RFID Application Development with the Accada Middleware Platform

Christian Floerkemeier, Christof Roduner, and Matthias Lampe

Abstract—**The proliferation of radio frequency identification systems in application domains such as supply chain management requires an IT infrastructure that provides RFID device and data management and supports application development. In this paper, we discuss these application requirements in detail. We also contend that the characteristics of passive RFID technology introduce constraints that are unique to the development of middleware for the RFID domain. These constraints include the occurrence of false negative reads, tag memory variations, the heterogeneous reader landscape, and the limited communication bandwidth available to RFID readers. To address these constraints and the application requirements for filtered and aggregated RFID data, we developed Accada, an open source RFID platform. The paper shows that the Accada implementation, which is based on a set of specifications developed by the EPCglobal community and a number of extensions, such as the surrogate concept and the virtual tag memory service, addresses the majority of the application requirements and limitations of passive RFID technology.**

Index Terms—**EPC Network, Middleware, RFID**

I. INTRODUCTION

Radio Frequency Identification (RFID) technology has recently seen growing interest from a wide range of industries such as retail, pharmaceutical, and logistics [1]. In these domains, RFID technology holds the promise to eliminate many existing business problems by bridging the economically costly gap between the virtual world of IT systems and the real world of products and logistical units. Common benefits include more efficient material handling processes, elimination of manual inventory counts, and the automatic detection of empty shelves and expired products in retail stores [2]. RFID technology has a number of advantages over other identification technologies. It does not require line-of-sight alignment, multiple tags can be identified almost simultaneously, and the tags do not destroy the integrity or aesthetics of the original object. The location of tagged objects can thus be monitored automatically and continuously.

In traditional RFID applications, such as access control, there was little need for an RFID middleware because the RFID readers were not networked and the RFID data were only consumed by a single application. In novel application domains, such as supply chain management and logistics, there is no longer a 1-to-1 relationship between reader and application instance, however. In these domains, many readers distributed across factories, warehouses, and distribution centers capture RFID data that need to be disseminated to a variety of applications. This introduces the need for an RFID infrastructure that hides proprietary reader device interfaces, provides configuration and system management of reader devices, and filters and aggregates the captured RFID data. This frees applications from the need of maintaining connections to individual reader devices or dealing with the idiosyncracies of proprietary reader protocols.

From an application development perspective, it is also important to abstract from the low-level RFID data captured and translate them into more meaningful application events. The detection of an RFID tag with tag ID 3455.3454656 by reader 8745653 would thus result in the corresponding business event that a shipment of razor blades arrived at dock door 14 of the warehouse. Many benefits commonly associated with RFID require sharing these business events across the supply chain [2].

In this paper, we analyze these application requirements in detail. We also present characteristics of passive RFID technology that are unique to the development of middleware for the RFID domain. The main contribution of this paper is a middleware platform called Accada[1]. Accada is an open source RFID infrastructure which implements the interfaces defined in the EPC Network specifications by EPCglobal[2]. The EPC Network, originally proposed by the Auto-ID Center [3] and further developed by the members of EPCglobal, is currently one of the predominant standardization effort of the RFID community. The analysis in this paper shows to what extent the Accada platform that builds on current industry RFID standards addresses the application needs and constraints of passive RFID technologies that we identified.

The rest of this paper is structured as follows. In Section II, we detail the requirements for an RFID infrastructure. Section III provides a brief overview of RFID technology and outlines the constraints imposed by the characteristics of RFID. In Section IV, we describe our Accada RFID platform. We continue by presenting some sample applications that were developed with Accada in Section V. In Section VI, we show how our implementation addresses the application needs and technology constraints, and we review limitations of our implementation and present future work. Before we conclude in Section VIII, we discuss related work in Section VII.

Manuscript received April 30, 2007.

C. Floerkemeier is with the Auto-ID Lab, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (phone: 617-324-1984; e-mail: floerkem@mit.edu).

C. Roduner and M. Lampe are with the Institute for Pervasive Computing, ETH Zurich, 8092 Zurich, Switzerland (e-mail: {roduner,mlampe}@inf.ethz.ch).

[1] www.accada.org
[2] www.epcglobalinc.org

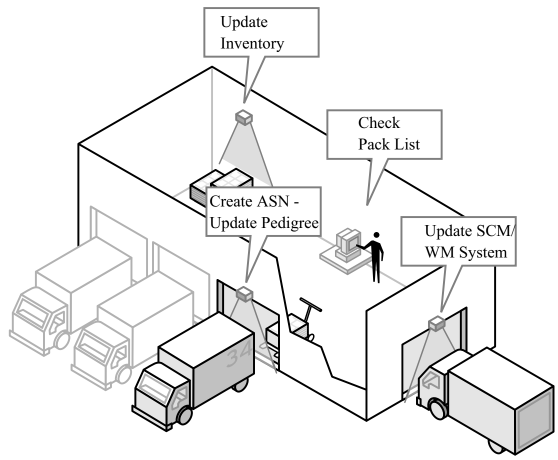


Fig. 1. RFID readers in a distribution center feeding captured data to different applications (source: EPCglobal)

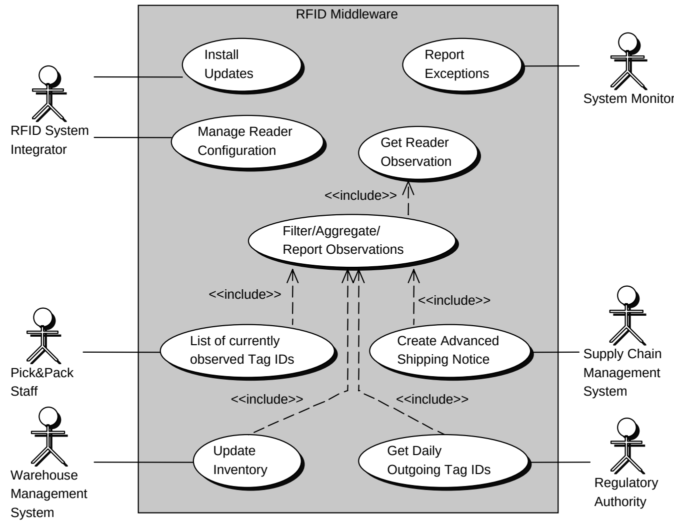


Fig. 2. Use case diagram of RFID usage scenario in a distribution center

II. Application Requirements

The following paragraph outlines a supply chain scenario – a distribution center of a pharmaceutical company (cf. Fig. 1) – that represents a common application of RFID technology. The corresponding use case diagram is shown in Fig. 2. We use this scenario and the use cases to derive the application requirements that an RFID middleware should address.

Goods arrive at the distribution center and are identified by the readers at the dock door. The captured information is transmitted at once to the supply chain management (SCM) system, which provides track and trace functionality. The goods are then placed in the warehouse, where RFID readers scan the inventory. At regular intervals the readers trigger the legacy warehouse management (WM) systems to update the inventory counts of the corresponding product categories. Goods are picked from the warehouse and packed at the corresponding pick and pack station. An RFID reader monitors the tagged items currently packed so that a local application can support staff with a near real-time comparison of items actually packed and the items on the pack list. Before the shipments are loaded into the trucks at the loading dock, they pass a reader that scans the tag on the pallet and passes this information to the supply chain management system, which sends an advance shipping notice (ASN) to the recipient of the shipment. On a nightly basis, all tag IDs of the items packed and shipped are transmitted to the healthcare authorities to comply with pedigree legislation. To maintain an adequate service level, the readers report exceptions to a remote system monitor. RFID system integrators can inspect a configuration of a reader and reconfigure reader devices remotely.

