameters to the group (Sun Microsystems, 2006 a).

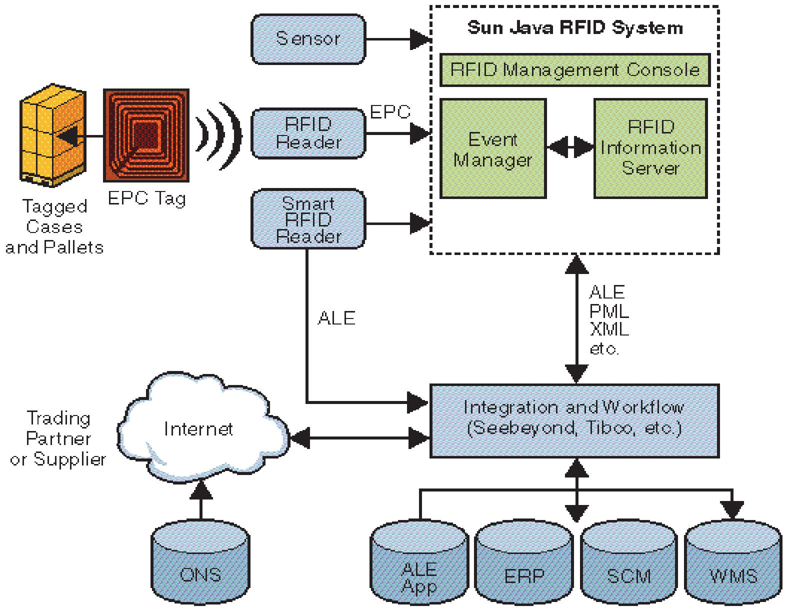

Fig. 6. The Sun Java RFID system function in the EPC network (Sun Microsystems, 2006 c)

All of these middleware designs aim at providing a scalable solution for gathering, filtering, and providing clean RFID data to the end-user. However, there are still many open issues. The reliability of RFID data needs to be improved since inaccurate data could misguide the application users. The accumulation of RFID data generated in high volumes, may lead to slower queries and updates, therefore efficient RFID data management solutions such as data transformation, aggregation, and dissemination should be investigated. Raw RFID data is not of significant value until it is aggregated with other data to obtain appropriate inferences, and transformed into a suitable form for application level interaction. Also, the applications with high security requirements are increasingly using RFID; therefore support for data security and confidentiality is needed. However, such support should maintain a desirable system performance. RFID also raises the privacy concerns because of its potential to leak proprietary information and ability to track private information such as the spending history of a consumer. Technical solutions must be implemented to ensure that private data is not compromised with (Sheng et al., 2008).

While the Savant middleware architecture provides features for cleaning the data and interfacing with different kinds of RF readers, it has limited built-in functionality for addressing business rules management, dealing with all types of sensor devices and providing data dissemination, filtering, and aggregation. Also, none of WinRFID and IBM WebSphere considers the business rules policies implementation, especially the ones concerned with security and privacy.

As compared to the related work described herewith, the distinguishing aspects of our FlexRFID middleware solution are as follows: the FlexRFID design aims to provide the applications with a device neutral interface to communicate simultaneously with many different hardware devices, creating an intelligent RFID network. It also provides an interface to access the hardware for the management and monitoring purposes. The FlexRFID provides all data processing capabilities along with the security and privacy features included in the data processing layer and enforced by a policy based management module for the business events, referred to as the Business Rules layer. The modular and layered design of FlexRFID allows integration of new features with little effort. The design also permits seamless integration of different types of enterprise applications. More detail about the FlexRFID middleware architecture is presented in the next section.

5. FlexRFID: a flexible middleware for RFID applications development

The FlexRFID middleware architecture takes into account the design issues discussed above. As shown in Fig. 7, FlexRFID is part of a three-tier architecture consisting of: the backend applications layer, FlexRFID middleware layer, and hardware layer consisting of diverse types of sensors and devices.

