A Web Based Peer-to-Peer RFID Architecture

Harinda Fernando[1(✉)] and Hairulnizam Mahdin[2]

[1] Asia Pacific Institute of Information Technology, Colombo, Sri Lanka
harinda@apiit.lk
[2] Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Malaysia
hairuln@uthm.edu.my

Abstract. To realize the maximum benefits of RFID technology in large scale distributed environments, the use of an architectural framework which fulfils the specific requirements of those systems is paramount. Unfortunately, the existing frameworks are designed at a high level to allow the development and deployment of a number of fundamentally different systems. Therefore, specialist systems based on this kind of framework will run into a number of issues due to the nature of those applications and their unique needs. In this paper, we present web based P2P architecture for distributed RFID systems specifically targeted at distributed RFID systems. We carry out a comparative analysis of the proposed which shows that our architecture has a number of significant advantages over other existing systems.

Keywords: RFID architecture · Distributed systems · Secure middleware

1 Introduction

In current ICT world there a major emphasis on Internet of Things (IOT) and in the future IOT technologies such as RFID will be used in fundamentally different systems. Due to the differences of these systems their specific functional and performance requirements will also change. The current globally accepted RFID architecture is a generic framework knows as the EPC Global Architecture Framework (EPCGAF) [1]. Therefore much of the specific requirements of specialist RFID systems cannot be met using it, creating a strong need for the creation of Specialist RFID architecture frameworks that can be used to build specialist systems such as global supply chain systems. The developed frameworks must also ensure compatibility with existing systems using the EPCGAF but must be able to provide additional functional and performance benefits as required by specific specialist systems.

By developing a Peer-to-Peer web based RFID architecture we fulfill the requirements of RFID enabled Global supply chain systems. Our approach increase both scalability and availability of the overall system while reducing the processing overhead required when using data provided by external partners. We have also leveraged the preexisting relationships between partners to simplify the issues of node churn and new partner identification for the Peer-to-peer system.

T. Herawan et al. (eds.), *Recent Advances on Soft Computing and Data Mining*,
Advances in Intelligent Systems and Computing 549, DOI 10.1007/978-3-319-51281-5_55

The remainder of this paper is structured as follows. A brief review of previous works is discussed in Sect. 2. In Sect. 3, we describe the proposed architecture. The results of are analyzed and presented in Sect. 4. Finally, some concluding remarks are given in Sect. 5.

2 Related Work

One of the first peer-to-peer RFID architectures is proposed in [2], and uses a hybrid method for peer resolution. Information discovery is done using the traditional EPC while service discovery is done using a DHT based system. The proposed architecture removes some of the bottlenecks and scalability and availability issues associated with the EPCGAF. But because the peers are networked by chain linking the address entries, if any participants are not available the chain would break and information access would be compromised.

In [1] the authors present a peer-to-peer, DHT based alternative to the EPCGAF. This architecture allows for the use of any type of tag identifier to allow greater interoperability with other architectures. The hash values of the tag identifier, which are also used to identify information sources about that tag, are mapped to a distinct location in the network where the participants can retrieve the entry directly. The actual data lookup is carried out using either a direct or indirect search. In direct search the object identifiers are used as keys and looked up in the DHT key space. For indirect searches indices have to be created and updated periodically. But all information associated with the tags remains in the participant's local system and other partners retrieve the required data from that one location. The proposed architecture has a number of improvements over the EPCGAF including greater scalability, because the data look-up is done in a distributed manner and interoperability. The main issue with this architecture is the partner data sources creating a single point of failure as well as scalability issues.

In [3] the authors present another peer-to-peer based RFID resolution framework which is based on the original proposal presented in [2] but without chain links. This system uses the EPC company prefix number (CPN), to map the keys to the nodes which contain data about it. The nodes in the system are then arranged in a logical circle based on their node ID. Because the first part of the node ID is based on the country this ensures that the nodes in the logical network are arranged with physically closer peers arranged logically closer to each other as well. The data resolution is done by going along the circle till a peer with the required information is found. While its scalability and resistance to failure is higher than the EPCGAF it still has scalability and availability issues at the actual data services. In addition, it also has issues with duplicate data look, retrieval and formatting as retrieved data is discarded once it's used and must be re-retrieved when needed again.

