



Collaborative Project

ASPIRE

Advanced Sensors and lightweight Programmable
middleware for Innovative Rfid Enterprise applications

FP7 Contract: ICT-215417-CP

WP2 – Requirements and specifications

Public report - Deliverable

Review of State-of-the-art Middleware, Tools and Techniques

Due date of deliverable: M6
Actual (re)Submission date: 12.01.2010

Deliverable ID:	WP2/D2.1		
Deliverable Title:	Review of State-of-the-art Middleware, tools and Techniques		
Responsible partner:	INRIA Michel Cezon, Guillaume Vaudaux-Ruth, Luc Laurens – INRIA		
Contributors:	Loïc SCHMIDT - INRIA Nektarios Leontiadis – AIT John Soldatos -AIT Humberto Moran – OSI Ramiro Robles – IT		
Estimated Indicative Person Months:	5		
Start Date of the Project:	1 January 2008	Duration:	36 Months
Revision:	1.8		
Dissemination Level:	PU		

PROPRIETARY RIGHTS STATEMENT
The Aspire consortium release this document under
Creative Commons Attribution-ShareAlike License

Document Information

Document Name: Review of State-of-the-art Middleware, tools and Techniques
Document ID: WP2/D2.1
Revision: 1.8
Revision Date: 7 January 7, 2010
Author: INRIA
Security: PU

Approvals

	Name	Organization	Date	Visa
<i>Coordinator</i>	Neeli Rashmi Prasad	CTIF-AAU	12.01.2010	Approved
<i>Technical Coordinator</i>	John Soldatos	AIT		
<i>Quality Manager</i>	Anne Bisgaard Pors	CTIF-AAU	12.01.2010	Approved

Reviewers

Name	Organization	Date	Comments	Visa
<i>Ramiro Robles</i>	IT	07.01.2010		Approved
	NORM			
	PTV			

Document history

Revision	Date	Modification	Authors
0.1	25 Feb 08	First draft	Guillaume Vaudaux-Ruth, INRIA
0.2	28 Feb 08	Complete the section about OSS RFID middleware	John Soldatos, Nektarios Leontiadis, AIT
0.3	29 Feb 08	Add description of 3 OSS RFID Library	Loïc SCHMIDT, POPS
0.4	11 Mar 08	Updated template (reviewers' table)	Luc Laurens, INRIA
0.5	26 Mar 08	Inclusion of proprietary middleware	Humberto Moran, OSI
0.6	30 Apr 08	Complete the section about the SOTA of collaborative tools	Luc Laurens, Guillaume Vaudaux-Ruth, INRIA

Contract: 215417
Deliverable report – WP2 / D2.1

0.7	6 May 08	Additions to OSS, corrections, comments	Nektarios Leontiadis, John Soldatos, AIT
0.8	14 May 08	Updated RFID Library chapter content	Loïc SCHMIDT, POPS
0.9	22 May 08	Inclusion of the section “Integration of the RFID platforms into the mobile telecommunications infrastructure”	Ramiro Samano Robles, IT
1.0	2 June 08	Added NovaForge sheet, and minor change	Guillaume Vaudaux-Ruth, INRIA
1.1	3 June 08	Added References, Completed section 8.2. on missing pieces of state-of-the-art RFID middleware platforms	John Soldatos, AIT
1.2	4 June 08	Update “SAP and NetWeaver” sheet	Humberto Moran, OSI
1.3	9 June 08	Updated document thanks to reviewers’ feedbacks	Julien Vinay, PV Ramiro Samano Robles, IT
1.4	13 June 08	Updated RiFiDi, Sun, corrections, additions	Nektarios Leontiadis, AIT
1.5	31 March 09	Revised version after review	INRIA POPS
1.6	2 April 09	Corrected inconsistencies in section 5	Nektarios Leontiadis, AIT
1.7	5 January 10	Revised version after review	INRIA POPS
1.8	7 January 10	Final corrections and review	IT

Content

Section 1	Executive Summary	6
Section 2	Introduction	8
2.1	What is RFID?	8
2.2	Motives for advanced middleware platforms	8
2.3	Middleware functionalities, reader architectures and RFID standards	9
2.3.1	RFID standards: EPCGlobal and ISO	11
2.4	ASPIRE objectives	14
2.4.1	Open-source software (OSS)	15
2.5	Scope and organization of the document	16
Section 3	State-of-the-art OSS Middleware Platforms	18
3.1	Overview	18
3.2	Fosstrak (previously Accada)	19
3.3	RiFiDi	20
3.4	Singularity	21
3.5	Radioactive Foundation	22
3.6	Mobitec (CUHK RFID Middleware)	24
3.7	RFIDSuite	26
3.8	Logicalloy	27
3.9	Sun Java System RFID Software	28
3.10	RFID Library	30
3.10.1	RFID-perl	30
3.10.2	RFID C library	31
3.10.3	RF-Dump	31
Section 4	State-of-the-art Proprietary Middleware Platforms	32
4.1	Overview	32
4.2	OAT Systems and OAT Foundation Suit	35
4.3	SAP and NetWeaver	37
4.4	IBM and WebSphere	40
Section 5	Other approaches to RFID middleware design	42
5.1	Trends in middleware platform design	42
5.2	Integration of RFID readers into wireless mobile telecommunication networks	48
5.3	Middleware solutions for mobility applications	50
Section 6	Synthesis	54

Section 7 List of Acronyms.....56
Section 8 List of Figures57
Section 9 List of Tables.....58
Section 10 References and Bibliography.....59

Section 1 Executive Summary

Modern RFID applications at the item level require advanced middleware tools to cope with the huge amount of generated data. However, due to the relative slow adoption of RFID standards, current middleware solutions are still implemented on a per-case basis. This fact considerably increases the total cost of ownership (TCO). ASPIRE consortium aims at solving this and other issues of RFID deployments by improving state-of-the-art RFID middleware tools. ASPIRE middleware solutions aim at being lightweight, programmable, user and privacy friendly, royalty free, open source, and scalable. In order to achieve these goals, it is first necessary to have a complete literature review and analysis of existing middleware platforms, architecture approaches and complementary tools. This deliverable presents a detailed study, review and assessment of the state-of-the-art Open Source Software (OSS) and proprietary components used in RFID middleware.

An introductory section (Section 2) provides a general overview of RFID technology, the motives for advanced middleware platforms and the general objectives of the ASPIRE project. It also illustrates the importance of this deliverable, whose purposes are as follows:

- To provide insights on the features, strengths and weaknesses of state-of-the-art middleware suites. These insights will serve as valuable input towards devising the ASPIRE middleware architecture, while at the same time selecting components that could be reused in the scope of the ASPIRE middleware codebase.
- To identify areas for innovative middleware contributions of the ASPIRE project, mainly in terms of programmability, configurability, versatility and privacy-friendliness of the ASPIRE middleware. These contributions will be specified within other deliverables of this work package (notably D2.3, D2.4) and accordingly implemented mainly in the scope of WP3 and WP4.

Section 3 is dedicated to RFID OSS middleware whereas section 4 is dedicated to RFID proprietary (commercial) middleware. Note that while the ASPIRE middleware should not be directly compared to vendor products, proprietary suites could provide ideas and design inputs to the ASPIRE architecture and middleware. The characteristics of each relevant component are described and matched with ASPIRE expected requirements and specifications. Furthermore, technological and functional gaps identified between existing components and ASPIRE SMEs requirements (being part of D2.2) may be listed and could contribute to final ASPIRE framework specifications. We have found several limitations in analyzed OSS middleware platforms (e.g. configurable Business Events Generation (BEG), support for sensors, actuators, integrated development environments (IDEs) and visual tools, readers connectors and virtualization for non EPC-RP or EPC-LLRP compliant readers, and end-to-end management) which will be addressed in the ASPIRE project. As for commercial solutions, resourceful and with high licensing cost, they do not provide tools for RFID development for business users. An additional section deals with proposed middleware design methodologies found in relevant papers. Emphasis is made in those architectures

that deal with integration of RFID readers and services into wireless mobile telecommunication networks. These ideas are likely to be explored within the research to be carried out within WP5 of ASPIRE, which is dedicated to added value sensing, resource-limited and mobility scenarios. A final section summarizes our findings and provides recommendations on RFID tools that may be used to implement the ASPIRE software and middleware. It is concluded, for example, that the best two OSS candidates to be reused by ASPIRE is the UJF (University Joseph Fourier) and Fosstrak branches due to the compliance to EPC standards, their open source nature and the relative size of their developers community. Several other gaps were found also during the research reported in this deliverable and that will be adopted in the future by ASPIRE solutions. Examples of these are the lack of business layer connectors between ALE and EPCIS services, the lack of appropriate management framework for middleware components and the lack of architectural, and data processing schemes for added value sensing parameters.

Section 2 Introduction

2.1 What is RFID?

RFID (Radio-frequency identification) is the generic name given to any type of system that employs radio frequency (RF) signals for purposes of identification of objects, animals and humans [1][20][34]. An RFID system consists of tags or transponders, readers or interrogators, a middleware platform, and the back-end application servers [11]. Tags are in charge of responding to readers' requests via RF signals. Readers are in charge of requesting and collecting data from tags, and provide information to the middleware platform. The latter one performs, in turn, aggregation and filtering of unwanted data, and forwards refined information to upper layers. Finally, application servers collect RFID information from the middleware and transform it into an event with business meaning. As compared to optical bar scanners, RFID exhibits excellent recognition rates, high reliability, low dependency on environment, large storage space, and it doesn't need of line-of-sight for a reliable detection [35]. Due to these attributes, RFID promises to finally break the barrier between the physical world of objects and the virtual world of software [1].

The origins of RFID technology can be traced back to the 1940s where the principles behind radar technology and modulation schemes based on reflected power were first developed [8]. The first RFID system, however, had to wait until the invention of the transistor in 1948 and the development of integrated circuits in the 1960s. The 1960s and 70s saw the first commercial application of RFID, also called EAS system (Electronic Article Surveillance), which is a one-bit RFID system. Late 70s and early 80s witnessed further advances of integrated circuits based on CMOS (Complementary Metal-Oxide Semiconductor) technology and memory gates based on EEPROM (Electrically Erasable Programmable Read Only Memory) technology. These advances paved the way for multi-bit transponders, and hence, to new applications of RFID such as toll payment and access control [8]. Finally, during the last two decades further advances in microelectronics and radio frequency transceivers (e.g., MMIC- Monolithic Microwave Integrated circuits) allowed reduction of cost and size, while increasing the range of passive transponders. These last two facts have, in turn, facilitated RFID technology to enter into new identification domains at the item level [8].

2.2 Motives for advanced middleware platforms

Conventional applications of RFID such as access control or toll payment require a simple and unchallenging middleware solution [1]. New RFID application domains at the item level, such as inventory control and supply chain management, generate a wide variety and a huge amount of raw RFID data, thus requiring more advanced middleware platforms [1]. In addition, RFID passive technologies suffer from natural impairments such as false positive and false negatives that require new methodologies to predict and correct detection errors [1]. Other new application domain of RFID is given by the area called Mobile RFID. In Mobile RFID, readers are also provided with mobility, which makes the

technology even more ubiquitous at the expense of new issues such as reader collision and resource constrained environments. The mobility feature brings up another potential application of RFID, which is given by added value sensing. Added value sensing is useful in scenarios where environmental parameters may provide an extra feature to supply chain management applications. Therefore, new middleware platforms need to modify their architectures or procedures so as to cope with the new sensing data information. Another recent trend in RFID, for example, is given by the increasing autonomy and processing capabilities of readers. In the future, middleware solutions for this type of readers will have to focus on business layers, as conventional filtering, aggregation and context aware algorithms can be provided by the reader platform itself. Finally, perhaps two of the main issues in current RFID deployments are related to privacy and security threats [39]. Future middleware platforms must be able to deal and solve most of the issues in these areas.

2.3 Middleware functionalities, reader architectures and RFID standards

A middleware, as the name suggests, is a piece of software that lies between a lower layer processing device or software and an upper layer server or software, usually at the application level. Therefore, in RFID the middleware platform serves as a connection between the reader or interrogators and the back end application servers. Typically, an RFID middleware platform performs aggregation of data across different readers, filtering of unwanted or noisy RFID data, forwarding of relevant data to subscriber servers or application-level systems, and persistent storage for context aware and other added value services [1]. However, an RFID middleware is often given the task of managing, monitoring and configuring the different readers and interrogators [29]. According to [29] the main functionalities hosted by an RFID middleware platform can be classified as follows:

- **Configuration service set (CSS)**
 - *Network Interface configuration.* Discovers and sets reader networking parameters and identity, e.g. the IP address.
 - *Firmware management.* Distribute and manage firmware version on readers
 - *Antenna, Tag population and memory selection.* Specify reader antennas and tag population to be inventoried. In case of tag memory access, specifies memory fields to be accessed.
 - *Base Service set scheduling* Sets how different BSS services, such as tag inventory, access, and deactivation, are triggered and stopped.
 - *RF transmitter configuration* Sets transmit channel, hop sequence, and transmit power for readers.
 - *Air Interface Protocol Configuration* Configures timing, coding and modulation parameter of a specific air interface protocol on the readers.
- **Data processing service set (DPSS)**
 - *Filtering.* Removes unwanted tag identifiers from the set of tag identifiers captured, e.g. based on the product type or manufacturer encoded in the identifier.

- Aggregation. Computes aggregates in the time domain (entry/exit events) and the space domain (across reader antennas and readers) and generates the corresponding 'super'-events.
- Identifier Translation Translates between different representation of the identifier, e.g. from raw tag object identifier in hexadecimal format to EPC in URN notation.
- Persistent storage Stores RFID data captured for future application requests.
- Reliable messaging Allow RFID data to be delivered reliably in the presence of software component, system and network failures.
- Location/Movement estimation Detects false positive reads of far-away tags that are outside the 'typical' read range and estimate the direction of movement.
- Application Logic execution Interprets the RFID data captured in an application context and generate the corresponding application events, e.g. detect whether a shipment is complete.
- **Monitoring service set (MSS)**
 - Network connection monitoring Check that the reader can communicate captured RFID data over the network
 - RF environment monitoring Check RF noise and interference levels to safeguard reliable identification operation
 - Reader Monitoring Check that the reader is up and running and executing BSS as configured for example via monitoring the number of successful/-failed read and write operations.

Now, not all these functionalities are mandatory to be hosted by the middleware. This depends on the reader architecture employed. Two types of architectures can be followed: one in which many of the above functionalities are hosted by the reader itself, which will be called decentralized reader architecture, and the one in which all the functionalities, except the basic ones used in the reader, will be hosted by the middleware platform or a controller appliance (see figures below).

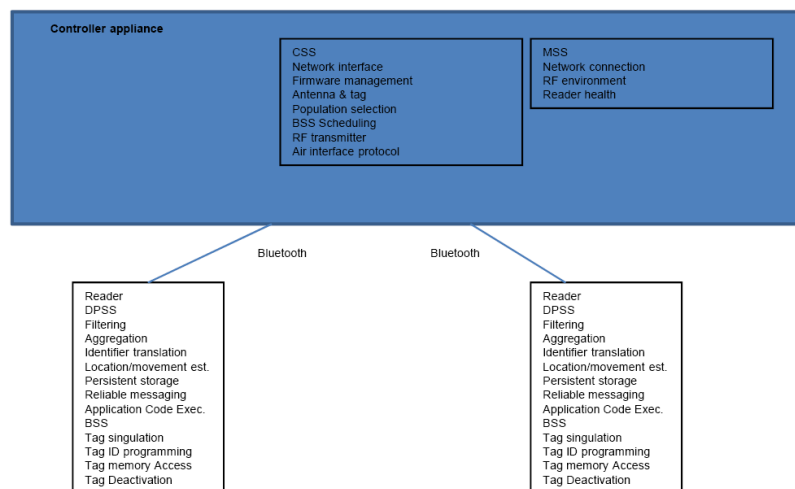


Figure 1 Decentralized reader architecture

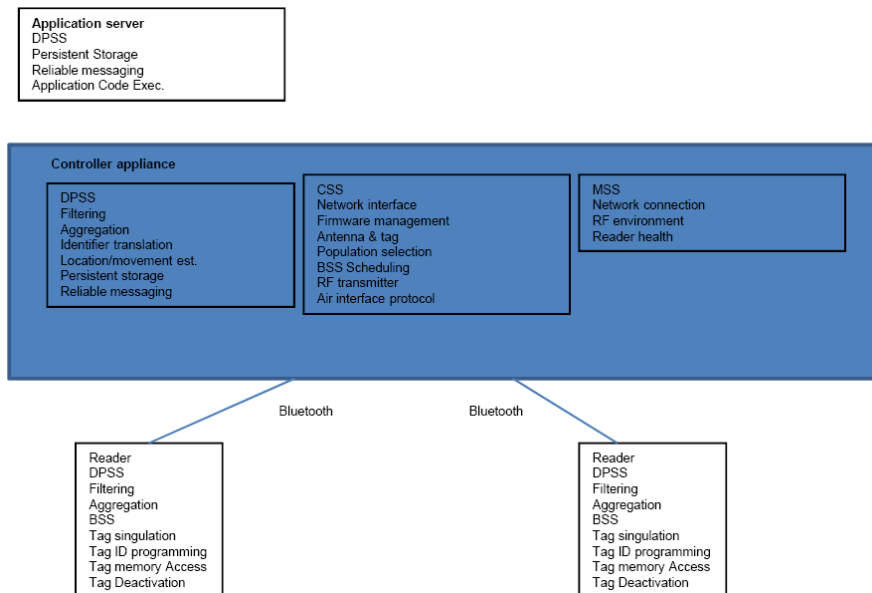


Figure 2 Centralized architecture

ASPIRE will mainly focus on centralized architectures. However, within centralized architectures emphasis will be made on low cost readers with limited processing functionalities. This work is part of the objectives in WP5.

