WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications

B. S. Prabhu, Xiaoyong Su, Harish Ramamurthy, Chi-Cheng Chu, Rajit Gadh
UCLA - Wireless Internet for the Mobile Enterprise Consortium (WINMEC)
420 Westwood Pl., Los Angeles CA 90095

Abstract

Radio Frequency Identification (RFID) middleware is a new class of software which facilitates data and information communication between automatic identification physical layer and enterprise applications. It provides a distributed environment to process the data from tags read by the readers, translates the data where necessary, and routes it to a variety of backend applications using suitable technologies such as Web, Remote and Windows Services. This paper reports different challenges and the corresponding research approach in developing a RFID middleware to provide a seamless environment from the edge of the enterprise network; moving data from the point of transaction to the enterprise systems. Key features of the RFID middleware architecture are encapsulation of communication details, large-scale network management, intelligent data processing and routing, hardware and software interoperability, system integration and system extendibility.

To deal with high volume data, WinRFID middleware is supported by novel algorithms and data representation schemes capable of processing large amounts of data, rectifying errors in real-time, identifying patterns, correlating events, reorganizing and scrubbing data and recovering from faults and exceptions. Interoperability involves simultaneous distributed working of receivers/readers and transponders/tags at different frequencies using different protocols, with read/write capabilities, different read rates, and other characteristics as a layer transparent to the applications. Network management involves deployment, initialization and control of receivers and transponders, which can be organized into a hierarchical structure with operational syntax and semantics attached to each or a group of receivers, transponders and concentrators or even the edge computers.

Keywords : Radio Frequency Identification, RFID, Middleware, Wireless Sensors, Automated Data Collection, .NET Framework

1.0 Introduction

Globalization and accelerated innovation cycles are forcing industry to adopt new technological improvements in manufacturing automation, process execution, engineering practices and control applications. At the same time it is desired that these improvements inject flexibility into the system for it to respond quickly to alterations and disruptions in real time. The key enabler for this to happen is *ubiquitous information flow*.

Towards this end, industry is looking for a new paradigm, which can provide real time visibility for most of the activities to the collaborating partners involved; enabling quick decisions and shorten process times. Activities, which are in the forefront of this plan, are supply chain management, inventory control, asset management and tracking, theft and counterfeiting prevention, access restrictions and security, hazardous material management, and others.

A number of mobile wireless technologies have all been significant catalysts for this transformation and have also spurred the development of inventive business process facilitators such as speed, quality, timeliness, adaptability and depth of information. But, amongst them Radio Frequency Identification (RFID) has drawn a lot of attention of the industry as it has already demonstrated the potential to enhance efficiencies of activities across business processes by providing a means to affix unique identification and related information to individual items and enable the items to travel with the information, which can be utilized as the items pass through the different process stages, increasing productivity, minimizing errors, improving accuracy, and potentially reducing labor costs [Finkenzeller 2003, Zawada and O'Kelly 2003, IBM 2003].

RFID does not require line-of-sight access to communicate – without requiring physical contact, multiple tags can be identified, and tags mounted on products or consignments can survive in harsh environments such as extreme temperatures, moisture, and rough handling. The technology significantly enhances the velocity of information flow by overcoming the limitations of other manual, data-collection methods [Välkkynen, et al., 2003]. With proper deployment it is deemed to enable industry to focus on real time optimization of the activities by providing accurate, timely visibility of the various stages of the activities and make intelligent strategic decisions [Want, et al., 1999, Sharp 2000, Sharp 2003, Zhekun, et al., 2003, Gadh, Aug. 2004, Gadh, Oct. 2004].

However, there are also some major weaknesses associated with the currently available RFID technology, which has hampered the technology going primetime to some extent. In some of the recent industry pilots and studies done by researchers and industry, it was found that a number of systems had a lot of problems - failed or provided erroneous reads, had problems in handling large amounts of data generated by the tags, were found costly for some applications, lack mature standards, burdened with collisions due to multiple-tag reads, failed in the presence of metal and liquid based products, etc. [Kellam 2003, Floerkemeier and Lampe 2004, Lewis 2003, Brewin 2003, Brandel 2003]

Nonetheless, the industry is very interested in RFID technology because it is expected to provide a means to bring passive objects online; integrate physical assets into the overall IT infrastructure and subject to intelligent decision making directly from the information available on these physical objects thereby increasing efficiencies, reducing losses, providing superior quality control, and other benefits [AIMGlobal 1998, McFarlane 2002].

This situation provides an excellent research opportunity and the ongoing research work under the aegis of Wireless Internet for the Mobile Enterprise Consortium (WINMEC) in UCLA, attempts to propose a RFID ecosystem to mitigate a number of the above mentioned problems by architecting a new generic paradigm using Remoting, Windows and Web Services technologies and frameworks based distributed middleware.

Before we begin discussing the middleware for RFID applications, concepts of RFID technology, middleware architectures, and enterprise class distributed systems will be introduced in the sections to follow.

1.1 Radio Frequency Identification (RFID) System

RFID is an automatic identification technology that can be used to provide electronic identity to an item/object. A typical RFID system consists of transponders (tags), reader(s), antennas and a host (computer to process the data) as shown in Figure 1.

History of RFID can be traced back to 'friend or foe' long-range transponder systems fitted to aircraft in World War II, followed by a number of reported scientific works up to 70's; in 80's and 90's they appeared in commercial applications such as animal tracking, vehicle tracking, factory automation, and toll collection and finally in the 21st century the technology is on the verge of exploding by virtue of its capability to facilitate 'reality mining' mainly in retail, food and drug supply chains, security and trade [Landt 2001, Cardullo 2004, Evilsizer 2004, Boone 2004]

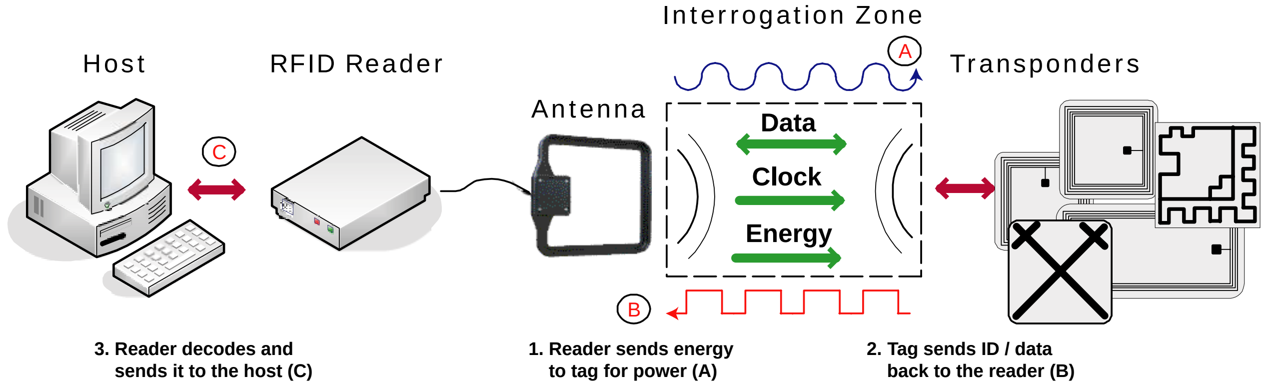

Figure 1. Typical RFID System

Communication in RFID occurs through radio waves, where information from a tag to a reader or vice versa is passed via an antenna. Unique identification or electronic data is stored in RFID tags, which can consist of serial numbers, security codes, product codes, and other object specific data. Using an RFID reader, the data on the tag can be read wirelessly, even without line-of-sight access, even when tagged objects are embedded inside packaging or even when the tag is embedded inside an object itself. RFID reader can read multiple RFID tags simultaneously [Finkenzeller 2003, AIMGlobal 2000, Enciu 2000, Zhekun, et al., 2004]. RFID technologies that work at different frequencies are currently available and their selection mainly depends on the requirements of the end application.

The strong interest shown by industry in RFID is mainly on account of the following features, which potentially would lead to better business and workflow processes:

- Tags can store or archive data which can be modified and updated during various stages of the processes.
- Automation of data collection at a high rate, eliminating the need for manual scanning
- Accurate data collection hence less problems due to erroneous data for decision making
- Less handling of goods and hence less labor required
- Simultaneous identification of multiple tagged items in the read area

To better understand RFID a discussion on the types of tags and their protocols is warranted. The following section briefly describes the salient points of the currently available tags and supporting protocols.

1.1.1 Types of RFID tags

RFID tags come in a wide variety of shapes, sizes, capabilities and materials – from as small as tip of a pencil or a grain of rice to as big as a 6” ruler. They come in a variety of shapes such as key fobs, credit cards, capsules, bands, disks, pads, etc. Tags can have metal external antennas, embedded antennas or the latest being printed antennas.

Tags can either be ‘passive’ - which work without a power source or ‘active’ - which have an embedded power source. The tags can either be read only or read/write capable. The range of sensing the tags from a reader can vary from a few centimeters to a few meters depending on the power output, radio frequency used, and the type and size of tag antennas.

