Evaluating RFID Infrastructures

Sebastian Frischbier, Kai Sachs, Alejandro Buchmann
Databases and Distributed Systems Group
{frischbi, ksachs, buchmann}@dvs1.informatik.tu-darmstadt.de
Dept. of Computer Science, TU Darmstadt, Hochschulstr. 10, 64289, Germany

Abstract

Investing in RFID technologies along the Supply Chain seems to be the choice of time as a lot of literature and media coverage is about its benefits [1, 2, 3, 4]. Nevertheless, the productivity enhancements achieved by using these technologies strongly depend on the software infrastructure being used. Thus, the extent of benefits regarding productivity can not easily be predicted without evaluating existing RFID infrastructures from a technical *as well as* a business-related point of view. Most research available today is made from a clearly business-related point of view or is restricted to abstract software models and recommendations [5, 6]. Given the costs and complexity of implementing an RFID infrastructure, many companies adopt preexistent solutions. However, the decision of which commercial solution to invest in should be based on a structured and comparable analysis. Surveys and reports like [2] and [3] show that companies still struggle with this problem. In this paper we point out economic *as well as* technical criteria based on our experiences [7, 8] and current literature with special regard to business-aspects. Putting these criteria together, an evaluation model for RFID infrastructures is formed which suits both technical as well as business-related considerations. Our model allows an easy characterization and comparison of different implementations being incomparable on the first sight. Due to its flexibility, it could be used either focusing on software infrastructures already being used by a company (e.g. ERP, SCM) or independently. Our model could easily be customized by stressing or neglecting assorted criteria.

1 Introduction

1.1 About Infrastructures

In the context of *Radio Frequency Identification (RFID)*, the term *RFID infrastructure* describes the IT infrastructure which is necessary to collect, filter and enrich raw RFID data before being processed by the backend systems (i.e., business intelligence systems like ERP) [9]. In this paper we focus on the software components doing this job and therefore the terms *middleware* and *infrastructure* are to be used exchangeably.

In order to provide a uniform technical description of each vendor's solution, we have derived a set of evaluation criteria. Furthermore we have defined three phases the act of processing RFID data typically has to go through if working properly. This was done by identifying and generalizing the several steps to be performed. Hence the abstract task of preprocessing data could be distiguished into three phases:

1. collecting data by managing the RFID reader(s)
2. enriching this collected data for further use (e.g. by filtering, aggregating, etc.)
3. exchanging enriched data with backend systems

Thus we have an n-tier design approach for RFID middleware (usually a 3-tier architecture presuming one layer for each phase). We show in the next sections, nearly all solutions meet this approach.

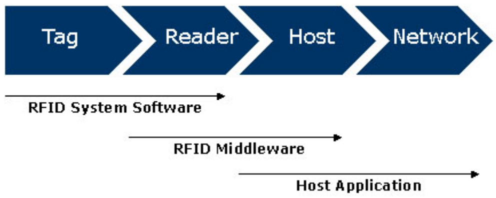

Figure 1: Components called middleware regarding [10]

1.2 The next Step

In our opinion, the current state of RFID technology is very much the same as that of application server (a.k.a. *enterprise systems*) such as the *Java Enterprise Application Server (J2EE* and now *Java Enterprise Edition 5)* was some years ago. Today, the market of enterprise sytems is consolidated and still growing, being worth billions of dollars. Application servers were the first systems offering a new combination of features and services for deploying and using software components in heterogeneous systems. Furthermore they allowed a better decoupling of the presentation, business logic and data storage layers. Hence, the usability and *r*eusablilty of software components became easier and less expensive.

These characteristics could be found again in the RFID technology. Similar to application servers, RFID infrastructures use standardized interfaces which allow a substi-

tution of systems to some extent.
There has been some struggle at the beginning of the era of application servers not only regarding standardisation but also regarding the question of how to categorise them and compare their features. This caused companies to be reluctant to adopt this new technology at first. With the establishment of a standardisation and certification system (e.g. [11]), this behaviour changed.
We believe, that there are strong parallels between both technologies, app. servers then and RFID now. The markets related to RFID are growing with an enormous speed: "*According to Gartner the RFID market will grow from $504 million in 2005 to $3 billion in 2010*" [4].
Thus, the need for comparable systems becomes obvious. To provide such systems and the tools to measure their suitability is our next challenge.