The *Diverse Types of Sensors and Devices layer* comprises RFID readers, sensors and other industrial automation devices. Such approach allows incredible flexibility in the selection of devices, lets companies build their enterprise solutions without handling low-level programming, and allows creating an intelligent sensor network, where RFID readers are choreographed with other devices. There are diverse makes and models of devices, which require a middleware layer that monitors, manages, coordinates, and obtains data from the different devices. In FlexRFID, these functions are taken care of before processing the raw data and applying business logic to them. Our approach is to use a *Device Abstraction Layer* (DAL) that abstracts the interaction with the physical network of devices. The FlexRFID middleware incorporates three other layers which are: *Business Event and Data Processing Layer* (BEDPL), *Business Rules Layer* (BRL), and *Application Abstraction Layer* (AAL) (Ajana et al., 2009).

5.1 Device abstraction layer (DAL)

The Device Abstraction Layer of the FlexRFID middleware is responsible for interaction with various devices and data sources independently of their characteristics. The *Data Source Abstraction Module* (DSAM) of the DAL provides a standard view of data regardless of the data source protocol (e.g. EPC Gen2, ISO 15693, and ISO14443A), air interface (e.g. UHF, HF), power supply, type, and memory size of a device. The *Device Abstraction Module* (DAM) of the DAL provides a common interface to access hardware devices with different characteristics such as protocols, air interface, and host-side communication interface (e.g. USB, Serial Port, Ethernet port). The DAM exposes simple functions like open, close, read, write, etc. that trigger the complex operations of the devices. Both, the DSAM and the DAM allow the FlexRFID middleware to be extendable to support various data sources and devices. The *Device Management and Monitoring Module* (DMMM) of the DAL is responsible for dynamic loading and unloading of the driver libraries or device adaptors. This allows the FlexRFID middleware to be light weight as libraries are loaded based upon request. The DMMM configures the devices as specified by the upper layers, and also monitors and reports their status (Ajana et al., 2009).

5.2 Business event and data processing layer (BEDPL)

The BEDPL acts as a mediator between the DAL and the AAL. The services accepted by the BEDPL are first authorized by the *Business Rules Layer* (BRL) and then allowed to issue commands to the DAL in order to get the raw data and process them accordingly. Similarly the raw data are carried from the DAL, processed, and passed on to the AAL by this layer. Services provided by the BEDPL are described as follows (Ajana et al., 2009).

5.2.1 Data dissemination

A diverse set of applications across an organization are interested in the captured information. The captured data are therefore broadcasted by the data dissemination service to all the interested entities. In addition, different applications require different latencies. For example, low latency for the notifications is desired by the applications that need to respond immediately to objects' events. In contrast, some legacy applications need to receive batched updates on a daily schedule (Floerkemeier et al. 2007).

5.2.2 Data aggregation

The fine-grained data has implicit meanings and associated relationships with other data, and need to be aggregated into summaries and/or proper inferences for applications that can not deal with the increased granularity. For example, it is common that an application is only interested in an event when an object enters or leaves a certain area. Other applications may only need a total count of objects belonging to a specific category rather than a serial number of each object detected. The data aggregation service provides such kind of functionality (Floerkemeier et al. 2007).

5.2.3 Data transformation

Raw data present little value until they are transformed into a form suitable for application-level interactions. So, from an application perspective, it is desirable to provide a mechanism that turns the low-level captured data into the corresponding business event. For example, a detection of a number of tagged books at the exit door of a library can be automatically translated into a books checked out event. This requirement is taken care by the data transformation service (Floerkemeier et al. 2007).

5.2.4 Data filtering

The volumes of data generated by the different devices require significant data filtering to extract the most important information. Also, different applications are interested in different subsets of data captured. There are filtering policies available in the FlexRFID middleware policy repository of the BRL, therefore the data filtering service filters data depending on the filter characteristics provided by the application. This offers flexibility in handling multiple filtering formats (Floerkemeier et al. 2007).

5.2.5 Duplicate removal

Multiple devices may generate duplicate readings of the data, for example tags in the vicinity of a RFID reader are read continuously. This results in a large amount of repeated data, and therefore duplicate removal service prevents the reporting of these duplicate data. The application specifies a time window, so that the same data read within it are only reported once (Ajana et al., 2009).

