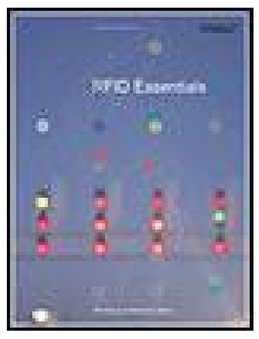
downloaded from: lib.ommolketab.ir



RFID Essentials

By Himanshu Bhatt, Bill Glover

.....

Publisher: O'Reilly
Pub Date: January 2006
Print ISBN-10: 0-596-00944-5
Print ISBN-13: 978-0-59-600944-1
Pages: 276

Table of Contents | Index

Overview

Radio Frequency Identification (RFID) is rapidly changing the way businesses track inventory and assets. From Wal-Mart and Tesco to the U.S. Department of Defense, early efforts are already showing benefits, but software, integration, and data processing for RFID still present a challenge. If you are a developer or an architect charged with developing an RFID system, this book is for you. Drawing on extensive experience, Bill Glover and Himanshu Bhatt provide you with essential information on this emerging technology.

With the knowledge you gain in these pages, you will possess the information and understanding you need to start designing, building, or integrating with RFID systems.

In *RFID Essentials* you will find information on:

- ---- Tags and tag protocols, including the Electronic Product Code (EPC)
- ---- Readers and reader protocols
- ---- RFID middleware
- ---- Security and privacy
- ---- Managing RFID devices
- ---- RFID's impact on your architecture

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Bill Glover has been writing software since 1981 and has worked as a programmer, lead developer, or architect on systems of all sizes, from small, automated systems controlling dams and feedmills up to a complete redesign and reimplementation of one of the world's busiest travel web sites. Bill first worked with RFID in 1995, tracking individual cattle using ear tags. He is currently a Senior Java Architect with Sun Microsystems, Inc., and works with Sun's RFID consulting practice and the RFID Test Center.

Himanshu Bhatt heads the U.S. RFID Practice and Software Technology Lab for Sun Microsystems, Inc. Prior to assuming this role, Himanshu was responsible for business development and consulting in emerging areas of technology. Himanshu has over 16 years of experience in the architecture and development of distributed, multitier systems using a host of technologies for Fortune 1000 companies. Himanshu has spoken at industry conferences such as JavaOne and the LoneStar Symposium and has published articles on Java/J2EE technologies.

"The Information Age is over. We're entering an era where network connectivity is almost ubiquitous - it's participate or perish."

--Jonathan Schwartz, President and COO, Sun Microsystems, Inc.

"Unique competitive advantage erupts from enterprises that couple the RFID technologies laid out in RFID Essentials with modern business integration using service-oriented architectures. This is the book to read in order to understand this new landscape."

--Mark Bauhaus, Senior Vice President, Sun Microsystems, Inc.

"This is a must read for RFID Software and Solution architects and is highly recommended for anyone needing to gain more insight into the myriad of components, standards and technologies that make up an RFID solutions environment."

--Bryan Tracey, Chief Architect, GlobeRanger Corporation

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

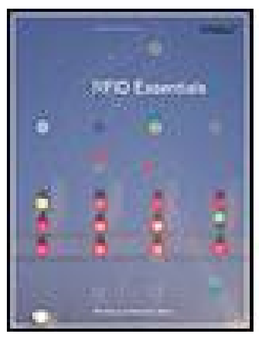
"The authors have done a commendable job of covering a lot of ground in the RFID space, including the infrastructure needed to share the volumes of data RFID will likely generate."

--Graham Gillen, Senior Product Manager, VeriSign

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

PREV



RFID Essentials

By Himanshu Bhatt, Bill Glover

.....

Publisher: O'Reilly
Pub Date: January 2006
Print ISBN-10: 0-596-00944-5
Print ISBN-13: 978-0-59-600944-1
Pages: 276

Table of Contents | Index

- Copyright
- Praise for RFID Essentials
- Preface
  - Who This Book Is For
  - Structure of This Book
  - Conventions Used in This Book
  - Comments and Questions
  - Safari® Enabled
  - Acknowledgments
- Chapter One. An Introduction to RFID
  - Section 1.1. The Case for RFID
  - Section 1.2. The Eras of RFID
  - Section 1.3. Application Types
  - Section 1.4. Summary
- Chapter Two. RFID Architecture
  - Section 2.1. A Confluence of Technologies
  - Section 2.2. Key Functionalities
  - Section 2.3. RFID System Components
  - Section 2.4. Systemic Quality Considerations
  - Section 2.5. Architecture Guidelines
  - Section 2.6. System Management
  - Section 2.7. Summary
- Chapter Three. Tags
  - Section 3.1. Basic Tag Capabilities
  - Section 3.2. Physical Characteristics
  - Section 3.3. Power Source
  - Section 3.4. Air Interface
  - Section 3.5. Information Storage and Processing Capacity
  - Section 3.6. Standards
  - Section 3.7. Summary
- Chapter Four. Tag Protocols
  - Section 4.1. Protocol Terms and Concepts

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

- Section 4.2. How Tags Store Data
- Section 4.3. Singulation and Anti-Collision Procedures
- Section 4.4. Tag Features for Security and Privacy
- Section 4.5. Learn to Troubleshoot Tag Communications
- Section 4.6. Summary

Chapter Five. Readers and Printers

- Section 5.1. Physical Components of an RFID Reader
- Section 5.2. Logical Components of an RFID Reader
- Section 5.3. Parts of an RFID Printer and Applicator
- Section 5.4. Types of Readers
- Section 5.5. Layout for Readers and Antennas
- Section 5.6. Configuring Readers
- Section 5.7. Summary

Chapter Six. Reader Protocols

- Section 6.1. Parts of a Reader Protocol
- Section 6.2. Vendor Protocols
- Section 6.3. EPCglobal Protocol Overview
- Section 6.4. Simple Lightweight RFID Reader Protocol
- Section 6.5. Future Protocols
- Section 6.6. Summary

Chapter Seven. RFID Middleware

- Section 7.1. Motivations
- Section 7.2. Logical Architecture
- Section 7.3. Application Level Events Specification
- Section 7.4. Commercial RFID Middleware
- Section 7.5. Summary

Chapter Eight. RFID Information Service

- Section 8.1. RFID Data
- Section 8.2. The EPCglobal Network
- Section 8.3. The Object Naming Service
- Section 8.4. The EPC Information Services
- Section 8.5. Summary

Chapter Nine. Manageability

- Section 9.1. Edge Deployment Options
- Section 9.2. Capabilities Needed for Edge Management
- Section 9.3. Standards and Technologies
- Section 9.4. Summary

Chapter Ten. Privacy and Security

- Section 10.1. Privacy and Security Issues
- Section 10.2. RFID Privacy
- Section 10.3. RFID Security
- Section 10.4. Summary

Chapter Eleven. The Future

- Section 11.1. Standards
- Section 11.2. Technology
- Section 11.3. Business

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

- Section 11.4. Summary
- Appendix A. EPC Identity Encodings
  - Section A.1. GS1 SGTIN Encoding
  - Section A.2. GS1 SSCC Encoding
  - Section A.3. GS1 GLN/SGLN Encoding
  - Section A.4. GS1 GRAI Encoding
  - Section A.5. GS1 GIAI Encoding
- Appendix B. References
  - Section B.1. Organizations, Standards, and Specifications
  - Section B.2. Important Papers on RFID
  - Section B.3. Related Books
  - Section B.4. Periodicals
  - Section B.5. History
  - Section B.6. Free Tools
- Glossary
- About the Authors
- Colophon
- Index

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

RFID Essentials

by Bill Glover and Himanshu Bhatt

Copyright © 2006 O'Reilly Media, Inc. All rights reserved. Printed in the United States of America.

Published by O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (safari.oreilly.com). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Editor:	Mary O'Brien
Production Editor:	Colleen Gorman
Copyeditor:	Rachel Wheeler
Proofreader:	Lydia Onofrei
Indexer:	Angela Howard
Cover Designer:	MendeDesign
Interior Designer:	Marcia Friedman
Illustrators:	Robert Romano, Jessamyn Read, and Lesley Borash

Printing History:

January 2006:	First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The *Theory in Practice* series designations, *RFID Essentials*, and related trade dress are trademarks of O'Reilly Media, Inc.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. O'Reilly Media, Inc. is independent of Sun Microsystems.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 0-596-00944-5

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

[M]

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Praise for RFID Essentials

"RFID is one of those rare 'change the world' technologies that will force a rethinking of many value-chain strategies. In that context, Bill and Himanshu's comprehensive expert-in-a-book creation should be on the must-read list for strategists and IT professionals who see RFID in their future. Unique competitive advantage erupts from enterprises that couple the RFID technologies laid out in *RFID Essentials* with modern business integration using service-oriented architectures. This is the book to read in order to understand this new landscape."

MARK BAUHAUS, SVP: BUSINESS INTEGRATION,IDENTITY, AND APPLICATION PLATFORM SOFTWARE

"RFID, or Radio Frequency Identification, got legs with mandates from giants such as Wal-Mart and the U.S. Department of Defense; however, the use is going beyond the supply chain. Companies are seeing the benefits of RFID in other areas such as asset tracking and drug authentication. No matter what the need, the first step in any engagement is understanding the technology and how it can be used to gain business benefits. This book gives a good foundation on RFID fundamentals and is a must-read if you are considering adopting RFID for your business or applications."

JULIE SARBACKER,DIRECTOR,SUN MICROSYSTEMS RFID BUSINESS UNIT

"The authors have done a commendable job of covering a lot of ground in the RFID space, including the infrastructure needed to share the volumes of data RFID will likely generate. While the short-term issues seem focused on read rates and hardware prices, in the end we may see yet another transformational use of the Internet in exchanging data about serialized assets and products."

GRAHAM GILLEN,SENIOR PRODUCT MANAGER,VERISIGN

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Preface

Like so many others, this book was written because we couldn't find one like it. We needed something to hand to all of those people who have come to us asking for "a good book to read on RFID." When we looked for candidates we found some great books, but most were aimed at electrica engineers or top-level managers, with very little for those of us who are in between. This book is for developers, system and software architects, and project managers, as well as students and professionals in all of the industries impacted by Radio Frequency Identification (RFID) who want to understand how this technology works. As the title suggests, this book is about RFID in general and not just the most recent developments; however, because so much is going on in the area of RFID for the supply chain and especially the Electronic Product Code (EPC), we have devoted considerable space to these topics. Regardless of the type of RFID work you may be doing, we think you will find something useful here.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Who This Book Is For

This book is for developers who need to get that first RFID prototype out the door; systems architects who need to understand the major elements in an RFID system; and project managers who need to divide work, set goals, and understand vendor proposals. Students and instructors should find enough detail here to use this book as at least a supplementary text for a study of RFID. Even those with considerable experience in RFID should find this book a useful update on the latest developments, with enough of the fundamentals to serve as a reference. This book is probably not fo anyone who wants either a cursory overview of the technology or a deep discussion of supply chain management, manufacturing, access control, or conspiracy theories. These are all interesting topics in their own right, but this is only one book!

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Structure of This Book

There are 11 chapters, 2 appendices, and a glossary in this book. RFID is a broad subject, so while we think all of the sections of the book are necessary to tell the whole story, not every section will appeal to every reader. The book's structure divides neatly into two broad areas of interest: a general overview (the "Everyone" section), which should be of interest to any reader, and a section containing more-technical discussion ("Developers and Architects").

Everyone

Chapters 1 and 2 cover the basics of RFID and give readers a framework for further discussion. Most readers will also want to cover Chapters 9-11 for discussions of key topics in RFID.

Developers and Architects

Anyone who will actually be creating or interfacing with an RFID system or who wants to understand how all of the pieces work should read Chapters 38, where we get into the details of tags, readers, middleware, and the RFID information service.

What's in Each Chapter

The chapters proceed in what we hope you will find is a natural progression, starting with an introduction and overview, a chapter on architecture, and then a discussion of each component of that architecture. Following our discussion of all of the pieces, we take a look at the qualities of the entire system that deserve special attention, such as manageability, security, and privacy. We then wrap things up in the final chapter with a look at the future of RFID. Here's a brief tour:

Chapter 1, *An Introduction to RFID*

> Defines RFID and introduces some of the fundamental concepts

Chapter 2, *RFID Architecture*

> Describes the parts of an RFID system, their relationships to each other, and some of the functional and service-level requirements specific to RFID

Chapter 3, *Tags*

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Describes the tags that attach an identity to an item and communicate that identity to readers

Chapter 4, *Tag Protocols*

Examines how tags talk to readers

Chapter 5, *Readers and Printers*

Describes the readers that communicate with tags and connect RFID-tagged items to the network

Chapter 6, *Reader Protocols*

Covers how readers talk to middleware and applications

Chapter 7, *RFID Middleware*

Describes the middleware that manages RFID information and edge devices

Chapter 8, *RFID Information Service*

Examines the storage and use of RFID information

Chapter 9, *Manageability*

Discusses some of the specific concerns related to managing RFID devices on the edge of the network

Chapter 10, *Privacy and Security*

Discusses real concerns regarding the impact of RFID on security and privacy, as well as dispelling some of the myths

Chapter 11, *The Future*

Provides a look at where RFID may take us in the next few years

Appendix A, *EPC Identity Encodings*

Provides additional encodings to complete the discussion of EPC identity encodings introduced in Chapter 4

Appendix B, *References*

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Lists resources available for those who want more information on RFID or who would like to become involved in defining the technology as it develops

Glossary

Includes key acronyms and terms to help you sort out some of the jargon used in this field

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Conventions Used in This Book

The following is a list of the typographical conventions used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, pathnames, and directories

`Constant width`

Is used for examples and the contents of files

`Constant width italic`

Shows text that should be replaced with user-supplied values

You should pay special attention to notes set apart from the text with the following icons:

Note: This is a tip, suggestion, or general note. It contains useful supplementary information about the topic at hand.

Warning: This is a warning or note of caution, often indicating that your money or your privacy might be at risk.

---

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the U.S. or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/rfid

The authors also maintain a site where you can find a list of RFID resources, along with news and updates:

http://www.rfidessentials.com

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

http://www.oreilly.com

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at http://safari.oreilly.com.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Acknowledgments

We would like to thank the many people who have provided both tremendous support and direct help to make this a better book. Much of what is good about this book is due to their diligence and care, and any mistakes or weaknesses are entirely ours.

We would like to thank our editors at O'Reilly Media, Inc.: Mike Hendrickson for getting us off to a great start; Andrew Odewahn for seeing this through; and Susan Brown Zahn, Marlowe Shaeffer, and Mary O'Brien for bringing it all together. Many people went out of their way to help by reading early drafts and providing detailed advice and reviews. We would like to thank Chuk Biebighauser, Jeff Coffin, Bryan Helm, Ravi Kalidindi, Brendan McCarthy, Ben Menasha, and Tim Seltzer. We were also lucky enough to have some very challenging technical reviewers critique our penultimate draft, and this is a much better book because of their help. Many thanks to Kevin Ellison from Sun Microsystems, Inc.; Graham Gillen from VeriSign, Inc.; John Hill, Jim Both, and Dennis Dearth from ESync; Mark Johnson of RFID Tribe; Steve Lazar of Texas Instruments; Mark Palmer from Progress Software; Ken Traub from ConnecTerra; Bryan Tracey from GlobeRanger, Inc.; and Marc Linster of Avicon. A special thanks goes to Paul Bodifee of the "Signals Collection 4045." We did not use his photos in the final draft, but his information about WWII-era RF technology was a great inspiration (see Appendix B for a link to the Signals Collection web site). We would also like to thank the entire RFID team at Sun Microsystems for their passion for this space and for their support.

Bill would also like to thank his family for tolerating a sleepless, grumpy, and sometimes overwhelmed husband, father, and son for these past months. Thanks to Janise and Rhian, and thank you to Bill's parents, Janet and Bill.

Himanshu would like to thank his wife, Teju, for her encouragement and support through the writing of this book. He would also like to thank his little ones, Rohan and Ritu, for their innocence, boundless energy, and enthusiasm. Finally, thanks to his parents, Vinod and Yashavati, and his siblings, Rajendra and Jigu, for their lifelong love and guidance.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Chapter One. An Introduction to RFID

In Twelfth Night, Shakespeare wrote, "Some are born great, some achieve greatness, and some have greatness thrust upon them." RFID is one of the more recent four-letter abbreviations to have greatness thrust upon it in a flurry of industry mandates, governmental legislation, and hyperbole. RFID stands for Radio Frequency Identification, a term that describes any system of identification wherein an electronic device that uses radio frequency or magnetic field variations to communicate is attached to an item. The two most talked-about components of an RFID system are the*tag*, which is the identification device attached to the item we want to track, and the*reader*, which is a device that can recognize the presence of RFID tags and read the information stored on them. The reader can then inform another system about the presence of the tagged items. The system with which the reader communicates usually runs software that stands between readersand applications. This software is called *RFID middleware* . Figure 1-1 shows how the pieces fit together.

Much of the recent interest surrounding RFID has arisen from mandates and recommendations by government agencies such as the U.S. Department of Defense (DoD) and the Food and Drug Administration (FDA), and from a few private sector megacorporations. For instance, in an effort to improve efficiency, Wal-Mart called for its top 100 suppliers to begin providing RFID tags by early 2005 on pallets shipped to its stores. This mandate caused the companies in Wal-Mart's supply chain to focus on implementing RFID solutions. Companies worked to decide which tags and readers to use how to attach tags to (or embed them in) containers or products, and how to test the read rates for RF tags on pallets as they moved through doors and onto trucks. Several companies have announced their support for what are now commonly known as *tag and ship* applications, which tag a product just before shipping it somewhere else, but few of these companies have moved beyond minimum compliance with the mandates to using the information on RFID tags to increase efficiency in their own internal processes.

Figure 1-1. Spreadsheet of CDDiscs and days

The mandates have also focused most of these early implementations on tagging, and thus on the physical side of the RFID systems. However, while it is important to both select tags and readers and

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

find just the right arrangement of antennas to recognize tags as they move through docks and conveyor belts, the true benefit (and complexity) of RFID systems doesn't come from reading the tags, but from getting the information from those reads to the right place in a usable form. The first 100 were only the beginning of the Wal-Mart RFID rollout. Many more suppliers will be tagging pallet and cartons and some individual items by the end of 2006. Meanwhile, the biggest news in RFID may surround the ePedigree initiatives aimed at reducing counterfeiting and improving efficiency and safety in the distribution of pharmaceuticals. By then, many more new initiatives will have been launched to apply RFID to other industries in ways we can hardly predict (although we'll try in Chapter 11).

In the pages to come, we explain the essentials of an RFID system, and in order to put these concepts in perspective, we will also briefly discuss the history, current status, and future of the technology. This book will give you the information and understanding you need to take on your first RFID project, but we hope you'll find it just as useful once you become a seasoned veteran in the field.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.1. The Case for RFID

RFID technologies offer practical benefits to almost anyone who needs to keep track of physical assets. Manufacturers improve supply-chain planning and execution by incorporating RFID technologies. Retailers use RFID to control theft, increase efficiency in their supply chains, and improve demand planning. Pharmaceutical manufacturers use RFID systems to combat the counterfeit drug trade and reduce errors in filling prescriptions. Machine shops track their tools with RFID to avoid misplacing tools and to track which tools touched a piece of work. RFID-enabled smart cards help control perimeter access to buildings. And in the last couple of years, owing in large part to Wal-Mart and DoD mandates, many major retail chains and consumer goods manufacturers have begun testing pallet- and case-level merchandise tagging to improve management of shipments to customers.

Part of what made the growth in RFID technologies possible were the reductions in cost and size of semiconductor components. Some of the earliest RFID tags were as big as microwave ovens, and the earliest readers were buildings with large antennas, as described in Chapter 3. Figure 1-2 shows a modern RFID tag (in the clear applicator) and a reader.

Figure 1-2. A tag and reader (image courtesy of Merten G. Pearson, D.V.M.)

Note how the bar code on the applicator matches the code read on the reader. The tag is inside the

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

applicator in this picture and is about the size of a grain of rice. It's very similar to the glass capsule tag shown in Figure 1-3.

Figure 1-3. The VeriChip is smaller than a dime (image courtesy of Applied Digital)

Like RFID tags, the size of tag readers is shrinking. While most tag readers are still the size of a large book, smaller and less expensive readers may open up opportunities for many new RFID applications that, over the coming years, could become a normal and mostly unnoticed part of our lives. Figure 1-4 shows one of the smallest readers currently available.

As individuals, we must consider what impact this technology will have on our lives. Such an efficient and unobtrusive tracking mechanism can be used in ways that raise concerns about individual privacy and security. As citizens, we must understand the benefits and costs of this technology and its impact on us. Conversely, as developers, we know that "unobtrusive" is a euphemism for "works correctly because a great deal of effort went into design, implementation, and testing." It is our job as managers, architects, and developers to make the technology work so well that it disappears. The following pages will provide an introduction to how RFID works.

Figure 1-4. SkyeTek's SkyeRead M1-mini (image courtesy of SkyeTek)

1.1.1. Advantages of RFID over Other Technologies

There are many different ways to identify objects, animals, and people. Why use RFID? People have been counting inventories and tracking shipments since the Sumerians invented the lost package. Even some of the earliest uses of writing grew from the need to identify shipments and define contracts for goods shipped between two persons who might never meet.[*] Written tags and name

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

badges work fine for identifying a few items or a few people, but to identify and direct hundreds of packages an hour, some automation is required.

[*] Lawrence K. Lo, "AncientScripts.com: Sumerian," http://www.ancientscripts.com/sumerian.html.

The bar code is probably the most familiar computer-readable tag, but the light used to scan a laser over a bar code imposes some limitations. Most importantly, it requires a direct "line of sight," so the item has to be right side up and facing in the right direction, with nothing blocking the beam between the laser and the bar code. Most other forms of ID, such as magnetic strips on credit cards, also mus line up correctly with the card reader or be inserted into the card reader in a particular way. Whether you are tracking boxes on a conveyor or children on a ski trip, lining things up costs time. Biometrics can work for identifying people, but optical and fingerprint recognition each require careful alignment similar to magnetic strips. Facial capillary scans require you to at least face the camera, and even voice recognition works better if you aren't calling your passphrase over your shoulder. RFID tags provide a mechanism for identifying an item at a distance, with much less sensitivity to the orientation of the item and reader. A reader can "see" through the item to the tag even if the tag is facing away from the reader.

RFID has additional qualities that make it better suited than other technologies (such as bar codes or magnetic strips) for creating the predicted "Internet of Things."[*] One cannot, for instance, easily add information to a bar code after it is printed, whereas some types of RFID tags can be written anc rewritten many times. Also, because RFID eliminates the need to align objects for tracking, it is less obtrusive. It "just works" behind the scenes, enabling data about the relationships between objects, location, and time to quietly aggregate without overt intervention by the user or operator.

[*] This term was originally attributed to the Auto-ID Center. We will discuss both this term and the Auto-ID Center in more detail later in this book.

To summarize, some of the benefits of RFID include the following:

Alignment is not necessary

A scan does not require line of sight. This can save time in processing that would otherwise be spent lining up items.

High inventory speeds

Multiple items can be scanned at the same time. As a result, the time taken to count items drops substantially.

Variety of form factors

RFID tags range in size from blast-proof tags the size oflunch boxes to tiny passive tags smaller than a grain of rice. These different form factors allow RFID technologies to be used in a wide variety of environments.

Item-level tracking

Ninety-six-bit RFID tags provide the capability to uniquely identify billions of items (more abou

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

this in Chapter 3).

Rewritability

Some types of tags can be written and rewritten many times. In the case of a reusable container, this can be a big advantage. For an item on a store shelf, however, this type of tag might be a security liability, so write-once tags are also available.

1.1.2. The Promise of RFID

As previously mentioned, the capability to attach an electronic identity to a physical object effectively extends the Internet into the physical world, turning physical objects into an "Internet of Things." Rather than requiring human interaction to track assets, products, or even goods in our homes, applications will be able to "see" items on the network due to their electronic IDs and wireless RF connections.

For businesses, this can mean faster order automation, tighter control of processes, and continuous and precise inventories. Business partners will finally be able to share information about goods end to end through the supply chain and, just as importantly, to instantly identify the current location and status of items. For example, pharmacists will be able to track how long perishables have been out of refrigeration.

Military personnel, law enforcement officers, and rescue workers may soon use RFID tags to help build and configure complex equipment based on rules enforced by tag readers. RFID already tracks expensive and sensitive assets used in each of these fields.

For individuals, RFID could provide more effortless user interfacesso-called "smart" systems that could tell you, for example, which clothes in your closet match. Smart medicine cabinets could warn you against taking two drugs that might interact negatively. It's even conceivable that supermarkets of the future may not have checkout standsyou may fill your cart with goods that a reader in the cart will scan and add to your total. Video monitors on the shelves will offer specials on complementary products; they may even offer to guide you to all of the ingredients for a recipe, based on some of the items you've already chosen. As you walk out the door, you will place your thumb on a pad on the cart handle to approve payment. A shoplifter, however, wouldn't make it very far before the readers recognized unpurchased items passing beyond the sales floor.

Some of these applications are already running in pilot stages. Libraries and video stores use RFID to thwart theft. Some shoppers in Japan use RFID-enabled cell phones to make purchases from vending machines. Businesses use RFID to track goods, and animal tracking has been around for years.

RFID will enter the home and the supermarket aisle when the prices of readers and tags become low enough and when the information infrastructure to use and maintain the new technology is in place. Some of these applications may seem far-fetched, but they are things we know we can do with a bit of engineering. What RFID promises most is to surprise us with uses we can't even imagine at this stage of adoption.

Note: For more on the future of RFID, see Chapter 11

---

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.2. The Eras of RFID

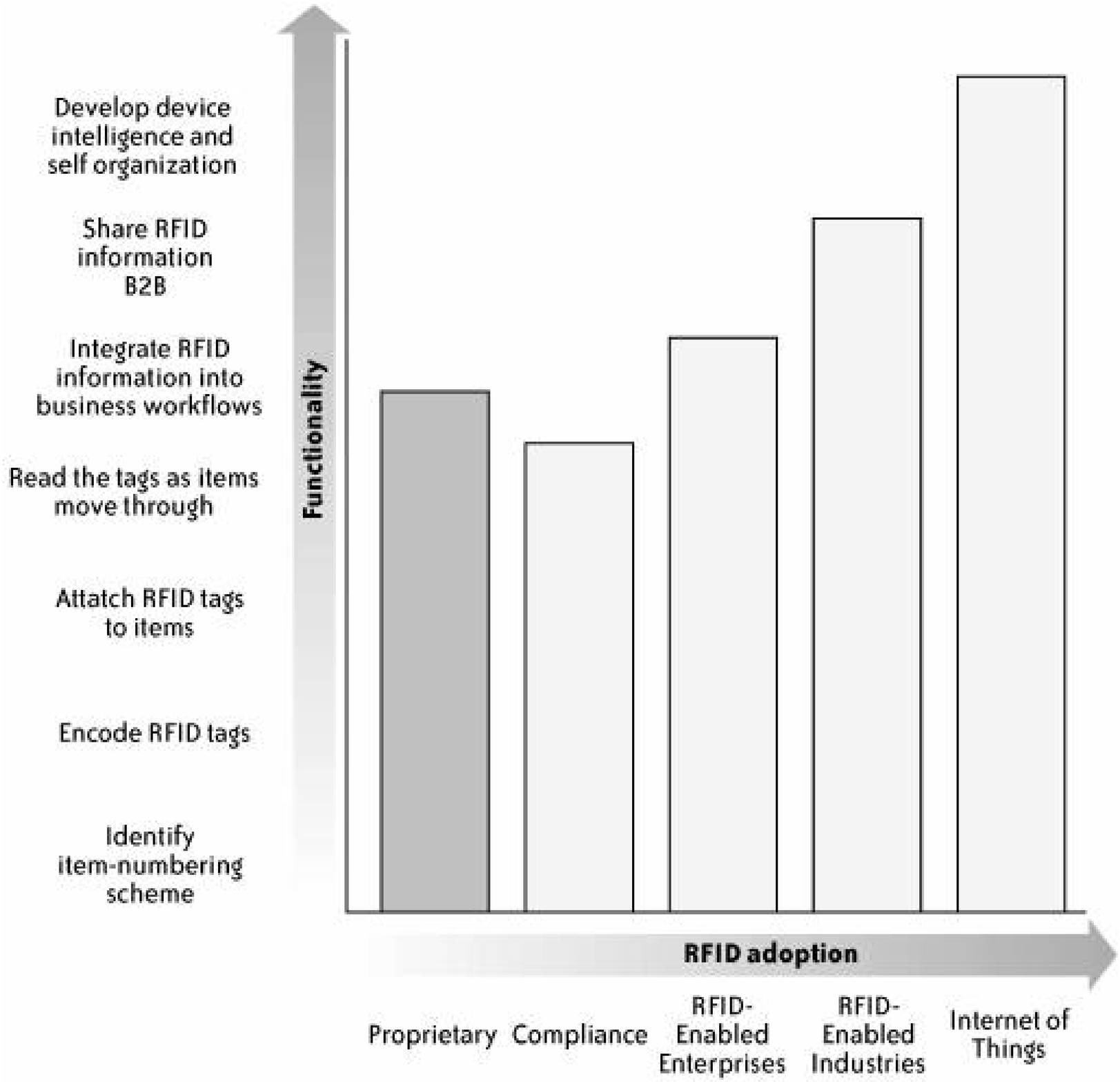
The progress of RFID adoption divides naturally into eras: the Proprietary era, the Compliance era, the RFID-Enabled Enterprise era , the RFID-Enabled Industries era, and the Internet of Things era. Ir Figure 1-5, you can see when some of the capabilities of RFID technology became, or will become, available.

In the beginning, during the Proprietary era, businesses and governmental entities created systems designed to track one particular type of item, and this tracking information typically remained within the same business or governmental entity. In the Compliance era (the present era), businesses implement RFID to meet mandates for interoperability with important customers or regulatory agencies but often don't use the RFID data themselves. The future will bring the era of the RFID-Enabled Enterprise, where organizations will use RFID information to improve their own processes. The era of RFID-Enabled Industries will see RFID information shared among partners over robust and secure networks according to well-established standards. The final RFID era that is currently foreseeable is the era of the Internet of Things. By this time, the ubiquity of RFID technology and other enabling technologies, combined with high standards and customer demand for unique products based on this infrastructure, will lead to a revolutionary change in the way we perceive the relationship between information and physical objects and locations. More and more, we will expect most objects in our daily lives to exist both in a particular place, with particular properties, and in the information spaces we inhabit. For instance, a park bench has a particular shape, color, and location, but with a tag, it can also have a list of notes left by people who have stopped there to rest. The list is just as real as the color of the paint, and just as much an attribute of the bench. This is now a green metal bench on the north end of Shaker Park with "a great view of the sunrise," according to "pigeon-guy."

Figure 1-5. New capabilities will come with deeper adoption

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



Companies, regions, and even individuals will move through these eras at different paces. Even now, some users of RFID are touching on the RFID-Enabled Industries era as emerging standards make this possible, while others are still in the Proprietary era. In many other areas, RFID has not been adopted at all.

1.2.1. The Proprietary Era

For almost 60 years now (triggered by the development of transistors in 1947), businesses and governmental entities have used RFID to track items and provide access control to facilities. The smaller size and greater durability of transistors made it possible to attach transmitters to valuable items, and over time developments such as improved batteries, integrated circuits, and microchips reduced the cost of the transmitters (tags), allowing tracking of less valuable items. Some of the applications in this era included the tags used to track rail cars and the chassis tags that have been used since the 1980s to track automobiles through an assembly line. In the 1970s and 1980s, RFID was used for tracking dairy cattle. In the 1990s, the beef industry began tracking cattle using $5.00 ear tags. Expensive, proprietary RFID tags, which were usually recycled, were a major characteristic

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

of this era. The reuse spread the cost out, such that a single use might cost only a few cents. Some of the systems developed during this era were technically advanced and tightly integrated into business processes, but they were characterized by both poor support for sharing information between trading partners (incompatible IDs, for instance) and costly reader and tag components.

1.2.2. The Compliance Era

The steep drop in semiconductor prices and widespread adoption of broadband networking at the end of the 20th century triggered an era we call the Compliance era. In this era, the U.S. DoD and large retailers such as Wal-Mart and Tesco began asking their suppliers to tag pallets (and sometimes individual items) with RFID tags. Their mandates required that the tags conform to emerging standards. The anticipated volume of tags that will be purchased to meet these mandates has pushed these same standards much closer to universal adoption, which has greatly reduced the cost of components.

Because this tagging effort has arisen from efforts to comply with a mandate rather than a perceived business opportunity for suppliers, many suppliers have implemented so-called tag and ship applications , which apply RFID tags to pallets or cartons as they leave the supplier's control and ship to a retailer or government customer. The supplier then uses the information on the tag to fill out an advanced shipment notification (ASN), notifying the recipient to expect goods tagged with particular identifiers. The supplier does not tag the goods early enough in its own cycle to take advantage of tracking to improve its processes.

In short, some of these suppliers currently may see RFID as an added expense and burden. They are driven almost entirely by cost efficiency in complying with the mandates, and we do not see applications that integrate RFID into internal business processes or extensive sharing of RFID information between partners in this era. Also, the new, less expensive tag technology is still prone to manufacturing defects, and, due in part to early implementation of the tag standards, often Compliance-era tags do not perform as well in practice as the tags in the Proprietary-era systems. Thus, while adoption of RFID is on the rise, there has actually been a slight slump in its capabilities (in the sense of how the technology is used, if not the purposes to which it is applied). Fortunately, new, more efficient standards and rapid improvement in both tag quality and throughput are quickly closing the gap.

1.2.3. The RFID-Enabled Enterprise Era

As standards stabilize and component costs fall, many organizations will begin to implement RFID tracking within their internal processes. This will allow them to measure the pulse of their distributior systems for materials, assets, and products and to keep real-time inventories of items, such as the location and age of perishable goods. During this era, declining costs will inspire a steady transition from tracking shipping units to tracking individual items. Other types of sensors will join RFID to monitor information such as the highest temperature to which an item has been exposed or whether gases produced by spoilage are present. Labor-intensive bar code inventories will, in many cases, be replaced with scan-by inventories, allowing someone simply to walk down the aisle with a handheld reader. Similarly, portal readers at the door will record the entry and exit of every item in a shop or warehouse.

