



# Professional Sitecore 8 Development

A Complete Guide to Solutions and  
Best Practices

---

Phil Wicklund  
Jason Wilkerson  
Foreword by John West

Apress®

# Professional Sitecore 8 Development

A Complete Guide to Solutions and  
Best Practices



**Phil Wicklund**

**Jason Wilkerson**

**Apress®**

***Professional Sitecore 8 Development: A Complete Guide to Solutions and Best Practices***

Phil Wicklund  
Minneapolis, Minnesota, USA

Jason Wilkerson  
Minnetrista, Minnesota, USA

ISBN-13 (pbk): 978-1-4842-2291-1  
DOI 10.1007/978-1-4842-2292-8

ISBN-13 (electronic): 978-1-4842-2292-8

Library of Congress Control Number: 2016959582

Copyright © 2016 by Phil Wicklund and Jason Wilkerson

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Acquisitions Editor: Susan McDermott

Developmental Editor: Laura Berendson

Technical Reviewer: Kamruz Jaman, Mike Reynolds, Nick Wesselman

Editorial Board: Steve Anglin, Pramila Balen, Laura Berendson, Aaron Black, Louise Corrigan,

Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham,

Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing

Coordinating Editor: Rita Fernando

Copy Editor: Kezia Endsley

Compositor: SPi Global

Indexer: SPi Global

Cover Image: Selected by Freepik

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springer.com](http://www.springer.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com](http://www.apress.com).

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at [www.apress.com/bulk-sales](http://www.apress.com/bulk-sales).

Any source code or other supplementary materials referenced by the author in this text is available to readers at [www.apress.com](http://www.apress.com). For detailed information about how to locate your book's source code, go to [www.apress.com/source-code/](http://www.apress.com/source-code/).

Printed on acid-free paper

*I dedicate this book to my mother, who unceasingly drilled into me:  
“If there’s a will, there’s a way!”*

*—Phil*

*I dedicate this book to my wife, Anna. Without you, I could never have  
accomplished this. For real. Also, to the Sitecore Community;  
your passion and excitement is infectious and it drives me.*

*—Jason*

# Contents at a Glance

**Foreword ..... xiii**

**About the Authors.....xv**

**About the Contributor .....xvii**

**About the Technical Reviewers .....xix**

**Acknowledgments.....xxi**

**Introduction .....xxiii**

**■ Chapter 1: Getting Started ..... 1**

**■ Chapter 2: Building Your First Component ..... 39**

**■ Chapter 3: Data Templates and Content ..... 77**

**■ Chapter 4: Back-End Dev Architectures ..... 115**

**■ Chapter 5: Improving the Design with Patterns ..... 151**

**■ Chapter 6: Front-End Dev Techniques ..... 187**

**■ Chapter 7: Unit Testing Sitecore ..... 209**

**■ Chapter 8: Search-Driven Solutions ..... 241**

**■ Chapter 9: Programming the Customer Journey ..... 279**

**■ Chapter 10: Sitecore PowerShell Extensions..... 319**

**■ Chapter 11: Extending the Experience Editor ..... 379**

**Index..... 419**

# Contents

- Foreword ..... xiii**
- About the Authors.....xv**
- About the Contributor .....xvii**
- About the Technical Reviewers .....xix**
- Acknowledgments .....xxi**
- Introduction .....xxiii**
- Chapter 1: Getting Started ..... 1**
  - What Is Sitecore? ..... 1**
    - Developer Platforms versus Portal Solutions ..... 3
    - Sitecore’s Key Features and Capabilities ..... 4
    - Infrastructure and Services Architecture Options..... 9
    - A Tour of the Sitecore Admin Interface ..... 15
    - What’s New in Sitecore 8?..... 18
  - The Sitecore Hierarchy ..... 20**
    - Templates ..... 21
    - Content ..... 23
    - Layouts and Components ..... 24
  - Setting Up Your Development Environment..... 27**
    - Tools Every Sitecore Developer Needs ..... 27
    - Setting Up SQL Server ..... 27
    - Installing Sitecore..... 31
  - Summary ..... 38**

- **Chapter 2: Building Your First Component** ..... 39
  - Building a View Rendering ..... 39
    - Working with Layouts and Placeholders..... 39
    - Creating a Component ..... 51
    - Deploying to Sitecore ..... 52
    - Adding a Component to a New Page ..... 54
  - Building a Controller Rendering ..... 61
    - Controller Renderings versus View Renderings..... 61
    - Creating a Controller Rendering ..... 62
  - Summary ..... 75
- **Chapter 3: Data Templates and Content** ..... 77
  - Data Template Fields ..... 77
    - Field Sections ..... 78
    - Field Types ..... 79
    - Custom Field Types..... 83
    - Field Sources ..... 90
    - Field Versioning ..... 90
    - Field Validation ..... 90
    - Custom Field Validators ..... 92
  - Data Template Inheritance..... 95
  - Working with Standard Values ..... 97
    - Configuring Default Values ..... 98
    - Configuring Insert Options..... 103
  - Templates and Configurations..... 103
    - Creating Our Branch Templates ..... 103
  - Content Management Fundamentals ..... 105
    - Where Do You Manage Content? ..... 105
    - Architecting Your Content Tree ..... 107
    - Managing Rich Media..... 109
    - Content Personalization..... 110
    - Content Versioning..... 111

