XMPP is a robust protocol used for a wide range of applications, including instant messaging, multi-user chat, voice and video conferencing, collaborative spaces, real-time gaming, data synchronization, and search. This book teaches you how to harness the power of XMPP in your own apps and presents you with all the tools you need to build the next generation of apps using XMPP or add new features to your current apps. Featuring the JavaScript language throughout and making use of the jQuery library, the book contains several XMPP apps of increasing complexity that serve as ideal learning tools.

- Learn about XMPP's instant messaging features, such as rosters, presence and subscriptions, and private chats
- Covers XMPP stanzas, stanza errors, and client protocol syntax and semantics
- Discusses service discovery, data forms, and publish-subscribe
- Addresses XMPP programming-related topics such as application design, event handling, and combining existing protocols
- Details the best ways to deploy XMPP-based applications
- Explains how to use Strophe's plugin system and how to create your own plugins

Jack Moffitt is CTO of Collecta, where he leads a team that produces several XMPP-related projects including Strophe (a family of libraries for XMPP communication), Palaver (a group chat server), Punjab (an HTTP to XMPP gateway service), and Speeqe (a simple web-based group chat client). He also served several terms on the Board of Direction and Council of the XMPP Standards Foundation.

Wrox Professional guides are planned and written by working programmers to meet the real-world needs of programmers, developers, and IT professionals. Focused and relevant, they address the issues technology professionals face every day. They provide examples, practical solutions, and expert education in new technologies, all designed to help programmers do a better job.
Related Wrox Books

Beginning JavaScript, 4th Edition
ISBN: 978-0-470-52593-7
Serving as a great introduction to JavaScript, this book offers all you need to start using JavaScript on your web pages right away. It's fully updated and covers utilizing JavaScript with the latest versions of the Internet Explorer, Firefox, and Safari browsers and walks you through the basics of JavaScript: what it is, how it works, and what you can do with it.

Beginning JavaScript and CSS Development with jQuery
ISBN: 978-0-470-22779-4
With this unique, project-oriented book, author Richard York teaches even the most novice of JavaScript users how to quickly get started utilizing the JavaScript jQuery Library to decrease the amount of code that needs to be written and tested. A four-color code syntax highlighting system provides a visual reinforcement and allows you to see the nuts and bolts that compose each line and section of code. With this helpful guide and hands-on exercises, you'll be able to put jQuery to work for you and avoid having to write code from scratch.

Beginning XML, 4th Edition
The perfect resource for beginning XML programmers, this guidebook clearly shows you what XML is, how to use it, and what technologies surround it. The authors build on the strengths of previous editions while covering the latest changes in the XML landscape such as XQuery, RSS and Atom, and Ajax. The most recent specifications are presented along with best practices to follow when developing XML solutions. You'll delve into the state of the art for XML and databases, discover how to query XML information, retrieve data, and create new XML documents. In addition, you'll learn how to publish information on the web, design dynamic interactive graphics, and make interactive forms.

Professional JavaScript for Web Developers, 2nd Edition
ISBN: 978-0-470-22780-0
This book covers JavaScript from its very beginning to the present-day incarnations that include support for the DOM and Ajax. It also shows you how to extend this powerful language to meet specific needs and create seamless client-server communication without intermediaries such as Java or hidden frames. You'll explore the components that make up a JavaScript implementation, with specific focus on standards such as ECMAScript and DOM. You'll also learn how to utilize regular expressions and build dynamic user interfaces. This valuable insight will help you apply JavaScript solutions to the business problems faced by Web developers everywhere.