1.3 Performance vs. Usability

RFID infrastructures have to perform a difficult task while dealing with a very heterogeneous environment (e.g. readers, databases, label printers, backends etc.). The more complex and heterogeneous a system is, the more you want to use techniques for staying atop the jungle of different platforms and hardware. One way is to use software layers (*indirections*) to provide a uniform programming interface while covering the heterogeneity beneath. A good example for this technique is the Hyper Text Transfer Protocol (HTTP) hiding differences in operating systems and hardware to the Internet user. Using indirections allows to concentrate on implementing the business logic without having to think about heterogeneity of the underlying system. The problem is that this increase in comfort is payed by a decrease in performance. A level of abstraction hides the underlying heterogeneity to the layer above, the new implemented layer itself has to deal with it by transforming different message formats from the underlying layers to the uniform format offered by the new layer.
As we will see further on, this becomes a big problem when dealing with a critical mass of RFID data being processed by protocols based on the *eXtensible Markup Language (XML)*. Since XML became the standard for data exchange and processing in the last years, performance problems in connection with XML are a well known problem and in focus of current research literature [12].

2 Criteria For Evaluation

The bibliography dealing with RFID middleware offers several criteria for evaluating RFID systems [9, 3]. We have summarized the most common ones, restructured them, and extended them with special regards to business aspects. Other criteria are based on our experience and research activity in the area of RFID infrastructures and other middleware environments.
We decided to group these criteria in three main groups:

1. *Technical Criteria*
2. *Integration Criteria*
3. *Economic Criteria*

The technical criteria deal with technical possibilities offered by a specific infrastructure and are therefore more general. The criteria of the second group are mostly related to the interfaces an RFID infrastructure offers in regard to the *already existing software environment* of a company. The last one covers the economic aspects like costs or human resources. Some of these criteria are related to others. However we think it makes sense to evaluate them separately.

2.1 Technical Criteria

2.1.1 Scalability

An infrastructure has to offer the possibility to be scalable for different required workloads in an easy way. If the number of transactions increases it should not be difficult to scale up the enviroment to avoid a collapse and keep a certain quality of service. Being in the line of fire, the RFID middleware has to offer features for dynamically balancing processing loads and handle large amounts of data and their preprocessing (like database accesses, data exchange, aggregation etc.) [13]. This topic covers the question whether a system is extendable and how to extend an already implemented system.

2.1.2 Commitment to Standards

Supporting common standards simplifies upgrading, migrating and scalability of an existing infrastructure. Concerning this topic, we concentrate on the exchange of information between the devices, infrastructure and backend systems. This topic goes hand in hand with the question of application integrability. We consider that it is to divide this topic further into two categories: *technical standards* such as J2EE or XML and *standards of data and business logic exchange* such as EPC PML.

2.1.3 Data Processing Capabilities

Besides collecting data, RFID middleware needs to filter and enrich raw RFID data in order to transform those flows into single events. In this regard, the following questions arise: What is the level of compression (e.g. by aggregation)? Are there any possibilities to configure the subset of information needed according to the connected backend systems (e.g. highly compressed and batched reports vs. raw data streams [9, 13])? What about attaching meta data from backend systems or local repositories to read data? Could the original raw data be preserved (e.g. for detailed analysis)?

2.1.4 Sharing of System Functionality

In reality information has to be spread across sites, countries and even across different organisations. Thus, RFID infrastructures have to support a sharing of system functionality by their architecture. To rely simply on a wide range of supported standards does not necessarily mean to

be able to share systemfunctionalities properly. Relying on standards is always a good choice but in this case it has to be supported by a modularized software architecture. Using a component oriented design *in connection* with a wide range of standards seems to be a step into the right direction from the point of view of the software design. Due to these reasons we treat this topic as a single item although being closely related to the question of standards being supported.
Sharing of system functionality itself should also include the ability to share information with partners in the business process. An interesting approach for instance are the *EPC Information Services (EPC IS)* [14].

2.1.5 Performance

The question of performance has always to be seen in relation to the questions of estimated throughput rates, response time and availability of a system under stress. In case of only few messages to be processed, too much care about performance is not necessarily required. Having thousands or millions of messages to be processed (in a short period of time) requires a well designed system being able to bear the burden. In addition to these general suggestions, more concrete questions arise: How could we measure and monitor the performance of a running system? Are there any comparable benchmarks available for this system? Are there any tools for measuring performance being shipped with? What are the benchmarks of a solution given server scenarios?
As we will see, especially parsing of XML-based communication is a main threat to performance. This is even worse due to the fact that XML is a commonly used technique to solve the problem of the communication in heterogeneous systems.

2.2 Integration Criteria

2.2.1 Integration into existing Software Environments (Application Integration)

The RFID middleware has to cooperate with several systems like Warehouse Management systems (WMS), Supply Chain Management systems (SCM), Enterprise Resource Planning systems (ERP), Business Intelligence systems (BI) or Customer Relationship Management systems (CRM). Therefore, the following questions arise: Does each infrastructure require a specific environment/system to work properly? How strong are the dependencies between these environments? Are there any adapters available or have other precautions been taken (e.g. by means of a service oriented architecture)? These questions are closely related to the issue of commitment to standards (previously discussed in Section 2.1.2). Nevertheless, we believe it is convenient to treat them as a separate concern.