2.3.1 RFID standards: EPCGlobal and ISO

It is difficult to provide a general classification of RFID systems due to the large variety of flavors in which they are currently presented. For example, RFID systems can be classified by their operational frequency: high frequency (HF), ultra high frequency (UHF), microwave bands, and ultra-wide band. RFID systems working on each one of these frequency bands have different properties and potential different applications. For example, it is known that as the operational frequency increases then the gain of an antenna with fixed physical dimensions also increases. This means that RFID systems at higher frequencies can potentially achieve larger reading ranges than at lower frequencies.

Another possible classification of tags is by standard. Standards aim to represent a set of common technical specifications to be followed by different manufacturers and hence provide interoperability and common interfaces. Different standards, thus, are created for different types of applications and different technical characteristics. In RFID two main types of standard prevail today: EPC (Electronic Product Code) and ISO (International Standard Organization). While EPC are slightly biased towards UHF tags for item level applications, ISO standards are biased to HF and LF tags for different kind of applications such as access control and animal tracking. Table 1 presents a classification of tags based on the different standards and the main characteristics of each one of them.

TECHNOLOGY	BAND	RANGE (meters)	DATA	CONFIDENTIALITY	INTEGRITY	AVAILABILITY
EPC Class 0/0 ⁺ (supply chain)	UHF	3	64-or 96 bits R/W block	None in standard	Parity bit. CRC error detection	Identification rate >1000 tags
EPC Class 1 Generation 1 (supply chain)	UHF	3	64-or 96 bits R/W block	None in standard	Commands have 5 parity bits CRC error detection	Lock command permanent and not protected
EPC Class 1 Generation 2 (supply chain)	UHF	3	R/W block	Masked reader to tag communications using the one time pad stream cipher Tags addressed by 16-bit random numbers	CRC error detection	Numerous readers can operate in dense configurations
ISO/IEC 18000-2 (item management)	LF	<0.010	Up to 1 Kbyte R/W	No protection on the read command Read talks first protocol No encryption or authentication	CRC error detection Permanent, factory set 64-bit ID Optional, lockable identifier code	None in standard
ISO/IEC 18000-3 (item management)	HF	<2	R/W	Reader talks first protocol 48-bit password protection on read command Quiet response in which tags won't respond to readers	CRC error detection No write protection in Mode 2 Mode 2 has 48 bit password on write commands	Multiple tag modes are non-interfering
ISO/IEC 11784-11785 (animal tracking)	LF	<0.10	64-bit identifier	Reader talks first protocol Tags addressed by 16-bit random numbers	Retagging counter CRC error detection	None in standard
ISO/IEC 10356 (contactless smart cards)	HF	<2	R/W	Reader talks first protocol Masked reader to tag communications Tags addressed by random number Quiet mode	CRC error detection	Probabilistic/slotted random anti-collision algorithm Multiple tag modes are non interfering
ISO/IEC 15693 (vicinity smart cards)	HF	1.5	Up to 1kbyte R/W	No protection on the read command No onboard encryption or authentication	Optional protections on write command Error checking on air interface	Optional Password protection on the lock command

Table 1 Different types of tags and their technical characteristics [41]

Since EPC standards look more promising for tackling item level applications in the UHF bands, particular emphasis will be made in this deliverable on middleware platforms that deploy this set of standards. The EPC set of standards is mainly concerned with the processing of data in centralized RFID architectures.

This means that this set of standards assumes that a central node or core is in charge of the coordination, configuration, collection and other functions of the platform. In **Error! Reference source not found.** we can observe the architecture framework proposed by EPC. The modules are divided into three general functionalities: Identification, capturing and exchange. These functionalities more or less resemble those of the OSI (Open System Interconnection) model. For example the identification modules are concerned with the format and mapping rules to handle identification strings and other parameters required by upper layer protocols. The capture functionality covers physical, medium access control and link layer parameter definition (e.g. the tag protocol). Going up in the architecture reference model in Figure 1, the capture functionality also includes reader protocols that define the rules, cycles, and report formats for reader or interrogators to pass information to the middleware platform. The main core of the middleware, namely the filtering and collection module, is then defined by the ALE (Application Level Event) standard, and further complemented for the communication with upper business layers by the EPC Information Services (EPCIS) standard.

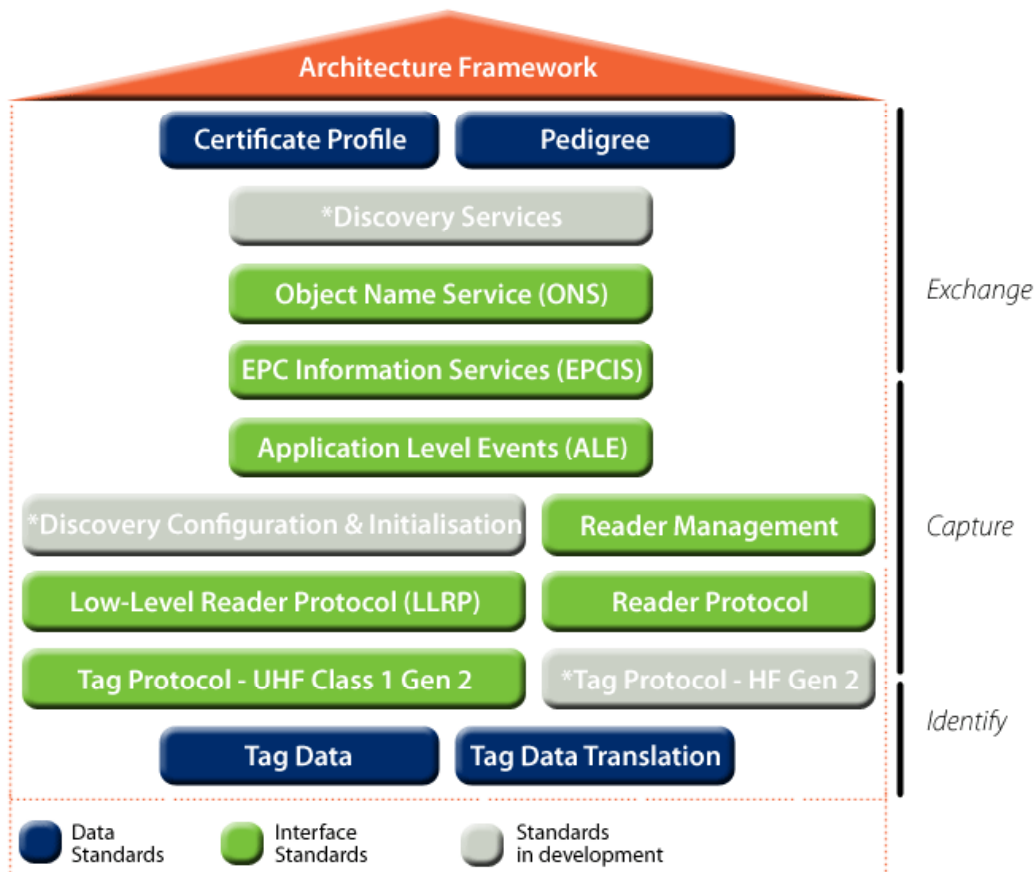


Figure 3 EPC Set of Standards

2.4 ASPIRE objectives

The research carried out in ASPIRE will provide a radical change in the current RFID deployment paradigm through innovative, programmable, royalty-free, lightweight and privacy friendly middleware. This new middleware paradigm will be particular beneficial to European SMEs, which are nowadays experiencing significant cost-barriers to RFID deployment.

European networked enterprises in general and SMEs in particular are still reluctant to adopt RFID, since they perceive RFID as unprofitable or too risky. This is largely due to the fact that the adoption of RFID technology incurs a significant Total Cost of Ownership (TCO). ASPIRE will significantly lower SME entry costs for RFID technology, through developing and providing a lightweight, royalty-free, innovative, programmable, privacy friendly, middleware platform that will facilitate low-cost development and deployment of innovative RFID solutions. This platform will act as a main vehicle for realizing the proposed swift in the current RFID deployment paradigm. Portions (i.e. specific libraries) of the ASPIRE middleware will be hosted and run on low-cost RFID-enabled microelectronic systems, in order to further lower the TCO in mobility scenarios (i.e. mobile warehouses, trucks). Hence, the ASPIRE middleware platform will be combined with innovative European developments in the area of ubiquitous RFID-based sensing (e.g., physical quantities sensing (temperature, humidity, pressure, acceleration), mobile, low-cost), towards enabling novel business cases that ensure improved business results.

The ASPIRE RFID middleware paradigm, as well as the unique and novel characteristics of the ASPIRE RFID middleware platform requires a set of OSS components which will be researched in existing OSS communities or developed by the Consortium. Specifically, ASPIRE will create a core (OSS based) middleware infrastructure, which will be appropriately enhanced with a set of programmable features/functionalities and associated tools. The ASPIRE core middleware infrastructure will attempt to overcome the limitation of existing OSS middleware platform for RFID applications. Nevertheless, ASPIRE will attempt to license and reuse components from existing middleware suites, which a view to economizing resources and allocating them in features that are not currently available as part of the state-of-the-art RFID middleware suites. As a result this deliverable reviews state-of-the-art middleware, with a dual objective:

- First, to provide valuable input to the ASPIRE architecture, as well as to ASPIRE middleware reuse tasks. Indeed, ASPIRE developers will consult this deliverable in order to identify potential reuse possibilities. The deliverable has its confidentiality status set to "Public", and can therefore be studied by other contributors to the ASPIRE middleware (e.g., OSS community developers).
- Second, to facilitate ASPIRE researchers in identifying novel features and functionalities to be implemented within the ASPIRE middleware suite. Such features include programmability aspects (to be implemented within WP4), but also enhancements to the existing EPC-oriented middleware suites (to be implemented within WP3).

2.4.1 Open-source software (OSS)

As defined by wikipedia, Open source software (OSS) is computer software for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that meets the Open Source Definition or that is in the public domain. This permits users to use, change, and improve the software, and to redistribute it in modified or unmodified forms. The Open Source Definition is as follows

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

2.5 Scope and organization of the document

This deliverable presents a summary and assessment of the state-of-the-art RFID middleware platforms. The assessment is by no means exhaustive as a detailed evaluation of each platform requires a complex methodology (e.g. [25] and [14]), procedures, and standard metrics that constitute a whole research project by their own and that in some cases have not been well defined in the literature. Instead we will present the assessment of interesting features for ASPIRE such as architectural innovative approaches, flexibility for reusability, standard compliance, user friendliness, etc. In summary, the aims of this document are the following:

- To provide insights on the features, strengths and weaknesses of state-of-the-art middleware suites. These insights will serve as valuable input towards devising the ASPIRE middleware architecture, while at the same time selecting components that could be reused in the scope of the ASPIRE middleware codebase.
- To identify areas for innovative middleware contributions of the ASPIRE project, mainly in terms of programmability, configurability, versatility and privacy-friendliness of the ASPIRE middleware.

The organization of this deliverable is as follows: Section 3 is dedicated to the evaluation of the following OSS middleware platforms: Fosstrak (previously Accada), Rifidi, Singularity, RadioActive, Mobitec, UJF RFID Suite, Logicalloy, and SJS RFID Suite. Section 4 focuses on the following proprietary platforms: iMotion from GlobeRanger, Websphere from IBM, Integration Platform from Manhattan Associates, OAT Foundation Suit from OAT Systems, Sensor capabilities from Oracle, NetWeaver from SAP, SmartChain from Savi Technology, and RFID Intechange from Tibco Software. Section 5 is dedicated to other approaches to middleware design that have been reported in papers and other publications and that are not included under the classification of OSS or Proprietary. Since details of these platforms are not very well known only a brief reference is made to their main characteristics and their innovative features. Emphasis is made on platforms that deal with issues found in mobile RFID deployments. This section provides an initial approach to the work to be developed in WP5. Finally, the last section summarizes our findings.

Section 3 State-of-the-art OSS Middleware Platforms

3.1 Overview

Some OSS RFID middleware platforms are already available even if they do not match completely what customers expect from them. Additionally, they usually do not implement completely or even correctly all available specifications, which are defined by EPCglobal. This is very essential in the design of the ASPIRE middleware, as the compliance to the standards is mandatory to achieve its goal and to support open loop systems. To benefit from existing solutions, we need to have a look at their features but also to the kind of customers who may be interested by those features. As part of our effort towards the best evaluation of the existing OSS RFID middleware, we installed and tested each one of them, as long as it was available. Thus, we were able to have a hands-on experience and, as a consequence, a better understanding of how they operate and also their pros and cons. We paid greater attention on the Fosstrak (previously Accada) and on the UJF software, which are the most complete and thus promising for inclusion in the ASPIRE project.

The following table summarizes the standards compliant systems of the OSS RFID middleware platforms that we were able to test and evaluate. The next section will provide more information about each one of them.

Available Implementations	Applied EPCglobal Standards						
	ONS	EPCIS	ALE	RM	LLRP	RP	TP
Fosstrak (previously Accada)		X	X	X	X	X	
Rifidi			*		X		X
**Singularity		X	X	X			
RadioActive	X	X	X			X	
Mobitec			X	X			X
UJF RFID Suite	X	X	**	X		X	X
*Logicalloy		Partial (Capture)	X				X
SJS RFID Software			X				

Table 2. Standards compliance of OSS RFID middleware platforms

* The Rifidi project can be connected and used with the Logicalloy ALE server

** The UJF RFID Suite uses the Singularity ALE server

3.2 Fosstrak (previously Accada)

URL : <http://www.fosstrak.org>

Module versions:

- Reader module: 0.5.0
- Filtering and collection module: 1.0.2
- EPCIS module: 0.4.2
- Tag data standard: 0.9.0

License : LGPL

Language: Java

Editor/Main contributor : ETH

Platform client : Web/Desktop Client

Platform server : OS portable


Description



Fosstrak is an open source RFID prototyping platform that implements the EPC Network specifications (see also [1]). It is intended to foster the rapid prototyping of RFID applications and to accelerate the development of an Internet of Things. The Fosstrak platform consists of three separate modules: the reader, the filtering and collecting middleware and the EPC information service (EPCIS) module. These modules implement the corresponding roles in the EPCglobal Network and have been identified by the Aspire partners as a strong candidate for inclusion in the implementation of the Aspire middleware.