Professional XML
From document type definitions and XQuery to Web services and form development, this book takes you through all of the XML capabilities that companies want to implement today. It not only covers the basics of XML and the XML specification, but also shows you the XML-based applications that are driving the tech industry forward, such as blogging and alerts.
PROFESSIONAL
XMPP PROGRAMMING
WITH JAVASCRIPT AND JQUERY

INTRODUCTION .......................................................... xix

PART I XMPP PROTOCOL AND ARCHITECTURE
CHAPTER 1 Getting to Know XMPP .................................. 3
CHAPTER 2 Designing XMPP Applications ..................... 23

PART II THE APPLICATIONS
CHAPTER 3 Saying Hello: The First Application ............... 39
CHAPTER 4 Exploring the XMPP Protocol: A Debugging Console .... 63
CHAPTER 5 Microblogging in Real Time: An Identica Client .... 87
CHAPTER 6 Talking with Friends: One-on-One Chat ......... 103
CHAPTER 7 Exploring Services: Service Discovery and Browsing .... 145
CHAPTER 8 Group Chatting: A Multi-User Chat Client ....... 165
CHAPTER 9 Publishing and Subscribing: A Shared Sketch Pad Introduction .... 203
CHAPTER 10 Writing with Friends: A Collaborative Text Editor .... 251
CHAPTER 11 Playing Games: Head to Head Tic-Tac-Toe ....... 299

PART III ADVANCED TOPICS
CHAPTER 12 Getting Attached: Bootstrapping BOSH ........... 377
CHAPTER 13 Deploying XMPP Applications .................... 387
CHAPTER 14 Writing Strophe Plug-ins .......................... 401

APPENDIX A Getting Started with jQuery ...................... 419
APPENDIX B Setting Up a BOSH Connection Manager ....... 429

INDEX ................................................................. 441
Dedicated to my wife Kimberly and our son Jasper, whose loves, hugs, and smiles make every day the best day ever.
EXECUTIVE EDITOR
Carol Long

PROJECT EDITOR
Ed Connor

TECHNICAL EDITOR
Dave Cridland

PRODUCTION EDITOR
Kathleen Wisor

COPY EDITOR
Kim Cofer

EDITORIAL DIRECTOR
Robyn B. Siesky

EDITORIAL MANAGER
Mary Beth Wakefield

MARKETING MANAGER
David Mayhew

PRODUCTION MANAGER
Tim Tate

VICE PRESIDENT AND
EXECUTIVE GROUP PUBLISHER
Richard Swadley

VICE PRESIDENT AND EXECUTIVE PUBLISHER
Barry Pruett

ASSOCIATE PUBLISHER
Jim Minatel

PROJECT COORDINATOR, COVER
Lynsey Stanford

COMPOSITOR
Craig Johnson, Happenstance Type-O-Rama

PROOFREADER
Carrie Hunter, Word One

INDEXER
Robert Swanson

COVER DESIGNER
Michael E. Trent

COVER IMAGE
© Punchstock/Glowimages
ABOUT THE AUTHOR

JACK MOFFITT is a hacker and entrepreneur based in Albuquerque, New Mexico. He has founded several startups built on XMPP technology including Chesspark, a real-time, multi-user gaming platform, and Collecta, a real-time search engine for the Web. He has started and contributed to numerous XMPP related open source and free software projects including the Strophe XMPP client libraries, the Punjab XMPP connection manager, the Palaver multi-user chat component, the Speeqe group chat application. He also has served several terms on both the XSF Board of Directors and the XSF Council. Previous to his XMPP work, he created the Icecast streaming media server, managed the Ogg, Vorbis, and Theora codec projects, and co-founded the Xiph.org Foundation, a standards organization for royalty-free multimedia technologies for the Internet. He is passionate about free software and open source, open standards, and Internet technology. His favorite programming languages include JavaScript, Erlang, and Python. You can find him at http://metajack.im, blogging about start-ups and code, as @metajack on Twitter and Identica, or often spreading the word of XMPP at technology conferences.
ACKNOWLEDGMENTS