The frequency bands / emission power for the RFID systems are limited according to governmental regulations [FCC 2001]. The choice of a particular frequency depends on application requirements such as absorption in liquids, the reflection on surfaces, tag densities, power demand, size of tags, location of tags, exposure to temperature range, data transmission speed and data rates. Even within the same class of tags, say EPC Class 0 and Class 1, problems exist as far as tags working with a particular reader is concerned on account of different antenna designs and tag sizes. Some of these factors are contradictory, and hence an optimal combination will have to be identified for each application [Kern 1999]. Table 1 lists the current popular RFID technologies in use and their typical characteristics, highlighting the range of applications where it can make a mark.

Frequency Band	Read Range	Characteristics	Typical Applications	RFID protocols
Low 100 – 500 kHz	Up to 4-6 in	Short to medium read range Inexpensive Low reading speed Can read through liquids	Access control Animal identification Inventory Control Vehicle immobilizer	ISO/IEC 18000-2
High 10 - 15 MHz	Up to 8 feet	Short to medium read range Potentially inexpensive Medium read speed Can read through liquids	Access control Smart cards Item tracking Electronic surveillance	ISO/IEC 18000-3 EPC HF Class 1 ISO/IEC 15693 ISO 14443 (A/B) I-Code, Tag-It, Hitag, MiFare
Ultra-high 850 – 950 MHz	10-20 feet	Long read range High reading speed Reduced chance of signal collision Problems with liquids and metal	Railroad asset monitoring Toll collection systems Supply chain Item tracking	ISO 18000-6 EPC Class 0, Class 1
Microwave 2.4 – 5.8 GHz	< 3 feet	Moderate read range Chance of signal collisions Very high data rates Problems with liquids and metal	Railroad asset monitoring Toll collection systems Airline baggage tracking	ISO/IEC 18000-4

Table 1. Current RFID Technologies

As RFID is embraced by industry and they start to internalize the technology in their in-house processes, it is expected that the infrastructure should support different RFID technologies to provide best benefits to a number of business processes, by utilizing the appropriate features of different RFID technologies such as read/write range, data rate, interference, etc., at different stages of the business or workflow processes.

1.2 Middleware technologies

Middleware is multidisciplinary and tries to merge features and knowledge from diverse areas such as distributed systems, networks, and even embedded systems [Carvalho 2003, Colyer 2003, Kelkar and Gamble 1999, Vinoski 2002]. It refers to the software layer which resides between the physical layer components (hardware), firmware or operating systems, which deal with low level system calls and communication protocols, and the upper layer standalone or even distributed enterprise applications generally interacting via the network. The boundaries between layers is not very sharp and as software evolve the features of middleware become part of operating systems, firmware, application frameworks and other layers of the IT infrastructure [Barbosa and Porto 2001].

But, they are an essential part of the latest generation distributed systems – both new developments or by way of integration of existing applications and services. The anatomy of a typical middleware is as shown in figure 2. The success of the architecture depends on how well the different pieces in different layers fit together or made to fit together by modifying some of the modules.

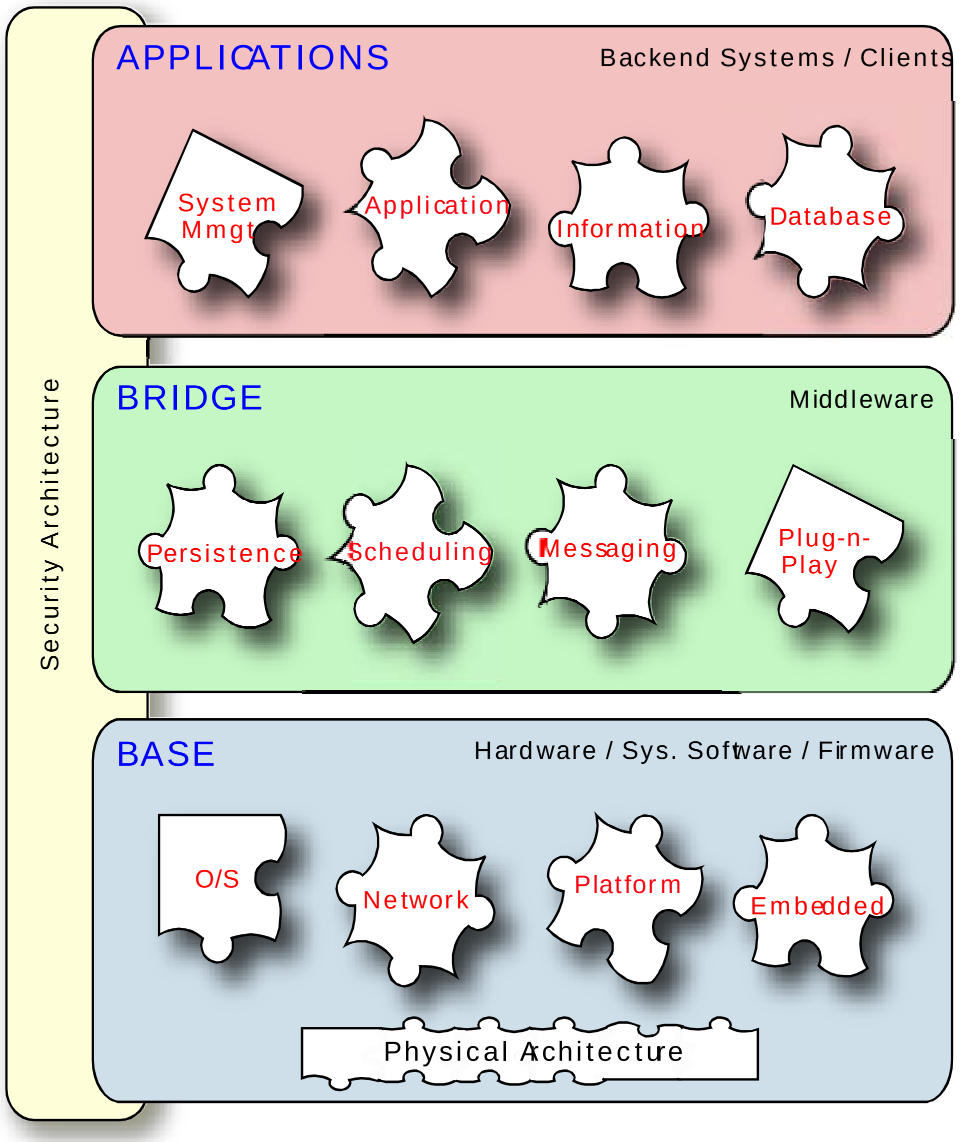

Figure 2. Anatomy of a middleware supported IT infrastructure

Middleware systems generally support the interaction of disparate application programs, collaborative groupware and other federated workflow systems. It seeks primarily to hide the underlying networked environment's complexity by insulating applications from heterogeneous hardware, explicit protocol handling, distributed data repositories, networking technologies and others and provide quality of service guarantees, security, scalability, ubiquity and ease of integration of applications and systems. [Middleware Domain Team 2001, Britton 2001, Geihs 2001, Sutton 2000, Hartwich 2003, Lowe and Noga 2002].

Thus, in developing a successful middleware a number of features must be considered. Some of the important features are network, languages and operating systems independence; architecture interoperability (object-oriented, client/server, push/pull, web services), plug-n-play operation of different modules and components; service location, and message and data routing; scheduling transactions through publish / subscribe schemes; mechanisms for fault tolerance and recovery from failures; and end-application specific features such as events, persistence, adapters, etc. [Colyer 2003, Carvalho 2003, Gimenez and Kim 2001, Meer, van der, 2002].

Middleware functions fall broadly into three main categories – application specific, information exchange, and management and support functions. Figure 3 shows an accepted categorization of the middleware systems in use currently [Bishop and Karne 2003, Vinoski 2002, Emmerich 2000]. Each of the categories caters to a principal requirement of an enterprise IT infrastructure. However, as the infrastructure matures, features of other categories will have to be integrated and a fully working middleware will have features of many of the categories shown in the figure. In section 3.0 we will see how some of these features are relevant to RFID middleware, how can they be integrated and what value they add to the infrastructure.