3 Proposed Architecture

The role of the proposed RFID architecture is to organize and manage RFID infrastructure throughout the enterprise in order to capture tag events, generate RFID data in real time, store it with minimal loss and share that data with partners using Peer-to-Peer web services (Fig. 1). The middleware in our architecture is developed to be modular and is required to carry out the tasks listed below:

- Filter and collect the data received from multiple readers.
- Carryout security tasks to ensure the integrity and confidentiality of data.
- Translate the tag identifier and data retrieved to information.
- Generate transaction data based on business events.
- Retrieve, aggregate, filter and format RFID tag data.
- Act as the communication hub for different components.

Data Cleaning and Filtering Module - The Data Cleaning and Filtering Module (DCFM) of the middleware is in charge of accepting RFID tags reads from multiple different readers, cleaning it by removing false reads and filtering duplicate reads and collating the reads from multiple readers [4]. Please note that there are a number of different data filtering and collection mechanisms proposed in recent literature that the developer could implement. For further details on the different challenges in RFID

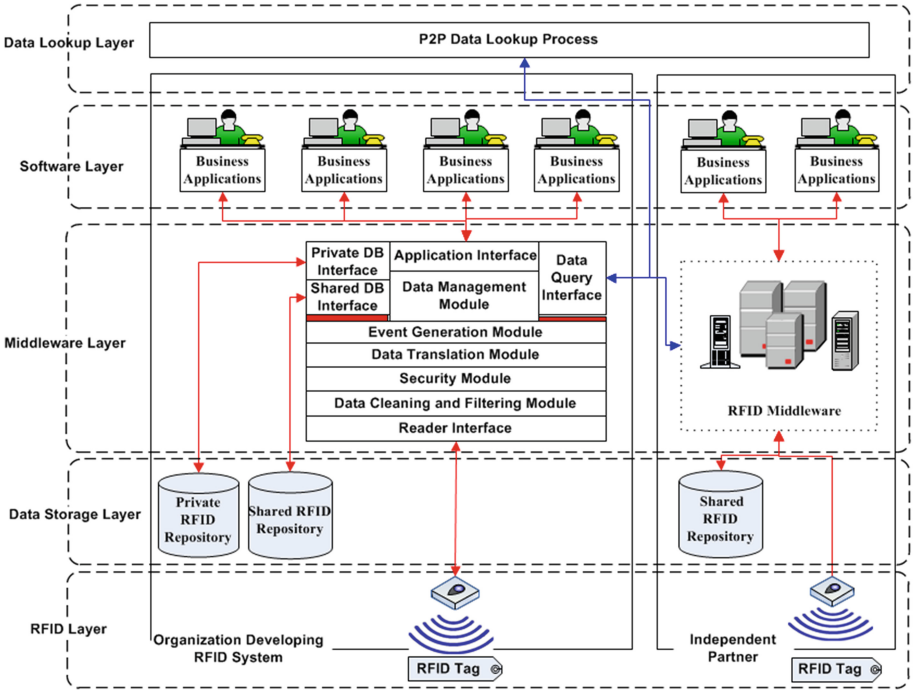

Fig. 1. P2P networked RFID architecture

data filtering and management and a comparison of the different possible approaches refer to [5].

Security Module - Overall, the minimum security functionality that the security module needs to provide includes mutual authentication of tags and readers, transmission confidentiality and integrity and tag anonymity. In addition, security requirements such as storage confidentiality and integrity, non-repudiation, tag malware protection and access control need to be implemented as well [6]. There are a number of security solutions and protocols, that can be used by this module, that offer varying levels of protection and features as discussed in [7].

Data Translation Module - The data translation module in our middleware needs to carry out two types of data translations. It needs to translate raw binary data from the tag into a format usable by the system and it also needs to translate the information received from the databases and business applications into raw binary data to be stored on the tag. For the first task the module splits tag data into different fields and for each field retrieves the data translation rules and applies them. To translate database data to tag storable data the module identifies which field each data should be stored to, retrieves the data translation rules and applies them to the received data. Once the translation is complete, the update is forwarded to readers.