WE ALL STAND ON THE SHOULDERS OF GIANTS, and I am fortunate to have stood on many friendly ones throughout my career and while writing this book. Thanks to Carol Long and Ed Connor for the encouragement, handholding, reminders, and patience that every author needs. Thanks also to Jason Salas who not only encouraged me on this project but made the appropriate introductions. Thanks also to Dave Cridland for his work ensuring the technical quality of this book and his tireless humor. I’m hugely indebted to Peter Saint-Andre, patron saint of XMPP, and the rest of the XMPP Standards Foundation members for their advice, criticism, and friendship over the years. Finally, the biggest thanks of all to my wife; not only did she encourage me in this project and put up with my long hours and absence, she also worked hard as my first reader and made many helpful suggestions to the text.
| CONTENTS |
|-----------------|--------|
| Making XMPP Applications | 33 |
| The Browser Platform | 33 |
| Basic Infrastructure | 33 |
| Protocol Design | 34 |
| Summary | 36 |

### PART II: THE APPLICATIONS

#### CHAPTER 3: SAYING HELLO: THE FIRST APPLICATION | 39

- Application Preview | 40
- Hello Design | 40
- Preparing the Way | 41
  - jQuery and jQuery UI | 42
  - Strophe | 42
  - flXHR | 43
  - XMPP Accounts | 43
- Starting Your First Application | 43
  - User Interface | 43
  - Application Code | 45
- Making Connections | 47
  - The Connection Life Cycle | 47
  - Creating a Connection | 48
  - Connecting Hello | 49
  - Running the Application | 53
- Creating Stanzas | 53
  - Strophe Builders | 54
  - Saying Hello | 56
- Handling Events | 57
  - Adding and Removing Handlers | 57
  - Stanza Matching | 57
  - Stanza Handler Functions | 58
  - Handling Responses in Hello | 59
- More Hellos | 62
- Summary | 62

#### CHAPTER 4: EXPLORING THE XMPP PROTOCOL: A DEBUGGING CONSOLE | 63

- Application Preview | 64
- Peek Design | 64
### CONTENTS

<table>
<thead>
<tr>
<th>Building the Console</th>
<th>65</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>66</td>
</tr>
<tr>
<td>Displaying Traffic</td>
<td>68</td>
</tr>
<tr>
<td>Making XML Pretty</td>
<td>71</td>
</tr>
<tr>
<td>Dealing with XML Input</td>
<td>74</td>
</tr>
<tr>
<td>Making Input Easier</td>
<td>76</td>
</tr>
<tr>
<td><strong>Exploring XMPP</strong></td>
<td>81</td>
</tr>
<tr>
<td>Controlling Presence</td>
<td>81</td>
</tr>
<tr>
<td>Probing Versions</td>
<td>81</td>
</tr>
<tr>
<td>Dealing with Errors</td>
<td>82</td>
</tr>
<tr>
<td><strong>Better Debugging</strong></td>
<td>85</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>85</td>
</tr>
</tbody>
</table>

**CHAPTER 5: MICROBLOGGING IN REAL TIME: AN IDENTICA CLIENT**

<table>
<thead>
<tr>
<th>Application Preview</th>
<th>88</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arthur Design</strong></td>
<td>88</td>
</tr>
<tr>
<td>Microblogging with Identica</td>
<td>89</td>
</tr>
<tr>
<td>Creating Your Account</td>
<td>90</td>
</tr>
<tr>
<td>Turning on XMPP</td>
<td>90</td>
</tr>
<tr>
<td><strong>Building Arthur</strong></td>
<td>90</td>
</tr>
<tr>
<td>Getting Started</td>
<td>91</td>
</tr>
<tr>
<td>Receiving Messages</td>
<td>93</td>
</tr>
<tr>
<td>XHTML-IM</td>
<td>94</td>
</tr>
<tr>
<td>Adding XHTML-IM to Arthur</td>
<td>95</td>
</tr>
<tr>
<td>Sending Messages</td>
<td>96</td>
</tr>
<tr>
<td><strong>Offline Messages</strong></td>
<td>97</td>
</tr>
<tr>
<td>Creating a Better Microblogger</td>
<td>101</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>102</td>
</tr>
</tbody>
</table>