Features

- Complete implementation of the EPCglobal defined standards
- Java interface that hides communication with a reader instance
- Configuration engine that allows the developer to specify a reader configuration in a configuration file
- Java interface that hides communication with ALE server

Strong and weak points

 Complete implementation of the EPCglobal protocol stack

  Medium sized development community

 Currently in alpha version

Compliance : EPCglobal

Architecture overview

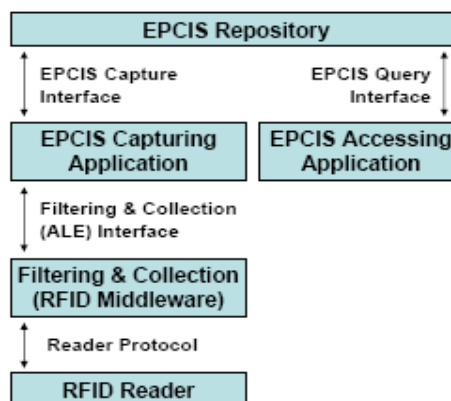


Figure 4 Fosstrak architecture

Evaluation

- community : Good: Medium sized development community
- use case coverage : Does not apply as it is a library implementing a specification
- extensibility: Excellent: It is a library that can be utilized in every third party application
- set-up costing : There is no cost acquiring the library. There is a cost in the development effort required to build an application that uses this library.

3.3 RiFiDi

URL : <http://www.rifidi.org>
Version: 1.4
License : LGPL
Language: C, Java

Editor/Main contributor : -
Platform client : OS portable
Platform server : OS portable







Description

The mission of the Rifidi project is to build a complete open source RFID hardware emulator. A hardware emulator is defined as software that mimics a hardware platform. They are usually used for development of applications and are especially popular in the embedded development industry. Rifidi's goal is to take the intangibles out of RFID development and to make Rifidi the First and Best tool for testing and developing RFID systems.

Features

- LLRP Virtual Reader Support
- Eclipse Based IDE
- Web Services Based Reader Emulation Engine
- Wizards for configuration

Strong and weak points

-  Edge Server Compatibility
-  Tag Streaming Utility to mimic the flow of RFID data
-  Alien ALR 9800 Gen 2 Reader Emulation
-  Lightweight XML-RPC Reader Engine
-  Too focused on the physical layer
-  Currently only available in beta version

Architecture overview

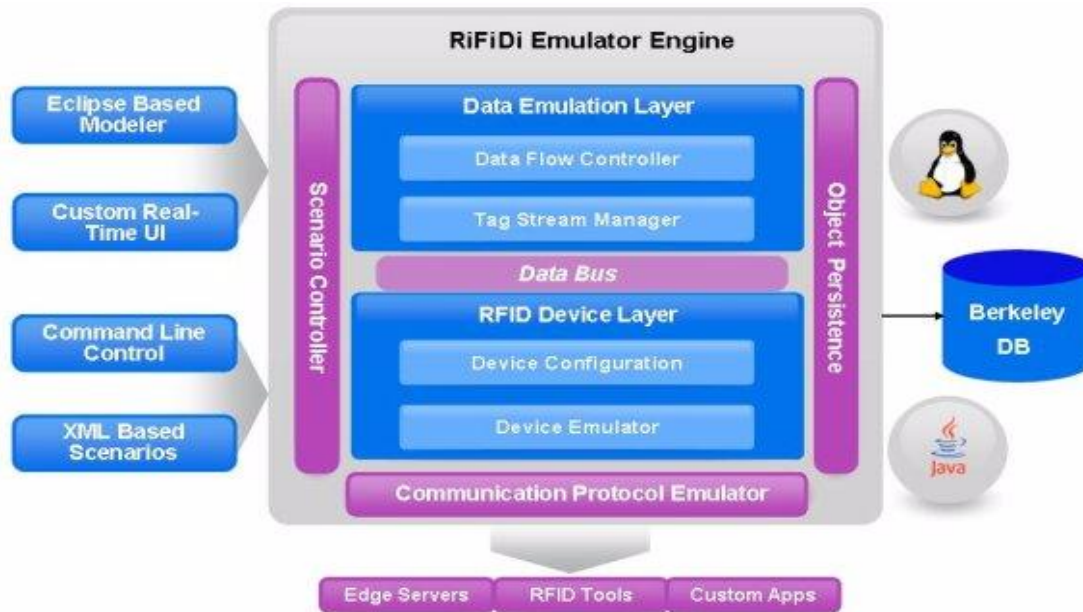


Figure 5 RiFiDi architecture

Evaluation

(poor/correct/good/excellent)

- community : correct
- use case coverage : poor
- extensibility : good
- set-up costing : poor

3.4 Singularity

URL : <http://singularity.firstopen.org/>

Version: 1.0 M2

License : Apache licence 2.0

Language: Java

Editor/Main contributor : firstopen.org

Platform client :

Mobile/Desktop/Web Client

Platform server : OS portable

Description


Singularity provides an open source EPC-IS that supports the EPCglobal™ specifications, as well as enables successful integration of EPC related information into the enterprise while the Middleware provides RFID/Sensor device and event management. Singularity has two major components, the Middleware and EPC Information Service (EPC-IS). The main goal of Singularity is to accelerate the evolution and adoption of RFID solutions. RFID Middleware and EPC-IS provides a platform that reduces entrance barriers, as well as

provides a base to allow commercial companies to accelerate their product offerings.

Features

- Distributed fault tolerant service component architecture Support X RFID readers.

Strong and weak points

 Currently only available in beta version

Compliance : EPC Global

Architecture overview

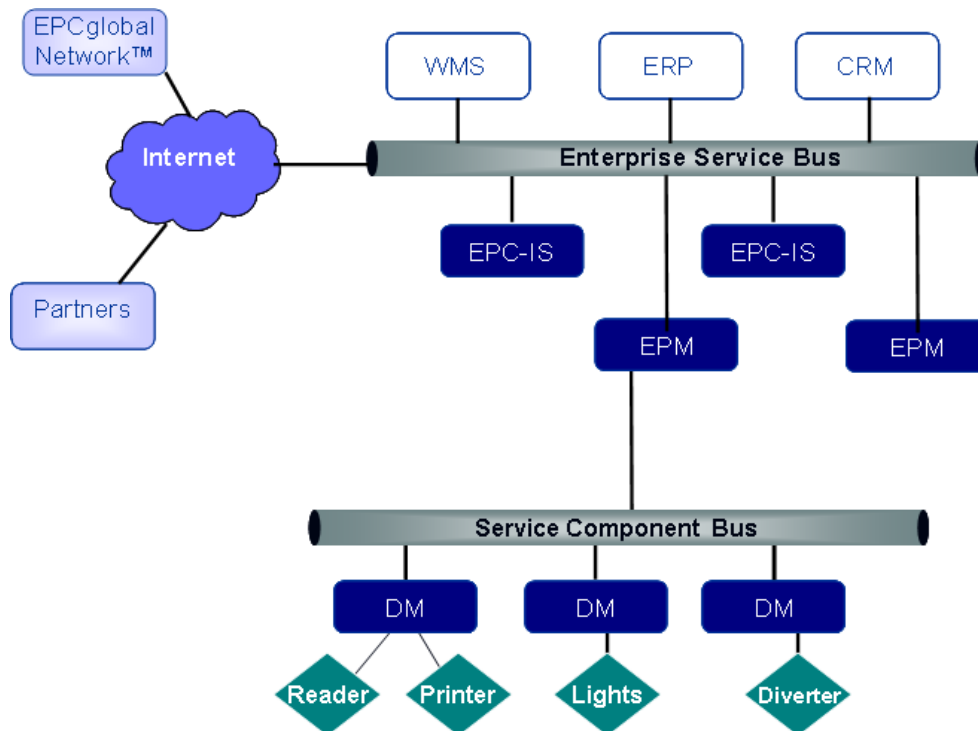


Figure 6 Singularity architecture

Evaluation

(poor/correct/good/excellent)

- community : poor
- use case coverage : poor
- extensibility : correct
- set-up costing : good

3.5 Radioactive Foundation

URL : <http://www.radioactivehq.org>
Version: 1.0
License : Apache licence 2.0
Language: Java

Editor/Main contributor : Radioactive
foundation
Platform server : OS portable

Description

The RadioActive Foundation is a group of projects with a single goal: is to provide a consistent suite of open source applications related to RFID. Its main objectives are:

- To develop a suite of RFID software that is needed to be in full compliance of the EPC standards and various other standards.
- To promote the use of RFID technology by providing a low cost access to RFID software.
- And to be able to compete and surpass any commercial product while being 100% open source and free

The three main RadioActive projects fall into three major groups:

- **Neutrino**

The Neutrino project is a set of components used for exchanging EPC related data between enterprises. It includes an implementation of the ONS, EPC-IS and DS (not yet released) standards. Its primary purpose is to communicate higher level "business events" which means it will rarely be used by middleware apps. Neutrino can even be used in deployments where no RFID reader is needed.

- **Fusion**

The Fusion project is a generalized middleware system that takes RFID (and other sensor) input and adds business contexts necessary to add meaning to that raw sensor input. It implements the ALE, EPC-IS and Reader Protocol standards.

- **Graviton**



The Graviton project implements the hardware sensor layer of an RFID deployment. It contains a reader simulator, an implementation of the Reader Protocol and Reader Management standards.

Features

- Contains a configuration, policy and management interface.

Compliance : EPC Global

Strong and weak points

-  Very limited activity
-  No available release

Architecture overview

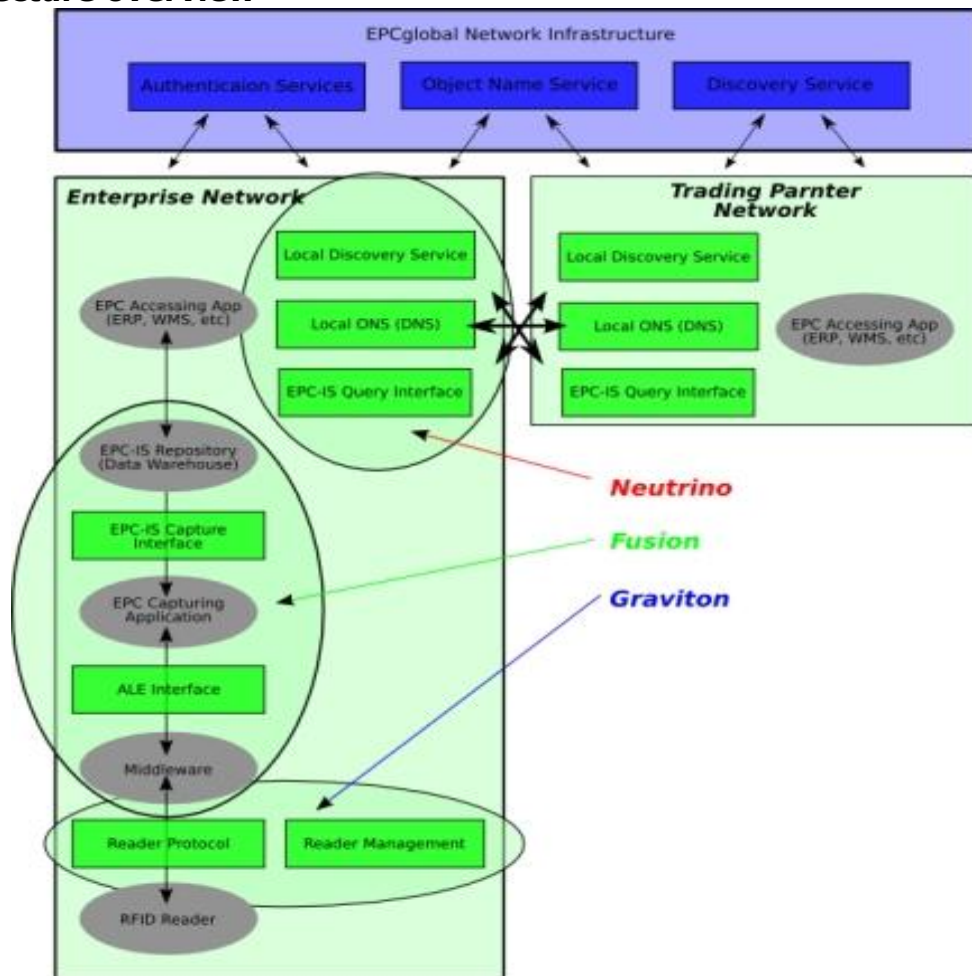


Figure 7 **Radioactive architecture**

Evaluation

(poor/correct/good/excellent)

- community : poor
- use case coverage : poor
- extensibility : unknown
- set-up costing : poor

3.6 Mobitec (CUHK RFID Middleware)

URL : <http://mobitec.ie.cuhk.edu.hk/rfid/middleware/project.htm>
 Version: 1.0
 License : No information
 Language: Java

Editor/Main contributor : Mobile Technologies Centre (MobiTeC)
 Platform client : Desktop
 Platform server : OS portable

Description

CUHK RFID System 1.0 is flexible and cost-effective software complying with EPCglobal middleware specifications. It follows the architecture framework specification of EPCglobal and the Application Level Events (ALE) Specification, Version 1.0. CUHK RFID System 1.0 provides a standard ALE interface for user applications to access the RFID network. The ALE interface is extended to support reading and writing of the tag memory. RFID readers can be connected to the server running this middleware through IP network and RS-232 adaptors. Through management console of CUHK RFID System 1.0, all readers in the RFID network can be configured, controlled, managed and monitored. User applications can be easily developed and integrated with the middleware system.

Features

- The ALE interface is extended to support reading and writing to tag memory
- It supports EPC and ISO tag standards
- The middleware also support CUHK CuTag reader and CuBadge reader which are using active tags
- All devices connected to the middleware server can be managed by the management console (is web-based easy-to-use graphical interface for the administrator to configure the system and monitor work status of all system components)

Strong and weak points

- + Device adaptors
- No source code available

Architecture overview

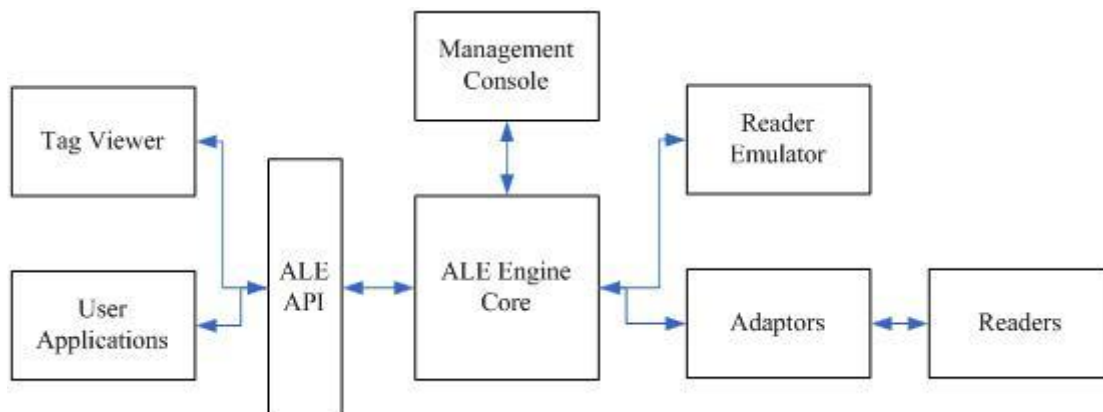


Figure 8 Mobitec architecture

Evaluation

(poor/correct/good/excellent)

- community : poor
- use case coverage : poor
- extensibility : poor
- set-up costing : No information

3.7 RFIDSuite

URL : -
Version: -
License : LGPL
Language: Java

Editor/Main contributor : LIG laboratory,
UJF
University
Platform server : OS portable

Description

RFID Suite is globally architected on the Edge-Premise-Server model that embeds ALE, EPCIS and ONS servers in a single software suite.