2.2.2 Customizability

Since the environments and flow of work of companies differ, an RFID infrastructure has to provide possibilities for customization to fulfil the requirements. Built-in criteria for filtering, processing, and routing data should be configurable. Also it should be possible to include customer code or adding third-party modules. Configurable features should cover as much functionality as possible, since configuring a system is much cheaper and easier to maintain than developing own customer code.

2.3 Economic Criteria

2.3.1 IT Landscape

A very general question is whether the RFID infrastructure matches together with IT guidelines of the company and the existing enviroment. This covers for instance the questions of which databases and operating systems are supported or if a specific vendor is favored. This is not a technical criteria but a long term challenge and thus a strategic management descision.

2.3.2 License Models and Hardware Costs

This criteria covers two main topics:

- What are the initiale costs for the RFID infrastructure?
- What are the future (license) and support costs?

The first item covers the costs for the initial hardware and software deployment. The second item deals with rates for future support (such as updates, service calls etc.). An example would be a question such as: How much would it cost to get support for a year?

2.3.3 Training Operators

Adding a new system into a company's flow of work might imply that some of the employees have to be trained to use this system properly. Depending on the influence of the system on the daily flow of work of employees, a training program could need to be arranged.
Basically, two groups of employees exist:

- End users
- System administrators

The first group covers all users whose flow of work is related to the RFID infrastructure. This includes employees registering stock receipts as well as managers, analysing SCM activies. Of course the training of these employees can be very different in content and complexity.
The second group includes the employees which manage and maintaine the overal system, ensureing that the devices are operated properly by the endusers.

2.3.4 Safety of Investment

The safety of an investment in an RFID infrastructure is influenced by mainly two aspects:

1. Support of future technologies

2. Market position and reliability of vendor

If the RFID infrastructure is to be used for a long period of time, both of this aspects should be fulfilled. The first is partly related to the question of standards being supported. Support for future technologies may not necessarily be expressed only by the range and extent of standards being supported and implemented. A firm basis for future technologies could also be achived by having a modular design which allows easy extension afterwards, or even by implementing technologies being at a research state. Nevertheless it is risky to invest in a solution relying mostly on future technologies too much because the risk of pushing ideas without future may not be underestimated. The short history of enterprise architectures and standards shows many examples for interesting ideas having not gotten into practice. However, this risk could be minimized by chosing a solution with focus on a solid base of proven technologies in connection with either a modular design or interfaces to non-stable technologies as an additional feature. The final question, whether to invest into solutions claiming to be best prepared for the future by relying on unproven technologies or not is a strategical question to be solved by the management.

The second aspect handles the vendor. Many vendors offered (good) software solutions but did not offer reliable support. For this purpose it is helpful to analyze the behavior of the vendor in the past. Did the vendor publish updates (e.g. service packs) regulary? Since when does it exist? Is it likely that it will still exist in three years? The vendor's current position on the market could be used as an indicator for that. Choosing a strong vendor is normally a guarantee for support of established technologies even in the future.

3 Selection of Vendors - SAP

In the following sections of this paper we want to attach our evaluation criteria from Section 2 to an available software infrastructure. This evaluation is supposed to be an expample and intended to show how to use certain criteria on a specific infrastructure.

Concentrating on the market leaders and other strong performers seemed to be the best way of giving a representative view of today's existing solutions. Hence we tried to identify those among the large number of RFID system vendors. Based on [3] we narrowed the range of probable candidates with regards to the amount and quality of the available documentation as well as to single significant characteristics of each candidate.

We chose *SAP* to be the commercial vendor being introduced in detail due to several reasons:

SAP is not only a leader in the market regarding business intelligence software and large scale infrastructures but has although introduced and supported many revolutionary technologies in this field [15]. Most large companies run SAP-based backend systems, relying on the long term experience of this vendor. SAP and its technologies are seen today as a quasi-standard regarding business intelligence.

Further solutions by other vendors (e.g. Microsoft, SAVI and Sun Microsystems) as well as Open Source approaches (e.g. Singularity, RadioActive and rfid project) will not be covered in this paper. For more detailed information about solutions by Sun we refer to [16] and [17] as well as to [18] regarding Microsoft's. An overview of interesting Open Source approaches in progress can be found in [19] (RadioActive),[20] (Singularity) and [21] (OSI)

SAP offers an RFID-add-on for its business intelligence landscape which is called *SAP Auto-ID Infrastructure (SAP AII)* and was a founding member of the Auto-ID center (now called EPC Global) in 1999. SAP began to develop its AII in 2001[22] and was one of the first vendors with a market-ready solution.