**CHAPTER 6: TALKING WITH FRIENDS: ONE-ON-ONE CHAT**

<table>
<thead>
<tr>
<th>Application Preview</th>
<th>104</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gab Design</strong></td>
<td>104</td>
</tr>
<tr>
<td>Presence</td>
<td>105</td>
</tr>
<tr>
<td>Messages</td>
<td>105</td>
</tr>
<tr>
<td>Chat Area</td>
<td>105</td>
</tr>
<tr>
<td>Roster Area</td>
<td>106</td>
</tr>
<tr>
<td><strong>Making the Interface</strong></td>
<td>106</td>
</tr>
</tbody>
</table>
## CONTENTS

### Building the Roster 109
- Requesting Rosters 111
- Handling IQs 113
- Updating Presence Status 114
- Adding New Contacts 116
- Responding to Roster Changes 117
- Dealing with Subscription Requests 119

### Building the Chats 122
- Working with Tabs 122
- Creating New Chats 123
- Sending Messages 124

### Best Practices for Instant Messaging 127
- Understanding Message Routing 127
- Addressing Messages Better 127

### Adding Activity Notifications 130
- Understanding Chat States 130
- Sending Notifications 131
- Receiving Notifications 132

### Final Touches 133
- Gabbing More 143
- Summary 143

### CHAPTER 7: EXPLORING SERVICES: SERVICE DISCOVERY AND BROWSING 145

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Preview</td>
<td>146</td>
</tr>
<tr>
<td>Dig Design</td>
<td>146</td>
</tr>
<tr>
<td>Finding Information</td>
<td>147</td>
</tr>
<tr>
<td>Disco#info Queries</td>
<td>147</td>
</tr>
<tr>
<td>Disco#items Queries</td>
<td>148</td>
</tr>
<tr>
<td>Disco Nodes</td>
<td>149</td>
</tr>
<tr>
<td>Creating Dig</td>
<td>149</td>
</tr>
<tr>
<td>Initial Disco Queries</td>
<td>153</td>
</tr>
<tr>
<td>Browsing the Disco Tree</td>
<td>155</td>
</tr>
<tr>
<td>Digging into Services</td>
<td>160</td>
</tr>
<tr>
<td>Finding a Proxy Server</td>
<td>161</td>
</tr>
<tr>
<td>Discovering Features</td>
<td>162</td>
</tr>
<tr>
<td>Looking for a Chat</td>
<td>162</td>
</tr>
<tr>
<td>Discovering More</td>
<td>163</td>
</tr>
<tr>
<td>Summary</td>
<td>163</td>
</tr>
</tbody>
</table>
## CHAPTER 8: GROUP CHATTING: A MULTI-USER CHAT CLIENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Preview</td>
<td>166</td>
</tr>
<tr>
<td>Groupie Design</td>
<td>167</td>
</tr>
<tr>
<td>Public Speaking</td>
<td>167</td>
</tr>
<tr>
<td>Group Chat Services</td>
<td>167</td>
</tr>
<tr>
<td>Entering and Leaving a Room</td>
<td>168</td>
</tr>
<tr>
<td>Sending and Receiving Messages</td>
<td>171</td>
</tr>
<tr>
<td>Anonymity</td>
<td>171</td>
</tr>
<tr>
<td>Creating Rooms</td>
<td>172</td>
</tr>
<tr>
<td>Understanding Roles and Affiliations</td>
<td>173</td>
</tr>
<tr>
<td><strong>Building the Interface</strong></td>
<td>175</td>
</tr>
<tr>
<td><strong>Joining the Room</strong></td>
<td>179</td>
</tr>
<tr>
<td><strong>Dealing with Presence and Messages</strong></td>
<td>183</td>
</tr>
<tr>
<td>Handling Room Messages</td>
<td>183</td>
</tr>
<tr>
<td>Tracking Presence Changes</td>
<td>185</td>
</tr>
<tr>
<td>Chat History</td>
<td>186</td>
</tr>
<tr>
<td>Keeping It Private</td>
<td>187</td>
</tr>
<tr>
<td>Describing Actions</td>
<td>190</td>
</tr>
<tr>
<td><strong>Managing the Room</strong></td>
<td>191</td>
</tr>
<tr>
<td>Changing Topics</td>
<td>191</td>
</tr>
<tr>
<td>Dealing with Troublemakers</td>
<td>192</td>
</tr>
<tr>
<td>Recruiting Help</td>
<td>194</td>
</tr>
<tr>
<td><strong>Improving Groupie</strong></td>
<td>201</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>201</td>
</tr>
</tbody>
</table>