Features

- EPCIS server
- ALE server
- ONS server

Compliance : EPCglobal

Architecture overview

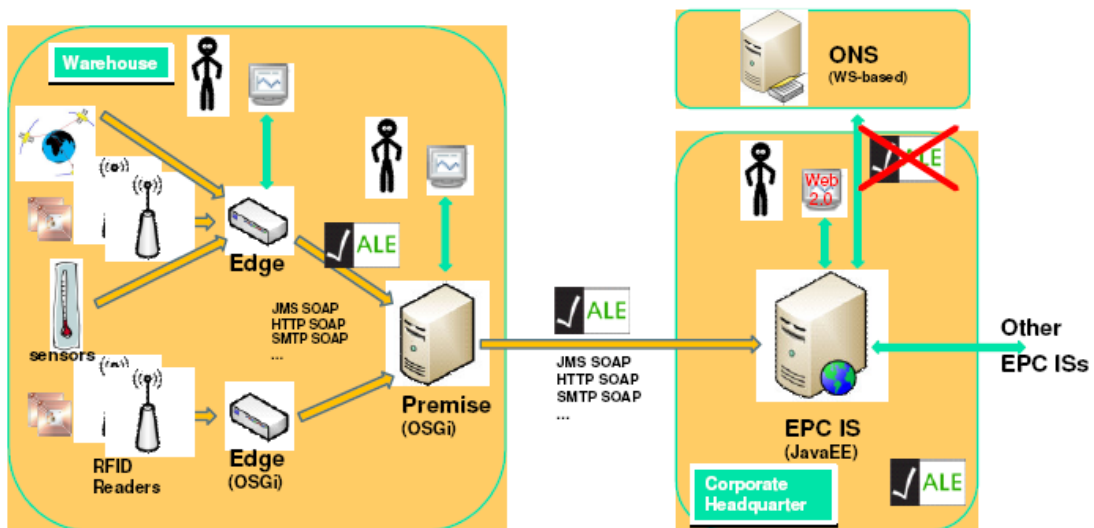


Figure 9 UJF RFIDSuite architecture

Evaluation

(poor/correct/good/excellent)

- community : poor
- extensibility : correct
- set-up costing : undefined

3.8 Logicalloy

URL : http://www.logicalloy.com/ Version: 1.2.1 License : Sleepycat / flexible OEM Commercial License Language: Java	Editor/Main contributor : LogicAlloy Inc. Platform server : OS portable
---	--

Description

ALE Server is a high performance, easy to use, cost effective middleware solution that eases the integration of RFID hardware with existing business systems. Based on EPCglobal architecture [2] and standards [7], it empowers your organization to quickly comply with mandates without making costly changes to your existing business systems.

ALE Server comes with built in hardware simulation which allows simulating the usage of EPCglobal GID-96 tags using your EPCglobal assigned General Manager Number and internally assigned object class.

Features

- High performance ALE server
- EPC-IS Integration
- Multiple notification channels
- ThingMagic Reader Support
- Web-based administration console



Compliance : EPCglobal

Evaluation

(poor/correct/good/excellent)

- community : poor
- use case coverage : good
- extensibility : good
- set-up costing : good

Strong and weak points

-  Simple configuration and management tools
-  Integration with existing business systems

3.9 Sun Java System RFID Software

URL : <https://sun-rfid.dev.java.net>
Version: 3.0
License : Sun Public Licence
Language: Java

Editor/Main contributor: LogicAlloy Inc.
Platform: Windows 2003, Solaris, Red Hat
Linux

Description

The Sun RFID Software integrates RFID devices into Java EE environments. It provides infrastructure to integrate, provision and manage readers and printers and it additionally routes business-level RFID events to backend applications. The RFID Software consists of three modules: the Event Manager, the Information Server and the Management Console. The RFID Event Manager processes streams of tag data coming from networked readers. It performs the filtering, aggregation, and counting of tag ID's. It also connects to back-end systems. The RFID Information Server is a Java EE application that serves as the interface for capturing and querying persistent tag-related data. The information stored includes both tag ID's and the attributes needed to map them to business events. The RFID Management Console is a web-based application that gives an operational view of readers and the Event Manager. It enables administrators to modify system parameters in a running environment.





Working together the Sun RFID Software components seamlessly tie edge events from RFID devices into end-to-end business solutions.

Features

- EPCglobal compatible ALE server
- Development kit available
- Possibility to use with JCAPS through RFID JCAPS

Compliance : EPCglobal

Strong and weak points

-  Simple configuration and management tools
-  Integration with existing business systems
-  Supports sensor data
-  Works best with other products of SUN

Architecture overview

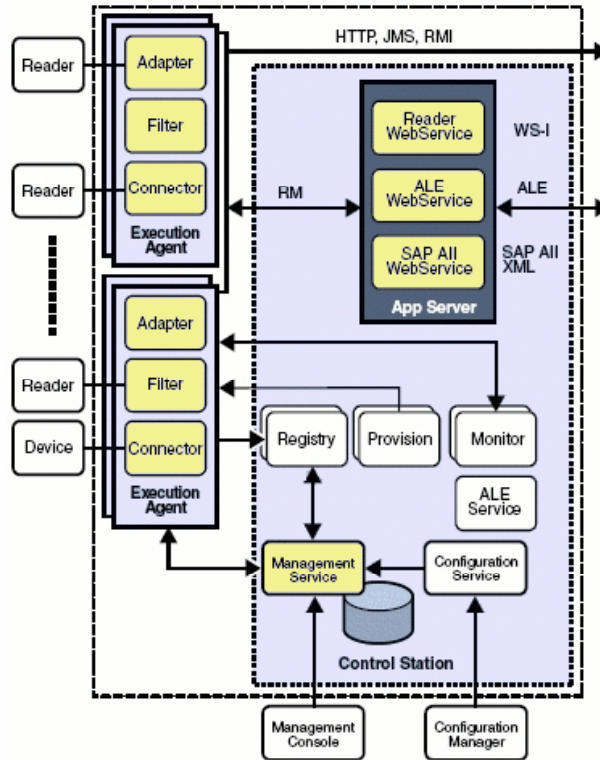


Figure 10 SJS RFID Software Architecture

Evaluation (poor/correct/good/excellent)

- community : good
- use case coverage : fair
- extensibility : good
- set-up costing : good

3.10 RFID Library

3.10.1 RFID-perl

URL :	Editor/Main contributor : This software is
http://www.eecs.umich.edu/~wherfid/code/rfid-perl/	an outgrowth of the Whereabouts project
Version:	at the University of Michigan. It was
License : University of Michigan's standard	primarily written by Scott Gifford.
license	Platform client :
Language: Perl	Platform server :

Description

This software allows the writing of independent code from types and brands of RFID readers and to facilitate the writing of new drivers.

For that, it provides an interface to RFID readers with common code to the various readers. The drivers of Matrics and Alien readers take 1000-2000 lines of codes. Writing new drivers should take a week and a similar amount of code. This software also supports EPC tags and code is using a tag interface.

Features

- RFID::Reader - Abstract class for a RFID reader
- RFID::Tag - Abstract class for a RFID tag object
- RFID::EPC::Tag - This class implements an EPC tag based on RFID::Tag. It allows tags to be created based on the fields of the various EPC tag types, and allows tag IDs to be parsed into their EPC components.
- RFID::Matrics::Reader - Abstract class for a Matrics RFID reader
- RFID::Alien::Reader - Abstract class for an Alien RFID reader

Compliance : EPC Global

3.10.2 RFID C library

URL: http://savannah.nongnu.org/projects/rfid/	Editor/Main contributor :
Version: 1.10	Platform client :
License : GNU General Public License v2 or later license	Platform server :
Language: C	

Description

RFID C library is a set of functions which allows the dialog with RFID device. It provides tools that can be used in a program which is notified when tags are within range of readers and to read or write data on RFID tags.

Supported Readers:

Series 6000- HF-I RFID Evaluation Kit (RI-K10-001A) product
Inside Contactless product range

Supported Tags:

All ISO-15693-3 conformant RFID tags

3.10.3 RF-Dump

URL : http://www.rfdump.org/	Editor/Main contributor : Lukas Grunwald at RFDUMP.org
Version: 1.5	Platform client :
License: GPL Licence. The Java application requires certain additional libraries and packages, some of them published under different licenses.	Platform server :
Language: C	

Description

RFDump is a backend GPL tool which can detect RFID tags with interoperability with RFID ISO readers. It allows to display and modify the user data of a tag and to show its meta information: tag ID, tag type, manufacturer, etc.

Supported Readers:

ACG Multi-Tag Reader or similar card reader hardware.

Section 4 State-of-the-art Proprietary Middleware Platforms

4.1 Overview

Competition is high in the market of proprietary RFID middleware, although not single solution is yet leading it. The enormous market potential has attracted leading technology vendors such as IBM, Oracle, Microsoft, SAP, Sun Microsystems and HP. However, some vendors have already retired from this competed market without being able to profit from it. Reasons include the lack of common standards and clear business benefits (in particular when infrastructure costs are driven by proprietary technologies), and privacy and security issues around RFID. Other vendors such as ConnecTerra and RedPraire merged their products and/or evolved into complete supply chain systems. Moreover, research on RFID middleware is scarce and the only reliable available report dates from 2004, when Forrester Research first studied the market for RFID middleware¹.

It is therefore not the intention of this document to undertake thorough research on the market of proprietary RFID middleware, which is by definition opaque because vendors do not openly and objectively publish their characteristics and price and have opted for low-profile PR since privacy and security issues started to undermine the image of their customers. Conversely, the purpose of this document is to provide a glance on the potential proprietary competitors for ASPIRE, and their relative competitive advantages and disadvantages. Consequently, this section focuses on the study of the most relevant proprietary RFID middleware products available on the market. Moreover, existing proprietary RFID middleware solutions are not considered competition for ASPIRE because its core product will be Open Source Software and distributed under a royalty-free license.

Importantly, many vendors have migrated their middleware solutions to Service Oriented Architecture (SOA), which achieves similar results but based on standard building blocks such as OSGI etc.

The following table summarizes the characteristics of the selected products and their comparison with ASPIRE's proposed characteristics (unconfirmed: based on claims made by their providers and on preliminary research):

¹ Leaver, Sharyn. Evaluating RFID Middleware, Forrester Research, August 13 2004.

Product	Vendor	ASPIRE proposed features*				
		Lightweight	User friendly	Affordable	Programmable	Privacy-friendly
iMotion	GlobeRanger	Y	Y	Y	N	N
WebSphere	IBM	Lightweight version available	N	Low-cost version available	Y	N
Integration Platform	Manhattan Associates	N	Y	N	N	N
OAT Foundation Suit	OAT Systems	Y	Y	Y	N	N
Sensor capabilities	Oracle	N	N	N	Y	N
NetWeaver	SAP	N	Y	N	Y	N
SmartChain	Savi Technology	N	Y	N	Y	N
RFID Interchange	Tibco Software	Y	N	Y	N	N

Table 3. Comparison of proprietary middleware platforms

* These middleware platforms are evaluated with regards with the expected features of the future ASPIRE solution

The following sections only focus on 3 of these solutions (i.e. OAT Systems, IBM and SAP). These three middleware platforms have been shown to be leading the current RFID market. We detail their characteristics and history.

Product	Vendor	ASPIRE proposed features*				
		Light-weight	User friendly	Affordable	Programmable	Privacy-friendly
iMotion	GlobeRanger	Y	Y	Y	N	N
WebSphere	IBM	light-weight version available	N	Low-cost version available	Y	N
Integration Platform	Manhattan Associates	N	Y	N	N	N
OAT Foundation Suit	OAT Systems	Y	Y	Y	N	N
Sensor capabilities	Oracle	N	N	N	Y	N
NetWeaver	SAP	N	Y	N	Y	N
SmartChain	Savi Technology	N	Y	N	Y	N
RFID Interchange	Tibco Software	Y	N	Y	N	N

Table 4. Comparison of proprietary middleware platforms (Synthesis of each tool)

* These middleware platforms are evaluated with regards with the expected features of the future ASPIRE solution

The following sections only focus on 3 of these solutions (i.e. OAT Systems, IBM and SAP). These three middleware platforms have been shown to be leading the current RFID market. We detail their characteristics and history.

4.2 OAT Systems and OAT Foundation Suit²

OAT Systems was one of the first software companies to offer RFID middleware. The company was founded in 2001 by Sanjay Sarma, their Chief Scientist, and Prasad Putta, responsible for business and technical strategic partnerships. Sarma co-founded MIT's Auto-ID Centre and has served as its Chairman of Research. OAT's headquarters are in Waltham, Massachusetts, USA.

The goal of most RFID deployments is to integrate data from multiple facilities or across trading partners to provide a clear view of a business's operations, inventory and asset movements. Therefore the key to being successful in translating the real-time world of RFID to the business world of ERP is a successful RFID middleware product. This has to be easy to use as well as having a flexible architecture allowing cost effective implementation of the software. Once you have this foundation then you can exploit the additional data that is generated by RFID.

OAT Systems has come a long way since 2004 and their first analyst review which highlighted some issues around enterprise-class integration and data management capabilities. Those issues have been solved and OAT Systems has used their experience of implementation to enhance the basic functions of RFID middleware so that it is easier to implement through the use of its preconfigured Use Cases and, in addition, the architecture topology for implementation is flexible so that the software can be implemented on a great variety of alternatives. On top of this OAT Systems is now delivering pre-configured application solutions that leverage its middleware.

Bloor sees OAT Systems as one of the scene-setters of the RFID Middleware market and therefore should be one of the first names on a selection list.

URL : <http://www.oatsystems.com/>

Modules:

- OATenterprise: provides centralised data management for analytics and visibility across the enterprise.
- OATxpress: is the runtime part of OAT Foundation Suite and supports RFID data management and real-time alerts for a single edge unit.
- **OATdesigner**: is a graphical tool which allows users to modify and create new business scenarios.
- **OATdevice manager**: OATdevice manager manages large-scale and dense device deployment scenarios, including non-RFID devices for alerts and exception handling.

² Sources:
Simon Holloway, Bloor Research, 13 Feb 2008.
OAT Systems website.

Features

- **Built-in adapters** for trading partner data.
- **Consistency engine** uses inference logic and business context to turn noisy, incomplete, and error-laden data into a clean view of a business's RFID data.
- **"Best practice" scenarios** for tagging, pallet building, shipping, and receiving goods come "out of the box" and enable organizations to reduce costs and time of deployment
- **Fast/thin deployment modes** enable businesses to optimize deployment architecture and manage readers across large numbers of remote sites (e.g., retail stores) by separating platform components
- **Hardware support for the range of devices** required to run an RFID deployment, including RFID readers, bar-code scanners, PLCs, and label printers
- **Device monitoring dashboard** provides up/down status of RFID hardware infrastructure and sends alerts in the event of performance failure
- **Maintains the EPC number management structure** and becomes the system of record for EPC number management

Compliance : EPCglobal

Architecture overview

The provider does not supply details of its architecture.

4.3 SAP and NetWeaver³

SAP RFID middleware is integrated with the application layer, so it is very difficult to classify and compare with other middleware software.

From the SAP website and whitepapers: « There are two major elements of SAP solutions for RFID, which are described in detail in the following sections:

- SAP Auto-ID Infrastructure facilitates the capture of serialized data from the devices at local sites and provides the business context to turn the data into meaningful business events. SAP Auto-ID Infrastructure commissions, configures, maintains, and translates serial numbers (EPC, UID, and others) as necessary for building the first-level product-information layer. It also communicates with the business applications (SAP or non-SAP) to access the business context information necessary to properly validate and store serial information. Communications are generally handled via the SAP NetWeaver® Exchange Infrastructure (SAP NetWeaver XI) component. SAP Auto-ID Infrastructure includes preconfigured business functionality for inbound receiving, outbound shipment, e-kanban, and more [.]
- Serialized information collected by SAP Auto-ID Infrastructure (or other EPCglobal compatible middleware) is often detailed, and is stored in the auto-ID infrastructure instance that is local to the originating site. Some of this information is routed to the SAP object event repository, where it is available to support applications that require visibility between sites in the enterprise, or between the enterprise and other trading partners, to support a full range of business processes. »

SAP is one of the leading RFID Middleware vendors with significant market presence. On the negative side, the SAP Auto-ID Infrastructure is a closed one and integration possibilities with non-SAP products are limited, particularly with other ERP and SCM systems.

URL : <http://www.sap.com/>

Modules:

- Mapping and rules processor – Messages arriving from devices on the plant or warehouse floor are mapped, based on device location and event type to configurable rules, which determine the business event that has occurred and what sequence of activities to execute.
- Activities – Discrete programs perform a specific action such as validations, updates, and communications required to execute the business process. Activities contain parameters such as exception criteria and thresholds. Activities are based upon the ABAP™ programming language workbench,

³

Sources:

Leaver, Sharyn. Evaluating RFID Middleware, Forrester Research, August 13 2004
SAP website.

which allows a company to modify existing activities and create its own to match its business requirements.