SAP AII is a component for the SAP NetWeaver platform.[1] SAP promotes it as an infrastructure providing "out-of-the-box functionality to fulfill requirements for RFID compliance in the logistics applications of the U.S. Department of Defense, the U.S. Food and Drug Administration, and large retailers such as Wal-Mart." [24]

In this section we will first discuss the abstract design of the SAP AII . In the second part we will have a closer look on the implementation including the different components of the SAP NetWeaver platform.

3.1 Abstract Design

We start by presenting the research approach of an RFID system including SAP AII as mentioned in [13]. It is designed as a 4-tier architecture which is illustrated in Figure 2:

1. *Device Layer*: RFID readers and other input/output devices (e.g. printers).
2. *Device Operation Layer (DOL)*: Reader management, low-level filtering and aggregation. Consists of one or more *Device Controlers (DC)*.
3. *Business Process Bridging Layer (BPBL)* consisting of one or more *Auto-ID Nodes (AINs)*, *an Auto-ID Administrator*[2] and a local Auto-ID repository which is independent of the backend system (to store local inventory information as well as additional master data). This layer serves as a negotiator between the DOL and the backend systems.
4. *Enterprise Application Layer* containing backend systems for business intelligence (SCM, CRM, ERP, etc.). Note that they are not restricted to SAP systems at this level.

According to [13] SAP AII's core consists of layers two (DOL) and three (BPBL). Comparing our abstract design scheme with AII's architecture shows that step one is translated into action by the Device Operation Layer, whereas steps two and three by the Business Process Bridging Layer.

[1]For further information about SAP NetWever see e.g. [23].

[2]From now on reffered to as *Auto-ID Cockpit (AIC)* [25][26]

3.1.1 Device Operation Layer

As mentioned above, one or more connected DCs set up the DOL. A DC manages several readers (attached via a publish/subscribe interface[3]). Furthermore a DC is responsible for a low-level filtering of read data, its transformation into events and handing over these events to the BPBL. The low-level filtering is done by so called *Data Processors*[4] which could be distinguished into six different types according to [13]:

1. *Filters* receive and filter incoming data according to a defined level (e.g. item-level vs. pallet-level)
2. *Enrichers* read meta data stored at the RFID tag memory of the current item and add them to the event.
3. *Aggregators* bundle low-level events to *higher-level events* such as *temperature_increased_event* which summarize the tracked values of several temperature sensors over time. [13])
4. *Writers* write new or changed data on tags.
5. *Buffers* keep temporary inventory information (tags being in a reader's scope)
6. *Senders* transform internal events to a defined format like PML/XML and send them to registered subscribers

A DC is designed to work in two modes. Being at the *asynchronous listening* mode, a DC *waits* for incoming events from the connected readers. The *synchronous* mode means a DC receives direct device operations (e.g. read/write commands) from the BPBL atop and gives an immediate feedback. In sum, the *asynchronous listening* mode refers to the layer beneath, the *synchronous* mode to the layer atop. Thus, being at asynchronous mode enables to execute orders by the level atop at the same time.

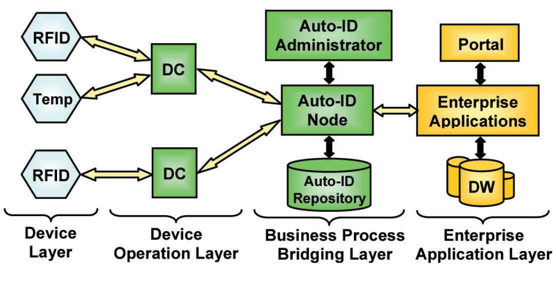

Figure 2: RFID system as described in [13]. Core SAP AII components (Device Operation Layer, Business Process Bridging Layer) are coloured green

[3]in this case, a DC subscribes at several readers in order to recieve RFID data. Readers subscribe at DCs in order to be notified for updates the other way round

[4]As we will see later on, the whole DOL is treated as a third-party component (provided by device vendors for example) by SAP in terms of implementation

3.1.2 Business Process Bridging Layer and Auto-ID Node

Similar to the DOL, the BPBL consists of one or more *Auto-ID Nodes (AIN)*. An AIN has to integrate data from several DCs into business processes defined at the backend systems. That means, aggregated and filtered RFID events from the DOL have to be interpreted in terms of business aspects in order to be suitable for the backend systems. This is done by applying predefined rules on those incomming events. A *Rule Engine* is used to manage a hierarchical structure of those rules. One or more actions could be assigned to each one of those rules. In addition rules can trigger other rules even in other AINs. For example, reading the EPC-tag of a tracked object followed by updating the status of that object (e.g. "a single item has been stored on a specific pallet", "object has left warehouse") at the local repository as well as notifying the backend systems.
Hence one could easily map business processes to events within a AIN and thus close the gap between raw RFID data and the underlying interpretation of that data in business processes. Later on we will refer to these rules as *Core Services* [25] (see *Implementation (Section 3.3* for detail). Due to that the AIN with its Rule Engine could be called the heart of SAP AII.