## CHAPTER 9: PUBLISHING AND SUBSCRIBING: A SHARED SKETCH PAD INTRODUCTION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SketchCast Preview</td>
<td>204</td>
</tr>
<tr>
<td>SketchCast Design</td>
<td>205</td>
</tr>
<tr>
<td>Everything Is Pubsub</td>
<td>205</td>
</tr>
<tr>
<td>Presenter’s Flow</td>
<td>205</td>
</tr>
<tr>
<td>Audience’s Flow</td>
<td>206</td>
</tr>
<tr>
<td><strong>Filling Out Forms</strong></td>
<td>206</td>
</tr>
<tr>
<td>What Is The Data Forms Extension?</td>
<td>206</td>
</tr>
<tr>
<td>Form Elements, Fields, and Types</td>
<td>207</td>
</tr>
<tr>
<td>Standardized Form Fields</td>
<td>210</td>
</tr>
</tbody>
</table>
## CONTENTS

### Working with Pubsub Nodes
- Creating Nodes 211
- Configuring Nodes 213
- Pubsub Events 215
- Publishing to a Node 215
- Subscribing and Unsubscribing 216
- Retrieving Subscriptions 218
- Retrieving Items 219
- Subscription Management 221

### Broadcasting Sketches Using Pubsub
- Building the Interface 222
- Sketching with Canvas 225
- Logging In and Making Nodes 228
- Publishing and Receiving Sketch Events 234

### Summary 249

### CHAPTER 10: WRITING WITH FRIENDS:
#### A COLLABORATIVE TEXT EDITOR 251

- Application Preview 252
- NetPad Design 252
- Operational Transformation 253
  - Basic Principles 253
  - Details of the Algorithm 255
  - Implementation 256
- Extending the XMPP Protocol 267
  - Ignoring the Unknown 267
  - XML Namespaces 268
  - Extended Elements 268
  - Extended Attributes 270
  - Contributing Extensions 271
- Designing the Protocol 271
  - Testing for Support 272
  - Requesting and Controlling Sessions 272
  - Editing Operations 273
- Building the Editor 274
  - The Initial Skeleton 274
  - Starting Editing Sessions 278
  - Chatting About Work 284
  - Making Edits 287
- Expanding NetPad 298
- Summary 298
CONTENTS

Growing Vertically 394
  Reducing Latency 394
  Minimizing XML Serialization 397
  Optimizing DOM Operations 398
Summary 399

CHAPTER 14: WRITING STROPHE PLUG-INS 401

Using Plug-ins 402
  Loading Plug-ins 402
  Accessing Plug-in Functionality 402
Building Plug-ins 403
Creating a Roster Plug-in 404
  Storing Contacts 405
  Getting and Maintaining the Roster 407
  Manipulating the Roster 411
Taking the Plug-in for a Spin 412
  Improving the Roster Plug-in 417
Summary 417

Appendix A: Getting Started with jQuery 419
Appendix B: Setting Up a BOSH Connection Manager 429

INDEX 441
INTRODUCTION

XMPP POWERS A WIDE RANGE OF APPLICATIONS including instant messaging, multi-user chat, voice and video conferencing, collaborative spaces, real-time gaming, data synchronization, and even search. Although XMPP started its life as an open, standardized alternative to proprietary instant messaging systems like ICQ and AOL Instant Messenger, it has matured into an extremely robust protocol for all kinds of exciting creations.

