A Software as a Service RFID Middleware for Small and Medium-sized Enterprises

Jürgen Müller, Martin Faust, David Schwalb, Matthieu Schapranow, Alexander Zeier, and Hasso Plattner

Hasso Plattner Institute for IT Systems Engineering
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany

{juergen.mueller|matthieu.schapranow|alexander.zeier|hasso.plattner}@hpi.uni-potsdam.de
{martin.faust|david.schwalb}@student.hpi.uni-potsdam.de

1 Introduction

Strategies for enhancing supply chain performance often require data sharing between supply chain partners to remove inefficiencies such as out-of-shelf situations, incorrect inventory levels, and incorrect demand and sales forecasts [1]. Radio Frequency Identification (RFID) combined with the Electronic Product Code (EPC) makes it possible to use unique identifiers for each object [2]. RFID furthermore allows the automation of collecting events and lowers costs for enhancing information granularity about an object traveling through the supply chain [3].

Given the fact that Small and Medium-sized Enterprises (SMEs) are integrated into a global Supply Network (SN) with modern Supply Chain Management strategies like Vendor-Managed Inventory, Collaborative Planning, Forecasting and Replenishment, and Demand-Driven Supply Network, they have to invest in technologies like RFID if they want to become or stay part of the SN [4]. SMEs, e.g. Wal’marts suppliers, are often subject to mandates or legal requirements which require the companies to implement a certain technology [3]. This will probably happen in the pharmaceutical supply chain over the next few years [5]. In contrast to large enterprises, SMEs do not have the financial capacity to invest in modern technology which might not even bring the promised advantages [6]. This often results in a difficult situation for SMEs. On the one hand, they depend on being part of a modern SN. On the other hand, they only want to invest in technologies which have proven to bring substantial advantages. Furthermore, SMEs are facing various problems by the integration of RFID technology. The IT resources of SMEs are limited to the needs of the companies’ everyday business [6] without additional capacities for an RFID implementation. The high upfront costs coming with such an implementation are also a burden for many SMEs.

In the end, the SME has to make a make-or-buy decision. We claim that it is possible to serve SMEs with a Software as a Service (SaaS) solution [7] especially for their purposes. In other business areas this already proved to be successful as examples like Salesforce.com show. In this contribution we will present the — to the knowledge of the authors — first SaaS solution for a RFID middleware especially for SMEs.

In this paper we will first describe the requirements on a SaaS RFID middleware. Then we point out our solution and describe its architecture. After that we point out business models and a conclusion summarizes the paper.

2 Requirements

Requirements should be elicited in a structured manner. In this section we will state requirements on a SaaS RFID middleware for SMEs and structure them compliant to the ISO/IEC 9126 standard [8].

2.1 ISO/IEC 9126 Standard

The ISO/IEC 9126 standard describes software engineering product quality. It divides product quality into three parts: quality in use, external quality, and internal quality [8]. Quality in use considers the user’s quality requirements when the software is used. External quality is the set of software product characteristics from an external view when the software is executed. Internal quality describes the rigor related to software design and construction. As proposed in ISO/IEC 9126-1 [8] this classification can be used for structuring requirements.

2.2 Requirements on the System In Use

Simple usage In order to enable the employees of SMEs to use the software even without special training the system has to be easy and intuitive to install and use.

Reliability The system has to be as reliable as an on-premise installation. Because the customers store valuable business information inside their RFID systems, the provider has to guarantee for the reliability of the solution.

Productivity Experiences show that the acceptance of a system rises if the users productivity is increased by the system [9]. Hence, this has to be ensured by the system.
2.3 External Requirements on the System

**Scalability** To operate efficiently, the SaaS RFID middleware provider wants the solution to be scalable. Therefore it should be applicable to small customers and large customers at the same time. This results in two problems: (1) provide enough processing power to serve very large customers and scale down efficiently to be able to serve as many SMEs as possible with one server. Because of the desired target audience the latter has the most impact on total costs of the provider.

**Less hardware at SME** In order to achieve simple installation and maintenance, only RFID readers and - if necessary - RFID writers should be located at the SME site. IT personnel is expensive and often SMEs do not have IT experts employed, which can administer the RFID solution. Therefore the less hardware on-premise, the lower the maintenance costs.

**Security** The SaaS RFID middleware solution must be as secure as a conventional on-premise solution. Because the RFID data include companies’ secrets it is absolutely necessary to store this information in a secure manner. A SaaS RFID middleware solution offers SMEs the chance to get security-standards of large enterprises which they probably could not realize on their own.

