Develop Cloud-Native Applications

Cloud Architecture Patterns

O'REILLY®

Bill Wilder

Cloud Architecture Patterns

If your team is investigating ways to design applications for the cloud, this concise book introduces 11 architecture patterns that can help you take advantage of several cloud-platform services. You'll learn how each of these platform-agnostic patterns work, when they might be useful in the cloud, and what impact they'll have on your application architecture. You'll also see an example of each pattern applied to an application built with Windows Azure.

The patterns are organized into four major topics, such as scalability and eventual consistency, and primer chapters provide background on each topic. With the information in this book, you'll be able to make informed decisions for designing effective cloud-native applications.

Learn about architectural patterns for:

- **Scalability.** Discover the advantages of horizontal scaling. Patterns covered include Horizontally Scaling Compute, Queue-Centric Workflow, and Auto-Scaling
- **Eventual consistency.** Learn how to maintain data consistency across a distributed system. Patterns covered include MapReduce and Database Sharding
- **Multitenancy and commodity hardware.** Understand how they influence your applications. Patterns covered include Busy Signal and Node Failure
- **Network latency.** Learn how to deal with delays due to network latency. Patterns covered include Colocation, Direct-to-Storage and Multi-Site Deployment

Twitter: @oreillymedia
facebook.com/oreilly

US $24.99 CAN $26.99
ISBN: 978-1-449-31977-9

O'REILLY®
oreilly.com

Cloud Architecture Patterns

Bill Wilder

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Cloud Architecture Patterns

by Bill Wilder

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 (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Rachel Roumeliotis
Production Editor: Holly Bauer

Proofreader: BIM Publishing Services
Indexer: BIM Publishing Services
Cover Designer: Karen Montgomery
Interior Designer: David Futato
Illustrator: Elizabeth O'Connor, Rebecca Demarest

Revision History for the First Edition:

2012-09-20 First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781449319779* for release details.

ISBN: 978-1-449-31977-9

[LSI]

Table of Contents

Preface

This book focuses on the development of *cloud-native applications*. A cloud-native application is architected to take advantage of specific engineering practices that have proven successful in some of the world's largest and most successful web properties. Many of these practices are unconventional, yet the need for unprecedented scalability and efficiency inspired development and drove adoption in the relatively small number of companies that truly needed them. After an approach has been adopted successfully enough times, it becomes a *pattern*. In this book, a pattern is an approach that can be duplicated to produce an expected outcome. Use of any of the patterns included in this book will impact the architecture of your application, some in small ways, some in large ways.

Historically, many of these patterns have been risky and expensive to implement, and it made sense for most companies to avoid them. That has changed. Cloud computing platforms now offer services that dramatically lower the risk and cost by shielding the application from most of the complexity. The desired benefit of using the pattern is the same, but the cost and complexity of realizing that benefit is lower. The majority of modern applications can now make practical use of these heretofore seldom used patterns.

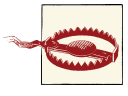
Cloud platform services simplify building cloud-native applications.

The architecture patterns described in this book were selected because they are useful for building cloud-native applications. None are *specific* to the cloud. All are *relevant* to the cloud.

Concisely stated, cloud-native applications leverage cloud-platform services to cost-efficiently and automatically allocate resources horizontally to match current needs, handle transient and hardware failures without downtime, and minimize network latency. These terms are explained throughout the book.

An application need not support millions of users to benefit from cloud-native patterns. There are benefits beyond scalability that are applicable to many web and mobile applications. These are also explored throughout the book.

The patterns assume the use of a cloud platform, though not any specific one. General expectations are outlined in *Scalability Primer* (Chapter 1).



This book will *not* help you move traditional applications to the cloud "as is."

Audience

This book is written for those involved in—or who wish to become involved in—conversations around software architecture, especially cloud architecture. The audience is not limited to those with "architect" in their job title. The material should be relevant to developers, CTOs, and CIOs; more technical testers, designers, analysts, product managers, and others who wish to understand the basic concepts.

For learning beyond the material in this book, paths will diverge. Some readers will not require information beyond what is provided in this book. For those going deeper, this book is just a starting point. Many references for further reading are provided in Appendix A.

Why This Book Exists

I have been studying cloud computing and the Windows Azure Platform since it was unveiled at the Microsoft Professional Developer's Conference (PDC) in 2008. I started the Boston Azure Cloud User Group in 2009 to accelerate my learning, I began writing and speaking on cloud topics, and then started consulting. I realized there were many technologists who had not been exposed to the interesting differences between the application-building techniques they'd been using for years and those used in creating cloud-native applications.

The most important conversations about the cloud are more about architecture than technology.

This is the book I wish I could have read myself when I was starting to learn about cloud and Azure, or even ten years ago when I was learning about scaling. Because such a book did not materialize on its own, I have written it. The principles, concepts, and patterns in this book are growing more important every day, making this book more relevant than ever.