5.2.6 Data replacement

Usually the rate at which the devices insert data in the channel buffer is slower than the read rate of the applications. However, in case the application is not responsive enough or not executing, the channel buffer gets full, and leads to buffer overflow problem. The data replacement service allows the application to specify the action to be taken in case of channel buffer overflow. The application specifies the data replacement policy stored in the BRL policies repository, which will be executed by the data replacement service (Ajana et al., 2009).

5.2.7 Data writing

Certain special data sources like RFID tags provision additional memory space for both ID and additional data. The FlexRFID middleware handles both the reading and writing of data to this additional memory (Floerkemeier et al. 2007).

5.2.8 Privacy

RFID based tracking solutions could trigger RFID tags attached to the personal belongings to reply with their ID and other private information, therefore increasing the potential of unauthorized surveillance mechanism that would pervade large parts of our lives. FlexRFID design supports dedicated privacy enhancing feature through the privacy module. The business rules of this module are stated in the privacy policy of the BRL (Ajana et al., 2009).

5.3 Business rules layer (BRL)

The BRL is a policy-based management engine that defines the rules that grant or deny access to resources and services of the FlexRFID middleware, and enforces different types of policies for filtering, aggregation, duplicate removal, privacy, and different other services. This is achieved by determining the policies to apply when an application requests the use of a service in the BEDPL. The *Middleware Policy Editor* (MPE) allows storing, retrieving, and removing policies from the *Middleware Policy Repository Database* (MPRD). When an application needs to access a service that is protected by the Business Rules Layer, the request passes through the *Middleware Policy Enforcement Point* (MPEP) which asks the *Middleware Policy Decision Point* (MPDP) whether to permit or deny access to the service by applying the privacy rule, and how the service will be processed depending on its type. The MPEP gives the MPDP the authority of decision making; whether or not to grant the application access to the service based on the description of the application attributes, and

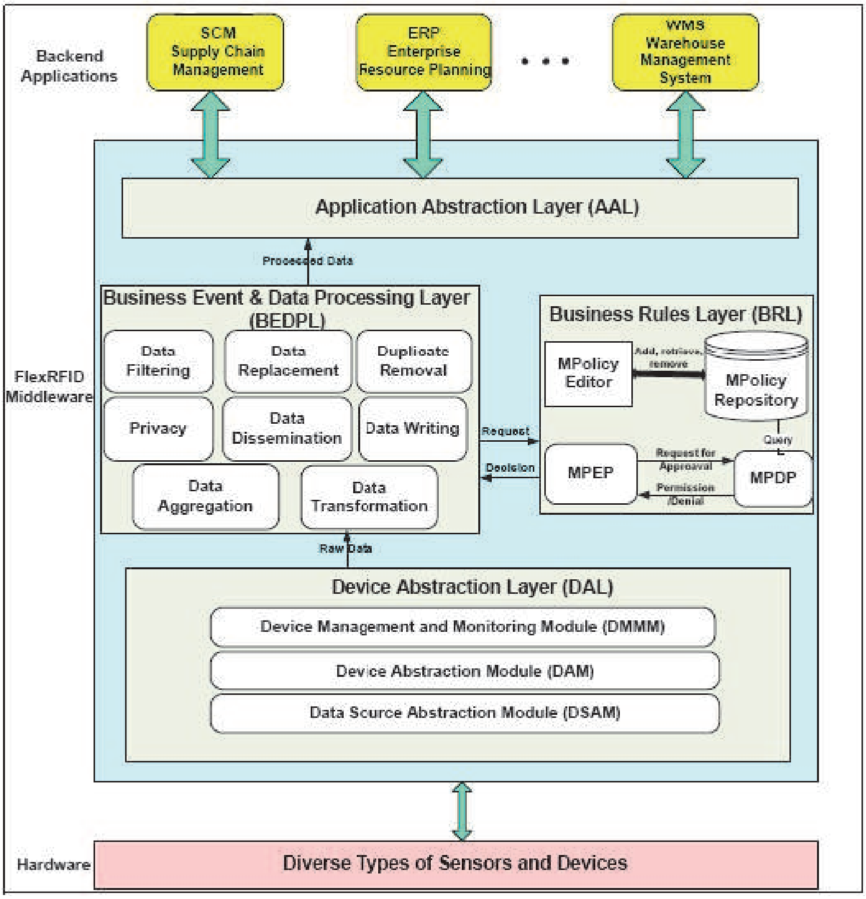
Backend Applications
SCM Supply Chain Management
ERP Enterprise Resource Planning
WMS Warehouse Management System
FlexRFID Middleware
Application Abstraction Layer (AAL)
Processed Data
Business Event & Data Processing Layer (BEDPL)
Data Filtering
Data Replacement
Duplicate Removal
Privacy
Data Dissemination
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 Scurce Abstraction Module (DSAM)
Hardware
Diverse Types of Sensors and Devices