Based on an analysis of different RFID applications including the above, we identified the following application requirements an RFID middleware should meet:

RFID data dissemination. The information captured by a reader is usually of interest not only to a single application, but to a diverse set of applications across an organization and its business partners. The captured RFID data must thus be broadcast to the entities that indicated an interest in the

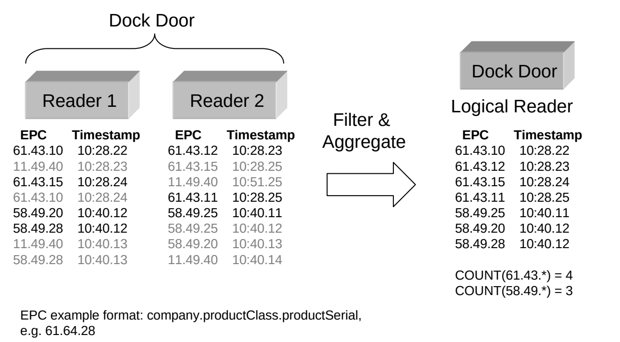


Fig. 3. Filtering and aggregation of RFID data: data of the two dock door readers are combined, duplicate EPCs are eliminated, EPC 11.49.40 is filtered out, and quantities of product categories are calculated. Eliminated ‘reads’ are shown in grey.

data, such as the SCM system. Due to the event-driven nature of many processes monitored with the help of RFID systems, there is a need to support asynchronous messaging as well as a query-response model. In our example, regulatory authorities execute a daily query to retrieve data whereas the pick and pack application is notified asynchronously whenever a new tag arrives. Different applications also require different latencies. Applications that need to respond immediately to local interaction with the physical objects, such as the pick and pack application, require a short notification latency that is comparable to the observation latency. Legacy applications that are not designed to handle streaming data might need to receive batched updates on a daily schedule.

RFID data aggregation. RFID systems generate a significant amount of data that can be aggregated in a number of different ways (cf. Fig. 3). RFID data can be aggregated in the time domain, e.g., by generating entry and exit events, and in the space domain, e.g., by combining data from different readers and reader antennas that observe the same location. In our scenario, a single dock door is monitored by two RFID readers. The RFID events are aggregated and the original reader ID is replaced by the ID of a logical reader that was defined for the dock door location. Legacy IT systems typically cannot handle the instance-level data captured by RFID data. There is thus also a need for count aggregates that report the quantity of objects belonging to a specific category (i.e., product class) rather than the serial number of each object detected.

RFID data filtering. A common feature of all applications that make use of the captured data is the desire to receive filtered RFID events rather than all RFID data captured (cf. Fig. 3). Different applications are interested in different subsets of the total data captured, based on reader, reader antenna, and tag involved. In the previous supply chain scenario, the SCM system might only be interested in tags attached to certain products.

Writing to a tag. Some tags feature not only memory space for an identifier, but also for additional data. RFID middleware should thus provide means to write to and read from this additional memory. This additional memory can then be used to store application data such as expiry dates in order to facilitate data exchange where no network access is available.

Trigger RFID readers by external sensors. In many applications, there is no need to operate RFID readers continuously. Due to the limited bandwidth available, it is even undesirable to have readers transmit while no tags are present [4]. To initiate the tag inventory process at a reader when there are tagged objects arriving in the read range, external sensors, such as motion sensors, should thus be able to trigger the RFID readers. In our example, the readers at the dock door only operate when items enter their read range. The arrival is detected by a motion sensor in front of the dock door.

Fault and configuration management. The proliferation of readers mandates fault and configuration management. This includes monitoring the health of RFID readers and accessing the RFID reader configurations remotely. The result is that RFID readers can be integrated into IT service management, just like any other computing hardware.

RFID data interpretation. From an application perspective, it is also desirable to provide a mechanism that interprets the captured RFID data in a given business context and that turns low-level RFID events into the corresponding business events. For example, the detection of a number of tags at a dock door can thus be automatically translated into a “shipment complete” event.

Sharing of RFID triggered business events. Many benefits commonly associated with RFID require data sharing across the supply chain [2]. In order to realize those benefits, an infrastructure that captures RFID triggered business events and makes them available to authorized parties is essential.

Lookup and Directory Service. During its lifetime, an RFID-tagged item usually passes the readers of many different parties. These parties typically store read events and other related data in their own respective information systems. In order to locate these various databases containing data on a given item, a lookup service is needed.

Tag identifier management. RFID allows for the unique identification of objects through the identifier stored in the memory on the RFID tag. Different numbering schemes exist for such an identifier and have to be supported by an RFID middleware. One prominent example is the electronic product code (EPC) [5] that comprises three parts, namely the product manufacturer ID, the product type ID, and the serial number. The EPC is available in different representations. This includes a representation that is suitable for storage in tag memory as well as representations as uniform resource locators. To convert between the different representations, a tag identifier translation mechanism is required [6].

Privacy. The intended deployment of RFID-based tracking solutions in today’s retail environments epitomizes for some the dangers of an Orwellian future: unnoticed by consumers, embedded RFID tags in our personal devices, clothes, and groceries can unknowingly be triggered to reply with their ID and other information, potentially allowing for a fine-grained yet unobtrusive surveillance mechanism that would pervade large parts of our lives [7], [8]. An RFID middleware should consider these consumer fears and the legal guidelines that apply for data collection. If RFID communication protocols support dedicated privacy enhancing features [9], [10], the RFID middleware will also need to support their use. For the “kill”-command specified in [10], this means, for example, that RFID middleware must provide the appropriate “kill”-passwords to the reader.

Other application requirements that relate to security and scalability are not discussed here in detail, since they are not unique to RFID middleware design.

III. Constraints Imposed by the Characteristics of RFID

Before describing how the application requirements listed in the previous section can be met, we will outline the constraints imposed by the characteristics of RFID. We believe that these constraints have a significant impact on the design of an RFID middleware and introduce aspects that are unique to the RFID domain. Any RFID middleware design that fails to include these will result in inefficient data capture and consequently low data quality.

There are a wide variety of different RFID systems that address the requirements of individual applications, e.g., with respect to range, transmission speed, susceptibility to environmental interference, and cost. Different passive RFID systems can be distinguished by the frequency band they operate in, the coding, the modulation, the medium access techniques used, and the supported command set. Since RFID design is generally driven by tradeoffs between different properties – e.g., read range, data transfer rates, identification speed, tag and reader form factor – there is no single RFID technology that proves superior in all possible application domains. For the purpose of this paper, the characteristics common to all passive RFID systems are most important, since they have a strong impact on RFID middleware design. These include:

Reliability issues. Due to the field nulls caused by, e.g., multipath fading (at UHF) or absorption by objects in the range of the reader, there is no guarantee that a tag will stay powered while in the assumed range of the reader. The result is a false negative read. Such false negative reads can also be caused by collisions on the air interface and by transmission errors [11].

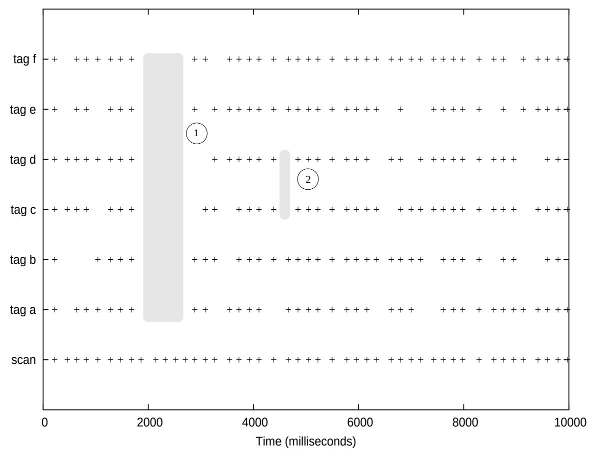


Fig. 4. Example of a series of tag reads showing regular reads by an RFID reader (HF) of six tags which are present in its read range [11]. The "missing" reads indicate that not all tags are detected on each scan (false negative reads). The figure also illustrates two different causes of false negative reads: interference problems (1) and collisions on the air interface (2).

False negative reads result in the fact that a tag will not be continuously detected on consecutive scans by a reader (cf. Fig. 4) although the tag remains in the assumed range of the reader. RFID middleware should address this characteristic of passive RFID technology by providing time aggregates, such as entry and exit events, which represent the most likely time of arrival and departure of a tagged object. Multipath effects can also lead to false positive reads. A false positive read occurs when a tag is detected although it is not within the range typically associated with an RFID reader antenna. This could, for example, mean that a tag passing through dock door 1 is "by mistake" detected by a reader at dock door 2, leading to a wrong location estimate. Ideally, appropriate data analysis performed on the reader or in the middleware could distinguish the weak tag reply by a tag embedded in a pallet from a reply by a tag located far away from the reader antenna.