In response to demand, more manufacturers will begin tagging items with standards-compliant tags

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

at the point of origin, taking over from the suppliers and distributors who performed this role in the Compliance era. Business integration products and inventory tools will begin to fully support individual item tracking.

However, even with widespread internal adoption and tagging at the origin of the supply chain, it will take time for businesses to develop the agreements and security to allow organizations to share RFID information with one another (so-called business-to-business, or B2B, communication). While businesses will continue to share whatever B2B information they have shared in the past, the new RFID information will be used largely within the enterprise.

1.2.4. The RFID-Enabled Industries Era

In this era, RFID standards, RFID information networks, business agreements, and comprehensive security and privacy policies will solidify to the point where entire industries and supply chains can share appropriate information reliably, trusting that only authorized users can see any sensitive information. This will probably include a redefinition of what constitutes sensitive information, as unexpected revelations are likely to arise from the study of detailed instance data (where only aggregates and estimates were available before). Safety overstock inventories will drop, along with fulfillment times and costs, due to theft and error. Simply knowing "what was where when" provides a powerful tool for applications that we have only begun to realize. Expect to see new products, partnerships, and even whole cottage industries develop by harnessing the advantages of this new flow of information.

For many years, pundits have predicted that businesses will begin to share information much more freely, but it hasn't yet happened on the scale they predicted. Why might RFID information be different? The key is that RFID information is very concrete. It can be difficult for partners to agree on what an item description should contain, but RFID information can be as simple as a universally standardized ID code, a timestamp, and a universally standardized location code. (For more information on ID and location codes, seeChapter 4.)

While we will realize all of these advantages, significant challenges will also arise as managing all of this information, including the sensors and software that produce it, becomes more difficult. Ad hoc or dated architectures will creak and even fail under the load.

1.2.5. The Internet of Things Era

This final era will be triggered by widespread adoption of RFID technology and the associated demand for easier management of distributed sensor networks, as well as by a reduction in the cost of smart devices and tags. Lower costs and greater demand for information will commercialize existing technologies already in use so that military and manufacturing applications can create self-organizing networks of cheap, expendable components with extremely low incremental maintenance and management costs. This technology will finally make it possible to adopt RFID technology on retail floors, in farm fields, and in homes. It will expand the group of businesses adopting the technology to include even the smallest entrepreneurs. RFID tags will cease to simply be labels applied to items and will more often be added as integral parts at the time of manufacture or as part of the packaging. At this stage of development, the idea that an item has a digital identity will become as basic as the idea that an item has a color or weight or size.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

In this era, physical objects will be tied to the Internet through their digital identities. Just as we expect to be able to do a quick Internet search to find the answer to an obscure trivia question, we will expect to be able to wave a soda bottle past a cell phone and find out where and at what specific time the soda was manufactured and, if we like, the last known location of every other bottle of that soft drink manufactured within the same hour at a particular location. Invisible digital graffiti associated with identities attached to physical objects will surround us in the form of messages posted to the Internet. But this could go far beyond messages like "Joey was here"for example, a how-to video for using a piece of equipment could be associated on the Internet with the equipment's tagged ID. By this era, we probably won't think about RFID technology any more than we think abou electrical technology today. We will simply expect it to work.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.3. Application Types

Certain broad types of RFID applications characterize whole approaches to this technology and are different enough in considerations and implementation to warrant a separate discussion. The tree in Figure 1-6 shows RFID's relationship to other identity systems, as well as the relationships between different types of RFID.

The term "autoid," short for automatic ID, describes any automated system for attaching an identity to an item. Real-Time Location Systems (RTLSs) are automated systems for tracking the location of an item. Notice that RFID is related only indirectly to RTLSs and that RFID is only one type of automated identity system. We will discuss RTLSs and how they can complement RFID in Chapter 11.

Obviously, we can't fit all of the possible uses of RFID into five simple categories, so we've left out a few applications. For example, while we don't directly discuss payment systems, pay-at-the-pump systems based on RFID raise some (but not all) of the same concerns as access control systems. Refer to the "Access Control" section for information on considerations and implementation; although payment systems don't always have issues with tailgating, both types of system must overcome counterfeiting and require strong audit procedures. We've also left out some unique applications, such as using RFID tags on compatible pieces of equipment to coordinate assembly of mobile structures in the field for military deployments and trade shows. However, these five categories are inclusive enough to provide at least some sense of the issues and considerations involved in typical RFID applications. The future will bring even more varied applications, but they will raise many of the same concerns as the applications in these categories.

Figure 1-6. Relationships among the various types of RFID applications

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.3.1. Access Control

Access control applications are RFID systems used to selectively grant access to certain areasfor example, RFID tags attached to an automobile or held in a person's hand as a card, key chain, or wristband may allow access to a road, building, or secure area.

1.3.1.1. Considerations

Primary considerations for this type of application are:

Anti-counterfeiting

Counterfeit tags must be recognized and attempts to use or manufacture them discouraged.

Tailgating

Tailgating occurs when an unauthorized person or vehicle enters just behind an authorized person or vehicle before the gate or door can close.

Emergency access

In an emergency, the access control system must allow emergency personnel or vehicles access to secured locations. It must also allow nonemergency personnel to evacuate without getting in the way of the emergency response team. Yet emergency access provisions must no provide an attractive "exploit" that allows unauthorized persons to defeat the system by staging a false emergency.

1.3.1.2. Implementation

To address tailgating and emergency access considerations, most access control implementations include a barrier to entry such as a gate or door. The timing of the door and additional sensors can "singulate" vehicles or persons who enter, restricting access to one at a time.

The anticounterfeiting consideration is the most difficult to address. Many existing systems try to keep the construction and programming of their tags a secret in hopes that no one will reverse-engineer them, but several of these systems have already proven vulnerable to such attacks. Fortunately, most of these systems were noncritical or were backed by audit procedures that caught reverse-engineering attempts. Strong encryption of both a randomly generated challenge message and digital signatures might reduce the opportunity for counterfeiting in cases where the added cost of more complex tags is acceptable, but in general, a human guard, video camera, or police interceptor vehicle is sufficient to discourage all but the most incorrigible. The lesson is that RFID systems do not operate in a vacuum, and effective systems often involve supporting automation, other sensors, and manual processes to ensure success.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.3.2. Tag and Ship

Tag and ship applications are minimal RFID systems that allow a user to associate an RFID tag with an item, apply the physical tag to the item, and then verify that the tag operates properly while attached to the item. In some cases, these systems even use pre-encoded tags to further reduce cost (see the "Implementation" section for a warning about this practice).

1.3.2.1. Considerations

Primary considerations for this type of application are:

Cost

Because the drivers for this sort of application typically comply with a mandate, keeping cost low is the primary concern of the end user. This includes initial cost and the total cost of ownership (TCO) over time, from upgrades and repairs to monitoring and maintenance.

Isolation

Tag and ship systems are in some cases the first automated system to be deployed at a given location. The support and maintenance infrastructure needed for such systems, and even the floor space they take up at the dock or shipping area, often aren't readily available.

Tag failure

Manufacturing defect rates are still high for the smart labels(paper tags with embedded RFID antennas and chips) used in this type of application. Since the logistics of trying to reorder missing numbers would be difficult, the system may have to discard one or more labels before finding a functioning label to apply to the item. For pre-encoded labels, this means discarding serial numbers.

Impact on processes

Because this type of application is typically implemented in response to an external mandate rather than a perceived internal need, the application's impact on the throughput of what may be a finely tuned manual process can be an important issue. RFID labels appear as only an added cost from this perspective, and each extra second spent attaching labels raises that cost

1.3.2.2. Implementation

To implement a tag and ship system, the first requirement is to create or purchase a tag and ship device with a low TCO. A self-contained appliance that operates as simply and reliably as possible

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

should be the best fit. Repairs and maintenance should be no more difficult than changing a light bulb. This means it is often cheaper to buy a few spare devices than to maintain and manage a single device or pair of devices. Owning spare devices can, for example, lower the costs of maintaining and supporting the device in harsh environments where visits from IT personnel are rare.

For the device to resolve the other primary considerations, it should operate simply and quickly and impact manual processes as little as possible. This means there cannot be a complex workaround for failed labels or a multistep process for applying the labels. The simplest way to implement this system is to use labels that are encoded by the appliance itself rather than those pre-encoded by a tag vendor. This enables you to simply skip a failed label and write the same identity to successive new labels until one succeeds. Wherever possible, a tag and ship system should also use an automated "print and apply" device that encodes the label, prints human-readable and bar code information in ink on the surface of the label, and either presses the adhesive label onto the item or blows it on with compressed air. When items are too awkwardly shaped for this approach, the system should provide a visual prompt to indicate how the tag should be applied manually (position and orientation) to reduce errors by the operator.

1.3.3. Pallet and Carton Tracking

One of the most commonly mentioned forms of RFID, pallet and carton tracking, essentially puts a "license plate" on a shipping unit made up of one or more individual items.

1.3.3.1. Considerations

Primary considerations for this type of application are:

Pallet or carton integrity

This type of tracking works best with a shrink-wrapped pallet that contains only one type of item; the pallet ID is then associated with a simple item count. It also works well with a mixed pallet that has a more complex manifest, describing counts for more than one item. Pallet and carton tracking can be ineffective if there is a possibility that the pallet or carton may be broken down and reconstituted. In this case, the counts or manifest may become invalid.

Pallet orientation

Pallets have six sides. Given that the bottom is typically inaccessible, we still have five choices when deciding where to attach the tag. Because most dock doors are roll-ups, placing a reader overhead can be difficult, so few implementations tag the top of the pallet. Most pallets have a orientation, and shippers typically place pallets with a certain side facing out, so in most cases it isn't necessary to put tags on all four sides. Even if putting a tag on each side seems reasonable, this means creating and reading a whole set of duplicate tags, which can cause problems for both printers and readers. The front of a pallet is blocked by the forklift, although it is possible to place a reader there. (We talk more about antenna placement in general and forklift pitfalls in particular in Chapter 5.) The back of the pallet is blocked from the readers by the pallet itself and then by the forklift upon passing through the portal. This leaves the left an

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

right sides available for tagging. No strong consensus exists on which to use, so most implementations place readers on each side but tag only one side of the pallet.

Interfering contents

You might expect that a tag on the outside of a box would be easily visible to a reader, regardless of its contents, but this is not so. Imagine the reader as a bright light and the tag as a small mirror. Could you see a small mirror attached to a larger mirror, even if the light of a 300-watt floodlight illuminated it? A tag on a box of metal cans can be just as difficult for a reader to distinguish. If the product contains metal, have mercy and put the tag on a thin foam backing. The added distance will usually create enough space between the tag and the reflected signal to greatly improve read rates.

1.3.3.2. Implementation

Because the considerations for pallet and carton tracking have to do with implementation, we have already covered several implementation ideas in the preceding section. To sum things up:

- Track wrapped pallets with known contents
- Place tags on the left or right side (as seen from the forklift) but not on both sides
- Put a foam backing on the tag to raise it a bit if the contents of your pallet are high in metals or liquids
- Check Chapter 5 for antenna alignment tips and warnings about forklifts.

1.3.4. Track and Trace

One of the earliest uses of RFID was to track dairy cattle. Now, companion animals and livestock of all types are routinely tagged with injectable glass capsules or button ear tags. These tags are used to identify lost pets and to sort, care for, and track the history of livestock. In recent years, RFID has also been increasingly used to track produce and pharmaceuticals. Information from livestock, produce, or pharmaceutical tracking can be critical in the event of a public health threat.

1.3.4.1. Considerations

Primary considerations for this type of application are:

Information sharing

By definition, track and trace applications require information sharing. One of the key requirements when merging information is coordinated identification. If one producer claims this flat of strawberries should be called 12345 and another claims that a different flat of

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

strawberries should be called 12345, how can the inconsistency be reconciled? What if one producer reuses numbers and sends a flat 12345 this week and another flat 12345 next week? In any track and trace system, each identity must be unique across all producers and for as long a period as the information must be maintained. This can be accomplished in several ways, but the simplest method is either to assign a prefix to each producer to put at the beginning of their identities or to assign blocks of identities to each producer from a central authority (which accomplishes the same thing). Whatever method is used must be universally enforcedotherwise, the integrity of the data in the system will be suspect.

Role- and instance-based access control

Sharing information includes pooling information with competitors. Track and trace systems must have a provision for role- and instance-based control over access to information. In simplest terms, a role is a job, such as veterinarian or retailer, while an instance is a particular person. For example, a retailer may need general information but should not be able to view sensitive information about individual producers or manufacturers. A veterinarian should be able to view detailed information, but only for clients with whom she has a professional relationship. A government inspector should be able to see which animals or produce might have been commingled with a certain suspect lot, but might not need to see any other information.

1.3.4.2. Implementation

Rather than "rolling your own" identity management system ("identity" meaning user identity for role- and instance-based access control), when implementing a track and trace system you should buy an off-the-shelf identity management and access control solution in the form of an integrated software stack. Several enterprise software vendors offer these. Likewise, you should not create a completely new and proprietary system for provisioning identities. Because so many entities need to share track and trace information, you should ensure that your system makes use of the emerging identity standards (such as those described in Chapters 2 through 9).

1.3.5. Smart Shelf

A smart shelf system is a set of shelves, or some other container (such as a refrigerator), that constantly keeps track of the individual items it contains. If an item is removed or added, the shelf immediately updates the inventory. By tying the identity of an item to its attributes, such as expiration date or lot number, a system using smart shelvescan immediately locate all expired products and products from a certain lot. An example of a smart shelf system is a system that contains indicators such as horns or lights that warn users if a product has been removed from refrigeration for too long and should therefore be discarded. Similarly, if the user removes two drugs at the same time that are known to interact negatively, the system signals a warning.

1.3.5.1. Considerations

Primary considerations for this type of application are:

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Item-level inventory support

The most important consideration in a smart shelf system is the necessity of resolving inconsistencies between existing applications and a system that handles individual inventory. Most inventory systems currently deal only with tuples made up of stock keeping unit (SKU) codes and a count.

Physics and hardware

Developing a reliable smart shelf system from readers, antennas, and standard shelving is a daunting task due to the complexity of choosing components, placing antennas, and modeling the possible side effects.

Handling spurious reads

A reader may sometimes fail to recognize a tag. This can be due to interference or absorption of the RF signal. For instance, someone may reach for one item and briefly block the signal response of several others. Also, passing carts full of items by the reader at once may cause false positives to appear on a shelf. The systems must be able to deal with these reads in a manageable way.

1.3.5.2. Implementation

Several vendors provide prebuilt smart shelf units. Such units are expensive, but the cost may be justified when compared to the TCO of creating your own version. To handle spurious and intermittent reads, many readers can perform "smoothing," in which the reader configuration sets a threshold number of reads to which a tag must fail to respond before the reader marks its absence. A similar threshold can require several reads before an item is added to a shelf, so tags in a passing cart have a smaller chance of being around long enough to cause confusion. With this kind of threshold in place, tags will likely have weaker signals, which could alternatively cause them to be ignored.

Interfacing with existing inventory systems can be complicated; however, you can greatly simplify things by using middleware to handle individual item events and sending inventory changes only to backend systems. Again, you should compare the TCO of custom middleware to the cost of buying a commercial product.

1.3.6. Challenges

Any new technology introduces both costs and benefits, and RFID is no exception. Let's begin with the challenges it presents:

Cost

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

The most-discussed cost element for RFID systems is the cost ofindividual tags. However, this is just a part of the overall cost. Successful adoption of RFID will require changes to business processes and information systems, personnel training, and, in some cases, customer education.

Accuracy

Many of today's RFID systems are far from robust. Most of the projects we surveyed in early 2005 reported a read accuracy rate of between 80 and 99 percent. What this means is that if we move a pallet carrying 100 tagged boxes past an RFID reader, the reader will fail to recognize anywhere from 1 to 20 tags. There are many reasons for this, but most stem from the inherent challenges in moving liquids and metals using RF communications.

Implementation

Introducing RFID will invariably change your business processes, from how items are labeled to how they are selected, palletized, cycle counted, and so on. For each step in the business process, you will need to incorporate capabilities for exception processingthat is, what to do if the RFID technologies are not working properly. For example, if an RFID reader stops working or a label falls off, how will you continue? Apart from the RF side, there are many challenges to overcome in integrating the RFID data and procedures into your existing systems and the business processes they support.

1.3.7. RFID Adoption Guidelines

With things changing so quickly, it can be difficult for organizations to decide where, when, and how much to commit to any particular RFID product or standard. The following guidelines offer a strategy for approaching RFID. Simple as the steps may seem, ignoring any one of them can lead to lost opportunities at best, and failure at worst.

Determine the business need.

The current process exists for a reason. Before you even think about changing it, make sure you know its strengths, weaknesses, and reasons for being. Don't start from the premise, "Where can I use RFID?" Instead, start by asking, "How can I improve this process?"

Evaluate potential changes.

Carefully assess the costs and benefits of any potential changes. For example, if you want to automate a manual process, ask yourself a few questions about the change. Would a bar code work better than the manual procedure? Could the process be changed in some way to take more advantage of RFID, or conversely, to eliminate it from your process? How will you handle equipment failures or other types of failure? Note that RFID is sometimes mistakenly referred to as a "replacement for bar codes." Bar codes are actually better than RFID for some applications, and they're considerably less expensive. Also, with RFID there is a temptation to

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

develop automated systems with minimal human oversight. But what happens if someone puts a box full of RFID tags on a pallet and accidentally runs it through the reader? Will your systems believe you've shipped 100 pallets? With bar codes, someone would probably scan each pallet manually, which would prevent this error.

Develop a long-term roadmap.

Instead of implementing RFID systems in an ad hoc manner, develop a long-term business justification for adopting RFID and formulate a vision of how your business processes will look in an RFID world. Follow this up by developing a master plan that shows which systems will need to change and how. Show what your application and infrastructure architecture will look like after deploying RFID. Doing the design up front will give you a clear goal and will also promote the necessary discussions between you and your business units, end users, operations staff, IT staff, and business partners. Don't underestimate how much RFID will impact these stakeholders.

Start small.

Develop a proof of concept (prototype) to validate your assumptions. It's better to fail small and learn early than to fail large and have to recover. Don't be afraid to revise the roadmap and architecture based on mistakes in your prototype, and be ready to start the cycle over again, learning more each time and managing risk as you progress.

Run in parallel with existing systems.

Take a lesson from mountain climbersdon't let go of the last toehold until you are sure the new one won't crumble away beneath you. Not only is it more responsible to run the new system in parallel with the old system for a while, but it can also lead to valuable insights that were invisible when looking at either system in isolation. Don't forget to test the workarounds and recovery strategies you identified earlier. You might be surprised to discover that all of the RFID readers in the building fail when you turn on the exhaust fan for the first time. Only when everything runs smoothly should you begin to depend on the new technology.

Be flexible.

Now that you've made these changes, you have a brand new process. The old process took time to develop, and the new one will take time to mature. While this is happening, be ready to take advantage of new capabilities. Readers with new features and smarter controllers will come out; meanwhile, your personnel should be looking for innovative ways to make use of the new equipment. Watching carefully how people use a piece of equipment can provide important clues for streamlining the process.

Share with partners.

You have a great system in place, but your suppliers are still sending signals by carrier pigeon. Work with your less-enlightened trading partners and show them how to improve their own processes. RFID is an evolving technology, so taking a leadership role will allow you to define the agenda and the standards for future integration. Wal-Mart is an example of a company tha

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

has approached a potentially disruptive technology by choosing to lead in its development. As Dr. Alan Kay said, "The best way to predict the future is to invent it." If you keep it all to yourself, you'll just have to change your system when one of your partners chooses a completely different approach.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

1.4. Summary

In this chapter, we discussed the following:

- RFID is a technology that allows a small radio device attached to an item to carry an identity for that item
- RFID has been around a whilelong enough that we can divide its history into eras and begin to predict future trends.
- In the current era (Compliance), cheap semiconductors and fast Internet connections have encouraged retailers and governmental agencies to require suppliers to place RFID tags on shipping units such as pallets and cartons. However, most suppliers are just tagging pallets and shipping them without using the information internally, and even retailers are simply breaking down the pallets on receipt
- As the components get cheaper and the information infrastructure becomes more defined and robust, RFID will be used for an increasingly broad array of tasks.
- There are five main categories of RFID applications. Knowing the type of application in question can tell us quite a bit about special considerations and implementation.
- This is a volatile time for RFID, so we must take a disciplined approach to both acquiring knowledge about it and adopting the technology within our organizations. Keep an eye on key players and standards as we move through this era to see which way the technology will shift.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Chapter Two. RFID Architecture

For our purposes, an architecture may be defined as a decomposition of a particular computer system into individual components to show how the components work together to meet the requirements for the entire system. With this definition in mind, we can confidently say that there is no such thing as a single, universal RFID architecture that fits all requirements for all systems. Likewise, there is no set number of variations on a single theme. Because of a recent confluence of technologies, RFID systems now offer some key functionalities that have a distinct and predictable impact on the architectures of systems that use it. In this chapter, we describe the components that RFID adds to the architectures of these systems and how RFID affects systemic qualities (i.e., nonfunctional requirements of the system, such as performance, security, scalability, and manageability). From these observations we will derive some architectural guidelines for systems that incorporate RFID.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.1. A Confluence of Technologies

RFID may be seen as the next logical step in the progression of tracking systems and sensor networks because of technological advances in several fields. Let's look at some of the developments that have made RFID possible.

Hype and Reality

Every so often a new technology is introduced that seemingly promises to solve a myriad of problems. In 2001, one of the authors was invited to participate in a technology roundtable on web services hosted by the Chief Technology Officer (CTO) of a Fortune 500 company. Each of the panel members was asked to make an opening remark about the very much hyped web services technologies. One of the vendor participants began by declaring web services "revolutionary" and likening their effects on the IT ecosystem to those of a tsunami. The CTO immediately interrupted and stressed that he wanted to have a pragmatic discussion about web services and did not want anyone to indulge in hype. Declaring web services revolutionary would mean that all learning thus far, including sound software development and architectural principles, does not apply to them.

We have found that, in the early adoption phases of emerging technologies, the amount of caution, past learning, and pragmatism applied to projects that make use of those technologies is inversely proportional to the amount of hype that surrounds them.

History teaches us that hype is generally more profitable for sellers than buyers. We have all seen our share of failed projects in which caution and past learning were abandoned in the name of a new wave. The CTO's comment in 2001 about web services rings very true today for RFID systems. Therefore, instead of indulging in hype and issuing heady promises, we'll take a look at how today's RFID technologies have evolved from the advances we have seen over the past five decades in device, platform, and software technologies.

Advances in semiconductor technologies

RFID would have remained a niche technology if it was not for Moore's law and the ability of the semiconductor industry to produce chips that package processing power at a level that makes it affordable for the mass RFID market.

Intelligent devices

Advances in semiconductor technologies haven't just brought down the cost of RFID chipsthey

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

are also the primary drivers behind the development of intelligent devices, including sensors such as RFID readers. Smarter devices and virtually ubiquitous bandwidth have opened up a host of mobility and edge-based applications. RFID is one implementation of the general idea o a "Network of Things" connected together to provide automation beyond the edges of corporate data centers. Smart homes, smart cars, and other smart objects are additional applications that require processing at the edges. Current implementations of smart home systems incorporate a variety of IP-enabled household devices connected to residential gateways that are in turn connected to the Internet. Figure 2-1 depicts the vision of intelligent devices connected to the Internet.

Figure 2-1. Network-enabled life

Broadband wired and wireless networks and cheaper edge processingservers

The availability of pervasive broadband data networks, coupled with affordable yet powerful servers, has led to the development of architectures that move processing to where the business processes are carried out. This means that it is now easier to deploy pieces of enterprise applications in edge locations such as warehouses and stores.

Edge processing capability

Edge processing capability derives from having powerful yet low-cost personal computers and servers deployed at the edges of the enterprise network as well as a broadband connection to the data center. RFID systems put greater computing, data management, and bandwidth

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

requirements on these edges. This is not a unique phenomenon, however, but a continuation o the overall trend. By "edge," we mean any location where business processes are carried out that is outside the data center or central office (for example, on production lines, in warehouses, or at retail locations). Figure 2-2 illustrates this.

Service-oriented architecture

Successful adoption of RFID technologies in your enterprise will depend on how well you integrate the RFID data into your business processes. RFID readers can generate a lot of data. If it is exposed unfiltered to downstream applications, it can overwhelm them. To prevent applications from being flooded with RFID data and to isolate them from physical devices such as readers and antennas, you can use sophisticated middleware components such as event managers. Service-oriented architectures allow us to develop and deploy loosely-coupled modules that interface with each other using web services-based standards. As we will see later in this chapter, many of the RFID middleware components are based on web services standards, and the overall RFID system architecture follows the principles widely accepted today as the underpinnings of service-oriented architectures.

Figure 2-2. The enterprise edge

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.2. Key Functionalities

What does an RFID system look like? Because there are many possible uses for RFID systems, there will naturally be differences in their architectures. For instance, as described inChapter 1, a typical tag and ship application implemented by a consumer packaged goods manufacturer would focus primarily on automating RFID tagging of products and ensuring that the tags can be read by the specified readers at higher than the minimum acceptable read accuracy. Generally speaking, these systems focus on the physical side of the implementation, and outside of generating fairly simple reports such as advanced shipment notifications (ASNs), they tend to have minimal data management/exchange requirements. (The ASN is a manifest indicating what items the receiver of a shipment should expect and what the identities of those items will be.) On the other hand, a pharmaceutical company that wants to track the movement of drugs from manufacturing plants to distributors to retail pharmacies would want up-to-the-minute information, including details on where a particular product is at any point in the process, how and where it was manufactured, and where it has been. It is very likely that both the manufacturer and the retailers will also need some of this tracking information. Thus, such a system will require not only item-level tracking capabilities, but also some degree of business-to-business (B2B) information exchange.

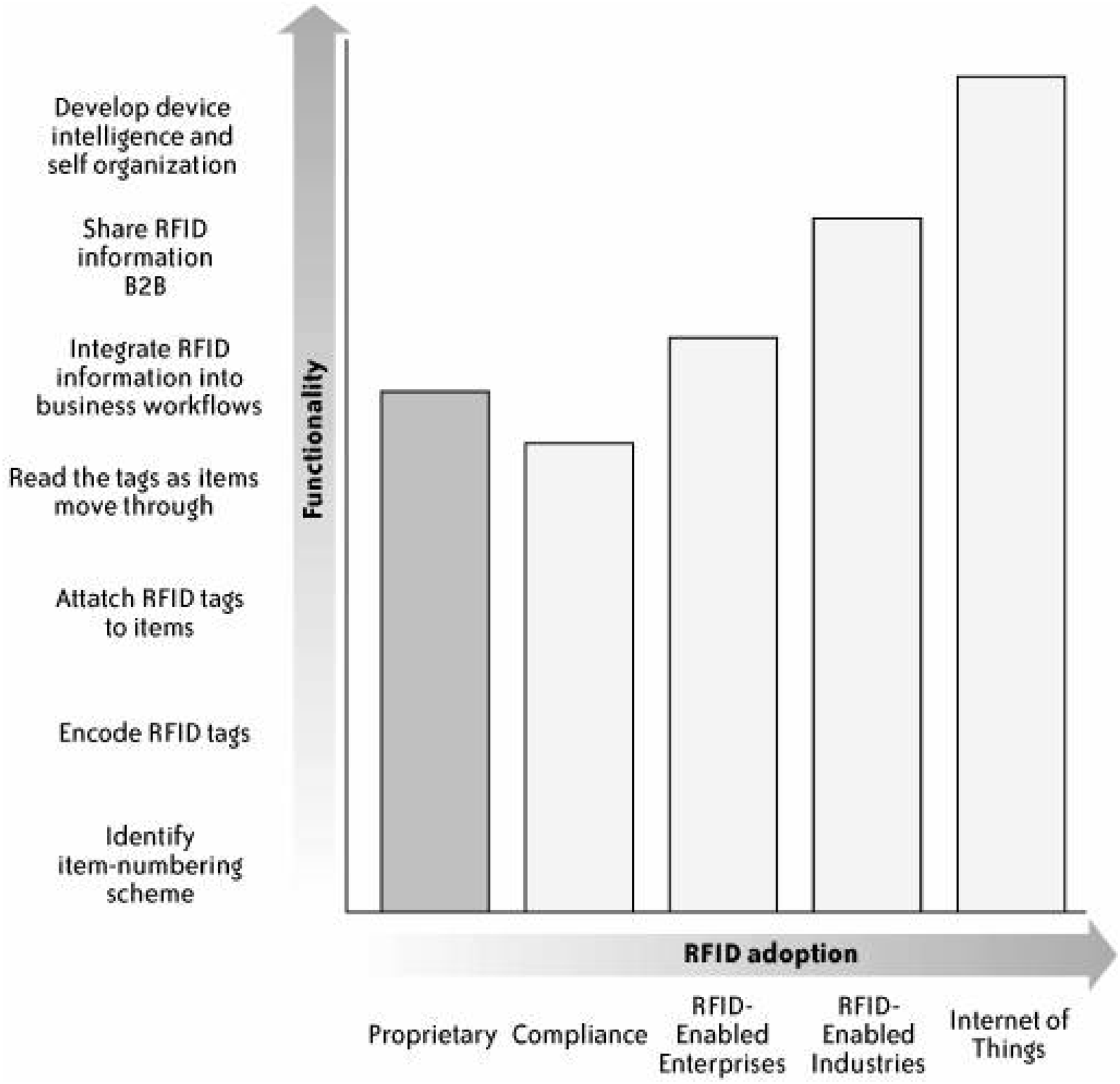
Figure 2-3 lists the five eras of RFID described in Chapter 1 and shows the corresponding capabilities required of the RFID systems in each.

As you can imagine, RFID systems will continue to evolve to meet a wide spectrum of needs and thus will require various architectures. As we stated at the beginning of this chapter, it would be impossible to define a general-purpose RFID system architectureor implementation suitable for all the uses of RFID. However, certain capabilities are required of almost every RFID system. In this chapter, we'll focus on those broad underpinnings. Once we've covered the basics, we'll provide some guidelines for developing architectures that will help you get started.

Figure 2-3. Capabilities required of different types of RFID systems

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



Generally speaking, an RFID system must provide some, if not all, of the following features and capabilities:

- The ability to encode RFID tags
- The ability to attach encoded RFID tags to items
- The ability to track the movement of tagged items
- The ability to integrate RFID information into business applications
- The ability to produce information that can be shared between businesses
- The ability to develop self-organization of intelligent devices

Let's take a closer look at each of these items.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.2.1. Encoding RFID Tags

Encoding RFID tags is a two-step process. The first step is to select an identification scheme to uniquely track the items in question. Once this is done, you can attach those identities to the RFID tags. Let's look at both steps in greater detail.

2.2.1.1. Deciding on an item-numbering scheme

Identification is the act of recognizing the identity of an object or item. But what is an identity? In RFID, an *identity* is a string of letters and numbers attached to an item to allow a person or an automated system to recognize either that item's type or even the unique item itself. As an analogy, consider the system used to identify books in a library. When you visit a library to find books on RFID, you can determine from the card catalog or computer database that these books will be in the 600sthat is, that each will be labeled with a code starting with a 6 in the Dewey Decimal System(or, more likely, the Universal Decimal Classification (UDC)system, which grew out of Dewey's older system). UDC numbers beginning with 6 all deal with technology. Using the UDC number, we can fin all of the copies of a particular book, film, or audio recording in the same place on a shelf or digital storage medium. For example, Klaus Finkenzeller's*RFID Handbook* (John Wiley & Sons) has the number 658.7'87 in UDC.

Note: Notice how UDC uses characters beyond the Arabic digits to create identities. For this reason, calling a UDC identity a "number" can be misleading. This is also true of identity schemes we discuss later in this book. Yet many of these codes are called numbers in formal usage; where it is necessary for clarity, we will retain this usage.

Under this system, 100 copies of the same title would each have the same UDC code. The code, then cannot tell us which copy of the book we are holding. To include that information we would have to add another identifier: a serial number. For example, we might create our own serialized UDC identity with a space separating the serial number from the UDC code. In such a scheme, copy 5,023 of *RFID Handbook* would be numbered 658.7'87 5023, allowing us to confidently refer to an individua book rather than a title or a printing.

Note: Individual identity schemes used and proposed for actual libraries and bookstores are slightly more complex than what we've described here. We are not proposing "space and a number" as a new encoding, but this discussion serves as a good example of how you might provide for individual item tracking.

Consider the need to uniquely identify entities, be they products, containers, physical assets, animals or even human beings. For a large enterprise, there could be millions of items moving through the supply chain at once. A numbering system could be used to uniquely identify these items, but its value would be limited if no one outside of your company could understand it. An industry or universal standard can help.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Multiple Identities

An object can have multiple identities. Consider the*RFID Handbook*. In addition to its UDC identifier, the same edition of the same book has the U.S. Library of Congress code TS160.F5513 and the International Standard Book Number (ISBN) 0-470-84402-7. Different identifiers are required by different organizations. The Library of Congress uses the Library of Congress Number, and to that organization the ISBN is just another attribute of the book, like its title. To a bookseller, the ISBN is the identity. It's even printed on the back as a bar code. If the local library uses UDC, it views the ISBN and the Library of Congress Number merely as attributes. Likewise, to our families, we may be known by our names, but to the Social Security Administration, we are known by our Social Security Numbers (SSNs). How can we keep up with all of this? The answer lies in what happens during the event known as the "birthing of an identity." When an item first receives its identity, the attributes of that item are associated with the identity in some directory or database. For a book, the title, author, publisher, and printing dateas well as all of the various codes for identifying itare associated with the new key identity. Similarly, paperwork for registering a baby's name and applying for an SSN is typically completed immediately after the birth, and this information is recorded in government databases.

In 1999, a group of universitiesM.I.T. (US), Cambridge (UK), Adelaide (Australia), Keio (Japan), Fudan (China), and St. Gallen (Switzerland)formed the Auto-ID Centeralong with industry partners such as Sun Microsystems and Gillette. Their hope was that they could develop standards that would reduce the cost of individual tagsthe component that has long been the primary expense of RFID applicationsand that standard identifiers could facilitate information-sharing among industry partners In October of 2003, EPCglobal, Inc. took over managing the standards, and a group now called the Auto-ID Labs continued as a separate research organization. EPCglobal is a joint venture between European Article Number International (EAN International; now GS1), the Uniform Code Council (UCC; now GS1 US), and industry partners that aims to reproduce the success of EAN.UCC standards for bar codes in the area of RFID. EPCglobal is developing standards that describe the components and architecture for what it calls the "EPCglobal Network." The idea is that this network of compatible tags, readers, and information systems will allow systems over an entire supply chain to integrate, from manufacturers to distributors to retailers. The work of EPCglobal centers around what is known as the Electronic Product Code (EPC).

