DEVELOPMENT OF JAVA BASED RFID APPLICATION PROGRAMMABLE INTERFACE FOR HETEROGENEOUS RFID SYSTEM

by

MOHAMMED F. M. ALI

**Thesis submitted in fulfillment of the requirements
for the degree of Master of Science**

January 2010

**SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITI SAINS MALAYSIA**

ACKNOWLEDGEMENTS/DEDICATION

During one and half year, I have gained invaluable knowledge that has guided me for my research. Humbly, I acknowledge that my success is not only related to me but its possession to all the people whose guide me, teach me, prayer for me and encourage me. From them, I learned the knowledge, patient, wisdom, humility and also how to gain access to my goals.

Firstly, I would like to thank Assoc. Prof. Dr. Kamal Zuhairi Zamli for giving good advice and sometimes disputable comments during the course of this work. Additionally, I would like to acknowledge the source of funding from Technofund: Building Smart Communities, headed by my co-supervisor Dr Widad Ismail. I would also wish to thank my close friends, both of whom are PhD candidates: Mohammed I. Younis and Muataz H. Salih for their help. I wish them a successful outcome on their PhD research. To all my fellow postgraduates from RFID research group, thank you so much for the numerous stimulating discussions/general advices and help with experimental setup.

I am grateful to all my friends from Universiti Sains Malaysia, for being the surrogate family during the duration of my research. Also, I am grateful to all my friends in my dear wounded country (Iraq), to all of them, thank you so much for continuously contacting me and give your undivided encouragement.

Given the opportunity, I would also like to dedicate my gratitude to the dearest people to my heart, my father - Falah bin Mohammed and my mother - Sana binti Jamel. This thesis is for both of you. Not to forget, my beloved wife Sura binti Alaa', thank you for bearing the time without full time husband. Finally, to all my family members, I am forever indebted for your understanding, support, endless patience and encouragement when it matters the most. You were all like bright candles in the dark days and difficult times, so thank again.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GPS	Global Positioning Systems
RFID	Radio Frequency Identification
TCP/IP	Transmission Control Protocol/Internet Protocol
API	Application Programming Interface
UPC	Universal Product Code
EPC	Electronic Product Code
RFIDTM	Radio Frequency Identification for Tracking and Monitoring
UML	Unified Modelling Language
OS	Operating System
JCAPS	Java Composite Application Platform System
DSP	Digital Signal Processing
UHF	Ultra High Frequency
LF	Low Frequency
GUI	Graphical User Interface
UTP	Unshielded Twisted Pair
JDBC	Java Database Connectivity
OO	Object Orientation
ANSI	American National Standards Institute

Pembangunan Aplikasi Antara Muka Boleh Aturcara RFID Berasaskan Java Untuk Sistem Heterogen

ABSTRAK

Pembangunan applikasi RFID adalah tugas yang amat sukar dan mencabar. Antara kesukaran dan cabaran termasuk ketidakpiawaian perisian dan perkakasan daripada pembekal peralatan, masalah keserasian antara sistem operasi serta kekurangan kepakaran dalam pengaturcaraan RFID tingkat bawah yang sukar dipelajari. Untuk mengatasi masalah ini, satu aplikasi antara muka boleh aturcara (API), dipanggil RFIDTM (RFID Tracking & Monitoring) Application Programmable Interface, untuk Sistem Heterogen RFID telah direkabentuk dan diimplementasi. API ini telah berjaya digunapakai dalam prototaip applikasi menggunakan beberapa jenis dan konfigurasi pembaca aktif dan pasif meliputi Pembaca RF Code Mantis II 433.92MHz, pembaca RF modul cc1020ua 433 GHz Texas Instruments, Pembaca USM 2.45 GHz Contactless Active Integrated Reader (CAIR) dan Pembaca pasif UHF.

DEVELOPMENT OF JAVA BASED RFID APPLICATION PROGRAMMABLE INTERFACE FOR HETEROGENEOUS RFID SYSTEM

ABSTRACT

Developing RFID based applications is a painstakingly difficult endeavor. The difficulties include non-standard software and hardware peripherals from vendors, interoperability problems between different operating system as well as lack of expertise in terms of low level programming for RFID (i.e. steep learning curve). In order to address these difficulties, a reusable RFIDTM (RFID Tracking & Monitoring) Application Programmable Interface (API) for heterogeneous RFID system has been designed and implemented. The API has been successfully employed in a number of our application prototypes including tracking of inventories as well as human/object tracking and tagging. Here, the module has been tested on a number of different types and configuration of active and passive readers including that of RF Code Mantis II 433.92MHz Reader, RF Reader module cc1020ua 433 GHz Texas Instruments, our in-house USM 2.45 GHz Contactless Active Integrated Reader (CAIR) and UHF Passive Reader.

CHAPTER 1

INTRODUCTION

Since ancient times, humans have quested for the improvement of their quality of life. Solutions were sought. They invented the wheel and developed other tools to enhance and facilitate daily work such as in agriculture, irrigation, and trade. With the passage of time, new requirements have emerged. Factories, laboratories, and advanced technology proliferated to meet human needs. Inevitably, the increasing population, together with the increasing human needs, have further led to the spread of business and industries. At this day and age, and as a result of amplified burdens, humans have once again begun to look for ways to reduce such complexities in the field of information exchange and data transfer, for example, among the branches of companies, factories, warehouses, and across suppliers and recipients. Increased complexity has also been noted in the tracking of goods and document information. These reasons have led humans to use modern technology, of which at present is deemed the prime solution for humans' problems.