Facebook uses XMPP technology as part of its chat system. Google uses XMPP to power Google Talk and its exciting new Google Wave protocol. Collecta has built a real-time search engine based extensively on XMPP’s publish-subscribe system. Several web browsers are experimenting with XMPP as the basis of their synchronization and sharing systems. Dozens of other companies have XMPP-enabled their web applications to provide enhanced user experiences and real-time interaction.

The core of XMPP is the exchange of small, structured chunks of information. Like HTTP, XMPP is a client-server protocol, but it differs from HTTP by allowing either side to send data to the other asynchronously. XMPP connections are long lived, and data is pushed instead of pulled.

Because of XMPP’s differences, it provides an excellent companion protocol to HTTP. XMPP-powered web applications are to AJAX what AJAX was to the static web site; they are the next level of interactivity and dynamism. Where JavaScript and dynamic HTML have brought desktop application features to the web browser, XMPP brings new communications possibilities to the Web.

XMPP has many common social web features built in, due to its instant messaging heritage. Contact lists and subscriptions create social graphs, presence updates help users keep track of who is doing what, and private messaging makes communication among users trivial. XMPP also has nearly 300 extensions, providing a broad and useful range of tools on which to build sophisticated applications. With only a handful of these, along with the core protocol, amazing things can be built.

This book teaches you to harness the promise of XMPP in your own applications, enabling you to build applications that are social, collaborative, real time, or all of the above. You will develop a series of increasingly sophisticated XMPP applications, starting from “Hello, World!” and finishing with a collaborative text editor, a shared sketch pad, and a real-time, multi-player game. By the end, you will have all the tools you need to build the next generation of applications using XMPP or to add new real-time, push, or social features to your current applications.

WHO THIS BOOK IS FOR

This book is written for developers interested in making XMPP applications. You need not have any previous experience with XMPP, although it will certainly be helpful if you do. The book starts from the assumption that you’ve heard great things about XMPP and are looking to dive right in.
The JavaScript language is used to develop all the applications in the book because it is an easy language to understand, is familiar to a large number of programmers, and comes on every computer with a web browser. Even though this book uses JavaScript, all the concepts and applications could be developed in any language; most of the “hard parts” are not related to the programming language, the libraries used, or the web browser. You do not need to be a JavaScript expert to understand and work with the code in this book.

It is assumed that you understand the basic front-end web technologies, CSS and HTML. If you’ve ever written a little HTML from scratch and changed a few CSS styling properties, you should be fine.

This book also makes use of two libraries, jQuery and Strophe. It is helpful if you have used jQuery before, but if you haven’t, a short primer is included in Appendix A. The Strophe library is explained fully as the applications are developed.

WHAT THIS BOOK COVERS

The XMPP protocol and its extensions cover a lot of ground. This book focuses on the pieces of XMPP in wide use. The following topics receive much attention:

- XMPP’s instant messaging features like rosters, presence and subscriptions, and private chats
- XMPP stanzas, stanza errors, and client protocol syntax and semantics
- Extending XMPP stanzas
- Service discovery (XEP-0030)
- Data Forms (XEP-0004)
- Multi-User Chat (XEP-0045)
- Publish-Subscribe (XEP-0060)

Although these topics are all approached from the client side, almost all of it is equally applicable to XMPP bots or server components and plug-ins.

The book also covers XMPP programming related topics such as application design, event handling, and combining simple protocol elements into a greater whole. Along the way, a few web programming topics are also discussed such as the Canvas API.

XMPP is now more than 10 years old and quite mature. This book covers the 1.0 version of the core protocol. The XMPP protocol parts of this book should work unchanged in future versions of the protocol, just as HTTP 1.0 clients can easily communicate with HTTP 1.1 servers.