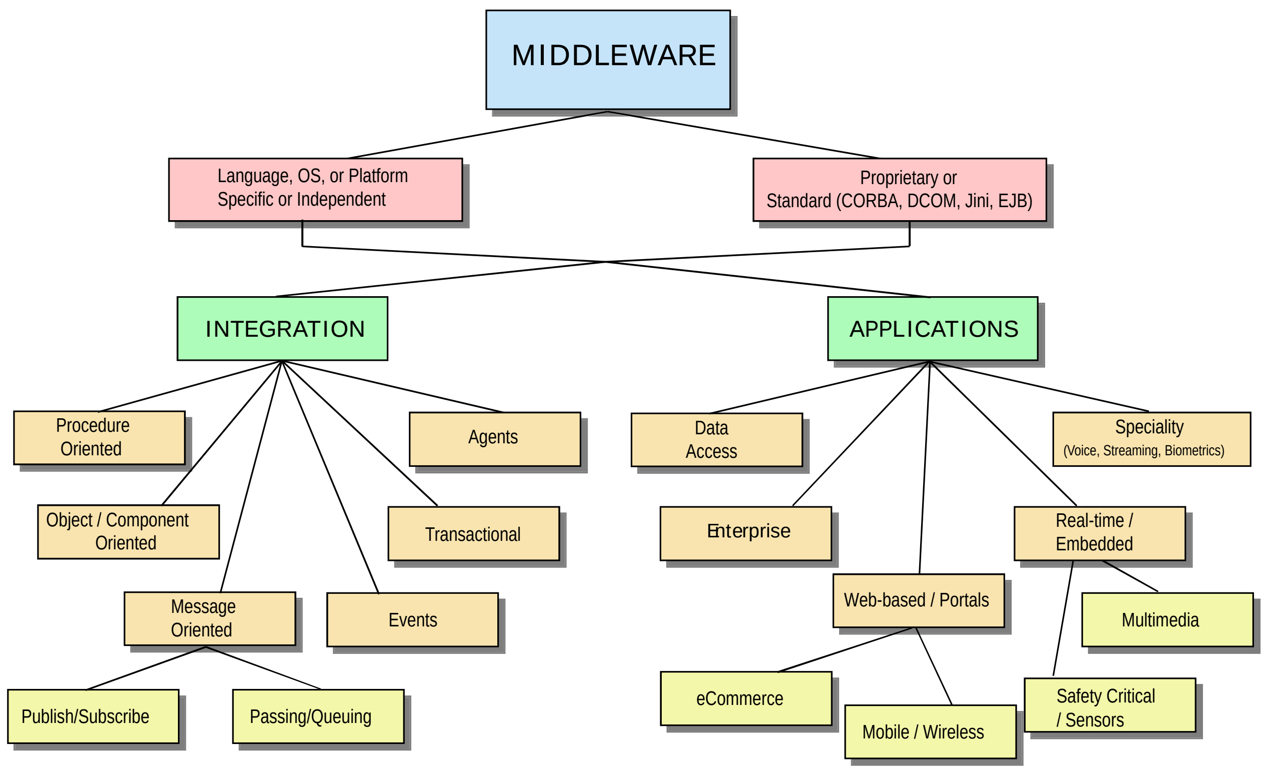

Figure 3. Categories of Middleware

1.3 Web Services

Web services are middleware components using Extensible Markup Language (XML). They are reusable components that can be accessed by multiple clients at the same time; or allow two or more web-applications to communicate with each other; or can be used as a glue to patch new applications / services with legacy applications. They are self contained units of functionality exposing well defined and precise interface to receive or generate messages. They are registered with a directory service or a registry and discovered by users [Violino 2003, Karastoyanova 2003].

Web services architecture allows a decentralized computing model in which interactions between the web service components can happen over distributed domains (different host machines in the network) using existing Internet technologies without much reengineering in comparison to earlier component models like CORBA, DCOM, COM+ and others where the component ownership generally resides in a single trusted domain like an enterprise Intranet [Ganesh et al., 2004].

It is this decentralized deployment option, which can be exploited in an RFID infrastructure, where the capabilities of sub-domains of different value chain partners of an enterprise can be utilized to provide real-time access to data as an integrated seamless environment. The decentralized model also bestows the architecture a unique advantage of maintaining, updating and adding the web services individually without disrupting the existent infrastructure, which may prove to be very valuable in RFID networks as new technology, protocols and standards will be introduced and also the architecture may be very dynamic.

Thus, it is expected that web services will play a pivotal role in helping RFID adopters to integrate RFID based applications into existing enterprise applications such as logistics, ware house management, inventory management, supply chain management, etc., and enable sharing of up to date if not real-time data about the tagged objects' location and shipping events and history, enabling quick decision making. Other important feature of a web service is its capability to be deployed to communicate between machines and to react to event triggers automatically in the background taking care of such tasks such as event logging, event verification, etc., most of the times not requiring any immediate attention of the operators, again a feature which will add substantial advantage to RFID based solutions.

It is this feature of web services, in addition to remoting and Windows services that will be examined and investigated in WinRFID – RFID middleware at WINMEC. These services will be used as building blocks of the WinRFID.

2.0 RFID Middleware

RFID middleware is a new breed of specialized software that sits between the RFID hardware (readers) and the enterprise applications or conventional middleware. The main goal of this middleware is to process data from tags collected by the readers deployed in the RFID infrastructure, or to write ID numbers and/or business process data to the tags while commissioning of these tags for assigning to individual items. In addition, it deals with a number of important issues related with avoidance of data duplication, mitigating errors and proper presentation of data. RFID middleware is being developed and made available by some software vendors on a service basis to suppliers of large retailers such as Wal-Mart, Target, and others, U.S. Dept. of Defense, and pharmaceutical companies who have to meet mandated deadlines in 2005, which requires tagging at pallet and carton levels. These vendors also conduct pilots and proof-of-concept projects for the suppliers.

According to a latest research report by Venture Development Corporation companies planning to implement RFID are mainly worried about data quality and data synchronization. Many of the research survey respondents indicated they were experiencing difficulty extending their RFID pilots because their legacy systems were not able to process the vast amount of information generated. In addition there were problems such as large number of missed tags producing a high volume of false negatives and readers reading the tags multiple times generating duplicate data [VDC 2004, Walker, et al., 2004].

RFID middleware will play a large role in reducing these problems and eventually in mitigating them. This is vouched for by the VDC survey and it is expected that the RFID middleware market will grow by 162% in the year 2005, from $16.4 million in 2004 to $43.1 million in 2005. Middleware will account for roughly 3% of RFID systems revenues in 2007, or $135 million [VDC 2004, Michielsen 2004]. This shows that RFID middleware will be a significant software suite in times to come.

All these issues will be dealt with in the sections describing the middleware research being undertaken at WINMEC.

2.1 Benefits of RFID

Benefits of deploying RFID will begin to accrue in phases. This is because of the scope of various mandates and the degree of compliance required. A clear ROI is still disputed, but obvious payback is expected in the form of increased labor, facility/equipment productivity; process improvements throughout the supply chain; reduction in theft and reduced inventory; and other benefits. The accompanying chart in figure 4 illustrates the various stages and the time line during which a number of enterprise activities will be benefited [A.T. Kearney 2003, A.T. Kearney 2004, Frontline-Sep 2004, Chappell, Durban, et al., 2003 Chappell, Ginsburg, et al., 2003, Boushka 2003, eForce 2004, Lee, et al., 2004].

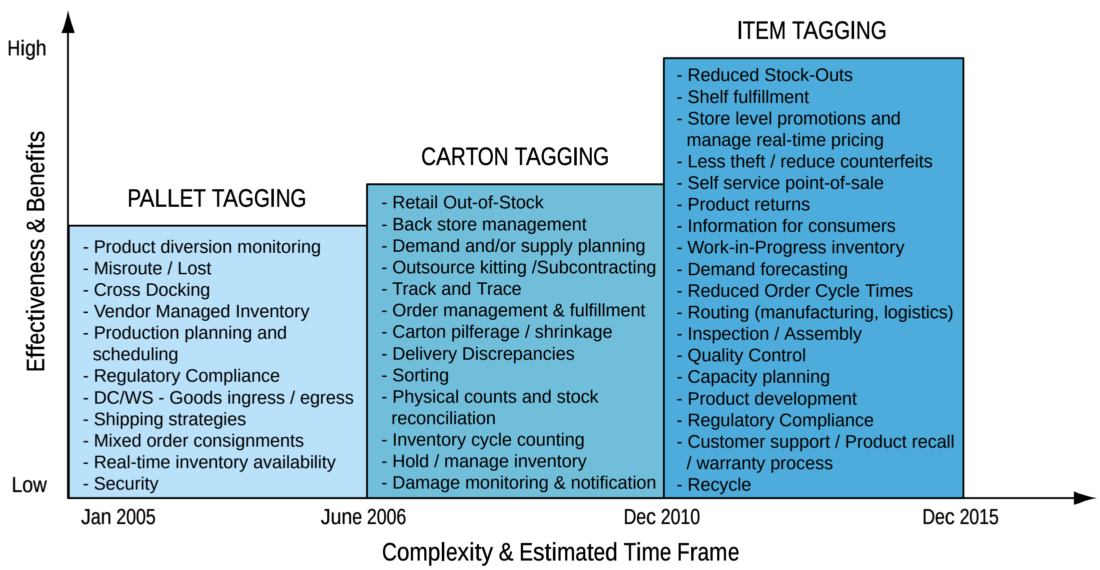

Figure 4. RFID benefits across supply chain activities

A good middleware solution will be greatly impacted by the technology adoption rate. Benefits will increase as tagging goes to the item level and at the same time it is expected to burden the middleware design. The design adopted by us in this work demonstrates that the solution framework can evolve with the increase in degree of difficulty of the adoption.