1.1 Overview

It has been well accepted that technological evolution makes life easier. Developments in electronics and computers can even be considered one of the most wondrous developments on the basis that it supports human life in many aspects. Modern technologies have changed the ways in doing business and work, not only because of their apparent modernity, but also because these technologies keep up with humans' comfortability. Distributed computing, the Internet, global positioning systems (GPS), and radio frequency identification (RFID) can all be considered important technologies to aid life. Manufacturers need to monitor, track, and control environmental equipment, workers, and goods (Khan and Ojha, 2009, Berenyi and Charaf, 2008, Oktem et al., 2008, Nohara et al., 2008), apart from delivering goods to retailers on time.

Similarly, transportation service providers are faced with increasing requirements from customers. They have become concerned with the status of goods in transit. Such requirement is especially important to the food industry, hazardous material management, and others (Garmin, 2009). Furthermore, some applications that require delivery and receipt of real-time information and access of goods play an increasingly important role in decision-making during operations management of a supply chain.

Today, the Internet has become a public, cooperative, and self-sustaining facility accessible to millions of people worldwide. On the physical side, the Internet uses a portion of the total resources of the current existing public telecommunications. Technically, the Internet can be distinguished through the use of a set of protocols called Transmission Control Protocol/Internet Protocol (TCP/IP). Two recent adaptations of Internet technology, the intranet and the extranet, also make use of the TCP/IP protocol (Microsoft, 2008). In fact, the networks and the Internet give a huge advantage with low cost communication for the transfer of electronic information and facilitation of communication services. The Internet strongly influences our society by reducing time and space boundaries in obtaining and communicating information. The impact of Internet services in the economics field is huge and has generated new forms of performing typical activities in the new e-way: e-commerce, e-banking, and so on. Moreover, using distributed computing, referred to as the ways by which a single computer program runs with more than one computer at the same time, gives an advantage to human processes in tracking and management. The different elements and objects of a program are run or processed using different computer processors, and as such can be considered similar to parallel computing and grid computing (Garg et al., 2009). The main features of a distributed system include the following (Nadiminti et al., 2006):

- Functional separation - Based on the functionality/services provided, capability, and purpose of each entity in the system
- Inherent distribution - Entities like information, people, and systems are inherently distributed. For example, different information is created and maintained by different people. This information can be generated, stored, analyzed, and used by different systems or applications, which may or may not be aware of the existence of the other entities in the system.
- Reliability - Long-term data preservation and backup (replication) at different locations
- Scalability - Addition of more resources to increase performance or availability
- Economy - Sharing of resources by many entities to help reduce the cost of ownership

Nowadays, the availability of multi-CPU systems, either tightly coupled (e.g., multi core system) or loosely coupled (e.g., GRID system), opens doors for completely concurrent application aspects. Of all the previous advantages in using multi-CPU systems, networking and distributed systems have contributed significantly to human life. Yet, another technology is needed to accommodate the increased speed of management, reduce time, and facilitate speed of businesses growth. This requires technology to be able to support auto tracking and management. At first glance, using the GPS seems to be the best solution, given that it is a satellite-based navigation system and that it has a variety of applications useful in many aspects like on land, at sea, and on air (Figure 1-1).

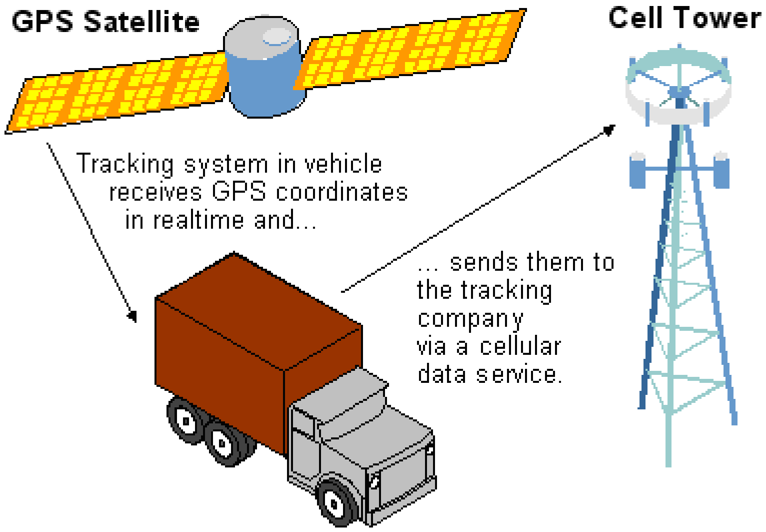

Figure 1-1 The GPS System (Encyclopedia2, 2007)

The GPS is a technology that has revolutionized many areas of our lives. It is made up of a network of 24 satellites placed into orbit by the U.S. Department of Defence (Figure 1-2) (Garg et al., 2009), but this system simply cannot meet all the needs of companies and factories due to the following reasons:

- GPS equipment, like a receiver, put on every shipped object, package, or envelope is considered too expensive.

- A GPS cannot be used for indoor tracking (e.g., inside buildings or in caves) and other subterranean locations because GPS-based systems are often utilized for large-scale outdoor environments.

Figure 1-2 The GPS Satellite System (Garmin, 2009)

For the reasons stated above, it may be difficult to determine if GPS can be used for general tracking, even if it is already useful for outdoor tracking. However, indoor tracking and automatic management for information can possibly develop given its relation to ensuring smoother businesses, apart from expanding its general purpose of tracking and monitoring both indoor and outdoor. One technology has achieved such: Radio Frequency Identification (RFID).