3.2 Architecture

What we have seen so far was a description of the abstract design of SAP's AII. Now we have a closer look on the concrete implementation of th AII. SAP AII 2.x consists of the BPBL with it's AINs [25, 22, 27]. The Device Operation Layer with its features as described above is referred to as third-party software [22]. As an additional component the SAP Exchange Infrastructure (XI) is inserted between SAP AII and backend systems so that there is no more direct connection (see Figure 5). Instead of AINs we have Core Services and Integration Services: Core Services consist of a Rule Engine as well as the assigned actions and are shielded to the rest of the system by the Integration Services. Hence we could call each pair of Core Services and Integration Services to form a single Auto-ID Node which can act autonomously.
According to [25] SAP AII consists only of three modules instead of AINs:

1. *Core Services* use a Rule Engine and Auto-ID Repository to perform transformation of RFID data to business process events
2. *Integration Services* encapsulate Core Services
3. *Auto-ID Cockpit (AIC)* manage Core Services and Integration Services

3.2.1 Core Services

The Core Services consist of the described Rule Engine and the assigned actions. These actions are classified according to their subject:

- *Action and Process Management:* Action Handling

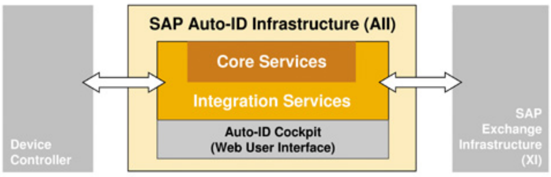

Figure 3: Core Services and Integration Services of SAP AII 2.0 taken from [25]

by Rule Engine as mentioned in *Architecture*, Event Queue, Event Message Dispatcher

- *Configuration and Admin Management* interfaces to devices/users, components, backend systems
- *Object Data Management* Supervising objects (expected actions, current state, trace).
- *Lean Master Data Management* meta data (e.g. product description) provided by backend systems and kept at the local Auto-ID repository.

3.2.2 Integration Services

They are used to enable the interaction between AII and the following three environments:

- *Human Integration*: Administration through Auto-ID Cockpit
- *Backend System Integration*: Connection to backend systems on the one hand by the use of the following two kinds of adapters: *Communication Adapters* (which provide support for several protocols[5]) and *Application Adapters* (to convert data directly). Provides API to access Core Services on the other hand.
- *Device Integration*: Similar to Backend System Integration

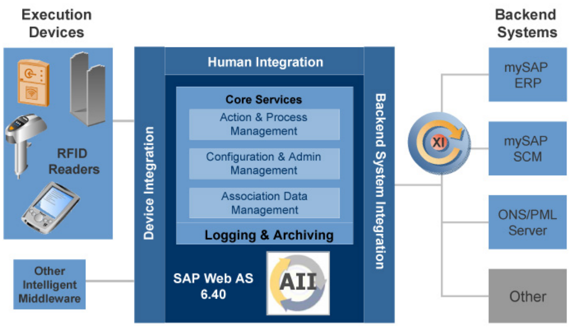

Figure 4: Scheme of an RFID system taken from [22] with SAP Exchange Infrastructure (XI) as adapter

[5]Supposedly XML via HTTP due to the fact that AII is based on SAP WebApplication Server. See *Implementation* for further detail.

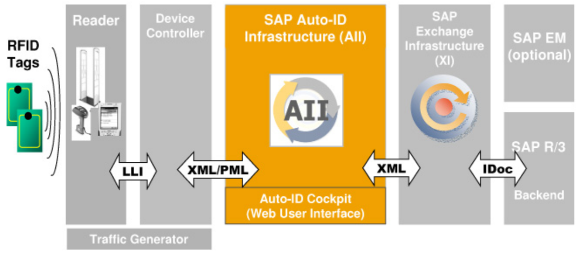

Figure 5: Scheme of an RFID system taken from [25]. SAP AII components are coloured orange. *IDoc* is an SAP-specific XML-Format