Content Approval and Publishing Workflows .....	112
Content Optimization .....	112
Deleting and Archiving Content .....	112
Content Cloning .....	113
Multilingual Content Translations .....	113
Summary .....	114
<b>■ Chapter 4: Back-End Dev Architectures .....</b>	<b>115</b>
The Model-View-Controller Design Pattern .....	115
Sitecore Modular Architecture .....	123
Setting Up Your Solution .....	128
Stubbing Out a Web Project .....	128
Creating Your First Module .....	142
Deploying Your Code .....	145
Summary .....	149
<b>■ Chapter 5: Improving the Design with Patterns .....</b>	<b>151</b>
Refactoring: Incorporating the Repository Pattern .....	151
Refactoring: Incorporating a Service Layer Pattern .....	158
Refactoring: Incorporating Object-Relational Mapping .....	169
Refactoring: Incorporating Generics .....	174
Refactoring: Incorporating Inversion of Control .....	180
Summary .....	186
<b>■ Chapter 6: Front-End Dev Techniques .....</b>	<b>187</b>
Presentation and Applying Layouts .....	187
Razor, CSS, JavaScript, and Gulp .....	195
Programming Razor and Forms .....	195
Getting CSS and JS on a Page .....	202
CSS and JS Minification with Gulp .....	202
AngularJS .....	206
Summary .....	208



- **Chapter 7: Unit Testing Sitecore .....** **209**
  - Getting Started Unit Testing..... 210
    - Making Tests Easier to Read with FluentAssertions ..... 213
    - Creating Dummy Data, the Easy Way, with AutoFixture ..... 214
    - Isolating Tests with Moq ..... 216
  - Unit Testing a Service..... 218
    - Refactoring to Include a Service Test Harness ..... 222
  - Unit Testing a Controller ..... 225
    - Refactoring to Include a Controller Test Harness ..... 229
  - Sitecore FakeDb ..... 233
  - Summary..... 239
- **Chapter 8: Search-Driven Solutions .....** **241**
  - Setting Up Your Index ..... 241
  - Searching Your Index..... 246
  - Creating a Computed Field..... 259
  - Adding an Autocomplete ..... 268
  - Refining Results with Faceted Search ..... 271
  - Summary..... 278
- **Chapter 9: Programming the Customer Journey .....** **279**
  - Extending Web Forms for Marketers..... 280
  - Programing Goals, Events, and Outcomes..... 291
  - Custom Personalization Rules..... 294
  - Extending Engagement Plans..... 300
  - Programming the xDB Contact..... 304
    - Identifying a Contact..... 304
    - Using Custom Tags ..... 306

Using Custom Extensions .....	307
Programing the Out-of-the-Box Facets .....	309
Building Custom Facets.....	311
Federated Experience Manager .....	314
Summary.....	317
<b>■ Chapter 10: Sitecore PowerShell Extensions.....</b>	<b>319</b>
Installing the PowerShell Extensions .....	320
Prerequisites.....	320
Installation Wizard .....	320
Module Highlights .....	327
PowerShell Console.....	328
PowerShell ISE .....	328
PowerShell Toolbox.....	329
PowerShell Reports .....	330
Out-of-the-Box SPE Features .....	330
Bundled Tools .....	330
Audit Reports.....	338
Bonus Features.....	340
Programming Sitecore with PowerShell .....	341
Learning PowerShell as a Language .....	341
Components of SPE .....	346
Building Your First Module.....	347
UI Settings .....	370
SPE Remoting .....	372
Resources.....	376
Summary .....	377

- **Chapter 11: Extending the Experience Editor ..... 379**
  - Configuring Dynamic Placeholders ..... 379
  - Customizing Renderings in the Experience Editor ..... 382
  - Custom Experience Editor Buttons and Properties..... 389
    - Configuring Custom Component Toolbar Buttons ..... 392
    - Configuring Custom Component Parameters ..... 396
    - Configuring Custom Component Experience Editor Buttons..... 401
  - Summary ..... 418
- Index..... 419**

# Foreword

Since the dawn of Sitecore, developers have needed introductory orientation as well as clear, comprehensive answers to their burning technical questions about the world's leading ASP.NET web content and experience management platform. But as with many emerging platforms, information lagged behind enthusiasm. In the beginning, there was disassembly, with most Sitecore solutions entirely dependent on the pioneering work of our original savior, Lutz Roeder.

Next came the Sitecore Developer Network forums and the advent of Sitecore developer documentation in PDF format, largely and somewhat unfortunately contributed by yours truly, the former @sitecorejohn. It is not even worth mentioning that those 500 pages of marketing fluff later written by someone with a name similar to mine who worked at their kitchen counter for three months in the fall of 2011. Today, there are so many sources of information about Sitecore that just identifying them can be a full-time job, with many of the most justified and valiant voices lost in the cacophony of distraction.