The RFID refers to small electronic devices consisting of tags, readers, and software. Tags are small transponders that respond to queries from an RFID reader by wirelessly transmitting a unique identifier (Jackson, 2004, BITKOM, 2005). These RFID tags are categorized as either passive or active (Bolotnyy and Robins, 2007). They are usually capable of carrying kilobytes of data, and each tag provides a unique identifier by which a reader can recognize between tags. An RFID reader is a device used to collect information from tags and send data to computers which have the software application. The software for RFID system can be divided into two parts: application programming interface (API) and application software. Defined loosely, RFID API works by fetching information, which is captured by the reader from the tags and then filtered and sent to the end-user, which consequently processes data using a software application. Actually the purpose of APIs is to support and increase the reusability of codes by using existing software artefacts or knowledge to create new software. Also API is a key method to significantly improve software quality and productivity.

In comparison, RFID is similar to the bar code technology or magnetic strips on credit cards, but RFID edges out its counterpart because RFID is considered smarter, more flexible, and automatic in term of collecting information, and can hold more data than bar codes. Moreover, RFID is capable of higher speeds of operation because of its use of radio waves to capture data from tags, and it does not require line-of-sight reading. Furthermore, RFID has the ability to recognize a pallet of mixed products by recognizing special numbers (i.e., tag's ID), thereby identifying all products containing individual RFID tags. Then, through a reader device, this technology can identify tags in a specific zone within the palletized load without having to move any material. Table 1.1 summarizes the differences between RFID and bar codes (Cheng-Ju et al., 2004, IEEE-USA's Committee on Communications and Information Policy, 2005).

Feature	RFID	Bar Code
Standards	Universal Product Code	Electronic Product Code
Capacity	More Capacity	Less Capacity
Reading mode type	Wireless	Optical
Reading direction	Not in line of sight	Line of sight
Ability to read many objects at the same time	Supports many readings	Reads sequentially
Range	From short to long range	Very short range
Types	Passive and active	Passive

Table 1-1 RFID versus Bar Codes

The RFID technology used from many years ago dates back to the World War II (Rieback et al., 2006), and it was used as a means to identify enemy or friendly airplanes. It can, in fact, be considered a popular and one of the most important research fields in recent years (Wei et al., 2006). Generally, RFID is used to indicate system solutions for tracking and monitoring objects both globally and locally. From mobile phones to safety-critical applications, RFID systems can be adopted in many aspects of our lives in order to improve the quality.

In these days the access to information easily, without delay and any complication has been considered one of the important aims in many fields. RFID is one of the best solutions for this aim because RFID is supporting real-time access, fast and smart detection for targets (tags), and has the ability for long-range area coverage reading.

In fact, the real-time availability of RFID information is deemed critical for many RFID applications such as in process control systems, manufacturing automation systems, critical applications, and coupling of active transponders for GPS and supply chains (Wang et al., 2005, IEEE-USA, 2009). In brief, RFID can improve operations and solve several very complex and time sensitive problems. There are many applications for the RFID system, and the more dominant applications include the following: supply chain management, inventory tracking, access control, library book checkout, cattle tracking, and passport tagging (Scassa et al., 2005). Some hospitals even use RFID to provide real-time tracking to locate doctors and nurses within the hospital and to track a patient's location and sets of equipment (Supply Insight, 2006).

The development in RFID has become beyond measure. Imagine people from the near future driving cars, opening doors, entering houses, and using various devices without the use of keys or passwords. All that have become possible with the use of special RFID devices which is small Tag (Figure 1-3), for example, implants inside a living body like an animal or a human being (Technovelgy, 2008).

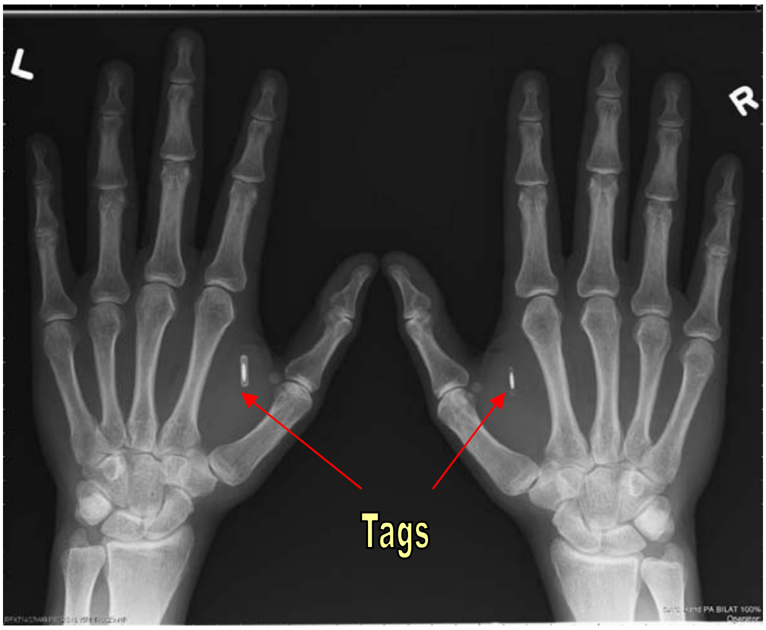

Figure 1-3 RFID Tag Inside a Living Body (Kranenburg, 2006)

At present, the RFID system and related networks have positive impacts in reducing overall project costs (Kovavisaruch et al., 2008). The ability to use these technologies has not only become cheaper, it has also become widely available and used by various segments of the society. The low cost cuts across equipment sets, tags, and sophisticated application support. Standardized protocols, real-time access, and ability to read multiple tags using long-range frequencies have further resulted to the diffusion of the RFID system. Indeed, networks can be considered an important and cheap technology, especially in the transfer of information using networks with RFID. Networks and distributed computers are provided the opportunity to expand RFID applications.