3.3 Implementation

3.3.1 SAP Web Application Server (WebAS)

The *SAP WebApplicationServer (WebAS)* contains a J2EE application server as well as an ABAP application server (called *Stacks*). ABAP stands for Advanced Business Application Programming and is a specific 4GL programming language used by SAP. We will have a closer look at it in the following section. WebAS provides support for open Internet standards (e.g. HTTP, HTTPS, SOAP and SMTP) and open document standards (e.g. HTML, XML) [27]. Using the SAP specific *Internet Communication Framework (ICF)*, programms written in ABAP can access Java or .NET components and vice versa. In addition, ABAP can process HTTP requests thus serving as a client as well as a server. At default, both stacks are installed. WebAS provides *Open SQL for Java* which encapsulates the underlying database from the developer.

3.3.2 SAP Exchange Infrastructure(XI)

The SAP Exchange Infrastructure(XI) is a standalone application and has already been introduced before. As part of the SAP NetWeaver platform it runs on the SAP Web Application Server with all its benefits. In addition, XI supports cross-component business process management (BPM) between different business applications. Based on the principles of an *Service Oriented Architecture (SOA)* for enterprise systems it allows easy integration of Web Services and open standards. A tool set allows the definition of messageing interfaces, mappings and routing rules to fit in all environments with standardized interfaces.
In our case it serves as an adapter between SAP AII and different backend systems [28].

3.3.3 SAP AII 2.1

SAP AII 2.1 is designed as a NetWeaver component and based on the *SAP Web Application Server 6.4 (WebAS)* as well. Since SAP AII 2.1 uses a pure ABAP-stack with its own scheme at the local Auto-ID-repository. SAP recommends Devices from Connecterra, ACSIS and Infineon [22] to serve at the Device Layer.

4 Evaluation

4.1 Technical Criteria

4.1.1 Scalability

SAP AII offers more than one possibility to scale the whole system by its architecture: Firstly, several instances of AIN (with attached DCs) could be combined by rules at the BPBL. Secondly, several DCs could be combined at the DOL. Due to the fact that the DOL is not implemented as a part of the "core" AII, this has to be achieved by connecting several third-party DCs to the Core Services (by using suitable adapters).

4.1.2 Commitment to Standards

SAP AII supports standards published by EPCglobal regarding EPC-tags including support for GTIN, EPC number range and EPC-tag generation. Nevertheless there is, as far as we know, no explicit support for Application Level Events (ALE). Supports of future EPC standards is planned for future releases.
Concerning interfaces, SAP AII supports common standards as mentioned before: XML, PML, HTTP, HTTPS, SMTP, IDoc (an SAP specific XML-subformat), J2EE (via WebAS' Java Stack) and .NET through the SAP Web Application Server and NetWeaver as we have seen earlier.
Something special about SAP AII is, that its interfaces became on its own a quasi standard supported by other vendors like Sun[29] or Infosys[30].

4.1.3 Data Processing Capabilities

The first way to influence the level of aggregation is by configuring the Data Processors at the Device Operation Layer. Further aggregation and enrichment could be reached by a suitable configuration of the Rule Engine at the Business Process Bridging Layer. Meta data could be attached by using the local Auto-ID repository.

4.1.4 Sharing of System Functionality

As we have seen several AINs (Core Services and Integration Services) act autonomously and could be combined by the use of appropriate adapters of each node's Integration Services. Together with the Backend System Integration information could be spread through the whole system. Missing support for processing information on a global scale via EPC IS (and Discovery Services) seems to be a disadvantage on the first view. It has to be mentioned that there is no concrete specification or implementation published by EPCglobal yet. Newer releases of AII are supposed to provide propper support for EPC IS.
Since the platform allows to access the components in different ways like J2EE, .Net or ABAP it is possible to share the system functionality.

4.1.5 Performance

SAP offers a whole set of different performance tools, e.g. SAP AII comes with a tool for simulating RFID traffic, called *Traffic Generator*. These tools could simulate traffic generated from (multiple) readers to the infrastructure as well as from backend systems to the infrastructure and allows heavy-stress tests of systems. In addition SAP offers multiple tools to measure, monitor and log the performance of a system and single components [6]. These tools are not part of the SAP AII but of the SAP NetWeaver platform. If running a SAP backend system some tools are integrated in the central monitoring of the whole system.
Performance experiments executed by our group show that the SAP AII scales even under heavy traffic well.[7]

4.2 Integration Criteria

4.2.1 Integration into existing Software Environments (Application Integration)

In theory there are several ways to integrate AII into existing environments:

- Using the NetWeaver-platform [23]
- Using the Exchange Infrastructure (XI) (recommended by SAP)
- Integrating custom adapters directly into AII
- Parsing the XML-streams provided by the Integration Services.