Assumptions This Book Makes

This book assumes that the reader knows what the cloud is and has some familiarity with how cloud services can be used to build applications with Windows Azure, Amazon Web Services, Google App Engine, or similar public or private cloud platforms. The reader is not expected to be familiar with the concept of a cloud-native application and how cloud platform services can be used to build one.

This book is written to educate and inform. While this book will help the reader understand cloud architecture, it is not actually advising the use of any particular patterns. The goal of the book is to provide readers with enough information to make informed decisions.

This book focuses on concepts and patterns, and does not always directly discuss costs. Readers should consider the costs of using cloud platform services, as well as trade-offs in development effort. Get to know the pricing calculator for your cloud platform of choice.

This book includes patterns useful for architecting cloud-native applications. This book is not focused on *how to* (beyond what is needed to understand), but rather about *when* and *why* you might want to apply certain patterns, and then which features in Windows Azure you might find useful. This book intentionally does not delve into the detailed implementation level because there are many other resources for those needs, and that would distract from the real focus: architecture.

This book does not provide a comprehensive treatment of how to build cloud applications. The focus of the pattern chapters is on understanding each pattern in the context of its value in building cloud-native applications. Thus, not all facets are covered; emphasis is on the big picture. For example, in *Database Sharding Pattern* (Chapter 7), techniques such as optimizing queries and examining query plans are not discussed because they are no different in the cloud. Further, this book is not intended to guide development, but rather provide some options for architecture; some references are given pointing to more resources for realizing many of the patterns, but that is not otherwise intended to be part of this book.

Contents of This Book

There are two types of chapters in this book: primers and patterns.

Individual chapters include:

Scalability Primer (Chapter 1)
: This primer explains scalability with an emphasis on the key differences between vertical and horizontal scaling.

Horizontally Scaling Compute Pattern (Chapter 2)
: This fundamental pattern focuses on horizontally scaling compute nodes.

Queue-Centric Workflow Pattern (Chapter 3)
: This essential pattern for loose coupling focuses on asynchronous delivery of command requests sent from the user interface to a processing service. This pattern is a subset of the CQRS pattern.

Auto-Scaling Pattern (Chapter 4)
: This essential pattern for automating operations makes horizontal scaling more practical and cost-efficient.

Eventual Consistency Primer (Chapter 5)
: This primer introduces eventual consistency and explains some ways to use it.

MapReduce Pattern (Chapter 6)
: This pattern focuses on applying the MapReduce data processing pattern.

Database Sharding Pattern (Chapter 7)
: This advanced pattern focuses on horizontally scaling data through sharding.

Multitenancy and Commodity Hardware Primer (Chapter 8)
: This primer introduces multitenancy and commodity hardware and explains why they are used by cloud platforms.

Busy Signal Pattern (Chapter 9)
: This pattern focuses on how an application should respond when a cloud service responds to a programmatic request with a busy signal rather than success.

Node Failure Pattern (Chapter 10)
: This pattern focuses on how an application should respond when the compute node on which it is running shuts down or fails.

Network Latency Primer (Chapter 11)
: This basic primer explains network latency and why delays due to network latency matter.

Colocate Pattern (Chapter 12)
: This basic pattern focuses on avoiding unnecessary network latency.

Valet Key Pattern (Chapter 13)
: This pattern focuses on efficiently using cloud storage services with untrusted clients.

CDN Pattern (Chapter 14)
: This pattern focuses on reducing network latency for commonly accessed files through globally distributed edge caching.

Multisite Deployment Pattern (Chapter 15)
: This advanced pattern focuses on deploying a single application to more than one data center.

Appendix A
: The appendix contains a list of references for readers interested in additional material related to the primers and patterns presented in the book.

The primers exist to ensure that readers have the proper background to appreciate the pattern; primers precede the pattern chapters for which that background is needed. The patterns are the heart of the book and describe how to address specific challenges you are likely to encounter in the cloud.

Because individual patterns tend to impact multiple architectural concerns, these patterns defy placement into a clean hierarchy or taxonomy; instead, each pattern chapter includes an *Impact* section (listing the areas of architectural impact). Other sections include *Context* (when this pattern might be useful in the cloud); *Mechanics* (how the pattern works); an *Example* (which uses the Page of Photos sample application and Windows Azure); and finally a brief *Summary*. Also, many cross-chapter references are included to highlight where patterns overlap or can be used in tandem.

Although the *Example* section uses the Windows Azure platform, it is intended to be read as a core part of the chapter because a specific example of applying the pattern is discussed.

The book is intended to be vendor-neutral, with the exception that *Example* sections in pattern chapters necessarily use terminology and features specific to Windows Azure. Existing well-known names for concepts and patterns are used wherever possible. Some patterns and concepts did not have standard vendor-neutral names, so these are provided.

Building Page of Photos on Windows Azure

Each pattern chapter provides a general introduction to one cloud architecture pattern. After the general pattern is introduced, a specific use case with that pattern is described in more depth. This is intended to be a concrete example of applying that pattern to improve a cloud-native application. A single demonstration application called *Page of Photos* is used throughout the book.