Fig. 7. FlexRFID middleware architecture (Ajana et al., 2009)

which policies will be applied to the services used by this application. The MPDP makes its decision based on the applicable policies stored on the system. The returned decision is Permit, Deny, Indeterminate or Not Applicable. Indeterminate is returned when there is an error in processing the request and Not Applicable when no policy that applies to the request could be found (Ajana et al., 2009). Policies are operating rules used to maintain order, security, consistency, or other ways of successfully achieving a task. Examples of policies that should be available in the Business Rules Layer are: Access policy, data replacement policy, quality of service policy, and privacy policy.

Different types of applications using the FlexRFID middleware may define rules to detect events and process them using the services provided by the middleware. *Primitive events* such as observations from readers may lead to actions such as change of location. *Sequence events* consist of a sequence of primitive events of the same type, defined by the order and closeness of intervals. *Composite events* are a combination of primitive events and sequence events, and may lead to actions such as aggregation of data. Here we present some examples of rules enforced by their corresponding policies (Ajana et al., 2009):

- The *filtering rule* filters data according to predefined policies by the applications. For example, multiple readers may generate duplicate readings. To filter this, the filtering policy will scan data within a sliding window to find if there are duplicate RFID tag readings from multiple readers, and delete the duplicate if it exists. A policy for duplicate removal could specify that if readings from reader Rx and Ry have the same tag ID value within time T, then one of them is dropped.
- The *location transformation rule* serves to transform RFID readers' observations into location changes. For example, Reader R1 is mounted at a warehouse departure zone and will scan objects before their departure. A policy for this transformation could state that any observation generated from reader R1 will change the object's location to a value different from its current location.
- The *data aggregation rule* is used to detect a sequence of ordered events and generate an aggregation relationship. For instance when pallets are loaded into a truck to depart, a sequence of readings on the pallets are done, followed by (with a distinctive distance) a separate reading of the truck's EPC. This sequence of events will aggregate as a containment relationship between the pallets and the truck.
- Privacy threats in an RFID application can include covert reading, tracking over time, and individual profiling. The *privacy rule* specifies whether an application has the right to access RFID tag data, can track them over time, and use them to generate events. Applications can load into the FlexRFID middleware's Business Rules Layer privacy policies specifying how to use and configure the RFID technology to maintain the privacy of data and prevent data from tracking and hotlisting.

5.4 Application abstraction layer (AAL)

The AAL provides various applications with an interface to the hardware devices, through which the applications request the set of services provided by the FlexRFID middleware with hidden complexity (Ajana et al., 2009).

6. FlexRFID applications

6.1 Smart library application

In the late 1990s, libraries began using RFID systems to replace their electro-magnetic and barcode systems. In North America approximately 130 libraries are using RFID systems, and hundreds more are considering it. The RFID self-check systems are increasingly becoming popular since they allow patrons to check-in or check-out many items, rather than one at a time. This reduces the number of library staff needed at the circulation desk. Inventory related tasks could also be done in a fraction of the time, as a portable reader can read a whole shelf of books, and then report which are missing or misplaced. Moreover, as books are dropped in the book return station, the reader reads the tag and uses the automatic sorting system to return the book back to the shelves. A RFID tag can be used for both identifying items and securing them, and there is no need to purchase additional tags for security or use security strips separately. As patrons leave the library, the tags are read to ensure that the items have been checked out. If the item is not checked-out, the RFID readers placed near the exit detect the presence of the tag and trigger an alarm (Ayre, 2004).