Tag memory. The design of RFID middleware is also influenced by the memory structure on the tags. The memory on the microchip embedded in the tag usually contains a unique identifier. This can be either a random serial number or an identifier code that incorporates information about the tagged object, e.g., its manufacturer. Most microchips also feature small amounts of additional random access memory. Due to the increased power required to write to the EEPROM on the microchip, the maximum distance between reader and tag for a "write" operation is a fraction of that for a "read" operation. In [12], Karthaus et al. present a UHF RFID tag microchip with EEPROM that consumes 12.5 μW in read mode and 35 μW in write mode. RFID middleware should thus not only provide a method to write to a tag, but should also consider the fact that it might not be possible to write to a tag, even if it was successfully identified previously.

Heterogeneous reader landscape. The diverse computing and networking capabilities of readers are also an important characteristic of RFID technology. Low cost readers usually support only a single antenna and a serial RS232 interface. More sophisticated reader devices support several antennas,

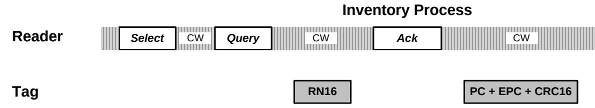


Fig. 5. "Select" command in the EPCglobal UHF Class 1 Generation 2 Protocol [10]. A particular tag population is selected before the inventory process is initiated with the "query" command.

a TCP host interface, and ample computing resources for on-device data processing. The result is that an RFID middleware design cannot assume that all RFID reader devices have the same capabilities. Many RFID deployments also feature handheld RFID readers, which might only have an intermittent network connection.

Reader collisions. RFID systems rely on the availability of unlicensed frequency bands. Radio transmissions in these unlicensed bands are governed by local radio regulations. In the UHF frequency band, which is particularly suitable for supply chain applications due to its superior read range, the European radio regulations which are based on the ETSI recommendation EN 302 208 permit the use of fifteen 200 kHz-wide channels between 865.0 MHz and 868.0 MHz by RFID readers [13]. However, there are only 10 sub-bands in which the maximum radiated power of 2 W ERP is available, and which thus allow for the largest read range. To avoid collisions with other reader devices and radio transmitters, ETSI EN 302 208 introduces the concept of "Listen Before Talk (LBT)". Prior to transmission, the RFID reader has to listen for the presence of another signal within the intended channel of transmission. The threshold level set for this Listen-Before-Talk scheme, -96 dBm in the worst case, implies that it is unlikely that two readers within the same facility will be able to operate in the same channel [14], [15]. Since large distribution centers might need to operate as many as 100 readers, it is evident that readers need to coordinate their activities somehow to avoid missing tags that pass by while the reader is not operating due to the limited number of channels available.[3]

Limited communication bandwidth. Another constraint is the bandwidth available per channel, which limits the data transmission rate between readers and tags. It restricts the number of tags that can typically be identified per second to the order of tens or hundreds. A tag labeling a shipment that arrives on a pallet carrying more than a thousand tagged items can thus easily be missed unless the identification of the "shipment" tag is prioritized. To facilitate the latter, some RFID protocols permit the selection of a certain group of tags based on data stored on the tag [10], [17]. In [10], there is a "select" command that selects a tag population based on a number of different criteria before the inventory process is initiated by the reader (cf. Fig. 5). The result

[3]There is a proposal to eliminate the LBT requirement in four of the ten channels [16], since the LBT scheme recommended in EN 302 208 is not taking the capabilities of the latest RFID air interface protocols into account, such as the dense reader mode in [10]. Eliminating LBT would alleviate the need for middleware to time division multiplex the readers in most applications. There is still a middleware service required that assigns the frequency channels to readers appropriately.

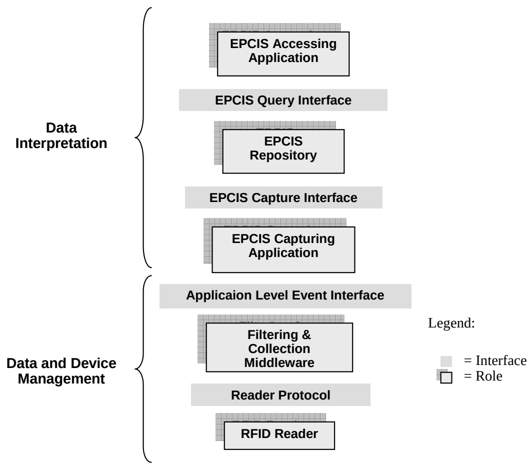


Fig. 6. EPC Network roles and interfaces [18]

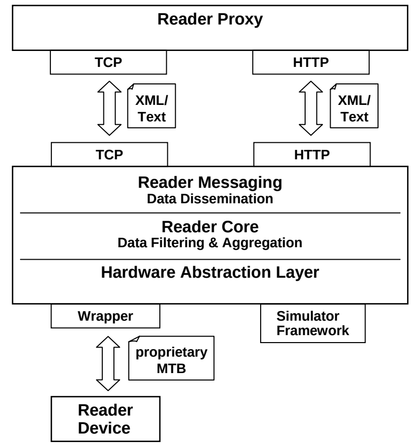


Fig. 7. Accada reader implementation supporting the EPCglobal RP and RM specifications

is that the limited communication bandwidth available is utilized efficiently, since only the tag population of interest is inventoried. The RFID middleware can utilize this feature of today's RFID communication protocols by propagating tag ID filters specified by any application to the reader devices.

IV. Accada – An Open Source RFID Platform

The goal of the Accada project is to develop an RFID platform that meets the application needs and constraints mentioned in the previous two sections. Our implementation is based on the standardized interfaces published by one of the predominant RFID standardization body – EPCglobal – but also features a number of extensions that address some of the challenges to RFID middleware design presented earlier. The Accada platform consists of three separate modules: the reader module, the filtering and collecting middleware module, and the EPC Information Services (EPCIS) module. These modules implement the corresponding roles in the EPC Network (cf. Fig. 6), which is named after the standardized tag identifier code, Electronic Product Code (EPC), which is stored on every tag. We begin our description with the reader module, which performs data dissemination, filtering, and aggregation at the reader level. We then continue with our filtering and collection middleware, which decouples readers and applications and provides additional aggregation and filtering functionality. The third module relates to the EPCIS part of the EPC Network, which deals with a framework to interpret the captured RFID data in an application context.

A. Reader module

The reader module of the Accada platform implements the EPCglobal Reader Protocol [19]. Our implementation features all mandatory and optional features defined in the Reader Protocol specification. This includes filters on tag ID and reader antenna, time aggregates, such as entry and exit events, and space aggregates, where multiple reader antennas can be logically grouped to a single source. There is also support for writing to tags and external triggers such as motion sensors. The data captured can be disseminated via a query-response and a publish-subscribe mechanism using different message transport bindings as shown in Fig. 7. Our reader implementation also features a Reader Proxy that implements the host side of the EPCglobal Reader Protocol. This facilitates application development because interaction with the resource-limited reader device is now as straightforward as a remote procedure call.

The Accada reader implementation can be used in three different modes. In the first mode, the reader implementation is deployed on a separate server as a surrogate and wraps a proprietary RFID reader protocol using the built-in hardware abstraction layer (cf. Fig. 7). Our implementation currently supports a variety of reader devices from different manufacturers.

To facilitate testing of RFID applications when there is no actual reader available, the Accada reader can also be used in a simulation mode. The built-in simulation framework that supports this mode includes a graphical user interface that allows the developer to drag and drop tags over different reader antennas (cf. Fig. 8). The simulation framework also provides a mechanism to schedule the detection of a tag at different times on different reader instances. This allows a developer to

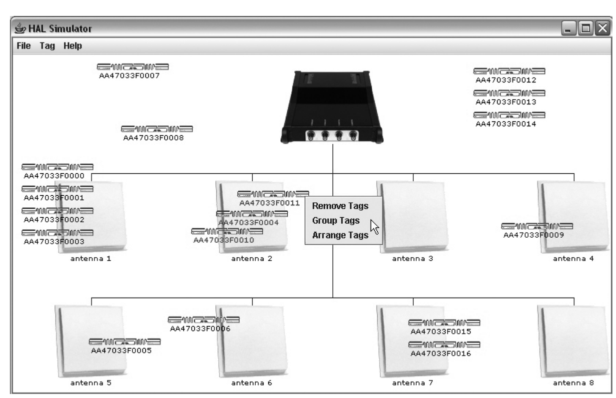


Fig. 8. Screenshot of the Accada reader simulator

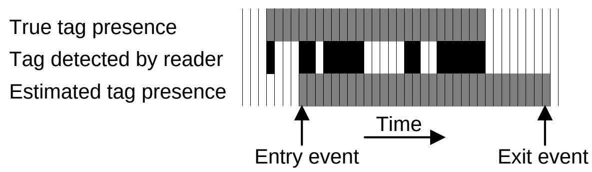


Fig. 9. Entry and exit events generated by an adaptive filter from RFID data.

simulate the movement of a tag population through the supply chain. In the third mode, the Accada reader implementation can also be deployed on an RFID reader itself to provide data dissemination, filtering, and aggregation capabilities.

