Internet of Things architecture for a RFID-based product tracking business model

Félix Jesús Villanueva, David Villa, Francisco Moya, Maria José Santofimia, Juan Carlos López
Department of Information Technologies and Systems
School of Computer Science
Ciudad Real, Spain
{*felix.villanueva, david.villa, francisco.moya, mariajose.santofimia, juancarlos.lopez*}*@uclm.es*

***Abstract*—Device and technology diversity along with application heterogeneity are two of the main factors that have an impact on the way how middlewares build IoT applications in a global and uniform fashion. The presented approach therefore proposes an Object-Oriented distributed middleware to set up an IoT ecosystem related to RFID-based product traceability business model. This paper also discusses the IoT requirements and how the proposed approach, apart from fulfilling them, can benefit all involved parties from developers and users to companies. As an example, goods tracking and recycle industry is used to show how IoT concepts can be evolved from laboratories to high scale industry.**

***Keywords*-Internet of Things architecture, Object-Oriented distributed middleware, product tracking , RFID.**

I. Introduction

Goods traceability is a growing application demand that, mainly individually, companies are implanting all over the world. In fact, this is the killer application in RFID technology, used by RFID manufacturers to prove how RFID tagged products can be monitorized, managed, stored, etc.

With the evolution of tag tehcnology is possible to tag almost every product keeping the cost down. However, these traceability applications traditionally:

- end when the product is sold to the user.
- are used by companies (manufacturers, supermarket, etc.) internally. Only big chains of supermarket can force its suppliers to tag its products.
- and, specially in product related to food chain, the information provided in the tag is only used for forensic information (who, where and when made this product).

This paper goes a step further including consumers and recycling companies in the traceability application. However, changing the way how entities (manufacturers, malls, consumers, etc.) share traceability information and the reasons why they do it. This approach implies a change of the existing business model since now all business entities have to take part in the chain.

This paper therefore proposes an evolution of goods trace applications, mainly motivated by the wish of obtaining a global ecosystem of goods traceability that enables high level services.

Regarding the technical aspect, this work proposes the use of Object-Oriented Distributed Middlewares (OODM) as global IoT infrastructure. OODMs have a long history of success in modeling, building and managing distributed systems with most of the features that a global IoT application need (high scalability, efficiency, modular design, etc.).

The information written in RFID tags is used for invoking remote methods in objects. As it will be thoroughly described later on this article, the proposed approach consists of routing, at application layer, the invoked method from the read tag to the service. This application level routing enables the use of different transport layers.

Summarily, two are the main contributions of this paper: in first place, this work proposes a comprehensive business chain for IoT infrastructure in which all the involved parties get benefits (obtaining either money or information); and in second place, this work also proposes an object oriented architecture that complies with all the needs of the considered business model.

The remainder of this article is organized as follows. Next section describes the proposed business model and it also establishes the requirements for the IoT architecture. Section III shows how previous approaches fail to handle these requirements. Section IV thoroughly describes the proposed middleware architecture along with message specification and data management. Finally, last section describes some of the most relevant aspects of the implemented prototype along with the main conclusions drawn for this work.

II. Good Recycle Business Model

Firstly, it has to be stated that the proposed architecture has been designed from a practical point of view and with an specific application framework in mind. The authors of this work strongly believe that the key issue for an IoT application to succeed is to report benefits to the IoT infrastructure investment requirements.

We are going to design our architecture from a practical point of view and with a specific application framework as guideline. In our opinion, a key need for a successfully IoT application is that all parts involved get benefits from required investment of the IoT infrastructure.

It has to be highlighted that the proposed application deals with the tracking of packaged goods. For example, perishable goods are tracked from packing and labeling (with RFID tags) to the collection/recycling of used packaging.