In addition one could access the ABAP-modules via SAP's *Java Connector* (for Java), DCOM (for .NET), or of course directly from ABAP. Another alternative is using the underlying J2EE - application server for JMX. The Exchange Infrastructure allows a pure data exchange, which is sufficient in most cases.
The cost of work to do this would be an interesting question to examine. However these costs depend heavily on the given enviroment and have to be evaluated for the specific case. Therefore we are not able to provide any numbers.

4.2.2 Customizability

The whole system (Core Services, Rule Engine, Integration Services, Devices, Local Repository) can be configured easily using the web-based Auto-ID Cockpit (or, in a SAP enviroment, via SAP admistration tools). SAP AII offers a set of different pre defined actions which should be sufficent for many cases.
Adapters, new actions, and modules developed by customers can be integrated using the NetWeaver platform. Third-party components for SAP AII can be easily installed as additional components on the system.

4.3 Economic Criteria

4.3.1 IT Landscape

Since this depends on the specific guideline of a company and scale of the planed infrastructure, we can only summarize some facts about the software platform of SAP AII:

[6] E.g. for Java applications: JARM (Java Application Response-time Measurement)
[7] System under test: SAP AII 2.0 Java version, Windows based system.

SAP provides its software for all standard plattforms like Windows, Linux, and Unix. Diverse databases are supported, but especially the strong support of MaxDB by SAP should be mentioned here. The open source database MaxDB (previous known as SAP DB) is hosted by MySQL and certified for SAP enviroments[31].

4.3.2 License Models and Hardware Costs

As we have already mentioned before, this criteria has to be applied on a given estimation of costs and could not be simulated by us.

4.3.3 Training Operators

SAP offers a full set of trainings regarding its systems. Regarding an individual team-training, no further information about the amount of an probable fee could be found. Regarding frequently offered workshops about RFID, the fee reaches from €150 to €395 (June 2006) [32].

4.3.4 Safety of Investment

As one of the major vendors, SAP is known for its reliability. Due to SAP's dominant position on the market SAP AII became a quasi-standard infrastructure. Therefore a future support is warranted.

4.4 Summary

SAP AII has offered multiple features: The first one is the integration into NetWeaver thus admitting easy integration into SAP based systems. The second one is the use of ABAP which is optimized for handling large masses of data and is independent from the underlying database by the use of OpenSQL. The Exchange Infrastructure allows the integration in non-SAP and SAP enviroments. Because the SAP AII includes a sufficient set of different business rules, which could be extended using the NetWeaver platform. It is complettly integrated in a SAP enviroment and can be centrally admistrated and monitored. The NetWeaver platform seems to offer the state-of-the-art technologies. The strong market position of SAP guarantees development and support in future.

5 Conclusion and Future Work

We believe that selecting a specific RFID infrastructure has a serious impact not only on the IT landscape of a company but influences other dimensions of the business-process as well. This leads us to the conclusion that it is very important to base the selection of an RFID infrastructure on a well-defined multi-dimensional model which covers both technical and economical aspects. Unfortunately, nowadays there exists no such model.

In this paper, we provided a selection of criteria to elaborate such a model based on both literature references and our own practical experiences. To give a better overview we have formed three main groups covering both technical and economical aspects. The criteria cover the most important aspects to be considered for evaluating RFID infrastructures. We applied the criteria on the SAP AII, one of the major RFID infrastructures on the market.

This model builds one of the major parts of a framework for evaluating, comparing, and testing RFID environments. While our model presented in this paper focuses on RFID middleware, other parts of such a framework should e.g. handle the support of RFID-technology offered by a backend system as well as giving a definition of a benchmark suite for RFID-based systems based on typical business interactions. One of our next research steps is to give an exact definition of the single parts and the complete framework.

6 References

[1] T3Ci. First steps towards achieving value from your RFID investment. White paper, T3Ci, 2006.

[2] CIO Online. Firmen zögern bei Investitionen in die Funkchips - RFID ist keine Datensammlung. `http://www.cio.de/technik/818500/index1.html`, February 2006. (last visited 06-17-2006).

[3] S. Leaver. Evaluating RFID middleware - Tech Choices. Technical report, Forrester Research, Inc., 2004.

[4] SMART LABEL Solutions. What is the ROI of RFID? Technical report, Smart Label Solutions, Inc., 2006.

[5] OATSystems SANJA SARMA and MIT. Integrating RFID. *ACM Queue*, 2(7):50 – 57, October 2006.

[6] Supply Insight. RFID integration platform. Technical white paper, Supply Insight, 2006.