**Standard compliance** EPCglobal has published standards for sharing EPC-related data across and within enterprises [10]. To allow the interaction with other companies the SaaS RFID middleware solution should implement the EPCglobal standard. This ensures that all advantages and the inter-operability of the SN can be achieved.

2.4 Internal Requirements on the System

**Efficiency for operator** The Total Cost of Ownership of a data processing center is mainly influenced by the total number of machines. Therefore, to maximize the profit of the SaaS provider, the solution should use the resources as efficiently as possible. The technical administration is also a cost factor. The more efficient the SaaS solution can be maintained, the less IT personnel is needed. Therefore, the architecture should be designed in a way that customer-specific needs can be adopted easily.

**Cost reduction by multi-tenancy** To leverage the long tail, i.e. many small customers, multiple customers are served by a single instance which results in a multi-tenancy middleware [11].

3 Solution

In this section we describe our proposed architecture for a SaaS RFID middleware. The basic idea is to install only a minimal amount of IT equipment on-premise and have the middleware provider to install most of it. Arising questions concerning multi-tenancy, scalability, and reliability are discussed.

---

1The notation is a Fundamental Modeling Concept (FMC) block diagram [14]
3.2 IT at the SaaS Provider

In the following we will describe the components of the middleware as shown in Figure 3.1 and discuss the flexibility and scalability of our approach.

Components The middleware contains a set of data per customer. This data is divided into configuration data, rules, and access rights. The configuration data contains general settings for the customers. With rules it is possible to configure the filtering and aggregation functionality and to specify actions which are executed if all conditions of a particular rule are fulfilled. The access rights guarantee that only authorized stakeholders can perform critical actions like submitting read events, changing the configuration or executing queries.

The configuration of all those settings is done by the account configuration component which provides a web interface for access from outside.

The hardware management component is connected to the agents running on the customers RFID readers and writers. It receives occurring read events and initiates their further processing by the event processing component.

The event processing component executes actions triggered by rules, e.g. the forwarding of events to enterprise resource planning systems.

For collaboration with other stakeholders the middleware provides an EPCIS [15] which can be queried about EPC data. Using this component we can provide a connection to existing enterprise information systems like SAP R/3.

Scalability The SaaS RFID middleware solution targets SMEs and the expected profit per customer is low. To still operate profitable it is necessary to have a highly scalable architecture. This is achieved by leveraging multi-tenancy on application level to allow several customers to share one server and by separating application and database servers so that they can be scaled individually. This enables the distribution over different application servers. If the load balancer is configured to route requests from the same destination to the same application server, the application servers can also gain a higher efficiency with the use of caching mechanisms, e.g. at the filter settings or the EPCIS configuration.

The EPCIS is a storage system for EPC data and therefore the performance depends on the underlying database installation. If one EPCIS is used for all customers, its database can be partitioned to achieve a maximum of performance [16]. Recent data could be held on faster servers and historical data on slower ones, because the access to EPCs which are accessed recently is more likely to appear again in the near future.

Flexibility Customers have the need to easily implement rules for their event processing to adapt to their changing requirements. Nowadays specialized personnel is needed to create, implement, and maintain the complex rules. To meet the targets of SMEs it is desired to have a rule configuration mechanism that can be used by people without special knowledge. We propose to do this with an intuitive and graphical application for modeling rules and the reflected business processes, e.g. like the Oryx editor [17]. This software should be able to model the rule-based business process via an already established modeling language like Event-driven Process Chains or Business Process Modeling Notation. We implemented a first prototype as shown in Figure 1 which is based on the Oryx editor. The customer can use this editor to define own rules in a flexible manner.

4 Business Model

We identified three different sources of revenue for the provider of a RFID SaaS middleware: fixed-fee revenues [12], transaction-based revenues [12], and the commercialization of information.

Fixed-fee contracts Having fixed-fee contracts, a user of the SaaS middleware pays a monthly usage fee. This fee includes a defined number of read events.

Transaction-based contracts Fixed rate contracts could be extended by transaction-based contracts which follow the principle of pay-per-use. Especially for SMEs with a low volume of RFID read events, this opportunity can lead to cost savings.

Sell Information A completely different approach to generate revenue is to sell the information included in the read events. This data has to be anonymized and the usage of this data is only allowed if the SME agreed to this and no company secrets are unveiled.

5 Summary

This contribution points out the possibility for SMEs to lower costs for RFID-related software investments. We first motivated our approach, listed requirements on a SaaS
RFID middleware, and then presented the first SaaS RFID middleware especially for SMEs. Using this middleware SMEs are in the situation that they can be part of a modern SN by conducting RFID implementations without high upfront costs for software. This strengthens the position of SMEs and makes them more competitive.

6 References