So, it is without doubt RFID would be a very disruptive technology with the potential to drastically change a number of business practices affecting a large number of industry verticals, and some of these verticals will be seeing exceptional benefits. And middleware architecture will be most appropriate as only such a distributed architecture would successfully wrap the differences and nuances of the technology and allow the dynamic character of this infrastructure to flourish as against a monolithic application. However, there are a number of challenges as laid out in the next section.

2.2 Challenges in adopting RFID

It is anticipated that the impact of introducing RFID in an enterprise IT infrastructure would be felt on many fronts but two main ones would be on the network and sharing of data generated by the RFID systems. The challenge posed will be from the point of integration of the RFID infrastructure with the existing IT network and the potential business process transformation enterprises would have to entertain to gainfully exploit the benefits of RFID [Fornicio 2004, Frontline-Aug 2004].

The demand progresses from the readers, which may be deployed at various locations connected to the edge host. The edge host will sieve through all the input data, filter it and glean information from it. The edge hosts will relay the information to intermediate aggregator servers, which in turn update the enterprise repository from where the value chain partners can utilize it for decision making.

Given this scenario, understanding and designing the data infrastructure would require answers to three key points – volume of data the system would generate, locations from which the data will be generated, and where and how long the data needs to be maintained.

The answers vary by the industry verticals. Many issues such as number of partners who will be using the data and the format in which they will require the data, regulatory mandates, granularity of tagging, and the degree of distributed nature of the RFID network would further impact the answers to the above needs.

In a typical RFID network it is anticipated a few hundred (some may require thousands) readers, tens of edge hosts and a few aggregators would constitute the RFID network infrastructure. At each of these nodes in the network, data in different formats and different quantities would reside at any given point. The amount of data at each of the node would probably be fairly limited. But it is speculated that the aggregation of data from many of these nodes would create large data volumes. Thus, the challenge lies in developing a distributed network to gather data from a large number of independent and fairly tiny data sources. This may also require an intelligent framework to aggregate and cross index the tiny data sources to enable the users use only the information of interest to them. The tiny distributed data sources would also be significant source of failures due to possible outages, requiring sufficient data redundancy to be built in the network architecture or to implement a mechanism to recover data after failure for seamless data visibility at all times.

In addition, any software solution that is developed today will at best be a stop-gap arrangement as the technology itself is evolving. New technologies are being introduced with different RF physics, transmission schemes, supporting different frequency bands, new protocols, new standards, multi-protocol support, changing governmental regulations, etc. Thus, any software solution developed now will have to evolve with the RFID technology – should be extendible and adaptable with minimum disruption to the deployed infrastructure.

Section 3.0 describes the architecture, different frameworks, data layers and integrating strategies being developed and implemented at WINMEC to with an aim to overcome some of these challenges.

3.0 RFID ecosystem research at WINMEC

The WinRFID - RFID middleware research work involves developing new algorithms and data structures, exploring the option of employing new paradigms viz., remote, Windows and web services, which have been used in the case of large distributed applications requiring a degree of autonomy and flexibility. Autonomous services can facilitate the incorporation of reasoning capabilities within the application logic which we think would be an ideal feature for large scale RFID based systems, which require effective use of interoperability between diverse business processes and diverse information and data required to achieve cooperation over the Internet and collaborating business partners.

In the following sections WinRFID architecture and some of the main modules will be discussed, highlighting the technological aspects that benefit a large distributed RFID ecosystem, and how it may alleviate the challenges delineated in section 2.2.

3.1 Architecture of WinRFID

WinRFID is a multi-layered middleware developed using .NET framework. There are five main layers. The first layer deals with the hardware – readers, tags and other sensors. The second layer abstracts the reader-tag protocols. Above that lies the data processing layer, which deals with processing the data streams generated by the reader network. Fourth layer constitutes the XML framework for data and information representation. The top layer deals with the data presentation as per the requirements of the end-users or different enterprise applications.

The communication, management, aggregation, formatting and customization of data, messages and information between these layers is marshaled by supporting services and modules such as the business rule engine, intelligent remote objects and coordinators and some libraries. Figure 5 shows the overall architecture of the middleware.

The functional and operational features of each of the layers will be described in the following sections.

3.2 Physical Layer – RFID Hardware

This layer deals with the abstraction of three elements of the RFID infrastructure, viz. readers, tags and the I/O module of the readers. The abstraction is so designed that it makes it very simple to derive any new specific reader, tag or I/O class extend the middleware capabilities in the advent of introduction of new RFID technology.

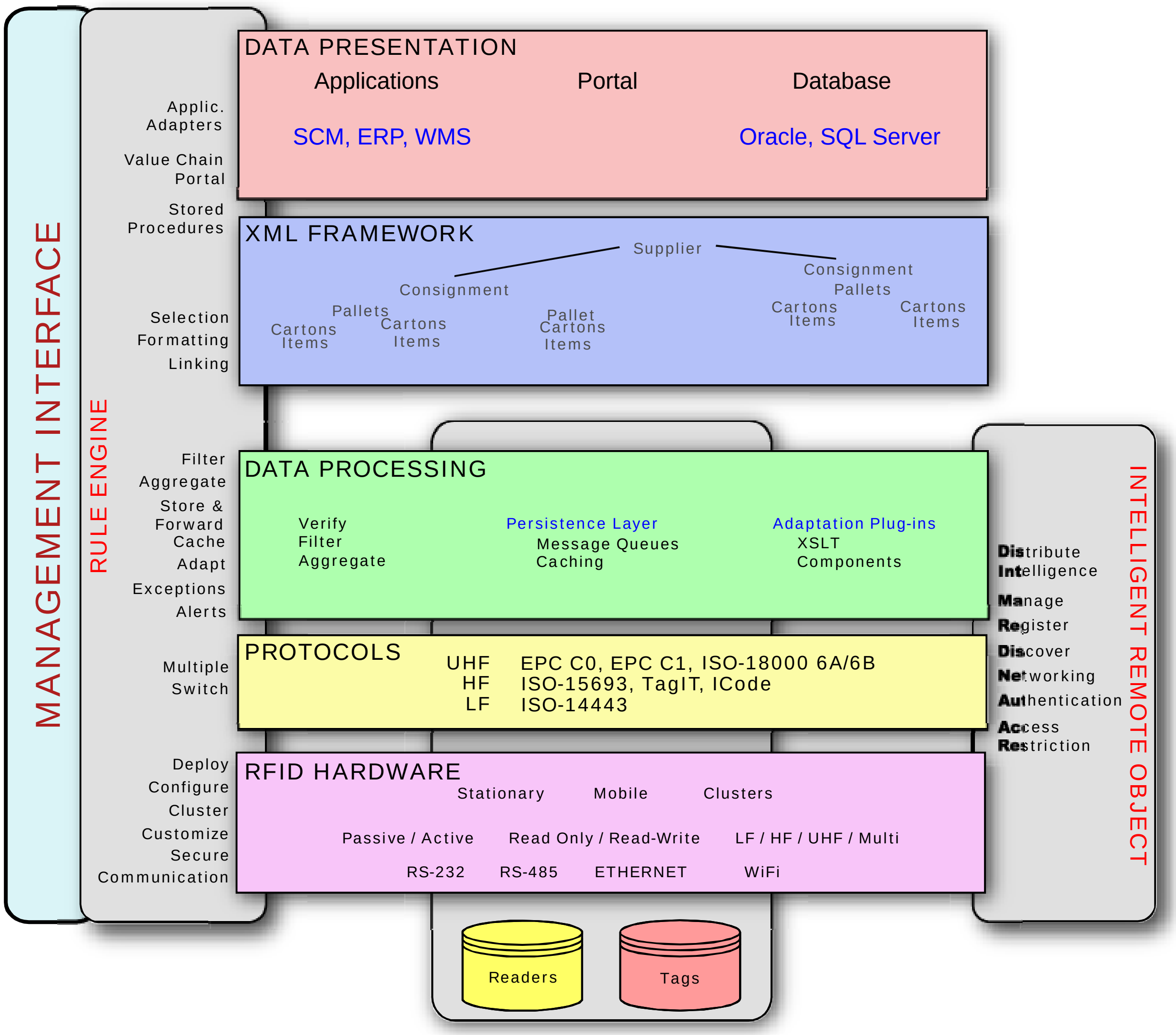

Figure 5 WinRFID Middleware Stack

The reader object assists in management, configuration, location assignment, associate tag protocol(s), security, and the interface for command engine or the dedicated API/SDK provided by the vendor. The reader object supports the requirements of physical readers, which can be stationary, mobile or handheld or even clusters of readers and facilitates their integration into the infrastructure agnostically. This feature addresses the idiosyncrasies of the readers such as read mode, support for number of antennas, command structures to interact with the tags, etc., by providing a common interface with high level methods to carry out these tasks. The reader object also provides a wizard interface to manage and configure the reader during deployment and after. Currently readers operating on LF, HF, and UHF frequencies; both stationary and mobile are supported for passive RFID. Support for active technology operating at 415MHz and 802.11b based RFID is being implemented.