Event Generation Module - The most important functions of an RFID system is the automated generation of transaction data concerning the tag's it identifies. Transaction data is created by associating EPCs with specific business events and transactions [8]. The event recognition module is responsible generating information based on tag reads and business rules and events and tasked with controlling and coordinating certain actions in the physical and digital environment [9].

Imagine the middleware receives the tag reads for new tags at a specific warehouse and it also receives information from the business applications that a logistics company X is delivering the goods for invoice for that warehouse from seller Y. By combining this data the Event Generation Module (EGM) generates the invoice received event for that transaction and associates all the newly picked up tags with that invoice. It may also initiate the opening of outbound logistics for sales for that specific good from that warehouse (Table 1).

Data Management Module - In networked RFID systems, different partners at different data storage locations store data about the tags used in the system. Additionally in our approach, the RFID data is shared using a P2P model rather than a client server model. Therefore, proper management and identification of this distributed data as its being saved and retrieved is required if the system is to work efficiently. In the proposed architecture, this task is the responsibility of the Data Management Module (DMM). Therefore the DMM is tasked with the responsibility of locating, retrieving, aggregating and formatting data from multiple different sources in such a manner as to most effectively respond to any single data request [10]. It is also responsible for deciding how the data generated by the EGM and data retrieved from external partners should be saved between the private repository and shared repository.

Table 1. P2P networked RFID architecture

Tag event detected	Actual business process	Business application data	Data stored in shared RFID repository	Business process triggered
Tags T_1 picked up by readers at warehouse	Receiving of new stock from logistics supplier S_1 delivered by truck TR_2	Goods for invoice I_1 received at warehouse W_1	Tag T_1 is at warehouse W_1 Tag T1 belongs to delivery invoice I_1	Start shipping out goods for orders which can be fulfilled Divert any more shipments if warehouse space is full
Tag T_1 leaves warehouse	Logistics supplier S_2 picks up stock for delivery using truck TR_1	Goods for sale SL_1 shipped from warehouse W_1	Tag T_1 is no longer at warehouse W_1 Tag T_1 belongs to sale invoice I_2	Request for more stock if extra space is available

3.1 P2P Technology in RFID Systems

The two big issues for P2P networks are node churn and security or privacy concerns [3]. In normal public P2P networks, partners/nodes are constantly joining and leaving the network. But in RFID systems, the network partners and therefore the P2P nodes are very stable. When one such node enters the system it is there permanently, except for occasional down time, till that partner leaves the supply chain. Therefore the issue of node churn does not apply to in the environment we are working in [3]. The other main drawback of P2P systems is security concerns because anyone can join the network. Therefore, access control, privacy and trust concerns come into play. However, partners in distributed RFID systems are business entities with existing business partnerships and connections. Therefore most of the security and privacy concerns plaguing public P2P networks do not apply to supply chain RFID P2P networks [2].

RFID data is currently categorized into two groups: Static data (data created at the birth of the object and which does not change) and Transaction data (data that is generated by different partners over the course of its lifetime).

However, sharing data that is constantly changing over a P2P network creates data synchronization problems while sharing only static data would defeat the purpose of using P2P technology, as it's only a very small percentage of the total data concerning any given RFID tag. Therefore, to remove data synchronization requirements and ensure that the highest amount of data can be shared via P2P we further split the transactional data into constant and updatable transaction data (as shown in Table 2).

