

## **An RFID-based Universal Lightweight Multi-Domain Auto-Monitoring System**

**Aniruddha Bhattacharjya**

**Department of Computer Science and Engineering**

**University of Calcutta**

**92, A. P. C. Road**

**Kolkata 700 009, India**

**and**

**Rajat Kumar Pal**

**Department of Information Technology**

**School of Technology**

**Assam University, Silchar**

**PIN 788 011, India**

### **Abstract**

Radio Frequency Identification (RFID) is one of the most promising technologies [8, 13-16] in recent years. People's attentions are attracted by RFID technology's abilities to track moving objects in real time [13, 14, 16]. By tracking items in real time, users are enabled to have improved visibility into moving objects' status in their companies or organizations and even in their supply-chain partners, providing opportunities to lower inventory carrying costs, as well as reducing the need for storage warehouses, thereby improving cash flow, boosting productivity, and reducing overhead [8, 13, 14]. To reap the full benefits of RFID technology through seamless and effortless system integration, enterprises need to use RFID middleware [5, 12] to bridge RFID hardware and their application systems. The system must rapidly implement functions to process a large quantity of event data generated by RFID operations [13, 16] and should be configured dynamically [5, 11] for changing businesses. Consequently, developers are forced to implement systems to derive meaningful high-level events from simple RFID events [8, 13-16] and bind them to various business processes. Although applications could directly consume and act on RFID events, extracting the business rules from the business logic leads to better decoupling of the system, which consequentially increases maintainability. An RFID tag [14, 23] can be stuck on or incorporated into a product, animal, or person, and identify tag information by using radio frequency. So, in this era RFID system will be able to replace barcode system for many merits.

A typical RFID system [12, 13, 22, 23] is usually composed of four parts: tag, RFID reader, RFID middleware, and application system. To make the RFID system's reader integration uniform and effective, and to meet the low calculation requirement of low-cost readers, an RFID-based Universal [3, 5, 7, 11, 12] Lightweight [1, 2, 6, 9, 10] Multi-Domain Auto-Monitoring [1, 2, 3, 5, 11, 12] system is presented here. Bearing in mind that the Electronic Product Code (EPC) standard [4, 20-22] has become a virtual standard of RFID system due to its rapid development, to meet the demand of EPC Reader Protocol Standard (RPS) [2, 6, 7, 9, 10], the presented OSGi-based [5, 17-19] Universal Lightweight Multi-Domain Data Tracking and Processing system consist of the reader protocol which is specified in four layers, respectively the Discovery layer, the Reader layer, the Tag layer, and the Notify layer [1, 2]. Varying from EPCRPS [2, 6, 7, 9, 10], in which special Messaging/Transport Bindings provide for different kinds of transport, the Messaging layer and the Transport layer are no longer intended to be operational in pairs. This reader protocol in the system makes the reader integration uniform and effective and tracked data may be governed from a remote location.

This system may be used for applications [13, 14, 16] like, time and attendance, animal tagging, animal husbandry, postal tracking, airline baggage management, paper money anti-counterfeiting, anti-counterfeiting in the drug industry, vehicle immobilizers and alarms, road toll collection, electronic article surveillance (EAS), access control, manufacturing processes with robotics, monitoring of offenders, passports.

**Keywords:** Radio Frequency Identification (RFID), Reader, Reader Protocol, Electronic Product Code (EPC), Service Management Framework (SMF), User Datagram Protocol (UDP), Open Services Gateway initiative (OSGi), Java Message Service (JMS), EXtensible Markup Language (XML).

## 1. Introduction

Radio Frequency Identification (RFID) is a technology that allows readers to detect a tagged item without line of sight or contact by using radiofrequency waves [4]. RFID is widely accepted by industries [8] to be an emerging technology for product identification [1]. RFID technology has already myriad commercial applications [8]. In manufacturing environment it can be used to track inventory, reusable containers, work in process, and finished products. In transportation and logistics it can be used for fleet management, pallet, container, cargo track, and improving asset utilization. In health care and pharmaceutical it can be used to reduce medical errors. In defence and aerospace it can be used to reduce counterfeiting of parts and improve safety by tracking hazardous materials.