Despite ease of use, there still remain obstacles for users and developers, which include diversities in applications and hardware. Non-standard hardware and software implementation leads to none achievement of the positive impacts of RFID for many users. With RFID's popularization and wide usage, development should then focus on software applications. It is worth noting that developers can build several software applications using a single RFID product. However, the real problem stems from how to widely support libraries or identify flexible Application Programmable Interface (API) that can supports heterogeneous RFID Readers and different connection interfaces on heterogeneous operating system. Additionally, support is also needed to accommodate a diversity of users and their different applications. As such, by using API for RFID purpose, the developer can develop numerous applications software depending on the need of one engine. This specific requirement has been highlighted amid the high prevalence of RFID use and increased demand in many modern applications. Figure 1-4 shows the number of case studies worldwide in 2008 regarding this technology (IDTechEX, 2009).

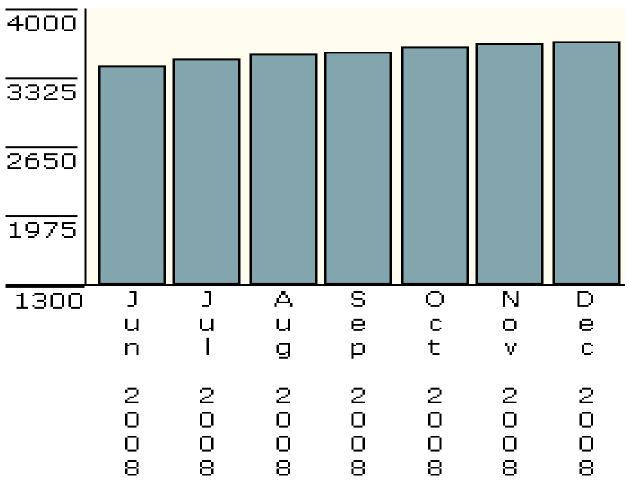

Figure 1-4 Number of Case Studies (IDTechEX, 2009)

1.2 Motivation

Both developers and users opt for products that can adapt in different environments. This raises the demand for independency and cross-functionality of products. To support this, we designed API that can dynamically and seamlessly support different operating systems, different RFID systems, and different connection interfaces. It aims to remove the gap currently filled by third-party programs and bridge the gap between industry and academic researchers by providing a shortcut development cycle.

Such APIs enable both technicians and researchers in both hardware and software domains to take vital advantages of API. In other words, the time to shift the gear towards real dynamic applications without reinventing the wheel is within the developers' hands.

To demonstrate the usage of these API, we implemented and evaluated a network-based RFID system for monitoring human attendance, RFID Distributed Database Management system, RFIDTM Protection System as an anti-thief system, Library Tracking and Monitoring, Clinic Tracking and Monitoring, Contact less RFIDTM Chat system, and RFID Terminal. This API has been used as a starting point in building all of our application programs (as proof of concepts).

1.3 Problem Statement

Despite its known potential, the development of the RFID system can be a painstakingly difficult process. Non-standard hardware and software implementations by many different vendors, interoperability and scalability problem, as well as lack of human expertise are amongst the reasons hindering the widespread use of the RFID system. In short, if an RFID system is to be made pervasively available, there is a need for a flexible API design that promotes reusability and redundancy, thus avoiding time wastage, as well as supporting interoperability within different operating systems and hardware implementations.

1.4 Thesis Aims and Objectives

The main aim of this research is to design, produce, and evaluate new Application Programmable Interface (API), called RFIDTM (RFID Tracking & Monitoring), which can solve some of the perceived deficiencies in the current RFID systems. The objectives of the work undertaken are:

- To develop RFIDTM with an open architecture that can be extended horizontally (by adding functionality) and vertically (supporting different vendors or providers).
- To ensure that the RFIDTM supports heterogeneous RFID system consisting of heterogeneous RFID readers and operating system.
- To evaluate the applicability of the RFIDTM for RFID system development.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 presents a literature review of API in general and RFID API or RFID middleware that supports the RFID system including issues related to the latter. It presents surveys and analyses of some of RFID APIs that support RFID systems mainly: Sun Java System RFID Software, JCAPS RFID Developer's Kit, Microsoft Biztalk RFID, Sybase RFID Anywhere, WinRFID middleware, Savant middleware, IBM WebSphere RFID, FlexRFID middleware, and SmartRF middleware. Towards the end of Chapter 2, a comparison between each technique is made based on their features and justification for the development of Radio Frequency Identification for Tracking and Monitoring (RFIDTM) API. In particular, the analysis highlights the processes that combine various features that can support the RFID fields and its developer.

Chapter 3 discusses and justifies the development of RFIDTM. Specifically, the chapter elaborates on the RFIDTM design, which consists of seven classes: RS-232 Engine, UTP Server Engine, UTP Client Engine, Filter Raw Information, MYSQL Data Base, Server Information, and Saving Class. It also explains the main core of RFIDTM, which strongly supports heterogeneous RFID readers and distributed computer systems. Additionally, its features and its support capabilities are also explained.

As a proof of concept, Chapter 4 demonstrates the various applications supported by RFIDTM, including their features and capabilities. Each application is demonstrated differently through case studies that utilize different types of RFID readers. Such differentiation estimates the high flexibility and reusability for RFIDTM.

The conclusion of this work is given in Chapter 5, wherein the comparison of RFIDTM with other RFID APIs is revisited. Also, the novelty of RFIDTM is summarized. Conclusions are drawn from the experiences gained from this work and the significance of the findings, along with considerations for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

The previous chapter has highlighted the potentials of RFID and challenges of adopting RFID systems. Building and complementing the previous chapter, this chapter revisits the main components of RFID in order to highlight the current achievements and difficulties in which the research contribution can be sought. In doing so, this chapter also presents in-depth literature review and analysis on the state-of-the-art on RFID APIs.