- Routing engine – The interpreted data in SAP Auto-ID Infrastructure is mapped to the relevant business objects inside the SAP Business Suite family of business applications, including the SAP ERP application, to facilitate automation of business processes. Using the mapping functionality of SAP NetWeaver XI, it is also possible to map to processes and data in non-SAP applications.
- Prepackaged, configurable support for business processes – Many business processes are supported by preconfigured content, including inbound receiving, outbound shipments, and RTI tracking. These processes, as described in the previous section, can be deployed out of the box or can be flexibly configured to meet specific organizational requirements.
“Prepackaged and configurable” means that rules and activities are created and packaged in a way to support the processes mentioned above, including all document and status associations.
- Serialized number and format management – Supports encoding and writing RFID tags, including Gen 2 RFID tags
- Operational or perpetual database – Stores site-level EPC serialization data and associated observation and event data, multilevel data aggregation, and associated business data. This repository integrates with the object event repository (see next section) and can be used to facilitate local reporting.
- Integration with business planning and execution applications via SAP NetWeaver XI – Supports preconfigured integration with SAP ERP, as well as the ability to integrate to non-SAP back-end enterprise resource planning (ERP) applications
- Analytical reports – Predefined content for the SAP NetWeaver Business Intelligence (SAP NetWeaver BI) component allows for the tracking of a range of critical key performance indicators (KPIs) such as tag read and write statistics or supply chain metrics including cycle times and dwell times.

Features

SAP Auto-ID Infrastructure features a number of important enhancements:

- Service enablement of numerous SAP Auto-ID Infrastructure services to facilitate the development of customized processes that leverage serialization
- Support and integration for the SAP object event repository, SAP enterprise EPCIS technology. The SAP object event repository is an enterprise-level serial number repository for serialization standards such as EPC, UID, and others. Together with SAP Auto-ID Infrastructure, this repository is intended to be the system of record for all enterprise serialized information. The object event repository is based on the core requirements specified by EPCglobal, including the EPCIS capture interface and the EPCIS query interface.
- The generic document interface, which enables easy integration of ERP and other legacy systems’ documents with SAP Auto-ID Infrastructure for the enablement of serialized processes beyond the standard documents – With

this tool, it is possible for the customer to set up configuration to facilitate the download from the ERP system into the auto-ID infrastructure (via SAP NetWeaver XI) of any standard business document, without the need for customizing.

Compliance : EPCglobal, UID and others (configurable)

Architecture overview

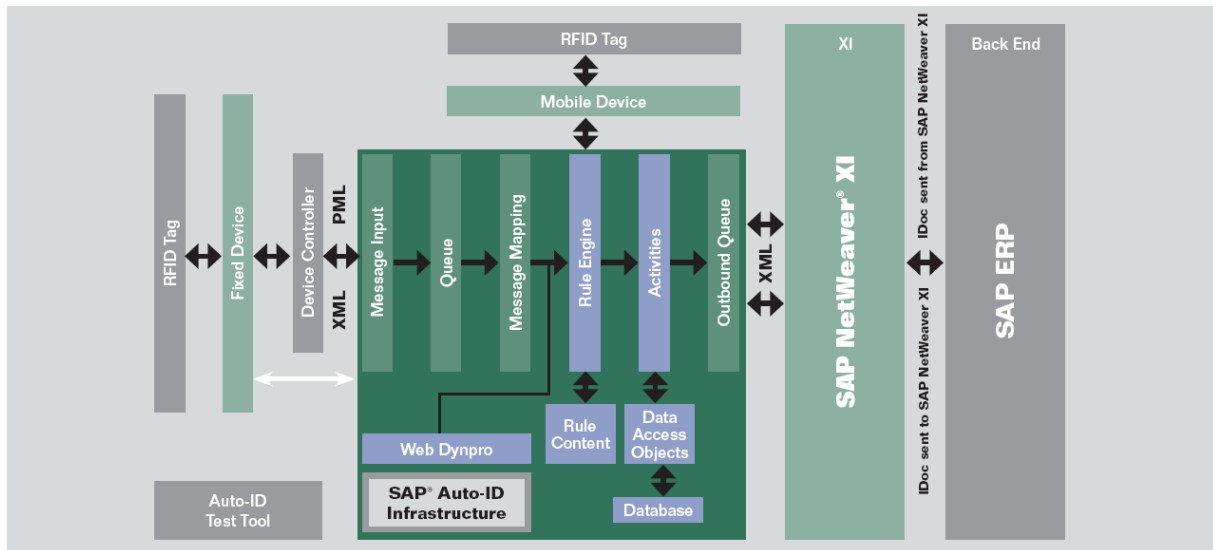


Figure 11 NetWeaver architecture

4.4 IBM and WebSphere⁴

Originally, IBM was leveraging on its mature WebSphere integration platform to provide RFID services. Additionally, it was developing a series of tools to provide RFID transactions with business sense and communicate with the integration. Particularly, IBM extended WebSphere to include an “RFID Premises Services” which is an application platform that performs the functions of the Premises domain within the IBM RFID solution architecture. The Premises Server processes RFID information and events from the RFID readers, controllers and automation equipment of the Edge domain, and provides access to RFID information to the Business Processing Integration domain.

More recently, IBM has integrated its RFID tools in the “IBM PLACES Middleware”, which is Middleware that allows the integration of various positioning technologies and the consolidation of them with ease into a single, managed interface middleware that includes a set of easy-to-use, location-based programming APIs. These type of middleware is defined by IBM as “Location-Aware Service (LAS) is middleware”, or middleware that lets application providers take advantage of location-based services from multiple vendors, while providing application developers with an easy-to-use, yet powerful, application programming interface (API). IBM PLACES (Points-of-interest, Locations, and Asset Catalog for Enterprise Services) Middleware is an implementation of LAS that provides indoor, location-aware services implementation that allows real-time positioning using various radio frequency (RF) technologies, asset tracking, geocoding, mapping, and directory services from the user's Web application.

Other RFID tools provided by IBM include the “RFID Integrated Solution Enablement”; the “RFID Device Development Kit”; and the “Application Level Events (ALE) Preview for RFID” (see definitions below).

For this reason, IBM’s RFID Middleware solutions are not “turn key” and rather oriented toward the development of tailor solutions. For instance, IBM PLACES Middleware provides a set of front-end Java™-based standard APIs that developers can call to perform location-aware functions such as real-time coordinate positioning, geocoding, or mapping. Developers can focus on the application without worrying about tedious implications. On the back end, IBM PLACES Middleware consolidates various positioning technologies using a Java-based, configurable adapter framework. It also integrates various location warehouses.

URL : <http://www.ibm.com/Modules/tools>:

- Theseos Query Engine for Traceability Networks: A demonstration of a technology that allows supply chains to track items quickly and efficiently

⁴ Sources:
IBM website.

without compromising confidentiality and control of participating organizations.

- IBM PLACES Middleware: Middleware that allows the integration of various positioning technologies and the consolidation of them with ease into a single, managed interface; middleware includes a set of easy-to-use, location-based programming APIs.
- IBM PLACESadmin: A visual Web interface for creating indoor, location-based Web applications.
- RFID Integrated Solution Enablement: A Model-driven development and life-cycle management system for embedded solutions.
- RFID Device Development Kit: A set of tools and techniques for interfacing RFID readers and other related hardware into IBM's RFID middleware solution. (This is an ETTK technology.)
- Application Level Events (ALE) Preview for RFID: An implementation of the EPCglobal Filtering and Collection Work Group's ALE (Application Level Events) specification. (This is an ETTK technology.)

Features

- Lightweight version available
- Configurable, flexible, possibility of tailoring
- Multiple choices available
- Openness of interfaces

Compliance: EPCglobal and others (configurable).

Architecture overview

Since IBM offers many different tools for RFID and these can be combined in different ways, there is no single architecture for this provider. However, the Websphere Integration Reference Architecture is provided:

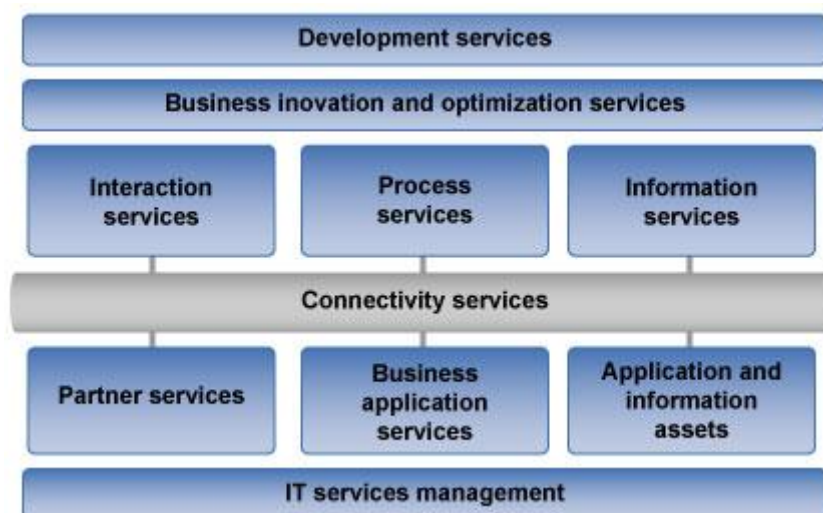


Figure 12 WebSphere architecture

Section 5 Other approaches to RFID middleware design

5.1 Trends in middleware platform design

This section provides a brief summary of other approaches and trends in RFID middleware design that either do not fall in the categories presented in previous sections or are only mentioned in journal or conference proceedings. In general, these works address the same gaps in middleware development than those found in the previous two sections. These gaps include the following:

- a) Middleware platforms need to address in a better way their integration with upper layer back end servers at the business level
- b) Middleware platforms need to devise innovative schemes for manipulation of data due to the huge amount of reads generated by item level applications.
- c) Middleware platforms need to cope with the unreliable nature of the underlying physical layer reading processes.
- d) Middleware platforms need to solve and tackle the increasing privacy and security threats in RFID solutions
- e) Middleware platforms need to consider the particular features of RFID systems integrated in mobile devices and with resource constrained hardware platforms.
- f) Middleware platforms need to be updated as regards the management of their components and the creation of RFID applications.
- g) Middleware platforms need to integrate better with Internet networking technologies

Addressing the points (a) and (f) above we find the work in [35] where the authors introduce an application level architecture for RFID middleware. The architecture combines a process engine with SCA (Service Component Architecture) and an OSGi component library. The architecture is especially designed to facilitate application development using several business processes modeling tools and concepts. JMX is used for managing the different components of the system. The architecture is reproduced in Figure 16. Details can be found in [35] .

To cope with a wide variety of requirements on middleware platforms, the authors in [32] propose a component-based reconfigurable Middleware (CRRM) that can be adapted to different applications. The layers and the architecture used for implementation are displayed in Figure 13.

Another middleware architecture based on components is presented in [36], where the authors use a combination of OSGi and JBPM (Java Business Process Management) to implement workflow concepts. The architecture proposes the extraction, generation and dynamic management of RFID application components, which enables component reuse and plug & play in the RFID business process, simplifies the reuse of RFID application workflow.

With the aim of providing a high level of scalability to a middleware platform, the authors in [37] present an approach based on a publish/subscribe mechanism

and on SOA (Service Oriented Architecture). The idea on this proposal is to use the loose coupling philosophy of SOA so as to get rid of the one to one relationship between readers and middleware platform. In this way, a large number of readers can be directly introduced in the architecture without needing to change application or architectural design in the middleware. The mechanisms simply consists of using a publish/subscribing mechanism that is in charge of forwarding low level data and upper layer requests to the adequate destination. RFID middleware servers can therefore be easily aggregated or removed from the deployment as required. Component and architectures are shown in Figure 14 and Figure 15.

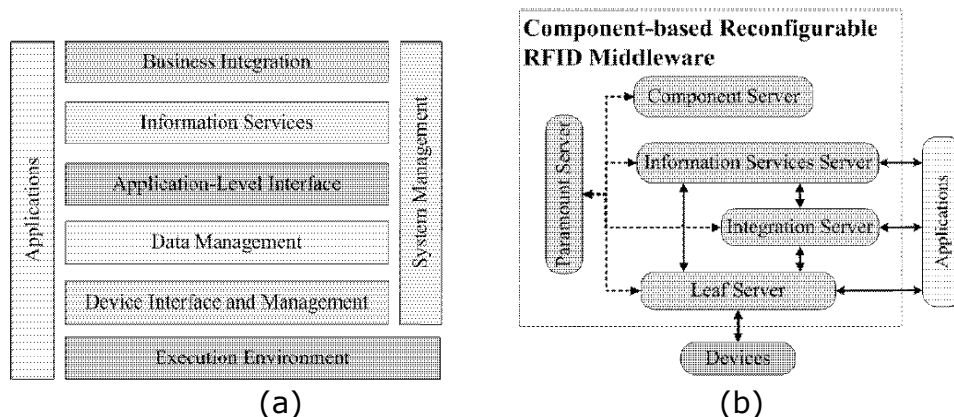


Figure 13 (a) Layers and (b) architecture of CRRM (taken from [32].)

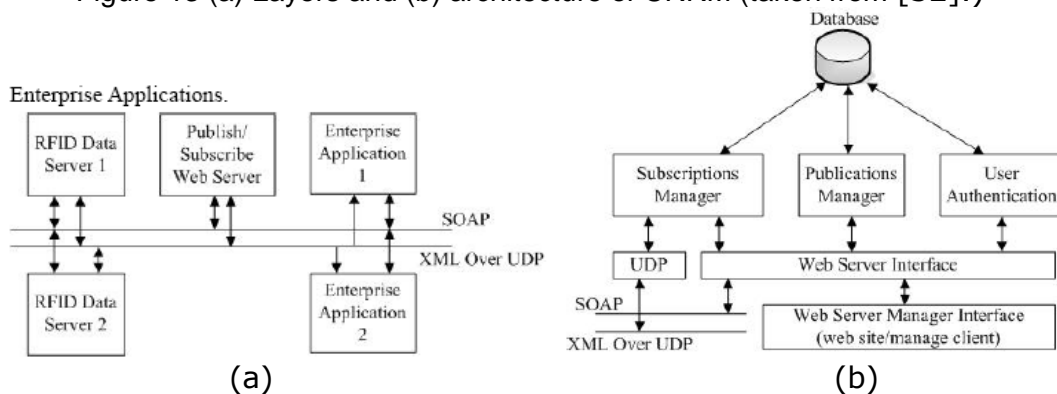


Figure 14 (a) Components of the publish/subscribe system and (b) components of the publish/subscribe server (taken from [37])

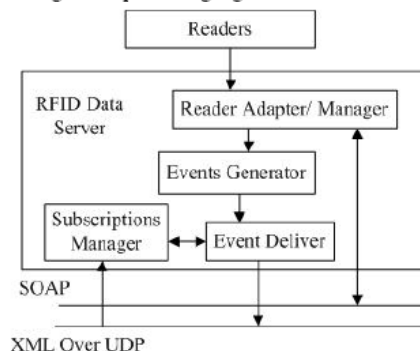


Figure 15 Components of RFID Data server (taken from [37])

Dealing with points (a) and (b) we find the work in [11] where the authors propose a descriptive language and an automated detection method for complex

events in RFID. This work recognizes the need for an appropriate modeling tool of business events with complex operations such as aggregation of items and nesting. The descriptive language is complemented by an automated method to process, detect and optimize the information retrieved by each RFID event.

Using a similar approach, the authors in [19] propose an NFA (non-deterministic Finite automate)-based RFID event detection technique for Complex Event Processing (CEP). The authors address CEP problems over multigranularities RFID streams, i.e. they address cases where tags are grouped in boxes, pallets or single units. Their analysis aims at optimizing the computation complexity while preserving adequate RFID event detection levels.

The work in [18] proposes a new middleware architecture, based on multi-agent software modules, that uses the PASSI methodology to address scenarios with asset management. This architecture is reproduced, for convenience, in Figure 17.

Addressing points (b) and (d), the authors of propose tag search protocols with enhanced security features and with low complexity. These lightweight protocols are mainly intended for passive tags with resource constrained processing complexity. The protocols are based on Linear Feedback Shift Registers (LFSR) and Physically Unclonable Functions (PUF), and mainly prevent attacks from eavesdropper readers and tag or reader spoofing. The complexity of the protocols, as stated by the authors, is no more than 1400 hardware gates on tags.