The tag object abstracts the operations and payload formats such as access designated areas of memory, read, write, ID structure, and the syntax of the commands or the API calls. It also provides a common interface to tags that are read only, read/write, passive, or active and working on different frequencies. To the user of this object the top level operational methods wrap around the individual commands or the API calls specific to the tag type just like in the case of the readers. Support is built-in not only for reading and writing tag ID but also read/write user or process-data on to tags that have additional memory. With each type of tag is associated a protocol and the tag object maps the tag operations to the protocol syntax and semantics. Currently support is available to tags of all frequencies and protocols listed in Table 1 above sans the Microwave range.

The next component in this layer is the I/O object. This object abstracts the functionality of different I/O protocols such as RS 232, RS 485, TTL and Ethernet, which are currently used for communications between the RFID readers and the edge hosts. Support is also built in for communication protocols such as HTTP, Telnet and TCP. So, using the reader wizard the I/O module of the reader can be switched from one to other based on the physical I/O connection employed.

In the next section we will see how components in this layer interface with the protocol component in the next layer of the middleware.

3.3 Protocol layer

In the case of a comprehensive RFID middleware, support for multiple tag protocols and the capability to add new ones as they become available is imperative. To facilitate this, in WinRFID, the protocol component is also abstracted to wrap the command syntax and semantics of a variety of published protocols such as ISO 15693, ISO 14443, ISO 18000 – 6 A/B, ICode, EPC Class 0 and EPC Class 1. It deals with protocol specifics such as byte-based, block or even page reading and writing, structure and length of the command frames, partitioning of the tag memory space, checksums, etc.

The essence of this layer is the protocol engine, which will parse and process the raw data from the tags in accordance with any particular standard protocol as mentioned above. The physical layer or the reader object subscribes to the protocol parser service with the type of protocol the reader will have to communicate with for the tags it will negotiate. This is done using the configuration wizard of the reader object.

When the data is parsed using the selected protocol by the reader, it is still in a raw format and is passed on to the data processing layer for further processing.

3.4 Data Processing layer

Given the state of the current RFID technology, the read and write operations in a reading area is influenced mainly by the tag density, read / write distance from the reader antenna(s), orientation of the tags, material of the item which is tagged, and spatial resolution between tags (closeness of tags to each other).

Many of these characteristics introduce inconsistencies in reading or writing such as multiple reads of the same tag, some tags not being read, erroneous reads, etc. These issues are addressed by having processing rules which will weed out duplicate reads, verify the tag reads, and when advanced records are available such as advanced shipping notices, this layer reconciles the records with the tag reads. Any discrepancy is processed as exceptions and a variety of alerting systems are available for resolution – emails, messages, or user defined triggers.

At the same time due to the business process requirements the reading of the tags may have to be intelligently processed. Demands such as consolidation of carton and pallet information from a particular supplier or vendor, information of particular product, data about items passing through a dock / ware house door, items passing during a particular time frame, reads from particular reader or cluster of readers, etc., are quite common place in warehousing, supply chain and logistics. This capability is quiet challenging as individual tagged items (cartons or pallets) from a consignment may arrive at the designated warehouse or distribution center in a staggered manner over a period of time through multiple inward doors by different transport mediums, but the middleware will have to keep tab on all the items received and outstanding, and reconcile the consignment contents and provide a status view when queried. In this case specific rules can be incorporated to aggregate and classify the data accordingly and made available in a variety of formats, which will be discussed in the next section. In this layer provision is also made to log the activities based on user selectable criteria.

In addition to these features, there is a data persistence component, which provides a local data store. This component is based on message queues. This design facilitates asynchronous processing of the data streams coming from the lower layers of the middleware and gives sufficient time for the above rules to work on the raw data and covert or adapt them for the upper layers or subscription services.

The intelligence to define these requirements and process it accordingly is built in this layer. This is achieved by way of a customizable business rule engine and a framework for adding custom data adaptation plug-ins. Discussion on features of these modules follows later.

3.5 Extensible Markup Language (XML) Framework

The raw cleaned (verified and filtered) tag data from the physical layer data streams is formatted in a variety of ways to a high-level XML based representation. The information is filtered, cleaned, aggregated and adapted as per the custom plug-ins, which can be added to the middleware services. The attempt is to provide data in a format amenable to decision making at the application layer as shown in figure 5.

The layer is supported by default templates and tag libraries using which the raw data is pulled from the message queues in the data processing layer. The data from the queues is corralled based on the rules or plug-ins as described in various criteria (particular supplier or vendor, particular product, etc.) in the previous section. The data from the framework can be published to registered connectors or connectors can subscribe to specific data. The data is designed in such a way that it is possible to easily search containers based on key fields such as suppliers, vendors, order number, consignment type, etc. The search pointers are required to set up the correct link between data sources and the connectors.

We expect the XML based representation would facilitate the ease of data consumption by enterprise applications such as warehouse management, supply chain management, enterprise resource planning and others. This is because most of these systems have adapters to export XML based data and wizards to design templates to parse the XML tree within these systems are also quite mature.

This framework can be deployed as in-memory database or native XML database.

3.6 Data Presentation Layer

This is the application layer and it gets the data for visualization and decision making from the XML framework. Currently, we have considered only the portal and the database connectors. We are attempting to link the XML framework with SharePoint server to provide a portal interface. The main feature of this portal would be capability to set up secure subscriber (for e.g. a value chain partner in a supply chain) accounts with complete authentication and access control. Each subscriber can then subscribe to the information of his interest. The data delivery format can be default as provided by the middleware or the subscriber can register data adapter plug-ins as web parts in SharePoint. Each supplier can also provide access rights to these web parts for other subscribers to share from the community. All such web parts would be available through a library.

Other features of the portal are plugging the RFID data into graphic visual widgets for presentation. This would be extended to provide data in other denominations like charts, graphs, etc. From each of these widgets the portal will let the subscribers to make decisions like trigger events for re-routing, re-assign, billing, alert, etc.

The other connector is the database connector. Currently the middleware can populate SQL Server and Oracle RDBMS. The databases get populated in an asynchronous fashion in a trickle mode – a process with least priority so as to avoid the edge hosts getting locked up. Priority of the resources is skewed towards processing the activities of the lower three layers as shown in figure 5 and the upper layers being catered to in the background at lower priority.

3.7 Services in the WinRFID

Using the Window Services, Web Services and Remote Objects based modules, middleware modules are deployed as independent components which run as self-contained modules in dispersed machines. These services run unattended with or without a user interface, run within their own process space and can start up during the OS boot process. They can be configured and managed over the Net and can be set up in the polling mode (service monitors applications) or event mode (application sends events).

The following sections provide a glimpse of the main services of WinRFID, which impart dynamic distributive control to the middleware.

3.7.1 Reader Windows Service

This service is hosted by the edge host in the WinRFID network. We have used Windows service here because of its better performance using TCP channel for communication in comparison to web services. It also provides the container services for the remote object based reader coordinator.

It is used in monitoring the physical connection of the readers to the edge hosts; health of the readers, which is monitored at predefined intervals; authenticates the readers based on the approved reader repository; and above all provides a means for other applications to discover and interact with it. Windows services' feature of remote activation during the O/S boot up is also a benefit in an RFID network (with large number of edge hosts with readers) as the service boots up when the edge host is remotely booted in case of any service breakage or the system needs to be reset.

3.7.2 Remote object based service

This service is built on the .NET remoting framework. It acts as the coordinator at the edge hosts for managing the readers physically connected to the edge host. This service allows the coordinator to directly interact with applications or services running remotely on other machines. It provides features such as activation and transactional lifetime support, communication channels (TCP or HTTP) for passing data and messages, etc. It allows formatters for encoding and decoding data and messages and these can be used to provide security to the content before they are transported over the communication channel. Thus, the main advantage of this framework is that it allows secure custom binary encoding for the payload which reduces the size of the content transported over the network, as against the bulkiness of the payload in a web services based solution as the content and messages are encoded in XML.

These features of the service are exploited for a variety of functions of WinRFID such as reader deployment, configuration and management; allow clients to subscribe to different formatters for events and data; and consolidated functions such as management of read and write cycles using multiple antennas connected to a particular reader, clustering of readers and aggregate the data stream, plug-in hooks for data subscription from edge hosts, portals or applications. This concept is shown in figs. 6 and 7.

3.7.3 Reader Web Service

Web services technologies are based on standards and use XML based languages for message and data passing, and platform support services framework based on Universal Description, Discovery and Integration (UDDI) for subscription, discovery, transactions, etc. They provide platform, O/S and programming language transparency to the distributed system.

In WinRFID the reader web service is functionally equivalent to the above two services put together. This service allows readers connected to any platform (UNIX, Linux, etc. with say Java programming environment) that support the standard web services framework. Simple Open Access Protocol (SOAP) is used for payload transactions – service request and response, and Web Service Description Language (WSDL) [details found elsewhere W3C, MS-WSDC] is used for service description, discovery and query responses.

