RFID middleware design – addressing both application needs and RFID constraints

Christian Floerkemeier and Matthias Lampe
Institute for Pervasive Computing
ETH Zurich, Switzerland
floerkem|lampe@inf.ethz.ch

Abstract: Radio Frequency Identification (RFID) technology has a lot of potential to bridge the gap between the virtual world of enterprise IT systems and the real world of products and logistical units. In this paper we analyse the requirements RFID middleware solutions have to meet in order to manage large deployments of RFID readers and the amount of RFID data these readers capture. We also present the RFIDStack – an RFID middleware platform that does not only focus on the application needs, but also considers the constraints imposed by passive RFID technology.

1 Introduction

Radio Frequency Identification (RFID) systems have become common in places where access control and tracking of physical objects is required. More recently, RFID systems have begun to find greater use in industrial automation and in supply chain management. The proliferation of readers and tags in these domains is largely driven by the need to bridge the economically expensive gap between the virtual world of IT systems and the real world of products and logistical units [FD03]. It was recognized early on that the successful widespread adoption of RFID as an economical replacement of bar-codes depended not only on low cost tags, but also on a middleware to deal with the large amounts of captured RFID data [SBA00].

In this paper we analyse the requirements such RFID middleware solutions should meet in order to manage large deployments of readers and the amount of data these readers capture. We argue that the characteristics of passive RFID technology introduce constraints that are unique to the development of middleware for the RFID domain. We also present the RFIDStack, a middleware platform that does not only focus on the application needs, but also considers the limitations of passive RFID technology in order to deliver high-quality data.

The paper is organized as follows: In Section 2, we list related work. Section 3 discusses the application requirements RFID middleware should meet. Section 4 provides a brief overview of RFID technology and outlines the constraints imposed by the characteristics of RFID. In Section 5, we briefly introduce the RFIDStack – an RFID middleware solution that addresses these requirements and constraints.

2 Related Work

The concept of a distributed network infrastructure to deal with the captured RFID data and to turn them into the corresponding business events was initially proposed by the Auto-ID Center, an industry-sponsored research program to foster RFID adoption [SBA00]. The requirements for such an RFID network infrastructure have recently also been the focus of a number of studies [BLHS04, CKRS04, Thi05]. The work in this paper focusses on the architectural component that is responsible for managing the readers and processing the low-level data. Our work is thus closely related to the idea of the Savant in [CKRS04] and that of a device controller in [BLHS04]. However, our work differs from the above because it explicitly integrates the constraints imposed by passive RFID.

3 Application Requirements

RFID deployment is driven by the need to close the gap between the real world and the virtual world. This means that large numbers of readers will be monitoring object movements throughout the supply chain. They will be feeding the captured data not to a single application, but to a variety of enterprise IT systems and local computing applications, e.g., point-of-sale systems [BLHS04].

To illustrate the data requirements these applications have, we will consider a typical supply chain scenario – a distribution center of a pharmaceutical company:

Goods are picked from the warehouse and packed at the corresponding pick&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 pick&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 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. The readers monitoring the warehouse shelves scan these at regular intervals and trigger the legacy warehouse management systems to update the inventory counts of the corresponding product categories.

This scenario represents application needs that are common for many business processes supported by RFID. It illustrates that different applications are interested in a different subset of the captured data. Common to all applications is however the desire to receive filtered and aggregated RFID events rather than raw streams of RFID data. Other application requirements that relate to security, scalability, and performance are out of the scope of this paper, but are discussed in [BLHS04, CKRS04, Thi05].

4 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 quality data.

While all passive RFID systems are made up of readers and tags, a wide variety of different RFID systems exist that address the requirements of individual application scenarios. They usually differ with respect to frequency of operation, coding and modulation techniques, timing, anti-collision algorithm, and supported command set [Fin00].

For the purpose of this paper, the important characteristics common to all passive RFID systems are the limited bandwidth available for reader-to-tag communication, as well as the problem of false negative reads resulting from the battery-less nature of the tags and the heterogeneous reader and tag landscape. The limited bandwidth implies that, if certain tags or readers are at a contain point in time of no interest to any application, they should ideally refrain from using up the scarce bandwidth. For readers, this means that they should switch themselves temporarily off and make the channel bandwidth available to other readers in the vicinity, if the locations they monitor are of no interest to any application. Similarly, tags attached to objects that need not be identified at a certain location should not participate in the singulation procedure initiated by the reader. To facilitate the latter, some RFID protocols permit the selection of a certain group of tags based on data stored on the tag, e.g. manufacturer code or product type [FSL04].

False negative reads are another characteristic of RFID. Due to field nulls, e.g., resulting from multipath fading or absorption by objects in the range of the reader, there is no guarantee that a tag stays powered while in the assumed range of the reader. Similarly, false negative reads can also be caused by collisions on the air interface or transmission errors [BFHF03]. Due to such false negative reads, tags might not be continuously detected on consecutive scans by a reader.