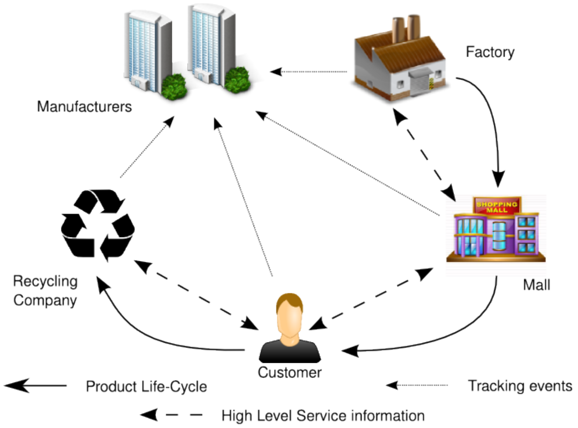


Figure 1. Proposed Business Model overview

Obviously, as it can be observed, this scenario brings to light the essential role played by scalability, heterogeneity and security issues in any architecture intended to support these type of applications.

Figure 1 depicts an overview of the IoT infrastructure proposed for supporting the business model. A manufacturer has several factories making and packaging its products. In the process of packaging, the factory attaches a RFID label to each product. When the factory dispatches a product to a shopping mall, it sends a tracking event with the product information to the manufacturer. In a similar way, the mall sends a tracking event to the manufacturer when it receives the product and another one when it sells the product to a user.

The user who acquires the product also sends the manufacturer a tracking event when he or she stores the product at home. The user generates another event (addressed to the manufacturer) when he or she consumes the product and it is thrown to the bin. Finally, the recycle company sends a tracking event when the user puts the product container in one of its recycling bins.

All entities send its tracking events to the manufacturer. The manufacturer has the role of an information broker giving information to all involved parties. Later on this article, technical details will be provided about how this platform is implanted. However, this section focuses on the high level services that can be provided by the proposed platform.

The manufacturer has a full real-time image about who, where and when to make, sell, store, consume and recycle its products. Any manufacturer can optimize its store management, product design, marketing strategies, factory productivity, client management, etc. with the managed information provided by the proposed infrastructure.

The mall can ask each manufacturer about who bought each product, grouped by shopping basket, and get information about user's profile and consume patterns. The mall can also offer segmented marketing campaigns from these consumption patterns, directly offer discounts to users, etc. Additionally, automatic stock information can be enabled.

The recycle company can pay-per-recycle to the users individually increasing its business. Once again, recycle companies and malls can optimize their process in the same way manufacturer do.

The key point in this model is the user. User needs to install RFID readers at home and provide profile information, as it will be explained later on this article. All the involved entities have to therefore make an effort to involve users in the model.

The user could obtain the following benefits:

- Get money benefits by recycling from recycle companies: In some malls users can find recycle machines and get discounts in the shopping basket by each plastic bottle, tin, cartons, etc. inserted in the machine. Our model could enable a big market paying to users monthly or annually by its recycling habits.
- Receive special product bargains from malls and manufacturers according to their consumption patterns, preferences, fidelity, etc.
- Recording (in a local store) the product movement in their homes, users could enable the access to these information to third-part application. External applications could investigate about their consumption patterns to analyze and to optimize features like quality, price, trademark alternatives, etc.

The main benefits and costs for each of the involved parties can be summarized in table I.

The proposed model exhibits certain relevant features that, under the authors point of view, should be part of any IoT business model:

- Incremental: The general overview of the architecture supports an incremental enrollment of entities as the business model succeeds in the market, although it might also be useful from the very beginning.
- Fault tolerant: Despite the fact that users are the weak link of this business model they also are at the same time the key entities to get this model to succeed. Even if few manufacturers and recycle companies adopt this model, the global business model could work.
- Scalability: A succeeding business model would involve and integrate thousands of companies playing different roles.
- Architecture reuse: the architectural technology required for the IoT business model proposed here could be reused for other activities.

These features are translated into requirements for the architecture design as described later on this paper.