EAN.UCC bar codes currently track things, so why can't we use the same system in RFID? In fact, we can, and some bar code encoding systems are already being used to make RFID tags. These systems are designed, for the most part, to track product categories, not individual items, but with the addition of a serial number, optical codes such as bar codes and two-dimensional "scatter codes" could track individual items. Item-level tracking and RFID are converging, and the point at which they are converging is the EPC. In EPCglobal's Version 1.1 Tag Data Standard, one general identity type is defined: the General Identifier (GID). Five specific identity types derived from EAN.UCC product codes are also defined. These specific identity types add individual asset references or serial numbers to existing EAN.UCC identifiers such as the Serialized Global Trade Item Number (SGTIN) and the Serial Shipping Container Code (SSCC). (The GID and the five System Identifiers are discussed further in Chapter 4.)

An example GID can be expressed as a uniform resource identifier (URI) in the form:

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

```
urn:epc:id:gid:GeneralManagerNumber.ObjectClass.SerialNumber
```

An example GID might look like this:

```
urn:epc:id:gid:00012345.054322.4208
```

As you can see, the `urn:epc:id:gid:` part of the GID is static and serves as a header that indicates the type of identifier as well as what fields should follow based on the specifications defined by EPC. This header is followed by value fields, the number and length of which are determined by that header. These three segments respectively identify the*General Manager Number*, the *Object Class*, and the *Serial Number* for the GID.

The General Manager Number identifies the organization (usually a company or trading group) responsible for assigning the numbers in the two succeeding fields. The Object Class identifies the product family or type. Finally, the Serial Number identifies the specific instance of the Object Class being tagged. This approach of delegating ownership of a particular range of numbers to a General Manager provides the flexibility to allow organizations to manage their own product numbers without having to submit changes to a central authority while ensuring that multiple organizations will not produce the same identities for different items.

2.2.1.2. Encoding identities on RF tags

Once you have chosen an item identification method, you must consider where and how this identification is going to be encoded onto an RFID tag.*Encoding* is the process by which rules are used to turn a human-readable message into machine-readable code. Each type of identification tag, from bar codes and optical scatter codes to magnetic stripes and RFID tags, has a particular encoding that allows it to represent an identity.

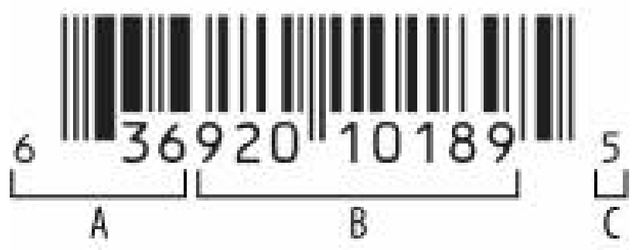
Theoretically, once an identity is established for an item, we could simply write the identity on a label and place it on that item. Other people would then be able to identify the item without difficulty (depending on our handwriting). However, an automated system might have more trouble. Printing the identity in a particular font might make the job easier for the system, but if the identity only needs to be read by the automated system, why not print the identity in a form that it can read very easily?

The ubiquitous bar code is the result of this reasoning. In a bar code, lines of specific thicknesses indicate particular letters or numbers. Bar codes come in different types, and each one has rules that describe how it can be formed from a particular kind of identity. The rules that determine both how we turn numbers and letters into bars and which other numbers and letters we can add to make a valid tag are called the *tag encoding* , or simply the encoding. Thus, the bar code may contain the item's identity, a number to indicate what type of bar code it is, and, in many cases, a number that represents the organization that assigned the identity.Figure 2-4 shows a bar code for the ISBN discussed in the sidebar "Multiple Identities."

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Figure 2-4. An ISBN shown as a bar code



In Figure 2-4, the callouts A, B, and C indicate the sections of the bar code. The section marked A contains the digits 978, a prefix indicating the book industry. Callout B indicates the ISBN number itself. Often bar codes also have a *check digit*, which is a value used by the automated reader to verify that it didn't misread the tag. Callout C marks the check digit, which happens to be 1 in this case. If you have sharp eyes, you may have noticed that the original ISBN ended with a 7. This was the check digit for the original 10 characters of the ISBN, but since we added the industry prefix, we had to recalculate the check digit. Even when we are writing an ISBN as text we must follow the appropriate encoding rules.

In order to select the proper encoding to write an identity to an RFID tag, you must know both the type of identity you want to write and the memory capacity and type of the tag itself. We will discuss EPC encoding in greater detail in Chapter 4 and in Appendix A, but for now, we will use the example GID code introduced previously. As a "pure identity," a GID cannot be written to a tag of any type without some sort of encoding. For our example, let's assume we want to write this to a 96-bit Class I EPC tagthat is, a tag that can store a 96-bit ID and conforms to EPC standards for tags that can be written by end users. First, we take the parts of the GID and arrange them in the right order for the tag, leaving out any fields that are not part of the tag encoding. Fortunately, GIDs contain only relevant fields that are already in the right order. We then add the additional information necessary for the tag type and produce an appropriate URN that can be understood by the reader or event manager. The format of this URN for a GID in a 96-bit tag is:

```
urn:epc:tag:gid-96:FilterValue.GeneralManagerNumber.ObjectClass.SerialNumber
```

For our example GID, the URN would look like this:

```
urn:epc:tag:gid-96:0.00012345.054322.4208
```

Note: The Filter Value indicates the logistical type of whatever is being tracked (e.g., a pallet or carton). We left it as zero for this example. Chapters 4 and 6 discuss encoding and filter values in detail.

If your application talks directly to a reader, you may need to produce these tag-specific URNs. If your application talks to some sort of RFID middleware or to a smart reader that incorporates some

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

elements of RFID middleware, you may be able to use a pure identity URN like the one mentioned previously. The reverse is also true: a reader may give you a tag-specific URN, while middleware can give you a pure identity on a read. However, more and more readers have begun to support the reporting of pure identities.

2.2.2. Attaching RFID Tags

Attaching tags to items is a process that requires time. As with many other processes, attaching tags manually is the most obvious but least efficient method. Bar codes are often applied using print-and-apply machines that press or blow an adhesive tag onto an item as it passes by on an assembly line, and similar devices exist for smart label RFID tags. Many of these devices also write both a bar code and human-readable information to the paper tag before applying it. (Of course, the devices, techniques, and issues involved in attaching other form factor tags to items are different, and almost any style of tag is more durable and less prone to manufacturing defects than a smart label. We'll discuss other types of tags in Chapter 3, but we'll use smart labels as an example here to point out some of the considerations for attaching tags.)

One important consideration when weighing the cost of adopting an automated method for attaching smart labels is the relatively high defect rate of these RFID tags due to their delicacy and sensitivity. Any automated or manual system must take into account the likelihood that many tags on a roll may be defective. Some manufacturers mark any tags on the roll that test as defective, but often they miss some, while others fail after the test. Rolls typically have some unmarked bad tags, which means that tags must be verified before they are printed and applied to the items, and then again after they are applied. This means that if you purchase pre-encoded tags, you cannot associate an item with its tag (known as "birthing the tag") until this second verification is complete (although as tag durability improves by means of new application methods and packaging technologies, the benefi of birthing pre-encoded tags late in the process will diminish). Many informal studies claim an insignificant rate of failure for tags that pass this second verification. Attaching these paper tags to more rigid items seems to help protect the delicate antenna from breaking even under considerable abuse.

2.2.3. Tracking the Movement of Items

Attaching a tag at the point an item is shipped out the door benefits receivers because they can track the movement of the item from the sending dock to their receiving dock. The shippers benefit only in that they have complied with the wishes of the receivers. If an item is tagged earlierfor example, on receipt from the manufacturer or during its productionmore of the supply chain can benefit from knowing how and when the item moved through the chain. It can be valuable to know that a particular shipped item arrived on the dock at 2:05 p.m. Consider how much more valuable it could be to know that a particular item sat in storage bay 11 of the warehouse for six weeks while newer stock shipped from storage bay 10, emptying that bay twice? Or being able to ask your system for the locations of all of the purple widgets made on Friday the thirteenth of May between 9:30 and 10:30 a.m. (the time when the purple widget jabber was misaligned). The cost of this knowledge is the widespread deployment of readers throughout your business in addition to changes to processes to ensure that readers see tags when they arrive at a new step in your process.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Reading RFID Tags

An RFID tag communicates with the outside world by sending and receiving electromagnetic signals. *Active* tags have batteries that power their communications, while *passive* tags reflect back a signal generated by another device known as a *reader*. *Semi-passive* tags have a built-in power source so that they can perform certain functions independently, but they still use the reader's radio signal to power communications.

Just as rules define the process of encoding information on a tag, rules also define the way a tag talks to a reader. This set of rules is called an *air interface* , because the tag and reader interface with each other "over the air." An air interface describes the frequencies the tag and reader use to communicate, as well as a communications protocol. This protocol defines which commands the tag understands and how a tag responds to the reader. The protocol also includes rules for communicating with one tag among many and for inventorying all tags present.

2.2.4. Using RFID Data in Business Applications

At the time of writing, the majority of the work being done with RFID systems is focused on the physical sidethe tags and readers. It is important to ensure that the right tags, readers, and antennas are selected and that they are configured and aligned to achieve the required read rates. You will realize the true benefits of RFID technologies, however, only when you integrate the tracking information from RFID components into your business applications. Most likely, using RFID information will require integration with, and modifications to, your existing applications. Integrating RFID information with your enterprise applications is no different than integrating other data sources Architectural approaches, technologies, and products available for application integration apply equally well to RFID solutions.

2.2.5. Sharing RFID Data B2B

Once companies integrate RFID data internally and adapt their business processes to leverage that data, they will find reasons to share it with their business partners and will begin to provide business to-business services based on the data. Consider an RFID-enabled pharmacy. A pharmacist working in such a pharmacy might be able to retrieve up-to-the minute FDA warnings or other relevant information about a drug just by carrying its container close to an information terminal. The pharmacy's point of sale (POS) system could request a web service provided by the drug manufacturer or some other information intermediary. The drug company could obtain complete tracking information on every drug that it has manufactured and shipped to a particular region by enabling the distributors, shippers, and pharmacies to transfer this information to its tracking systems.

Any of these scenarios assumes the capability of these respective companies to share information. O course, "one-off" solutions exist to enable these B2B workflows, but long term, these are not the most cost-efficient, the most flexible, or the quickest ways to share information. Sharing information and workflows across enterprise boundaries is not a new phenomena. As we will see later in this

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

chapter, a host of options are available for accomplishing it. In addition to the technology options that are generally available, industry bodies such as EPCglobal are proposing mechanisms for standardizing how we exchange this information. Chapter 8 describes some of these options in more detail.

2.2.6. Self-Organization of Intelligent Devices

As an increasing number of devices are connected to the Internet, the task of provisioning, configuring, monitoring, and managing them poses an ever-growing challenge. An average retail outlet might a dozen servers connected to a couple of dozen POS terminals, at most. However, when the same retail outlet adds RFID-enabled smart shelves and POS terminals, a couple of hundred antennas and readers could be connected to the infrastructure fabric. RFID middleware standards such as application-level events (described later in this chapter) help decouple the enterprise applications from readers and antennas, but configuring those components to work with RFID middleware can be a time-consuming task. Technologies such as Jini and mesh networks, and old standbys such as SMTP provide dynamic configuration and self-healing characteristics so that RFID middleware can adapt to changes in the physical configuration of readers and other sensors. Chapter 9 describes these and other RFID system management issues and options.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.3. RFID System Components

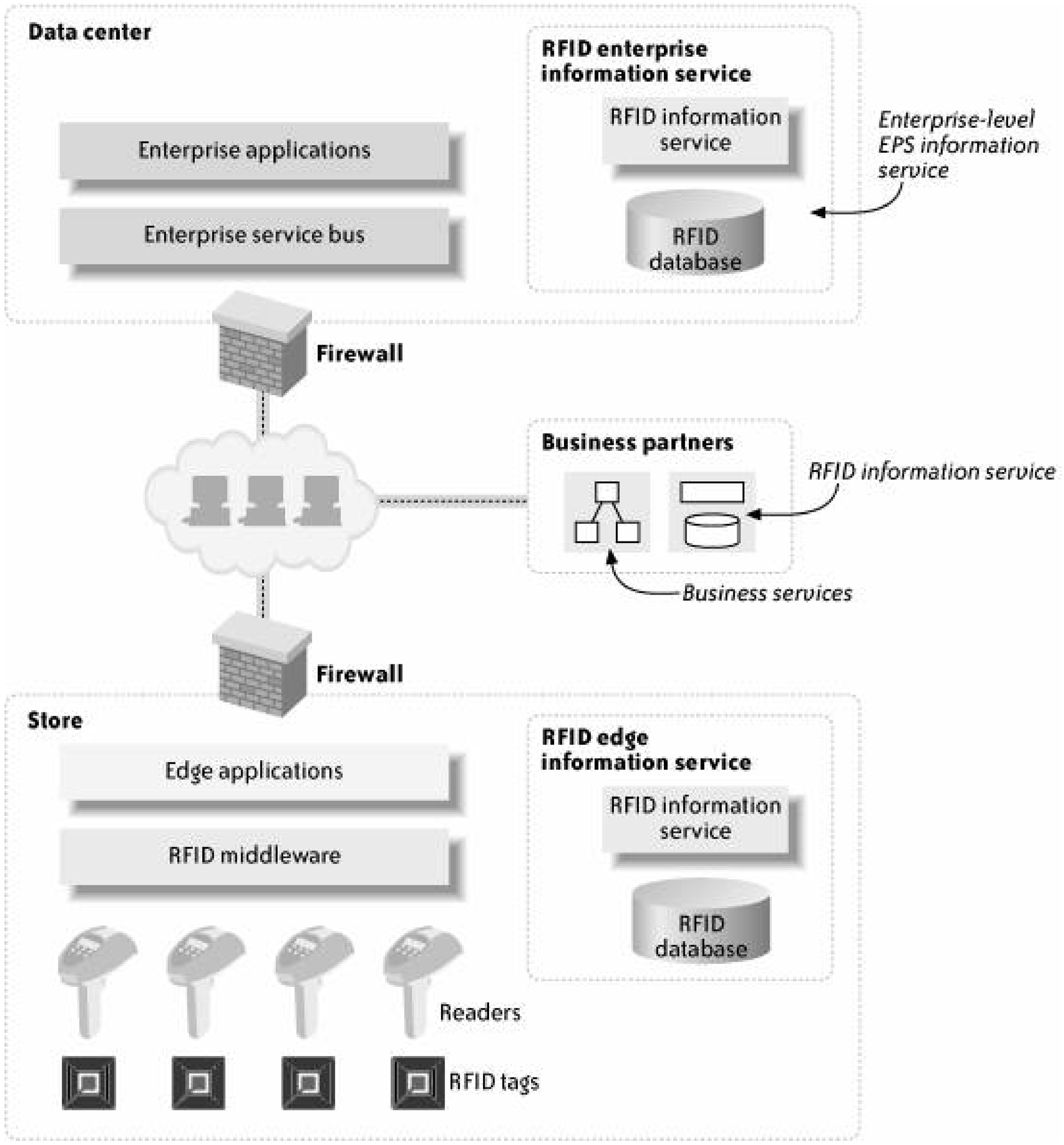
Figure 2-5 shows the primary components of an RFID system. We will explain each of these components in detail soon, but first let's look at the big picture, starting with the components typically present at the edges.

Figure 2-5 shows the typical components found in a retail store. In the bottom-left corner of the diagram, there is a set of RFID tags that represent the tagged merchandise. The store also has readers stationed within the shelves and at the checkout lanes. These readers may read tags hundreds or even thousands of times per minute, but most of these reads will not be interesting to our application. The readers must also be configured and managed and must know how to work together to cover blind spots should a reader fail. The box marked RFID middleware represents one or more software modules that handle these responsibilities. The box marked Edge applications represents any enterprise applications that have components running inside the storefor instance, POS system components. The box marked RFID information service represents a mechanism to store RFID events and related data at the edge. As you can see, we are showing similar RFID information service boxes in the enterprise's data center and in its business partners' data center. This is because RFID information is stored at various points in the infrastructure: at the edges, within the data center, and with business partners. We will provide more detail on what this information looks like later in this section. For an in-depth discussion of RFID information networks, seeChapter 8.

Figure 2-5. RFID system components

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



The two other components shown inside the enterprise data center in Figure 2-5 are the enterprise service bus and enterprise applications. The enterprise service bus is any mechanism that your company may have selected for application integration. Standards-based products that facilitate this are now available. Enterprise applications are any applications that are clients of, or are otherwise affected by, RFID data in your enterprise.

With this basic introduction complete, let's look at the primary RFID components in greater detail.

2.3.1. Tags

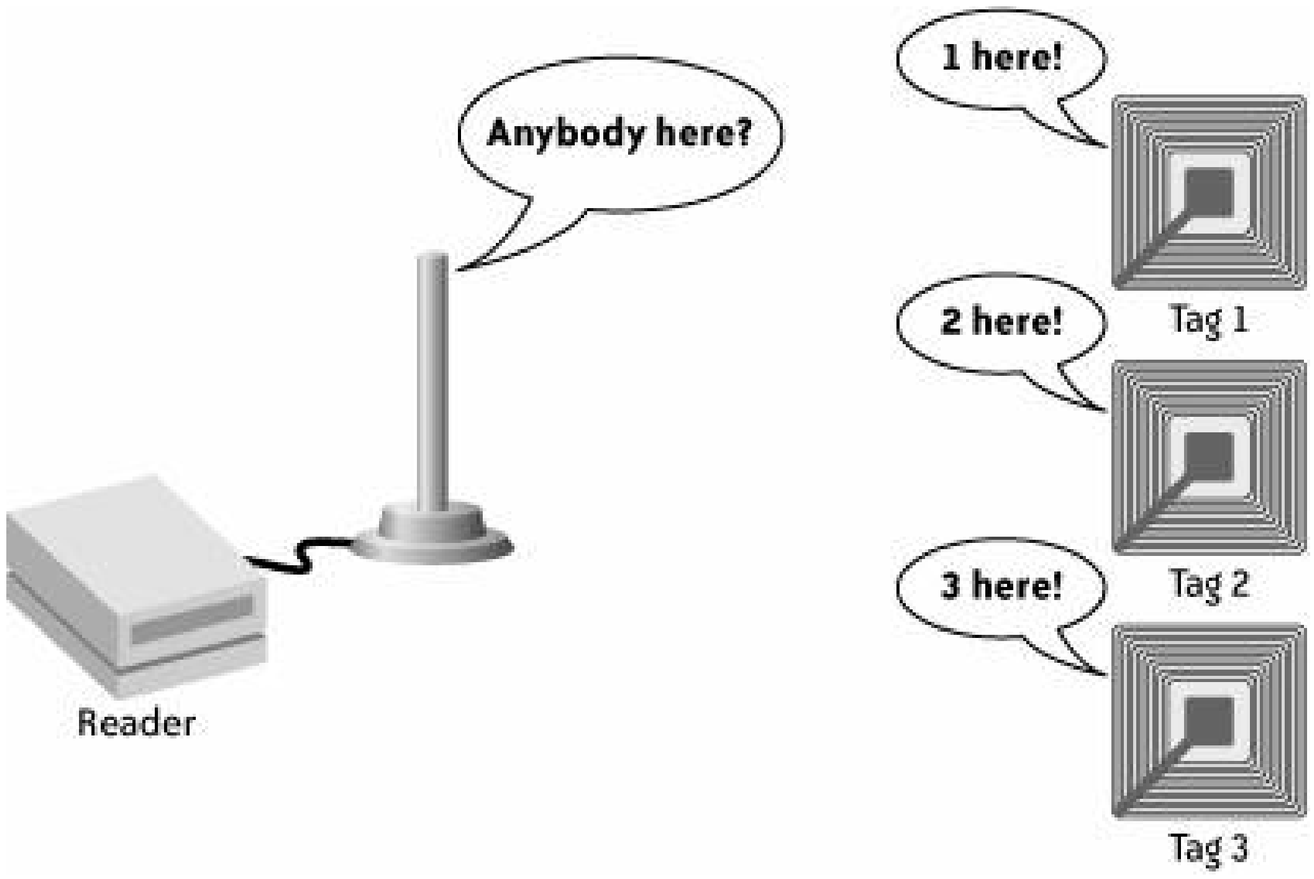
downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

As we discussed in Chapter 1, the term "RFID" is typically used to describe systems wherein a base station of some sort (a reader) is able to recognize another electronic device (a tag) using one of several possible wireless transmission mechanisms. These mechanisms may include microwave but not infrared or visible light systems. Since a reader is able to identify a particular tag, the system car claim to have identified the object to which that tag is attached. Tags may be housed in small plastic buttons, glass capsules, paper labels, or even metal boxes. They may be glued to a package, embedded in a person or an animal, clamped to a garment, or hidden in the head of a key.

To understand how an RFID tag notifies a reader about its presence and identity, consider the simple scenario depicted in Figure 2-6. In this figure, the RFID reader transmits radio signals at a preset frequency and interval (usually hundreds of times every second). Any radio frequency tagsthat are in the range of this reader will pick up its transmission because each has a built-in antenna that is capable of listening to radio signals at a preset frequency (as we'll see inChapter 3, the size and shape of the antenna determine what frequencies it will pick up). The tags use energy from the reader's signal to reflect this signal back. Tags may modulate the signal to send information, such as an ID number, back to the reader.

Figure 2-6. Communication between RFID tags and a reader



Different kinds of tags and readers fit different applications and environments. Deciding which tag or reader to use involves determining the best fit for your needs. The tag you choose can greatly affect the cost of the system. Readers also have a wide range of prices and features.

Important characteristics of RFID tags include the following:

Packaging

As noted previously, tags may be housed in PVC buttons, glass vials, paper labels, or plastic cards. They may be combined with jewelry, hung from key chains, or built into the heads of

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

keys. The DIN/ISO 69873 standard defines a standard for tags that may be inserted into holes built into machining tools. Some tags used in auto assembly lines are designed and packaged to survive the intense heat of paint-drying chambers. In short, the ways in which tags can be packaged are remarkably varied. Figure 2-7 shows two tags: the quick-payment tag in the key fob is exposed to show the antenna and chip, and the head of the key also contains a vehicle immobilization chip.

Figure 2-7. Two RFID tags (image courtesy of Texas Instruments)

Coupling

Coupling refers to the means by which the reader and tag communicate. Different coupling methods allow for different strengths and weaknesses. Coupling choices especially affect the range of communications, the price of tags, and the conditions that might cause interference.

Power

Many RFID tags use some sort of "passive" system, wherein an electromagnetic field or a pulse of radio frequency energy emitted by the reader powers the tag. In other ("active") tags, a battery powers a microchip or additional sensors. However, active tagsstill use power from the reader for communications. The third type of tag is the so-called "two-way tag," which powers its own communications and may even be capable of communicating directly with other tags without a reader.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Information storage capacity

Tags provide varying amounts of storage capacity. Read-only tags are set to store a particular value at the factory. Users can set a value to write-once tags one time, while the value stored by a write-many tag may be changed many times. Some tags are also able to gather new information, such as temperature or pressure readings, on their own. Tags range in storage capacity from the 1-bit tags used for theft prevention to tags used in auto assembly lines, which may store thousands of bytes.

Standards compliance

Many types of RFID systems conform to particular national and international standards. A developer working on "an ISO 11785 system" is actually working on a system compatible with that standard. Some standards, such as the Class system used by EPCglobal, specify frequencies, coupling type, information storage capacity, and more.

2.3.1.1. Selecting tags

Many considerations are involved in selecting tags. They include the following:

Required read range

Active tags provide a longer read range than passive tags. For retail applications, the read ranges offered by passive tags are usually sufficient.

Material and packaging

Different materials have differing RF characteristics. For example, liquids may impede the flow of radio waves. Metal containers also pose interference challenges to the readers.

Form factors

RFID tags come in different sizes. The form factor for the tags used for individual products will depend on the packaging used for these products.

Standards compliance

It is important to consider whether most of the readers that are generally available will understand the RF tag you select. EPCglobal and the International Standards Organization (ISO) provide standards for communications between RFID tags and readers.

Cost

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

The cost of a single RFID tag plays an important role in its selection because most applications use many tags.

Chapters 3 and 4 provide more information about RFID tags and how they work.

2.3.2. Readers

RFID readers, also called *interrogators*, are used to recognize the presence of nearby RFID tags. An RFID reader transmits RF energy through one or more antennas. An antenna in a nearby tag picks up this energy, and the tag then converts it into electrical energy via induction. This electrical energy is sufficient to power the semiconductor chip attached to the tag antenna, which stores the tag's identity. The tag then sends the identity back to the reader by raising and lowering the resistance of the antenna in a kind of Morse code. This is only one scenario, and different tags can work in slightly different ways, but this is typical of the way readers and tags interact. We describe methods of communicating between tags and readers in more detail in Chapter 3.

Readers come in many shapes and sizes and can be found in stationary, as well as portable, handheld varieties. For our purposes, you can think of readers as the point at which tags connect to the network. Figure 2-8 shows how a reader fits between tags and the outside world and illustrates the components of a reader.

Figure 2-8. Parts of a reader

A reader is a system comprised of four distinct subsystems:

Reader API

The reader API is the application programming interface that allows programs to register for and capture RFID tag read events. It also provides capabilities to configure, monitor, and otherwise manage the reader.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Communications

Readers are edge devices, and like any other RFID device, they are connected to the overall edge network. The communications component handles the networking functions.

Event management

When a reader sees a tag, we call this an *observation*. An observation that differs from previous observations is called an *event*. The analysis of observations is called *event filtering* . Event management defines what kinds of observations are considered events and determines which events are interesting enough to merit being either put in a report or sent immediately to an external application on the network.

Antenna subsystem

The antenna subsystem consists of one or more antennasand the supporting interfaces and logic that enable RFID readers to interrogate RFID tags.

2.3.2.1. Selecting readers

Your selection of a reader is constrained by the tag you select. Some readers are compatible with certain types of tags but not with others. Readers, as powerful radio transmitters, must also conform to local regulations concerning frequency, power level, and duty cycles (how often the reader is actually transmitting). When selecting a reader, also pay close attention to the physical environment in which it will be operated. The reader must be small enough to be out of the way of personnel and equipment and tough enough to withstand exposure to dust, shock, humidity, or temperature extremes. Finally, an often-overlooked consideration in selecting a reader is how well it will cooperate with your current IT monitoring and management tools.

2.3.3. RFID Middleware

Choosing the right tags and readers and determining where to put the antennas is only the first step in building a working RFID system, because identifying items is only the first step in managing them. The capability to read millions of tags as they move through the supply chain and the need to tie tag codes to meaningful information will generate large amounts of data with complex interrelationships. One of the primary benefits of using RFID middlewareis that it standardizes ways of dealing with the flood of information these tiny tags produce. In addition to event filtering, you also need a mechanism to encapsulate the applications so as to prevent them from knowing the details of the physical infrastructure (readers, sensors, and their configurations). Ideally, you want a standards-based, application-level interface to the RFID infrastructure that applications can use to request meaningful RFID observations.

Figure 2-9 shows the primary components of RFID middleware.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Figure 2-9. Components of RFID middleware

2.3.3.1. Motivations for using RFID middleware

There are three primary motivations for using RFID middleware: providing connectivity with readers (via the reader adapter), processing raw RFID observations for consumption by applications (via the event manager), and providing an application-level interface to manage readers and capture filtered RFID events.

Let's start with a simple approach to processing data collected by the readers, as shown in Figure 2-10. The sheer inadequacy of this approach will help to illustrate several points. In this figure, we expose the output of the readers directly to our applications. Today's readers provide minimal event filtering. From the perspective of the enterprise applications, using this approach for anything but a very small deployment or a proof of concept would be akin to trying to shove a bowling ball through a keyhole.

Figure 2-10. Exposing applications directly to RFID readers

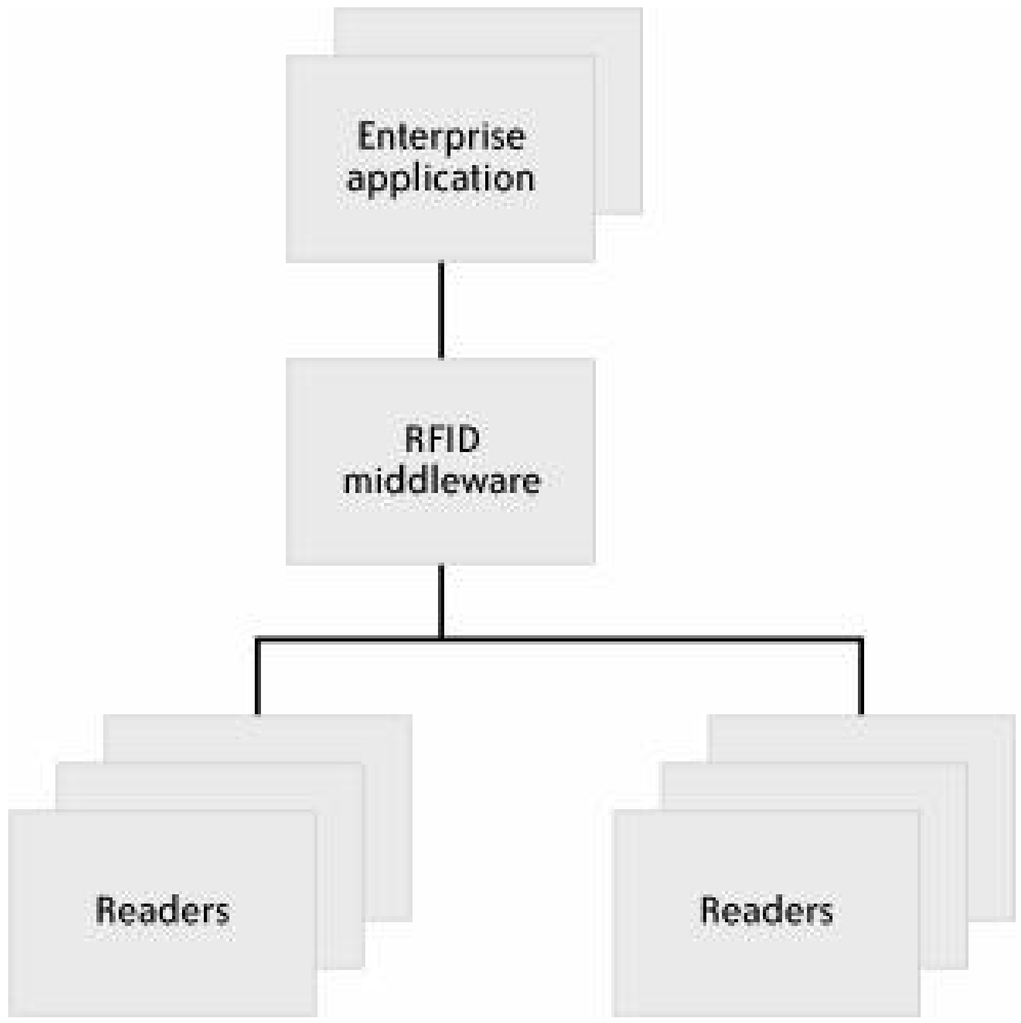
In Figure 2-11, event-processing middleware is introduced between the readers and the applications. This approach is suitable for small-scale deployments using the capabilities provided by application

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

integration products.

Figure 2-11. Using RFID middleware



2.3.3.2. The reader adapter

Several dozen types of RFID readers are available in the marketplace today, and each has its own proprietary interface. It would be impractical to expect application developers to learn different types of reader interfaces. Reader interfaces, as well as data access and device management capabilities, differ widely, so you should try to use middleware that shields you from having to learn the idiosyncrasies of individual readers. The reader adapter layer encapsulates the proprietary reader interfaces so that they don't come in contact with the application developers.

2.3.3.3. The event manager

A fully deployed, RFID-enabled supply chain for an enterprise could have hundreds or even thousand of readers scanning at hundreds of reads per minute. Most of these observations would be too fine-grained to be meaningful to applications, so you want to encapsulate the reader interface to keep the applications from being bombarded by raw data. Enterprises will thus need to deploy special-purpose RFID middleware near the edges of their IT infrastructures.

Readers are less than 100 percent accurate at picking up tags in their vicinity. Suppose 100 items were present near a reader that was set to scan a couple of hundred times a minute. Chances are that each scan of the reader would pick up anywhere from 80 to 99 of these items. The fact that item

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2, for instance, is sensed by the reader 80 percent of the time across multiple read cycles would be a good indication that it is still around. However, this scenario demonstrates why RFID scans from readers are considered "raw" and require further processing to derive meaningful business events. Suppose this reader was positioned for a smart shelf system. Would you want to expose unfiltered reader observations to your enterprise applications? For all but trivial applications, you would probably want these observations to be processed further before sending them downstream. One example is a "smoothing" filter that can take a series of reader scans and aggregate the results in a way that accounts for the less-than-perfect read accuracy of some of the RFID technologies.

If you think exposing the observations of one reader directly to your enterprise applications would be a bad idea, consider that a typical RFID implementation incorporates at least a few dozen RFID readers. Imagine the amount of data that would be generated by all of these readers and how much filtering would be required!

An RFID event manager aggregates raw RF read data from multiple data sources (such as readers) and consolidates and filters it based on previously configured application-level event filters. Most event managers then feed the filtered data to the backend systems.

Let's take a closer look at how an event manager would be used in a smart shelf scenario. Imagine that for a certain application, each reader scans shelves around 10 times a minute. Each scan returns a set of observations, and each observation might look like the following:

```
Reader Observation
timestamp, reader code, antenna code, RF tag id, signal strength
```

To get a sense of the volume of data generated by the smart shelf readers, consider the following example. An electronics retailer called Nirvana Electronics wants to implement a smart shelf system. The store averages 25 items per shelf and 4 shelves per rack, for an average of 100 items per rack. Each of Nirvana's 10 stores has 20 isles, each containing 20 racks (10 on each side). So, a Nirvana store has 400 racks, which means the average total inventory would be around 40,000 items.Table 2-1 recaps these totals. (Aren't you glad we didn't start with a grocery store or warehouse example?)