These are some of the reasons why I welcome *Professional Sitecore 8 Development* as a means to restore sanity to an increasingly disjointed archipelago of information. This book provides the best starting point available today for developers new to Sitecore technologies, including the web Content Management System (CMS) and Experience Platform (XP).

For anyone unfamiliar with the platform, this book covers all of the fundamentals that a developer new to Sitecore needs to understand from the beginning. For those familiar with Sitecore, it provides a wealth of information that even seasoned software engineers should find invaluable.

In particular, this book outlines core concepts about developing with Sitecore and demonstrates how to incorporate modern design patterns into projects in order to deliver high-quality, scalable, and testable solutions. The book shows how to apply Helix design and development principles, a set of overall objectives and conventions from Sitecore itself that support the objective of achieving optimal and maintainable solutions. On top of this, it demonstrates the strongest aspects of development as implemented in the Habitat project also provided by Sitecore. This type of foundation becomes more critical as the complexity, importance, and interconnectedness of WCM continues to increase in order to enable the next generation of digital marketing practices.

As the former Chief Technology Officer for Sitecore, I have known the authors of this new work, Jason and Phil, since they appeared in the community years ago. To his tremendous credit, Jason has rapidly achieved the coveted Sitecore Most Valuable Professional (MVP) status, while Phil manages an entire engineering team responsible for implementing solutions based on Sitecore digital marketing technologies. Phil previously wrote the *Practical Sitecore 8 Configuration and Strategy* book published in 2015. The insights in this new work come from their real-world experiences implementing advanced solutions for Sitecore customers in North America.

I continue to be impressed with the Sitecore software and ecosystem and was delighted when Jason requested that I write the forward for this volume. This book makes a significant and important contribution that is now available to the entire Sitecore community, which I have watched grow, evolve, and progress at an ever-increasing rate as the software cadence and feature set has accelerated. This book may just be the best starting point for implementing what I still consider to be the best Web Content Management System available today.

I have an additional reason to be thankful for such a book. In my new role as senior vice president of experience platforms for POSSIBLE at @cmsvortex, I now benefit from these efforts as would any consumer rather contributing to them as a producer. As I continue to develop in my own career, reading draft chapters of this book has rewarded my time investment by providing a valuable training resource.

Regardless of your background, whether with or without Sitecore, I am confident that you will benefit from this well-structured and researched volume. As a final word, I would like to personally thank all of you for being members of the Sitecore community!

—John West  
Senior Vice President, Experience Platforms, POSSIBLE  
Former Chief Architect and CTO, Sitecore

# About the Authors



**Phil Wicklund** is the manager of digital technology services at RBA, Inc.—a digital and technology consultancy.

His passion is leadership. It sounds simplistic, or worse, cliché, but his passion runs deep and no other word hits closer to the mark. He loves leadership because he loves casting a vision for a better future; a future of personal growth, betterment of the human condition, ways to create future value for companies, and a future of shared accomplishment with his teammates. Such a future is one worth fighting for, and he has never been afraid to take the path of maximum resistance to make a goal a reality.

He has led and managed countless large-scale, enterprise projects as a project manager and/or technical lead. He also has experience managing a staff of 36+ consultants. Additionally, he's been blessed with countless opportunities to interface with and advise CXO-level customers on creating value for their companies.

He has also written three technology books; teaching is his second passion. He loves the feeling at the end of a day when he has been able to apply his expertise to help someone in need. He sincerely hopes this book helps a lot of people learn a tool that isn't always the easiest to learn.

He is currently in his third year of doctoral studies at the University of St. Thomas. Learning, you could say, is his third passion. He is working to extend his studies in technology and business by deepening his understanding of organizational leadership, working toward a dissertation that combines the three disciplines.



**Jason Wilkerson** is the Discipline Lead of the Sitecore practice at RBA, Inc., a Sitecore Platinum Partner. In 2016, he had the honor of being selected as a Sitecore Technology MVP, primarily due to his commitment to the Sitecore community as a speaker, blogger, and thought leader.

Jason's passions lie deep in architecture and technology. With an unrivaled eye toward quality and elegance, he strives to make every customer's solution the most intuitive and maintainable it can possibly be—not only from a technical standpoint, but also from the user's perspective.

With almost two decades of experience building enterprise-class software, as well as leading teams of developers, project managers, business analysts, and quality assurance testers, he's been able to assemble an impressive list of successful projects, leaving nothing but satisfied customers behind him.

## ■ ABOUT THE AUTHORS

In his free time, Jason is also an avid singer, guitar player, and bassist. From a very early age, the same passion that has driven him in technology has driven his love for music and the honing of his talents.

It is this creative outlet, he believes, that provides a unique perspective in the realm of technology. His training and education in computer science provides the foundation for the engineering side of software, but his creative nature serves as the backdrop for creating intuitive, user-centric customer solutions.

