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, Methods, Tools and Techniques

Due date of deliverable: M6
Actual Submission date:

Deliverable ID:	**WP2/D2.1**
Deliverable Title:	Review of State-of-the-Art Middleware
Responsible partner:	INRIA
Contributors:	Michel Cezon, Guillaume Vaudaux-Ruth, Luc Laurens – INRIA Loïc SCHMIDT - INRIA Nektarios Leontiadis – AIT John Soldatos -AIT Humberto Moran – OSI
Estimated Indicative Person Months:	5

Start Date of the Project: 1 January 2008 Duration: 36 Months

Revision: 1.5
Dissemination Level: PU

Document Information

Document Name: **Review of State-of-the-Art Middleware**
Document ID: **WP2/D2.1**
Revision: **1.5**
Revision Date: **31 March 2008**
Authors: **INRIA, AIT, OSI, IT**
Security: **Public**

Approvals

	Name	Organization	Date	Visa
Coordinator	Neeli Rashmi Prasad	CTIF-AAU		
Technical Coordinator	John Soldatos	AIT		
Quality Manager	Anne Bisgard Poos	CTIF-AAU		

Reviewers

Name	Organization	Date	Comments	Visa
	IT			
	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
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

Content

Section 1 - Executive summary

This deliverable presents a detailed study, review and assessment of the state-of-the-art Open Source Software (OSS) components used in RFID middleware.
A preliminary section (Section 2) provides a general overview of the ASPIRE project, and illustrates the importance of this deliverable, whose purposes are:

1. 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 selecting components that could be reused in the scope of the ASPIRE middleware codebase.
2. 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 5 is dedicated to the RFID OSS middleware whereas the section 6 is dedicated to RFID proprietary (commercial) middleware. Note that while the ASPIRE middleware should not be directly comparable 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 from the business users.

A last section summarizes our findings and provides recommendations on RFID tools that may be used to implement the ASPIRE software and middleware.

Section 2 - Context and overview

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).

The last section summarizes our findings. The present deliverable reviews not only OSS middleware suites (directly comparable to the ASPIRE middleware platform), but also some of the most prominent proprietary vendor products.

Section 3 – 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 4 - Integration of the RFID platforms into the mobile telecommunications infrastructure

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 Fig.1:

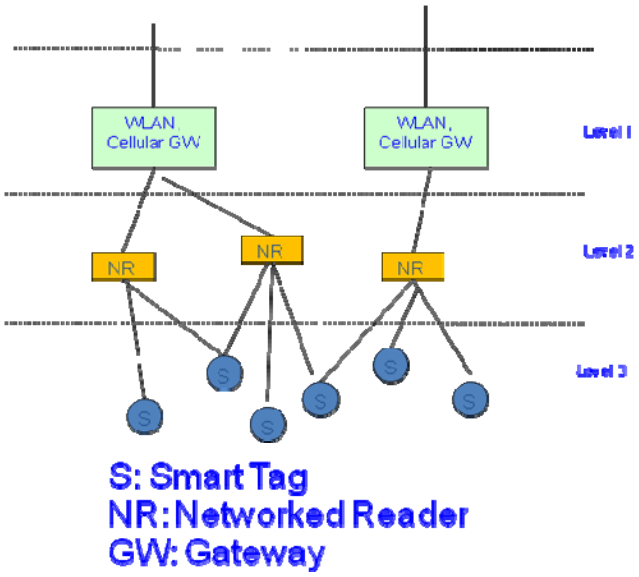

Figure 1: Generic wireless network architecture for the ASPIRE project

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

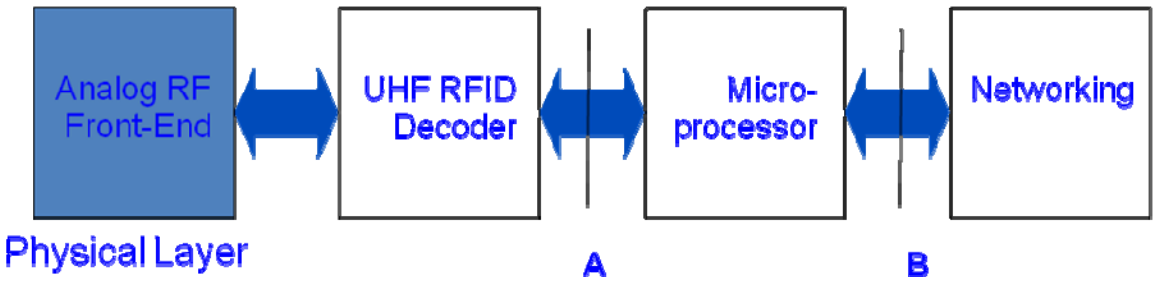

Figure 2: Communication interfaces

Section 5 - State of the art of OSS RFID Middleware

5.1 Overview

Some OSS RFID middlewares are already available even if they do not match completely with what customers expect. 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 middlewares 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**	
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**				