The Accada reader module also features adaptive filters that automatically adjust the time constants of the filters that remove any noise resulting from false negative reads [11]. Without our adaptive filters, these time constants need to be specified manually, which is difficult in practice. A sample aggregate computation is shown in Fig. 9. In this figure, the time constant is set to eight time units. To eliminate false positive reads, tags need to be detected a few times before an entry event is triggered.

Our reader module uses a virtual tag memory service (VTMS) that facilitates writing to a tag by shielding the application from the characteristics of RFID tag memory: limited memory size, different memory organizations, reduced write range. Hosts simply provide key-value pairs that should be written to a set of tags. The reader module then checks with the VTMS for the appropriate tag memory block and page to write to. If the write succeeds, the reader module acknowledges this to the host and stores a backup copy of the data in the virtual representation of the tag in the VTMS. If the tag memory gets corrupted at a later stage or the host wants to access the memory of the tag while the tag is outside the range of any reader, the data can be made available via this virtual memory. If the write to the tag fails due to insufficient power, the key-value pair will be stored in the VTMS and flagged as “open”. The reader will retry the write command at a later point of time. If there is insufficient memory space, the host will receive an appropriate error message and the key-value will be stored in the virtual tag memory only. The host can also indicate that the virtual memory of a tag can only be accessed once the tag is in the read range of the particular reader.

As with all enterprise IT systems, monitoring and configuration management are important issues in large-scale RFID deployments. The reader module therefore also implements the EPCglobal Reader Management Protocol in order to provide fault and configuration management capabilities. While the EPCglobal Reader Management Protocol specifies both an SNMP/UDP and an XML/TCP message transport binding, the Accada reader module only supports the SNMP binding.

B. Filtering and Collection Middleware module

The Accada filtering and collection middleware represents a single interface to the potentially large number of readers that make up an RFID system deployment. This allows applications to define a standing query and subsequently subscribe to it. The subscription is then used to configure the corresponding reader devices using the EPCglobal Reader Protocol (cf. Fig. 10). This feedback to the event producing readers allows the readers to use the limited communication bandwidth efficiently. Such feedback can lead to an appropriate adaptation of the queries executed by the reader over the air interface, e.g. targeting a particular tag population at a higher sampling rate or switching off completely to make the bandwidth available to another reader. The filtering of the RFID data is then no longer carried out in the event router, but over the air interface (cf. Fig. 5). Once the readers capture relevant tag data, they notify the middleware, which combines the data arriving from different readers in a report that is sent to the subscribed applications according to a predetermined schedule. Since the middleware receives data from multiple readers, it can provide additional aggregation functionality. Redundant read events from different readers observing the same location can thus be omitted, reducing the amount of filtering and aggregation required in any application interpreting the captured RFID data. The interface between the filtering and collection middleware and a host application is based on the EPCglobal Application Level Events (ALE) Specification [20]. The ALE specification defines a SOAP message transport binding for the subscription communication channel and an XML and TCP/HTTP message transport binding for the notification channel.

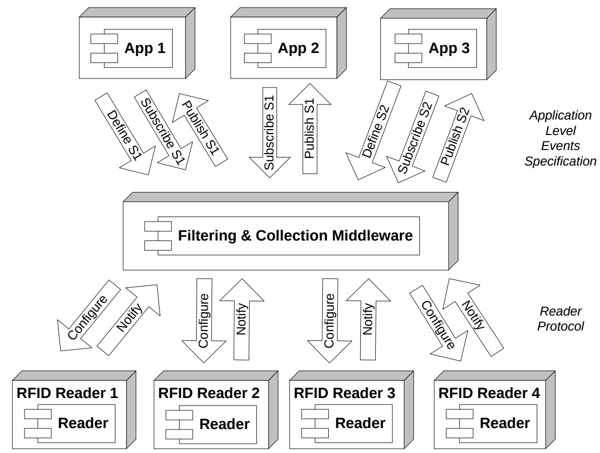


Fig. 10. Filtering and collection middleware

C. EPCIS module

The EPC Information Services (EPCIS) [21] of the EPC Network is the component that is responsible for receiving the application-agnostic RFID data from the filtering and collection middleware, translating them into the corresponding business events, and making those business events available. The EPC Information Services themselves consist of three parts (cf. Fig. 6): an EPCIS capturing application that interprets the captured RFID data, an EPCIS repository that provides persistence, and an EPCIS accessing application that retrieves the EPCIS events from the repository.

EPCIS Capturing Application. The EPCIS capturing application subscribes to the filtering and collection middleware from where it receives uninterpreted RFID read events. Its main task is to translate these uninterpreted events (e.g., detection of tag with ID 3455.3454656) into meaningful business events (e.g., arrival of a shipment of razor blades) that are known as *EPCIS events*. If necessary, it is also the responsibility of the capturing application to keep track of multiple filtering and collection events and transform them into a single EPCIS event. This situation arises, e.g., when a number of individual items are commissioned on a pallet. In order to generate a business event out of the plain RFID data obtained from the filtering and collection middleware, the capturing application may need to acquire additional information, such as interactive user input or data from business information systems. In general, implementations of the capturing application differ significantly from each other depending on the business process in which they collect RFID data and will thus have to be developed on a case-by-case basis. However, the Accada platform provides a generic capturing application for simple deployments where there is no need to acquire data from external sources (i.e., other information systems or interactive user input). The Accada generic capturing application can be configured to register an arbitrary number of subscriptions with one or several instances of the filtering and collection middleware. For each subscription, a simple event template can be specified that will be used to transform every incoming ALE event into a single EPCIS event. This event is then written into a freely configurable EPCIS repository (see below). Configuration of both subscriptions, templates, and target repository is done in an XML file, requiring no code to be written. It thus hides from developers the complexity of the filtering and collection middleware as well as the EPCIS event generation and communication with the repository.

EPCIS Repository. The repository provides persistence for EPCIS events. The Accada platform includes a module that adds two EPC Network-conformant interfaces to a relational database backend (currently MySQL), turning it into an EPCIS repository. New events can be stored in the repository through the *capture interface*, while historical events can be retrieved through the *query interface* (cf. Fig. 11). Current EPCIS standards do not define a query language, but rather use a few predefined queries that can be parameterized. Accada provides both synchronous and asynchronous (including scheduled) query processing capabilities. The Accada EPCIS repository provides a TCP/HTTP binding for the capture interface, a SOAP binding for the query interface (allowing for the immediate execution of queries and the registration of subscriptions), as well as an HTTP binding for the notification channel.

EPCIS Accessing Application. Applications that retrieve data from the repository act as EPCIS accessing applications. As such an application (e.g., a warehouse management system) is not a middleware component, it is not included in the Accada platform.