# About the Contributor



**Michael West** (author of Chapter [10](#)) is a passionate technologist and Application Architect at Concentra—a national healthcare company in Dallas, Texas. In his roles as developer, manager, designer, and innovator, he begins with empowering others and ends by helping them succeed. During Michael's time working in IT, he has taken on responsibilities in UX design, UX development, DevOps, environment governance, and release management. He enjoys taking on new challenges and is known for bringing innovative solutions to the software industry. Michael received the 2015 and 2016 Sitecore Technology MVP award in recognition of his active contribution to the Sitecore community. He is a key contributor to the popular open source module Sitecore PowerShell Extensions (SPE), found on the Sitecore marketplace. Teaching others is another passion of his. Many of his thoughts and ideas are shared with the community through user groups, video tutorials, and written material.



# About the Technical Reviewers



**Kamruz Jaman** is an independent Sitecore specialist and four-time Sitecore MVP. Kamruz has over 15 years of development and architecture experience using the Microsoft technology stack, having spent the majority of his career in industries such as finance, technology, ecommerce, pharmaceuticals, distribution, insurance, and travel. Having worked exclusively with Sitecore for the past seven years, Kamruz is heavily involved in the Sitecore community and has spoken at user groups and conferences. He also likes to blog and share his knowledge with the community via Slack, Stack Overflow, and the Sitecore forums.



**Mike Reynolds** is a Sitecore MVP with over eight years of Sitecore architecture and development experience and is an independent consultant with a proven track record of pushing the envelope of the Sitecore platform to deliver best-of-breed solutions for clients. A well-known thought leader in the Sitecore community, Reynolds was recognized as a Sitecore Technology MVP in 2014, 2015, and 2016 for his community contributions.



**Nick Wesselman** is a four-time Sitecore MVP and manages product development for Active Commerce, “Sitecore ecommerce done right.” Nick has been developing content management and ecommerce applications for over 15 years and has contributed to Sitecore modules such as WeBlog and Sitecore PowerShell Extensions

# Acknowledgments

We would first like to acknowledge the support and help we received from our technical editors—Kamruz Jaman, Mike Reynolds, and Nick Wesselman—as well as our contributing author, Michael West. The impact of their feedback, support, and guidance over the past six months cannot be understated. Gentlemen, you have our sincerest thanks!

Additionally, we wish to offer our thanks to the Apress team, most notably Susan McDermott and Rita Fernando. Susan, thank you for giving us the opportunity to write this book and partner with such a reputable brand, Apress, once again. Rita, thank you for your patience, guidance, and help these past months as we slogged through the material together. Apress is truly a best-in-class publisher, and we're honored to continue our partnership.

Lastly, we wish to thank the Sitecore community itself. The community is large and robust, filled with thousands of thought leaders who really deserve the credit for this book. Everybody knows you can find whatever you need online these days. This book is really an assemblage of all the great materials and best practices championed by so many fine, hard-working individuals. We raise our glass to YOU!

# Introduction

We believe this book represents the most foundational, core aspects of Sitecore that every Sitecore developer needs to understand. The goal of this book is to take developers who are new to Sitecore all the way from setting up their local development environment to more advanced topics, such as pipelines or extending the Experience Editor, and many things in between.

The earlier chapters focus on an introduction to the platform, including its core capabilities and features, as well as getting started with development. Later chapters start layering in “best practices” that even seasoned Sitecore veterans will find beneficial.

Chapters toward the end of the book start to home in on key capabilities, such as search and the marketing features, where you will really start to dive into the particulars.

Again, our goal is to lay a foundation of “core” Sitecore development concepts. Many of these are for beginners, but many are still quite advanced. The Sitecore platform is sooooo big, there are just too many nooks and crannies; we could never hope to cover all of them. Literally an entire book of just “Advanced Sitecore Development Techniques” could be written if ever that were to be attempted (any takers?). Hopefully you agree that the following 11 chapters represent the most important topics each and every Sitecore developer should understand and master. Beyond these, we look to the Sitecore community to fill the gaps with blog posts or future books.

---

■ **Note** This book isn’t meant to provide production-ready code samples, but rather it teaches the patterns for successfully building quality, testable, maintainable code. Our goal is to teach you how to fish, so to speak.

---

## How This Book Is Structured

- Chapter 1, “Introduction to Sitecore,” will answer the question, “what is Sitecore?” Additionally, the chapter quickly gets pragmatic by showing you, step-by-step, how to set up your development environment.
- Chapter 2, “Building your First Component,” helps you do just that, build your first component. Components are the most basic building block of any page within Sitecore. This chapter shows you how to do the stereotypical “hello world” example.
- Chapter 3, “Data Templates and Content,” extends component development by focusing on the core foundational aspects of any Sitecore deployment: templates. Templates are the basis of everything in Sitecore, and you need a solid understanding of them before you advance in the book.
- Chapter 4, “Back-End Dev Architectures,” extend that first “hello world” example you wrote in Chapter 2 by discussing common patterns for Sitecore development, such as MVC and modular architectures.