Regardless the field of applications, with RFID deployment scales-up, tag traffic is channelled to the applications requiring data through network-centric infrastructure. So, there is a huge demand for an universal Lightweight [1, 2, 6, 9, 10] Multi-Domain Auto-Monitoring [1-3, 5, 11, 12] system, which does not depend upon only network-centric infrastructure, but can work in various domains[13, 14, 16] of applications like time and attendance, animal tagging, animal husbandry, postal tracking, airline baggage management, paper money anti-counterfeiting, anti-counterfeiting in the drug industry, vehicle immobilizers and alarms, road toll collection, electronic article surveillance (EAS), access control, manufacturing processes with robotics, monitoring of offenders, passports, and so and so forth. So, here, an RFID-based Universal [3, 5, 7, 11, 12] Lightweight [1, 2, 6, 9, 10] Multi-Domain Auto-Monitoring [1-3, 5, 11, 12] system is presented.

The rest of this paper is well thought-out as follows. Section 2 introduces the related literature survey. Section 3 describes the architecture of the whole system. Section 4 describes the results of experiments. Finally, Section 5 concludes the paper.

## 2. Literature Survey

Several vendors and organizations have projected allied system architectures or protocols to universalize and standardize the communication protocol [1, 2, 6, 7, 9, 10] between readers and RFID middleware in their Auto-Monitoring system [1-3, 5, 11, 12, 15]. The most distinctive one is the Simple Lightweight RFID Reader Protocol (SLRRP) [9]. SLRRP is enacted by Reva Company and its cooperation partners. The fundamentality of the protocol is TCP over IP. Self-defined functions are transported over SLRRP channel [9]. The main advantage of adapting SLRRP is that different readers and middleware systems can be integrated easily with efficiency improving and error decreasing. However, SLRRP comes from manufacturer proprietary protocols, which entails following drawbacks.

- ❖ The reader logical function is defined in hard code, which makes the system almost impossible to be expanded.
- ❖ The parameters of commands are fixed, and dynamic parameters are not supported, which decrease the flexibility of the whole system.
- ❖ The physical interface of readers only supports TCP/IP, and the bottom protocol lacks of reliability restriction.

These leads to make a Universal [3, 5, 7, 11, 12] Lightweight [1, 2, 6, 9, 10] Multi-Domain Auto-Monitoring [1-3, 5, 11, 12] system, which does not depend upon only network-centric infrastructure [9], but can work in serial or network interfaces with multi-domain applications.

One of the main standards for making Auto-Monitoring [23] system is the EPC Reader Protocol Standard (EPCRPC) [20-22], which specifies the interaction between a device capable of reading (and possibly writing) tags, and

application software [7]. These two parties are herein referred to as the Reader and the Host. EPCRPS [20-22] is specified in three distinct layers, respectively, the Reader Layer, the Messaging Layer, and the Transport Layer. Taking into consideration that the EPC standard has become a virtual standard of RFID system due to its fast development, EPCRPS is adopted as the basis in the paper to describe the working process and function of RFID reader communication protocol. Varying from EPCRPS in which special Messaging/Transport Bindings provide for different kinds of transport, the Messaging layer and the Transport layer are no longer intended to be set in pairs. Our reader protocol [1-3] in the RFID-based Universal [3, 5, 7, 11, 12] Lightweight [1, 2, 6, 9, 10] Multi-Domain Auto-Monitoring [1-3, 5, 11, 12] system makes the reader integration uniform and effective, and tracked data may be governed from a remote location.

### 3. System Architecture

In this section the detailed architecture of our RFID-based Universal Lightweight Multi-Domain Auto-Monitoring system is presented. In our architecture we used the Alien RFID Readers (Model-8800, Model-9800) and Class 1 Gen 2 RFID Passive Tags. The Dynamic Host Configuration Protocol (DHCP) mode of configuration eliminates the need for the user to perform network configuration for the devices, when changes takes place. The architecture is implemented here in JAVA, the run-time platform is Service Management Framework (SMF), which is Open Services Gateway initiative (OSGi) compliant. Therefore, the run-time environment is supported by embedded OS, like WINDOWS CE, Embedded LINUX as well as 32-/64-bit OS environment (SUSE LINUX, WINDOWS XP/2000). Each layer consists of one or more OSGi bundles delivering specific set of services. OSGi framework [17, 18] is more dynamic as well as a framework of scalable components. OSGi [19] gives a definition of a standard, service-oriented computing environment, and provides with an open, service oriented, easy-to-deploy lightweight components model. Based on OSGi framework, this work puts forward the Universal Lightweight Multi-Domain Auto-Monitoring system with service component architecture. The reader communication protocol uses publish/subscribe Java Message Service (JMS) infrastructure. Therefore, the architecture ensures a lightweight dynamic modularity for many applications. To meet the demand of EPC RPS, our reader communication protocol is specified in four layers, respectively, the Discovery Layer, the Reader Layer, the Tag Layer, and the Notify Layer.