The Accada EPCIS module includes both a graphical capturing and accessing application in order to facilitate the development of applications using EPCIS. During the development

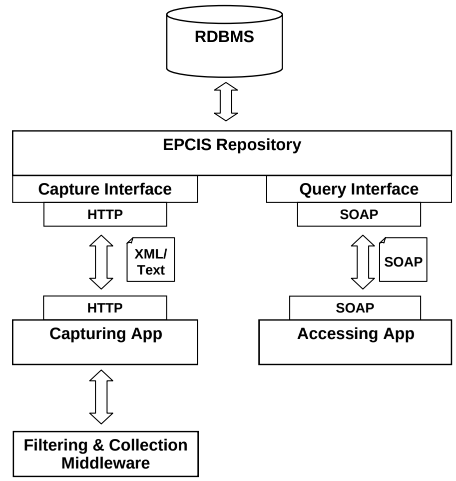


Fig. 11. Accada EPCIS module featuring an EPCIS repository as well as capture and accessing applications

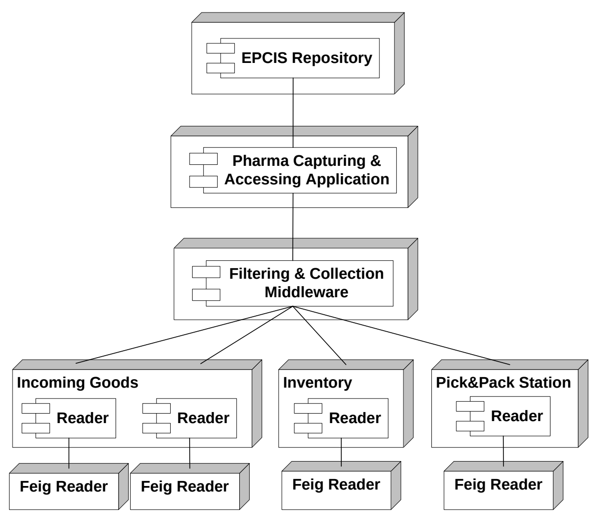


Fig. 12. Deployment diagram of pharma distribution center demonstrator

of an Accada-based application, these tools can be used in the testing and debugging phase to interactively store events in and read events from a repository.

V. Sample Applications

In order to illustrate the value and maturity of the Accada platform, we present two RFID applications that use Accada. The first example is a supply chain application that is based on the scenario described in Section II. The other application is a pervasive gaming application, where a deck of cards is equipped with RFID tags. Both have been in routine use in our research group.

A. Pharma Distribution Center Application

The pharma distribution center application comprises all three modules of the Accada platform and features four different reader devices (cf. Fig. 12). These reader devices

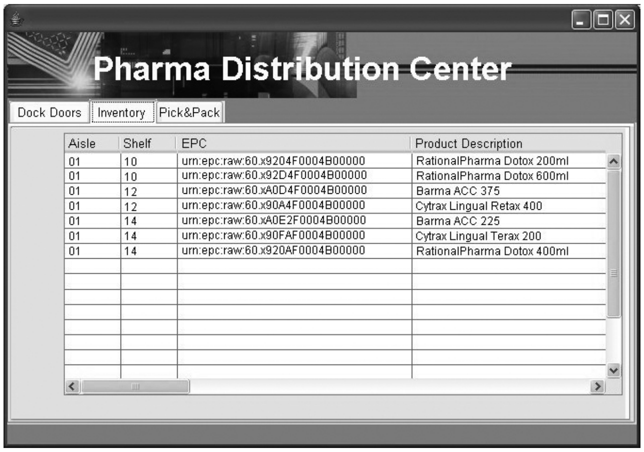


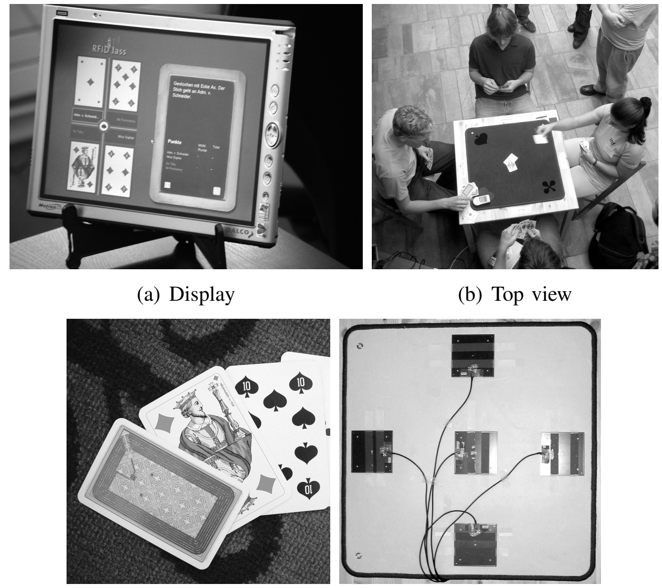
Fig. 13. Screenshot of the EPCIS accessing application

Fig. 14. Anti-counterfeiting application

are used to identify incoming goods, to monitor the inventory in the warehouse, and to identify the items present at the pick and pack station:

- *Incoming goods.* The data captured by two readers are aggregated by the filtering and collection middleware. The corresponding EPCIS capturing application subscribes to it in order to receive tag appearance events from the logical reader "incoming goods" and checks whether incoming shipments are complete. It then generates a corresponding EPCIS event and stores it in the EPCIS repository.
- *Inventory.* The reader is configured to scan the inventory every five minutes. Any tags that arrive or disappear are reported via the filtering and collection middleware to the EPCIS capturing application. An EPCIS accessing application makes the data available to the warehouse staff (cf. Fig. 13).
- *Pick and pack station.* The reader at the pick and pack station continuously monitors the desk of the staff and updates the list of tags currently present in near real-time. The information is displayed on a screen to support the staff.

In this context, we also developed a second application that demonstrates how the events collected in a pharma supply chain can be shared with third parties using the Accada EPCIS repository. Our second demonstrator is a simple anti-counterfeiting application that incorporates the idea of checking whether a product has a plausible track and trace history in order to determine if it is authentic or counterfeit [22]. The demonstrator consists of a Java ME MIDlet running on a Nokia 3220 mobile phone that is equipped with a Near Field Communication (NFC) module. When the user touches a medicine package to which we have attached an NFC-compatible RFID tag, the EPC stored on the tag is read and used to retrieve the corresponding EPCIS events from the repository. Depending on the EPC's history, the user is then informed if the product is likely to be a counterfeit or not (cf. Fig. 14).[4]



(a) Display (b) Top view

(c) RFID-equipped cards (d) Reverse side of cover

Fig. 15. Smart playing cards

B. Smart Playing Cards

Computer-augmented tabletop games have recently received considerable attention. In contrast to the most common form of pervasive gaming, where computer games are mapped onto real-world settings, augmented tabletop games build on old-fashioned (board) games that are enriched with unobtrusive information technology. Smart playing cards [23] represents a conventional card game that is augmented with information technology to automate score keeping and to advise novice players (cf. Fig. 15).

In this smart card game, we use a single short-range RFID reader (Feig ID ISC.MR100) together with a multiplexer (Feig ID ISC.ANT.MUX-A) that switches five different antennas (cf. Fig. 16). The antennas are integrated into a cover that is commonly used to facilitate the playing (cf. Fig. 15). We minimize the time it takes to detect card movements by dynamically eliminating antennas of no interest by configuring the Accada reader module appropriately. Initially, we only subscribe to receive entry events on the four "player" antennas because the central antenna is of no significance during card distribution. After the system has determined the owner of each card, the outer antennas are deselected and the application

[4] In our demonstrator, we consider the trivial case where each product has a fixed path through the supply chain and its authenticity can be established by simply comparing its track and trace data with the fixed path.

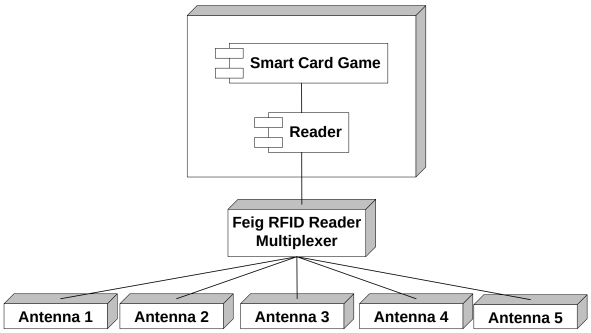


Fig. 16. Deployment diagram of smart playing cards application

subscribes to entry events on the central antenna only. Since the filtering and aggregation takes place within the reader module, there is no need to develop custom filtering software in the application.

Since EPCIS models supply chain processes and not gaming applications, the card game does not use the Accada EPCIS. There is also no need to implement the filtering and collection middleware based on the ALE interface, since we are dealing with a single reader instance only.