In a nut shell, RFID can be seen as breakthrough technology that provides the means and capability to collect and process information quickly, automatically and correctly. This technology identifies the target (tags) with radio signal automatically and obtains the unique identification and other information of a special entity (Chow et al., 2006). The technology helps to avoid users from wasting time by automatic scanning, identification and data capture for objects and entering that data directly into computer systems as opposed to manual operation (Wikipedia, 2009). For this reason, the use of RFID technology for monitoring and track processes are gaining popularity in recent years (e.g. the wireless induction system (Jian et al., 2008)).

In general, RFID based system has three main components: reader, tag, and software (Figure 2-1). Non-standard hardware and software implementations by many different vendors are amongst the reasons hindering the widespread use of the RFID system. As such, we can assume the increasing difficulty in the implementation of the RFID system for users and a potential increase in cost resulting from change in the types of readers (i.e., as opposed to support for more functionality). These have led the changes in pre-built application programs to accommodate support for a specific reader.

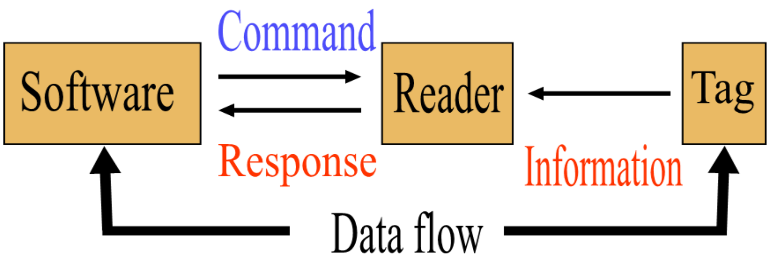

Figure 2-1 Typical RFID System with Application Software, Reader, and Tag

In the early days of computing, a primary concern involves building the hardware. Software was almost expected to take care of itself. The consensus held that hardware was difficult to change unlike software. Most people in the industry would carefully plan for hardware development but afford considerably less forethought to software, which most think as easy enough to change while doing work. In reality, the process of developing software is expensive, while maintaining and improving them is even a more expensive feat. This is most true in the case RFID, which practices non-standardizations. Moreover, designing good-quality API is difficult (SoftwareEngineering-BestPractices, 2007), and lengthy, especially in terms of learning how to build them.

As such, a popular research topic involves finding ways to solve, or at least reduce, difficulties in acquiring information from RFID systems without the need to change and reprogram applications, rebuild system, and address issues of time wastage and increased cost. This thesis suggests and implements API that considers the abovementioned setbacks. As such, this solution considers the first step in finding the cure for non-standardization in the RFID system. Expressed differently, the author believes that it is time to shift the gear towards real dynamic applications without reinventing the wheel.

Viewing for RFID system perspective, the API can be viewed in the following manner (see Figure 2-2).

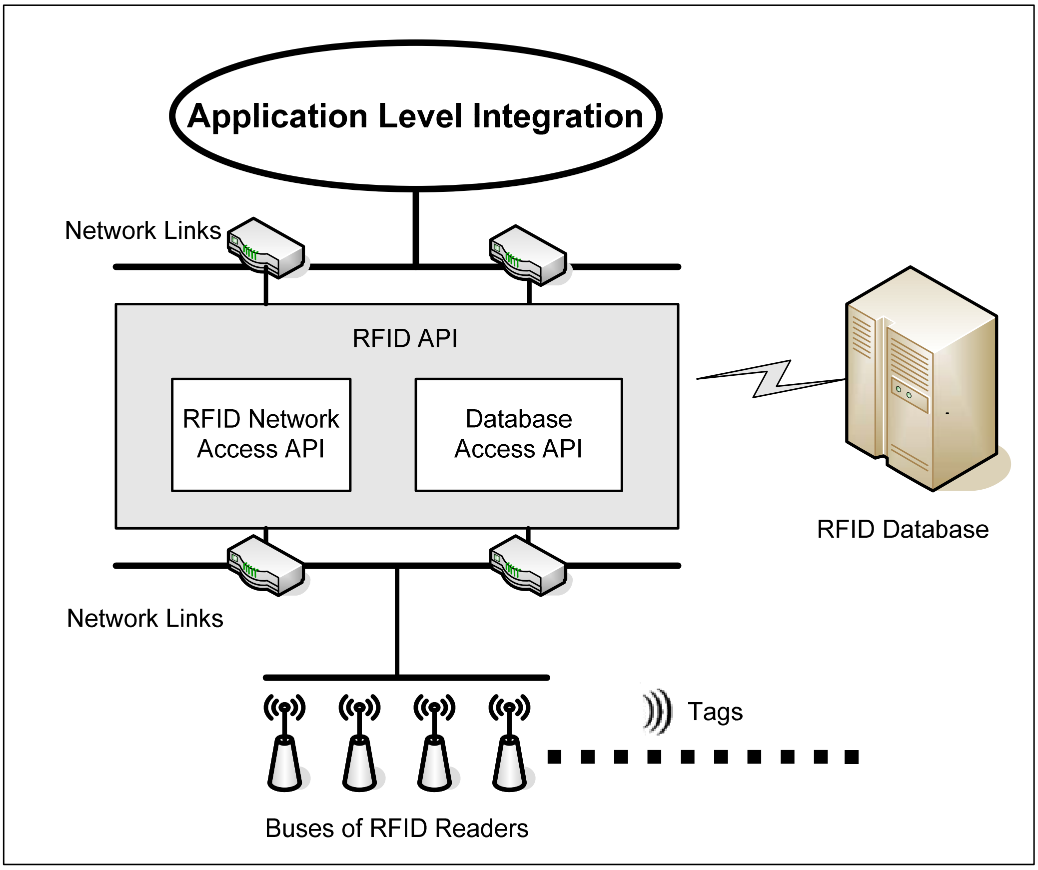

Figure 2-2 RFID API

Herein, RFID API serves as the interface from the application level to achieve system level connectivity to (possibly) buses of readers as well RFID databases.