A significant impediment to library use of RFID is privacy concerns associated with an item-level tagging. The tag contains static information that can be easily accessed by unauthorized readers. The privacy issues are generally described as tracking and hotlisting.

Tracking refers to the ability to track the item movement or the person carrying the item by correlating multiple observations of the item's RFID tag. Hotlisting allows building a database listing the items and their corresponding tag numbers and then using an unauthorized reader to get who is checking out items on the list. Therefore, libraries implementing RFID should use and configure the technology to maintain the privacy of patrons (Ayre, 2004).

Smart library management applications require data to be automatically read, analyzed and written back. Every patron is issued a RFID tagged library card that stores both personal information and information of the library items borrowed. Upon borrowing an item, the patron card is checked if he/she is permitted to borrow. Then, depending on the permissions, the application updates the borrowing status of the patron and the internal library database or rejects the request.

We developed a smart library RFID prototype using FlexRFID, which provides services to borrowers without having to go through an employee at the library. This prototype aims also at helping library staff to track items placed at the wrong places, and identifying most read documents in the library. This allows the visualization of important events and alerts in real time. The most important events are: item check-in, item check-out, shelf management, and item theft.

In order to illustrate the value and maturity of the FlexRFID middleware, the smart library prototype makes use of its services such as filtering, duplicate removal, transformation, aggregation, and is tested with different devices such as bar code readers, RFID readers, and sensors. A solution to the security and privacy concerns is also provided by the FlexRFID's security and privacy modules managed by policies. The smart library prototype is developed using Microsoft Visual Studio .Net. The prototype is coded using C# as a language and uses the Data Writing, Data Replacement, and Duplicate Removal services of the FlexRFID BEDPL module. The hardware used in testing the prototype consists of Intermec IF4 fixed RFID reader, Intermec 915 MHz ID Card, Intermec passive tags, and sensors used to initiate and stop the reading of tags at the entry/exit points of the library.

6.2 Supply chain management application

RFID technology has gained greater prominence and a higher level of adoption due to its recent advancements and decreasing costs across the years. The applications of RFID in the SCM have vast potential in improving effectiveness and efficiency in solving supply chain problems. RFID tags are placed on objects so that they can be uniquely identified. These objects in motion are traced throughout the supply chain from manufacturer's shop floor, to warehouses, to retail stores. Such a visibility of accurate data brings opportunities for improvement and transformation in various processes of the supply chain, and allows a wide range of organizations to realize significant productivity gains and efficiencies (Ajana et al., 2010).

Some of the key questions to be answered when applying RFID to SCM are: (1) what would be the benefits of RFID integration in supply chain? (2) What are the risks, challenges, and recommendations in adopting and implementing RFID in supply chain? (3) What processes in supply chain will be affected by RFID, and where does this technology have the potential of creating the most business value? (Ajana et al., 2010)

RFID promises to revolutionize supply chains and usher in a new era of cost savings, efficiency and business intelligence. Some of the main benefits of integrating RFID in SCM are: Automatic non-line-of-sight scanning, labor reduction, enhanced visibility, asset tracking, item level tracking, traceable warranties and product recalls, quality control and regulation, and ability to withstand harsh environments (Ajana et al., 2010). Major issues that inhibited the adoption of RFID in SCM are: the cost of tags, tag readability, the need for new data structures for RFID data management, data ownership and sharing, standardization, business process changes, and privacy (Ajana et al., 2010).