Currently, in WinRFID we have integrated mobile readers attached to WIN CE / Pocket PC PDAs and use the reader web service to transact with the reader over 802.11b and Bluetooth wireless connectivity. Since, PDAs are computationally challenged and do not support remote object references, in WinRFID we use the web services and pass on all the intensive processing from the PDA.

3.8 Rule Engine

The philosophy of a rule engine is solving a problem using a set of logical rules specific to the problem domain. They have been very popular in solutions requiring processing of large sets of rapidly changing data – like in an RFID network. The other main advantage is that the rules can be flexibly updated and modified by even end users as and when the system requirements change, without requiring the services of system developers.

Driven by these benefits, and the nature (large sets of data from multiple sources) and status (changing standards, protocols) of the RFID technology we have designed a tightly coupled rule engine into the architecture of WinRFID. An attempt is made to craft the architecture in a flexible way to the extent that it provides a means to even the end users to incorporate their own rule chunks by way of 'plug-ins', the concept of which is described in the next section. The 'inference engine' of the rule engine is based on forward chaining (data driven) as defined by RuleML [Boley 2001, Boley, et al., 2001, Wagner 2002].

As per the design, the rule engine influences a number of processes and activities of WinRFID. As shown in figure 6, a number of processes such as raw data filtering, aggregating, exception handling and alerting at the edge node, and data adaptation, options to publish or subscribe the data to and from the enterprise applications and others are all driven by the rules. The main objective of using a rules system in WinRFID is to convert data and messages from lower layers to *actionable information* for the upper layers; based on the business or process semantics as perceived by the user of the information.

This module is being refined with different sets of rules, and also being tested for accuracy, reliability, integrity and performance.

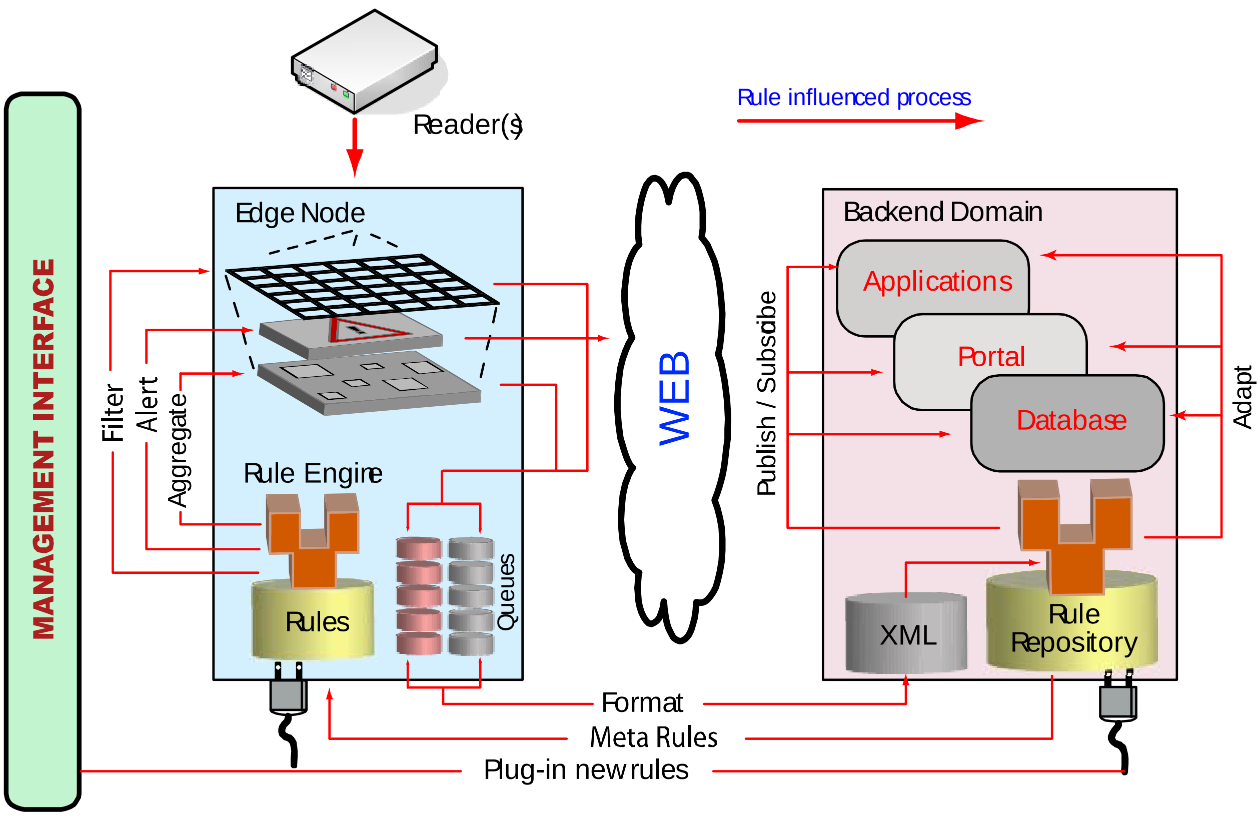

Figure 6. WinRFID Rule Engine Architecture

3.9 Extendibility through Plug-ins

The .NET framework facilitates adding of functionality to the applications through runtime plug-ins easily. This feature is exploited in WinRFID for adding new reader modules to the remote object service, protocol modules to the system repository, data adaptation plug-ins in data processing layer, and can be extended to add other sensors or auto-ID technologies at the physical layer. We expect this feature will help refine and reinforce WinRFID with minimal rework as an when readers, new protocols, and standards become available. The major benefit of this feature is these modules or assemblies as they are called in .NET can be added to existing infrastructure with minimum disruption if not totally disruption free. Figure 7 shows the concept of WinRFID plug-ins as assemblies of .NET framework.

For this feature to work the added assembly will have to the discovered, which can happen at run time. The discovery can be facilitated in a number of ways but we employ two methods. First, an XML configurations file with the information of the assembly to be used – its name, location and methods to activate is registered in a registry. This method is used in adding new reader, tag protocol or business rules. This is the method employed in WinRFID currently. Second method uses reflection (explanation of reflection can be found in Liberty 2003, MSDN) where the discovery method can be automated by storing the plug-in in publicized location relative to the application directory.

In WinRFID, plug-ins framework allows external value chain partners to add data adaptation plug-ins to transform the XML data to the required format, add new reader objects, and add business rule chunks.