Table 2-1. Average inventory (items)

Location	Average inventory (items)
1 shelf	25
1 rack (4 shelves)	100
1 store (400 racks)	40,000
All 10 stores	400,000

Now consider the volume of data generated as the RFID system reads this data:

- Each scan returns observations that contain information on all the items that were recognized t

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

be on the targeted shelves.

- 25 items per shelf x 4 shelves per rack x 10 scans per minute = 1,000 observations per minute per rack.
- 1,000 items read per minute x 400 racks = 400,000 observations per minute.
- 400,000 items read per minute x 60 minutes per hour = 2,400,000 observations per hour.
- Let's say the store is open 10 hours a day. 10 hours x 2,400,000 items read per hour = 24,000,000 observations per day, per store.
- 10 stores = 240,000,000 observations between all the stores.

This data is summarized in Table 2-2.

Table 2-2. Average number of observations

Location	Number of observations
1 rack per minute	1,000
1 store per minute	400,000
1 store per day	24,000,000
All stores per day	240,000,000

Wow! That's some serious volume, and we haven't even counted the observations that would be coming from the checkout lanesbut you get the point. Processing all these observations will require some serious architectural planning. But before we start drawing boxes and arrows, we need to understand a little more about the quality and relevance of the information contained in the observations.

If you simply pass all these observations to the enterprise applications running in your data centers, you will not only overwhelm those applications but also possibly push the networks and other infrastructure elements to the limit. Also, the downstream applications will likely find the raw RF reader observations uninteresting. For instance, say a customer picks up a DVD from a shelf and puts it on another shelf. This would generate a series of observations from the readers on the respective shelves. However, if we look at this in the context of Nirvana's order management system, this information would be uninteresting because the store inventory has not changed. In fact, even if the customer buys that DVD, the order management system might not care about this event if the store inventory for that DVD does not fall below a certain level, as established by the store's business rules.

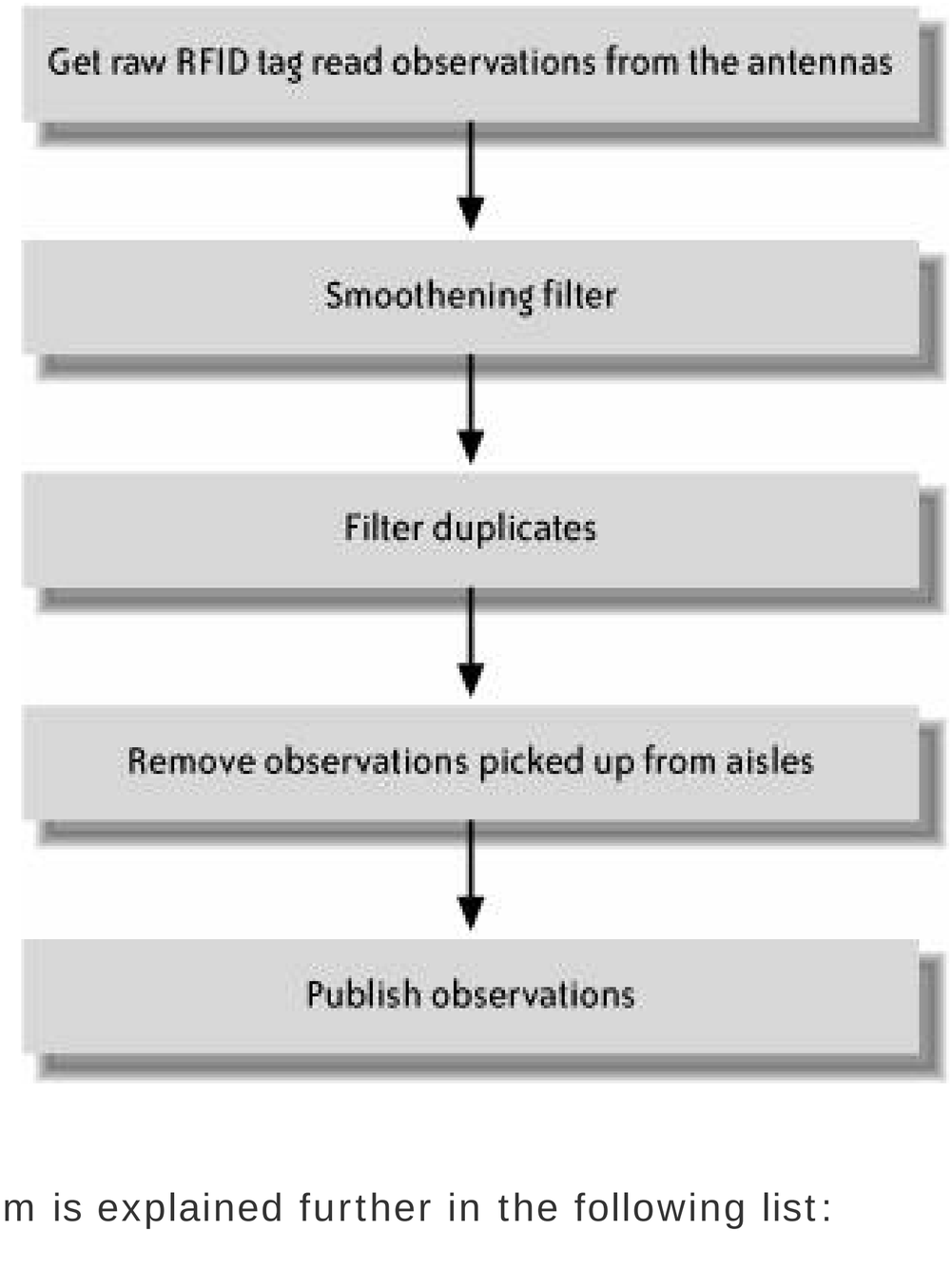
This example illustrates the need for mechanisms to aggregate observations over time and across multiple readers. It also illustrates the need to filter, consolidate, and transform raw RF reader observations. This is why your RFID system needs middleware that runs inside the edges of your enterprise data centers. This way, only the observations that are important to your corporate applications are sent to those applications. The rest of the data is filtered out by the RFID middleware.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

So, what sorts of things must be filtered out? Since the antennas are in close proximity (two per rack), their read ranges will overlap. Therefore, the observations coming from them will need to be filtered to remove duplicate reads. Also, as individual scans tend to be less than 100 percent accurate, these observations need to be aggregated over several read cycles to smooth them out. (As explained in Chapter 5, the inexact nature of reads is the product of several factors, including the nature of radio frequency transmissions, alignments of the RFID labels, obstructions, and environmental factors.) As customers move along the aisles with products in their hands or in shopping carts, the nearby readers might pick up these items, too. We should filter out any such spurious observations to avoid sending a flood of inaccurate observations to the inventory control system. Figure 2-12 illustrates a filtering and smoothing system devised to address typical scenarios experienced in retail stores.

Figure 2-12. Event filters



Each element of the system is explained further in the following list:

Get raw observations (EPC reads)

The readers pick up raw event data (observations).

Smoothen observations

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Today's readers are less than 100 percent accurate in picking up RF tags with a single scan, so we analyze the raw reader data over several scans and pick up observations based on averages. For example, if 70 percent of observations tell us that an RCA cable box is on a counter, we accept that information.

Filter out duplicates

Duplicate observations caused when more than one antenna identifies an item should be filtered based on relative signal strengths.

Filter out observations from aisles

Observations of items moving through the aisles will generally have lower signal strengths and be transient in nature. They should be filtered out.

Publish observations

After the necessary filtering, our data is ready to communicate downstream.

RFID readers already provide some filtering capabilities, and as readers become smarter, they will take over more and more of the filtering tasks themselves. So why do we need to do additional filtering in the middleware? As we've seen, some filtering requires information from multiple readers, other sensors, or even other systems (such as the inventory system, in the case of comparing stock levels). This higher-level filtering will always need to occur in a system located above the readers in the event hierarchy.

2.3.3.4. The application-level interface

The application-level interface is the top layer in the RFID middleware stack. Its primary purpose is t provide a standardized mechanism that enables applications to register for and receive filtered RFID events from a set of readers. In addition to this, the application-level interface also provides a standard API for configuring, monitoring, and managing RFID middleware and the readers and sensors it controls. Many RFID middleware vendors provide proprietary interfaces designed for these purposes. More recently, EPCglobal has published the Application Level Events (ALE) specification for standardizing the event management portion of RFID functionality. We cover the ALE specification in greater detail in Chapter 7.

Finally, note that RFID middleware comes in various shapes and sizes. What we just presented was a logical breakdown of what middleware is and what it does. In practice, you might find that various middleware vendors provide modules that you can deploy on particular reader types for specific type of applications.

2.3.3.5. Relevant EPCglobal standards

There are a number of EPCglobal standards that are important to RFID middleware. Two of these standards are:

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Reader protocols

There are efforts underway within EPCglobal to create standards for the reader and printer network protocols. (We cover reader protocols in Chapter 6.) RFID middleware vendors plan to support these standard protocols as they become available. We expect that within the foreseeable future, custom drivers will continue to be required for most device types.

Application Level Events (ALE)

ALE is an upcoming EPCglobal standard that provides a declarative interface so that applications can interact with the filtering and collecting engine. The ALE standard supports request/response and pub/sub-style programming. This approach has proven to be vastly simpler than the early real-time workflow model of Savant, and it has replaced all Savant efforts within EPCglobal.

2.3.4. The RFID Service Bus

An enterprise service bus (ESB) is a distributed integration platform designed for application connectivity, data transformation, guaranteed transactions, and messaging. An RFID service bus is a type of ESB used to integrate applications using RFID data. Depending on the implementation you choose, ESB products offer web services, messaging, business process orchestration, and other capabilities. Generally, ESBs orchestrate business processes across application, or sometimes enterprise, boundaries.

As we know, RFID systems will not function in a vacuum. They have to coexist with and extend the capabilities of enterprise applications such as warehouse management systems (WMSs), enterprise resource planning (ERP) systems, or point of sale systems. Do not underestimate the importance of adapting these applications and systems to utilize RFID-enabled data. Extending the capabilities of your existing applications is an important activity that must be carefully thought out before you begin any RFID adoption process.

Many companies use ESB architectures, which are also referred to as *integration servers* , to enable disparate applications to collaborate with each other. At a minimum, such software provides interfaces that enable you to integrate various systems that use differing technologies. It provides adapters to understand input coming from these systems, and then it translates that input into a common format (usually XML). Integration servers also provide capabilities that enable you to define and execute business processes. It is possible that the RFID event manager product you use provides rudimentary integration server capabilities.

The primary purpose of an RFID service bus is integrating application-level events captured by event managers into workflows that occur at the enterprise's edges (edge workflows). Specific vendor implementations may vary (some RFID middleware vendors provide scaled-down versions of ESBs), but an RFID service bus is essentially an integration server that runs the relevant edge workflows and provides integration capabilities with modules of enterprise applications such as POS or WMS packages. The RFID service bus also integrates into the ESB and can be configured to supply specific events and observations. Because a typical ESB product can be complex and expensive, it's common to see simpler application server-based custom implementations designed to serve this need.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.3.4.1. The RFID information service

It is important to realize that an EPC, or for that matter, any other item identification system, is just a unique identifier and by itself does not provide any product details. EPCglobal envisions that collaborating businesses and industries will set up a network of EPC Information Service (EPCIS) servers to provide an on-demand repository of information related to individual EPCs. Information made available using the EPCIS servers could include the last observed location of an item carrying an EPC (based on an RF reader observation), as well as pricing information and product manuals, if appropriate. While it will take years to realize the vision of an industry-wide network of EPCIS servers, a necessary intermediate step would be to implement a database that maps the identifiers used in RFID tracking data to a richer set of information relevant to business applications. An example might be mapping product reference information to its identifier. Even small and non-EPC applications will need to tie the EPC codes to product tracking information (or other reference data related to the product type or instance) and so may need to use and update information managed by an RFID information service.

EPCglobal intends to leverage existing and evolving data sources for the RFID information service. For example, the Serialized Global Trade Item Number (SGTIN) has been proposed as the unique EPC identifier for the consumer packaged goods and retail industries.

2.3.4.2. The RFID information network

As RFID-tagged products move through the supply chain, various participants in the supply chain will need standards-based means to share their tracking information and to get reference information on products based on their product tags (EPCs). EPCglobal envisions a mesh of networked B2B EPCIS systems collaborating to provide a comprehensive reference source for EPCs. The EPCglobal Network is a vision coupled with an evolving set of standards that aims to provide a standard framework for product information exchange. Centered around EPC RFID technologies and the existing Internet infrastructure, the EPCglobal Network will offer the potential for increased efficiency and accuracy in tracking products between trading partners. Since the details of the EPCglobal Network are still emerging, we'll focus here on the principal features that might characterize an RFID information network.

Note: Note that some of the discussion in this book concerning the RFID information network is conceptual in nature. Where possible, we will draw the distinction between the EPCglobal Network specification and our conceptual model.

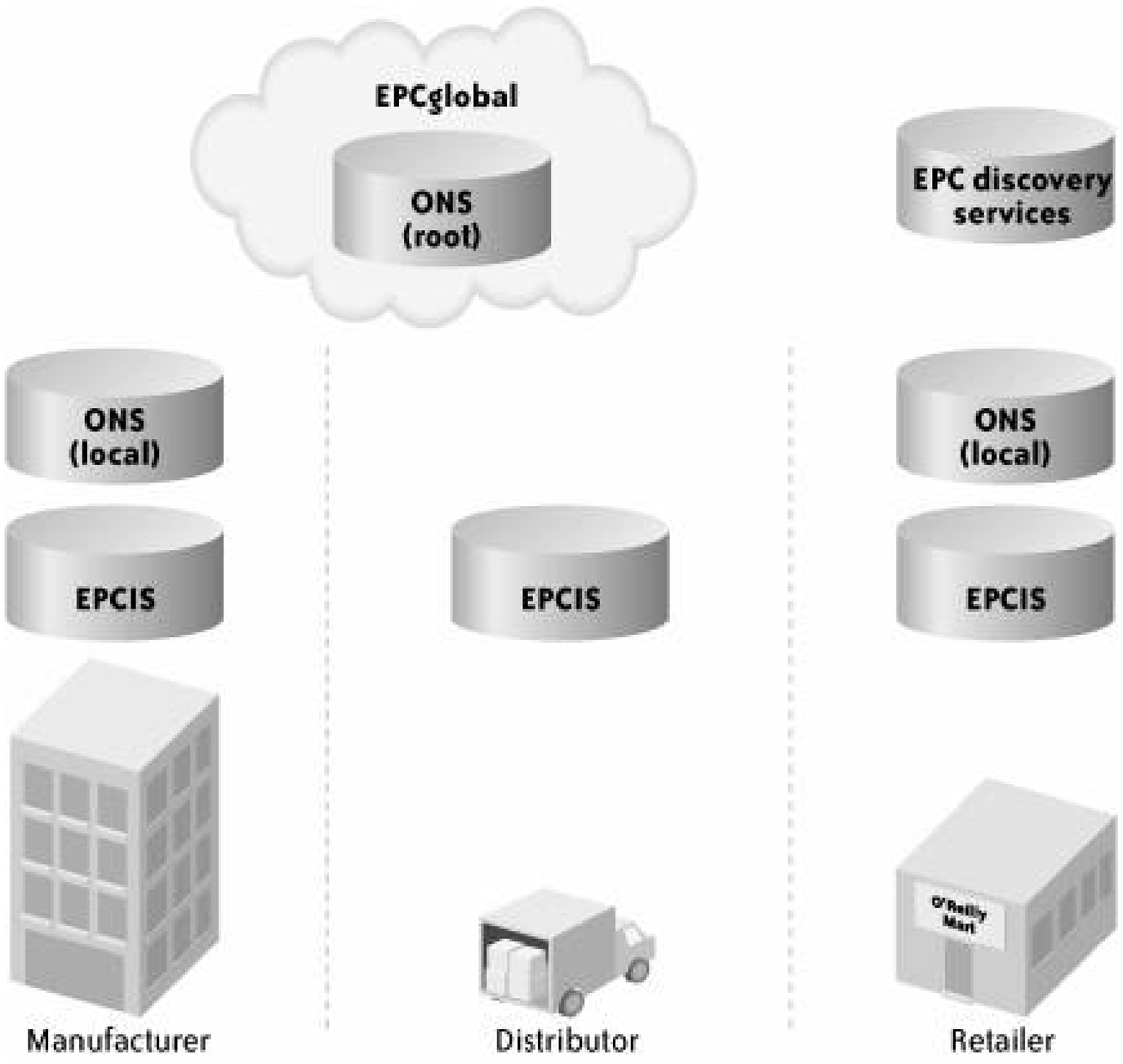
Figure 2-13 shows one vision of the RFID information network. As a product flows from its manufacturer to a distributor and then to a retailer, each of these parties captures its tracking information. How, you might ask, is the tracking information the distributor captures tied to the tracking information originally supplied by the manufacturer and then supplemented by the retailer? This is where the Object Naming Service (ONS) and the EPCIS come in. As shown inFigure 2-13, the manufacturer, distributor, and retailer maintain local EPCIS instances. The local EPCIS instances are tasked to maintain the EPC tracking information generated by the constituent, and they also help interface with other EPCIS instances that maintain relevant and desired EPC information. An EPCIS server can host, as well as facilitate access to, serialized product information generated by EPC-

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

tagged products. This means a distributor's EPCIS server can be configured to interface with a manufacturer's EPCIS server to access and/or aggregate related EPC data.

Figure 2-13. The RFID information network (image courtesy of VeriSign)



The ONS is a centralized, authoritative directory that routes requests for information about particular EPCs; it follows the architectural underpinnings of the Domain Name System (DNS). VeriSign maintains the root ONS server for the EPCglobal Network. The EPC Discovery Service is an authoritative registry of every EPCIS server that has information about a particular product. The EPCIS thus works with the ONS and the EPC Discovery Service to realize the vision of a distributed information network.

The RFID information network provides five principal services:

Assigning unique identities

Tracking items is not possible without the capacity to uniquely identify them. This is where the EPC comes in. Like the UPC or bar code, the EPC is an identification system for products. However, unlike the UPC, the EPC allows item-level tracking by identifying not only the manufacturer and product type, but also the serial number.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Detecting and identifying items

The identification system consists of EPC tags and readers. Each tag contains a microchip attached to an antenna. The EPC is stored on this tag. At the most basic level, the EPC provides a coding scheme for RFID tags that helps identify an item's manufacturer, product category, and unique serial number. The tag is applied to an item either during the manufacturing process or somewhere down the supply chain. The EPC readers use radio frequency waves to interrogate the EPC tags, which then communicate their EPCs back to the readers. EPC readers deliver information to local business information systems using EPC middleware.

Collecting and filtering events

EPC middleware provides specifications for services that enable data exchange between EPC readers and business information systems. Much of the raw EPC observations coming from the readers would be noise to the enterprise applications. Event management middleware is needed to facilitate the collection of observations from the readers and to filter and group them for consumption by the applications.

Storing and querying events

The EPCIS enables users to exchange data with trading partners based on EPCs. The EPCIS specification (not available at the time of writing) aims to provide standards to allow disparate applications to share EPC data. The specification provides standards for capturing and querying EPC data between trading partners.

Locating EPC information

To enable trading partners to share EPC observations, it is necessary to provide lookup or discovery services that can locate repositories for the required EPC data. For authoritative data from the manufacturer, this role is handled by the ONS. The ONS provides the means to look up service resources that provide further information on an EPC. The ONS is very similar to, and is in fact implemented on top of, the DNS technology that handles the billions of domain name queries on the Internet today.

As we noted earlier, the EPCglobal Network specifications are still being defined by the EPCglobal architectural committee and will be published under the EPCIS specification. EPCglobal has already published the standards for the ONS. In Chapter 8, we'll look at some of the alternatives for sharing RFID information across enterprise boundaries.

Before we move on to the systemic quality considerations for an RFID system, remember that successful implementations of business-to-business information-sharing networks are only partially dependent on technology. Data ownership and liability, business processes, security, and legal considerations all need to be addressed before you begin setting up such a network. We anticipate that the technological underpinnings of the EPCglobal Network will initially be developed around two-party, and then later multiparty, vendor relationships before any sort of industry-wide or global EPC information network is successfully implemented.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.4. Systemic Quality Considerations

As we know, a system's requirements come in two flavors: functional and nonfunctional. The functional requirements define what the system does, while nonfunctional requirements involve systemic qualities such as privacy and security, performance, scalability, manageability, extensibility, and maintainability. RFID systems demand vastly increased data-processing capabilities at the edges of corporate environments. The volume of raw RFID data that is captured and available for filtering, coupled with the need to automatically monitor and manage RFID devices (such as antennas and readers), requires a level of sophistication that was generally absent from the bar code-type systems that RFID systems replace.

Some of the most important systemic qualities for RFID systems are covered in the following sections.

2.4.1. Privacy and Security

Privacy and security considerations are a core part of the design and architecture of any RFID solution. Ironically, even though RFID solutions are sometimes offered as enhancements to security, the technology itself presents certain vulnerabilities. As in any enterprise system, security considerations for an RFID systemensuring the authenticity of the information stored on the tags themselves, securing the transmission of information between tags and readers, and ensuring overal application and infrastructure securitypermeate the various layers of its architecture.

Physical security measures involve efforts to prevent both corruption of tag data and interception of communication between tags and readers. For instance, a malicious tag reader that is not part of an RFID system can attempt to read tags in its vicinity. Wireless security focuses on securing the communication pipe between readers and tags since this communication can be intercepted using wireless sniffing and spoofing devices. Any security-sensitive application must carefully weigh the risks of both interception or alteration of tag-to-reader and reader-to-tag communications.

Of course, security considerations vary depending on business needs. For example, using an RFID-enabled credit card to purchase goods can be done securely as long as the fulfillment systems on the backend are prepared to handle and detect fraudulent data (as credit card systems are), but a ticketing system that depends on the authenticity of every tag read might be more easily compromised.

In addition to security considerations, individual privacy is a very important consideration in implementing RFID systems and introducing them to operators and users. We have dedicated Chapter 10 to a discussion of privacy and security.

2.4.2. Performance

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Performance is measured in terms of time taken to perform a unit of activity. Depending on the layer of the RFID system at which you're working, the performance considerations will vary. For instance, for the physical layer, the time required to recognize a tag is an important performance consideration. As with the performance metrics for other systems, this measure would be taken at peak throughput and averaged over a period of time appropriate to system usage. We might be tempted to state requirements in the form of scans per second, such as: "Each loading dock must recognize a newly arrived tag within 10 seconds on average at peak periods and 2 seconds on average at nonpeak periods." Because of the simultaneity of RFID inventories, however, it is much more common (and useful) to state RFID performance requirements in the throughput form, such as "The loading dock must recognize 96 tags per minute on average during peak load periods and 23 tags per minute on average during nonpeak periods."

As you move up to the RFID event manager and other application-level components, the performance characteristics you'll be most concerned with will be based on the time taken to perform a unit of application-level activity. For a tag and ship application, you could define this as the time taken to retrieve an EPC and print it. When building any application, performance considerations and requirements for various uses of the system should be defined and tested early on.

2.4.3. Scalability

Scalability has to do with how small a system may start and how quickly it must be ready to grow. For RFID systems, scalability involves much more than "CPU headroom." RFID scalability requirements are often stated like so: "The shipping system should be able to handle 12 loading docks initially and be able to scale to 60 loading docks in 3 years with no more than a 2-week cycle t install a group of 4 new loading docks and bring them online." However, one of the qualities of RFID systems that may be especially unfamiliar to IT personnel is the need to plan for site surveys and RF tuning as well as the installation of new equipment and software. The RF environment of an area 10 feet from the last install cannot be considered "just the same" as before the install, and there is no guarantee that an earlier site survey of that area is still valid if, for instance, the forklift for certain docks now parks in a different spot when not servicing a load. Once readers are in place, changes to the RF environment can usually be spotted quickly due to changes in performance, but during an install, the environment offers enough variables without adding unexpected signal absorption and reflection.

2.4.4. Manageability

RFID information servers are typically housed in a data center or server room. Operations staff can monitor and manage the servers using procedures developed for any other application server. Event managers and readers are found in loading bays and warehouses, in truck trailers and train yards, and even in corrugated tin shacks in the middle of fields. A reader may be just as complex as a server, requiring software or firmware upgrades, monitoring, and management. An event manager is likely to be located on server hardware but deployed close to the readers. These are edge systems and must be managed as such. One of the primary management differences between edge systems and other systems is that they are likely to be less accessible and more prone to environmental extremes such as heat, dust, and vibration. Furthermore, because a reader has a fixed location, if it should fail, another reader in a different location cannot be assigned to take over unless it is physically close to the failed reader. These characteristics of edge systems require both new practices and training for operations personnel and management software capable of managing these

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

relationships. Manageability requirements are often stated in terms of human resource utilizationfor example, "The entire shipping system should be able to operate and meet all availability and performance requirements with only one help desk technician on duty."

Adding sensors introduces additional opportunities for both automation and error. If RFID allows you to push a pallet of boxes onto a truck without stopping to scan bar codes manually, it also allows you to push the wrong pallet onto the truck if the right pallet is close enough for the reader to see it. This could be a problem especially if backup readers have powered up to cover a dead spot from a failed reader. Because RFID allows you to automate a process to the point of obviating human interaction, a failed reader could result in serious errors or lost information if there is no equally automated way to monitor the system's status.

We have dedicated Chapter 9 to this topic. For now, though, note that real-world RFID deployments will not succeed without automated provisioning, monitoring, and management of the RFID infrastructure.

2.4.5. Extensibility and Maintainability

For most organizations, RFID readers are the first step into a new space at the edge. Some industriesespecially within the areas of manufacturing, agriculture, and defensehave worked with distributed sensors for some time, but a company new to RFID may falsely assume that once it gets the RFID readers talking to the management applications, it's done. Mandates and contracts that today require RFID compliance could easily require temperature, humidity, shock, or magnetic field exposure tracking tomorrow. RFID systems are in a state of flux as technology advances and standards only begin to solidify. An RFID installation of any size must be prepared to plug in readers with different mounting requirements, different management interfaces, and even different manufacturers depending on what is available this week. Requirements in this area are often stated thus: "The receiving system should be able to use any EPC Class I reader from any vendor, with less than three person-days of configuration for a new type."

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.5. Architecture Guidelines

The infrastructure for RFID edge components must be incredibly robust. Ideally, the readers, RFID middleware, and so on should support effortless plug-and-play functioning. In other words, they should be the edge equivalents of telephones in terms of ease of use, provisioning, monitoring, and management. A human operating a bar code scanner will be able to tell if the scanner goes down, bu you'll need to employ automated means to monitor and manage RFID readers. Also, you'll want to shield your applications from the large number of readers deployed in a store or other location. To support all this, your edge architectures will need to be more flexible, scalable, robust, secure, and manageable than ever before.

As is always the case in a technology's early adoption phase, RFID standards and products are rapidly evolving. It is very likely that you will encounter a highly heterogeneous infrastructure, whereas types of RFID tags, readers, and other sensors will vary. Event managers based on standards such as ALE will have to coexist with other vendor-specific extensions to vertical applications such as point-of-sale and warehouse-management systems. When developing your RFID architecture roadmap, it is very important to take a close look at both your business and systemic quality requirements and how they might evolve over time. RFID systems are no different than any other distributed system in that you should plan for performance, scalability, security, manageability, maintainability, ease of use, and failover early on. The following section cover some important principles to consider when devising an RFID architecture roadmap for your company.

2.5.1. Begin with Business Requirements

While researching this book, we approached many experts in RFID technologies and asked them wha aspect was most overlooked when designing RFID systems. The almost unanimous response was business processes. In our experience as system architecture consultants, which has spanned severa new waves of technologies, we have often seen people forget that technology's ultimate purpose is t solve business problems. This is the case even more so with promising new technologies, as the newness and buzz surrounding these technologies drive a bottom-up way of thinking. While you're defining the business requirements and capturing functional requirements, pay careful attention to accounting for exception processing. These are the "what if" situations, such as what to do if a reade goes down, or a tag falls off a package, or you are not able to read a particular tag. The exception scenarios should provide capabilities to use alternate means to accomplish the task. For example, if a read registers only 50 packages of a product when an operator visually determines that the number should be closer to 100, you could create alternate workflows to obtain the inventory count manually

2.5.2. Don't Forget Your Existing Infrastructure

More often than not, RFID systems will have to coexist with bar code systems, providing alternate, more automated capabilities. When planning the architecture of your RFID system, pay careful attention to your existing systems and business processes.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.5.3. Process Data at the Edge Where Possible

Process as much data as possible at its source. This means that the raw RFID observations collected by readers should be filtered at the edge before sending only what is meaningful to applications. Develop a data-management architecture that keeps data needed for the workflow at the edge on the edge. If RFID data gives you enough information to sort items at the factory into different bins based on their models, then don't make that decision at corporate headquarters. Have the edge server make the call based on an understanding of which item types go in which bins. The list of which item types go in which bins can be updated from time to time via an automated process.

2.5.4. Track Items to the Level Your Business Processes Will Support

The item-level tracking ability offered by RFID systems can come in handy for many applications. You might wonder then why we are listing it as a challenge. The real challenge is not in identifying an RF tag uniquely, but in figuring out what you are going to do with that information. More information is not always better. Weigh the cost of generating, storing, and processing this information against the business benefits. Tracking items at instance level greatly increases the data that is generated as the items move through the supply chain. Returning to our earlier example, consider that each Nirvana store stocks 20 Samsung P725 cell phones. With item-level tracking, each store inventory will have a unique record for each instance of the phone. Similarly, each instance of the phone will be separately tracked as it moves through Nirvana's supply chain. In addition to adjusting to the increased volume of information, this finer-grained tracking data would need to be incorporated into the business processes and thus the systems that support them.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.6. System Management

The level and scale of automation for some RFID systems poses significant challenges for system administrators and managers. As stated earlier, if a manual bar code scanner is down, a human operator will immediately pick up on it and call for service. The automated processing capabilities tha RFID technologies offer make automated detection of system faults and stoppages indispensable. Similarly, we need to provide novel solutions for provisioning, monitoring, and managing our silently ticking antennas, readers, and other RFID infrastructure components.

Think ease of configuring, provisioning, managing, and monitoring when deciding on infrastructure components such as readers, sensors, event managers, servers, storage, and networks. Build redundancy into the architecture. Have a plan in place for what should happen when an antenna, a reader, or an event manager malfunctions. Consider using RFID middleware and other system management solutions to stay ahead of the difficulties that come with a fast-growing edge infrastructure.

2.6.1. Start with Architecture

Much of the time, enterprise architecture grows out of its selection of products. With minor exceptions, over time, a proprietary architecture that is locked into a particular product and vendor develops. Begin with well-defined interfaces between different layers, such as event managers, applications, and integration servers, and then choose products that fit your architecture. Later, this will pay in greater flexibility and independence.

2.6.2. Use RFID Middleware

As explained earlier in this chapter, you will find it necessary to use RFID middleware for all but trivia implementations. RFID middleware helps decouple your enterprise applications from the physical architecture of your RFID systems; those applications don't need to know about the reader vendors, their APIs, or the physical configuration of readers and antennas. In addition to this, RFID middleware provides the important function of filtering the highly fragmented data that comes from the readers. By cutting down the volume of data that passes over wide area networks, event filtering also allows you to define application-level events and pass more meaningful information to your enterprise applications.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

2.7. Summary

RFID and other, similar technologies are pushing increasingly more network intelligence toward the network's edge, where RFID readers, temperature and light sensors, and smart middleware manage events without requiring guidance from massive, central servers. This movement toward edge computing is changing the way companies think about their data and computing resources. We shouldn't underestimate how disruptive this will be to the way we design, deploy, and manage our enterprise systems. In large part, how we design systems for RFID now will shape the way we deal with other edge devices for many years to come.

In this chapter, we covered the following important points about RFID:

- A confluence of factors has led us to the tipping point, where RFID seems primed for widespread adoption. Technological factors include advances in semiconductor technologies, the advent of intelligent devices, widely available broadband networks, and web services standards available under the umbrella of service-oriented architecture. Nontechnological factors include the interest of leading government agencies such as the FDA and the DoD, coupled with megacorporations such as Wal-Mart and Tesco, in RFID technology.

- When planning the architecture of an RFID system, the key processes include selecting an item numbering scheme, encoding RFID tags, attaching RFID tags to items you want to track, reading the RFID-tagged items as they move through the physical environment, integrating RFID information into enterprise applications and business processes, and, finally, sharing RFID information within and between companies.

- As with most enterprise systems, RFID systems are multi-layered. The physical layer components include RFID tags, antennas, readers, and sensors.

- RFID middleware provides many important services, including decoupling the physical layer components from the application layer components, managing and monitoring physical layer components, and filtering raw RFID observations and generating high-level events that are more meaningful to applications. The Application Level Events (ALE) standard is the EPCglobal standard that is gaining momentum for RFID middleware. There are already several vendor products available that implement ALE. (ALE is explained in detail inChapter 7.)

- An RFID service bus is a type of enterprise service bus that serves to integrate application-level events captured by event managers into edge workflows.

- As companies deploy RFID systems, they will want to share some of the tracking information and other relevant information about the items being tracked. Various options are available for RFID information services. The EPCglobal Network (discussed further inChapter 8) is a vision of RFID information services proposed by EPCglobal.

- As with any enterprise application, it is paramount to understand systemic quality requirements before designing an RFID system. Important RFID system qualities include privacy and security performance, scalability, manageability, flexibility, and maintainability.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

The intent of this chapter was to introduce you to the important facets of RFID system architecture. The remainder of this book discusses the individual components in further detail.

PREV

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Chapter Three. Tags

The purpose of an RFID tag is to physically attach data about an object (item) to that item. Each tag has some internal mechanism for storing data and a way of communicating that data. Figure 3-1 shows diagrams of several representative RFID tags.

Not every sort of RFID tag has a microchip or a built-in power source, but every RFID tag has a coil or antenna of some sort.

While it is important to realize what all tags have in common, classifying tags helps in understanding how they work. In this chapter, we will categorize RFID tags by those criteria most likely to affect the capabilities of the tag within our applications. We will divide tags based on their physical characteristics, their air interfaces (how they communicate with readers), and their information storage and processing capacity. With these criteria in mind, we will then explore some of the standards that define these variations and talk about guidelines for matching tags to applications. We'll start by looking at some basic tag capabilities before launching into the discussion of tag categories.