XMPP has many extensions and several of these are also covered. For the most part, the book concentrates on extensions that are in a stable, mature state. For each extension used, the document number is always given, and if in doubt, you can always check the latest version of the extension to see if it has been changed or superseded.
The book was written with the 1.3 series versions of jQuery and the 1.7 series versions of jQuery UI. These libraries generally remain backward compatible to a large degree. Version 1.0 of the Strophe library is used, but future 1.X versions should also work fine.

HOW THIS BOOK IS STRUCTURED

This book is primarily organized as a walkthrough tutorial of a series of example XMPP applications. Each application increases in difficulty and teaches you one or more useful parts of the XMPP protocol and its extensions. These applications are stripped down for clarity, but they are examples of the kinds of applications XMPP developers create every day.

This book is divided into three parts.

The first part is an introduction to the XMPP protocol, its uses, and XMPP application design. Chapter 1 covers the use cases for XMPP, the history of the protocol, and its component parts. Chapter 2 explains when XMPP is a good choice for the job and goes into detail about how XMPP applications work, particularly for the Web.

The second part is the meat of the book and contains nine XMPP applications that solve a variety of problems. Each application is more complex than the last and builds on the concepts of the previous ones. Chapter 3 starts with a simple “Hello, World!” type example, and by Chapter 11 you build a real-time, multi-player game.

The last part covers a few advanced but important topics. Chapter 12 discusses attached sessions, a useful trick for security, optimization, and persistence. Chapter 13 goes into detail about how best to deploy and scale XMPP-based applications. Chapter 14 explains how to use Strophe’s plug-in system and how to create your own plug-ins.

WHAT YOU NEED TO USE THIS BOOK

This book makes use of web technologies and therefore requires almost no special tools. You can use, build, and run the applications in this book on virtually any platform. The libraries needed for the applications are explained in Chapter 3, and most can be used without downloading any code.

You will need some way to serve web pages such as a local web server or a hosting account somewhere. If you don’t have these readily available, you can use the Tape program to serve the files; Tape is a simple web server and is explained in Appendix B. It is an unfortunate requirement of browser security policy that you can’t easily run these applications directly from your local file system.

You will need an XMPP account (or multiple accounts in some cases if you want to test the code by yourself) to run the applications. You can avail yourself of any of the public XMPP servers for this purpose, although you will need to ensure that the server has support for publish-subscribe and multi-user chat; most do. You can also download and run your own XMPP server instead, although this is not covered in the book.
Chapter 12 requires some server-side assistance. The example uses the Python programming language along with the Django framework to provide this. This chapter is an advanced topic and is not needed for the normal applications in the book.

**CONVENTIONS**

To help you get the most from the text and keep track of what’s happening, we’ve used a number of conventions throughout the book.

Boxes like this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.

Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.

As for styles in the text:

- We highlight new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so: persistence.properties.
- We present code in two different ways:
  - We use a monofont type with no highlighting for most code examples.
  - We use boldface highlighting to emphasize code that is of particularly importance in the present context.

**SOURCE CODE**

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at http://www.wrox.com. Once at the site, simply locate the book’s title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book’s detail page to obtain all the source code for the book.
Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-54071-8.

Once you download the code, just decompress it with your favorite compression tool. Alternatively, you can go to the main Wrox code download page at http://www.wrox.com/dynamic/books/download.aspx to see the code available for this book and all other Wrox books.

ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to http://www.wrox.com and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at www.wrox.com/misc-pages/booklist.shtml.

If you don't spot “your” error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book’s errata page and fix the problem in subsequent editions of the book.

P2P.WROX.COM

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At http://p2p.wrox.com you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.

4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.
PART I
XMPP Protocol and Architecture

► CHAPTER 1: Getting to Know XMPP

► CHAPTER 2: Designing XMPP Applications
Getting to Know XMPP

WHAT'S IN THIS CHAPTER?