2.2 Application Programming Interface (API)

In most cases, if a programmer wants to create software, everything needs to start from scratch. However, with the progress in technology and software engineering allowing the reuse of components, UML, as a standardized general-purpose modelling language in the field of software engineering, API, and libraries have been developed. As a result, the process of creating software has changed considerably. Instead of creating functionality, much of today's software engineering is about integrating existing functionality or repacking the functionality using APIs (Zibran, 2008). A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together. In fact, a major purpose of APIs is to support and increase the reusability of codes by using existing software artefacts or knowledge to create new software. This is a key method to significantly improve software quality and productivity.

Reusability can be defined as the degree to which an item can be reused. An API serves as a foundation for creating applications and saves programmers the time necessary to code basic functionality from scratch. It is also worth noting that using Java to design APIs has been considered extremely useful in making APIs more powerful.

Using Java gives strong architecture in designing libraries and APIs. As a programming language, Java can be incorporated into a network and a heterogeneous world. Java is designed for a distributed environment on the Internet and can handle TCP/IP protocols (Herbert, 2002). Another important feature is its ability to create cross-platform programs and the ability to work on different operating systems like Microsoft Windows, Linux, Solaris, and Mac. As such, Java is a powerful platform that includes a complete set of APIs for distributed applications, and it allows programs to run on a heterogeneous system in the network or on the Internet, which means that it can run atop existing platforms (i.e. Write the program one time and Run Anywhere as shown in Figure 2-3), strongly implying cost reduction in desktop administration.

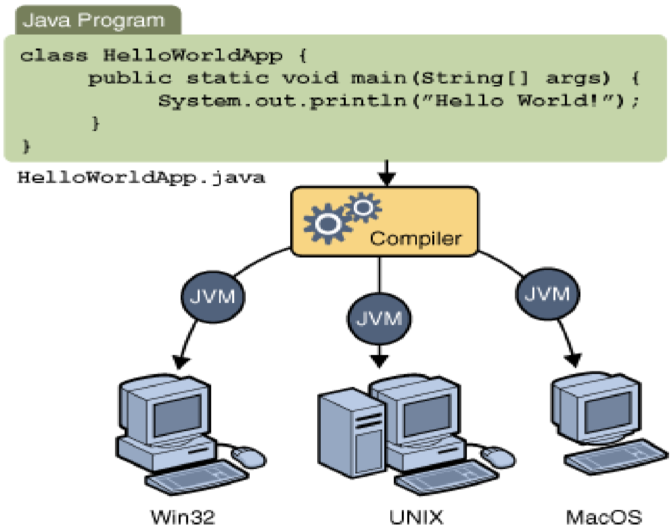

Figure 2-3 Supporting Heterogeneous O.S. System in the Network (Sun Microsystems, 2009)

In brief, with the current progress in technology in the RFID field and with the different venders' requirements, there is a need to use APIs with RFID systems that promotes interoperability.

2.3 RFID Application Programming Interface

There have been several attempts to support the RFID system with general APIs or middleware, but there is a lack of APIs that generally support divergence in the RFID system. In fact, some APIs are designed to support specific RFID readers. The growth of RFID in the application space has generated quite an interest in the RFID software development. Furthermore, there are many RFID middleware support software in the commercial field (Sun Microsystems, 2004b), but the most associated APIs have been developed merely to support the developers and researchers (Ghayal et al., 2008). In this section, some middleware and RFID APIs will be discussed.

Middleware refers to software or devices that connect RFID readers and the tags' collected information to end user or enterprise information systems. Middleware is a layer that helps makes sense of RFID tag reads and does fetching for information, filter tag data and some times monitor and manage devices. Defined loosely, RFID middleware applies filtering, formatting or logic to information captured by a reader so the information can be processed by a software application. Actually the middleware is an intermediary between reader and application software. The middleware also has some characteristics like independence which is mean that each of the RFID reader and the application system is independent of the RFID middleware (Ajana et al., 2009, Yulian et al., 2008).

Within the scope of this research work, there is a need to clarify the differences between API and middleware. Middleware can be seen as a level layer higher than API. In fact, it is the API itself that is used to develop middleware. Indeed, it is the API helps RFID middleware and application developers speed up development cycles. Nonetheless, as will be seen later, the differences between middleware and APIs can sometimes be a subtle one especially when the middleware also supports RFID development for heterogeneous readers.

There are some middleware available in the commercial and research domain which try to provide an environment for the application development there have been some proposals and research work involving middleware design and RFID data processing.

Some of middleware design to support one reader and other to support different readers. However, in this thesis, some of RFID middlewares are also illustrated as they behave like APIs. From these middleware we can discover the advantage and the limitation and discuss the features of each one. These RFID APIs and middleware are: Sun Java System RFID Software, JCAPS RFID Developer's Kit, Microsoft Biztalk RFID, Sybase RFID Anywhere, WinRFID middleware, Savant middleware, IBM WebSphere RFID, FlexRFID middleware, and SmartRF middleware.

2.3.1 Sun Java System RFID Software

Java-RFID is a Java programming library for RFID. Sun provides the Sun Java System its RFID software. This library will communicate with any RFID kit that has been tested with Texas Instruments and Microchip RFID kits (Java-RFID). Sun Java System RFID Software consists of four major modules (Sun Microsystems, 2006c), RFID event manager, RFID configuration manager wherein the graphical user interface (GUI) application is a component of the RFID event manager, an RFID management console that monitors and manages the status of the devices connected to the RFID event manager, and an RFID information server. In comparison, the two major components in the Java system RFID software are (Sun Microsystems, 2005) the Java system RFID event manager and the Java system RFID information server.