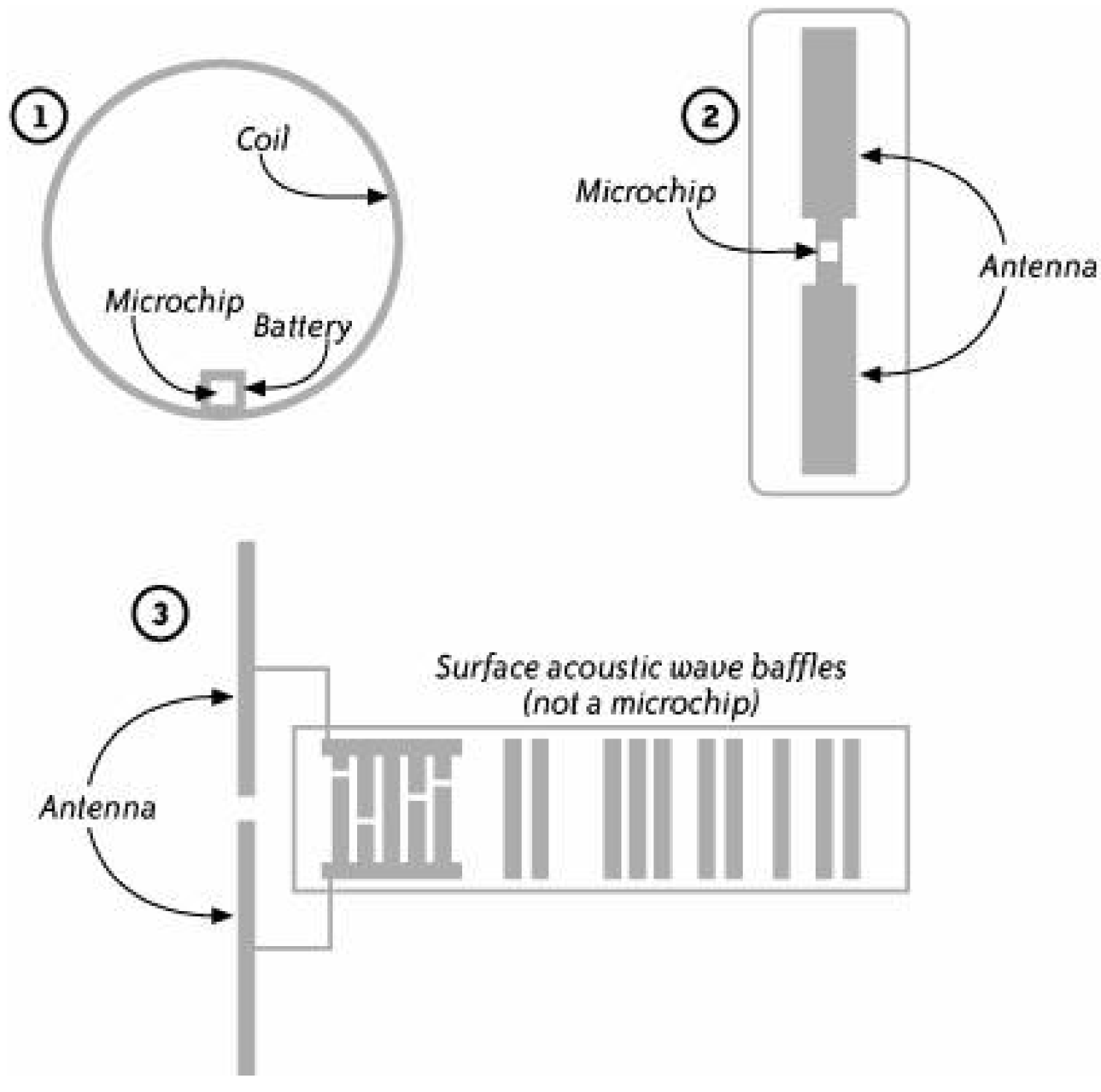
downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.1. Basic Tag Capabilities

Many basic operations can be performed with an RFID tag, but only two of them are universal.

Figure 3-1. Typical RFID tags



Attaching the tag

Any RFID tag must be attachable to an item in some way.

Reading the tag

Any RFID tag must be able to communicate information over some radio frequency in some way.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Many tags also offer one or more of the following features and capabilities:

Kill/disable

Some tags allow a reader to command them to cease functioning permanently. After a tag receives the correct "kill code," it will never respond to a reader again.

Write once

Many tags are manufactured with their data permanently set at the factory, but a write-once tag may be set to a particular value by an end user one time. After that, the tag cannot be changed except, possibly, to be disabled.

Write many

Some tags can be written and rewritten with new data over and over.

Anti-collision

When many tags are in close proximity, a reader may have difficulty telling where one tag's response ends and another's begins. Anti-collision tags know how to wait their turn when responding to a reader.

Security and encryption

Some tags are able to participate in encrypted communications, and some will respond only to readers that can provide a secret password.

Standards compliance

A tag may comply with one or more standards, enabling it to talk to readers that also comply with those standards; or, in the case of standards for physical characteristics, a tag may fit in a particular standard receptacle.

With these basics in mind, we can begin to explore the various categories of tags.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.2. Physical Characteristics

Because RFID tags must physically attach data to items of different shapes and sizes in different environments, they come in a wide assortment of shapes and sizes. Furthermore, they may be housed in many different kinds of materials. Some of the physical characteristics of various tags include:

- PVC or plastic buttons and disks, usually including a central hole for fasteners. These tags are durable and reusable.
- RFID tags shaped like credit cards, which are called "contactlesssmart cards ."
- Tags made into the layers of paper in a label, called "smart labels." These may be applied with automated applicators similar to those used for bar code labels.
- Small tags embedded in common objects such as clothing, watches, and bracelets. These small tags may also come in the form of keys and key chains.
- Tags in glass capsules, which can survive even in corrosive environments or in liquids.

Figure 3-2 shows just a few types of tag housings.

Figure 3-2. Tags come in many shapes and sizes (image courtesy of Texas Instruments)

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Packaging may be the most obvious way to divide tags into categories, and this is the property of a tag that most directly affects how the tag can be attached to an item. We've already mentioned that buttons typically attach with a fastener through a central hole. Keys and key chains attach as you would expect, and smart labels attach with an adhesive backing. A plastic medicine bottle may contain a glass capsule in its lid or base, and smaller capsules can be injected for tracking pets and livestock.

Warning: Any new RFID application should be tested in candidate systems, with tags in the same packaging and attached in just the same way as they would be in the production system.

---

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.3. Power Source

A common way of categorizing tags is by their source of power. This is also one of the main determining factors for the cost and longevity of a tag.*Passive* tags obtain all of their energy by some method of transmission from the reader.*Active* tags use an on-board battery to power communications, a processor, memory, and possibly sensors. Traditionally, tags that use battery power for some functions but still allow the reader to power communications have been termed "active" as well, but for clarity, we will use the more recent terminology for them:*semi-passive*. One additional type of tag is not only capable of supplying power for itself but is also able to initiate communications with other tags of its own kind without the aid of a reader. These tags are called *two-way* tags.

As you might expect, having a battery on board makes for a more expensive chip, but semi-passive and active tags have several advantages over passive tags. In the case of semi-passive tags, the read range may be longer because the passive communications can use all of the power provided by the reader for communications rather than sharing some of the power with the chip. An active tag may have an extremely long read range and may perform some functions in the absence of a readerfor example, using battery power for environmental sensors. This capability can be very useful for tags that identify items such as perishable goods.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.4. Air Interface

The *air interface* describes the way in which a tag communicates with a reader. By knowing a tag's air interface, we can determine the tag's read range and identify readers compatible with the tag. The following sections describe the attributes that comprise the definition of an air interface. The major attributes include the tag's power source, operating frequency, communication mode, keying, encoding, and coupling.

RFID in History

RFID has been in use longer than most people realize. It first appeared in the air war during WWII. British pilots noticed that sometimes, inexplicably, all of the German planes in sight would roll in unison. British intelligence later learned that German pilots were participating in a sort of manual RFID to help identify their planes to the newly invented German radar systems. By changing their radar images in response to a ground challenge, the pilots identified themselves as Luftwaffe.

The British later developed an automated system for their own aircraft that caused their planes to appear to pulse according to a preset rhythm when illuminated by the British "Chain Home" (CH) air defense radar. This new system was known as the MK I and was the predecessor of all Identify Friend or Foe (IFF) and aircraft transponder systems in use to this day.

Like RFID systems today, the most important components of the system were an *interrogator* and a *transponder*. In early IFF systems, the interrogator was the radar system itself and the transponder was an unwieldy box of tubes with dials and switches. The term "interrogator" gives us a clue as to how the system worked: the ground station sent out a radar signal, and the transponder receiving this signal reflected it back, causing the radar antenna to receive a stronger return than it otherwise would have. The transponder also "swept" the frequency of its return back and forth over a small range as it responded, causing the radar return to pulsate according to a specific rhythm.

3.4.1. Operating Frequency

The *operating frequency* is the electromagnetic frequency the tag uses to communicate or to obtain power. The electromagnetic spectrum in the range in which RFID typically operates is usually broken up into low frequency (LF) , high frequency (HF), ultra-high frequency (UHF), and microwave (see Table 3-1). Because RFID systems broadcast electromagnetic waves, they are regulated as radio devices. RFID systems must not interfere with other, protected applications, such as emergency service radios or television transmissions.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Table 3-1. RFID frequency ranges

Name	Frequency range	ISM frequencies
LF	30300 kHz	< 135 kHz
HF	330 MHz	6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz
UHF	300 MHz-3 GHz	433.920 MHz, 869 MHz, 915 MHz
Microwave	> 3 GHz	2.45 GHz, 5.8 GHz, 24.125 GHz

In practice, this means that the actual frequencies available to RFID are limited to those frequencies set aside as Industrial Scientific Medical (ISM). Frequencies lower than 135 kHz are not ISM frequencies, but in this range RFID systems are usually using powerful magnetic fields and operating over short ranges, so interference is less of an issue than it might be otherwise.

Different frequencies have different properties. Lower frequency signals are better able to travel through water, while higher frequencies can carry more information. Higher-frequency signals are also typically easier to read at a distance. Table 3-2 shows the read ranges for the different frequencies and how they have been used in different applications.

Table 3-2. Read range by frequency

Frequency	Typical max. read range for passive tags	Some typical applications
LF	50 centimeters	Pet identification and close reads of items with high water content
HF	3 meters	Building access control
UHF	9 meters	Boxes and pallets
Microwave	> 10 meters	Vehicle identification of all sorts

The recent cost reductions in UHF tags have led to an increased use of these tags in applications where LF and HF tags were more common in the past. UHF is not likely, however, to replace LF in implanted tags or microwave in long-range (greater than 10-meter read range) applications.

Regulatory bodies have chosen different ranges for UHF in different parts of the world. In Europe, South America, and much of Asia, UHF RFID tags operate from 865 MHz to 868 MHz. In North America, they operate from 902 MHz to 928 MHz, and India recently adopted a range of 865 MHz to 867 MHz. China has not announced which range it will favor, but there has been some indication that the range it chooses will support global standards (seeChapter 11). One possible global standard is the EPCglobal Gen2 standard, described in more detail later in this chapter. Gen2 tags operate from 860 MHz to 960 MHz and at a range of power levels designed to match varying local regulatory requirements.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.4.2. Communications Mode

Another way of distinguishing tags is by whether or not the tag and the reader can "talk" at the same time. This is known as the *communications mode*. As with wired communications, RF communications may be *full-duplex* (FDX) or *half-duplex* (HDX)that is, the tag and reader may talk at the same time (FDX) or take turns (HDX). In most cases, for passive tags, the reader provides power throughout the conversation, but in one variation on HDX, power transmission stops while the tag responds. A capacitor or some physical property of the tag allows it to store energy and respond while the power transmission is off. We will call this communications mode*sequential* (SEQ), in keeping with Finkenzeller.[*]

[*] Klaus Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, trans. Rachel Waddington (Chinchester, West Sussex, England: John Wiley & Sons, 2003).

3.4.3. Types of Keying

The term "keying " comes from the days of telegraphy, when an operator pressed a manual key to make long and short tones. Keying describes which attributes of an analog carrierthe analog carrier can be a wave or a fieldmay be modulated to represent the ones and zeros of a digital message. There are three main types of keying:

Amplitude-shift keying (ASK)

A type of keying that sends digital data over analog carriers by changing the amplitude of a wave in time with the data stream

Frequency-shift keying (FSK)

A type of keying that sends changes through the frequency of the wave (or how often a wave crest comes along)

Phase-shift keying (PSK)

A type of keying that sends changes through the distance by which the waves lead or follow a reference point in time

3.4.4. Encoding

Encoding determines the way the tag and reader will interpret changes in the analog carrier to represent digital data. Thus, an encodingis an agreement between sender and receiver regarding what the keyed changes mean. Morse Code, one example of an encoding, uses long tones to represent "dashes" and short tones to represent "dots." If we substituted "0" for "dash" and "1" for

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

"dot," Morse encoding would work for sending information over a serial bus.

It might seem odd to think of RFID as devices communicating over a serial bus, but even though the components are wireless rather than wired, most tagsand readers use serial communication because of the limited frequency range available. RFID even uses some of the same encodings that many of us first encountered in Universal Asynchronous Receiver/Transmitter (UART) chips. Various encoding schemes commonly used in RFID include:

Biphase Manchester encoding

This encoding uses a negative transition in the middle of the bit cell to mean a "1" value and a positive transition in the middle of the bit cell to mean a "0" value. Transitions that happen at the bit cell boundary do not encode a value and are used to "reset" the encoding to send the same value again in the middle of the cell. Encoding is independent of the type of keying or modulation, so this could be an ASK, FSK, or PSK transition. Because transitions are synchronized with the clock, this method is said to have the clock in the data stream. A recipient can derive the clock signal from this encoding and so regain synchronization, which makes biphase Manchester encoding well suited for transferring large messages. ISO 18000-6 type B tags use biphase Manchester encoding to talk to a reader.

Pulse interval encoding

This is similar to Morse Code, where values are encoded by the length of the pause, or interval between, short pulses. A pause of a specific length represents a "1," while a pause twice as long represents a "0." This has the advantages of requiring less power and being resistant to noise. It may be a problem for longer transmissions, though, because the data stream does no contain a clock. ISO 18000-6 type A tags use pulse interval encoding to send messages to a reader.

Biphase space encoding

In biphase space encoding a type of encoding that is often used for reader-to-tag communicationtransitions happen at each clock tick (actually, at the edges of the bit cell, but it amounts to the same thing). If a transition occurs in between these "ticks" of the clock, it represents a "0"; no extra transition represents a "1." This encoding method contains the clock in the data stream and so is well suited for long messages. ISO 18000-6 readers use biphase space encoding.

Pulsed RZ encoding

This is a term we've coined to describe a type of encoding in which four short pulses in the first half of a bit cell represent a "1," while four short pulses in the last half of the bit cell represent a "0." The "RZ" means "return to zero," which is a term used to describe encodings in which the signal returns to a point that is neither high nor low (i.e., at zero) for some part of the bit cell. EPC HF readers use pulsed RZ encodingwith four pulses for each symbol.

EPC Miller encoding

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Miller encoding encodes a number by having a transition in either direction at the half-bit point EPC UHF Gen2 uses a version of this encoding that can have several timing pulses in a bit, wherein a "transition" is actually represented by the lack of one of the timing pulses. The number of transitions to use as a timing pulse in a bit is called the "M" value. For example, for M=2 we would have 2 pulses per bit cell to mark the timing of the clock. If one of these pulses does not occur, this represents a "1"; otherwise, the bit cell represents a "0."

"1 of 256" and "1 of 4"

These two forms of encoding are used in tags that conform to ISO 15693. For "1 of 256," values from 0 to 255 may be encoded in the data stream by the timing of a pulse. There are 512 timing slots per frame during which a pulse may occur, and the value is encoded by multiplying it by 2 and then adding 1. This gives the slot locationfor example, a value of 12 would be encoded by a pulse at slot 25. Only odd slots are used for data, as even-numbered time slots are used for End of Frame (EOF) signals. Counting the odd slots leaves 256thus the name of this encoding and its capability to send an entire byte in each frame. "1 of 4" encoding works in a similar way but uses only eight total time slots, which allows for 2 bits to be sent at a time.

FSK subcarrier encoding

This is another term we've coined for want of a better one. This sort of encoding is the exception to our earlier claim that RFID communicates over a serial bus and that encoding and keying are different. In this type of encoding, two subcarriers represent "1" and "0," respectively, and the sender uses FSK to create pulses on these subcarriers in time with the data stream. EPC UHF tags use FSK subcarrier encoding to talk to a reader.

3.4.5. Coupling

A tag's coupling mechanism determines the way a circuit on the tag and a circuit on the reader influence each other to send and receive information or power. The type of coupling a tag uses directly affects the read range between the tag and reader. We can group the different read ranges loosely into those systems where the read range is*close* (within 1 cm), *remote* (1 cm to 1 m), or *long-range* (more than 1m). A synonym for remote coupling is "vicinity coupling." In this section, we will discuss backscatter coupling , inductive coupling, magnetic coupling, and capacitive coupling. Capacitive and magnetic coupling are examples of close coupling, inductive coupling is a type of remote coupling, and backscatter coupling may be remote to long-range.

Along with range, the choice of coupling mechanism strongly affects which frequency the tag should use. Inductive coupling typically works best in either the LF or HF range. Backscatter coupling works better with higher frequencies. Magnetic coupling is most efficient at 110 MHz, and capacitive coupling usually runs at about 10 MHz to provide a fast enough clock for communications and processing.

RFID coupling, especially in the case of passive tags, may seem a bit mysterious at first. How can a tiny tag with no battery participate in a radio conversation? Let's find out.

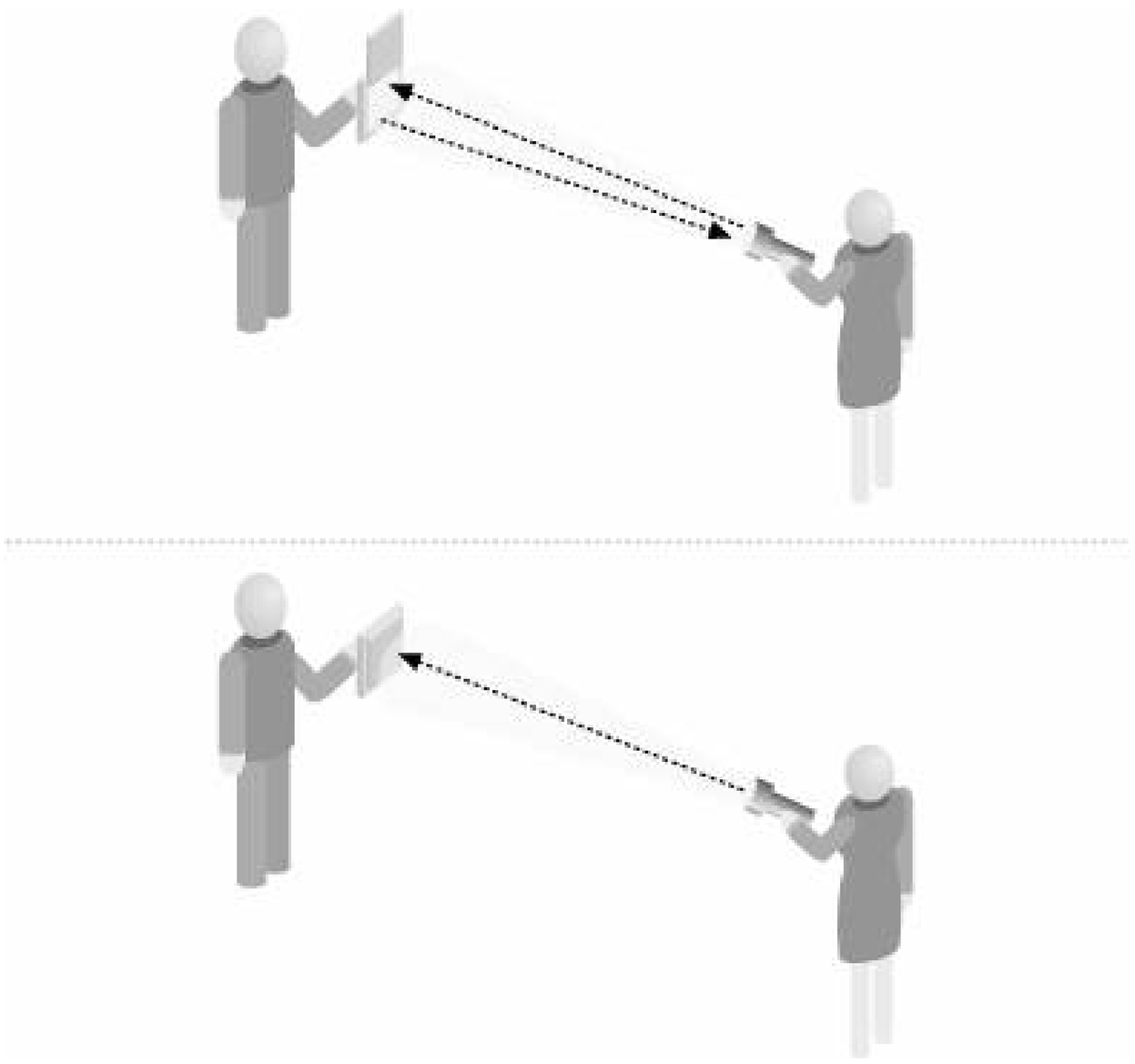
downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.4.5.1. Backscatter coupling

Backscatter coupling provides an elegant solution to the puzzle of how to make an RFID tag without a battery. The name itself, "backscatter," describes the way the RF waves transmitted by the reader are scattered back by the tag. That is to say, the waves are reflected back to the source to send a signal. Imagine the reader as a flashlight and the tag as a signaling mirror with a cover, as shown in Figure 3-3.

Figure 3-3. Backscattering an RF signal is like reflecting light with a signaling mirror



The person holding the flashlight sends signals by turning the beam on and off. When left on, the flashlight provides a signal carrier that allows the person holding the mirror to respond by exposing and concealing the mirror. In effect, this emulates a second, much weaker flashlight. A backscatter tag uses a similar trick to reflect or backscatter the RF signal generated by the reader.

So how can a reader actually power the tag's chip as well as its communication? Imagine for a moment that the shutter of the signaling mirror is controlled by a microchip. If a photovoltaic cell powered the microchip using energy from the light of the flashlight beam, our example would be very similar to a passive backscatter tag. If this microchip were powered by a small battery, but the device still used reflected light from the flashlight to communicate, it would be similar to what we referred to earlier as a "semi-passive" backscatter tag.

Notice that in our example, we are sending back the same frequency electromagnetic waves from the

downloaded from: lib.ommolketab.ir

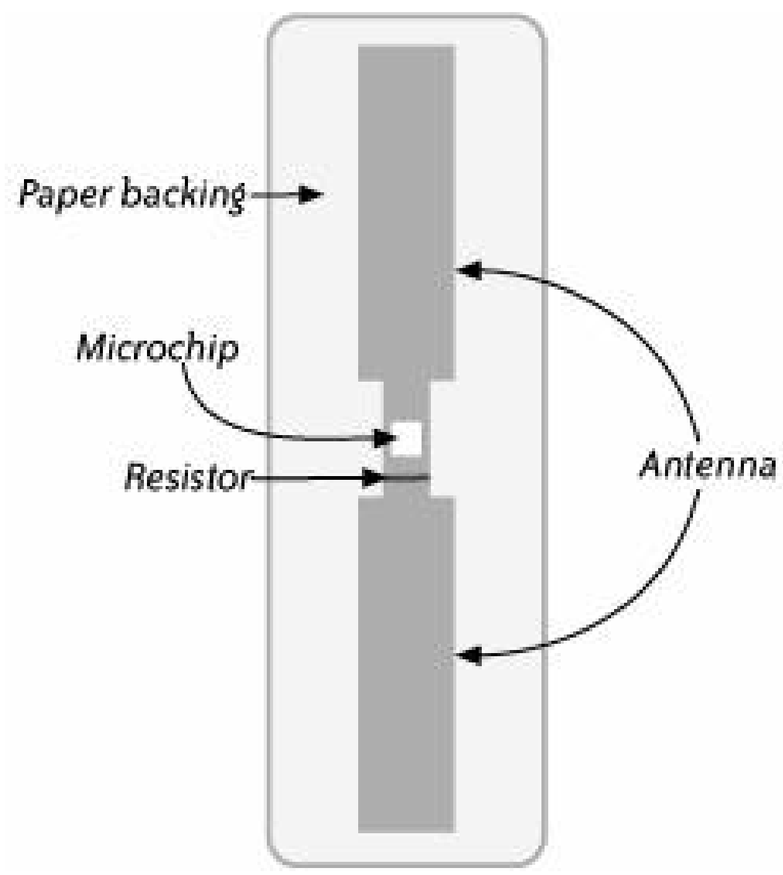
downloaded from: lib.ommolketab.ir

mirror as we receive from the flashlight. The term "backscatter" is used today to describe tags that reflect the same frequency emitted by the reader but change various qualities of that reflection to send information to the reader. Some tags do this via physical properties of the tags themselves, and others by switching a load connected in parallel to the tag's antenna that causes the antenna to reflect less efficiently when the load is applied. The distinguishing characteristic to remember is that backscatter tags reflect back the same frequency that the reader supplies to power them.

Because the reader and tag are each using the same frequency for communications, they typically take turns. As mentioned earlier, we call this a half-duplex (HDX) communications mode. The reader continues to power the tag even when the reader is waiting for the tag or receiving from it, so backscatter systems are not the kind of system we described as SEQ. In our flashlight and mirror example, we used ASK as the keying because we did not change the phase or frequency of the signal.

Components similar to those in our example are packaged in a more convenient form in Figure 3-4.

Figure 3-4. A modern "smart label"-style RFID tag



Besides reflecting power back to the reader for communications, the tag's antenna conducts some of that energy to provide power for a small chip. The chip controls a resistor between the two halves of the antenna. This resistor is like the shutter in our flashlight and mirror example. When the two halves of the antenna connect directly with low resistance, they reflect the reader's signal with a high amplitude. When the resistor separates the two halves of the antenna, they reflect the reader's signal with a lower amplitude. By switching the resistor in and out of the circuit, the chip is able to create a load-modulated ASK signal to transmit a unique ID number stored in the chip's memory.

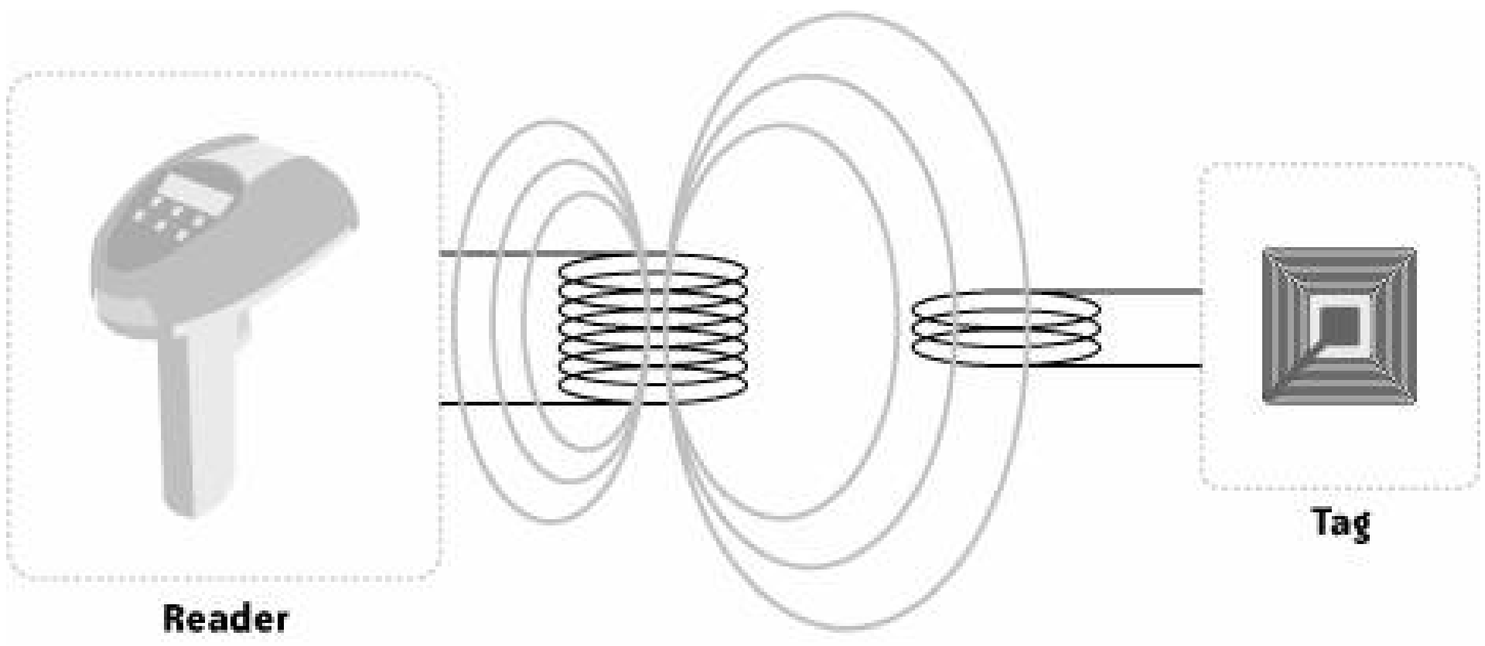
3.4.5.2. Inductive coupling

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Inductive coupling is a common type of remote coupling. Some of the most common types of tags in recent years have been various kinds of inductively coupled tags, including tags that follow the ISO 15693 standard for vicinity-coupled smart cards. A reader powers inductively coupled tags by using a coil antenna to generate a magnetic field. The field drives current through a coil on the tag by induction in much the same way that a transformer transfers energy between two coils. This is illustrated in Figure 3-5.

Figure 3-5. Inductive coupling uses transformer coils



For this reason, this type of coupling is sometimes called*transformer coupling*. The field provides ample energy to power a microchip, which may then communicate with the reader by load modulation in almost the same way as in the backscatter example. A resistor that is switched on and off in the tag causes fluctuations in the magnetic field, which create voltage changes in the reader's antenna. Unlike with backscatter coupling, however, the changes are so minute that designing a reader to recognize them directly would be prohibitively expensive. Thus, one of three methods is used to generate a tag response frequency.

In the *subcarrier* method, the tag switches its resistor on and off very quickly. Thus the tag seems to generate two new frequencies, one above the fundamental frequency and one below. For example, if the reader is operating at 13.56 MHz and the tag switches its resistor at 500 kHz, the two new frequencies will appear at 13.810 MHz and 13.310 MHz. These two new frequencies are called subcarriers. The reader can recognize them easily because they differ from the frequencies the reader itself generates. The tag sends information to the reader by modulating one or more of the subcarriers.

Another trick used with inductive coupling is the*subharmonic* method, which divides the fundamental frequency by an integral valueusually twoto produce a frequency for the tag to use in its responses. If, for instance, the reader generates a fundamental frequency of 128 kHz, a tag using a subharmonic based on one half of the fundamental frequency would respond at 64 kHz. (Division by other values is possible but rare.)

Both subcarrier and subharmonic inductive coupling use FDX mode to communicate, since the reader and tag transmit on different frequencies. Unlike with RF backscatter, there is no need to take turns. Any sort of keying can be used to encode the data, but the most commonly used keying is, again, ASK.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

The remaining type of inductive coupling is sequential (SEQ) and uses FSK for modulation. Interestingly, it is also technically a variation on the backscatter method, since it uses the same frequency to send data from the tag and from the reader. In sequential operation, the reader field charges a capacitor on the tag and then shuts off. The tag then uses an oscillator to create its own magnetic field at the same frequency as the reader's (because the antennas on the reader and tag are both tuned to that frequency). The reader is able to detect this field because its own field is off. The tag encodes data using FSK, by accelerating and decelerating the circuit's oscillation.

Tags using inductive coupling are passive or semi-passive and have capacities of up to 2 kilobits. Read-only, write-once, and rewritable tags are all available with this type of coupling. Inductively coupled reader antennas are coils that generate a magnetic field, so this type of coupling has relatively large antennas and short read ranges. Typical read ranges can be as much as 25 cm with a 25-cm reader antenna and 40 cm with a 40-cm reader antenna, but 10-cm read ranges are the most common. This is far enough that inductive coupling may be classed as a remote coupling.

3.4.5.3. Magnetic coupling

Magnetic coupling is a close coupling that is similar to inductive coupling in that the reader and tag form a pair of transformer coils. The major difference is that the reader coil in magnetic coupling is a round or u-shaped ferrite core with windings, as shown in Figure 3-6. The tag must be within 1 cm of the reader and placed over the air gap in the core. Because the distance is so small and the coupling is so strong, magnetic coupling uses direct ASK with load modulation for simplicity.

This type of coupling can power complex chips. Close coupling systems usually require the tag to be inserted into the reader, so close coupling is ideal for smart cards. (ISO 10536 defines standards for magnetically coupled smart cards.)

3.4.5.4. Capacitive coupling

Capacitive coupling, another form of close coupling that works best with the tag inserted into the reader, is also usually used for smart cards (as defined under ISO 10536). Capacitive coupling dispenses with antennas and replaces them with electrodes. The reader and tag each have conductive patches that together form a capacitor when held exactly parallel to each other without touching. As Figure 3-7 shows, the circuit created is identical to one in which the reader and tag are directly connected through a capacitor and share a common ground. As with magnetic coupling, this type of coupling can easily power complex tags, and it may use simple ASK with load modulation to transfer data.