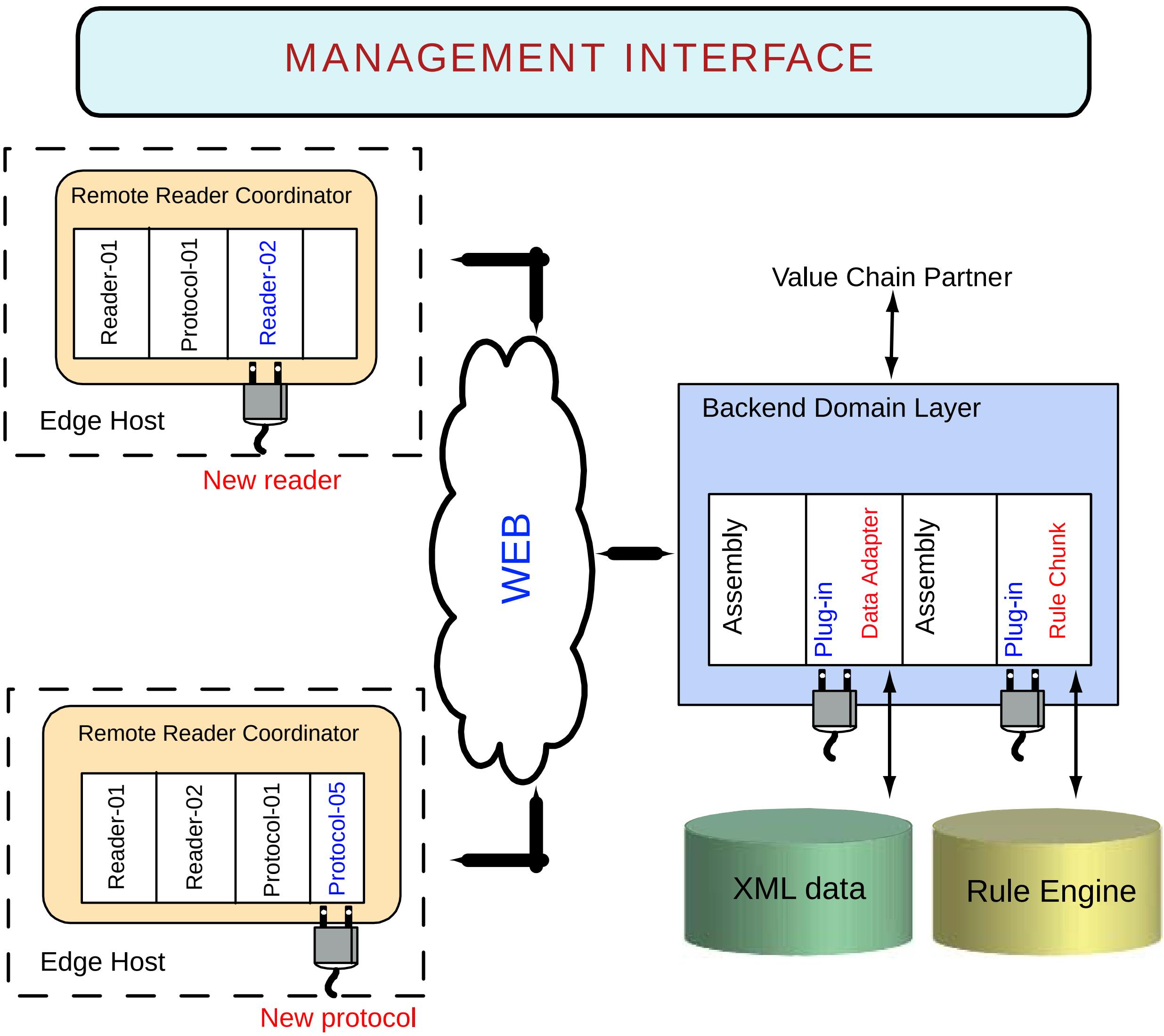

Figure 7. Plug-ins concept for extending WinRFID

With this feature of WinRFID, we are working on simulating an EPC network with the functionality of each of the modules of the network – savant, naming, discovery, information and trust embedded in WinRFID's services. Based on the success of this exercise it would be possible to experiment with EPC and non-EPC technologies working together and exploiting their synergies.

4.0 Summary

WinRFID is an RFID-technology agnostic middleware and a holistic distributed application. The design of the architecture is federated with the functional, system, business and process knowledge residing in self contained software units - the different services providing a variety of independent and complementary capabilities.

Our experience has been encouraging from the point of view of system specific features such as reliability, extendibility, scalability and ease of use. This has been possible due to the architecture providing access to end users right up to the remote object based reader coordinator on the edge of the network, and customize it even during run-time by injecting new process rules, adding new hardware, support new protocols and standards. In our view this feature is very important in order to assimilate and sustain the number of changes across the entire gamut of RFID technology that is expected, as well as effortlessly deploy a solution exploiting the best of breed RFID technology, catering to the needs of sought solutions for a variety of enterprise verticals. As a result, it is our opinion that RFID technology rollouts and internalization will be a journey, not a destination for quite some time.

From the point of view of incorporating business process knowledge and semantics we feel confident that the rule-based engine will prove its worth as it is very flexible and facilitates an apt description of the process activity syntax and semantics which can blend with the RFID data to assist quick decision making.

We have been successful in supporting a substantial number of useful reader-tag technologies, protocols and standards, and providing a transparent tier from which enterprises can build solutions, focusing only on the end application. The middleware has been deployed on an RFID testbed at WINMEC and various experiments conducted [http://www.wireless.ucla.edu/rfid/research/].

In the next phase support for a reconfigurable sensor platform will be added [Ramamurthy, et al., 2004, http://www.winmec.ucla.edu/rewins/]. We believe that RFID in conjunction with a variety of sensors (temperature, pressure, chemical, motion, etc.) will provide a big value-add to a number of supply chain, security and logistics activities, enabling new effective business models and improving existing practices multifold.

5.0 References

A.T. Kearney, 2003, Meeting the Retail RFID Mandate - A discussion of the issues facing CPG companies. *A research analysis report*, ***November***, A.T. Kearney, Inc. Chicago, IL, USA.

A.T. Kearney, 2004, RFID/EPC : Managing the transition (2004 - 2007). *A research analysis report*, ***August 1***, A.T. Kearney, Inc., Chicago, IL, USA.

Agha, G, 2002, Adaptive Middleware. *Commun. ACM* ***45***, 6, 30-32.

Agha, G., Frølund, S., Kim, W., Panwar, R. Patterson, A., and Sturman, D., 1993, Abstraction and modularity mechanisms for concurrent computing. *Parallel Distrib. Tech.* ***1**,* 2, 3–15.

AIMGlobal, 1998, A Study of Data Carrier Issues for the Next Generation of Integrated AIDC Technology. *A Research Study Report*, AIM, Inc, Warrendale, PA, USA. [http://www.aimglobal.org/technologies/rfid/resources/dcstudy/datacarrier_study.htm]

AIMGlobal, 2000, Draft Paper on the Characteristics of RFID-Systems, ver. 1.0. Aim Inc., ***July***, Warrendale, PA, USA.

Astley, M., Sturman, D., and Agha, G., 2001, Customizable middleware for modular distributed software. *Commun. ACM* ***44**,* 5, 99–107.

Barbosa, A.C.P, and Porto, F.A.M., 2001, Configurable Data Integration Middleware System. *Proc. of Intl. Workshop on Information Integration on the Web – Technologies and Applications (WiiW 2001)*, April 9-11, Rio de Janeiro, Brazil.

Bishop, T., and Karne, R., 2003, A Survey of Middleware. *Proc. of 18th Intl. Conference on Computers and their applications*, March 26-28, Honolulu, Hawaii, 254-258.

Boley, H., 2001, The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations. *Web Knowledge Management and Decision Support at the 14th International Conference on Applications of Prolog, INAP 2001*. Tokyo, Japan. October 20-22

Boley, H., Tabet, S., and Wagner, G., 2001, Design Rationale of RuleML: A Markup Language for Semantic Web Rules. *Proc. of Intl. Semantic Web Working Symposium (SWWS'01), Infrastructure and Applications for the Semantic Web*, Stanford, CA, USA, July 30 - August 1, pp. 381-401.

Boone, G., 2004, Reality Mining: Browsing Reality with Sensor Networks. *Sensors* ***21***, 9, September.

Boushka, M., Ginsburg, L., Haberstroh, J., Haffey, T., Richard, J., and Tobolski, J., 2003, Auto-ID on the Move -The Value of Auto-ID Technology in Freight Transportation. *Research report*, Accenture, February.

Brandel, M., 2003, Smart tags, high costs, *ComputerWorld*, Dec. 15.
[http://www.computerworld.com/softwaretopics/erp/story/0,10801,88130,00.html

Brewin, B., 2003., RFID users differ on standards. *ComputerWorld*, Oct. 27.
[http://www.computerworld.com/softwaretopics/erp/story/0,10801,86486,00.html]

Britton, C., 2001, Classifying Middleware. *The Business Integrator Journal* ***Winter***, 27-30.

Cardullo, M. W., 2004, Genesis of the versatile RFID tag. *RFID Journal*, April 24.
[http://www.rfidjournal.com/article/articleview/392/1/2/]

Carvalho, H.S., Murphy, A.L, Heinzelman, W.B., and Coelho, C.J.N., 2003, Network-Based Distributed Systems Middleware. *Proc. of Intl. Middleware Conference*, June 16-20, Rio de Janeiro, Brazil, 13-20.

Chappell, G., Durdan, D., Gilbert, G., Ginsburg, L., Smith, J., and Tobolski, J., 2003, Auto-ID on Delivery: The Value of Auto-ID Technology in the Retail Supply Chain. *Research report*, Accenture, February.

Chappell, G., Ginsburg, L., Schmidt, P., Smith, J., and Tobolski, J., 2003, Auto-ID on Demand: The Value of Auto-ID Technology in Consumer Packaged Goods Demand Planning. *Research report*, Accenture, February

Colyer, A., Blair, G., and Rashid, A., 2003, Managing Complexity in Middleware. *Proc. of Second AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, Boston, Massachusetts, 21-26.

eForce, 2004, RFID Solution Lifecycle Management, Presentation by eForce IT team. [http://www.eforceglobal.com/ppt/eF_RFID_Assesment_Overview.ppt]

Emmerich, W., 2000, Software Engineering and Middleware: A Roadmap. *Proc. of the Conf. on the Futures of Software Engineering*, June 4-11, Limerick, Ireland, 117-129.

Enciu, G, 2002, Applications of RFID in Electromechanical Systems. *Proc. of Intl. Sym. on Electrical Engineering*, June, 3-4, Valahia University of Targoviste, Romania.

Evilsizer, J., Patel, B., Robles, D., and Mintz, M., 2004, Radio Frequency Identification (RFID), *INFO3229 Business Data Communications Technical Briefs*, Belk College of Business, Univ. of N. Carolina, Charlotte, Spring.

Federal Communications Commission (FCC), 2001, Part 15 – Radio Frequency Devices. The Office of Engineering and Technology, Federal Communications Commission, U.S. Government, Oct. 1.

Finkenzeller, K., 2003, *RFID Handbook : Fundamentals and Applications in Contactless Smart Cards and Identification, 2nd ed.* John Wiley and Sons, West Sussex, England.