RFID can provide major benefits in the following SCM processes (Ajana et al., 2010):

- **Demand Management:** The use of RFID allows eliminating inaccuracies in data due to human errors, and provides timely data both at the item level and in aggregate about the market demand of a particular product.
- **Order Fulfillment:** Order fulfillment is a key process in meeting customer requirements and improving the effectiveness of supply chain. RFID can reduce the cost of operations in order fulfillment, and enables suppliers to automatically and accurately determine the location of an item, to track its movement through the supply chain, and to make instantaneous business decisions.
- **Manufacturing Flow Management:** The use of RFID helps manufacturers with their Just-in-Time (JIT) assembly lines by tracking where every item is in the manufacturing process and supply chain.
- **Returns Management and RFID:** RFID facilitates return management by helping retailers know if they sold the item being returned. Through the use of the *ESM* (Electronic Security Marker), RFID can tie the relationship of a particular product to a given sale and then to the return.

SCM applications target many aspects depending on supply chaining processes. One of these major aspects is inventory control. We focused on the use of FlexRFID middleware to provide input to existing tools and applications of inventory control. FlexRFID middleware deals with RFID data streaming, reactivity, integration, and heterogeneity that represent a challenge for e-logistics and SCM systems (Ajana et al., 2010):

- **Streaming**: RFID devices are becoming cheaper and widely deployed and it is now increasingly important to perform continual intelligence analysis of data captured. To relieve the SCM applications from dealing with the streaming nature of data and the fact that the data might be redundant, even unreliable in certain cases, the FlexRFID middleware is able to process such unreliable real time sensing data before delivering it to the backend system.
- **Reactivity**: RFID has promised real time global information visibility for SCM participants. To benefit from such visibility, the SCM participants have to be able to identify the interested situations and react to such situations when they happen. The events associated with the triggers have to be reported in a timely manner and notification has to be sent to interested SCM participants. The FlexRFID middleware handles this through its Business Event and Data Processing Layer and policy based Business Rules Layer.
- **Integration**: The design of FlexRFID middleware allows it to scale and support different devices and data sources that may be used at numerous points of inventory control such as Point of Sale (PoS), and smart Shelves.

The advantages of using FlexRFID for inventory control can therefore be summarized as follows:

- Report RFID data about location and inventory level in real time so that the inventory control application could place an automatic order whenever the total inventory at a warehouse or distribution center drops below a certain level.

- Report and aggregate accurate data at the PoS that will be used by the SCM application to monitor demand trends or to build a probabilistic pattern of demand that could be useful for products exhibiting high levels of dynamism in trends.
- Reduction of the Bullwhip effect, which means an exaggeration of demand in upward direction in a supply chain network. FlexRFID will provide accurate and real time information on actual sales of items that can be used for decision making and that will diminish the magnitude of the bullwhip effect. Reducing bullwhip effect would benefit industries where instances of supply-demand imbalances have high costs attached to them.
- Capturing data that gives total visibility of product movement in the supply chain. This will help to make early decisions about inventory control in case there is any interruption in the supply. This results into reduction of total lead-time for arrival of an order. Pharmaceutical and perishable product industries could benefit from this to increase total useful shelf life of items.
- Reduced inventory shrinkage: FlexRFID can transform the capture of RFID data into inventory shrinkages events including thefts and misplacement of items.
- FlexRFID allows issuing policies by the inventory control applications for items as per the requirements. E. g.: first-in-first-out (FIFO) policy for items such as, vegetables, and bread.

7. Conclusion and future work

A number of enterprise applications using RFID technique introduce a need for an infrastructure that hides proprietary device interfaces, facilitates configuration and monitoring of the devices, and processes the captured data. This chapter introduces RFID middleware and its design issues, presents some existing middleware solutions, and details the FlexRFID middleware framework that we developed to address the application requirements stated above. FlexRFID has four important layers: the Device Abstraction Layer (DAL), the Business Event and Data Processing Layer (BEDPL), and the Application Abstraction Layer (AAL). FlexRFID enables the following: communication with different types of devices; implementation of functionalities by ensuring the business rules using policy-based management; and seamless integration of various enterprise applications. The smart library application has been developed to show the usefulness of the designed middleware solution. Also the scenarios of integrating FlexRFID with an inventory management application have been set.

With respect to the future work we intend to develop all the possible scenarios and specific events that could be triggered in an SCM application for inventory control, integrate the FlexRFID middleware with an open source system for inventory control (e.g. TechLogic Inventory Control System, Opentaps…), and show how the different layers of FlexRFID middleware will work to deliver enhanced visibility of inventory in various stages of supply chaining.