VI. Discussion

The Accada platform was developed to meet the application needs and constraints of passive RFID technology listed in Section II and Section III. Both of them – needs and constraints – make the development of middleware unique for the RFID domain. Tables I and II show that our Accada implementation addresses the majority of the application requirements and constraints of passive RFID technology. In the following subsections, we discuss each module, list its limitations, and present future work.

Application Requirement	Addressed by
Data dissemination	Publish-subscribe messaging system that decouples readers and applications in space and time.
Data aggregation	Aggregates in the time (appearance and disappearance) and space domain (across reader antennas and multiple readers) as well as a count aggregate.
Data filtering	Filters on tag ID, tag memory, reader antenna, and reader ID.
Writing to a tag	Virtual tag memory service that facilitates writing to a tag memory.
Trigger RFID readers by external sensors	Dedicated read triggers in the reader module.
Fault and configuration management	Implementation of the EPCglobal Reader Management Protocol.
Tag identity	Implementation of the EPCglobal Tag Data Translation Protocol.
RFID data interpretation	Generic capturing application that facilitates the interpretation of RFID data and the translation into meaningful business events.
Sharing of RFID-triggered business events	EPCIS module to store and retrieve business-level events.
Privacy	Subscription feedback mechanism that prevents RFID readers from collecting data without any application indicating an interest in them.

TABLE I
Application Requirements addressed in Accada

RFID Constraint	Addressed By
Reliability issues	Aggregates that reduce the volume of data generated to events that reflect the status change of a tag (appearance/disappearance). This is supported by our adaptive filters in the reader module that automatically adapt the time constants of the smoothing filters.
Tag memory	Virtual tag memory system that abstracts from the different tag memory organizations and provides redundancy.
Heterogeneous reader landscape	Surrogate concept where (embedded) computing devices support those RFID readers with limited computing resources.
Limited communication bandwidth	Filtering over the air interface enabled by the subscription feedback mechanism in the filtering & collection middleware.
Reader collisions	Subscription feedback mechanism to the event producing readers allows readers to detect when no application is interested in the data it captures.

TABLE II
Constraints of Passive RFID Technology addressed in Accada

A. Reader module

The usage of the Accada reader module is envisioned in three different modes: as a surrogate, where the software is deployed on a separate (embedded) computer, embedded on a reader itself, and as a simulation tool with a graphical user interface with no actual reader hardware connected. The reader module implements all mandatory and optional features specified by the EPCglobal Reader Protocol [19]. This includes data dissemination, filtering, aggregation, writing to tags, and external triggers (cf. Table I). Our reader module implementation also features a number of add-ons which are not part of the above specification. This includes the virtual tag memory service, the adaptive filters to compute the aggregates, and the surrogate mode, all of which help to address the constraints of passive RFID technology (cf. Table II). The virtual tag memory service facilitates writing to a tag by shielding the application from the particularities of RFID tag memory: limited memory size, different memory organizations, reduced write range. The adaptive filters help to deal with the false negative reads and to smooth the noisy RFID data. Due to our surrogate concept, where the Accada reader software is deployed on a separate (embedded) computer and connects to reader devices via our hardware abstraction layer, our implementation also addresses the heterogeneity of the reader landscape and in particular those reader devices with proprietary message transport bindings and limited resources. This surrogate mode supports only a limited number of different reader devices, which do not speak the EPCglobal Reader Protocol yet. We hope that other researchers will see the benefit of the existing open source Accada implementation and contribute drivers for the reader devices they use.

The fault and configuration management capabilities of our reader module are based on the EPCglobal Reader Management Protocol [18]. An Accada reader can thus be managed and configured like any other IT equipment via the Simple Network Monitoring Protocol (SNMP). The Accada reader implementation currently uses the standard edition of SUN's Java Virtual Machine rather than a microedition. The result is that the Accada reader implementation can only be embedded on those reader devices that feature significant computing resources. In the future, we should consider porting the software to be compliant with one of the Java profiles for limited devices.

The EPCglobal Reader Protocol Version 1.1 currently consists of a large number of optional features. This design choice seems to be a direct result of the heterogeneity of the reader landscape, where reader devices with significant computing resources were envisioned to provide the majority of the optional features and low-end reader devices would only support the mandatory features and possibly a small number of the optional features. However, this makes application development a significant challenge, since there is no service discovery available that could advertise the supported features. Future versions of the EPCglobal Reader Protocol could address this issue by providing different reader profiles with corresponding feature sets.

The EPCglobal Reader Protocol Version 1.1, which we implemented in Accada, abstracts from the underlying air interface protocols. This was an explicit design choice, since it shields the application developer from the RF communication details. The disadvantage of air interface abstraction is that advanced features particular to an air interface are not available. This reduces the RFID performance in challenging application environments, e.g. where many readers operate in close proximity or tags move fast. For users of the EPCglobal UHF Class 1 Generation 2 Protocol, this means, for example, that advanced features such as the dense reader mode are not exposed. There is also no way to externally control protocol timings and coding schemes to deal with interference.

While the EPCglobal Reader Protocol supports tag memory access, the commands can only be triggered by the application. It is not possible to schedule inventory, write, read, or kill operations. The result is that a tag might have left the range of the reader before the reader reported the tag identifiers detected and received the corresponding write command. The recently released EPCglobal Low Level Reader Protocol [24] addresses these shortcomings. LLRP does not specify any filtering and aggregation functionality on the RFID reader though. This means that all filtering and aggregation needs to be carried out in the middleware.

B. Filtering and Collection Middleware module

The Accada filtering and collection middleware supports data dissemination, filtering and aggregation. At first sight, this seems redundant to the capabilities provided at the reader level. However, it is essential to support filtering and aggregation at both levels. There might be readers that do not support the optional filter and aggregation functionality due to limited computing resources. Moreover, the middleware is also capable of aggregating information captured by a number of different readers. It would be helpful if the Reader Protocol and Application Level Events Specification of EPCglobal used the same nomenclature and specified common functionality in the same way.

While the decoupling of RFID event consumers and producers is desirable, the limited bandwidth available to RFID systems requires a feedback mechanism for readers to determine whether applications are interested in the RFID data they produce. Our filtering and collection middleware is thus not just a messaging service that receives events from the event producers and routes these to subscribed applications, but it also feeds the subscriptions registered to the appropriate readers. This feedback can then lead to an adaptation of the queries executed by the reader over the air interface, e.g. targeting a particular tag population at a higher sampling rate or switching off completely to make the bandwidth available to another reader. The filtering of the RFID data is then no longer carried out in the messaging service, but over the air interface (cf. Fig. 5). If such a feedback mechanism is missing and readers simply co-ordinate access to the radio channel independently of the application's needs, the quality of the captured data will suffer. A reader configured to read any tag might miss a fast-moving pallet tag – potentially the only tag an application is interested in. Likewise, a reader listening for tag replies and occupying a radio channel – though no application desires its data – will potentially cause a dock door reader that is unable to find a free channel to miss an outgoing shipment. Such a subscription feedback mechanism is also beneficial from a privacy perspective. It prevents RFID readers from collecting data and possibly logging those data without any specific application indicating an interest in the data.

Our filtering and collection middleware is currently not scalable to very large numbers of readers and notification volumes. There is no mechanism to farm out the event aggregation and routing to multiple servers in a tightly-coupled local area federation. This would require methods for handover of connections to facilitate load sharing. There are also no security mechanisms in place to authenticate readers and applications.

C. EPCIS module

While Accada supports collaborating parties in sharing their RFID-related data through the EPCIS module, it does not currently provide means to look up all EPCIS repositories containing data on a given EPC. As the current EPCglobal specifications do not address the issue of discovering repositories, collaborating parties will have to use some out-of-band mechanism to advertise their repositories. However, for the special use case of finding a manufacturer's repository, EPCglobal provides a global infrastructure service, named Object Naming Service (ONS) [25]. As this service is merely an extension of the existing Domain Name Service (DNS) infrastructure, and using it is as simple as looking up a domain name in the DNS, Accada does not provide any dedicated module for it. Future releases of Accada should, however, provide components that facilitate the advertisement and discovery of repositories without exposing confidential data.