- Chapter 5, “Improving the Design with Patterns,” extends Chapter 4’s discussion around back-end techniques by adding many best practices, such as ORMs, dependency injection, and abstracting through patterns.
- Chapter 6, “Front-End Dev Techniques,” focuses on the front-end developer a bit by discussing layouts, placeholders, CSS, and JavaScript.
- Chapter 7, “Unit Testing Sitecore,” walks you through the ins and outs of unit testing in and around Sitecore, including popular tools such as AutoFixture, Moq, and FakeDb.
- Chapter 8, “Search-Driven Solutions,” focuses on Search: how to index, search, and refine content.
- Chapter 9, “Programming the Customer Journey,” focuses on the marketing capabilities of Sitecore and how to develop and customize them.
- Chapter 10, “Sitecore PowerShell Extensions,” shows you how automation helps with many of the administrative capabilities within Sitecore, as well as how to program Sitecore with PowerShell.
- Chapter 11, “Extending the Experience Editor,” explains how to customize the Experience Editor to make it friendlier for content authors and marketers.

## Contacting the Authors

Phil Wicklund:

- [phil@sitecoreconfig.com](mailto:phil@sitecoreconfig.com)
- <http://sitecoreconfig.com>
- <https://linkedin.com/in/philwicklund>
- Twitter: @philwicklund

Jason Wilkerson:

- [jasonmwilkerson@gmail.com](mailto:jasonmwilkerson@gmail.com)
- <http://citizensitecore.com>
- Twitter/Slack: @longhorntaco
- <https://linkedin.com/in/jasonmwilkerson>

## CHAPTER 1



# Getting Started

What do you need to know about Sitecore before getting your hands on the code? Whenever I read development books, I always want to get into the code as quickly as possible. Sure, there's always tons of great background information that helps you understand the code better. Sitecore is no exception. In fact, that's part of the problem with writing a Sitecore book—it's just such a huge platform!

Critics of this chapter might comment that we've skipped too much introductory material for a first chapter. This chapter is really NOT an introduction to Sitecore. This chapter is an introduction to Sitecore development. As such, we'll cover some of the most fundamental aspects of Sitecore, but just enough to get you to a point that you can understand the code, what it's doing, and how it's working.

This means answering the "What is Sitecore" question from a high-level perspective. The goal here is to give you a 30,000-foot view of Sitecore's capabilities, its server roles, and how data flows in Sitecore. Additionally, we'll introduce you to Sitecore's administrative user interface, the Launchpad. Finally, we'll discuss Sitecore features in relation to previous versions of Sitecore to give you a view back.

Next we'll cover the Sitecore hierarchy. Namely, you need to know the core "building blocks" of Sitecore. You'll create data templates, layouts, placeholders, and components during the course of building out your Sitecore digital properties. Understanding these four building blocks represents the most foundational knowledge you need to have before you see a single line of code. Without that backdrop, the code won't make sense.

With that foundation laid, the next step will be to set up your development environment. This includes setting up a local SQL server, installing Sitecore, and getting the necessary tools in place, most obviously Microsoft Visual Studio.

With all that under your belt, you can finally start building something! However, we're going to save that for Chapter 2.

## What Is Sitecore?

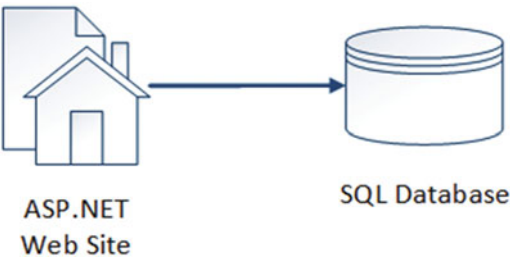
Sitecore is HUGE, Huge, huge! Where do you even start? You could literally write an entire book on just the features and capabilities of Sitecore without ever seeing a single line of code. Oh wait, I did that; it's called ***Practical Sitecore 8 Configuration and Strategy*** (Apress, 2015), to be specific (#shamelessplug).

Most of the features and capabilities discussed in that book are there to help the marketer and/or content administrator do their job, create great user experiences for their customers, and manage their campaigns. This book, however, is geared toward the developer, not the marketer per se. The best place to start, in that case, is with a gross oversimplification of Sitecore to ensure we're all on the same page.

---

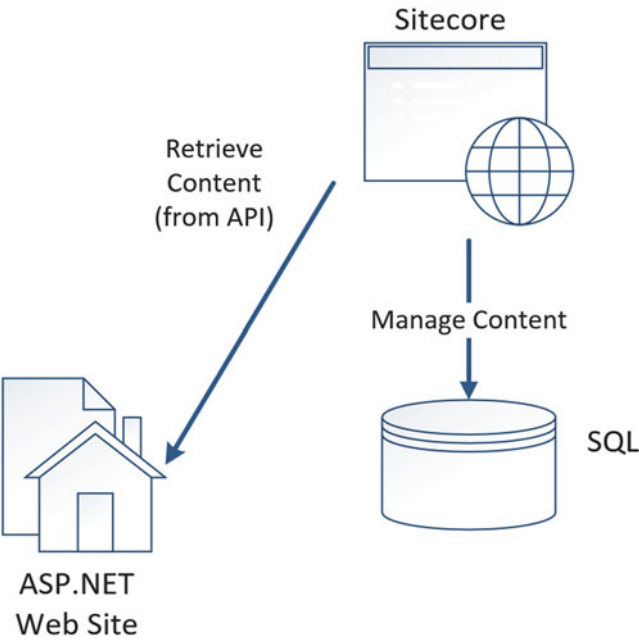
**Electronic supplementary material** The online version of this chapter (doi:[10.1007/978-1-4842-2292-8\\_1](https://doi.org/10.1007/978-1-4842-2292-8_1)) contains supplementary material, which is available to authorized users.