Another approach to security is given by the authors in [17] where they use an access control scheme as a security layer of a reconfigurable middleware platform based on OSGi. The middleware platform is especially designed for providing security in ubiquitous environments. The proposed middleware architecture is displayed in Figure 18.

Further security issues are tackled by the security-enhanced RFID middleware platform proposed in [31]. This platform deploys a novel context aware access control service at the level of the ALE service. The access control scheme prevents from unauthorized users to have access to consolidated data provided by the ALE or FC server. The proposed middleware architecture is displayed in Figure 19, while the access control mechanism and the access control policy are displayed in Figure 20.

Another approach to tackle points (b) and (c) is given by the data cleaning model used by the authors in [15]. The authors propose a virtual spatial granularity concept together with a Bayesian estimation algorithm to cope with false positives and false negative reads. The virtual spatial granularity concept exploits the fact that tags across a supply chain follow similar movement and spatial locations. The algorithm classifies tags according to their spatial movements and, hence, improves their probability of correct detection by estimating their next movement.

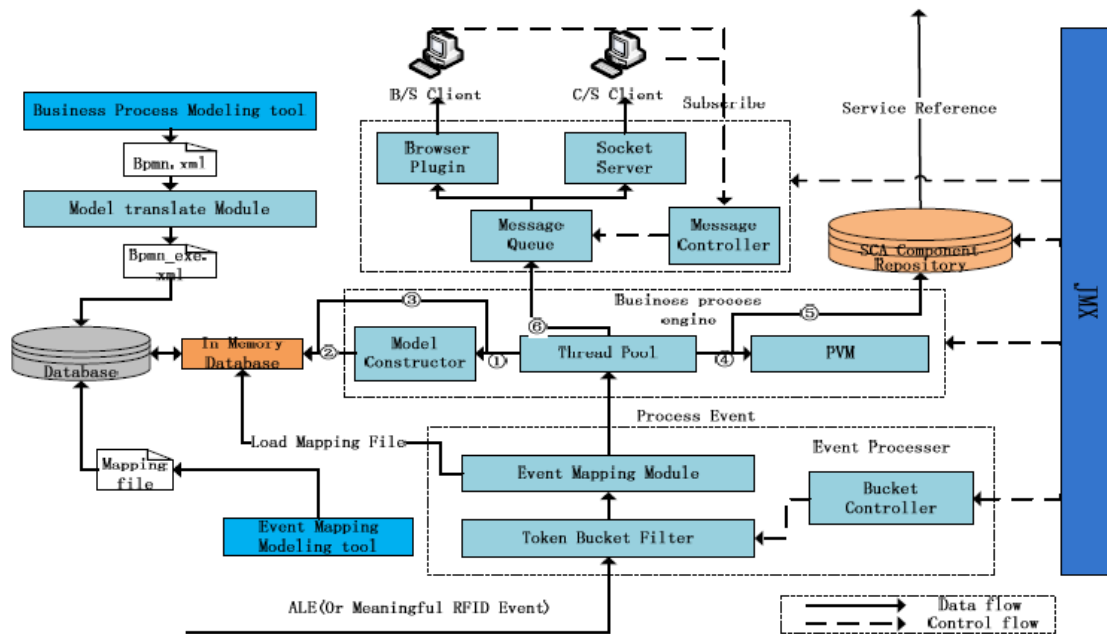


Figure 16 RFID SCA middleware architecture proposed in [35].

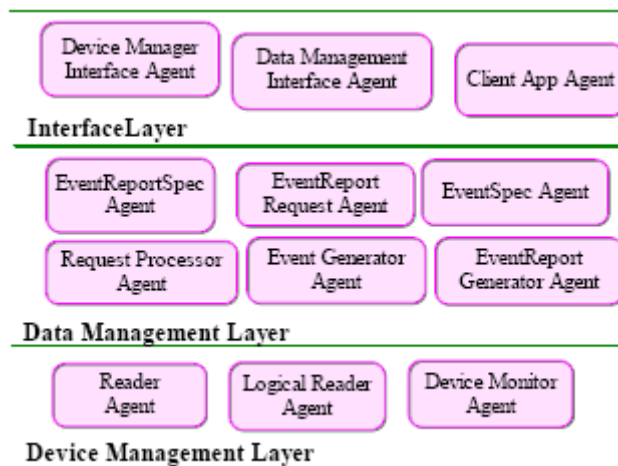


Figure 17 Agent-based middleware architecture proposed in [18][35].

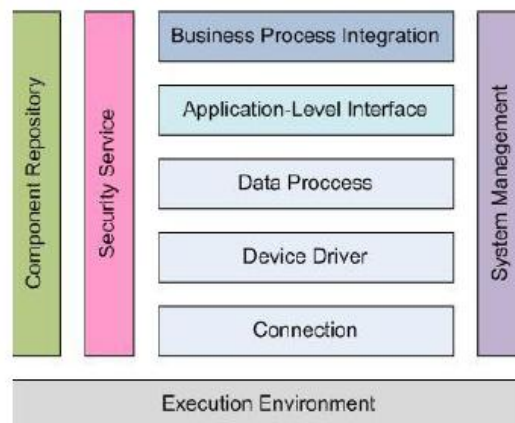


Figure 18 Architecture of a reconfigurable OSGi based middleware architecture with a security layer based on access control for ubiquitous scenarios (proposed in [17])[35].

Yet another approach for improving reading reliability in RFID systems is given by the work in [9] and [26]. The authors have proposed a middleware architecture called RF²ID which is based on the concept of context aware and virtual reader and path abstraction model. Additionally, their design is oriented to organize queries in an efficient manner and provide high levels of reliability and scalability. The main concept consists of creating virtual readers, which consider the unreliable nature of each interrogator, and virtual paths that serve as a higher level abstraction that can identify and follow a tag moving across the environment. A virtual path, hence, can cope with false negative and false positive reads of a tag moving across different virtual readers. The architecture of RF²ID is displayed in Figure 21.

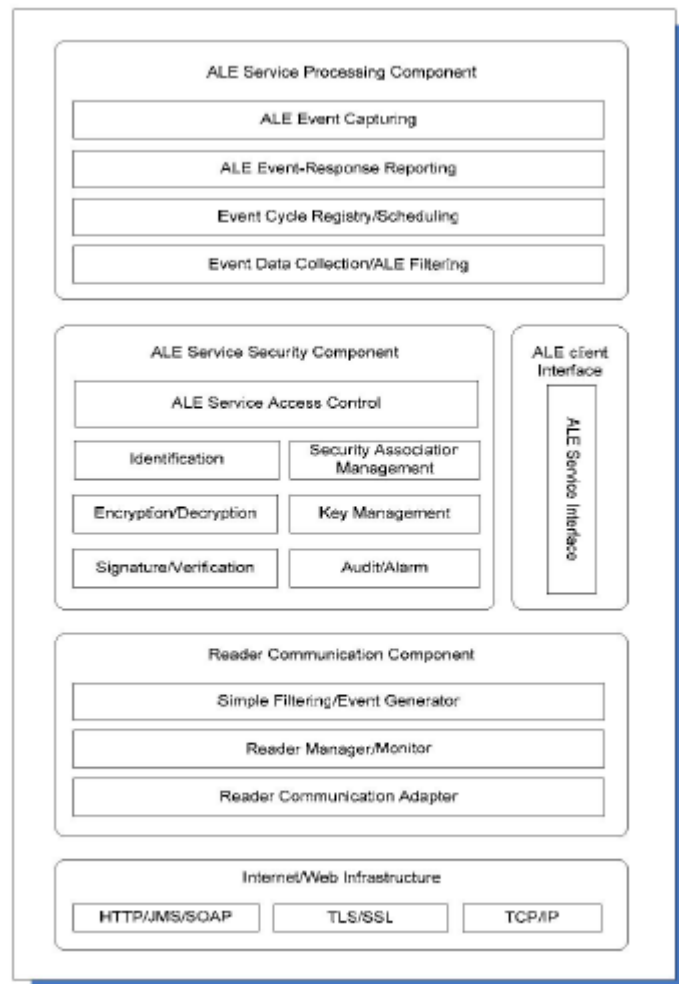


Figure 19 Security-enhanced middleware Architecture with context aware access control (proposed in [31])[35].

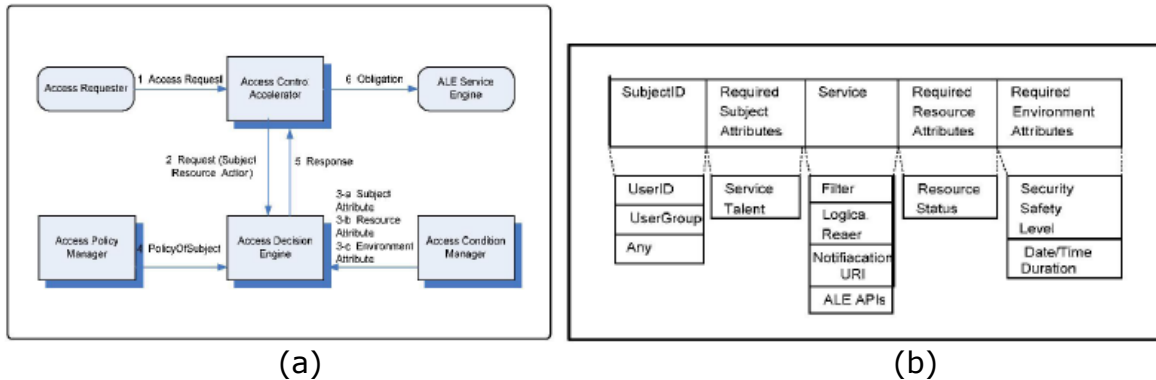


Figure 20 (a) Access control mechanism and (b) access control policy for the security-enhanced RFID middleware platform in [31][35].

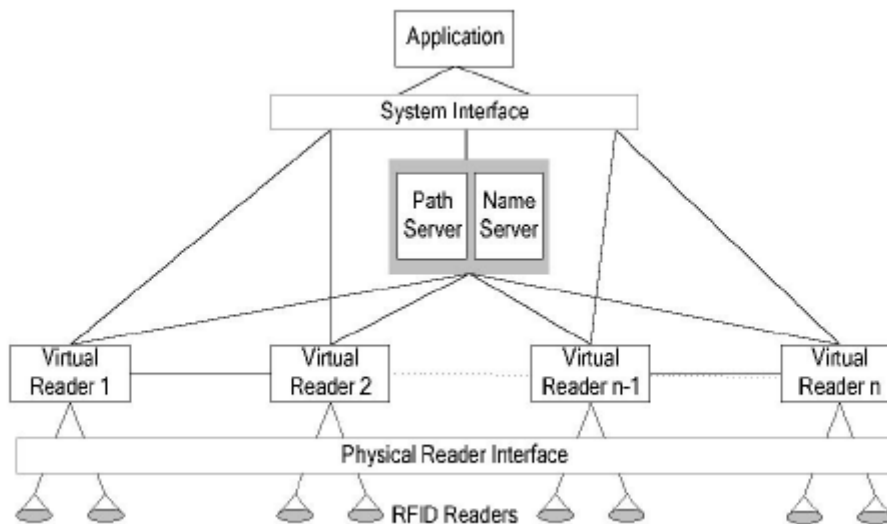


Figure 21 Architecture of RF²ID [26][35].

Another work on data cleaning models is reported in [24], where the authors propose a P2P (Peer to Peer) collaborative model. The model exploits redundancy information that is exchanged between the different nodes across the path of a mobile tag. The authors claim to achieve considerably good amounts of cleaned data with low complexity figures.

Coping with point (f), the authors in [13] propose a middleware platform for easy application development called FlexRFID. The middleware architecture is especially focused on the business layer and on facilitating both application development and device management. The architecture is displayed in Figure 22.

The increasing interaction of RFID with Internet is another fact middleware platforms must consider [38]. This interaction further increases security and privacy issues. To cope with these problems the authors in [28] propose and interoperable internet scalable security framework (IISS) for RFI networks. ISS performs authentication and authorization based on an aggregation of business rules, enterprise information, and RFID tag information.

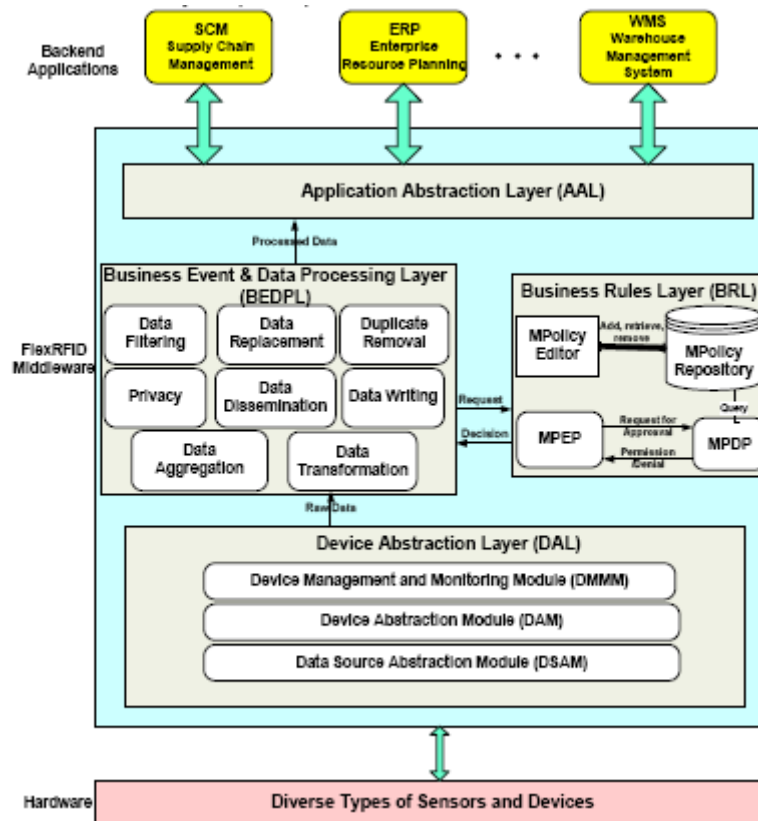


Figure 22 FlexiRFID middleware architecture in [13][35].

5.2 Integration of RFID readers into wireless mobile telecommunication networks

The purpose of this section is to give an overview of how the RFID platforms and related technologies can be integrated into the mobile telecommunications infrastructure, and to present the current developments, challenges and possibilities given by this integration.

Conventional RFID consists of mobile tags which respond to requests from fixed readers by sending relevant data. However, in recent years the concept of fixed reader has changed, as this functionality can be now embedded or somehow attached to a mobile phone or portable device with access to public telecommunication networks. This completely new approach is different from current implementations of ordinary RFID: now the readers are mobile and the tags are fixed, instead of the other way around. M-RFID (Mobile-RFID), as it has been called, has some advantages over conventional RFID: no wires to fixed readers are needed and several mobile readers are enough to cover larger areas than those covered by fixed readers.

Integrating RFID capabilities into mobile or portable devices extends the use of RFID technology beyond the typical supply chain management into areas like customer service, marketing or brand management. Services that use the

information stored on RFID tags and communicate with peer objects will help to create an environment of smart objects which can sense, for example, environmental parameters such as temperature, position and acceleration. Mobile phones provide the platform for user communication with such smart objects. Besides, the usage of the public telecommunication infrastructure enables RFID applications to fulfill the requirements of those SMEs whose infrastructure is distributed across different geographical locations, which matches the ASPIRE objectives.

Among the different applications that arise from the integration of RFID into mobile networks we can find:

- Information retrieval
- Data transmission
- Automated messaging
- Presence indication

Mobile payment

Various consortiums like the Near Field Communication Forum (NFC), European Telecommunications Standards Institute (ETSI) and EPCglobal are developing M-RFID solutions. For example, the NFC standards describe the transport protocol and data exchange methods as well as mechanisms for data collision control during initialization, among other specifications. NFC operates at data rates of 106 kbits/s and 212 kbits/s, slightly lower than Bluetooth.

RFID has been identified as an enabler for ubiquitous computing, i.e., the integration of computation into the environment: any device, anytime, anywhere. RFID enabled mobile phones could represent the first step in this direction. In this context, the issue of RFID-IPv6 mapping is another important field of study, as many active RFID tags of the future will have to be integrated into communication networks, using an identifier that most probably will be or have a direct relation with an IPv6 address.