The *Discovery Layer* used in order to use and control a reader, its network address or the serial port number on the host where it is connected first is known by it. Dynamic Host Configuration Protocol (DHCP) mode of configuration eliminates the need for the user to perform network configuration for the device. This layer provides the classes, `SerialDiscoveryListenerService` and `NetworkDiscoveryListenerService`, needed to automatically search for and discover readers using both of these connection modes. In both cases, the service is created, an object is registered as the recipient of discovery events, and the service is started. Once started, the service runs on its own thread until it stops automatically (for serial discovery) or is told to stop (for network discovery). Here `NetworkDiscoveryListenerService` class is developed along with `SerialDiscoveryListenerService` class. Each alien reader is configured, by default, to broadcast heartbeat messages over its local subnet. These messages are UDP (User Datagram Protocol) packets containing small XML documents, which detail the reader's type, name, and contact information. The class that performs these listening duties is called `NetworkDiscoveryListenerService`. Once this class is instantiated and started, it runs in its own thread until it is stopped. While running, it listens for reader heartbeats on the listener port (which is specified in the constructor), calling either the `readerAdded()` or `readerRenewed()` methods of a registered `DiscoveryListener` when it detects a reader. Part of the heartbeat sent out by the reader indicates the time until the next heartbeat is expected. If this time expires before the next heartbeat is received, then the service assumes the reader goes to offline and the `readerRemoved()` method is called.

The *Reader Layer* consists of primary classes for communication between readers and the host (RFID middleware or applications) either over the network or a serial port. Typically the reader object is obtained from a `DiscoveryItem` object, however, if the location (either serial port or network address) is known, a reader object can be instantiated directly without the need of any discovery classes.

The *Tag Layer* is one of the most important layers as tags play a very important part in the RFID reader and tag system. For this reason there is a single class devoted to storing and manipulating tag information: the `Tag` class. Additional classes and an interface are helpful for managing raw tag data and tag lists within the applications. The `Tag` class has the following members, each of which is accessible through getters and setters in the Application Programming Interface(API): *Tag ID* – a string representing the tag's ID, *Discover Time* – the time the tag is first

seen by the reader, *Last Seen Time* – the time the tag is last seen by the reader, *Count* – the number of times the reader has read the tag since it is first seen, *Antenna* – the (transmit) antenna number where the tag is last seen, *Protocol* – the air protocol used to acquire the tag's ID.

The *Notify Layer* consists of classes work in conjunction with a reader running in autonomous mode. In autonomous mode the reader is configured to read tags over and over again without the need for human interaction. The reader can be configured to send messages to listening services on the network when specific events occur, such as a timer expiring, tags added/removed from the taglist, successful/unsuccessful programming, etc. The notify classes implement such listening services, constantly waiting and listening for notification messages from readers, and converting these messages into Java objects, which are then available to the developed application. The key class in the notify layer is *MessageListenerService*. This is a service that listens at a specified port for incoming reader notification messages. A *Message* object encapsulates a collection of metadata about the notification message itself, and an array of *Tag* objects is extracted from the taglist portion of the notification message. It contains the following members, all of which are available through getter and setter accessor methods: *ReaderName* – the name of the reader, *ReaderType* – the type of the reader, *IPAddress* – the IP address of the reader, *MACAddress* – the MAC address of the reader, if provided, *CommandPort* – the port number on which to send commands of the reader, *Time* – the date and time the message is sent out, *StartTriggerLines* – indicates which external inputs triggered the reader to start, *StopTriggerLines* – indicates which external inputs triggered the reader to stop, *TagList* - an array of *Tag* objects extracted from the notification.

The architecture of the RFID-based Universal Lightweight Multi-Domain Auto-Monitoring system is depicted in Figure 1.