The main drawback of this business chain is the externalization of sensible information for companies and consumers. However, this hurdle can be easily overcome thinking that:

Table I
MAIN ENTITIES AND THEIR BENEFITS IN OUR RECYCLE BUSINESS MODEL

Role	Benefits	Cost
Manufacturer	Consumer profiles	IoT Infrastructure
	Real-Time demand	Information broker
	Product Consumption patterns	
Malls	Consumer profiles	IoT Infrastructure
	Real-time demand	Share Company Information
	Trademark Consumption patterns	
Users	Recycle profits	IoT Infrastructure
	Consumption patterns	Share Personal information
Recycle Company	Increase demand	IoT Infrastructure
	Pay-per-Recycle Business model	Share Company Information

- All information is segmented by manufacturer trademark so, there is no single entity storing all information related to users or malls.
- A manufacturer already knows the product it sells to the malls, so only traceability is added to the handled information.
- As explained later on this paper, the user information provided at a first stage is only an anonymous ID. Afterwards, manufacturers and malls get a profile in which the information included is fully controlled by the user.

The following section analyzes the state of the art solutions that have been proposed to the date to tackle this type of business models.

III. PREVIOUS WORK

RFID is a fast-growing global technology with multiple application fields. There are applications for security, store management, automotive, contactless payments, etc. Product tracking is among the most promising application fields [1] enabling several applications like eliminating inventory inaccuracies, inventory shrinkage, fine grain product recalls, anti-counterfeiting, etc.

Fine-granularity tracking products has several problems which can be grouped as showed underneath:

- Technological issues: Dealing with heterogeneity in standards, hardware and stack software used in these applications seems to be one of the key problems.
- Architectural issues: Depending on specific implementations, scalability, time response, security and privacy, etc. are issues that have also to be tackled.
- Data-Mining issues: Again, depending on specific implementations, algorithms to deal with massive data have to be applied.

Of course, in the literature, several models for dealing with these problems can be found. EPCglobal [2] designs from scratch all software stacks to deal with massive RFID events. However EPCglobal provides a general architecture without having any business model in mind. EPCglobal defines a set of entities which have a query interface to know information about RFID events inside of each entity. An entity uses a central service discovery service for finding other entities, there is a central service for getting the object name of each product, etc. so as a conclusion, EPCglobal is thinking in pulling model meanwhile the solution proposed here focuses on how raw events are transmitted in a push model. So, despite the fact that some modules might be interesting from the point of view of the proposed approach, in the overall, this architecture does not fit with the proposed business model.

The SmartProducts [3] architecture proposes a Java distributed channel-based publish/subscribe system for interaction among objects (represented by a set of OSGi bundles). In our business model, more complex things like cold stores, washmachines, refrigerators, etc. will get access to a more elaborated information attending to their needs. Also from developers point of view, we will see how the proposed solution standardizes the protocol while keeping free the decision about what platform to use, or what language and tools are more suitable for each application.

The use of P2P-based solutions deal with robustness of the system, In [4] a hierarchical P2P based RFID code resolution network structure is proposed. These type of works are centered in deploying an infrastructure to lookup a specific tag or support all type of applications. In this sense, the proposed model is, by design, load balanced, distributed and even fault tolerant. As it will be explained later on this article, the aforementioned drawbacks of this type of applications are directly overcome by resorting to well-known solutions and some key design decisions.

IV. THE OBJECT-ORIENTED MIDDLEWARE FOR IOT

In a traditional object-oriented middleware, the design of any application starts with a specification of the interfaces of all services designed to be remotely accessed. After the implementation phase, the service runs in a server and it publishes its identity that have to be used to reach

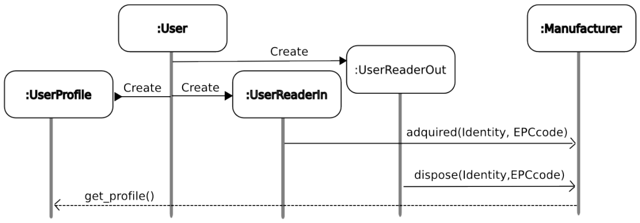


Figure 2. Sequence Diagram

the provided service from any point in the internetwork. The middleware provides method encapsulation and direct invocations.

In the proposed platform, the manufacturer includes its identity within the product code in the RFID label attached to the product. The service running in the RFID reader only has to interpret the printed RFID identity and invoke the method in the corresponding manufacturer service.