➤ The history of XMPP
➤ XMPP networks and connections
➤ XMPP’s three building block stanzas

The eXtensible Messaging and Presence Protocol (XMPP) is, at its most basic level, a protocol for moving small, structured pieces of data between two places. From this humble basis, it has been used to build large-scale instant messaging systems, Internet gaming platforms, search engines, collaboration spaces, and voice and video conferencing systems. More unique applications appear every day, further demonstrating how versatile and powerful XMPP can be.

XMPP is made of a few small building blocks, and on top of these primitives many larger constructions have been made. Within XMPP are systems for building publish-subscribe services, multi-user chat, form retrieval and processing, service discovery, real-time data transfer, privacy control, and remote procedure calls. Often, XMPP programmers create their own, unique constructions that are fitted exactly for the problem at hand.

Most social media constructs that have propelled web sites like Facebook, MySpace, and Twitter into the forefront are also baked into XMPP. Within XMPP, you’ll find rosters full of contacts that create a social graph with directed or undirected edges. Presence notifications are sent automatically when contacts come online and go offline, and private and public messages are the bread and butter application of XMPP systems. Developers will sometimes choose XMPP as the underlying technology layer simply because it gives them many social features for free, leaving them to concentrate on the unique pieces of their application.

The possibilities are vast, but before you can begin, you need to know about XMPP’s different pieces and how they fit together into a cohesive whole.
WHAT IS XMPP?

XMPP, like all protocols, defines a format for moving data between two or more communicating entities. In XMPP’s case, the entities are normally a client and a server, although it also allows for peer-to-peer communication between two servers or two clients. Many XMPP servers exist on the Internet, accessible to all, and form a federated network of interconnected systems.

Data exchanged over XMPP is in XML, giving the communication a rich, extensible structure. Many modern protocols forgo the bandwidth savings of a binary encoding for the more practical feature of being human readable and therefore easily debugged. XMPP’s choice to piggyback on XML means that it can take advantage of the large amount of knowledge and supporting software for dealing with XML.

One major feature XMPP gets by using XML is XML’s extensibility. It is extremely easy to add new features to the protocol that are both backward and forward compatible. This extensibility is put to great use in the more than 200 protocol extensions registered with the XMPP Standards Foundation and has provided developers with a rich and practically unlimited set of tools.

XML is known primarily as a document format, but in XMPP, XML data is organized as a pair of streams, one stream for each direction of communication. Each XML stream consists of an opening element, followed by XMPP stanzas and other top-level elements, and then a closing element. Each XMPP stanza is a first-level child element of the stream with all its descendent elements and attributes. At the end of an XMPP connection, the two streams form a pair of valid XML documents.

XMPP stanzas make up the core part of the protocol, and XMPP applications are concerned with sending and responding to various kinds of stanzas. Stanzas may contain information about other entities’ availability on the network, personal messages similar to e-mail, or structured communication intended for computer processing. An example stanza is shown here:

```xml
<message to='elizabeth@longbourn.lit'
    from='darcy@pemberley.lit/dance'
    type='chat'>
    <body>What think you of books?</body>
</message>
```

In a typical client-server XMPP session, a stanza such as this one from Elizabeth to Mr. Darcy will travel from Elizabeth’s client to her server. Her server will notice that it is addressed to an entity on a remote server and will establish an XMPP connection with the remote server and forward the message there. This communication between servers resembles the e-mail network, but unlike e-mail servers, XMPP servers always communicate directly with each other and not through intermediate servers.

This direct communication eliminates some common vectors for spam and unauthorized messages. This is just one of the many ways in which XMPP is designed for security. It also supports encrypted communications between endpoints through use of Transport Layer Security (TLS) and strong authentication mechanisms via Simple Authentication and Security Layers (SASL).

XMPP is designed for the exchange of small bits of information, not large blobs of binary data. XMPP can, however, be used to negotiate and set up out-of-band or in-band transports, which can move large blocks from point to point. For these kinds of transfers, XMPP functions as a signaling layer.