The Page of Photos application, or PoP for short, is a simple web application that allows anyone to create an account and add photos to that account.

Each PoP account gets its own web address, which is the main web address followed by a folder name. For example, *http://www.pageofphotos.com/widaketi* displays photos under the folder name *widaketi*.

The PoP application was chosen because it is very simple to understand, while also allowing for enough complexity to illustrate the patterns without having sample application details get in the way.

This very basic introduction to PoP should get you started. Features are added to PoP in the *Example* section in each pattern chapter, always using Windows Azure capabilities, and always related to the general cloud pattern that is the focus of the chapter. By the end of the book, PoP will be a more complete, well-architected cloud-native application.

The PoP application was created as a concrete example for readers of this book and also as an exercise for double-checking some of the patterns. Look for it at *http://www.pageofphotos.com*.

Windows Azure is used for the PoP example, but the concepts apply as readily to Amazon Web Services and other cloud services platforms. I chose Windows Azure because that's where I have deep expertise and know it to be a rich and capable platform for cloud-native application development. It was a pragmatic choice.

Terminology

The book uses the terms *application* and *web application* broadly, even though *service*, *system*, and other terms may be just as applicable in some contexts. More specific terms are used as needed.

Conventions Used in This Book

The following typographical conventions are used in this book:

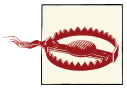
Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

Constant width italic
: Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Cloud Architecture Patterns* by Bill Wilder (O'Reilly). Copyright 2012 Bill Wilder, 978-1-449-31977-9."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online (*www.safaribooksonline.com*) is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley

Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

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

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States 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://oreil.ly/cloud_architecture_patterns*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

This is a far better book than I could have possibly written myself because of the generous support of many talented people. I was thrilled that so many family members, friends, and professional colleagues (note: categories are not mutually exclusive!) were willing to spend their valuable time to help me create a better book. Roughly in order of appearance…

Joan Wortman (UX Specialist) was the first to review the earliest book drafts (which were painful to read). To my delight, Joan stayed with me, continuing to provide valuable, insightful comments though to the very last drafts. Joan was also really creative in brainstorming ideas for the illustrations in the book. Elizabeth O'Connor (majoring in Illustration at Mass College of Art) created the original versions of the beautiful illustrations in the book. Jason Haley was the second to review early (and still painful to

read) drafts. Later Jason was kind enough to sign on as the official technical editor, remarking at one point (with a straight face), "Oh, was that the same book?" I guess it got better over time. Rahul Rai (Microsoft) offered detailed technical feedback and suggestions, with insights relating to every area in the book. Nuno Godinho (Cloud Solution Architect – World Wide, Aditi) commented on early drafts and helped point out challenges with some confusing concepts. Michael Collier (Windows Azure National Architect, Neudesic) offered detailed comments and many suggestions in all chapters. Michael and Nuno are fellow Windows Azure MVPs. John Ahearn (a sublime entity) made every chapter in the book clearer and more pleasant to read, tirelessly reviewing chapters and providing detailed edits. John did not proofread the prior sentence, but if he did, I'm sure he would improve it. Richard Duggan is one of the smartest people I know, and also one of the funniest. I always looked forward to his comments since they were guaranteed to make the book better while making me laugh in the process. Mark Eisenberg (Fino Consulting) offered thought-provoking feedback that helped me see the topic more clearly and be more to the point. Jen Heney provided helpful comments and edits on the earliest chapters. Michael Stiefel (Reliable Software) provided pointed and insightful feedback that really challenged me to write a better book. Both Mark and Michael forced me to rethink my approach in multiple places. Edmond O'Connor (SS&C Technologies Inc.) offered many improvements where needed and confirmation where things were on the right track. Nazik Huq and George Babey have been helping me run the Boston Azure User Group for the past couple of years, and now their book comments have also helped me to write a better book. Also from the Boston Azure community is Nathan Pickett (KGS Buildings); Nate read the whole book, provided feedback on every chapter, and was one of the few who actually answered the annoying questions I posed in the text to reviewers. John Zablocki reviewed one of the chapters, as a last-minute request from me; John's feedback was both speedy and helpful. Don McNamara and William Gross (both from Geek Dinner) provided useful feedback, some good pushback, and even encouragement. Liam McNamara (a truly top-notch software professional, and my personal guide to the pubs of Dublin) read the whole manuscript late in the process and identified many (of my) errors and offered improved examples and clearer language. Will Wilder and Daniel Wilder proofread chapters and helped make sure the book made sense. Kevin Wilder and T.J. Wilder helped with data crunching to add context to the busy signal and network latency topics, proofreading, and assisted with writing the Page of Photos sample application. Many, many thanks to all of you for all of your valuable help, support, insights, and encouragement.

Special thanks to the team at O'Reilly, especially those I worked directly with: editor Rachel Roumeliotis (from inception to the end), production editor Holly Bauer, and copy editor Gillian McGarvey. Thanks also to the other staffers behind the scenes. And a special shout-out to Julie Lerman (who happens to live near the Long Trail in Vermont)