Various technical implementations for RFID in mobile telecommunications already exist. For example, Nokia has developed the first RFID-enabled cell phone in cooperation with VeriSign. The Nokia 5140 RFID Kit, a GSM phone with RFID reading capability was introduced in March 2004 [6].

In the context of the ASPIRE project, a generic architecture for the integration of the RFID infrastructure into wireless networks is shown in the figure below:

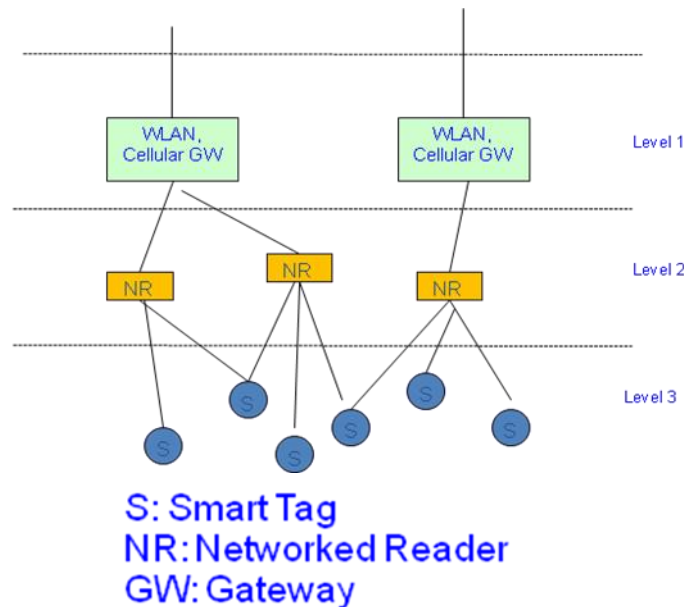


Figure 23: Generic wireless network architecture for the ASPIRE project

One of the future goals of research within the ASPIRE project is to define the functionalities of the wireless network architecture previously shown and then specify the communication interfaces between the physical layer hardware and the network. A diagram showing the communication interfaces to be defined within the ASPIRE project is displayed in Fig.2

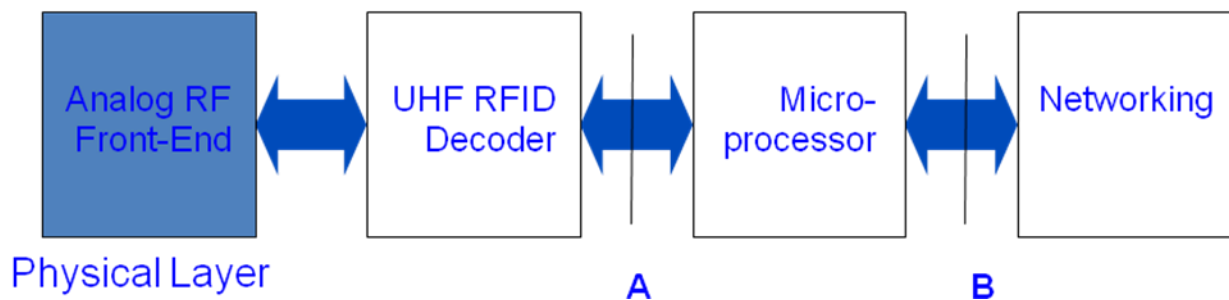


Figure 24: Communication interfaces

5.3 Middleware solutions for mobility applications

This subsection provides a brief review of middleware solutions for mobility applications. We recall that, as part of the activities of WP5, ASPIRE will also tackle different networking and mobility scenarios, especially those with low cost and resource limited readers. The summary provided here is by no means exhaustive, as it only aims to present relevant examples of the different research trends identified so far in the project.

Recognizing the resource constrained features of Mobile RFID applications, the authors in [32] have proposed an OSGi Extension service for Mobile RFID

applications. The extension has been called Mobile RFID Service Extension (MRSE), and it consists of a component-based, service-oriented architecture that provides developers with the possibility of constructing customized mobile RFID middleware with different requirements. This provides with a flexible method to create RFID middleware tools tailored to the resources of the mobile reader. The layers and the architecture of MRSE are displayed in Figure 25.

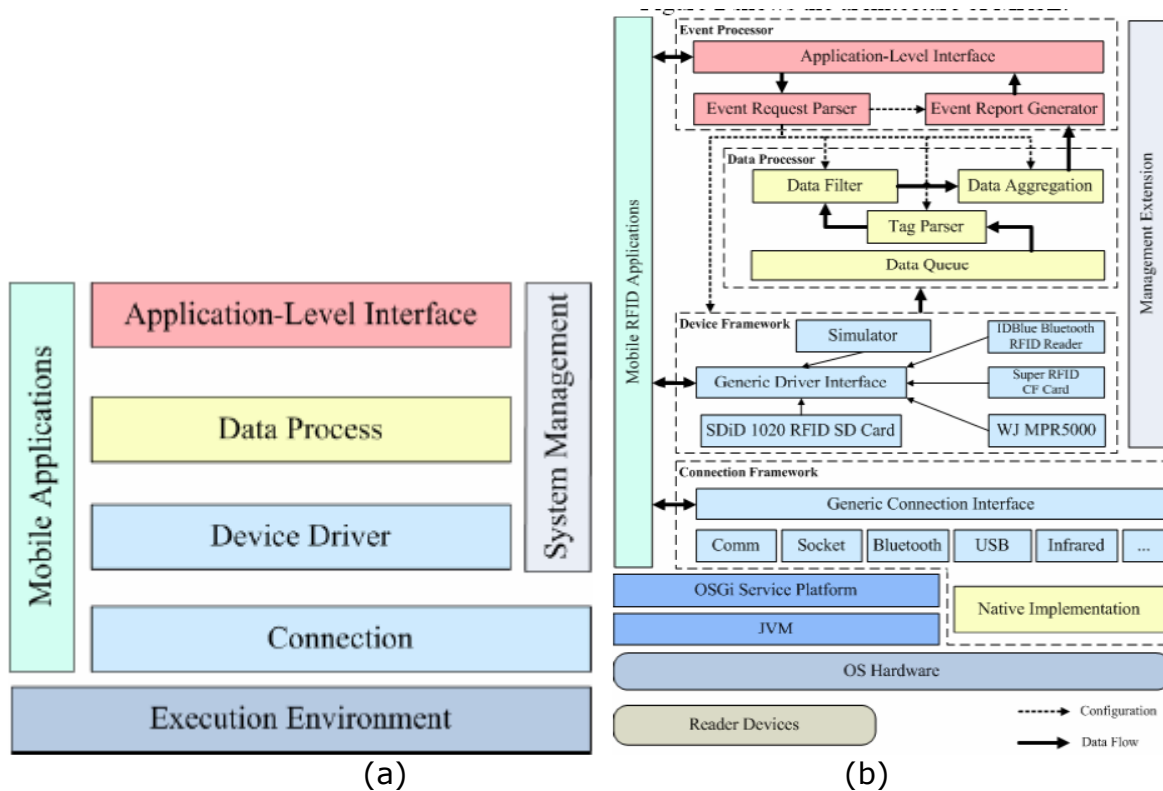


Figure 25: (a) Layers and (b) architecture of MRSE [32]

The work in [20] provides a detailed summary of recent trends in RFID and a Java based software framework for integration in mobile phones. The authors propose an RFID software framework in which the operating system (Java-based) will have special functions that will constant monitor possible RFID events.

Another work related to mobile RFID applications is presented in [16], where the authors address the problem of inconsistency of mobile networks by proposing a synchronization mechanism. The proposed mechanism is lightweight so as to fit the requirements of resource constrained mobile devices. The architecture is suited for mobile networks that have limitations in bandwidth and that present certain degree of inconsistency of connection at the application layers. The architecture for lightweight synchronization is displayed in Figure 27.

Another issue found in mobile RFID networks is the one of protocol compatibility. RFID networks in local area networks use protocols especially suited for wireline

environments. However, mobile applications use especially designed and lightweight wireless transmission protocols such as WAP, WML, etc. Therefore, it is mandatory to provide a framework for protocol conversion and information retrieval between the wireline and wireless worlds. The authors in [22] have proposed a framework for seamless information retrieval between an EPC network and a Mobile RFID network. For such purpose they have suggested proxy servers and routing address translation schemes for interconnecting the two worlds. A similar approach is used by the authors of [27] where they propose an integrated mobile RFID service architecture between B2B and B2C networks (mobile based). They use a service gateway that provides translation of semantics and data structures between the two worlds.

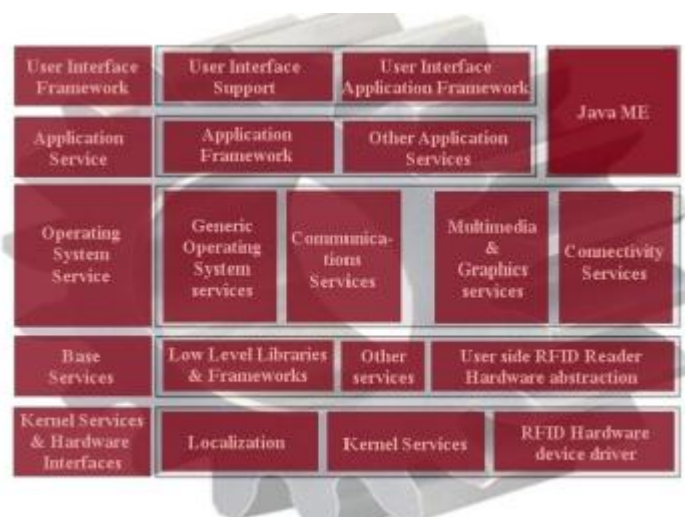


Figure 26: Proposed RFID software framework in [20].

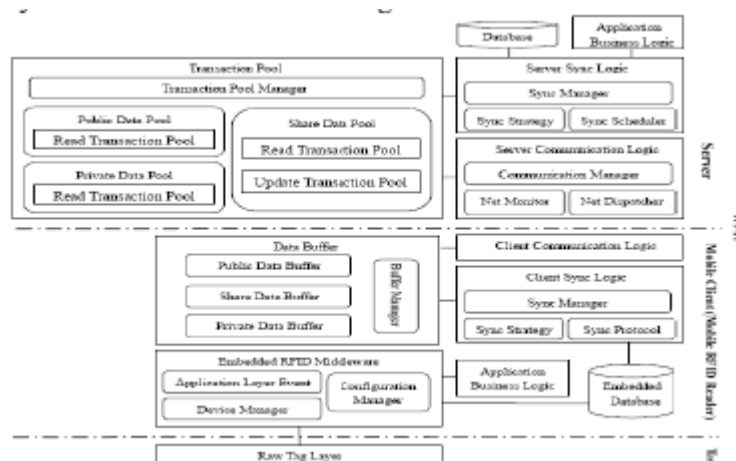


Figure 27: Lightweight Data Synchronization Architecture (taken from [16]).

Similar to the issues found for RFID middleware solutions in the wireline world, in the wireless world there is an increasing need for efficient and simple application development environments. Addressing this issue the paper in [21] proposes an application development methodology based on the so called ID-services middleware architecture. The approach used in this work is to use an intelligent

middleware that deals with most of the complex low level details of RFID, while leaving application development in simpler terms. A major feature of ID-services is the use of object oriented programming techniques to resemble objects in the physical world (mobile devices). The virtual objects are called proxy objects. The architecture of Id-services is displayed in Figure 28.

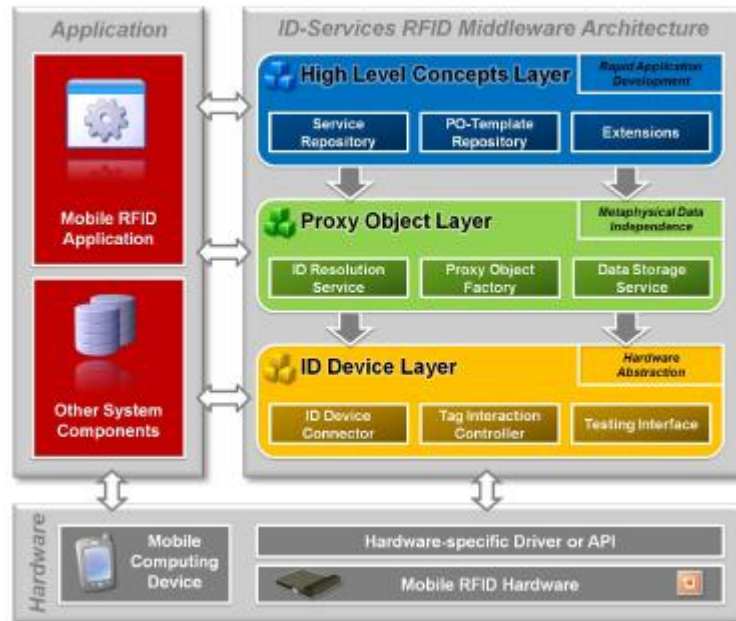


Figure 28: ID-services middleware architecture (taken from[21]).

Finally, providing mobility to readers will result in new privacy and security issues that have to be dealt by the middleware platform. To partially address this issue the authors in [23] propose a lightweight security mechanism for mobile RFID based on WIPI (Wireless Internet Platform for Interoperability). WI-PI provides an appropriate framework for developing of secure RFID applications by enabling business to provide new services to mobile customers and securing their services and transactions. The architecture of WIPI is displayed in the figure below.

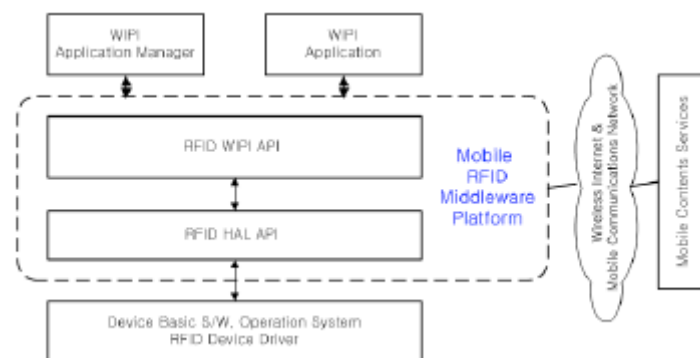


Figure 29: Wi-PI middleware architecture (taken from [23]).

Section 6 Synthesis

In this section we highlight major limitations of RFID middleware platforms (notably OSS middleware platforms), which prevent them to be widely adopted from European companies (notably SMEs).

Most OSS middleware platforms analyzed above are quite immature, overly EPC centered and do not provide any tools to essentially facilitate RFID development and deployment. As a result, they are subject to several limitations, some of which are inherent to the EPC architecture. Specifically, the most prominent of these limitations relate to the following areas:

- **Configurable Business Events Generation:** None of the current middleware implementations provides support for configurable and automated translation of filtered data (i.e. ECR reports) to business events (i.e. EPC IS Events). RFID developers are therefore still required to allocate programming effort in mapping ALE outputs to information sharing constructs. We strongly believe the configurable interpretation of RFID readings in a specific business context should be an essential functionality of any RFID middleware suite.
- **Support and integration for sensor data:** In addition to identifying objects many applications (e.g., cold chain management) need to detect and consume physical measurements (e.g., temperature, humidity, weight, acceleration (for shock-tracking), lighting). Hence, middleware frameworks must provide the means to integrate sensors and accordingly make their data accessible by the applications. EPC Global covers mainly the coding of things identifiers. While ALE reports can include (as extensions) physical measurements acquired by RFID sensor tags or sensors attached to the environment (e.g., RFID interrogator, container) at reading time, current middleware frameworks do not provide support for the consumption of these metrics. This is they do not cater for aligning the coding of these measurements with main international units, quantities standards and specifications (such as ISO 31-0, JSR 275, Open Geospatial Consortium GML, Google KML). Middleware frameworks must therefore provide support for adapting and using sensor readings in accordance to these coding schemes.
- **Integration of Actuators:** Experience with automatic identification applications manifests that there is often a need to quickly interact with the physical world based on a wide range of actuating functions such as locks, LEDs or mechanical controllers. Hence, RFID middleware frameworks need to be enhanced with actuator control frameworks.
- **Reader Connectors and Virtualization:** EPC-RP and EPC-LLRP prescribe reader protocol standards aiming at achieving vendor independence. In the current reader landscape however, there are still many readers that do not fully support these protocols. As a result there is still a need to provide an adaptation layer for non EPC-RP or EPC-LLRP compliant readers, similar to the HAL (Hardware Abstraction Layer) implementation of the Fosstrak project for EPC-RP. Most important, a middleware suite should include a uniform interfaces for communicating with upstream EPC layers (e.g., ALE).
- **End-to-End Management:** Non-trivial RFID solutions are supported by highly heterogeneous infrastructures comprising multiple tags, readers, sensors, as well as a host of middleware servers. Managing such an infrastructure end-to-

end is certainly asset towards facilitating the deployment and operation of RFID solutions. The EPC architecture and related middleware products emphasize on single reader management (e.g., based on the Reader Management Protocol) and do not support complete end-to-end of the RFID middleware solutions.