5.2 Fosstrak (previously Accada)

URL : http://www.fosstrak.org
Module versions:

- Reader module: 0.4.1
- Filtering and collection module: 0.4.0
- EPCIS module: 0.4.1
- Tag data standard: 0.4.0

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

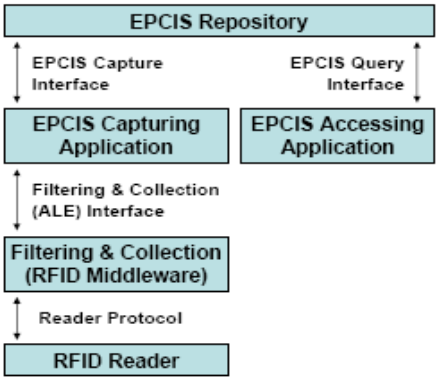

Figure 3 Fosstrak architecture

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.

5.3 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

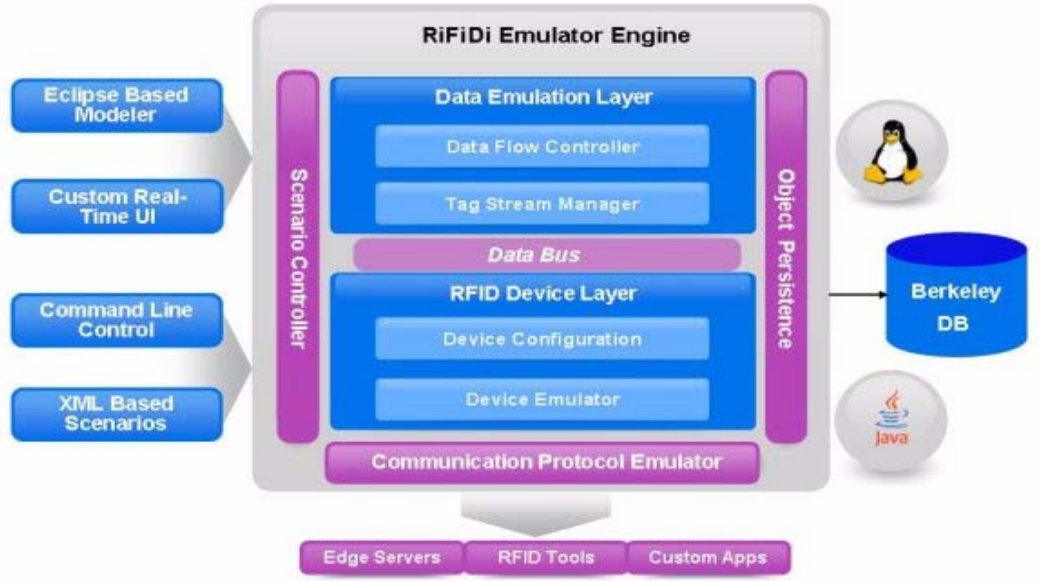

Figure 4 RiFiDi architecture

Evaluation
(poor / correct / good / excellent)

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

5.4 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 **Strong and weak points**