Next we are intending to integrate FlexRFID with a healthcare application, and in the context of Situational Awareness; being aware of what is happening around users and understand how information, events, and actions will impact their goals, both now and in the near future. This will allow us to evaluate the FlexRFID middleware with multiple hardware configurations and applications’ requirements.

8. Acknowledgements

We would like to express our sincere appreciation to AlAkhawayn University and ENSIAS School in Morocco, for their support of this research work.

9. References

Ajana, M. E., Boulmalf, M., Harroud, H. & Elkoutbi, M. (2010). FlexRFID in the Supply Chain: Strategic Values and Challenges, *Proceedings of NGNS 2010 5th International Conference on Next Generation Networks and Services*, Marrakesh, Morocco, July 08-10, 2010

Ajana, M. E., Boulmalf, M., Harroud, H. & Hamam, H. (2009). A Policy Based Event Management Middleware for Implementing RFID Applications, *Proceedings of WiMOB 2009 5th International Conference on Wireless and Mobile Computing, Networking and Communications*, ISBN 978-0-7695-3841-9, Marrakesh, Morocco, October 12-14, 2009

Al-Mousawi, H. (2004). Performance and Reliability of Radio Frequency Identification (RFID), 01.01.2011, Available from: http://student.grm.hia.no/master/ikt04/ikt6400/g28/Document/Master_Thesis

Association for Automatic Identification and Mobility (n. d.). What is RFID?, In: *AIM*, 27.02.2011, Available from: http://www.aimglobal.org/technologies/RFID/what_is_rfid.asp

Auto-ID Center (n. d.). EMS Specification, 28.02.2011, Available from: http://www.quintessenz.org/rfid-docs/www.autoidcenter.org/media/feb03_board/oatsystems.pdf

Ayre, L. B. (2004). Position Paper: RFID and Libraries, In: *Galecia Group*, 05.03.2011, Available from: http://www.galecia.com/included/docs/position_rfid_permission.pdf

Burnell, J. (2008). What Is RFID Middleware and Where Is It Needed?, In: *RFID Update*, 27.02.2011, Available from: http://www.rfidupdate.com/articles/index.php?id=1176

Clark, S., Traub, K., Anarkat, D. & Osinski, T. (2003). Auto-ID Savant Specification 1.0, In: *Auto-ID Center*, 28.02.2011, Available from: http://www.amece.org.mx/amece/Documentos/estandares/epc/WD-savant-1_0-20030911.pdf

Eisma, A. (2008). Data Capture in IBM WebSphere Premises Server™, In: *OSGi Alliance*, 28.02.2011, available from: http://www.osgi.org/wiki/uploads/CommunityEvent2008/23_Eisma.pdf

Floerkemeier, C., Roduner, C. & Lampe, M. (2007). RFID Application Development with the Accada middleware Platform. *IEEE Systems Journal*, Vol.1 No.2, pp. 82-94, ISSN 1932- 8184

Floerkemeier, C. & Lampe, M. (2005). RFID Middleware Design: Addressing Application Requirements and RFID Constraints, *Proceedings of SOC'2005 Smart Objects Conference*, pp. 219-224, ISBN 1-59593-304-2, Grenoble, France, October, 2005

Floerkemeier, C., & Lampe, M. (2004). Issues with RFID Usage in Ubiquitous Computing Applications, 04.03.2011, Available from: http://www.vs.inf.ethz.ch/res/papers/RFIDIssues.pdf

Glasser, D. J., Goodman, K. W. & Einspruch, N. G. (2007). Chips, Tags and Scanners: Ethical Challenges for Radio Frequency Identification. *Ethics and Information Technology*, Vol.9, No.2, pp. 101-109, ISSN 1388-1957

IBM Corporation (2009). IBM WebSphere Sensor Events, 27.02.2011, Available from: http://www-01.ibm.com/software/integration/sensor-events/index.html

Ishikawa, T., Yumoto, Y., Kurata, M., Endo, M., Kinoshita, S., Hoshino, F., Yagi, S. & Nomachi, M. (2003). Applying Auto-ID to the Japanese Publication Business to Deliver Advanced Supply Chain Management, Innovative Retail Applications, and