A. Architectural structure

Each entity has several input points in which RFID readers notify the reception of products and output points in which RFID readers notify which product leaves the entity. According to the type of the entity (user, mall, etc.), the complexity of input/output points are more/less relevant. A typical input point for a user is the refrigerator and an ouput point is the dustbin. In case of a mall, the complexity is more relevant, the input point is a set of RFID readers at store department which could make complex operation like aggregation, batching of sending events, etc. meanwhile the output points are the check out operators.

Each entity also has a profile resolver in which other parties can get information related to the entity. Again, the type of entity sets up the type of information that a third party can get. Maybe the most sensitive point is the user profile.

Attending to the entity type, the profile provided is related to the company type, name, placement, etc. In the case of the user, the information could be more elaborate and related to the type of family, number and age of members, incoming salary, etc.

A public key infrastructure protects the information communications between entities. At this point reliable security mechanisms provided by generic middlewares assure secure communications at different levels, achieving a total integrity and security of communications.

The internal infrastructure or even data management in each entity depends on the type and size of the entity. We will focus in common points, that's it, how the information is communicate between entities. Effectively, the key point in this architecture is how the information is addressed to manufacturers only with the information which we read in RFID tags. For meeting this we use what we call Inter-Domain Messaging (IDM) architecture.

B. IDM overlay architecture

Our previous work IDM (Inter-Domain Messaging) [5] is what we propose to building a generic invocation-oriented transport protocol. An invocation is a message to request a distributed object to execute a specific method with given arguments. It is exactly the same basic concept in which middleware technologies are based on, that is RMI (Remote Method Invocation) IDM provides a cross-layer technology-agnostic transport method for conventional method invocations.

Despite this fact, IDM works more like a network protocol although its datagrams may be encapsulated over any protocol, on any layer: TCP, UDP, IP, ethernet, etc. Depending on an application level specified QoS profile, IDM routers check the feature set of the available underlying protocols and choose the more appropriate to achieve the target object or next hop.

IDM imposes a global hierarchical addressing scheme for objects. These addresses are *object identities* in the object oriented middleware parlance, and they need to be globally unique. For our IoT proposal, IDM provides identifiers for all parties: all are represented in the system as distributed objects. The identifiers are independent on the network or protocol technology. Any client (particularly RFID readers) is able to invoke any other remote party in the system. The required information to perform that invocations is:

- The manufacturer object identifier coded in the product RFID tag.
- The product code, also in the RFID tag.
- The "owner", that may be hard-coded in each reader.

By means of IDM, each component may lay on a different network technology with any kind of protocol, network or link layer, open or proprietary, etc.

In the given scenario, customers may use wireless readers employing Zigbee or similar low rate radio technology. These readers can communicate to remote objects through an embedded IDM router connected to the customer ethernet switch. We can think about a simple customer kit composed by a reader for user recycle bin, a reader in the home door and the router/radio receptor in the ethernet switch.

The city recycling containers could be equipped with autonomous readers for the disposed products. There are two options here. Readers may store product codes in an internal memory and transfer them to the collector truck. If a more synchronous behavior is required, the reader may have a data connection through GPRS or similar.

Factories and malls require readers in the product store entries and the POS (Points of Sale in malls). These readers can be integrated in the their information systems. The case is similar for recycling companies.

It is worth mentioning that that the network topology is highly heterogeneous. Readers used in different point may require different precision, scope or communication interface. IDM provides an effective solution to deal with this heterogeneity.

C. Identifiers

All parties are globally accessible through their object identifiers. The specification of ProductID is represented by the following slice language in ICE[1]:

```
struct ProductID {
  int productKind;
  int serialNumber;
};
struct Tag {
  int manufacturerID;
  ProductID product;
};
```

Product RFID tags essentially contains tree fields

- Manufacturer service identifier. It is an IDM address that may be directly invoked from everywhere.
- Product kind identifier. A unique product number (per manufacturer) to identify each product (model).
- Serial number. A unique serial number (per manufacturer and product) to identify each product individually (product instance).