In the proposed environment, the transaction data needs to be associated with the original partner who generated that data when it's stored at a different location. Therefore, in our system in addition to the tag identifier we also use the unique identifier of the partner who generated that data to identify transaction data. In addition,

Table 2. Types of data

	Generated at	Generated by	Updated	Example
Static data	Birth of object	Manufacturer	No	Batch number of item is 3476 Item expires on 14/08/2012
Constant transaction data	Over lifetime	Supply chain partners	No	Item was checked into warehouse ×23 on 21/10/2010 Item was sold to supply partner Y as part of invoice 211
Updatable transaction data	Over life time	Supply chain partners	Yes	The next destination for item is warehouse ×56 There are 1863 lots of model Z at warehouse 34

Table 3. Comparison on stored transaction data details

	Other system	Proposed system	Details
Tag identifier	Yes	Yes	Unique to the tag: must be given by a global authority
Original partner identifier	No	Yes	Unique to each partner for each supply chain
Date generated on	No	Yes	The date on which the information was generated or last updated
Data class	No	Yes	Number indicating the data class: 1- Static data, 2- Constant transaction data, 3- Updatable transaction data
Transaction information	Yes	Yes	The actual information that was generated concerning the tagged object

to allow for stronger and more granular identification and filtering were also store the date on which that information was actually generated (Table 3).

When a partner requests specific data all of the above information it will also be transmitted for each transaction event. When a business application requires data concerning an object it will relay that request to the data management module which will identify which data services of which partners might contain the information required and retrieve that data.

When requesting P2P data from external data services additional information provided by the application can be used to retrieve only a subset of the total data at the external service. The system can ask the P2P data service for data concerning tag X, which was generated by partner Y between two specific dates. Once data is retrieved from external partners it will be stored on the local servers and shared with other partners via the P2P network. The potions of the retrieved data that has been classed as static data or constant transactional data (indicated by either 1 or 2 for data class) will be saved in raw form in the shared data repository while any data classed as updatable transaction data will forwarded to the business applications and not stored and shared

via the P2P network. All locally generated shared transaction data, regardless of data class, will be stored in the shared data repository and shared via the P2P network.

In our system the data services will be web based and will have a service profile, which contains Meta data about the service it offers. It also will generate its data profile, which contains information about the data it's sharing, and share these two profiles with other partners. The two extra data fields that are filled by the partners and allow the partner to track when he last downloaded the data profile for a particular external data service and when he should retrieve a newer data profile for that service. Each web data service will also create a list of all the tags the service has data about along with the original partner who generated that data and the last time data for that tag and partner combination was generated or updated.

The proposed service and data profile file sharing approach is based on the fact that partner chains pre-exist in distributed RFID systems. Once the initial discovery is done, the partners use direct communication to retrieve data from partner web data services. The partner profile distribution and lookup process is divided into two main parts: (1) partner data service discovery (2) partner data profile update.

When a completely new partner joins RFID system it is with the knowledge and approval of at least one existing partner and that existing partner will be able to directly get the service profiles of the new partner. That existing partner will then be tasked with distributing the new partner's service profiles to their up or down stream partners. When the other partners receive the new partners service profiles they will directly contact those services and retrieve their data profiles. The existing partner who initiated the new partner will also be in charge of forwarding all services profiles it has to the new partner. The process is shown in Fig. 2.

When a new data service is added that partner includes the service profile of the new service in all his existing data services. As all service-profiles have an expiry date partners need to regularly contact all the data services and refresh their service profiles. When this happens, any service profiles of new data services will be sent along with the current service profile for that particular data service. When the requesting partner receives the service-profiles they directly contact the new data services and request them for their data profiles as shown in Fig. 3.

Whenever a data service gets new data it will update its data profile to reflect the new data it has available. External partners will use the data profile time stamp and the data profile expiry fields in the service profile for each data service to update their data