[7] Sebastian Frischbier. Existing RFID infrastructures. `https://www.dvs1.informatik.tu-darmstadt.de/publications/pdf/ExistingRFIDInfrastructures.pdf`, Februar 2006. Seminar on RFID technologies - Database and Distributed Systems Group, Dept. of Computer Science TU-Darmstadt,Germany. (last visited 06-17-2006).

[8] Kai Sachs. Evaluation of Performance Aspects of the SAP Auto-ID Infrastructure. Master's thesis, TU Darmstadt, 2005.

[9] Christian Floerkemeier and Matthias Lampe. RFI middleware design - addressing application requirements and RFID constraints. Technical report, Institute for Pervasive Computing, Departement of Computer Science, ETH Zurich, Switzerland.

[10] Sun Microsystems. *RFID Field-Guide: Deploying Radio Frequency Identification Systems*. Prentice Hall.

[11] Sun Microsystems Inc. Java EE At a Glance. `http://java.sun.com/javaee/index.jsp`, June 2006. (last visited 06-17-2006).

[12] M. Nicola and J. John. XML parsing: A threat to database performance, 2003.

[13] Christof Bornhoevd, Tao Lin, Stephan Haller, and Joachim Schaper, editors. *Integrating Automatic Data Aquisition with Business Processes - Experiences with SAP's Auto-ID Infrastructure*, Toronto, Canada, 2004. SAP Research, Proceedings of the 30th VLDB Conference.

[14] The EPCglobal Network - Overview of Design, Benefits & Security. http://www.epcglobalinc.org/news/EPCglobal_Network_Overview_10072004.pdf, September 2004. (last visited 06-17-2006).

[15] SAP übertrifft eigene Prognose - Konzern steigert 2005 den Umsatz um 13 prozent und lässt Konkurrenz bei Softwarelizenzen weit hinter sich. *Handelsblatt*, 8:11, 2006.

[16] Sun Microsystems Inc. Sun Java System RFID Software. http://www.sun.com/software/products/rfid/index.xml, 02 2006. (last visited 06-17-2006).

[17] Sun Microsystems Inc. Developing Auto-ID Solutions using Sun Java system RFID Software. http://java.sun.com/developer/technicalArticles/Ecommerce/rfid/sjsrfid/RFID.html, 02 2006. (last visited 06-17-2006).

[18] J. Sirkaner. RFID Enabled Retail Supply Chain. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/RFIDRetSupChn.asp. (last visited 06-17-2006).

[19] The RadioActive Foundation. http://www.radioactivehq.org/. last visited 06-17-2006.

[20] Singularity. http://singularity.i-konect.com/. (last visited 06-17-2006).

[21] OSI - open source innovation. http://opensourceinnovation.org/projects.htm. (last visited 06-17-2006).

[22] Christoph Leflmöllmann. MCA202: Bringing RFID on the fast track, 2004.

[23] SAP. SAP NetWeaver: Providing the Foundation to Enable and Manage Change. http://www.sap.com/solutions/netweaver/index.epx, May 2004. (last visited 06-17-2006).

[24] SAP. Components & Tools of SAP NetWeaver: SAP Auto-ID Infrastructure. http://www.sap.com/platform/netweaver/components/autoidinfrastructure/index.epx, 2006. (last visited 06-17-2006).

[25] SAP Official. SAP RFID technology - SAP Auto-ID Infrastructure 2.0 (AII).

[26] SAP Draft. Sizing the scenario RFID-enabled slap & ship outbound processing using SAP Auto-ID Infrastructure, release 2.1, January 2005.

[27] SAP Master Guide Document. RFID-enabled supply chain execution powered by SAP NetWeaver - using SAP Auto-ID Infrastructure 2.1, March 2005.

[28] SAP Official. SAP Exchange Infrastructure. http://www.sap.com/solutions/netweaver/pdf/BWP_SB_ExchangeInfrastructure.pdf. (last visited 06-17-2006).

[29] Sun. Sun java system rfid software developer's toolkit 3.0. https://sun-rfid.dev.java.net/SJS_RFID_Toolkit.html, 2006. (last visited 06-17-2006).

[30] Girish Ramachandra and Ravishankar Hossur. 309404ce588ebd4000000009 Calling - SAP, Do You Copy? http://www.infosys.com/sap/sap-inside-rfid-special-feature.pdf, 2006. (last visited 06-17-2006).

[31] MySQL. Maxdb - Das professionelle Datenbanksystem. http://www.mysql.de/products/maxdb/, 2006. (last visited 06-17-2006).

[32] SAP. SAP Education: Chancen nutzen, Zukunft gestalten. http://www.sap.com/germany/services/education/index.epx, 06 2006. (last visited 06-17-2006).