The RFID readers get *manufacturer identifier* from the product tag and invoke the manufacturer service given the rest of tag content (the productID) as parameter. The productID has Electronic Product Code (EPC) format which is an universal identifier that provides an unique identity for every physical item.

Using EPC with IDM requires a small modification.IDM addresses must be hierarchical because the routing is subnet based. EPC manager field should be managed by an administrative that would assign address to product trackers (manufacturers or agent acting on behalf them).

D. Product tracking protocol proposal

The platform works as a typical distributed application. Its design is based on a set of object interface specifications. This section describes that interfaces and their influence in a actual implementation. The platform provides two basic (and independent) functions:

- An event service, to get product tracking. Here, manufacturers are the sink for all events sent by all other parties.
- A query service, that enables parties to get information from other parties.

This work proposes a specific set of interfaces for both services. Each party implements one or more interfaces depending on their roles. To improve scalability, each company may have a set of global identifiers, in each geographic or organizational zone.

The protocol messages for the event service (summarized in table IV-D) represent input/output events send to the manufacturer in each entity. The name of the operation associates with the state in the process-chain of the product. Note that not all messages are required to obtain a satisfactory operation. Some messages may be lost and may be interpolated with a good error margin. Event temporal precision is not required for most services.

Event	Reader location	Source ID	Destination ID
dispatched	Plant	Plant	Manufacturer
received	Store	Mall	Manufacturer
sold	Sell Terminal	Mall	Manufacturer
acquired	Fridge	User	Manufacturer
consumed	Bin	User	Manufacturer
dropped	Recycle container	Recycler	Manufacturer
recycled	Recycle plant	Recycler	Manufacturer

Table II
PROTOCOL MESSAGE SUMMARY

The query service is closely related to the possible ways of platform exploitation. Some example interface to implement these services must be:

- Stock control: Given a product class identifier, mall and manufacturers may inform about its current stock or sales forecast. This information is useful adjust production and provisioning to current market.
- Product profiles: Recycling companies, users and malls will be interested on product composition, storage conditions, expiry dates, etc. That information may be used to automatize storage process at mall, product classification in recycling processes, etc.
- Product lifecycle: Manufacturers will be interested on the geographical distribution of their products, times between manufacturing and disposing, expiry fulfillment, etc.
- User profiles: Customer shopping profiles are very valuable for all parties. Malls may infer consumption patterns to improve its offer, manufacturers may produce real-time market and demand analysis, etc.

E. Platform efficiency and data volume

Taking the information of the RFID tag (96 bits) plus the ownerID like mallID, userID, etc. (28 bits) and including the protocol header used by ICE (208 bits), a single event has a full length of 336 bits. Some approximate numbers could be provided to estimate the impact of deploying the proposed business model to the current market. Mercadona, one of the main supermarkets chains in Spain, according to its annual statistics [6] has 1310 malls supplied by more than 2000 providers with around 5300 different products. According to Mercadona's report, it sells around 8.500 millions of kilo/litres of products. Assuming 4 different products per kilo, a single Mall would send, in average, around 2,18 GBytes per year in information events to all providers.

[1] The Object-Oriented distributed middleware used in demo, available at www.zeroc.com

According to this, each mall would send around 7 MBytes per day (assuming 312 working days) which is easy-to-use amount of information. On the contrary, providers would receive the amount of events related to their products sells, but in this case, they can balance the amount of information duplicating the resources assigned to receive the events.

F. Security and legal considerations

As it has already being mentioned, all communications are ciphered by a public key infrastructure. The main information broker is the manufacturer company, since it is receiving all the events that are related to its products and it is providing information to the remainder entities. This role could be played by a company with this business chain, obtaining profits from managing the information with all the involved entities.

The information included in the customer profile is voluntary and it is compulsory that users give their permission and approval to share that information to different entities.