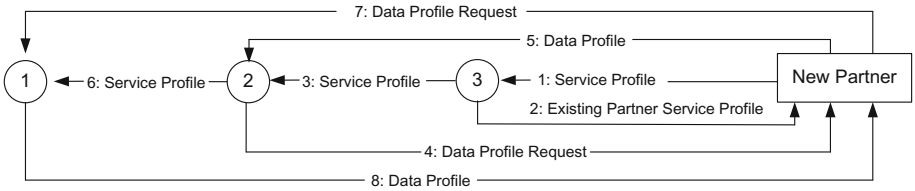

Fig. 2. New partner service profile sharing

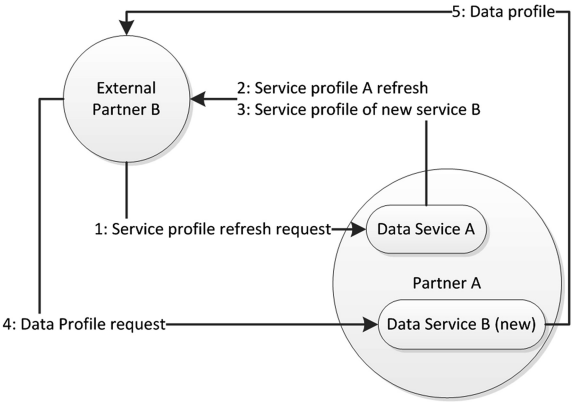

Fig. 3. New data service profile sharing

from a partner. Using these up-to-date data profiles partners can request the information they require from any number of data services rather than just the data service that originally generated it.

4 Comparative Analysis

In the EPCGAF, the lookup services offered by the EPCGlobal itself and the EPCIS are all centralized and based on client-server technology and scale badly. In the more recently proposed P2P based architectures [2, 3] the bottleneck that is created by the EPCGlobal lookup services is removed. However, the bottlenecks at the actual RFID data sources still exist because they are still a single server. In our architecture both the lookup process and the data sharing is done using modified P2P techniques. The comparison of architectures is presented in Table 4.

The structure of the EPCGAF introduces a number of Single Points of Failure (SPOF), at the lookup process and the EPCIS of partners, which affects the availability of the system [11]. In the more recent P2P architectures the SPOF at the data lookup is removed but the system still retains a SPOF at the data service components. In contrast, the architecture proposed by us removes both these SPOF and increase system availability and reliability in a number of ways. (1) By storing static data on the RFID tag in addition to the data service of the manufacturer, we reduce the dependency on external data sources and services, (2) By storing filtered, aggregated and formatted data in a local DB, we further minimize the dependence on all external services and (3) By using P2P technology, which is proven to have much better availability and reliability than client-server technology, we increase the overall reliability and availability of the networked system as a whole.

Unfortunately the security features that are provided by the EPCGAF are very basic [11]. In addition, the ONS service that it offers have a number of security issues such as vulnerabilities to DDoS attacks and cache poisoning [12]. The P2P architectures in [2, 3] remove the vulnerability to cache poisoning and partially removes the vulnerabilities

to DDoS attacks. However, they do not have any security features built into them; neither do they discuss the potential threats to the proposed system. Our architecture offers a number of advantages when it comes to system security and the P2P technology that our architecture employs eliminates the vulnerability to cache poisoning and reduces vulnerability to DDoS attacks.

Table 4. Comparison of architectures

		Proposed	EPCGAF	P2P architecture	Peer resolution framework
Scalability	Data lookup	High (P2P)	Low (Client-Server)	High (P2P)	High (P2P)
	Data sharing	High (Client-Server)	Low (Client-Server)	Low (Client-Server)	Low (Client-Server)
Availability	Data lookup	High (P2P)	Low (Client-Server)	High (P2P)	High (P2P)
	Data sharing	High (Client-Server)	Low (Client-Server)	Low (Client-Server)	Low (Client-Server)
Performance	Partner/server discovery	Uses chain distribution, done one time for each data service	Hierarchy based ONS, is repeated each time data is required	Uses DHT tables	Peers are arranged in a circle based on location
	Data lookup	First time networked than local, based on simple list shared by partners	Networked, based on hierarchy based ONS	Networked, based on DHT tables	Networked, based on profiles published by service
	Data sharing	P2P	Client-Server	Client-Server	Client-Server
	Data retrieval	Only done once for each data set	Lots of duplicate work	Not indicated	Not indicated