The design of RFID middleware is also impacted by the heterogeneous reader and tag landscape. This includes the memory structure on the tags and the different computing and networking capabilities of RFID readers. The microchip embedded in the tag can either contain a single random identifier or a structured identifier code that encodes information about the tagged object. Most microchips also feature small amounts of additional random access memory that can be selectively read out and written to. This additional memory could also contain data captured by on-board sensors. The computing resources and network interfaces of RFID readers vary. Low cost readers usually support only a single antenna and a serial RS232 interface. More sophisticated devices support several antennas, a TCP host interface, and ample computing resources for on-device data processing.

5 RFIDStack

This section presents an overview of an RFID middleware platform – the RFIDStack – that addresses the requirements and constraints described in the two previous sections. It is currently in use in various research groups to facilitate RFID application development. To address the characteristics of RFID – in particular the limited bandwidth, the problem of false negative reads, and the varying reader and tag resources – the RFIDStack features:

- a feedback mechanism by which readers are informed of the subscriptions currently registered. Such feedback can then lead to an appropriate adaptation of the queries exercised by a reader over the air interface, e.g., by targeting a particular tag population at a higher sampling rate or by switching off completely to make the bandwidth available to another reader. A simple data warehousing approach, where readers simply transmit all the captured data to a local database, in which the data are filtered and aggregated depending on the application queries, leads to reduced sampling frequencies. A reader configured to read any tag might miss a fast-moving pallet tag – potentially the only tag an application is interested in.

- smart filters and aggregates that address the problem of false negative reads and the application needs for filtered and aggregated data. Applications using the RFIDStack can specify the readers from which they would like to receive captured data. They are also able to define the tag population of interest. In addition to filtering functionality, special aggregates are supported that reduce the amount of application-level data processing required. These include entry and exit events that reduce multiple reads to the first appearance in and the disappearance of a tag from the read range. The RFIDStack also features a count aggregate type that abstracts from the serial numbers and provides the application with the total number of unique items present. A passage event that indicates the direction, in which a tagged object is moved, as a tag moves from one reader to another, is also provided. Whenever an application does not want distinguish between two readers, a virtual reader aggregate type allows it to virtually join their read range.

- a modular design that facilitates the filtering and aggregation functionality, either directly on a reader or on a surrogate with ample processing resources. This is additionally supported by a standardized query language.

- a tag memory abstraction that hides the complexity of different memory structures from the application.

The RFIDStack currently supports seven different types of readers from several different manufacturers. It is based on Java technology and uses the Elvin content-based messaging system [SAB+00].

6 Conclusion

This paper analyses the requirements RFID middleware solutions should meet in order to manage large deployments of readers and the amount of data these readers capture. We argue that the characteristics of passive RFID technology introduce constraints that are unique to the development of middleware for the RFID domain. We also present the RFIDStack, a middleware platform that addresses these limitations of passive RFID technology. It feeds application subscriptions to the readers to allow them to optimize bandwidth usage, provides smart filters and aggregates to reduce the data streams, and is able to manage a heterogeneous reader and tag landscape.

References

[BFHF03] 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.

[BLHS04] 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 30th VLDB Conference, Toronto, Canada*, pages 1182–1188. Morgan Kaufmann, 2004.

[CKRS04] S. Chawathe, V. Krishnamurthyy, S. Ramachandrany, and S. Sarma. Managing RFID Data. In *Proceedings of 30th VLDB Conference, Toronto, Canada*, pages 1189–1195. Morgan Kaufmann, 2004.

[FD03] E. Fleisch and M. Dierkes. Ubiquitous Computing: Why Auto-ID is the Logical Next Step in Enterprise Automation. Technical Report STG-AUTOID-WH-004, Auto-ID Center, 2003.

[Fin00] Klaus Finkenzeller. *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Sons, 2000.

[FSL04] Christian Floerkemeier, Roland Schneider, and Marc Langheinrich. Scanning with a Purpose – Supporting the Fair Information Principles in RFID protocols. In *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, Tokyo, Japan, November 2004.

[SAB+00] Bill Segall, David Arnold, Julian Boot, Michael Henderson, and Ted Phelps. Content Based Routing with Elvin4. In *Proceedings AUUG2K*, Canberra, Australia, June 2000.

[SBA00] Sanjay Sarma, David L. Brock, and Kevin Ashton. The Networked Physical World – Proposals for Engineering The Next Generation of Computing, Commerce & Automatic Identification. Technical Report MIT-AUTOID-WH-001, MIT Auto-ID Center, 2000.

[Thi05] Frederic Thiesse. Architektur und Integration von RFID-Systemen. In Elgar Fleisch and Friedemann Mattern, editors, *Das Internet der Dinge – Ubiquitous Computing und RFID in der Praxis*. Springer-Verlag, 2005.