Sitecore is a Web Content Management (WCM) solution on steroids. You could argue that Sitecore is really a Customer Experience Platform (CXP, or better yet WCM 2.0). However, arguing that would be going down the rabbit hole I just argued we shouldn't go down. Given that, Sitecore is two things at its core: 1) a web site you build and 2) a database within which you can manage content. The most simplistic Sitecore architecture diagram you could ever create is shown in Figure 1-1.



**Figure 1-1.** Sitecore, at its most basic definition, is a web site with a database

You'll see in Figure 1-2 that Sitecore itself isn't even depicted in Figure 1-1; however, that requires more explanation that is best saved for later. The point, fundamentally, is that a Sitecore site is just an ASP.NET web site that connects to a database. Sitecore helps you manage the data in that database; Sitecore is not the web site your customers will experience—that is something you build yourself.



**Figure 1-2.** Sitecore really is a web site within your web site that's used to manage content

Confused? You might be. Many people bring a lot of expectations and baggage to Sitecore. Namely, that Sitecore comes with a “bag of tricks” such as widgets, page templates, etc., that you can just drag and drop and in five minutes have a site, WordPress-style. For better (not really worse), Sitecore is just about the opposite of WordPress as you could ever get. To help explain the difference, let's compare the two types of solutions next: developer platforms versus portal/web solutions.

## Developer Platforms versus Portal Solutions

Sitecore is a developer platform, not a canned “portal solution” such as many popular horizontal portal solutions out there, such as SharePoint, WebSphere or Liferay. Some of those products make claims to be a platform, but you’ll see in a bit why Sitecore stands out as unique in this regard. This topic of “platform” versus “portal package” can be confusing, especially since Sitecore is often used as a platform for portals.

An analogy might help newbies understand how Sitecore differs from some of these other popular portal solutions. Let’s use the example of a manufactured/mobile home versus a custom-built home.

Features of manufactured homes include:

- *Fast and cheap*: The most obvious point to make with mobile homes is they are fast to set up and are cheap. You literally go to the lot, pick one out with the features you want, and then they drive it to your lot and drop it off.
- *Easily configurable*: Mobile homes have many options and configurations to select from, and they’re all optimized for people with little training to understand and use.
- *Rigid*: That configurability can turn to rigidity when your unique needs are not perfectly expressed in those options.
- *Customizable, to a point*: Yes, you can customize your mobile home, but doesn’t that defeat the entire point of buying a mobile home? If you do customize it, you run into a rapid decline in benefit with ever-growing costs, as you push the home beyond its original intent.
- *Ugly*: You don’t buy a mobile home to make a statement of elegance. You buy it for pure utilitarian purposes.

These seem like pretty good points and you can start to understand the tradeoffs that come with packaged portal solutions. You can get a *big* head start with them, and they all come with a nice “bag of tricks” to help you build sites quickly. However, this is never *YOUR* site; it’s always built from a template and never quite captures your unique brand, perspective, or characteristics. From a development perspective, such sites are a pain. You’re forced to play in a box, and customizing beyond the scope of that box is often painful and expensive, to the point where you begin to wish you had chosen a platform, not a packaged solution. With that segue in mind, let’s look at the features of a home foundation (software platform):

Features of a foundation:

- *Access to core utilities (e.g., water, sewage, power, and gas)*: If you buy a homestead lot (licensing), you get access to all kinds of goodies that help your life as a homebuilder. You can hook right up the water supply and power without having to worry about generating your own electricity. Sitecore provides a lot of rich “core” capabilities such as a personalization and rules engine, analytics, and content management, all without pigeon-holing you.
- *Solid foundation using industry-standard approaches (block, rebar, etc.)*: You know that *any* homebuilder can build on this foundation because it uses industry-standard home building techniques (ASP.NET, MVC, design patterns, test-driven-development, etc.). Block, rebar, standard dimensions, etc. enable anyone who swings a hammer to jump in and get to work without much specific training. Being a “Sitecore developer” is not far from being a typical ASP.NET MVC developer.

---

■ **Note** While being a “Sitecore developer” is not far from being a typical ASP.NET MVC developer, Sitecore MVC doesn’t fully follow the principles of ASP.NET MVC. There is a bit of a learning curve for even seasoned ASP.NET MVC developers when starting to develop a Sitecore MVC-based solution.

---

- *Ability to customize the foundation and blaze your own path (use solar panels instead of the electric grid):* Maybe you hate the idea of using the city’s sewage system and you want to build your own septic. Not a problem. Sitecore has a vast network of “pipelines” (pun intended) that you can tap into and change the way Sitecore works in a supported, standard fashion.
- *You get to build whatever you want on top of that foundation:* Sitecore makes no point to tell you what your house should look like. Bungalow or Rambler? Victorian or Craftsman? Single-page-application (SPA), ASP.NET MVC, or good ol’ fashioned ASP Web Forms? Doesn’t matter (although we’ll make a point in this book to outline best practices) to Sitecore. You decide what you build, what it looks like, and how it works.