Figure 3-6. Magnetic coupling uses a ferrite core antenna

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

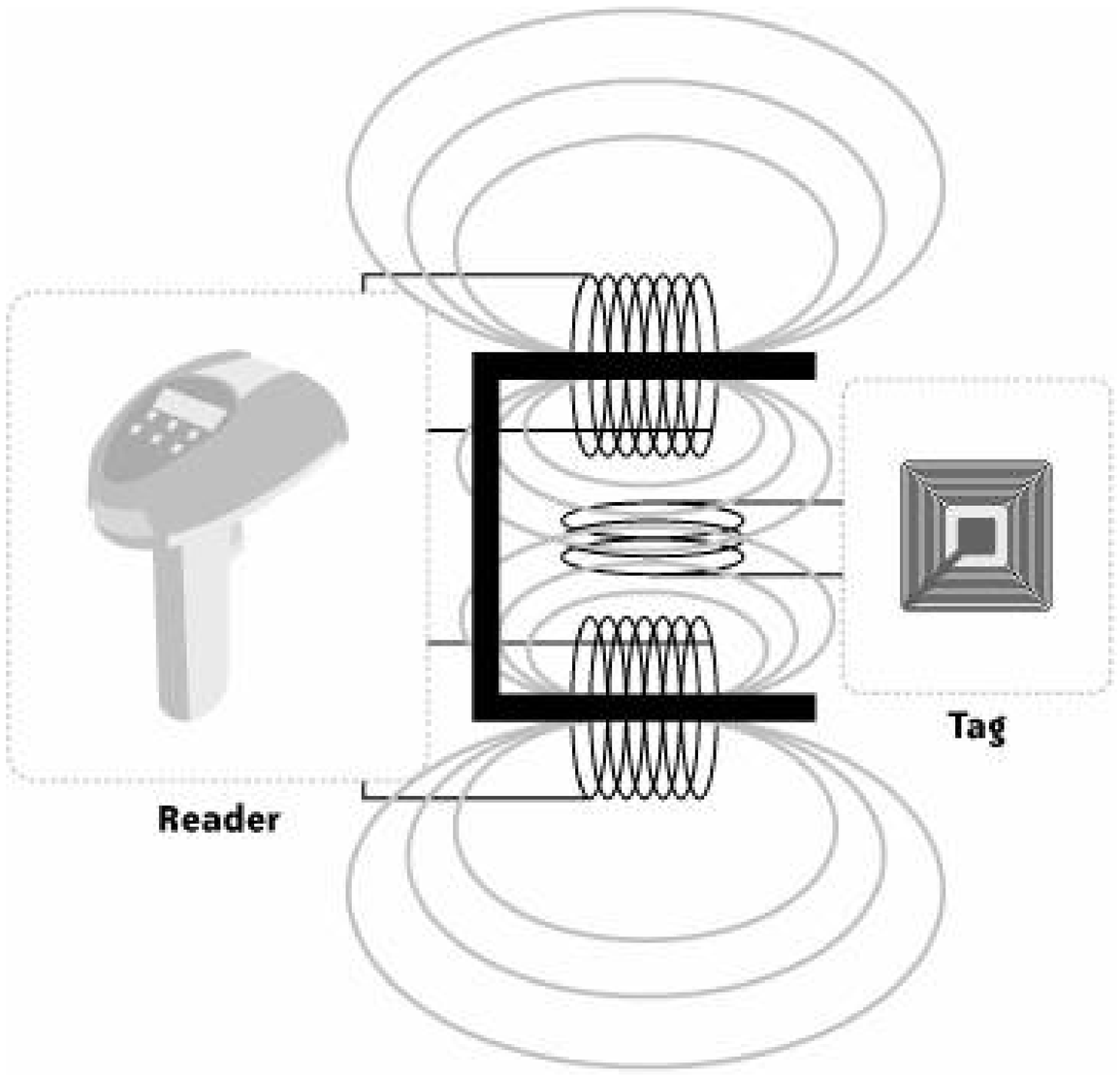


Figure 3-7. Capacitive coupling is akin to connecting the tag and reader through a capacitor

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.5. Information Storage and Processing Capacity

Information storage and processing capacity is the final major consideration when dividing tags into categories. RFID tags range widely in their capability to store information. The simplest tags store only 1 bit. Systems based on these tags, such as those commonly used in libraries and clothing stores for theft prevention, can only recognize the presence or absence of a tag; they cannot identify individual items. On the other hand, some tags may store kilobytes of data. Larger capacities usually require active tags, and even among passive tags, larger memory capacity directly increases the cost per tag.

How Many Bits Do I Need?

Some of the latest active tags ordered by the U.S. Department of Defense will be able to store as much as 256 bytes of information. How much storage do you need for your application? To get an idea, let's look at some numbers:

- Number of people on Earth: 6.3 * 10^9
- Number of grains of sand on Earth's beaches: 7.5 * 10^17
- Number of photons in the visible universe: 1 * 10^88

This means we can count everyone in the world in 33 bits (2^33 = 8,589,934,592). We can count all of the grains of sand on the world's beaches (not counting deserts) in 60 bits, and we can count every photon in the observable universe in 293 bits (less than 37 bytes).

For less tangible things such as license keys or electronic message IDs, there may be some value in going to a larger number, but probably not. A 256-byte RFID tag must be storing much more than a number. Larger capacities allow tags to store location IDs for every place they have visited, or sensor readings that have either come from on-board sensors or been sent to the tag by a reader. Electronic signatures or encryption keys might take up more room and so make use of larger memory sizes. At some point, tags may carry entire user manuals and item-care instructions for automated handling systems. But for now, to uniquely identify tangible items, most applications will not need more than 96 bits.

This section examines storage and processing capacities for 1-bit EAS tags, surface acoustic wave (SAW) tags, and state machines and microprocessors.

3.5.1. One-Bit EAS Tags

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Electronic Article Surveillance (EAS) tagsare typically used to prevent theft. Rented videos and library books typically have EAS tags attached in the form of thin strips or labels. Stores often tag clothing with EAS tags inside hard plastic clips or buttons that are difficult to remove without the correct tool. Some EAS tags are even designed to damage an item if removed incorrectly and so discourage theft.

EAS tags are often called "1-bit" tags because they are capable of communicating 1 bit of information. One bit may be used to store the answer to a yes or no questionin this case, the question is, "Is there a tag present?" If a tag is detected, the answer is "1" or "yes." If a tag is not present, the answer is "0" or "no." EAS tags are simple and inexpensive. At present, these are the most commonly used RFID tags.

Because of their simplicity, EAS tags often do not have microchips or memory storage and are always passive, using modulation appropriate to a given type of coupling to create a distinctive signature that may be recognized by a reader. The types of coupling available in EAS tagsare as numerous as in other types of RFID tagsfor example, EAS tags may use induction or backscatter coupling, as do the more complex tags. However, the way EAS tags generate a response is somewhat different.

Inductive EAS tags are simple, resonant circuits that cause a voltage drop across the reader's coil. The reader "sweeps" the frequency of its field to allow for minor tuning defects in the tags and thus further reduce their cost. Once an item is purchased, these tags may be deactivated by "burning out' the resonant circuit with a massive induced voltage.

In the case of backscatter, which is usually used at microwave frequencies, a diode generates a harmonic of the fundamental frequencythat is, a frequency some integral factor higher than the fundamental frequency, such as two or three times the fundamental frequency. This harmonic frequency is then modulated using ASK to create a distinct pattern. All of the tags of a particular type will have the same pattern, so this does not constitute a unique ID. The purpose is to differentiate between a responding tag and environmental noise of the same frequency. Once an item is purchased, these tags, which are slightly more expensive than inductive EAS tags, are removed and applied to other items in the store.

Frequency divider tags make use of a simple microchip and a coil. Powered at a fundamental frequency generated by the reader, the chip divides the fundamental frequency by two and modulates the subharmonic with ASK or FSK, creating a modulated wave that is detected by the reader. Like the tags mentioned earlier in this section, frequency divider tags use this modulation simply to help distinguish a tag from environmental noise. These tags are also usually removed and reused rather than being disabled.

Electromagnetic EAS tags use a strip of amorphous metal that causes a distinctively odd variation in magnetic flux when it is exposed to an oscillating magnetic field. This variation is at a harmonic of the oscillation frequency of the field (the fundamental). To differentiate between tags and environmental noise, the reader may also modulate the fundamental, thus causing a matching modulation in the harmonic returned by the tag. These tags are usually disabled at the point of purchase by magnetizing hard metal plates in the tags with a strong magnet. The plates will then keep the amorphous strip from reacting to the reader's field. Demagnetizing the hard metal plate will reactivate this type of tag.

Acoustomagnetic tags also contain an amorphous metal strip and a hard metal plate, but in these tags the hard metal plate is magnetized when the tag is active, and the amorphous strip is attached at only one end. When exposed to a reader's field, this strip begins to vibrate. The reader then shuts

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

off the field, but the strip inside the acoustomagnetic tag continues to vibrate like a tuning fork, generating a weak alternating electromagnetic field that the reader can then detect without interference from its own field or reflections. This is very similar to sequential mode communications, although no information other than the presence of the tag is communicated. To disable this type of tag, the hard metal plate is demagnetized. These tags are somewhat more secure than electromagnetic tags in that disabling them is more difficult for a thief or prankster to accomplish without an expensive deactivator.

3.5.2. Surface Acoustic Wave (SAW) Tags

In between the 1-bit tags and other, more advanced RFID tags is an ingeniously designed oddity. SAW tags operate in the microwave range as backscatter tags and have no processors, but unlike 1-bit tags, a SAW tag can be encoded at the point of manufacture to contain a number.

Figure 3-8 shows what a SAW tag looks like. The antenna at the left receives the microwave pulse from the reader and feeds it to the interdigital transducer (the block on the left). The transducer contains a piezoelectric crystal, which vibrates when it receives the microwave pulse. This vibration creates an acoustic wave that travels through the tag, encountering reflector strips (shown on the right).

Figure 3-8. A SAW tag

The strips reflect back part of the wave, causing the crystal to vibrate again and creating a backscatter reflection. The number and spacing of the reflector strips determines the number and timing of the pulses sent back to the reader, and so determines the number represented by the tag. Since the number must be set at the time of manufacture, SAW tags are read-only, and because they

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

have no logic circuit, many tags may respond simultaneously to a reader pulse without waiting their turn, causing difficulties in sorting out where one tag's response starts and another's stops. At the time of writing, practical size constraints also limit SAW tags to a 32-bit capacity.

Because there is a delay as the acoustic wave propagates through the tag and reflects back to the reader, SAW tags operate in the SEQ communications mode. The reader broadcasts its pulse and then waits, without needing to power the tag. The reader cannot really be said to communicate with the tag since its interrogatory pulse is always the same. Think of a SAW tag as if it were simply "shaped like" a particular number that the reader is able to "illuminate" so that it becomes visible.

3.5.3. State Machines and Microprocessors

Some tags have more complex logic circuits than others. Most 1-bit tags and all SAW tags have no logic circuits at all, while other types of tags have sophisticated state machines incorporated into custom chips. An on-board state machine allows these tags to participate in a complex anti-collision dance with the reader, which helps the reader to distinguish individual tags from a large group of similar tags (For more on anti-collision, seeChapter 4.)

More complicated processors can provide security through encryption or through control sensors built into the chip. Advanced smart cards even have powerful microprocessors that can be programmed and reprogrammed with new capabilities. As with memory capacity, more processing capacity means a higher cost, so finding the right balance for a given application can mean the difference between a profitable system and an expensive failure. The best guiding principle here is probably "don't buy more than you need." Fortunately, even the most advanced tags are dropping in price as production volume increases, so what might have been too expensive last year may be a good fit this year.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.6. Standards

Some RFID applications need to interoperate only with the procedures and systems of a single company. Others must share information with a global consortium of partners. No matter where in this spectrum an individual application falls, standardswill affect the choice of tags for that applicationif only because large manufacturing runs of tags fitting a particular standard may make those tags less expensive than equivalent proprietary tags, even though the latter may be more simply designed. Choosing a tag is, to a large extent, choosing the type of RFID system you intend to build, and tag standards often involve much more than the physical characteristics and air interface of a tag.

3.6.1. EPCglobal Tag Types

EPCglobal, Inc., a collaboration between GS1 and industry partners, defines a combined method of classifying tags that specifies frequencies, coupling methods, types of keying and modulation, information storage capacity, and modes of interoperability. As described inChapter 2, EPC tags are intended to carry EPC numbers, which are assigned by the specific management entities who own the object classes involved. (We will talk in detail inChapter 4 about encoding and decoding EPC numbers from tag bitstreams.)

Note: The European Article Number and Uniform Code Council groups, formerly known as "EAN.UCC," chose the new name "GS1" in 2005. "GS1" is not an abbreviation, but instead stands for "one global standard, one global system and one global organisation."[*] We will use "EAN.UCC" and "GS1" interchangeably in this and other chapters, as some of the specifications predate the name change.

Table 3-3 shows the different classifications of tags recognized by ECPglobal. These classifications, which began with the Auto-ID Center, have mutated as actual standards developed and vendors made suggestions. For instance, Class 0+ is a slightly modified (but still compliant) implementation of the Class 0 standard offered by a particular manufacturer. Version 1.1 of the EPC Tag Data Standard defines the contents and encoding for all EPC data carriers (tags) regardless of their class, but the air interfaces (that is, the coupling, frequencies, and communications protocols) are, at the time of this writing, specified only for Class 0 and Class I tags.

Table 3-3. EPCglobal tag classes

Class	Description
Class 0	Passive, read-only
Class 0+	Passive, write-once but using Class 0 protocols

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Class	Description
Class I	Passive, write-once
Class II	Passive, write-once with extras such as encryption
Class III	Rewritable, semi-passive (battery-powered chip, reader-powered communications), integrated sensors
Class IV	Rewritable, active, "two-way" tags that can talk to other tags, powering their own communications
Class V	Can power and read Class I, II, and III tags and read Class IV and V tags, as well as acting as Class IV tags themselves

The EPC tag classes should not be seen as an attempt to create a taxonomy of RFID tags. Instead, these specifications are intended as a roadmap, with the goal of replacing installed tags with somewhat backward-compatible, higher-class tags as they become available. We say "somewhat" backward compatible because there have been suggestions that the intention is not to provide backward compatibility in the tags themselves, but instead to upgrade reader software or firmware to support both existing and new tags during a transition. ECPglobal has consistently promised to provide a reasonable migration path for early adopters. The standard defines an air interface from 860930 MHz in the UHF range. Earlier Auto-ID Center working documents also defined an air interface for 13.56 MHz in the HF range.

EPC HF tags are inductive coupled FDX tags with read ranges of up to one meter. The encodings are not specifically named in the specification, but the reader-to-tag encoding seems to be biphase space encoding, whereas the tag uses pulsed RZ encoding to respond to the reader. These tags use a protocol called the Slotted Terminal Adaptive Collection (STAC) protocol, which allows a reader to select a single tag from among a group of tags. For this reason, this protocol is called a *singulation* protocol.

EPC UHF Gen2 tags use backscatter coupling and an HDX communication mode, allowing for read ranges of up to 10 meters with optimal orientation of the tag and reader. Reader-to-tag communications are ASK and pulse interval encoded, and tag-to-reader communications are encoded using FSK subcarriers with biphase space or Miller encoding. These tags use a different singulation protocol than the HF tags: the Slotted Random Anti-Collision (SRAC) protocoltakes advantage of the faster turnaround time in UHF communications. Earlier EPC UHF tags used still another protocol, known as Adaptive Binary Tree (ABT).

Note: Don't be confused by the terms "Gen2" and "Class II." Class II is a particular class of tag functionality, as shown in Table 3-3. "Gen2" is short for Class I Generation 2, which is a new generation of tag protocols for UHF Class I tags.

All EPC tags must also support the permanent, remote destruction of an individual tag using a password. This password is 24 bits in Gen2. All tags also support a Cyclic Redundancy Check (CRC), which is stored in the tag. The CRC in Gen2 is 16 bits. EPC has defined tags for 64-bit and 96-bit memory capacity and is working on a standard for 256-bit. We will discuss tag communications, CRC and anti-collision protocols including STAC, SRAC, and Slotted Aloha in more detail in Chapter 4.

3.6.2. IS0/IEC 18000 Tags

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

In the past, GS1 has backed both the EPC standards and the GTAG initiative, which includes the 18000-6 standard for UHF. This standard originally conflicted with the EPC UHF specification and the new EPC UHF Generation 2 specification, but tag manufacturers were already selling EPC Class 0, Class 0+, and Class I tags. Some have expressed concern that the 18000 air interfaces and command protocols are more complex than those defined by the EPC standards, which could lead to both more expensive tags and stall adoption of the standard. Chris Turner, who has worked on the development of the 18000 standards, disagrees.[*] He contends that the 18000 protocols are flexible enough to support an EPC on an ISO tag for a price equivalent to that of an EPC tag.

[*] Chris Turner, "EPC and ISO 18000-6," *RFID Journal*, 3 March 2003, http://www.rfidjournal.com/article/articleview/325/1/2/.

In late 2004, EPC and ISO announced a plan to adopt EPC Gen2 as the standard for 18000-6C tags, but at the time of this writing (in the summer of 2005), ISO has not formally adopted Gen2.Table 3-4 lists the individual 18000 standards.

Table 3-4. ISO/IEC 18000 standards

Standard	Title	Description
18000-1	Generic Parameters for the Air Interface for Globally Accepted Frequencies	Principles and architecture for an RFID standard
18000-2	Parameters for Air Interface Communications below 135 kHz	LF, two tag types, optional anti-collision Tag type A: FDX 125 kHz Tag type B: HDX 134.2 kHz Passive, inductive coupling
18000-3	Parameters for Air Interface Communications at 13.56 MHz	HF, two modes (both require a license from the IP owner) Mode 1: 105.94 kbps from tag to reader Mode 2: 423.75 kbps from tag to reader Passive, both use inductive coupling, FDX
18000-4	Parameters for Air Interface Communications at 2.45 GHz	Microwave, two modes Mode 1: Passive Mode 2: Semi-passive, tags talk first Passive, backscatter, HDX
18000-5	Withdrawn	Withdrawn (was for 5.8 GHz)
18000-6	Parameters for Air Interface Communications at 860 to 930 MHz	UHF, two tag types

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Standard	Title	Description
		Type A: Pulse interval encoding, Aloha anti-collision Type B: Manchester encoding, Binary Tree anti-collision Passive, backscatter, HDX Reader to tag uses biphase space encoding for both tag types (Type C: ISO may adopt EPC Gen2 as the type C tag in late 2005)
18000-7	Parameters for Air Interface Communications at 433 MHz	UHF, long range Read/write, active, HDX

3.6.3. IS0 15693 Vicinity Smart Cards

The ISO 15693 standard was originally intended as a specification for "vicinity cards." These contactless smart cards are typically used for access control but have also been used for many other applications, including supply chain and asset tracking. This standard defines one type of tag, which should be the size and shape of a credit card, inductively coupled, using ASK from reader to tag and ASK or FSK from tag to reader. ISO 15693 also defines an anti-collision procedure called the Slot Marker Method (more on that in Chapter 4) and two encoding methods. The method intended for fast reads is called "1 of 4," and the method intended for long range is called "1 of 256." Some vendors sell unique IDs, but these are unique only to the vendor; they are not globally unique as in ISO 18000 or EPC.

The new EPC Gen2 standard uses ISO 15693 IDs to identify tags. Think of the tag for a moment as an electronics product from a manufacturer rather than as a data carrierthis Tag ID (TID) is assigned by the tag manufacturer to identify its product. The tag then carries an Object ID (OID), which is the EPC for the object to which the tag is attached.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

3.7. Summary

If you have an understanding of the different types of tags, choosing a tag for a particular applicatior becomes easier. Tags with durable physical characteristics are available for harsh environments, and paper tags are available for controlled environments where the application calls for rapid, automated tag application. The air interface determines which tags may communicate with which readers, and the type of coupling defined for that air interface determines how far away a reader can be and still read a tag. Tags come in a variety of data storage capacities, and some even have additional information processing capabilities, but these additional capabilities come at a cost. By studying the standards, we also now know what EPC and ISO tags are and why (for now) they may need different readers.

With these characteristics and capabilities in mind, we can talk about guidelines for selecting the best tag for your application. We've only covered the essentials in this chapter, and the actual process of selecting tags for a given application still usually requires extensive testing and the help of a skilled RF engineer. However, here are some general rules of thumb to help match tags to applications:

- Use smart labels for automated application in a warehouse, but use PVC or glass for tougher environments.
- Use passive tags for the lowest cost, and semi-passive or active tags only as necessary for additional capabilities or greater read range.
- Use LF/HF tags for individual items.
- Use UHF tags for shipping units such as pallets.
- Use microwave tags for vehicles and long-distance reads.
- Where possible, to reduce cost, store only an identifier on the tag and look up the rest of the information. More storage capacity is more expensive.
- Follow the standards where you can, and watch what the largest adopters are doing.
- As always, don't reinvent the wheel!

This has been a complex chapter because the tag options are so numerous. As the Compliance era matures, the options in RFID systems will stabilize to the point where we can choose from a limited number of options with confidence that the elements will match.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Chapter Four. Tag Protocols

In the previous chapter, we discussed the "air interface ," the bedrock of tag communications, and the circuit designer's view of tag and reader interaction. In this chapter, we'll examine the protocols readers and tags use to exchange messages over that air interface as well as take a more detailed look into the information stored on tags.

The Jargon File,[*] a dictionary of technical jargon, defines a *protocol* as:

[*] Eric S. Raymond, "The Jargon File," v3.0.0.0, July 1993, http://catb.org/~esr/jargon/.

> A set of formal rules describing how to transmit data, especially across a network. Low level protocols define the electrical and physical standards to be observed, bit- and byte-ordering, and the transmission and error detection and correction of the bit stream. High level protocols deal with the data formatting, including the syntax of messages, the terminal to computer dialogue, character sets, sequencing of messages, etc.

By this definition, the air interfaces described in Chapter 3 would be low-level protocols, while the protocols described here are the high-level protocols that define the actual syntax of messages and the structure of the dialog between a reader and a tag. The first part of this chapter covers some important terms and concepts for understanding tag protocols and explains more about the relationship between bar code standards and tag encodings, which should be helpful for developers building EPC applications. (These same encodings appear again in the APIs in Chapters 6 and 7.) We then discuss various types of singulation and anti-collision procedures in depth to illustrate how a reader can recognize a tag across a crowded room. The chapter closes with sections on tag features for security and privacy and some tips on troubleshooting tag communications.

The singulation protocols may seem esoteric at first, but they have been the focus of heated debate during recent months and are in many ways responsible (for better or worse) for critical decisions on the timing of technology purchases and for adoption of RFID technologies across many industries. In particular, the publication of the Gen2 specification in early 2005, was one of the most anticipated events in RFID in early 2005.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.1. Protocol Terms and Concepts

Technical jargon develops around any new technology, and RFID is no exception. Some of these terms are quite useful, serving as a convenient way to communicate concepts needed to describe other concepts that will appear in the pages that follow. These terms include:

Singulation

This term describes a procedure for reducing a group of things to a stream of things that can be handled one at a time. For example, a subway turnstile is a device for singulating a group o people into a stream of individuals so that the system may count them or ask them for access tokens. This same singulation is necessary when communicating with RFID tags, because if there is no mechanism to enable the tags to reply separately, many tags will respond to a reader at once and may disrupt communications. Singulation also implies that the reader learns the individual IDs of each tag, thus enabling inventories. Inventories of groups of tags are just singulation that is repeated until no unknown tags respond.

Anti-collision

This term describes the set of procedures that prevent tags from interrupting each other and talking out of turn. Whereas singulation is about identifying individual tags, anti-collision is about both regulating the timing of responses and finding ways of randomizing those responses so that a reader can understand each tag amidst the plethora of responses.

Identity

As we discussed in Chapter 1, an identity is a name, number, or address that uniquely refers to a thing or place. "Malaclypse the Elder" is an identity referring to a particular person. "221b Baker Street London NW1 6XE, Great Britain" is an identity referring to a particular place, just as "urn:epc:id:sgtin:00012345.054322.4208" is an identity referring to a particular widget.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.2. How Tags Store Data

The high-level tag communications protocols know about the ID types that can be stored on a tag and so know, at a general level, how data is stored on the tag. However, since a reader only communicates with a tag via the air protocol, the actual physical layout of memory on a tag is left to the tag manufacturer. A discussion of the physical layout is beyond the scope of this book, but the logical or apparent layout of tag memory is specified in the EPC Class 0 and Class I (Generation 1) standards. Figure 4-1 shows that layout.

Figure 4-1. Tag data layout

The CRC is a checksum (described in more detail in the sidebar "CCITT-CRC"), the EPC is the ID on the tag, and the password is the "kill code" to disable the tag. Note that for a tag to be EPC-compliant under Class 0 and Class I Generation 1 standards, it must never transmit the password under any circumstancesnot even in response to proprietary diagnostic commands used only by the manufacturer.

This password is the same password described previously as a means of disabling(destroying) tags. (Under Gen2, this changed; we will describe Gen2 in more detail in the "EPC UHF Class I Gen2" section of this chapter.)

Version 1.1, Revision 1.26 of the EPC Tag Data Standardsdescribes EPC as a "meta-coding scheme" because it allows existing identifiers to be encoded as EPC identifiers, as well as allowing for the creation of completely new identifiers. This standard defines one encodingfor General Identifiers (GIDs), which is intended for creating new identification schemes, and five specific encodingscalled System Identifiers for particular uses. The System Identifiers are based on existing GS1 (EAN.UCC) identifiers.

Table 4-1 describes each of these special encodings and gives examples of their use. The application identifier is a number that can be used as a prefix in a bar code to indicate which identifier the number represents.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

CCITT-CRC

A Cyclic Redundancy Check (CRC) is a way of verifying that a block of data has not become corrupted. The sender of the data block calculates a value by treating the whole block as one large number and dividing it by a number called the *CRC polynomial* . The remainder of this operation is the CRC. The sender sends this CRC along with the data, and the recipient uses the same method to calculate a CRC over the data block for comparison. If the CRC from the sender doesn't match the CRC calculated by the recipient, the recipient may request that the data be resent. To generate CRCs, EPC protocols use the CCITT-CRC polynomial, which happens to be the same polynomial used for error detection in most disk drives and in the venerable XMODEM file transfer protocol. This protocol is a 16-bit CRC that uses the polynomial $x^{16} + x^{12} + x^5 + 1$. It catches about 99.998 percent of errors and can more or less be easily implemented. The pseudocode for calculating CRCs is as follows.

First, calculate the hex value for the polynomial. We do this by counting down from 15 (because this is a 16-bit CRC) and marking a 1 for each power that appears in the polynomial. For each power that does not apply in the polynomial, we mark a zero. This means we have a 1 at the 2^{12} position and a 1 at the 2^5 position. Since the polynomial ends with a +1, we then add a 1 to the end of our number, yielding 0001000000100001, or 1021 hex. We will call this number `CCITT` in the Java-like pseudocode below:

```
CCITT = x1021;
bytes[] = toBytes(x30500046501CC50000001070);
crc = 0xFFFF; // Set it to ones to start
i = 0;
while(i < bytes.length) {
  if (1 == (crc & 0x80) xor bytes[i])) //MSB of crc
    crc = (crc << 1) ^ CCITT;
  else
    crc = crc << 1;
 }
 print crc;
```

This is, of course, a terribly inefficient way of doing the calculation. The EPC also uses a CRC-5 for short queries, but a CRC-5 would be too small to work for the tag IDs. The Gen2 specification includes example code and a circuit diagram for implementing CCITT-CRC and CRC-5 in hardware. For a clever, fast CCITT-CRC software implementation, see Ritter.[a]

[a] a T. Ritter, "The Great CRC Mystery," *Dr. Dobb's Journal of Software Tools*, 11 no. 2 (1986): 2634, 7683. http://www.ciphersbyritter.com/ARTS/CRCMYST.HTM.

Table 4-1. Application identifiers

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Application identifier	Identifier	Name	Intended use	Example
(21)	SGTIN	Serialized Global Trade Item Number	Item tracking	Individual tires
(00)	SSCC	Serial Shipping Container Code	Shipping containers	An ISO/intermodal container
(414)	GLN	Global Location Number	Locations	An individual antenna in a warehouse
(8003)	GRAI	Global Returnable Asset Identifier	Rental and library items	A rentable hand-truck
(8004)	GIAI	Global Individual Asset Identifier	Asset tracking	An office chair
	GID	General Identifier	Individual IDs	New identification schemes

The GID is a special case. Unlike the System Identifiers, it does not simply adapt an existing identifie to work with EPC tags. The GID defines a header and three fields: the General Manager Number, the Object Class, and the Serial Number. The General Manager Number is assigned by EPCglobal to some company or entity and is guaranteed to be unique. An entity granted a General Manager Number is then responsible for assigning unique values for the dependent Object Class and Serial Number fields These dependent fields only need to be unique for a particular General Manager; different General Managers can use the same Object Class numbers and Serial Numbers without conflict.

The GS1 System Identifiers each serve a particular purpose in identifying portions of the global trade network, as you can see in Figure 4-2. Imagine a pallet of automobile parts bound overseas from the factory. Separate Serialized Global Trade Item Numbers (SGTINs) might identify each individual part while yet another SGTIN identifies the pallet composed of those parts. A railway dock door through which the pallets travel at the factory might be identified with a specific Global Location Number (GLN) . The shipping container, identified with a Serialized Shipping Container Code (SSCC), would contain this pallet (among others), as well as a set of reusable shipping braces that belong to the factory. Each brace could be identified using a Global Returnable Asset Identifier (GRAI)number. Having standards for all of these pieces of information allows for unprecedented information sharing among trading partners throughout the supply chain. For example, there are currently hundreds of different standards for location information alone, so the benefits and challenges of moving to a single system, such as GLNs, are equally great. None of the global identityschemes (such as the GTIN) are strictly or even chiefly concerned with RFID, but their existence and momentum are important to the successful adoption of RFID on a massive scale.

Remember that "identity" has a special meaning here, as described in the previous section Protocol Terms and Concepts." EPC defines three layers of identity: the Pure Identity Layer, the Encoding Layer, and the Physical Realization of an Encoding. A *pure identity* is an abstract name or number used to identify something. This identity remains the same regardless of the technology used to attach it to an item. An *encoding* is a procedure for combining the pure identity with syntax-specific information, such as a filter value or checksum, and then rendering that information into the syntax. A single pure identity could be rendered as a bar code encoding, an RFID tag encoding, or an EPC Uniform Resource Identifier (URI) printed on a piece of paper. The physical realization of an encoding is a particular transformation of that encoding that allows it to be stored as a bar code, written into a tag's memory, or physically realized through some other technology.Figure 4-3 gives an example of

downloaded from: lib.ommolketab.ir

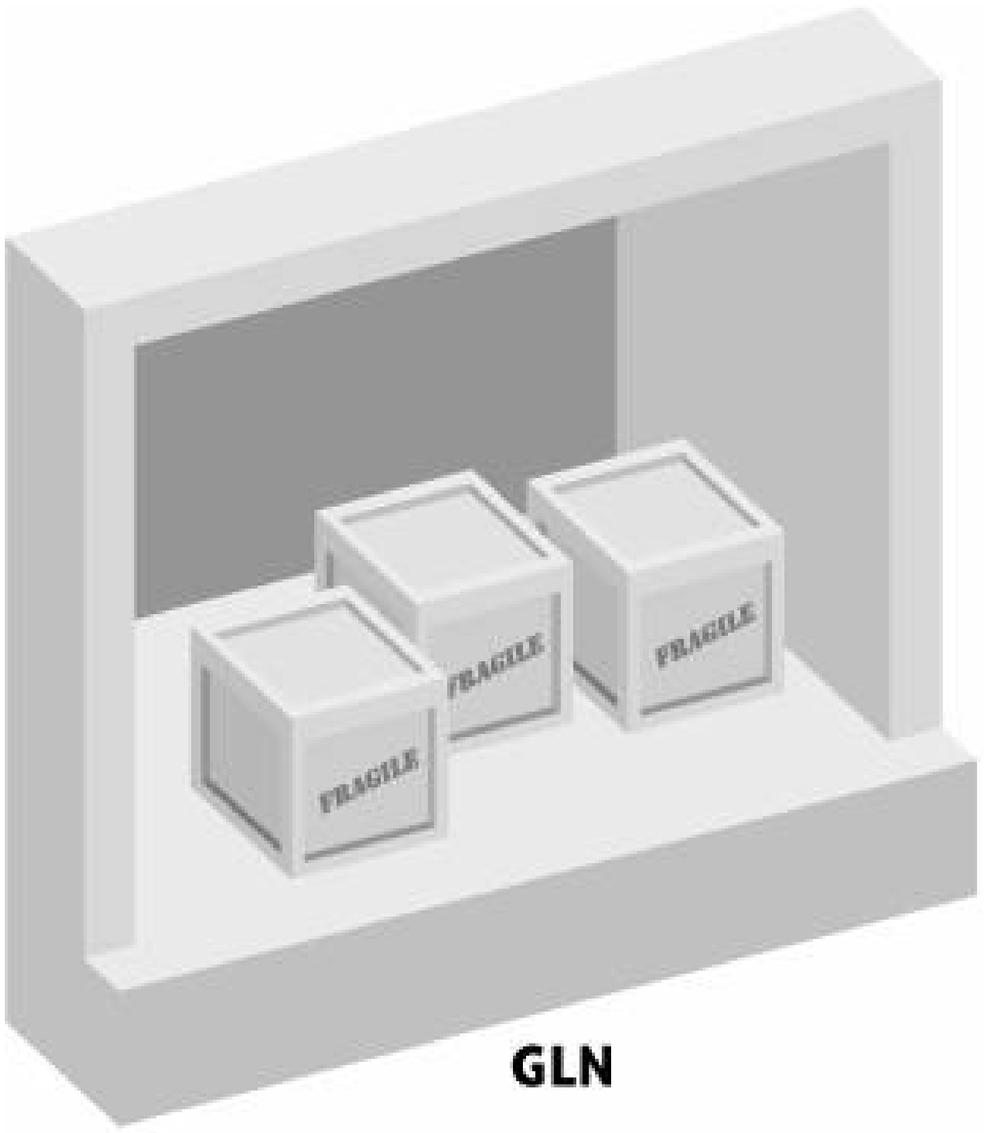
downloaded from: lib.ommolketab.ir

encoding a pure identity as a bar code or an EPC tag.