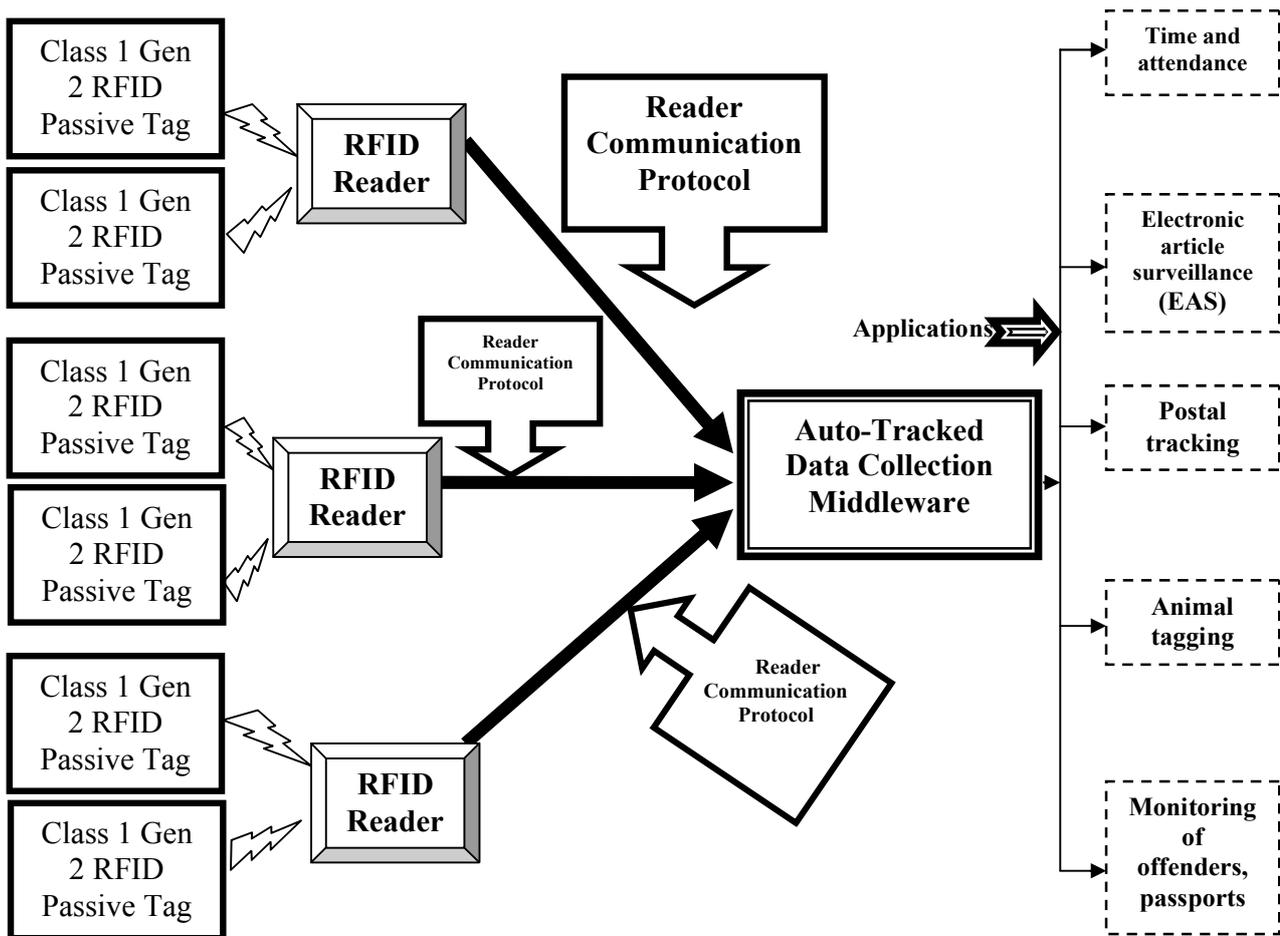


Figure 1: Architecture of the RFID-based Universal Lightweight Multi-Domain Auto-Monitoring system.



attendance, animal tagging, animal husbandry, postal tracking, airline baggage management, paper money anti-counterfeiting, anti-counterfeiting in the drug industry, vehicle immobilizers and alarms, road toll collection, electronic article surveillance (EAS), access control, manufacturing processes with robotics, monitoring of offenders, passports, and so and so forth. The experiment results show that the system can meet the requirement of lightweight, integrity, fundamentality, transparency, hardware independence, simplicity, safety, and reliability.