Lastly, Accada only allows for the persistence of business events, while uninterpreted RFID data is discarded once it has been processed. While this is sufficient for many applications, the availability of historical raw reads would be especially helpful in the case of an exceptional situation that is not mirrored in a business process and does thus not generate a persistent event. For instance, many companies must make sure that their production process adheres to regulations imposed by law or internal standards. It might, e.g., not be allowed to keep one chemical substance in close physical proximity of another, highly reactive substance. In order to ensure compliance with regulations, it is common practice to conduct audits. In these cases, it might be useful for an auditor to have historical location data at hand. By analyzing samples from the location history, dangerous conditions could be detected, e.g., in a safety audit or environmental audit. A possible approach to such needs would be to extend Accada to also provide a persistence and query mechanism for historical data on the reader or filtering and collection level.

VII. Related Work

The need for an RFID infrastructure and the application requirements towards such an RFID infrastructure have been discussed in a number of publications [4], [26], [27]. Initially, the concept of a distributed networking infrastructure for RFID was proposed by the Auto-ID Center, an industry-sponsored research program to foster RFID adoption [3], which coined the term EPC Network.

Our work is closely related to a component in a now outdated architecture of the EPC Network, which was called Savant [28]. While the Savant software addressed some of the application requirements presented in Section II (e.g., it featured functionality for coping with the idiosyncrasies of different kinds of readers and for cleaning the data), there was only limited built-in functionality that specifically addresses data dissemination, filtering, and aggregation. In the Savant implementation, there was also no predefined subscription language. Instead, the Savant concept allowed applications to register event filters programmed in the Java programming language, which could operate on a combination of notifications. This approach increases the expressiveness of the subscription language at the expense of performance and scalability [29]. We believe that a predefined subscription language as provided by [19], [20] is expressive enough to support RFID data filtering and aggregation. However, the Savant architecture did not feature the strict separation of application-agnostic filtering and aggregation and application-specific data interpretation that we presented in this paper and which is reflected in the EPC Network architecture [18]. In the Savant system architecture, user-defined operators are thus not only used to filter the data, but also to interpret the data in an application context. It is important to note that in this kind of system architecture with custom application code executed on middleware appliances and RFID readers, the ALE interface and the reader protocol interface in its current form are not applicable.

There are a number of other commercial and non-commercial RFID middleware products available, among others [26], [30], [31], [32], [33]. All of them address the application requirements for device and data management and data interpretation listed in Section II. To our knowledge, none of them incorporates the standardized interfaces developed by the EPCglobal community to the extent shown in this paper.

The requirements which an overall RFID network infrastructure should meet and the corresponding system architectures have also been studied by [26], [34], [35], [36], [37]. Our work differs from the above because it focuses in particular on the constraints imposed by passive RFID and how these can be addressed in an appropriate middleware design.

In Section III, we mentioned the limited bandwidth available to RFID readers. To address this problem, we proposed a middleware design that selectively propagates the subscriptions to the RFID readers. The implication is that readers which would capture data that no application is interested in at any given point of time will refrain from operating and occupying the scarce bandwidth. However, there are a number of complementary approaches to address this reader collision problem, which was first documented in [38]. They include approaches that involve frequency planning and time-division multiplexing schemes [10], [39], [40], but also methods such as shielding and triggering RFID readers with motion sensors [41].

The filtering and collection middleware functionality of our Accada platform differs from the one commonly found in publish-subscribe systems. In traditional publish-subscribe systems, there is no feedback path from the messaging service to the producers. Due to the restricted bandwidth available to the RFID readers on the air interface and the potentially large amount of data they produce, the subscriptions of the applications are fed back to the event producing reader instances. The result is that reader instances only disseminate data to the messaging service which are of interest to the applications. The event router Elvin [42] uses a similar kind of feedback mechanism called quenching.

Different from most other publish/subscribe messaging systems such as [29], [42], the subscription process used in the Accada filtering and collection middleware consists of two steps. In an initial step, applications define a subscription, where they specify notification latency, aggregates, and filters. In a second step, applications can then simply subscribe to a previously defined subscription. In Accada, the filtering and aggregation functionality is realized with dedicated software that computes the aggregates. Alternatively, one could also consider using messaging services with subscription languages that support compound filters or patterns which are matched against multiple notifications based on both their attribute data and on the combination they form [43], [44].

VIII. Conclusion

In novel RFID application domains, such as supply chain management and logistics, there are many RFID readers distributed across factories, warehouses, and distribution centers. The data the readers capture need to be disseminated to a variety of applications. This introduces the need for an RFID infrastructure that hides proprietary reader device interfaces, provides configuration and system management of the reader devices, and filters and aggregates the captured RFID data.

In this paper, we discuss these application requirements in detail. We also contend that the characteristics of passive RFID technology introduce constraints that are unique to the development of middleware for the RFID domain.

These constraints include the occurrence of false negative and false positive reads, tag memory variations, the heterogeneous reader landscape, and the limited communication bandwidth available to RFID readers. To address these constraints and the application requirements for filtered and aggregated RFID data, we developed Accada, an open source RFID platform.

The paper shows that the current Accada implementation, which is based on a set of specifications developed by the EPCglobal community and a number of extensions, such as the surrogate concept and the virtual tag memory service, addresses the majority of the application requirements and limitations of passive RFID technology. The paper illustrates that a publish-subscribe approach that provides an additional feedback path to the event producing RFID readers is appropriate for RFID middleware design.

Our work also discusses limitations of the existing Accada implementation. This includes the limited supported for proprietary reader protocols and the abstraction from the underlying air interface protocols between RFID readers and tags. The latter means that advanced features for challenging environments available in some air interface protocols cannot be utilized. Future versions of the Accada filtering and collection middleware could provide persistence for RFID data captured. The scalability of the filtering and collection middleware to a large number of readers could be improved by farming out the event aggregation and routing to multiple servers in a local area federation.

REFERENCES

[1] Economist, "The Best Thing since the Bar-Code," Feb. 2003.

[2] K. Alexander, T. Gilliam, K. Gramling, M. Kindy, D. Moogimane, M. Schultz, and M. Woods, "Focus on the Supply Chain: Applying Auto-ID within the Distribution Center," Auto-ID Center, Tech. Rep. IBM-AUTOID-BC-002, 2002.

[3] S. Sarma, D. L. Brock, and K. Ashton, "The Networked Physical World – Proposals for Engineering The Next Generation of Computing, Commerce & Automatic Identification," MIT Auto-ID Center, Tech. Rep. MIT-AUTOID-WH-001, 2000.

[4] C. Floerkemeier and M. Lampe, "RFID middleware design – addressing application requirements and RFID constraints," in *Proceedings of SOC'2005 (Smart Objects Conference)*, Grenoble, France, Oct. 2005, pp. 219–224.

[5] EPCglobal, "EPC Tag Data Specification 1.1," Nov. 2003. [Online]. Available: www.epcglobalinc.org

[6] ——, "EPC Tag Data Translation (TDT) Standard," EPCglobal, Jan. 2006. [Online]. Available: www.epcglobalinc.org

[7] S. Garfinkel and B. Rosenberg, Eds., *RFID: Applications, Security, and Privacy*. Addison-Wesley, 2005.

[8] M. Langheinrich, "RFID and Privacy," in *Security, Privacy and Trust in Modern Data Management*, M. Petkovic and W. Jonker, Eds. Springer-Verlag, 2006, ch. RFID and Privacy.

[9] C. Floerkemeier, R. Schneider, and M. Langheinrich, "Scanning with a Purpose – Supporting the Fair Information Principles in RFID protocols," in *Ubiquitious Computing Systems. Revised Selected Papers from the 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004), November 8-9, 2004, Tokyo, Japan*, ser. Lecture Notes in Computer Science (LNCS), H. Murakami, H. Nakashima, H. Tokuda, and M. Yasumura, Eds., vol. 3598. Berlin, Germany: Springer-Verlag, June 2005, pp. 214–231.

[10] EPCglobal, "Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9," 2005. [Online]. Available: www.epcglobalinc.org

[11] J. Brusey, C. Floerkemeier, M. Harrison, and M. Fletcher, "Reasoning about Uncertainty in Location Identification with RFID," in *Workshop on Reasoning with Uncertainty in Robotics at IJCAI-2003, Acapulco, Mexico*, 2003.