Figure 4-2. Some system identities in use

downloaded from: lib.ommolketab.ir

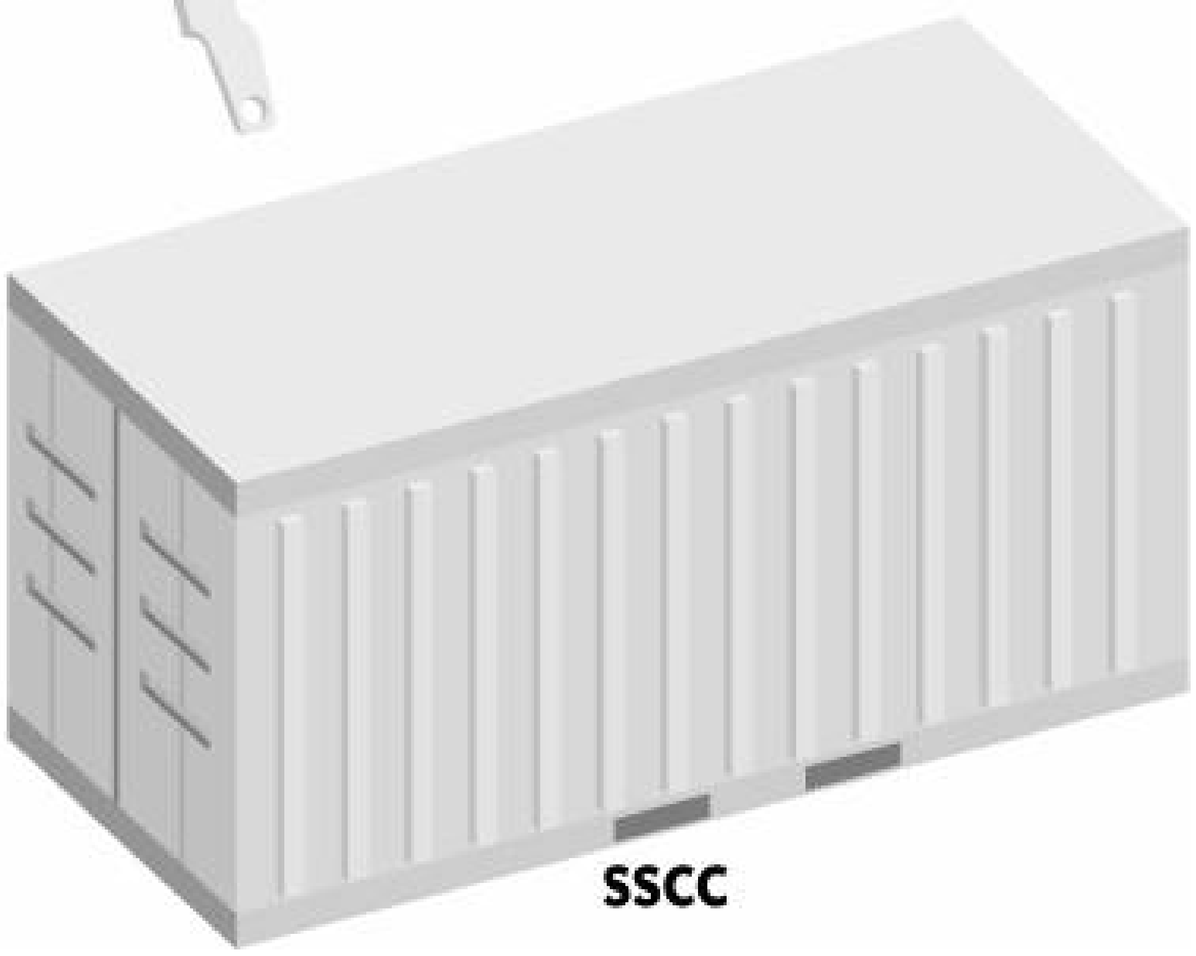
downloaded from: lib.ommolketab.ir



GLN

GTIN

GRAI

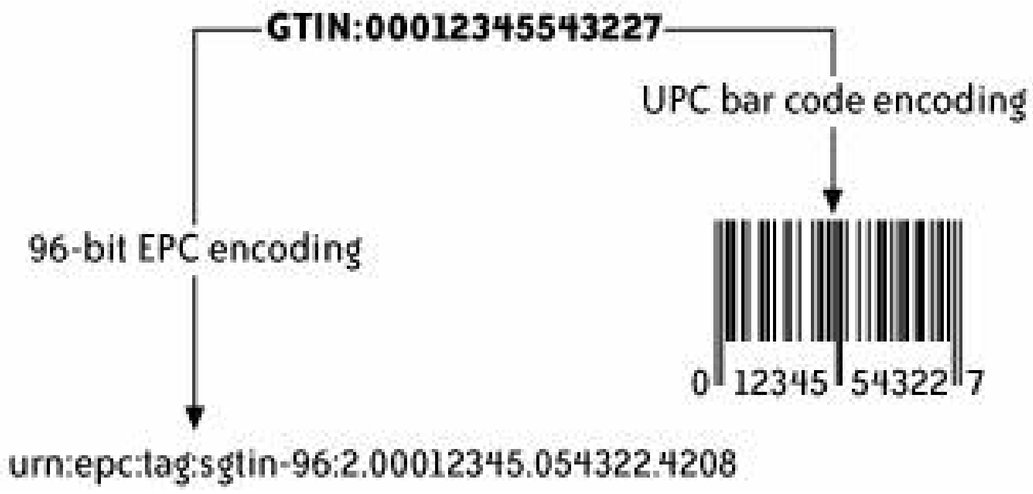


SSCC

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Figure 4-3. Encoding a pure identity



Notice that the GID is used as the fundamental encoding for EPC tags. Other encodings take an identity, which might be encoded in another form (such as a bar code), and convert it to a form that fits into a GID with a syntax similar to "Header.GeneralManagerNumber.ObjectClass.SerialNumber." This requires the EPC to take some liberties with the GS1 SGTIN. For example, while the guidance from GS1 is that the GTIN should never be parsed and treated as separate fields, an EPC must have a General Manager Number and an Object Class for the Object Naming Service (ONS) to work, so the EPC-SGTIN divides the GTIN. In this case, an SGTIN is an example of a pure identity. The EPC-SGTIN is the encoding, and the bits on an EPC tag representing the EPC-SGTIN are the physical realization of that encoding.

4.2.1. GS1 SGTIN Encoding

EPC readers and RFID middleware present tag data according to its EPC encoding. Interfacing with a reader or middleware requires at least a cursory understanding of tag encodings for debugging code. An understanding of these encodings is also required to define the events and filters introduced in Chapter 2, which we will describe in more detail inChapters 6 and 7. The SGTIN is a good example of an identity and its encoding, so we'll use it as an example in this chapter.

Note: For details on GLN, GRAI, and other encodings from Table 4-1, please see Appendix A.

EPC-SGTIN is an extension of the GS1 GTIN that assigns Company Prefixes and Item References for use in identifying particular classes of object. The common 12-digit UPC and 13-digit EAN bar codes are a subset of the GTIN. These types of codes are being merged with the 14-digit GTIN in 2005 by prepending zeros to the existing codes. The GTIN doesn't have an individual item serial number, so the SGTIN appends a serial number, the value of which is assigned by the General Manager.

Figure 4-4 shows a typical UPC bar code. To convert this UPC into an EPC and store it on an RFID tag, we must first convert it to a GTIN. This bar code has an Indicator Digit (0), a Company Prefix (12345), an Item Reference (54322), and a check digit (7). To convert this to a GTIN, we take the entire code as a string and add two zeros to the beginning, yielding a GTIN of 00012345543227. Notice that our Company Prefix has now become 00012345, an 8-digit number. We will then convert the GTIN to an SGTINwhich allows us to track individual itemsby adding a Serial Number (4208).

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Figure 4-4. UPC bar code

To represent a pure identity, the EPC uses a URI expressed in the URN notation. For an SGTIN, this notation is:

```
urn:epc:id:sgtin:CompanyPrefix.ItemReference.SerialNumber
```

This notation includes only the information necessary to tell one item from another; it does not include the GTIN check digit or a filter value. The Item Reference here is actually the Indicator Digit plus the Item Reference from the GTIN (Figure 4-5). Our example would thus be encoded as:

Figure 4-5. Conversion of GTIN to SGTIN

```
urn:epc:id:sgtin:00012345.054322.4208
```

To express an identity encoded as SGTIN-96, a value dependent on the tag type, the EPC uses a different format for the URN:

```
urn:epc:tag:sgtin-96:FilterValue.CompanyPrefix.ItemReference.SerialNumber
```

Using this notation, our example would be encoded as:

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

```
urn:epc:tag:sgtin-96:2.00012345.054322.4208
```

The steps to encoding a 96-bit EPC to a binary string are:

1. Find the appropriate header for the identity type.
2. Look up the partition value based on the length of the Company Prefix.
3. Concatenate the 8-bit header, 3-bit filter, and 3-bit partition fields.
4. Append to this the Company Prefix and other fields appropriate to the identity (Item Reference and Serial Number for an SGTIN).
5. Calculate the CRC and append the EPC to the end of the CRC.

4.2.1.1. Find the header

A header identifies each type of identity and its specific encoding.Table 4-2 shows examples of encodings for SGTIN for 96-bit tags and 64-bit tags. Notice how the 64-bit header starts with a one and is only two bits. Only 64-bit headers start with one.

Table 4-2. SGTIN header values

Type	Header
SGTIN-96	0011 0000
SGTIN-64	10

4.2.1.2. Find the partition

We have 96 bits to work with. Of these bits, the encoding designates 44 for the Company Prefix and Item Reference. Different companies have prefixes of different lengths. The partition number tells us how many bits we have left for the Item Reference field based on how many bits the Company Prefix takes up. To see how many bits Company Prefixes of various lengths take up, look at the Company Prefix section in Table 4-3. Our example of the Company Prefix of 00012345 (eight digits long) corresponds to partition 4 in Table 4-3. From the other columns in this row, we know that we will need 27 bits to encode the Company Prefix on the tag and will have 17 bits to encode the Item Reference.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Table 4-3. SGTIN-96 partition values

Partition	Company Prefix		Item Reference	
	Bits	Digits	Bits	Digits
0	40	12	4	1
1	37	11	7	2
2	34	10	10	3
3	30	9	14	4
4	27	8	17	5
5	24	7	20	6
6	20	6	24	7

4.2.1.3. Concatenate the header, filter value, and partition

Notice the field name "Filter Value." This is not a part of the SGTIN, but is instead a way of quickly selecting EPCs based on common logistical types. For example, a filter value of 1 might be used for smaller individual items, while 3 might be used for something large that ships individually, such as a refrigerator. Table 4-4 lists the SGTIN filter values. We will use a filter value of 2 in our example, to indicate a "Standard Trade Item Grouping" such as a pallet or carton. All encodings support a filter value of 0, and SGTIN and SSCC also support a 1, meaning "undefined." SSCC defines 2 as "Logistical/Shipping Unit." More filter values may be defined in the future.

Table 4-4. SGTIN filter values

Filter value	Binary code	Meaning	Example
0	000	Unspecified	Unspecified
1	001	Retail Consumer Trade Item	A single shaving razor
2	010	Standard Trade Item Grouping	A carton or pallet of razors
3	011	Single Shipping/Consumer Trade Item	A refrigerator (large, shipped alone)
4	100	Reserved	Reserved for future use
5	101	Reserved	Reserved for future use
6	110	Reserved	Reserved for future use
7	111	Reserved	Reserved for future use

Since we are building an SGTIN-96, the standard header value is 0011 0000, or 30 hex. Building the

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

SGTIN-96 is simply a matter of concatenating bits, starting with the header as the most significant bi (MSB), followed by the bits for the filter (3 bits) and the partition (3 bits). This yields the following (shown in 4-bit nibbles, with the last incomplete):

```
0011 0000 0101 00
```

4.2.1.4. Append the Company Prefix, Item Reference, and Serial Number

We now append the Company Prefix to the previous bits by specifying the 27 bits available for it so that they represent the appropriate value. This yields:

```
0011 0000 0101 0000 0000 0000 0001 1000 0001 1100 1
```

which can be represented in hex as 305000181C, with an extra bit set to 1.

We then append the Item Reference in 17 bits to the end of this number, which yields 305000181C B50C plus two bits set to 10.

Now we append the Serial Number, which is 38 bits long. This yields 305000181CB50C8000001070, which is a 12-byte, or 96-bit, number.

4.2.1.5. Calculate the CRC and append the EPC to it

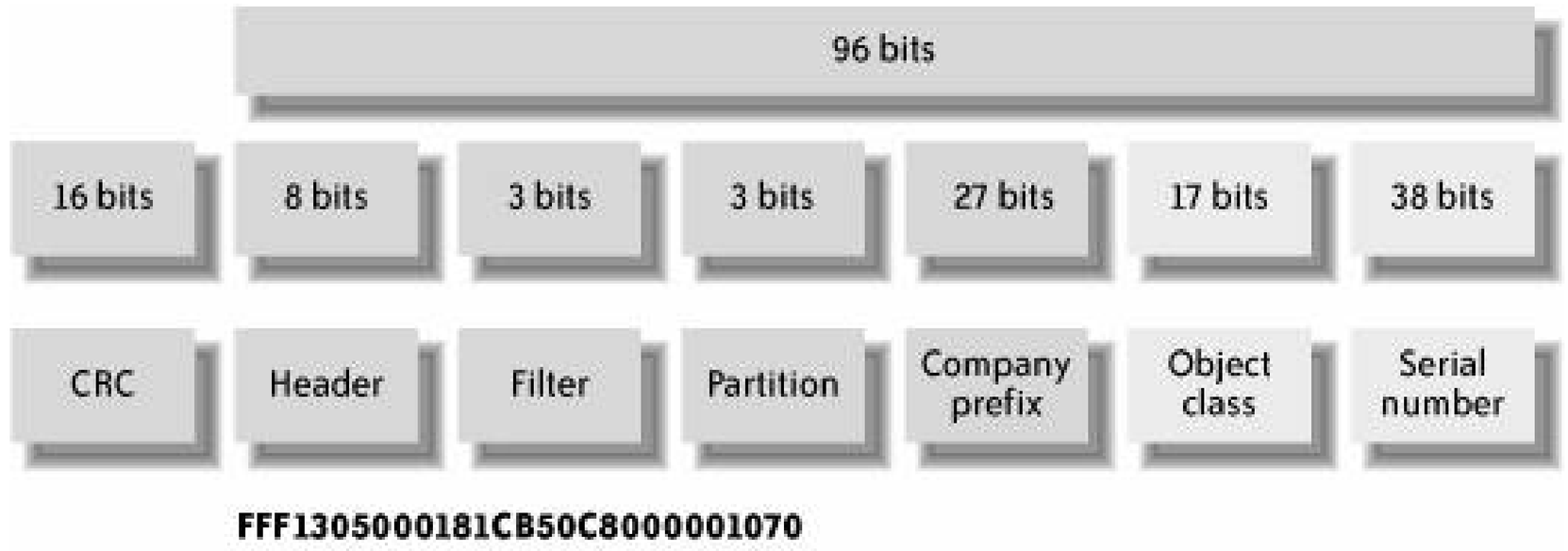
This value is then stored in the tag along with a 16-bit CRC (CCITT-CRC), which would be FFF1 in this case. The value with the CRC would be FFF1305000181CB50C8000001070.

Figure 4-6 shows the parts of the encoding again.

Figure 4-6. Encoding for an SGTIN-96 with a partition value of 4

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



For each system identity, the specification describes an alternative encoding for 64-bit tags. To help fit the identity into a 64-bit encoding, the Company Prefix is removed and a Company Prefix Index takes its place in the encoding. This index is an offset into a Company Prefix table. A Company Prefix Index is supplied as needed by GS1 to those entities that intend to use 64-bit tags. This table is limited to 16,384 entries, and this encoding scheme is intended as a temporary solution until the industry universally adopts tags that are 96 bits or larger. Note that the SGTIN-64 encoding is the only currently defined header that contains a 1 in the MSB, allowing it to be only two binary digits long (11 is reserved for other 64-bit encodings).

For headers, partition tables, and field types of the other EPC-defined 96-bit encodings, see Appendix A.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.3. Singulation and Anti-Collision Procedures

Our next topic concerns the way a reader and a tag use the air interface. There are many different ways for readers and tags to communicate, but the different methods can all be broadly categorized as Tag Talks First (TTF) or Reader Talks First (RTF). Logically, it would seem simplest for a tag arriving on the scene to announce its presence to all concerned. In practice, however, this is difficult unless tags are able to negotiate among themselves which tag will speak first. Some high-end active tags use TTF communications protocols, but the new crop of inexpensive smart labels and other passive tags exclusively use RTF protocols. In this section, we'll examine the most common of these protocols for RFIDSlotted Aloha, Adaptive Binary Tree, Slotted Terminal Adaptive Collection, and the new EPC Gen2 specification. In the later section "Tag Features for Security and Privacy," we'll look at how features of these protocols can affect security and data integrity in an application.

4.3.1. Slotted Aloha

Slotted Aloha is derived from a procedure known simply as "Aloha," which was originally developed ir the 1970s by Norman Abramson of Aloha Networks in Hawaii for packet radio communication. Aloha was the inspiration for the Ethernet protocol, and a variation of this procedure is still used for satellite communication as well as for ISO 18000-6 Type B and EPC Gen2 RFID tags.

Aloha itself is as simple as an anti-collision procedure may reasonably be and doesn't really include the concept of singulation in the usual sense. With this procedure, tags begin broadcasting their IDs as soon as the reader's field energizes them. Each tag sends its entire ID and then waits for a pseudorandom period of time before broadcasting again. The reader simply receives the IDs, depending on chance to ensure that each tag will eventually broadcast during a period when all other tags are quiet. The reader doesn't respond to the tags in any way. The advantages of this procedure are speed and simplicity. Tag logic is minimal, and with such a low protocol overhead, this is the top performer for read rate where only a few tags are present.

However, adding tags means reducing the chances that any given tag will happen to transmit during a quiet moment. This means waiting for tags to transmit again and again until a clear transmission comes throughan approach that, depending on the length of transmission, becomes unworkable for most item-tracking implementations at about 8 to 12 tags. Fortunately, Slotted Aloha greatly improves the scalability of the protocol by adding the concept of singulation and requiring tags to broadcast only at the beginning of a particular time slot, which cuts the chances of collision considerably. This along with comparisons of received signal strength by the reader, has led to a protocol that some have claimed is capable of reading almost 1,000 tags per second. The cost of this optimization is, as usual, increased complexity.

Slotted Aloha uses three commands to sort out tags: REQUEST, SELECT, and READ. The first command is a REQUEST, which provides a timing mark to any tags currently in range. The REQUEST command also indicates how many slots are available for use by the tags. The tags each select one o the offered slotsthat is, based on a number of options offered by the reader they randomly select a length of time to wait before responding to the REQUEST. The tags then broadcast their IDs at their

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

chosen slots. The reader, upon receiving a clear ID with no collisions in a particular time slot, issues a SELECT containing that ID. Only the tag whose ID matches this ID responds. The reader then issues a READ command, to which the tag responds by sending whatever data it has to share with the reader. (In a nonsecure system, there may be no further data .) With that conversation over, the reader then sends another REQUEST command. Figure 4-7 shows the state transitions of the reader during this conversation, while Figure 4-8 shows the state transitions of the tag.

Fewer slots means faster reads, but with many tags, more slots means fewer collisions. Various implementations of Slotted Aloha optimize the number of slots in different ways. The reader may increase the number of slots in a REQUEST following a collision and continue to increase them in successive REQUEST commands until a clear ID transmission occurs without collision. A reader may also use a BREAK command to inform waiting tags that a clear transmission has occurred and that they should not transmit. In some implementations, a tag will enter a SLEEP (also called DORMANT or MUTE) state for some period following a successful read, thereby allowing any remaining tags more of a chance to be selected. The dramatization in the following sidebar may help further your understanding of the Slotted Aloha protocol.

Figure 4-7. Slotted Aloha reader state diagram

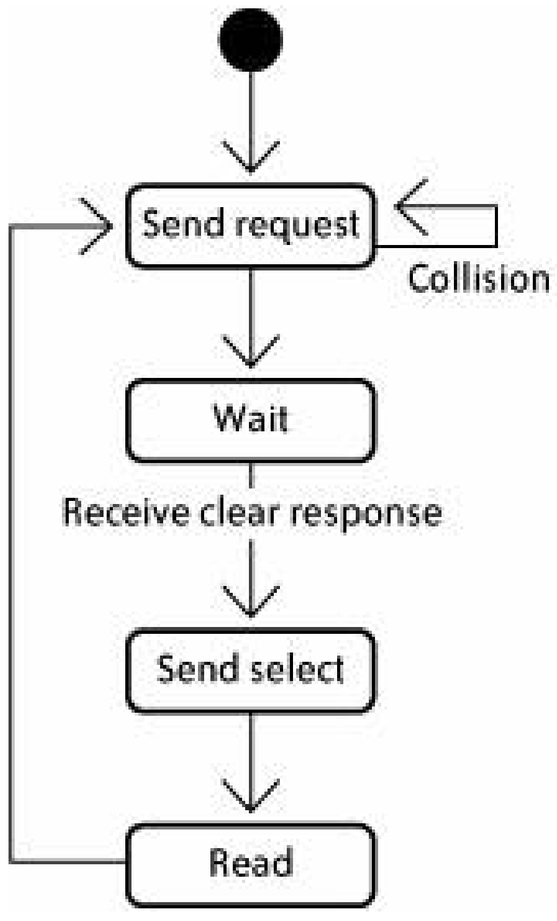
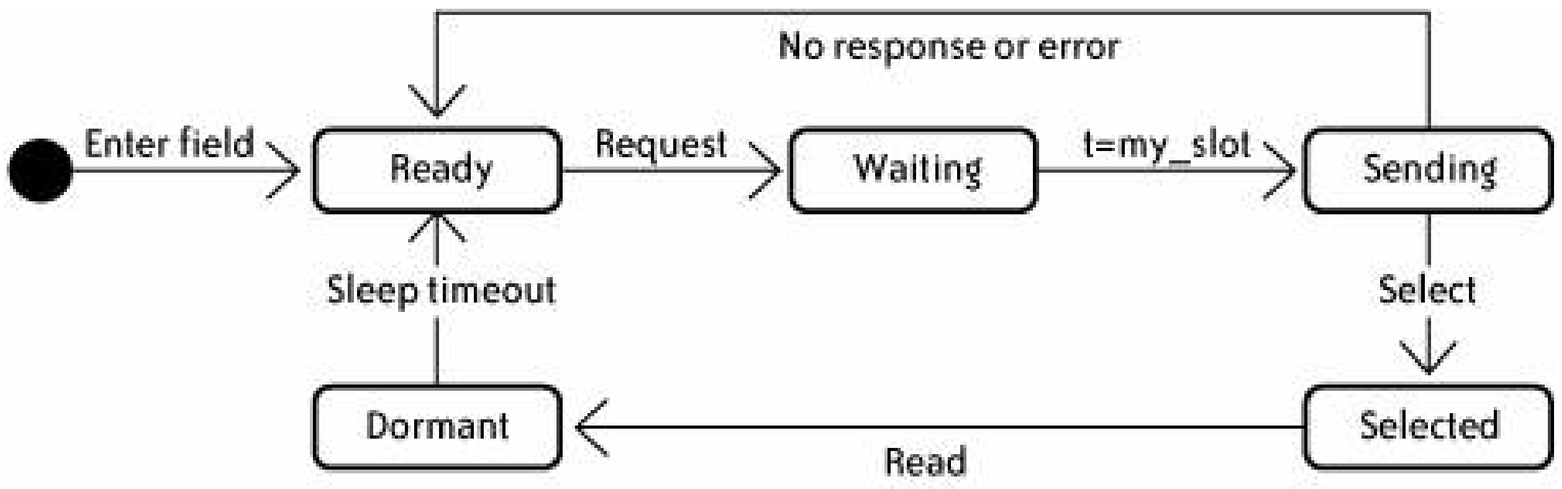


Figure 4-8. Slotted Aloha tag state diagram

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



Tag 5 puffs out his cheeks and sits in silence as Tags 1 and 2 roll their dice.... Rather than using timed intervals, the reader may indicate the end of a slot using a signal, called a "slot marker." Anti-collision protocols that use these signals are called Slot Marker protocols. A Slot Marker implementation of the Slotted Aloha protocol works much the same as that described above, except that the end of a slot is indicated by a signal rather than by the expiration of a set time. This allows some slots to be longer than others so that there is a better chance of a read in each slot.

4.3.2. Adaptive Binary Tree

EPC Class 0 and Class I Version 1.0 (Generation 1) UHF tags use a slightly more complicated approach to singulation and anti-collision known as an Adaptive Binary Tree procedure. As the name implies, this procedure uses a binary search to find one tag among many. Most of us are familiar with binary trees from other domains, but for the sake of clarity, we will revisit the basics here. Then we'll explain some of the nuances of doing a binary tree search using a query/response approach similar to the previous description of Slotted Aloha. Unlike with Slotted Aloha, with this protocol tags respond immediately. The EPC specification for the air interface of UHF tags uses two separate subcarriers for 1s and 0s in tag responses. Because the protocol doesn't care how many tags responded with a 1 or a 0 and only cares that one or more tags responded, this protocol neatly sidesteps most tag response collisions. The protocol does require that only one reader be involved, though, which may require careful configuration of readers that are near each other.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Slotted Aloha

A dramatization may help to explain the Slotted Aloha protocol. In this case, we have a single reader and three tags using a protocol that includes BREAK and SLEEP commands. Imagine you have four actors: one plays the role of Reader, and the others play Tag 1, Tag 2, and Tag 5.

We begin in an empty room. A large clock hangs on one wall, showing the current time in milliseconds. Reader stands on a podium and tags enter from a door, stage left.

Reader: Is anyone there? It is now time t. I have slots at t+10, t+20, t+30, t+40, and t+50. Please respond.

(Tags 1, 2, and 5 enter.)

Reader: Is anyone there? It is now time t. I have slots at t+10, t+20, t+30, t+40 and t+50. Please respond.

Tag 1: (rolls percentile dice and selects t+40)

Tag 2: (rolls percentile dice and selects t+20)

Tag 5: (rolls percentile dice and selects t+10)

(The clock chimes t+10.)

Tag 5: Five!

Reader: Quiet, all of you! I hereby select Five.

(All tags except 5 shake their heads in disgust but remain silent. Five grins foolishly.)

Tag 5: That's me! I'm Five!

Reader: Have you any data to share, Five?

Tag 5: FFF1305000181CB50C8000001070, I'm Five!

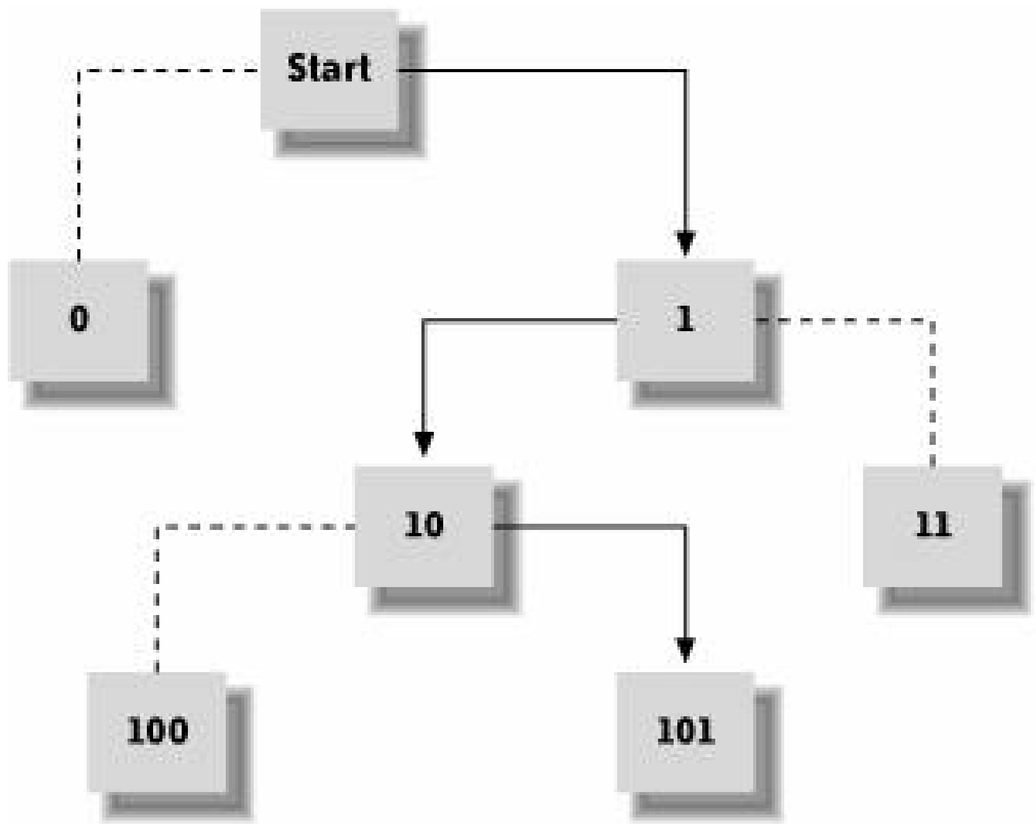
Reader: Yes, well shut up now, Five... Is anyone there? It is now time t. I have slots at t+10, t+20, t+30, t+40, and t+50. Please respond.

An easy way to imagine traversing or "walking" a binary tree is to imagine trying to guess a number made up of 1s and 0s by making guesses for each digit. When we start we have no information, so we ask, "Is the first digit 1?" If the answer is "yes," we can add that to our string of digits and ask, "Is the next digit 1?" If the answer is "no," we can add a 0 to our string of numbers. This question and answer repeats for each digit until we know the whole number.Figure 4-9 shows the "tree." The arrows show the correct digits at each step.

Figure 4-9. A binary tree

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



Now apply this strategy to finding one tag among many by the bits in the tag's ID. As before, we start with no information. The reader sends out a query that amounts to asking, "Does any tag have an ID with a 1 as the most significant bit?" All of the tags that answer "no" then cease to respond, while tags that answered "yes" are asked a similar question about the next bit. The tags continue to be narrowed down in this fashion until only one tag answers. In this way, a reader can typically narrow down the possibilities to one tag without going through the entire ID, although in the worst casea population of sequential IDseach search will necessarily go to the least significant bit (LSB).

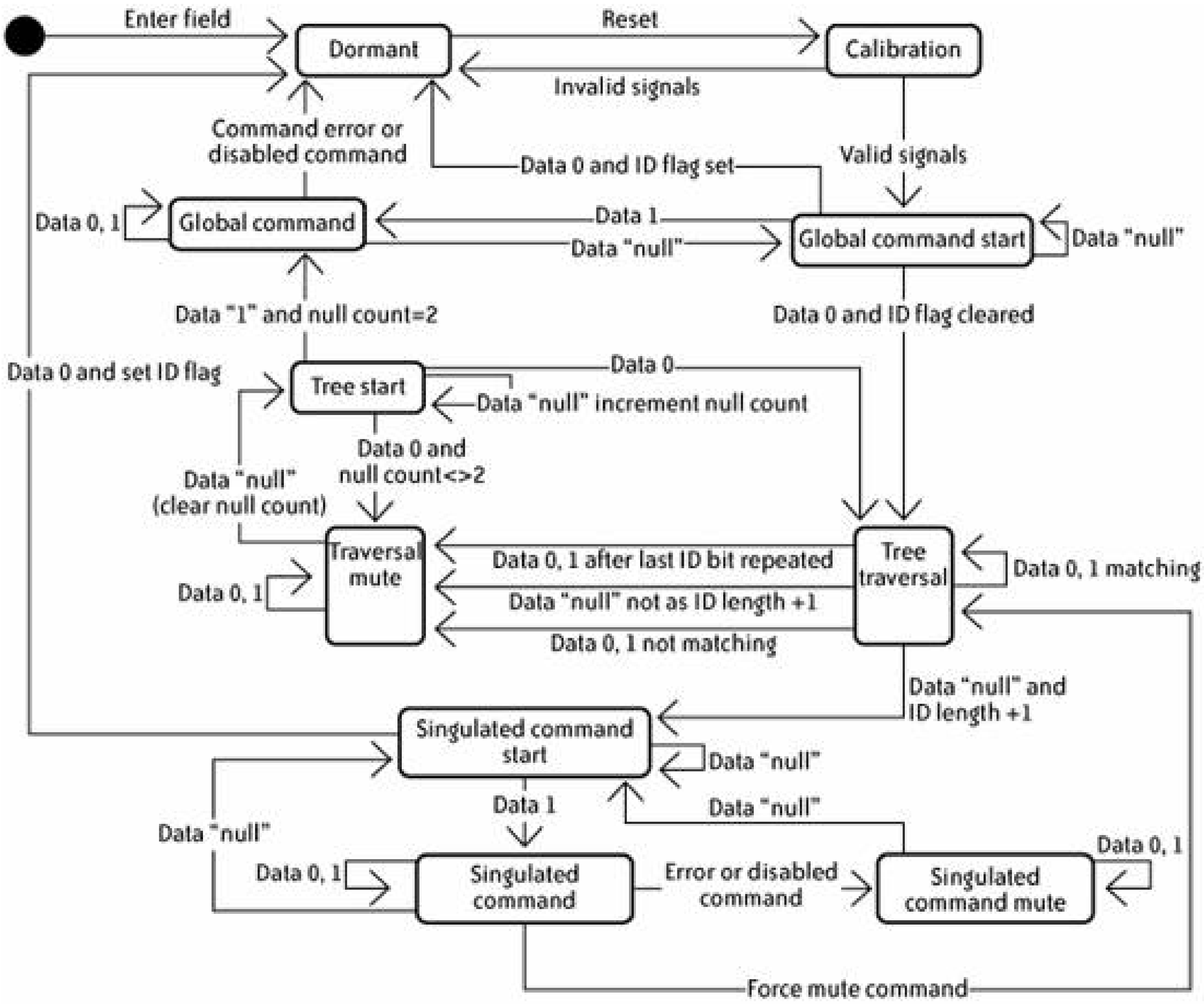
This "worst case" would be a frequent problem, since in most cases tags are applied sequentially, but the EPC standard allows for three different IDs for a tag: ID0, ID1, and ID2. ID2 is the EPC code, ID1 is a pseudorandom number that is set at the factory, and ID0 is a pseudorandom number that is generated as needed. If a reader asks each of the tags to generate a new ID0 and then singulates based on that ID, the probability of sequential IDs becomes very small.

With the Adaptive Binary Tree protocol, the interaction between the reader and tag is necessarily more complex than it is with Slotted Aloha. The protocol uses a state machine made up of four interdependent sections. The first is a set of states associated with global commands, including the all-important Dormant state. The next section is a state for calibrating communicationsspecifically, synchronizing the time-keeping oscillators on the tags with the timing of the reader. Differences in manufacturing, age of components, and even temperature can affect timing of circuits enough that this calibration is critical to achieving reasonable read rates. The next set of states is concerned with walking the binary tree, and the last set of states is used for communicating with a tag once it has been singulated. Figure 4-10 shows the state machine.