- Java System RFID Event Manager.
- Java System RFID Information Server.

The first component (event manager) is a Jini-based event management system that facilitates the capture, filtering, and eventual storage of events generated by RFID readers. On the other hand, the event manager is designed to process the streams of tag or sensor data coming from the reader devices. An RFID event manager consists of a control station and one or more execution agents as depicted in Figure 2-4. Filtering, aggregation, and counting of tag data are performed by the event manager.

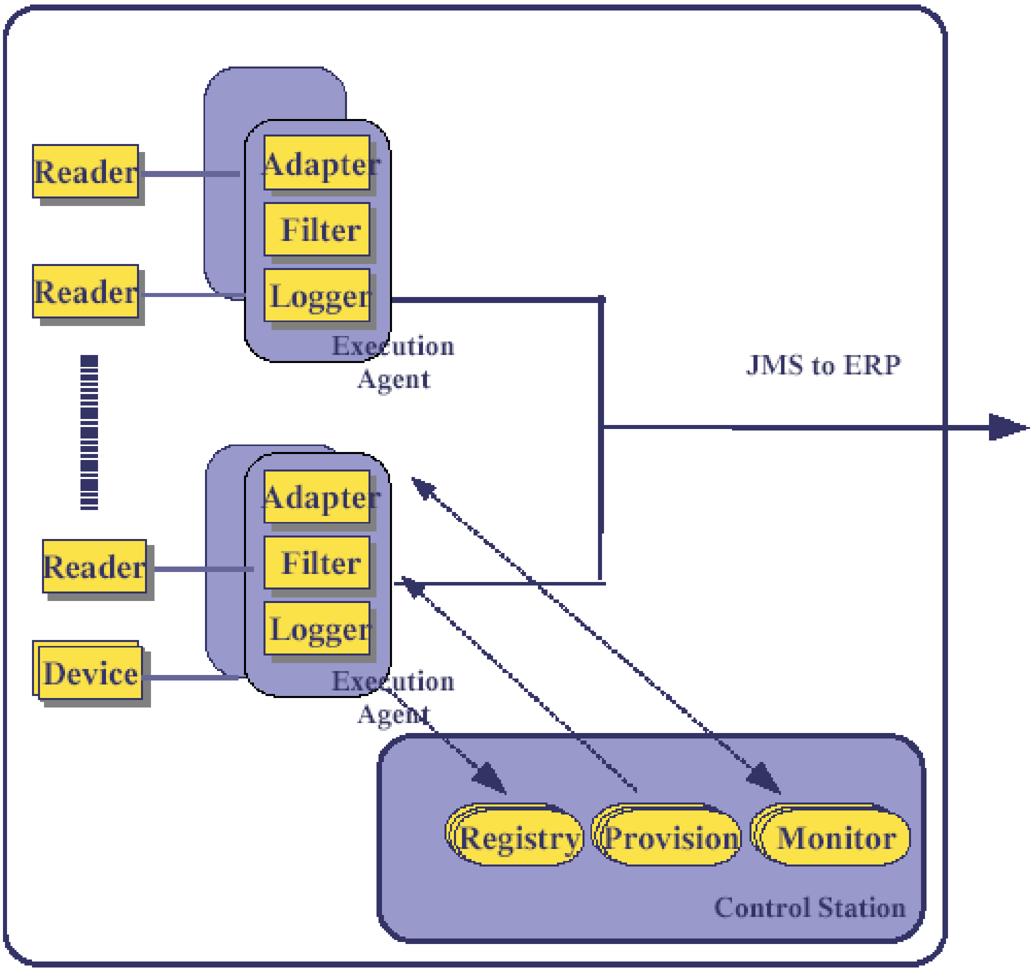

Figure 2-4 Sun Java System RFID Event Manager Architecture (Sun Microsystems, 2004a)

According to Figure 2-4, when a tagged object is scanned by a reader, all the information will be collected from the tags by the reader device and then sent to the event manager. Information will be processed by the Sun Java System RFID Software suite's RFID event manager. Succeeding this, the processed information can be stored in the RFID Information Server for future analysis. The information can also be sent continuously to third-party applications as it arrives at the RFID event manager (Sun Microsystems, 2006c, Sun Microsystems, 2005).

The RFID Information Server is a J2EE application that serves as an interface for the capture and query of Electronic Product Code (EPC)-related data. This EPC data includes a tag observation data from the RFID event managers, as well as information that maps EPCs to higher-level business data. The RFID information server translates a set of low-level observations into higher-level business functions and is designed to run on the Sun Java System Application Server and any J2EE-compliant application server.

Figure 2-5 shows how the Sun Java System RFID event manager and information server fit into the EPCglobal Network, for example, to enable a global business network (Sun Microsystems, 2004b, Sun Microsystems, 2005).