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

Currently only available in beta version

Compliance : EPC Global

Architecture overview

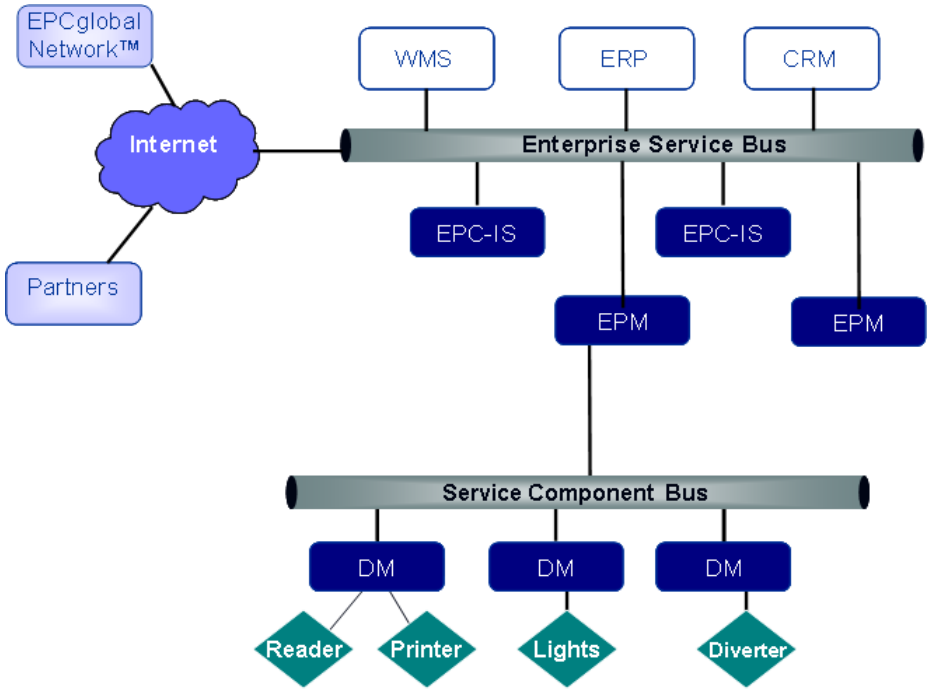

Figure 5 Singularity architecture

Evaluation (poor / correct / good / excellent)

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

5.5 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

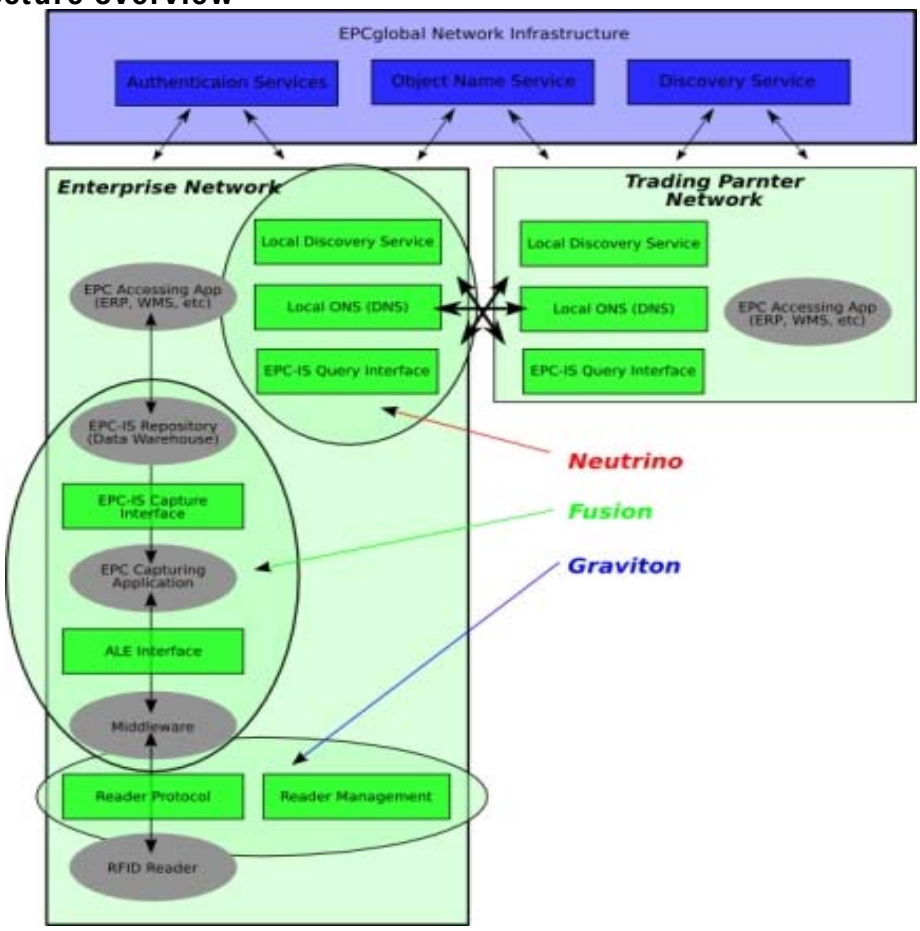

Figure 6 **Radioactive architecture**

Evaluation
(poor / correct / good / excellent)