Convenient and Safe Reader Services, In: *Auto-ID Center*, 27.02.2011, Available from: http://www.autoidlabs.org/uploads/media/KEI-AUTOID-WH004.pdf

Molnar, D. & Wagner, D. (2004). Privacy and Security in Library RFID: Issues, Practices, and Architectures, *Proceedings of ACM CCS 2004 11th Conference on Computer and Communication Security*, ISBN 1-58113-961-6, Washington, DC, USA, October, 2004

Parliament Office of Science and Technology (2004). Radio Frequency Identification (RFID), 01.03.2011, Available from: http://www.parliament.uk/documents/upload/postpn225.pdf

Phoenix Software International (2006). Optical Character Recognition (OCR): What You Need to Know, 27.02.2011, Available from: http://www.phoenixsoftware.com/pdf/ocrdataentry.pdf

Polniak, S. (2007). The RFID Case Study Book: RFID Application Stories from Around the Globe, In: *Abhisam Software*, 02.03.2011, Available from: http://www.bin95.com/case_studies/RFID_Technology_Applications.htm

Prabhu, B. S., Su, X., Ramamurthy, H., Chu, C. & Gadh, R. (2005 a). WinRFID: A Middleware for the Enablement of Radio Frequency Identification (RFID) Based Applications, In: *Wireless Internet for the Mobile Enterprise Consortium (WINMEC)*, 27.02.2011, Available from: http://www.techrepublic.com/whitepapers/winrfid-a-middleware-for-the-enablement-of-radio-frequency-identification/2349745

Prabhu, B. S., Su, X., Ramamurthy, H., Chu, P.,Qiu, C. & Gadh, R. (2005 b). WinRFID: Middleware for Distributed RFID Infrastructure, In: *Wireless Internet for the Mobile Enterprise Consortium (WINMEC)*, 27.02.2011, Available from: http://www.wireless.ucla.edu/techreports2/winrfid-middleware.pdf

Sheng, Q. Z., Li, X. & Zeadally, S. (2008). Enabling Next-Generation RFID Applications: Solutions and Challenges. *IEEE Computer*, Vol.41, No.9, pp. 21-28, ISSN 0018-9162

Sun Microsystems (2006 a). Sun Java™ System RFID Software 3.0 Developer's Guide, 27.02.2011, Available from: http://download.java.net/general/sun-rfid/Release30/Docs/Developers_Guide_819-4686.pdf

Sun Microsystems (2006 b). Sun Java™ System RFID Software 3.0, 28.02.2011, Available from: http://www.slgroup.com/Portals/0/docs/sample_docs/sun_rfid_datasheet.pdf

Sun Microsystems (2006 c). Introduction to the Sun Java System RFID Software, In: *Sun Microsystems*, Available from: http://download.oracle.com/docs/cd/E19486-01/819-4684/RFID-intro.html

United States Government Accountability Office (2005). Information Security Radio Frequency Identification Technology in the Federal Government, 27.02.2011, Available from: http://epic.org/privacy/surveillance/spotlight/0806/gao05551.pdf

Vacca, J. R. (2009). Computer and Information Security Handbook, In: *Morgan Kaufmann Publishers*, Available from: http://books.google.co.ma/books?id=TnE85sckwMAC&pg=PA206&lpg=PA206&dq=rfid+tags+power+supply+read+range+cost+type+of+memory&source=bl&ots=tWEVtCBId0&sig=1fy75A0dYwfKhIpN4sxwOzQz14Q&hl=fr&ei=UaBjTfzuI8H7lweKzIz7Cw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CBcQ6AEwAA#v=onepage&q=rfid%20tags%20power%20supply%20read%20range%20cost%20type%20of%20memory&f=false

Wal-Mart and RFID: A Case Study RFID Tags Advantages and Limitations (2007). In: *Tutorial-Reports*, 27.02.2011, Available from: http://www.tutorial-reports.com/wireless/rfid/walmart/tag-advantages.php