Floerkemeier, C., and Lampe, M., 2004, Issues with RFID usage in ubiquitous computing applications, *Draft Technical Paper submitted for publication*, Computer Science Dept., ETH Zurich.
[http://www.vs.inf.ethz.ch/publ/papers/RFIDIssues.pdf]

Fornicio, H., 2004, Middleware Filters RFID Data. *Managing Automation Magazine* ***19***, 2, March.
[http://www.managingautomation.com/maonline/magazine/read.jspx?id=196620**]**

Frølund, S., 1996, *Coordinated Distributed Objects: An Actor-Based Approach to Synchronization.* MIT Press, Cambridge, MA

Frontline-Aug., 2004, Confusion still common for RFID users. *Frontline Solutions* ***5***, 8.
[http://www.frontlinetoday.com/frontline/article/articleDetail.jsp?id=110305]

Frontline-Sep, 2004, RFID benefits will come in phases. *Frontline Solutions* ***5***, 9, September.
[http://www.frontlinetoday.com/frontline/article/articleDetail.jsp?id=122209]

Gadh, R., Aug. 2004, The State of RFID: Heading Toward a Wireless Internet of Artifacts. *Computerworld*, August 11.

Gadh, R., Oct. 2004, RFID: Getting From Mandates to a Wireless Internet of Artifacts. *Computerworld*, October 4.
[http://www.computerworld.com/mobiletopics/mobile/story/0,10801,96416,00.html]

Ganesh, J., Padmabhuni, S., and Moitra, D., 2004, Web Services and Multi-Channel Integration:A Proposed Framework. *IEEE International Conference on Web Services (ICWS'04)*, June 06 - 09, San Diego, California, 70-79.

Geihs, K., 2001, Middleware Challenges Ahead. *IEEE Computer* ***34***, 6, 24-31.

Gimenez, G., and Kim, K.H., 2001, A Windows CE Implementation of a Middleware Architecture Supporting Time-Triggered Message-Triggered Objects. *Proc. of 25th Annual International Computer Software and Applications Conference (COMPSAC'01)*, October 8 - 12, Chicago, Illinois, 181-189

Hartwich, C., 2003, *A Middleware Architecture for Transactional, Object-0riented Applications*. Ph.D. Thesis, Fachbereich Mathematik u. Informatik, Freie Universität Berlin, Nov.

IBM Wireless e-Business Group, 2003, RFID in Business Processes. *A Thought Leadership Technical Paper*.
[http://www.ibm.com/industries/wireless/doc/content/bin/SmartTagsInformationv.1.0.pdf]

Karastoyanova, D., and Buchmann, A., 2003, Components, Middleware and Web services**,** *Proc. of the IADIS International Conference WWW/Internet (ICWI2003)*, Algarve, Portugal, November 5-8.

Kelkar, A., and Gamble, R.F., 1999, Understanding the Architectural Characteristics behind Middleware Choices. *Proceedings of the 1 st Int'l Conference in Information Reuse and Integration*, ed. Rubin, S., Nov.

Kellam, L., 2003, P&G Rethinks Supply Chain. *Optimize Magazine* ***Issue 24***, Oct.
[http://www.optimizemag.com/printer/024/pr_supplychain.html]

Kern, C., 1999, RFID-Technology – Recent Development and Future Requirements. *Proc. of European Conference on Circuit Theory and Design (ECCTD '99)*, 29 Aug. - 2 Sep., Stresa, Italy.

Landt, J., 2001, Shrouds of Time – The History of RFID. An AIM Publication, Oct.
[http://www.aimglobal.org/technologies/rfid/resources/shrouds_of_time.pdf]

Lee, Y. M., Cheng, F., Leung Y. T., 2004, Exploring the Impact of RFID on Supply Chain Dynamics, *Proc. of 2004 Winter Simulation Conference (WSC'04)*, Volume 2, December 5-8, 2004, Washington D.C., 90 - 97.

Lewis, J., 2003, RFID: Small Package, Big Problem. *Course Paper : Information Security & Cryptography*, Dec., Information Security Laboratory, Dept. of Electrical Engg. and Computer Science, Oregon State University, USA. [http://islab.oregonstate.edu/koc/ece399/f03/final/lewis2.pdf]

Liberty, J., 2003, *Programming C#*. 3rd ed., O'Reilly Media, Inc., Sebastopol, CA, USA, May.

Lowe, W., and Noga, M.L., 2002, A Lightweight XML-based Middleware Architecture. *Proceedings of 20th IASTED Intl. Conf. Applied Informatics (IASTED AI 2002)*, ACTA Press, Feb 18-21, Innsbruck, Austria, 131-136

McFarlane, D., 2002, Auto-ID Based Control : An Overview. *White Paper*, AUTO-ID Center, Feb.

Meer, S., van der, 2002, *Middleware and Application Management Architecture*. Ph.D. Thesis, Electrotechnik und Informatik der Technischen Universität, Berlin, September.

Michielsen, E., 2004, RFID Middleware Market Competition Heats up. *RFID International newsletter **V II***, n 6, March 15.

Middleware Domain Team, 2001, A Middleware Framework for Delivering Business Solutions, *Middleware Architecture Report v. 1.0*, The COTS Enterprise Architecture Workgroup, The Council on Technology Services Commonwealth of Virginia, May.

MSDN, Reflection. *MSDN's .NET Framework Technology Samples web site*. [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpsamples/html/reflection.asp]

Ramamurthy, H., Prabhu, B.S., Gadh, R., 2004, Reconfigurable Wireless Interface for Networking Sensors (ReWINS). *Personal Wireless Communications: IFIP TC6 9th International Conference, PWC 2004 [LNCS 3260 / 2004]*, eds., Niemegeers, I., and Heemstra de Groot, S., Delft, The Netherlands, September 21-23.

Sharp, K.R., 2000, A Sense of the Real World. *Supply Chain Systems Magazine **20***, 9, Sept.

Sharp, K.R., 2003, Channel Tuning in the RFID Market. *Supply Chain Systems Magazine **23***, 5, May.

Sutton, S.M., 2000, Middleware Selection. *2nd Intl. Workshop on Engineering Distributed Objects (EDO 00)*, LNCS 1999/2001, Emmerich, W., and Tai, S. (Eds.), Nov., 2-3, Davis, USA, 2-7.

Tripathi, A., 2002, Challenges Designing Next-Generation Middleware Systems. *Commun. ACM **45***, 6, 39-42.

Välkkynen, P., Korhonen, I., Plomp, J., Tuomisto, T., Cluitmans, L., Ailisto, H., and Seppä, H., 2003, A user interaction paradigm for physical browsing and near-object control based on tags. *Proc. of Physical Interaction (PI03) – Workshop on Real World User Interfaces held during MobileHCI 2003*, Sep 8-11, Udine, Italy.

VDC - Venture Development Corporation, 2004, *Radio Frequency Identification (RFID) Middleware Solutions: Global Market Opportunity*, A Market Research Report, Natick, Massachusetts, USA, August.

Vinoski, S., 2002, Where is Middleware ? *IEEE Internet Computing **6***, 2, Mar-Apr, 83-85.

Violino, B., 2003, Linking RFID with Web services. *RFID Journal*, Oct. 6. [http://www.globeranger.com/papers/Article%20-%20RFID%20Journal%20RFID%20&%20Web%20Services.pdf]

W3C, Web Services Activity. *World Wide Web Consortium's Web Services Working Groups and Interest Groups web site*.
[http://www.w3.org/2002/ws/]

Walker, J., Mendelsohn, T., and Overby, C.S., 2004, Vendors Race To Fill A New Void: RFID Middleware. *TechStrategy Research brief,* ***Jan., 26***, Forrester Research, Inc., Cambridge, MA, USA.

Wagner, G., 2002, How to Design a General Rule Markup Language. *Proceedings of the Workshop XML Technologies for the Semantic Web (XSW 2002)*, HU Berlin, Institut fur Informatik, March 24-25.

Want, R., Fishkin, K.P., Gujar, A., and Harrison, B.L., 1999, Bridging Physical and Virtual Worlds with Electronic Tags. *Proc. of ACM CHI'99*, *Conference on Human Factors in Computing Systems*, May 15-20, Pittsburgh, USA, 370-377.

MS-WSDC, Microsoft's Web Services Developer Center. *Microsoft's .NET Web Services Portal.*
[http://msdn.microsoft.com/webservices/]

Zawada, L. H., and O'Kelly, P., 2003, Assess RFID's Transformational Potential. *.NET Magazine* **3**, 11, Oct. 2003

Zhekun, L., Gadh, R., Prabhu, B.S., 2003, A Study of RFID Smart Parts, *Tech Report # UCLA-WINMEC-2003-202*, Wireless Internet for the Mobile Enterprise Consortium, School of Engg. and Applied Science, UCLA, Los Angeles, USA.

Zhekun, L., Gadh, R., and Prabhu, B. S., 2004, Applications of RFID Technology and Smart Parts in Manufacturing, *Proc. of DETC'04: ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Sep., 28 – Oct., 2, Salt Lake City, Utah, USA.