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

5.6 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

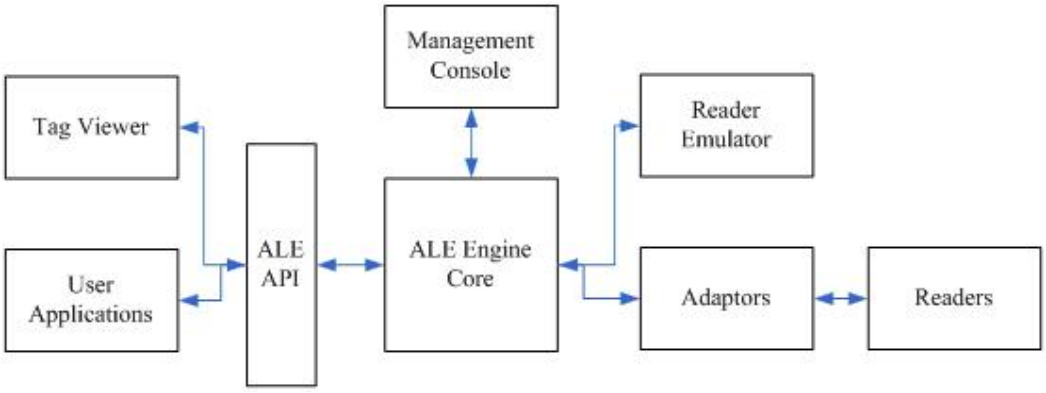

Figure 7 Mobitec architecture

Evaluation (poor / correct / good / excellent)

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

5.7 RFIDSuite

URL : - Version: - License : LGPL Language: Java	Editor/Main contributor : LIG laboratory, UJF University Platform server : OS portable

Description

RFID Suite is globally architectured 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

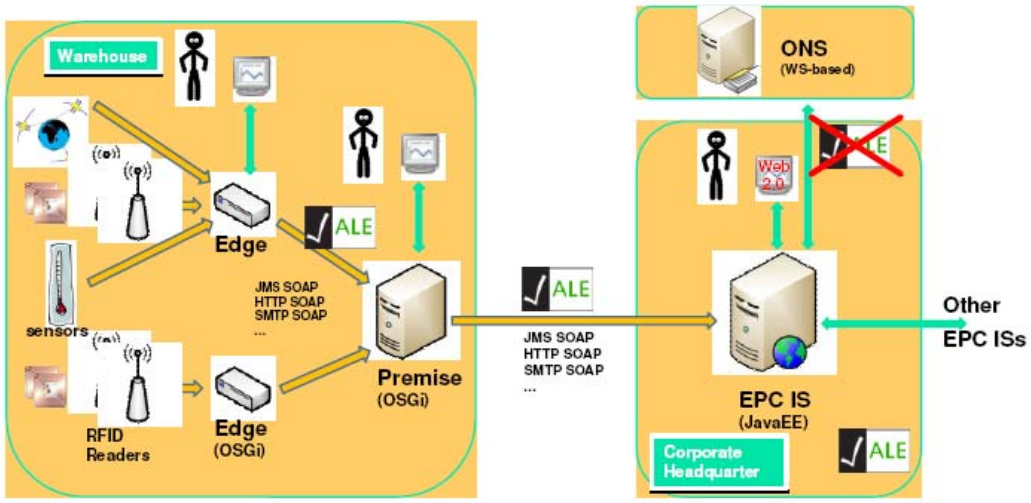

Figure 8 UJF RFIDSuite architecture

Evaluation
(poor / correct / good / excellent)

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

5.8 Logicalloy

URL : http://www.logicalloy.com/ Version: 1.1.3 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

5.9 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

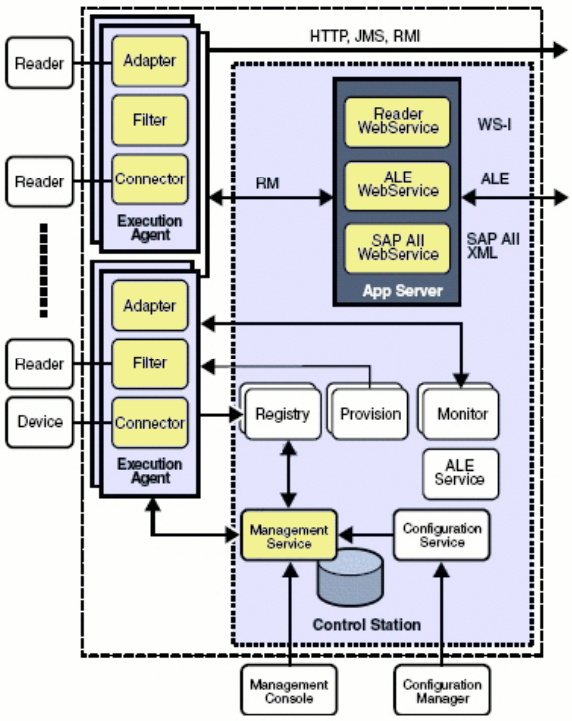

Figure 9 SJS RFID Sofware Architecture

Evaluation (poor / correct / good / excellent)

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

5.10 RFID Library

5.10.1 RFID-perl

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

5.10.2 RFID C library

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

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

5.10.3 RF-Dump

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

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 6 - State of the art of Proprietary RFID Middleware

6.1 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 middleware[1].