Figure 4-10. Adaptive Binary Tree protocol state diagram

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



Here's a breakdown of the Adaptive Binary Tree protocol states:

Global states

The following are global states, which may be entered at almost any point:

Dormant

The Dormant state is the initial state a tag enters when powered. It is also the state a tag enters after it has been read.

Global Command Start

After a successful calibration, the tag enters the Global Command Start state and waits for a 1 or a 0 from the reader. A 1 sends the tag to the Global Command state, while a 0 sends the tag to the Tree Traversal state (unless the tag has already been read, in which case a 0 causes the tag to proceed to the Dormant state).

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Global Command

In this state, the tag is ready to receive and process commands that affect all tags or groups of tags that have not been singulated. Some commands, such as kill, are not available as global commands.

Calibration

Whenever a tag receives a "reset" message from the reader, it enters the Calibration state and expects oscillator and data calibration pulses from the reader. If any of these are invalid, the tag returns to the Dormant state.

Tree walking states

The following states occur while the protocol is traversing the binary tree itself:

Tree Start

In this state, a data null will increment the null counter by 1. A 0 will send the tag to the Tree Traversal state. A 1 will send the tag to the Traversal Mute state unless the null counter is at 2, in which case the tag will go to the Global Command state. This null trick is used to address groups of tags that match (or do not match) a partial singulation string.

Tree Traversal

When the tag enters this state it immediately sends its most significant bit. The reader then responds with a bit. If it matches the bit the tag sent, the tag will send the next bit, and so on. If the bit does not match, the tag goes to the Traversal Mute state and waits for a data null. If the singulation is using a pseudoID, every 10th bit is considered a "boundary bit." If the singulation is using the EPC itself, the boundary bit is the last bit of the EPC plus the CRC. At the boundary bit, the tag sends the bit as usual; if the reader confirms this bit, the tag then sends the same bit again in the case of the EPC or the next bit in the case of a pseudoID. If the reader responds with a 1 or a 0, the tag enters the Traversal Mute state. If the response from the reader is data null, the tag enters the Singulated Command Start state. If the tag receives a data null at any other time, it enters the Traversal Mute state.

Traversal Mute

A tag waits quietly in this state until it receives a data null, at which point it proceeds to the Tree Start state and resets the local null counter.

Singulated states

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

These states occur when a single tag remains after the binary tree has been walked:

Singulated Command Start

A tag enters this state from the Tree Traversal state after the last bit of its ID has been confirmed and it has received a null from the reader. Any further nulls are ignored, while a 1 will send the tag to the Singulated Command state. A 0 will set an identified flag, and the tag will go to the Dormant state. This identified flag indicates that the tag has been read.

Singulated Command

Here, the tag receives 8-bit commands from the reader. If an error occurs, the tag enters the Singulated Command Mute state.

Singulated Command Mute

In this state, a tag waits quietly until it receives a data null, at which point it proceeds to the Singulated Command Start state.

4.3.3. Slotted Terminal Adaptive Collection (STAC)

The STAC protocol is similar in many ways to Slotted Aloha, but some of the details of this protocol make it more complex and otherwise different enough to warrant separate treatment. STAC is defined as part of the EPC specification for HF tags. Because it defines up to 512 slots of varying lengths, it is especially well suited to singulation of large populations of tags. This protocol also allows for the selection of groups of tags based on matching lengths of EPC code beginning with the MSB. Because the EPC code is organized by Header, Domain Manager Number, Object Class, and Serial Number (in that order) from MSB to LSB, this mechanism can easily select only tags belonging to a particular Domain Manager or Object Class. Since HF tags are often used for individual item identification, this is especially useful if, for example, the application would like to know how many of the items on a mixed pallet are boxes of A4 paper.

Like Slotted Aloha, STAC uses slots. Figure 4-11 illustrates how the slots are used.

Figure 4-11. STAC slots

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

The F or "fixed" slot is always present and is always the same length. Numbered, variable-length slots follow. These slots must begin with a slot named "0," and there must be enough of these numbered slots to equal some power of two. The exact number of slots is selected and continually adjusted by the reader to most effectively balance the need for quick reads and few collisions. Fewer slots make for faster reads, while more slots make for fewer collisions.

STAC defines only a small set of states and commands, but the steps of the protocol require some explanation. Figure 4-12 shows the states and the commands that trigger transitions.

Note that whatever state a tag may be in, it will return to the Unpowered state if it moves out of the reader's field. (The exception to this rule is the Destroyed state, which indicates that the tag has been permanently disabled and may not be read or reused.) A variation on this protocol allows for tags to remember that they were in the Fixed Slot state for a short time even if power is lost. In this variation, a Complete Reset command allows the reader to force tags to the Ready state when necessary, even if the tag remembers having been in the Fixed Slot state previously. This scenario may occur in the case of a tag moving down a conveyor between readers. Having been read by the first reader, the tag believes its state should be Fixed Slot. However, the new reader has not seen this tag yet and will need it to go to the Ready state when powered, so the new reader issues a Complete Reset.

Figure 4-12. STAC protocol state diagram

The list below describes each STAC state associated with singulation, plus the Destroyed state. The Write state, which in current EPC Class 0 HF tags is issued only by the tag manufacturer to set the tag's EPC ID, is not shown. The states are:

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Unpowered

When outside the reader's field, a tag is in the Unpowered state. Upon entering a reader's field the tag enters the Ready state.

Ready

In this state, a tag waits for a Destroy, Write, or Begin Round command. If the tag receives a Begin Round command containing either no selection or a selection of equal length to the tag's EPC, the tag enters the Slotted Read state.

Slotted Read

In this state, the tag picks a slot at random from those presented by the reader. The slot may be any slot except the F slot. If the tag receives a Fix Slot command after sending its information, it proceeds to the Fixed Slot state. If a Fix Slot, Close Slot, or Begin Round command comes along with a matching selection, the tag will remain in the Slotted Read state If a Begin Round command arrives with a selection that does not match, the tag will return to the Ready state.

Fixed Slot

When a tag enters the Fixed Slot state, it responds on the F slot and will continue to do so for any subsequent responses until loss of power occurs (i.e., exiting the reader field).

Destroyed

If a tag receives a Destroy command and the password contained in the command matches the password contained in the tag, the tag will send its ID one more time and then permanently cease to function. Once destroyed, a tag may not be returned to service.

4.3.4. EPC UHF Class I Gen2

The latest revision of the EPC UHF Class I air interface is called the "Gen2 protocol." Gen2 addresses some of the limitations of the first UHF protocol standard by defining protocol variations that are able to work under European (CEPT) and North American (FCC) RF regulations. It also specifies variations in the protocol for operating environments where more than one reader is within another's receiving range, including a special set of protocol requirements for environments where the number of reader is equal to the number of available communication channels.

The EPC Gen2 protocol supports much faster tag singulation than the previous protocol, with tag reads rates as fast as 1,600 tags per minute in North America and 600 tags per minute under the more constrained power and frequency ranges in Europe. One of the primary concerns it addresses is added security for the protocol. Key to the solution Gen2 offers is the recognition that signals the reader transmits may be received over a far greater distance than signals generated by the tags. The

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

specification considers two readers to be in the same operating environment if they are within one kilometer of each other.

The protocol describes three procedures for communication between readers and tags. A reader may *select* tags by asking tags to compare themselves to a bitmask, or it may *inventory* tags by singulating tags until the reader has recognized each tag within range. A reader may also *access* tags, which includes reading information from a tag, writing information to a tag, killing a tag, or setting the lock status for various sections of tag memory by memory bank number.

4.3.4.1. Tag memory

The Gen2 protocol recognizes an optional user memory area and a Tag Identifier (TID) in addition to the CRC+EPC, which is called an Object Identifier (OID) in the specification. Each section of tag memory is organized into an addressable bank (see Table 4-5), and read and write commands in the protocol accept a bank address to determine which bank the operation affects.

The new specification also requires that, in addition to the kill code, each tag have an access password. Though the tag is required to support commands that use an access password or kill code, either or both may have a value of zero. Any reader command that selects a bank may operate only on that bank. To switch banks, the reader must issue a new command.

Bitmasks

The Gen2 specification describes several different features in terms of bitmasks. For those of us who haven't written low-level code in a while, a *bitmask* is a series of bits that can be compared to another series of bits using the binary operators "|" (OR), "&" (AND), "~" (NOT), and "^" (XOR). As an example, imagine that you need to find out if a value named `thisID` (which, for our example, will be equal to `7`) has a 1 in the twos position and in the fours position that is, the second and third least significant bits, or 0110, which equals 6. You could do this easily by evaluating the following expression:

```
6 & thisID = 6
```

If this expression evaluates to true (not zero), those three bits are 1s. Bitwise AND returns a 1 if the corresponding bits are 1 in both operands. So, for the example above, we would do the following test:

```
0110  = 6
0111  = 7
-----
0110  = 6
```

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Notice that this test would return true for 6 and for 15, but not for 8 or 14, for instance. So why would we care if `thisID` has 1s in those bits? For the Gen2 protocol, this issue comes up in two places. First, Gen2 uses bitmasks to compare values in tag memory against a Select command that works very much like the example above. Second, and more complex, is the way Gen2 uses bitmasks as a "cover code" to protect sensitive communications from the reader to the tag. On a Req_RN request from the reader, the tag generates a 16-bit random number and sends it to the reader. Tags transmit at low power, so only a nearby receiver can intercept this communication. The reader then uses this 16-bit number to XOR blocks of data sent back to the tag. For example, if the reader wanted to send a block of data equal to 0xAFAF and the random number from the tag was 0x7E21, the reader would evaluate the expression as follows:

```
0xAFAF ^ 0x7E21 = 0xD18E
```

or:

```
1010 1111 1010 1111 = 0xAFAF
0111 1110 0010 0001 = 0x7E21
--------------------
1101 0001 1000 1110 = 0xD18E
```

The reader would then send 0xD18E to the tag. Next, the tag would repeat and then XOR the same cover code against this value by evaluating the following expression:

```
0xD18E ^ 0x7E21 = 0xAFAF
```

The reader has thus sent the data as ciphertext over its relatively powerful signal, which helps thwart eavesdropping. An observer trying to determine the data sent to the tag would have to know the cover code to retrieve the data 0xAFAF, and to do that, he would have to be near enough to have received the cover code when it was transmitted (one time) via the weak backscatter from the tag.

Table 4-5. Tag memory banks

Bank	Contents
00	Reserved (access password and kill code, protocol bits, etc.)
01	OID (CRC+EPC)
10	TID (Tag manufacturer's ISO 15693 serial number)

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Bank	Contents
11	User

4.3.4.2. Inventory commands

When a reader starts an inventory over a group of tags, this is called a*session*. A tag will participate in only one session (i.e., talk to only one reader) at a time, but it may time-slice up to four sessions, which makes it appear as though the tag is talking to four readers at once. To support this, tags keep four inventoried flags: S0, S1, S2, and S3. Inventoried flags may have one of two values, A or B. While stepping through an inventory with a group of tags, a reader will set the inventory flags for the particular session that "belongs" to that reader until they all share the same value. Think of this as moving all of the singulated tags into a set named "A" or "B" for a particular session.

A reader working on session zero cannot see the values of the inventoried flags for the other three sessions, but if another reader were to alter some other value on the tagfor example, if it were to lock the OID bankthis would affect all sessions.

During the inventory, the reader uses the Slotted Random Anticollision method. This method, as its name implies, uses slots to determine when a tag should respond to the reader, but unlike with Slotted Aloha, the tag actually chooses its own slot by setting a slot counter to a 16-bit random number at the start of an inventory round and then decrementing until it reaches zero. When a tag's slot reaches zero, it sends a new 16-bit random number to the reader. The reader will then use this number to mask blocks when communicating with the tag. This effectively creates a one-time pad encryption for reader-to-tag communications. Tag-to-reader communications are still unencrypted but are transmitted at much lower power. The inventory commands are:

Query

The reader begins an inventory by issuing a Query command, which specifies the session and the number of slots. Tags generate random numbers and use them to determine which slot to use for responses. Tags that pick slot zero enter the Reply state, while tags that pick another slot remain in the Arbitrate state (tag states are discussed in the later section "Tag states").

Query Adjust

A Query Adjust changes the number of slots in an inventory round. The Query Adjust comman can add one to the number of slots, subtract one from the number of slots, or leave the number unchanged. Tags generate random numbers and select random slots from the new range. Tags that pick slot zero enter the Reply state, while tags that pick other slots remain in the Arbitrate state.

QueryRep

When the reader issues a QueryRep, tags decrement their slot counters. If the counter becomes zero, the tag enters the Reply state; otherwise, the tag remains in the Arbitrate state

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

ACK

A reader responds to a tag with an ACK, sending a 16-bit value back to the tag. If the tag is in the Reply or Acknowledged state, the reply is the 16-bit random number just sent by the tag as it transitioned from the Arbitrate state to that state. If the tag is in the Open or Secured state, the 16-bit ACK number is the tag's handle.

NAK

A reader can emit a NAK, which is 8 bits long with a value of 0xC0. When a tag receives a NAK it returns to the Arbitrate state unless it is in the Killed or Ready state, in which case the tag ignores the NAK.

4.3.4.3. The Select command

A reader may choose not to inventory all of the tags within range. A Select command asks tags to compare the contents of a particular bank of memory against a bitmask (see the "Bitmasks" sidebar). If the bitmask matches the tag's memory, the tag sets either its selected flag (SL) to true or its inventoried flag (one of S0-S3) to a value specified by the Select command. A single Select comman can set the value of either the select flag or a single inventoried flag (one per command).

4.3.4.4. Access commands

The access commands allow a reader to change the contents of a tag's memory, read from that memory, lock memory banks, kill a tag, or request a 16-bit random number generated by the tag. The reader must identify a single tag in order to use any of the access commands. Access commands that transmit sensitive data such as passwords, IDs, or user memory from a reader to a tag may be sent as ciphertext using a cover code, as discussed in the "Bitmasks" sidebar. The Gen2 standard requires both tags and readers to support the following access commands unless otherwise indicated

Req_RN

Requests a random (or, more likely pseudorandom) number from the tag. The "randomness" o this number is strictly defined in the Gen2 specification.

Read

Reads data from a particular bank of tag memory.

Write

Writes data to a particular bank of tag memory. Before each write, the reader issues a

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

Req_RN. The tag responds with a new 16-bit random number, which the reader then uses as a cover code to protect the 16 bits of data sent in the Write command. For longer data streams, the Req_RN-Write sequence is repeated as many times as necessary.

Kill

Permanently disables a tag. This command is similar to Write in that the reader first requests a 16-bit random number from the tag, which it uses as a cover code for the first 16 bits of the ki password. The sequence is then repeated for the second 16 bits. Once this command completes, the tag will never respond to another request. However, if a tag's kill password is zero, the tag can't be killed and will ignore this command.

Lock

Sets the read/write privileges for banks of memory or specific passwords. This may be a "permalock," in which case these privileges can never be changed again.

Access (optional)

If a tag has a nonzero password, the reader can use the Access command to put a tag that is in the Open state into the Secured state. If the tag is already Secured, it will remain Secured. Since this command includes the password, it is issued twice with a cover codethat is, the reader issues a Req_RN, gets the cover code, uses it for the first half of the password, and then repeats the sequence for the last half of the password.

BlockWrite (optional)

This command is similar to Write but is able to write multiple 16-bit blocks at once without a cover code. Thus, this command doesn't need to be repeated block by block, nor does it require a previous Req_RN.

BlockErase (optional)

This command allows a reader to erase multiple blocks from a single tag memory bank. The BlockErase command is protected with a CRC-16 and includes the tag's handle (i.e., its 16-bit tag authentication number).

4.3.4.5. Tag states

We can see the high-level structure of the EPC Gen2 protocol in Figure 4-13.

Notice that this is a slightly simpler state machine than that defined earlier for the Adaptive Binary Tree protocol.[*] The tag states are:

[*] For a full diagram of Gen2, see *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz 960 MHz Version 1.0.9*, p. 41, Fig. 6.19

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

(http://www.epcglobalinc.org/standards_technology/EPCglobalClass-1Generation-2UHFRFIDProtocolV109.pdf).

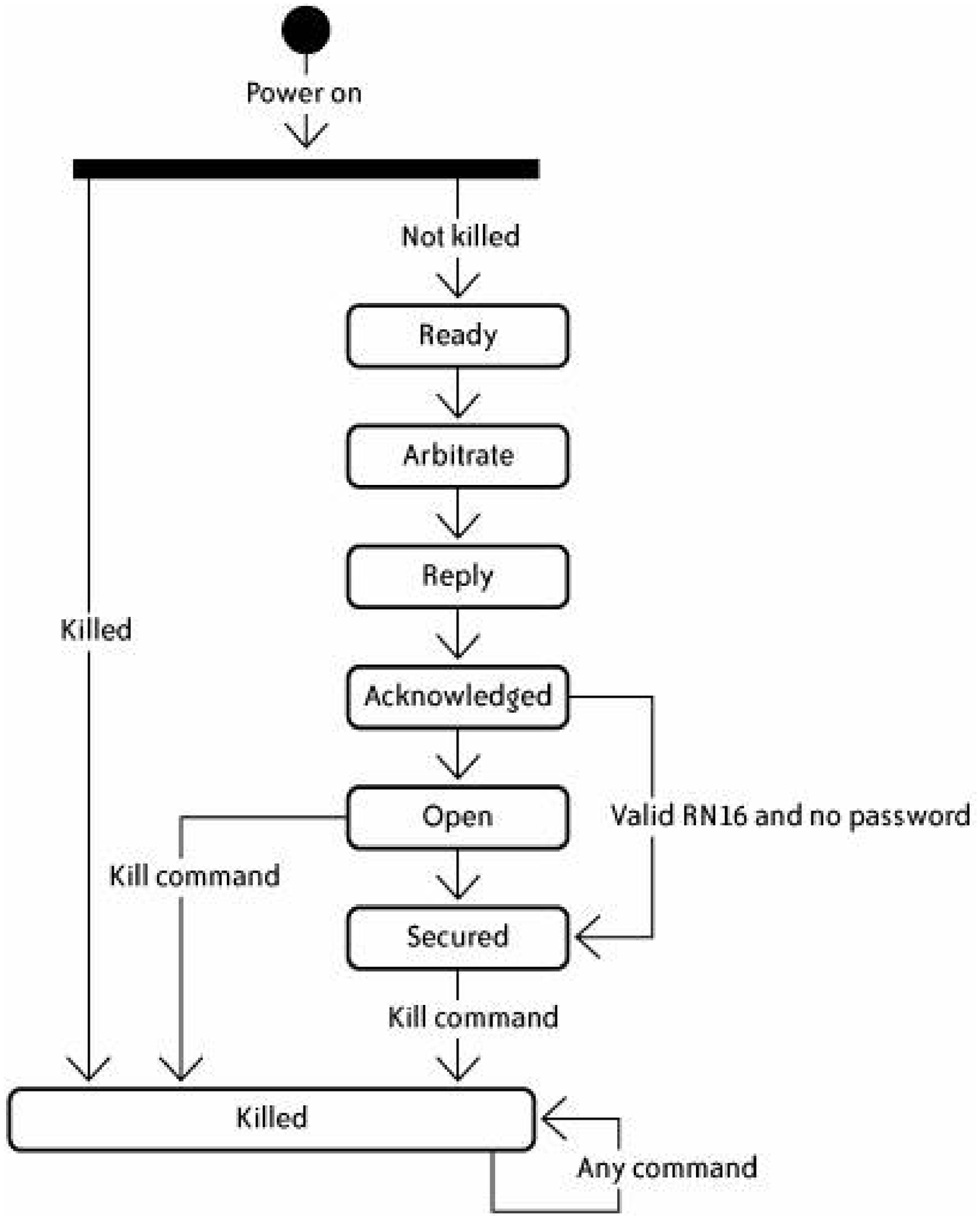
Ready

This is the state in which a tag waits when it is not part of the current inventory.

Arbitrate

This is the state in which a tag waits when it is part of an inventory, but not yet in slot zero.

Figure 4-13. Gen2 protocol state diagram



Reply

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

When a tag enters the Reply state, it generates a 16-bit random number and sends it to the reader. If it gets back an ACK, it enters the Acknowledged state; otherwise, it returns to the Arbitrate state.

Acknowledged

A tag may go to any state except Killed from this state.

Open

A tag with a nonzero password enters this state when it is in the Acknowledged state and receives a Req_RN from the reader. A tag in this state can go to any state except Acknowledged.

Secured

A tag with a password of zero enters this state when it receives a Req_RN while in the Acknowledged state, and a tag with a nonzero password enters this state from the Open state when it receives an Access command. A tag in this state can enter any state except Open or Acknowledged.

Killed

When a tag enters the Killed state, it sends a success response to the reader and is then permanently disabled; that tag will never respond to a request from a reader again.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.4. Tag Features for Security and Privacy

There are justifiable privacy and security concerns with any sort of identification technology, and RFID, as much as biometrics, has raised popular concern because of its possible impact on personal privacy. A common concern is that an unauthorized person will be able to obtain information from, or possibly even change information stored on, an RFID tag. There is some legitimacy to these concerns. A tag that does not use a secure protocol might be read by anyone, just as a bar code or printed label might be read by anyone. A writable tag might also be maliciously altered if it does not implement a secure protocol for communication with a reader. This alteration might be less obvious than someone placing a false bar code or printed tag on a piece of merchandise. Unlike printed tags and bar codes, which will always remain at risk, future secure protocols will make RFID tags very difficult to forge or even read without authorization. The more secure the tag, however, the more expensive it is, and an unencrypted, read-only tag is no more risky in the right application than a bar code is now. For these reasons, many supply chain applications currently use less-secure RFID implementations. As overall RFID prices drop due both to wider adoption and standardization, the cost of more secure will also drop, and these in turn will become more widely adopted.Chapter 10 covers security and privacy for the entire RFID system, but because some of the implementation details of RFID security are specifically concerned with the tag protocols, we'll take a closer look at tag destruction and data encryption here.

4.4.1. Destroying and Disabling Tags

A key concern of privacy advocates is the longevity of an identification mechanism such as RFID. Few privacy concerns arise from tracking pallets in a warehouse, or even from tracking individual items from the warehouse to the shelf. The point when an end user or customer first touches an item is the point where most privacy concerns arise. Within the store, expectations of privacy are tempered somewhat by the current precedent of surveillance from cameras and electronic anti-theft devices. But when RFID tags may still be attached and even active when a product leaves the store, there is a real possibility that tracking may continue outside the store, and this raises many concerns.

At the checkout counter, the respective interests of the seller and customer diverge. The seller, as represented by the manufacturer, the distributors, and the store itself, would very much like to know what happens to the item after it leaves the store. For example, sensors in the packaging that identify changes in temperature, acceleration, and shock could help the manufacturer design better packaging; readers in garbage collection systems could recover this information and report it back to the manufacturer. Supermarkets and distributors would like to know how long an item sat in the refrigerator or pantry before the customer used it. They would especially like to know what other products were in the trash in those households. It's the dream of market researchers everywhere to be able to definitively answer the question, "What else are my customers buying?"

The customer, in contrast, would rather not reveal too much information beyond the doors of the store. Discarded packaging for medicines might reveal sensitive information about a person's health. Simply adding up the cost of discarded goods and packages would provide a good estimate of the

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

household's real cash flow and pinpoint items worth stealing or households worth targeting in an easily manageable shopping list for thieves. Although much of this information is already available, it requires physically sorting through a household's weekly garbage. With a little imagination, the possible abuses for automated "dumpster diving" are numerous. Notice that this is still an issue even when the RFID tags are only on a product's packaging, which will likely be discarded. For the sake of privacy, many consumers would like to have any RFID tags disabled before leaving the store.

Privacy is not the only reason to disable a tag. Pharmaceutical manufacturers are interested in authenticating bottles as an anti-counterfeiting measure. Unless tags can be permanently and irreversibly altered or disabled in someway at the point of sale, these bottles could be refilled with counterfeit products and sold again.

So how is an RFID tag disabled? The implementation of the kill command is left to the manufacturer, but many use embedded fuses on a chip that physically disable it. Others set a "killed" value in memory, which disables the protocol state machine. In either case, the death of the chip is triggered by a Destroy command containing the appropriate password, which is set by the manufacturer. Even in the case of chips that are otherwise writable, EPC standards require that this password is unable to be changed or read by any command, including diagnostic commands intended for use only by the tag's manufacturer.

This raises its own set of security questions about how to distribute these kill codes. The password must somehow arrive at the point of sale, yet it cannot be public knowledge since this might allow a prankster to disable tags maliciously. The problem is clearer if we think of the lifecycle of a tag in a supply chain.

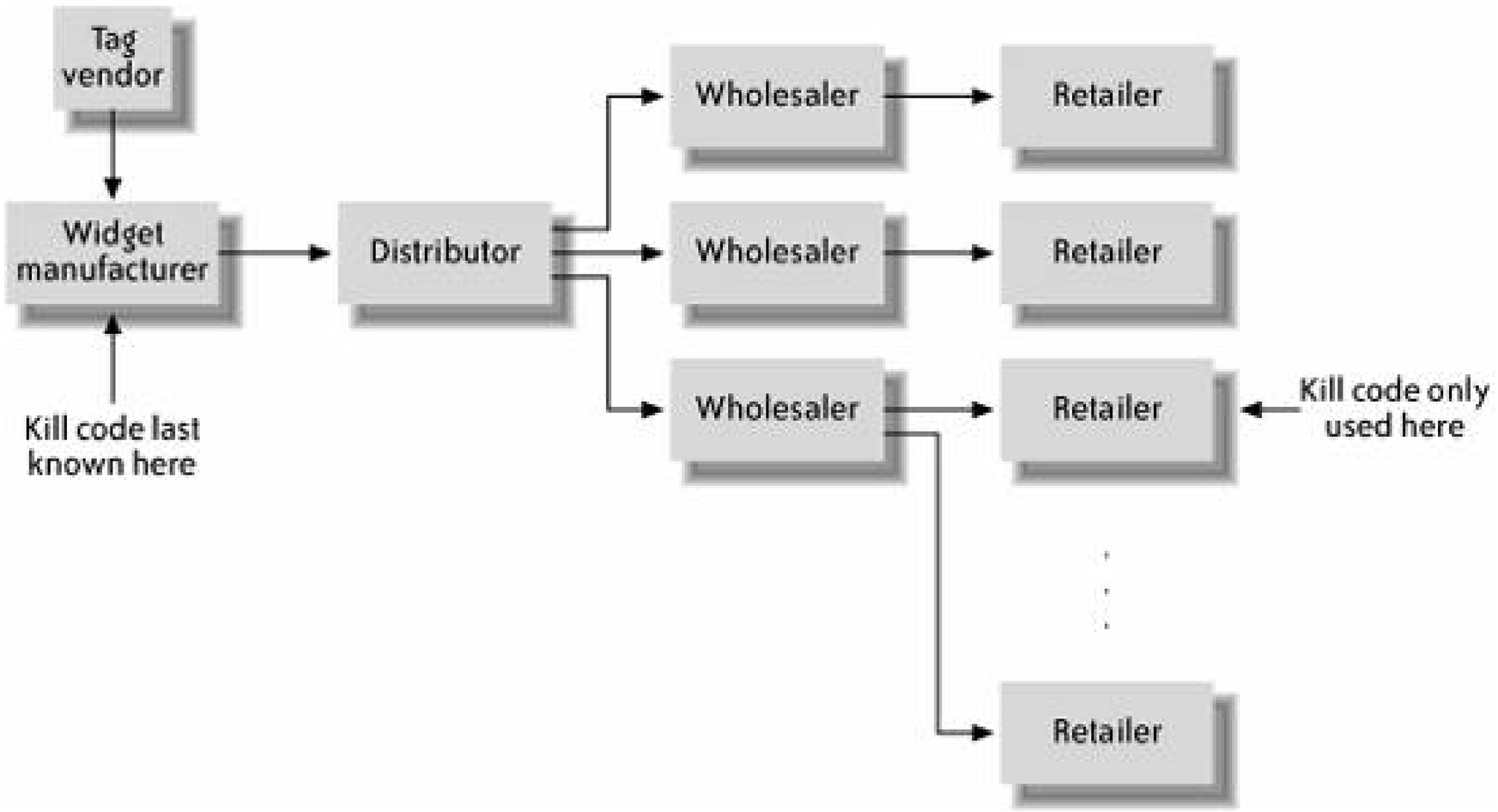
At the beginning of its journey to the store, each tag leaves its manufacturer in a box bound for a widget factory. For the sake of simplicity, we will assume that the tags are smart labels and that each tag on this particular roll will accept the same kill code (say, 23532). The kill code may be sent to the widget factory on a slip of paper in the box. The widget factory produces the widgets and places a tag on each of them. These widgets are placed on pallets and picked up by a distributor, who sends them to one of three wholesalers. Fifty different retailers buy these widgets, along with others from earlier and later runs. Some of these retailers are small and don't track IDs at all. Others disable tags at the checkout, while still others allow customers to choose whether the tags should be disabled or not. Which kill codes should be sent to which retailers, and how should it be done? Figure 4-14 illustrates the problem.

Creating a "push" solution in which the manufacturer sends the kill codes to each retailer won't work because the widget manufacturer has no easy way of knowing which retailers have which batches of their products at any given moment. A "pull" makes much more sense. This is where the Object Naming Service comes in. We'll discuss the details of how this works in Chapter 8, but for now, it's enough to know that the ONS allows a retailer to look up information concerning tags on its products. This information can include kill codes, which might be stored on a secure web server or on an EPC Information Service (EPCIS) server. Requests for kill codes can be authenticated with unique store identifiers and tracked by the manufacturer to discourage unauthorized access.

Figure 4-14. Life of a tag

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir



This example points out how even seemingly simple cases of use, such as theft prevention using EPC tags, may require a relatively sophisticated information infrastructure. EPCglobal's choice of a password-based method for disabling tags shows that the standards bodies expect that most users o EPC tags will be doing more than simple Electronic Article Surveillance. The build-out for this infrastructure is only now beginning, though, so in the meantime, a retailer or end user might choose to ignore the EPC kill codes altogether and simply destroy tags with a hammer or a sharp knife.

4.4.2. Encrypting Information

Before the Gen2 specification, any EPC Class I reader was able to read any EPC Class I tag. As we described in the earlier section "EPC UHF Class I Gen2," tags using the Gen2 protocol generate 16-bit random numbers, which readers then use to mask similar-sized blocks of data when communicating with the tags. This effectively creates a one-time pad encryption for reader-to-tag communications. Tag-to-reader communications are still unencrypted but are transmitted at much lower power and so are less vulnerable to snooping. This technique makes eavesdropping on RFID communications more difficult. The capability to secure a tag with a password helps to further thwart attempts to read or modify tags using unauthorized readers.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.5. Learn to Troubleshoot Tag Communications

Troubleshooting tag and reader communications is the sort of subject that could (and probably someday will) fill up an entire book, but a few general techniques can solve the most common problems. Not many of us have spectrum analyzers, but RF questions can usually be approached indirectly. First, think of ways to halve the problem. In the usual case, where one reader fails to talk to one tag, we can halve the problem by asking, "Is it the reader or the tag?" Try another tag with the reader. If the new tag works, you have a faulty tag. If the new tag doesn't work, you have a faulty reader (or two faulty tags; in this case, it's best to try a few more tags just to be sure). The reader uses one or more antennas to talk to the tags. Can you swap antennas with another reader that works?

Warning: Be careful! Only swap antennas from identical readers and consult the reader manual first for more precautions.

Chances are that instead of being broken, the reader is simply misconfigured. Most readers these days support multiple tag types. Is the reader configured to speak to an ISO 18000-6 Type A tag only? If so, an EPC tag won't work (this week). Is it configured (or built) for HF or UHF? What about the tag? Was it tuned to work on a metal can or a rubber tire? Try it on the appropriate item.

A simple trick for testing packaging is to place a tagone that you know to workin different locations on the packaging of an item already in the field of a reader that provides a visible read light. Move the tag around and watch the read light come on and go off. In this way, one can develop a very good idea of the RF profile of an item and its packaging in just a few minutes. To "hide" a tag, place your body between the tag and the reader. Squishy humans are great RF shields.

In a story (possibly apocryphal) often told around the RFID camp, two technicians testing a reader and tag combination found that they could read the tag at a great distance in some cases but only at a short distance in others. To demonstrate, one technician would hold a portable reader near an item bearing the tag and then slowly step back. At roughly a half a meter, the tag would cease to register This seemed to be the "dead zone." Another technician would then hold the end of a tape measure and the technician with the reader would step back to nearly 10 meters away. All the way from the outside of the dead zone to this limit, the reading was strong. An observer then pointed out that the metal tape measure might make an excellent antenna...

The lesson here is to look for unexpected variables. For example, if you get slow reads while using a filter, it might mean that many tags are responding but most don't fit the filter criteria. This could look as if a single tag is reading slowly, since the time spent turning off the excluded tags is not reflected in the single tag identity that's returned. Try reading a few times with no filters set.

downloaded from: lib.ommolketab.ir

downloaded from: lib.ommolketab.ir

4.6. Summary

Understanding tag protocols, including singulation and encodings for identities, prepares the way for our discussion of readers in the next two chapters, as well as for understanding the reader protocols and RFID middleware discussed in Chapters 6 and 7. While these tag protocols have been the area most affected by what David P. Meany of Cisco Systems called the "tornado of obsolescence" that surrounds RFID technology,[*] the much-anticipated Gen2 specifications from EPC have removed some of the uncertainty. Anyone working with RFID should keep up with at least the high points of the ongoing debate over adoption of this protocol, as well as some of the intellectual property issues currently in the news. The organizations and publications listed in Appendix B are a good place to start.

[*] David P. Meany, speaking at the 2004 ECPglobalUS conference in Baltimore, MD.

In this chapter we learned:

- That a protocol is a set of formal rules describing how to transmit data
- The meaning of the terms *singulation*, *anti-collision*, and *identity*
- How to encode identities for storage on an EPC RFID tag
- About singulation and anti-collision protocols, including Slotted Aloha, Adaptive Binary Tree, STAC, and the EPC Gen2 protocol
- About some of the features available in the tag protocols to increase security and privacy through encryption and careful planning
- A few tips on troubleshooting reader/tag communications

downloaded from: lib.ommolketab.ir