You can be a productive Sitecore developer in just a few months. However, it takes years to understand the nuances of SharePoint to make the same claim. That’s the beauty of a “platform” versus a packaged “portal solution”.

Another key differentiator of Sitecore is how it separates content from its presentation. Many portal solutions combine the two, to the point where it’s commonplace for the content itself to be HTML. Sitecore, on the other hand, treats it as severely as the separation of church and state. In Figure 1-2, you’ll notice an additional element not shown in Figure 1-1—Sitecore itself. Sitecore is an administrative user interface that sits in a folder of your web site, allowing content authors to administer content and marketers to do things marketers do, such as manage their campaigns, optimize pages, and so forth. Sitecore is *not* your site! Your site is your site! You build your ASP.NET web site to look and feel however it ought to look and feel, in whatever way best expresses your company’s brand. Sitecore doesn’t get in the way of that by forcing your presentation details to conform to Sitecore rules (like you so often find in portal solutions); Sitecore is totally separate from your site’s presentation.

---

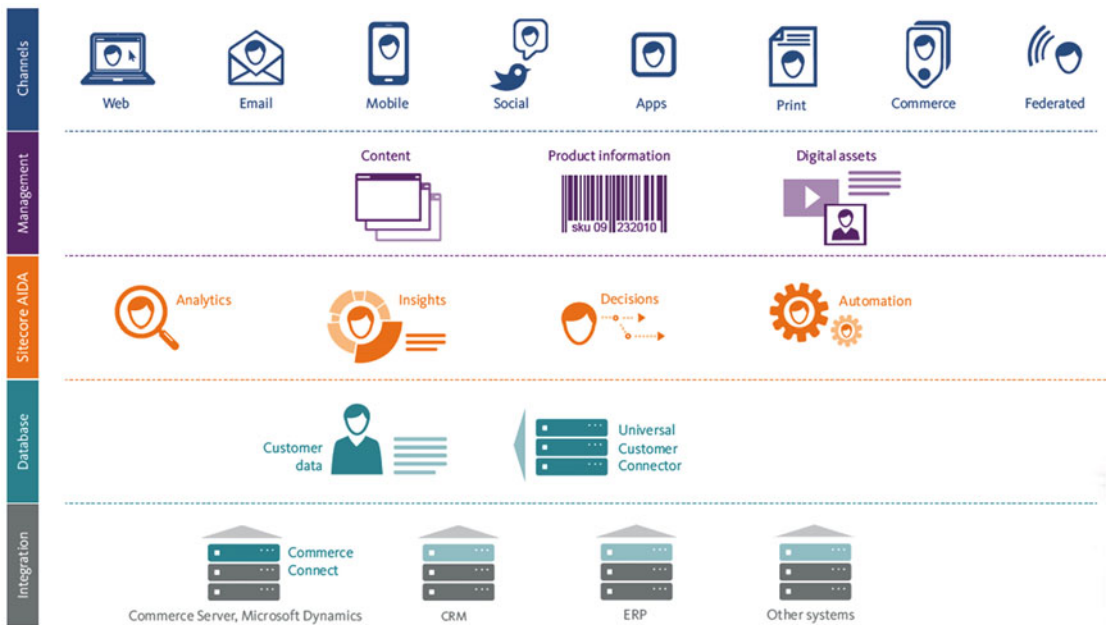
■ **Note** While I may have been speaking in extremes to make a point in the previous paragraph, Sitecore does provide a WYSIWYG (what-you-see-is-what-you-get) editor called the Experience Editor. This editor makes it very easy for marketers to edit content without having to guess what it’ll look like after they click Publish. However, the editor is smart enough to reflect your site’s custom presentation details while still being totally separate from your actual site. More on the Experience Editor in Chapter 2.

---

## Sitecore’s Key Features and Capabilities

It’s time to talk about what Sitecore can do, now that you have a general sense of what Sitecore is. Sitecore has published a helpful Solution Architecture diagram (Figure 1-3) that is a good tool to use to review Sitecore’s architecture, features, and capabilities.





**Figure 1-3.** Sitecore's de facto solution five-tiered architecture

Sitecore's architecture is broken into five main service tiers: Channels, Management, Sitecore AIDA, Database, and the Integration Tier. The specific features and capabilities of Sitecore are spread throughout this architecture in various forms.

## Channel Layer

Channels are the points of engagement from which customers enter as they interact with your brand. Sitecore provides a host of features and capabilities that maximize engagement across the following channels.

Key features and capabilities of the Channel layer include:

- **Email Experience Manager (EXM):** With EXM, you can author e-mails, set personalization rules for those e-mails, and track e-mail click-throughs and other analytics.
- **Print Experience Manager (PXM):** PXM allows you to dynamical generate print assets, such as brochures or manuals, on the fly. This is helpful because your brochures and digital properties can share the same content, for example, and you can even generate custom, personalized assets for customers on demand.
- **Federated Experience Manager (FXM):** FXM allows you to share the same content and analytics across all your digital properties, regardless if they are on the Sitecore platform.
- **Social connections:** Provides integration into Facebook, Twitter, etc. for your digital properties.