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 summarise 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	Comparison with ASPIRE				
		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

The following sections detail the characteristics and history of three selected leading proprietary RFID middleware products.

Product	Vendor	Comparison with ASPIRE				
		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

The following sections detail the characteristics and history of three selected leading proprietary RFID middleware products.

6.2 OAT Systems and OAT Foundation Suit[2]

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.

Features

- **Built-in adapters** for trading partner data.
- **Consistency engine** uses inferencing 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

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

- **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.

6.3 SAP and NetWeaver[3]

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 date 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, 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

[3] Sources:
Leaver, Sharyn. Evaluating RFID Middleware, Forrester Research, August 13 2004
SAP website.

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

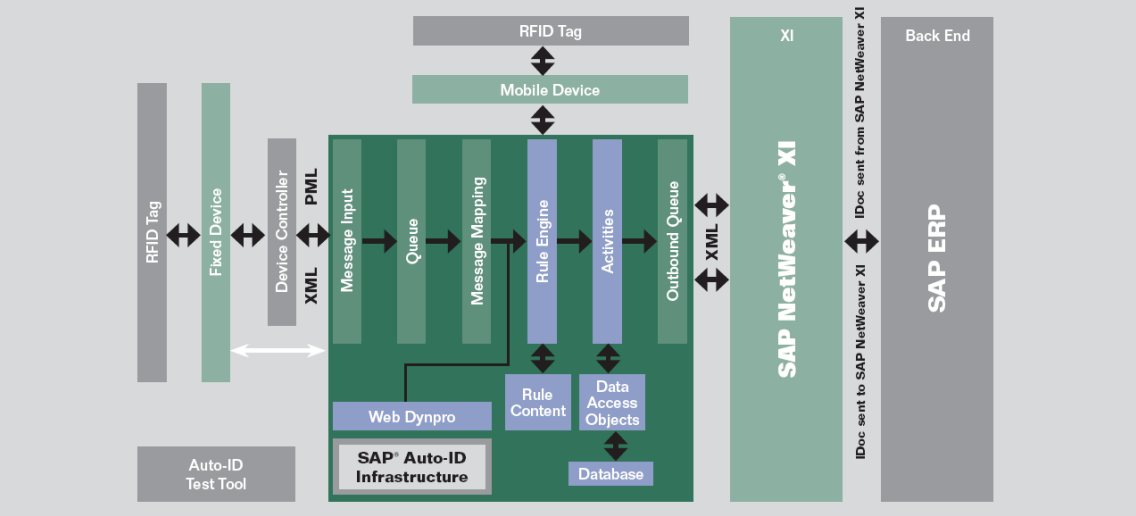

Figure 10 NetWeaver architecture

6.4 IBM and WebSphere[4]

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 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

[4] Sources:
IBM website.

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
- Openess 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:

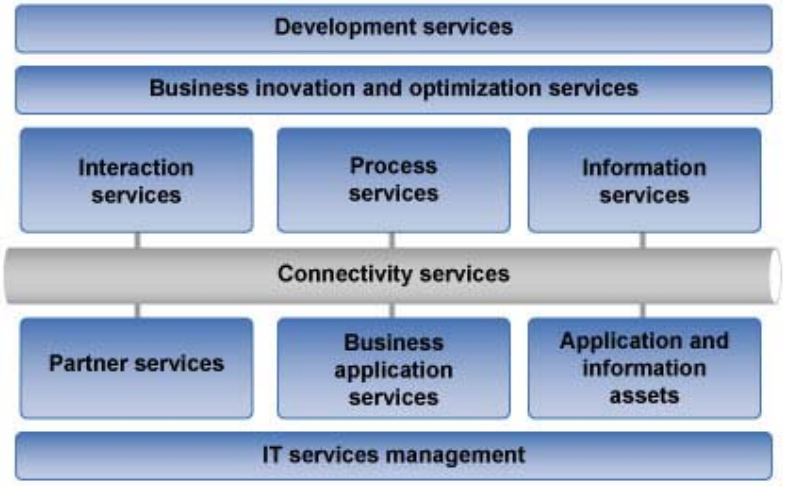

Figure 11 WebSphere architecture

Section 7 - Synthesis

7.1 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 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: Current middleware implementations do not provide support for configurable and automated translation of filtered data (i.e. ECReports) 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 to 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 Accada 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 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.

List of Figures

List of Tables

Section 8 - 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

Appendix A

A.1 License

See deliverable D3.1 ”Licensing and reuse”**}}}}}}}}}}}}}}}}}}**