[12] U. Karthaus and M. Fischer, "Fully Integrated Passive UHF RFID Transponder IC With 16.7-μW Minimum RF Input Power," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 10, pp. 1602–1608, Oct. 2003.

[13] ERM TG34, "Electromagnetic compatibility and Radio spectrum Matters (ERM);Radio Frequency Identification Equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W," European Telecommunications Standards Institute (ETSI), Tech. Rep. EN 302 208-1 V1.1.1, 2004. [Online]. Available: www.etsi.org

[14] G. Falcke, *The New RFID Standard in Europe*. [Online]. Available: www.rfidjournal.com

[15] K. S. Leong, M. L. Ng, and P. H. Cole, "The Reader Collision Problem in RFID Systems," in *Proceedings of IEEE 2005 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications (MAPE 2005)*, Beijing, China, 2005.

[16] ERM RM, "Electromagnetic compatibility and Radio spectrum Matters (ERM); Technical characteristics of RFID in the UHF Band; System Reference Document for Radio Frequency Identification (RFID) equipment; Part 1: RFID equipment operating in the range from 865 MHz to 868 MHz," European Telecommunications Standards Institute (ETSI), Tech. Rep. ETSI TR 102 649-1 V1.1.1, 2007. [Online]. Available: www.etsi.org

[17] International Organization for Standardization, "Information technology – Radio frequency identification for item management – Part 6: Parameters for air interface communications at 860 MHz to 960 MHz ," 2004.

[18] Architecture Review Committee, "The EPCglobal Architecture Framework," EPCglobal, July 2005. [Online]. Available: www.epcglobalinc.org

[19] EPCglobal, "Reader Protocol Standard, Version 1.1," June 2006. [Online]. Available: www.epcglobalinc.org

[20] ——, "The Application Level Events (ALE) Specification, Version 1.0," Sept. 2005. [Online]. Available: www.epcglobalinc.org

[21] ——, "EPC Information Services (EPCIS) Version 1.0, Specification," EPCglobal, Apr. 2007. [Online]. Available: www.epcglobalinc.org

[22] T. Staake, F. Thiesse, and E. Fleisch, "Extending the EPC network: the potential of RFID in anti-counterfeiting," in *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. Santa Fe, NM, USA: ACM Press, Mar. 2005, pp. 1607–1612.

[23] C. Floerkemeier and F. Mattern, "Smart Playing Cards – Enhancing the Gaming Experience with RFID," in *Proceedings of the third Pergames Workshop at Pervasive 2006*, T. Strang, V. Cahill, and A. Quigley, Eds., Dublin, Ireland, May 2006, pp. 79–88.

[24] EPCglobal, "Low Level Reader Protocol (LLRP), Version 1.0.1," EPCglobal, Aug. 2007. [Online]. Available: www.epcglobalinc.org

[25] ——, "Object Naming Service (ONS) Specification Version 1.0," Oct. 2005. [Online]. Available: www.epcglobalinc.org/standards_technology/EPCglobal_Object_Naming_Service_ONS_v112-2005.pdf

[26] C. Bornhövd, T. Lin, S. Haller, and J. Schaper, "Integrating Automatic Data Acquisition with Business Processes - Experiences with SAP's Auto-ID Infrastructure," in *Proceedings of the 30st international conference on very large data bases (VLDB)*. Toronto, Canada: VLDB Endowment, 2004, pp. 1182–1188.

[27] S. Sarma, "Integrating RFID," *ACM Queue*, vol. 2, no. 7, pp. 50–57, 2004.

[28] Oat Systems and MIT Auto-ID Center, "The Savant Version 0.1," Auto-ID Center, Tech. Rep. MIT-AUTOID-TM-003, 2002.

[29] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Achieving scalability and expressiveness in an internet-scale event notification service," in *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, Portland, Oregon, July 2000, pp. 219–227.

[30] S. Microsystems, "Java System RFID Software 3.0 Developer Guide," www.sun.com, Feb. 2006.

[31] OATsystems, "OAT C4 Architecture," www.oatsystems.com, 2006.

[32] B. S. Prabhu, X. Su, H. Ramamurthy, C.-C. Chu, and R. Gadh, "WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications," in *Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions*, R. Shorey and C. M. Choon, Eds. Wiley, 2005.

[33] Reva Systems, "Tag Acquistion Processor," 2007. [Online]. Available: www.revasystems.com

[34] S. Chawathe, V. Krishnamurthyy, S. Ramachandrany, and S. Sarma, "Managing RFID Data," in *Proceedings of the 30st international conference on very large data bases (VLDB)*. Toronto, Canada: VLDB Endowment, 2004, pp. 1189–1195.

[35] K. Römer, T. Schoch, F. Mattern, and T. Dübendorfer, "Smart Identification Frameworks for Ubiquitous Computing Applications," *Wireless Networks*, vol. 10, no. 6, pp. 689–700, Dec. 2004.

[36] F. Thiesse, "Architektur und Integration von RFID-Systemen," in *Das Internet der Dinge – Ubiquitous Computing und RFID in der Praxis*, E. Fleisch and F. Mattern, Eds. Springer-Verlag, 2005, pp. 101–118.

[37] F. Wang and P. Liu, "Temporal management of RFID data," in *Proceedings of the 31st international conference on very large data bases (VLDB)*. VLDB Endowment, 2005, pp. 1128–1139.

[38] D. Engels, "The Reader Collision Problem," Auto-ID Center, Tech. Rep. MIT-AUTOID-WH-007, 2001.

[39] J. Ho, D. W. Engels, and S. E. Sarma, "HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem," in *Proceedings of the RFID and the Extended Network Workshop at SAINT 2006*, Phoenix, AZ, USA, 2006, pp. 88–91.

[40] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: an anticollision algorithm for the reader collision problem," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, Anchorage, Alaska, USA, 2002, pp. 1206–1210.

[41] K. S. Leong, M. L. Ng, A. Grasso, and P. H. Cole, "Synchronization of RFID Readers for Dense RFID Reader Environments," in *Proceedings of the RFID and the Extended Network Workshop at SAINT 2006*, Phoenix, AZ, USA, Jan. 2006, pp. 48–51.

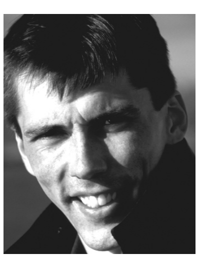
[42] B. Segall, D. Arnold, J. Boot, M. Henderson, and T. Phelps, "Content Based Routing with Elvin4," in *Proceedings AUUG2K*, Canberra, Australia, June 2000.

[43] M. Mansouri-Samani and M. Sloman, "GEM: a generalized event monitoring language for distributed systems," *Distributed Systems Engineering*, vol. 4, no. 2, pp. 96–108, 1997.

[44] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel, and M. Spiteri, "Generic Support for Distributed Applications," *IEEE Computer*, vol. 68–76, no. 3, p. 33, Mar. 2000.

Christian Floerkemeier received his Bachelor and M.Eng. degree in Electrical and Information Science with distinction from Cambridge University in the UK in 1999 and his Ph.D. from ETH Zurich, Switzerland, in Computer Science in 2006. He is currently a research scientist at the Massachusetts Institute of Technology (MIT), Cambridge. Before joining the Auto-ID Lab at MIT, he was Associate Director of the Swiss Auto-ID Lab at ETH Zurich. From 1999 until 2001, he worked as head of software development for Ubiworks, an Amsterdam-based software company. His research interests include radiofrequency identification systems and pervasive computing.

Christof Roduner studied computer science and business administration at the University of Zurich, Switzerland and the University of Uppsala, Sweden. He holds an M.Sc. degree from the University of Zurich. Before joining ETH Zurich's Institute for Pervasive Computing, he was responsible for the Internet business unit at a Zurich-based IT services company. He is a member of the Auto-ID Lab Switzerland. His research interests include infrastructures for smart objects as well as human-computer interaction.



Matthias Lampe holds an M.Sc. degree in computer science from Portland State University in Portland, Oregon and is currently a research assistant at the Institute for Pervasive Computing at ETH Zurich. As part of the industry sponsored research program M-Lab, he was involved in the design and implementation of various RFID based applications. He is also member of the Auto-ID Lab Switzerland. His main research interests are concepts and design issues for RFID middleware and application level system support for RFID applications.