The EPCGAF performs certain tasks quite inefficiently and most of the time, the system needs to at least access the manufacturer's EPCIS as well as the EPCIS of the partner containing the transaction data required when retrieving data. In addition the data recovered from another partner's EPCIS needs to be filtered, aggregated and formatted before it can be used by business applications [10]. As the EPCGAF requires that the data be retrieved from the partner's EPCIS each time it is needed, the system must filter, aggregate and format the same data whenever it is retrieved from partner's EPCIS. This creates unnecessary duplication of work, which affects system performance negatively. Because the EPCGlobal implements the ONS as a hierarchy, the system also has to complete a large number of processes to complete a data lookup. The P2P architectures reduce the number of steps required for the data lookup as its direct lookup. Additionally our architecture improves performance in a number of ways. Due to static data being stored on the tag, our architecture has a fewer number of situations requiring a data look-up. In our architecture, once data is retrieved and formatted its stored in the local private database. By doing this, our architecture significantly reduces the amount of duplicate filtering, aggregation and formatting done by

the middleware component of the system compared to the EPCGAF. This significantly reduces the load on the middleware and therefore improves the overall system performance. We also use the chain distribution method for locating new data services and partners. This approach is a lot more efficient than typical decentralized P2P node discovery methods such flooding, because partners directly query each other. In addition the P2P data sharing balances the loads more efficiently and reduces bottlenecks therefore improving overall system performance.

5 Conclusion

In the future RFID will be used in large number of different systems. In this paper we design and present a peer-to-peer RFID architecture that can run as a web service for systems that are distributed and of very large scale. To do this an efficient manner we have used peer-to-peer technology as well as developed a novel way of classifying and identifying different types of RFID based data. We have also proposed a mechanism through which service and data profiles can be used to simplify the data discovery and retrieval process. The comparative analysis shows that it will provide greater scalability, availability and performance than currently existing RFID architectures for the type of system it's developed for.

References

1. Uckelmann, D.: Quantifying the Value of RFID and the EPCglobal Architecture Framework in Logistics. Springer Science & Business Media, Heidelberg (2012)
2. Wakayama, S., Doi, Y., Ozaki, S., Inoue, A.: Cost-effective product traceability system based on widely distributed databases. J. Commun. **2**(2), 45–52 (2007)
3. Shrestha, S., Kim, D.S., Lee, S., Park, J.S.: A peer-to-peer RFID resolution framework for supply chain network. In: IEEE Second International Conference on Future Networks, pp. 318–322 (2010)
4. Kamaludin, H., Mahdin, H., Abawajy, J.H.: Filtering redundant data from RFID data streams. J. Sens. (2016)
5. Mahdin, H.: A review on bloom filter based approaches for RFID data cleaning. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) DaEng-2013. LNEE, vol. 285, pp. 79–86. Springer, Singapore (2014). doi:10.1007/978-981-4585-18-7_9
6. Fernando, H.S., Abawajy, J.: A security framework for networked RFID. In: Abawajy, J.H., Pathan, M., Rahman, M., Pathan, A.-S.K., Deris, M.M. (eds.) Internet and Distributed Computing Advancements: Theoretical Frameworks and Practical Applications, p. 85. IGI Publishing, Hershey (2012)
7. Piramuthu, S.: RFID mutual authentication protocols. Decis. Support Syst. **50**(2), 387–393 (2011)
8. Jia, F., Jeon, S., Hong, B., Kwon, J., Kwak, Y.S.: Flexible capturing application for enhanced generation of EPCIS events. Int. J. Distrib. Sens. Netw. (2014)
9. Tan, J., Wang, H., Li, D., Wang, Q.: A RFID architecture built in production and manufacturing fields. In: Third International Conference on Convergence and Hybrid Information Technology, vol. 1, pp. 1118–1120 (2008)

10. Musa, A., Gunasekaran, A., Yusuf, Y.: Supply chain product visibility: methods, systems and impacts. Expert Syst. Appl. **41**(1), 176–194 (2014)
11. Armenio, F., Barthel, H., Burstein, L., Dietrich, P., Duker, J., Garrett, J., Suen, K.: The EPCglobal architecture framework. Rapport technique Version, 1 (2007)
12. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: 2 attacking RFID systems. In: Security in RFID and Sensor Networks, p. 29 (2016)