V. Demo

We are implementing a demo in order to show the feasibility of our approach. Now, we are focusing in two main aspects, first, the software infrastructure, starting in the manufacturer part of the architecture (figure 3). The interface implemented by this service is:

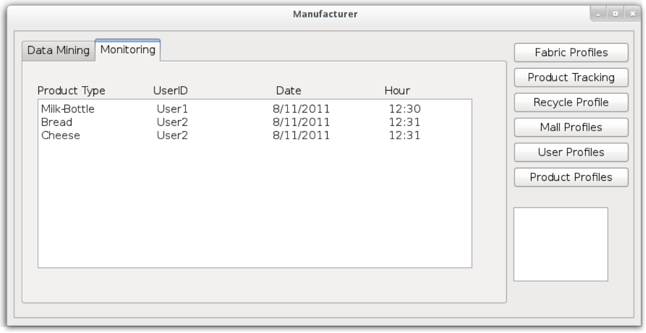
```
interface ProductTracker {
  void dispatched(Ice::Identity factoryID,  ProductID pid);
  void received  (Ice::Identity mallID,     ProductID pid);
  void sold      (Ice::Identity mallID,     ProductID pid);
  void acquired  (Ice::Identity userID,     ProductID pid);
  void dispose   (Ice::Identity userID,     ProductID pid);
  void dropped   (Ice::Identity recyclerID, ProductID pid);
  void recycled  (Ice::Identity recyclerID, ProductID pid);
};
```

The other aspect is RFID user reader deployment, evaluating different RFID readers to see which one could be more appropriate for installation at home. We are working with Omnikey CardMan RFID reader attached to a low-cost fonera wireless device. In last term, we could set up a network of RFID readers connected between them by mean of an ad-hoc network. This configuration could be suitable from large store in malls to a consumer small-home.

From the scalability point of view we are implementing a testbench IDM simulation with omnet++[7]. To this simulation work, we are planning to connect directly our software infrastructure and estimate some traces about realistic product transactions. The main idea is to do a realistic stress test to get estimations about bandwidth, average event generation, etc.

VI. Conclusion

This paper proposes an architecture for massive RFID product tracking. The author of this article strongly believe that only by involving user, companies will be involved in IoT ecosystems. In the proposed business model a novel global tracking system for products is proposed, in which users could obtain benefits from recycle activities.



Product Type	UserID	Date	Hour
Milk-Bottle	User1	8/11/2011	12:30
Bread	User2	8/11/2011	12:31
Cheese	User2	8/11/2011	12:31

Figure 3. Screenshot of manufacturer application

After a requirements phase collection, a scalabe and efficient architecture has been designed to support the proposed business model. It has to be highlighted that the architecture is also suitable and useful for a broad type of advanced services. Currently, deployment issues are being analyzed and the involvement of local manufactures is also being pursued.

Acknowledgment

This research was supported by the Spanish Ministry of Economy and Competitiviness under the project DREAMS (TEC2011-28666-C04-03) and by Indra under the Catedra Indra UCLM 2010.

References

[1] Y. Wu, D. C. Ranasinghe, Q. Z. Sheng, S. Zeadally, and J. Yu, "RFID enabled traceability networks: a survey," *Distributed and Parallel Databases*, vol. 29, pp. 397–443, Jun. 2011.

[2] G. Lee, J. Shin, D. Park, and H. Kwon, "Discovery architecture for the tracing of products in the epcglobal network," in *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*, vol. 2, dec. 2008, pp. 553 –558.

[3] D. Schreiber, "Smartproducts final architecture and specification of platform core services," 2011.

[4] W. Zhao, X. Liu, X. Li, D. Liu, and S. Zhang, "Research on hierarchical p2p based rfid code resolution network and its security," in *Frontier of Computer Science and Technology (FCST). 4 Int. Conf. on*, dec. 2009, pp. 103 –111.

[5] D. Villa, F. Villanueva, F. Moya, G. Urzaiz, F. Rincón, and J. López, "Object oriented multi-layer router with application on wireless sensor-actuator networks," in *16th IEEE International Conference on Networks, ICON*, 2008, pp. 1 –7.

[6] Mercadona, "Mercadona annual report 2010," Mercadona, Tech. Rep., 2010, http://www.mercadona.es.

[7] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Simutools '08*. Brussels, Belgium: ICST, 2008, pp. 1–10.