## References

1. Bhattacharjya, A. and Pal, R.K.: An RFID-based Ubiquitous Framework for Mobile Object Tracking with Reconfigurable Lightweight RFID Reader Protocol. Proceedings of International Conference on Distributed Computing and Networking, PhD Forum Publication, pp. 33-36 ( 2010)
2. Bhattacharjya, A. and Pal, R.K.: Layered OSGi-based Reconfigurable Lightweight RFID Reader Protocol. CD Proceedings of International Conference on Industrial Engineering and Operations Management (IEOM) ISBN: 978-984-33-0988-4, Id: 217 (2010)
3. Bhattacharjya, A. and Pal, R.K.: An RFID-based Self-Organizing Architecture for Remote Sensing. Accepted for publication in the Proceedings of International Conference on Informatics, Cybernetics, and Computer Applications (ICICCA2010), Id: 206 (2010)
4. Floerkemeier, C., Anarkat, D., and Osinski, T., Harrison, M.: PML Core Specification 1.0, AUTO-ID Center work paper (2003)
5. Fagui, L., Liu, H., LinKai, and Yongxiong. R.: OSGi-based Reconfigurable RFID Middleware. In 2nd International Conference on Anti-counterfeiting, Security and Identification (ASID 2008) Print ISBN: 978-1-4244-2584-6, pp. 410 - 415
6. Huang, D. and Kapoor, H.: Towards Lightweight Secure Communication Protocols for Passive RFIDs. Proceedings of the IEEE Secon (2009)
7. Tan, C.C., Sheng, B., and Li, Q.: Serverless Search and Authentication Protocols for RFID. Proceedings of Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom) (2007)
8. Hossain, M.M., and Prybutok, V.R.: Consumer Acceptance of RFID Technology: An Exploratory Study. IEEE Transactions on Engineering Management, Vol. 55, No. 2 (2008)
9. Krishna, P.: Simple Lightweight RFID Reader Protocol, SLRRP Working Group (2005)
10. Ouyang, Y., Hao, J., Zhang, T., Ren, Q., and Xiong, Z.: Research on Lightweight RFID Reader Protocol. In First IEEE International Conference on Ubi-Media Computing (2008)
11. Wu, J., Wang, D., and Huanye Sheng, H.: Design an OSGi Extension Service for Mobile RFID Applications. Proceedings of IEEE International Conference on e-Business Engineering (2007)
12. Wu, J., Wang, D., and Sheng, H., A Component-based Reconfigurable REID Middleware. Proceedings of 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Department of Computer Science and Engineering, Shanghai Jiaotong University (2007)
13. HighJump Software, A 3M Company: The True Cost of Radio Frequency Identification [Online]. Available: <http://highjumpsoftware.com/promos/rfid-cost-report.asp> (2004)
14. The History of RFID, Association for Automatic Identification and Data Capture Technologies, [www.aimglobal.org](http://www.aimglobal.org) (Oct. 2001)
15. RFID Journal Frequently Asked Questions, [www.rfidjournal.com](http://www.rfidjournal.com)
16. Data from: Radio Frequency Identification A Basic Primer, Association for Automatic Identification and Data Capture Technologies, [www.aimglobal.org](http://www.aimglobal.org) and from Wikipedia Encyclopedia, <http://en.wikipedia.org/wiki/RFID> (Aug. 2001)
17. OSGi Alliance, <http://www.osgi.org>
18. OSGi Technical Whitepaper, <http://www.osgi.org/documents/collateral/OSGITechnicalWhitePaper.pdf>
19. OSGi Alliance. OSGi service platform release 3 [EB/OL]. [http://www.osgi.org/osgi\\_technology/-spec\\_download3.asp](http://www.osgi.org/osgi_technology/-spec_download3.asp)
20. EPCglobal, EPCglobal Architecture Framework, Version 1.2 (Oct. 2007)
21. EPCglobal, EPCglobal Certificate Profile, Ratified Specification 1.0 (Mar. 2006)
22. EPCglobal, Reader Protocol Standard, Version 1.1 Ratified Standard (Jun. 2006)Auto-ID Savant Specification 1.0., Oat Systems & MITAuto-ID Center, <http://www.epc.org.mx/contenido/WD-savant-10-20030911.pdf>