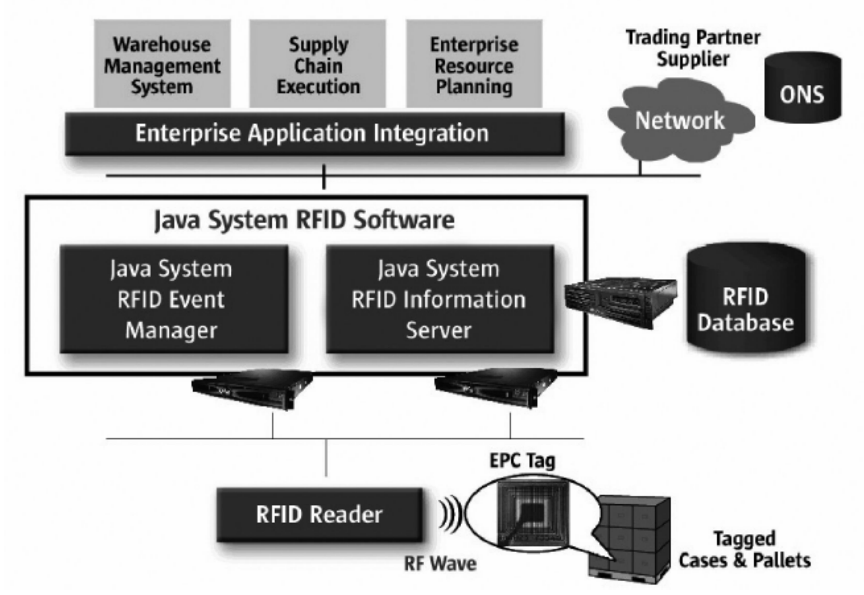

Figure 2-5 An EPCglobal Network(Sun Microsystems, 2005)

The Sun RFID software, as earlier mentioned, consists of an RFID event manager and information server. The software supports the commercial field. It has a shortened development cycle but maintains a highly reusable and reconfigurable API. By designing software depending on the Java language, the software can be supported in cross-platform functionalities. The Sun RFID Software also supports reader multi-connectivity and across different types of readers. Both the RFID event manager and information server play key roles in the EPC network as defined by EPCglobal, an international organization chartered with the standardization of RFID for use in business. As such, we can say that event managers can help developers send information to their application software.

2.3.2 JCAPS RFID Developer's Kit

The Java Composite Application Platform Suite (Java CAPS) RFID Developer's Kit extends Sun's Java Composite Application Platform Suite into the Sun RFID software. The JCAPS RFID makes RFID data available for business processes within and across enterprises. Meanwhile, JCAPS greatly simplifies implementing RFID in sophisticated enterprise application integration environments. The JCAPS RFID delivers powerful data transformation and integration capabilities for RFID solutions (Sun Microsystems, 2006a). The JCAPS RFID Development Kit is used to design eGate Projects aimed to process, validate, send, and receive RFID messages. Developers can thus access RFID messages (see Figure 2-6).

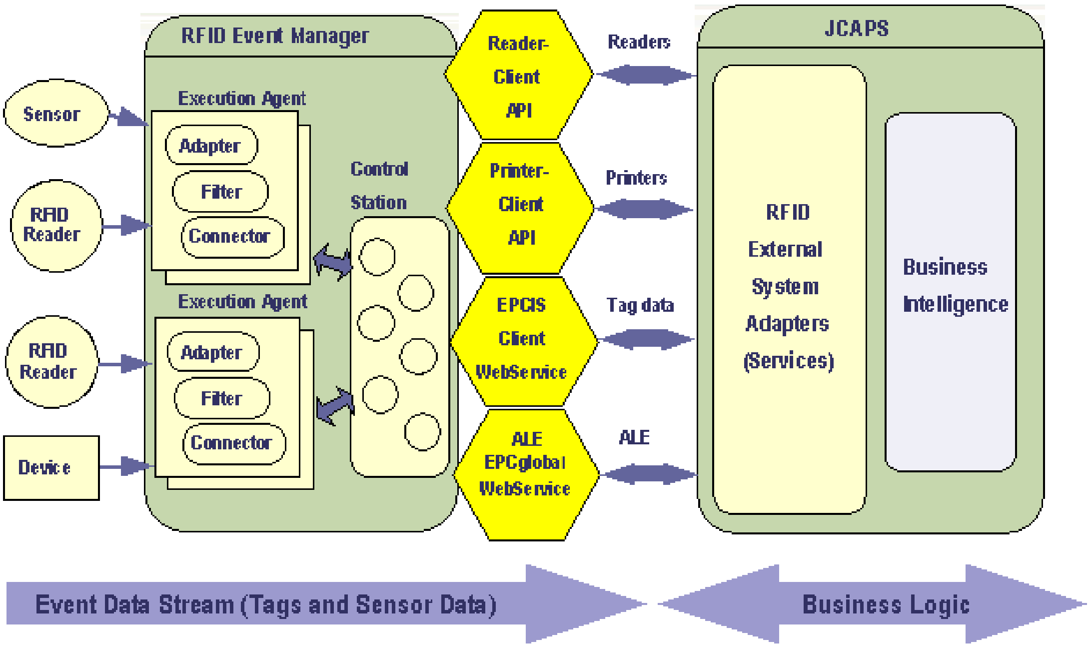

Figure 2-6 JCAPS RFID Developer's Kit (Sun Microsystems, 2006b)

Figure 2-6 clearly depicts the main parts by which APIs are dependent upon. The RFID Development Kit develops features contained in the Sun RFID Software, particularly the RFID event manager reader client API and the EPC information server interface. As such, JCAPS RFID has the same feature as the Sun Java System RFID Software, only it is made exclusively for electronic gate (eGate) and eInsight Projects and components.

2.3.3 BizTalk RFID

Microsoft also has provided a .NET platform-based RFID end-to-end utilization and designed, developed, and distributed application solutions known as the BizTalk RFID Server. By offering plug-and-play capabilities, flexibility, scalability, and extensibility, BizTalk RFID brings a myriad of advantages to application developers, systems integrators, hardware vendors, and end-users. BizTalk RFID is integral and is meant for the development, deployment, and management of diverse RFID and sensor implementations. BizTalk RFID is designed to allow all users to incorporate