- Programmability and (Visual) Integrated Development Environments: Integrated development environments (IDEs) and visual tools are a key prerequisite to boosting RFID implementation. Most OSS RFID platforms do not provide complete integrated environments enabling visual development of RFID applications. The sole exceptions are probably the Rifidi project (<http://www.rifidi.org>) which attempts to provide an open source IDE for RFID, as well as Sun's JCAPS (Java Composite Application Platform) for RFID. Rifidi lets you develop an RFID system entirely with Software components and removes dependencies on hardware and infrastructure that RFID typically demands. Nevertheless, Rifidi does not deal with EPC artifacts and cannot support the flexible programming and configuration of filtering, eventing and business-level interpretation of RFID readings. Also, JCAPS is immature and does not comply with EPC standards. In order for RFID deployment to go mainstream, complete IDEs enabling RFID consultants and business users to configure standards based solutions through minimal programming effort are urgently required.

In addition to these limitations, current OSS middleware implementations do not pay sufficient attention to privacy issues for consumer applications.

As far as commercial solutions are concerned: Most commercial IT vendors have released middleware platforms (e.g., Oracle Sensor Edge Server, BEA Weblogic RFID Enterprise Server), which provide rather robust EPC compliant middleware functionality for collecting, filtering and managing RFID data. On the downside these platforms do not yet provide tools for RFID development from the business users. Moreover, commercial products tend to be heavyweight (i.e. resourceful) and come with high licensing costs, since they are usually bundled with the vendors' enterprise middleware platforms (i.e. application server). Hence, a large number of companies (mainly SMEs) cannot rely on these platforms in order to innovate based on RFID technology.

These limitations will be addressed in the ASPIRE project, starting from Deliverable D2.3 and D2.4 that will provide specification for the ASPIRE middleware and its programmability. Accordingly the ASPIRE middleware and tools, developed in the scope of WP3 and WP4 will provide remedy for most of these limitations.

Section 7 List of Acronyms

AGPL	: Affero General Public License
ASPIRE	: Advanced Sensors and lightweight Programmable middleware for Innovative Rfid Enterprise applications
BSD	: Berkeley Software Distribution (License)
CMS	: Content Management System
CMS	: Content Management System
EJB	: Enterprise JavaBeans
EPC	: Electronic Product Code
ETSI	: European Telecommunications Standards Institute
GPL	: Gnu Public License
GUI	: Graphical user interface
J2EE	: Java Enterprise Edition (Framework)
JCAPS	: Java Composite Application Platform Suite
LDAP	: Lightweight Directory Access Protocol
Lucene	: Text search engine written in Java
NFC	: Near Field Communication Forum
OSS	: Open Source Software
PHP	: PHP Hypertext Preprocessor (Language)
RFID	: Radio Frequency Identification
RSS	: Really Simple Syndication (Format)
SCM	: Software Configuration Management
SME	: Small and Medium Enterprise
SOA	: Service Oriented Architecture
SOTA	: State Of The Art
VCS	: Version Control System
VoIP	: Voice over IP
WebDAV	: Web-based Distributed Authoring and Versioning (Protocol)
XMPP	: eXtensible Messaging and Presence Protocol

Section 8 List of Figures

Figure 1 Decentralized reader architecture10
Figure 2 Centralized architecture11
Figure 3 EPC Set of Standards.....13
Figure 4 Fosstrak architecture19
Figure 5 RiFiDi architecture21
Figure 6 Singularity architecture22
Figure 7 Radioactive architecture.....24
Figure 8 Mobitec architecture25
Figure 9 UJF RFIDSuite architecture26
Figure 10 SJS RFID Software Architecture.....29
Figure 11 NetWeaver architecture39
Figure 12 WebSphere architecture41
Figure 13 (a) Layers and (b) architecture of CRRM (taken from [32].).....43
Figure 14 (a) Components of the publish/subscribe system and (b) components of the publish/subscribe server (taken from [37]).....43
Figure 15 Components of RFID Data server (taken from [37]).....43
Figure 16 RFID SCA middleware architecture proposed in [35].....45
Figure 17 Agent-based middleware architecture proposed in [18].45
Figure 18 Architecture of a reconfigurable OSGi based middleware architecture with a security layer based on access control for ubiquitous scenarios (proposed in [17]).....45
Figure 19 Security-enhanced middleware Architecture with context aware access control (proposed in [31]).....46
Figure 20 (a) Access control mechanism and (b) access control policy for the security-enhanced RFID middleware platform in [31].....47
Figure 21 Architecture of RF²ID [26].....47
Figure 22 FlexiRFID middleware architecture in [13].48
Figure 23: Generic wireless network architecture for the ASPIRE project.....50
Figure 24: Communication interfaces50
Figure 25: (a) Layers and (b) architecture of MRSE [32].....51
Figure 26: Proposed RFID software framework in [20].....52
Figure 26: Lightweight Data Synchronization Architecture (taken from [16]).52
Figure 28: ID-services middleware architecture (taken from [21]).....53
Figure 29: Wi-PI middleware architecture (taken from [23]).53

Section 9 List of Tables

Table 1 Different types of tags and their technical characteristics [41].....12
Table 2. Standards compliance of OSS RFID middleware platforms18
Table 3. Comparison of proprietary middleware platforms.....33
Table 4. Comparison of proprietary middleware platforms.....34

Section 10 References and Bibliography

- [1] Christian Floerkemeier, Christof Roduner, and Matthias Lampe, 'RFID Application Development with the Accada Middleware Platform', IEEE Systems Journal, Vol. 1, Issue 2, pp.82-94, December 2007.
- [2] Architecture Review Committee, "The EPCglobal Architecture Framework," EPCglobal, July 2005 (available at: <http://www.epcglobalinc.org>)
- [3] T. Staake, F. Thiesse, and E. Fleisch, "Extending the EPC network: the potential of RFID in anti-counterfeiting," in SAC '05: Proceedings of the 2005 ACM symposium on Applied computing. Santa Fe, NM, USA: ACM Press, Mar. 2005, pp. 1607–1612.
- [4] S. Prabhu, Xiaoyong Su, Harish Ramamurthy, Chi-Cheng Chu, Rajit Gadh, "WinRFID –A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications", Invited chapter in Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, Rajeev Shorey, Chan Mun Choon, Ooi Wei Tsang, A. Ananda (eds.), John Wiley (to appear), available at: <http://www.wireless.ucla.edu/rfid/winrfid/>.
- [5] Walter Rudametkin, Lionel Touseau, Maroula Perisanidi, Andrés Gómez, Didier Donsez, "NFCMuseum: an Open-Source Middleware for Augmenting Museum Exhibits", accepted for public demonstration in the IEEE International Conference on Pervasive Services (ICPS 2008), Sorrento, Italy, July 6-10, 2008.
- [6] Florian Michahelles, Frederic Thiesse, Albrecht Schmidt, John R. Williams, "Pervasive RFID and Near Field Communication Technology," IEEE Pervasive Computing, vol. 6, no. 3, pp. 94-96, c3, Jul-Sept, 2007
- [7] EPCglobal standards, <http://www.epcglobalinc.org/standards>
- [8] Jeremy Landt, The history of RFID, IEEE Potentials, October-November 2005.
- [9] Nova Ahmed and Umakishore Ramachandran, "Load Shedding Based Resource Management Techniques for RFID Data," 2009 International Conference on RFID, 27-28 April 2009 Page(s):306 – 313.
- [10] Benham Jamali, Peter H. Cole and Cheng C. Lim, "CRISP: A Flexible Integrated Development Platform for RFID Systems," ICIEIA 2009, 4th IEEE Conference on Industrial Electronics and Applications.
- [11] Jiaqi Zhu, Yu Huang, and Hanpin Wang, "A Formal Descriptive Language and an Automated Detection Method for Complex Events in RFID," 2009 33rd Annual IEEE International Computer Software and Applications Conference.
- [12] Lars Kulseng, Zhen Yu, Yawen Wei, and Yong Guan, "Lightweight Secure Search Protocols for Low-Cost RFID Systems," 2009 29th IEEE International Conference on Distributed Computing Systems.
- [13] M.E. Ajana, H. Harroud, M. Boulmalf, and H. Hamam, FlexRFID: A Flexible Middleware for RFID Applications Development. WOCN 09, IFIP International Conference on Wireless and Optical Communications, 28-30 April 2009 Page(s):1 – 5.
- [14] John P. Curtin, Robert L. Gaffney and Frederick J. Riggins, "The RFID e-Valuation Framework Determining the Business Value from Radio Frequency

- Identification," Proceedings of the 42nd Hawaii International Conference on System Sciences, 2009
- [15] Baoyan Song, Pengfei Qin, Hao Wang, Weihong Xuan, and Ge Yun, "bSpace: A Data Cleaning Approach for RFID Data Streams Based on Virtual Spatial Granularity," 2009 fifth International Conference on Hybrid Intelligent Systems, Volume 3, 12-14 Aug. 2009 Page(s):252 – 256.
- [16] Liu Fagui, Jie Yuzhu, and Ruan Yongxiong, "A Framework for Lightweight Data Synchronization on Mobile RFID Devices," 2009 ISECS International Colloquium on Computing, Control, and Management, Volume 4, 8-9 Aug. 2009 Page(s):468 – 473.
- [17] Bo Du, Songyan Ju, and Dong Wang, "Access Control for OSGi-Based Reconfigurable RFID Middleware," ICCIT 09, fourth International Conference on Computer Sciences and Convergence Information Technology, 24-26 Nov. 2009 Page(s):1010 – 1014.
- [18] Libe V. Massawe, Farhad Aghdasi, and Johnson Kinyua, "The Development of a Multi-agent based Middleware for RFID Asset Management System using the PASSI Methodology," ITNG09, Sixth International Conference on Information Technology: New Generations. 27-29 April 2009 Page(s):1042 – 1048.
- [19] Shanglian Peng, Zhanhuai Li, Lin Chen, Yanming NIE, and Qun Chen, "Complex Event Processing over Multi-granularity RFID Data Streams," ICCSIT 2009, 2nd IEEE International Conference on Computer Science and Information Technology.
- [20] Rohit Pathak and Satyadhar Joshi, "Recent Trends in RFID and a Java based Software Framework for its Integration in Mobile Phones," First Asian Himalayas International Conference on Internet 2009. 3-5 Nov. 2009 Page(s):1 – 5.
- [21] Joachim Schwieren and Gottfried Vossen, "A Design and Development methodology for mobile RFID applications based on the ID-services Middleware Architecture," MDM 09, Tenth International Conference on Mobile Data Management Systems, services and Middleware.
- [22] Min Kyu Han, Il Woo Paik, Byung Hee Lee, and Jin Pyo Hong, "A Framework for seamless Information retrieval between and EPC Network and a Mobile RFID Network," Proceedings of the Sixth IEEE International Conference on Computer and Information technology (CIT06), Sept. 2006 Page(s):98 – 98.
- [23] Namje Park, Jooyoung Lee, Howon Kim, Kyoil Chung, and Sungwon Sohn, "A Layered Approach to Design of Light-Weight Middleware Systems form Mobile RFID Security (SMRM : Secure Mobile RFID Middleware System)," CIVVS, IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, March 30 2009-April 2 2009 Page(s):51 - 57.
- [24] Xiaogang Peng, Zhen Ji, Zongwei Luo, Edward C. Wong, and C.J. Tan, "A P2P Collaborative RFID Data Cleaning Model," The 3rd International Conference on Grid and Pervasive Computing Workshop, 25-28 May 2008 Page(s):304 - 309.
- [25] Oh Gi oug, Kim Doo yeon, Kim Sang and Rhew Sung yul, "A Quality Evaluation Technique of RFID Middleware in ubiquitous Computing," 2006 International Conference on Hybrid Information Technology (ICHIT06), Volume 2, 9-11 Nov. 2006 Page(s):730 - 735.

- [26] Nova Ahmed, Ranjish Kumar, Robert Steven French, and Urmakishore Ramachandran, "RF²ID: A Reliable Middleware Framework for RFID Deployment," IEEE International Parallel and Distributed processing Symposium, 26-30 March 2007 Page(s):1 - 10.
- [27] Sangkeum Yoo, Junseob Lee, Yongwoon Kim, and Hyungjun Kim, "An integrated mobile RFID service architecture between B2B and B2C networks," The 9th International Conference on Advanced Communication Technology, Volume 1, 12-14 Feb. 2007 Page(s):90 - 93.
- [28] Tingting Mao, John Williams and Abel Sanchez, "Interoperable Internet Scale Security Framework for RFID Networks," IEEE 24th International Conference on Data Engineering Workshop 2008, 7- 12 April 2008 Page(s):94 - 99.
- [29] Christian Floerkemeier and Sanjay Sarma, "An Overview of RFID System Interfaces and Reader Protocols," 2008 IEEE International Conference on RFID, The Venetian, Las Vegas, Nevada, USA April 16-17 2008, pp. 232-240.
- [30] Hyangjin Lee and Jeeyeon Kim, "Privacy threats and issues in mobile RFID," Proceedings of the First International Conference on Availability, Reliability and Security (ARES 06).
- [31] Jieun Song and Howon Kim, "The RFID Middleware System supporting Context-Aware Access Control Service," ICACT 2006. The 8th International conference on Advanced Communication Technology. Volume 1, 20-22 Feb. 2006 Page(s):4 pp. - 866.
- [32] Jie Wun, Dong Wang and Huanye Sheng, "Design an OSGi extension service for Mobile RFID Applications," IEEE International Conference on e-business engineering, ICEBE 2007, 24-26 Oct. 2007 Page(s):323 - 326.
- [33] Ming Ling Chuang and Wade D. Shaw, "RFID: Integration Stages in Supply Chain Management," IEEE Engineering Management Review, Vol. 35, No.2, second Quarter 2007, pp. 80-87.
- [34] Jie Wun, Dong Wang and Huanye Sheng, " A Component-based reconfigurable RFID Middleware," The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Nov. 5-8, Taipei Taiwan.
- [35] Shiqi Ma, Jiangtao Tang, and Dong Wang, "Process-based Application Level Architecture for RFID System," 5th International Conference on Wireless Communications, Networking and Mobile Computing, 24-26 th Sept 2009, pp. 1-5.
- [36] Fang Liu, Fagui Liu, Zehao Liu and Yang Zhang, "Application Component Development of OSGi and JBPM- Based RFID Middleware," 3rd International Anti-counterfeiting, Security, and Identification in Communication 2009.
- [37] Jian Yu and Shengli Lai, "Research of RFID Middleware Publish/Service Mechanism Based on SOA," WiCom 09, International Conference on Wireless Communicaions, networking and Mobile Computing 2009, 24-26 Sept. 2009 Page(s):1 - 5.
- [38] Geroge Roussos, Sastry S. Duri, and Craig Thompson, "RFID meets the Internet,"IEEE Internet Computing, Volume 13, Issue 1, Jan.-Feb. 2009 Page(s):11 - 13.
- [39] Ari Juels, RFID Security and Privacy: A Research Survey, IEEE Journal on Selected Areas in Communications, Vol. 24, No 2., February 2006.

- [40] Shailesh M. Birari, Mitigating the Reader Collision Problem in RFID Networks with Mobile Readers, Master Dissertation, Kanwal Rekhi School of Information Technology Indian Institute of Technology Bombay.
- [41] Ramaswamy Chandramouli, Tim Grance, Rick Kuhn, and Susan Landau. Security Standards for the RFID Market. IEEE Security and Privacy.