- *Sitecore Commerce Connect*: This interface allows any commerce system that implements the interface to natively talk to Sitecore. Your web site or app may need to submit an order, get the number of items in a shopping cart, or check inventory of a product. All these actions and more can be developed through this interface, thus making the backend commerce solution interchangeable.

Those channels are:

- *Web*: The web channel is the core channel users will engage in from a browser experience. The browser/web channel also becomes the primary mechanism to administer Sitecore.
- *E-mail*: Sitecore engages the e-mail channel primarily through the Email Experience Manager module. This module allows marketers to create e-mail campaigns by leveraging customer profile data, preferences, or segments. The Email Experience Manager also allows for easy tracking of customer engagement within the campaign.
- *Mobile*: Sitecore accommodates a separation of content and presentation details. This makes it very easy for developers to create compelling mobile web experiences within Sitecore. Sitecore also accommodates device detection for situations where an adaptive experience is preferred.
- *Social*: Sitecore allows for easy integration with Social channels such as Facebook and Twitter. When you post content in Sitecore, you can configure it to automatically post to one or many pages in Facebook. You can also allow users to log into your site with their federated Facebook credentials. There is a ton of possibility for a marketer in this channel, from finding brand advocates, to capturing customer intelligence, to proactively seeking feedback and advice.
- *Commerce*: Sitecore has rich commerce capabilities to tap into your customer shopping behaviors across all your channels. Sitecore Commerce 8 is a new capability released in early 2015 that enables B2C and B2B ecommerce managed within Sitecore and a Microsoft Dynamics backend.
- *Print*: Print is another channel that customers interface with regularly, such as through brochures and print ads. Print Experience Manager, a feature of Sitecore, allows marketers to easily capture personalization data to create compelling, targeted print experiences for customers.
- *Apps*: Mobile apps have become a critical way customers interact with your brand, especially when considering ecommerce. One of the benefits of Sitecore is the ease of access to centralized content. Content for the web can be reused for content in an app. additionally, you can use Sitecore's Federated Experience Manager to tie in App engagement data into Sitecore to, again, create a single view of how your customer is experiencing your brand.
- *Federated*: Federated channels might be channels you don't own but still represent a view into your customer. Perhaps a partner web site captures valuable information about your target customers. Through the use of Federated Experience Manager, you can easily capture those very important, non-Sitecore managed experience's data and combine it with native representations of your customer to create a truer representation of your customer's engagement across all your channels (and even partners).

## Management Layer

The Management layer of Sitecore is the core of the experience platform. This is where content administrators manage content. It is also where they configure personalization rules, view engagement scores, and view reports. It is also what collects, evaluates, manages, and analyzes customer intelligence from patterns, behaviors, and decisions gathered across your channels. Additionally, from within Sitecore, you can manage all of your company's digital assets, in one place, used across all channels.

Key features and capabilities of the management layer include:

- *Unified content management:* Sitecore provides a one-stop-shop to manage all your content across all your digital properties and apps. Additionally, you get access to powerful content editing tools, such as the Experience Editor, to make content administration a snap.
- *The Sitecore Launchpad:* The Launchpad is your portal to all things Sitecore Administration; see “A Tour of the Sitecore Admin Interface” later in this chapter.
- *Viewing analytics and reports:* Sitecore captures every interaction a customer makes on your digital properties. You can view all these analytics and reports from the Launchpad.
- *Digital asset management:* Sitecore provides a media library capability to manage assets such as videos, imagery, and documents.

## Sitecore AIDA Layer

Sitecore AIDA (Analytics, Insights, Decisions, and Automation) encompasses all the sexy features and capabilities over which marketers salivate. Therein lies the power to do advanced content testing, such as A/B or multivariate testing. You can also configure customer engagement goals, triggers, and reports to show progress and campaign outcome attainment. At its core, AIDA provides a rich suite of reporting and analytic capabilities so your marketing team can discover new insights and customer behavior patterns.

Key features and capabilities of the management layer include:

- *Experience profiles:* Profiles in Sitecore become the 360-degree view of a customer. Every event, goal, outcome, and interaction is tracked in the customer's profile. This provides rich data from which to personalize the experience later.
- *Personalization:* Every interaction in Sitecore can be personalized to fit each customer's unique needs, wants, pain points, buying patterns, etc. Sitecore provides two main ways to personalize content: rules-based personalization (e.g., if A, then B), and behavioral personalization, where Sitecore takes into consideration all the interactions in the current session or across multiple sessions to determine which content would best suit that customer (e.g., machine learning of sorts).
- *Experience optimization:* “Test everything” is Sitecore's motto. You can run tests that compare one page to another or compare multiple versions of a given page to find the combination that creates the most engagement from users. This is often called multivariate, or “A/B,” testing.
- *Engagement plans:* Otherwise called “experience automation,” engagement plans allow you to build a “customer journey”. These plans enable Sitecore to facilitate that journey without outside intervention from a marketer (sending e-mails, etc.), therefore advancing a customer further down the buying cycle.

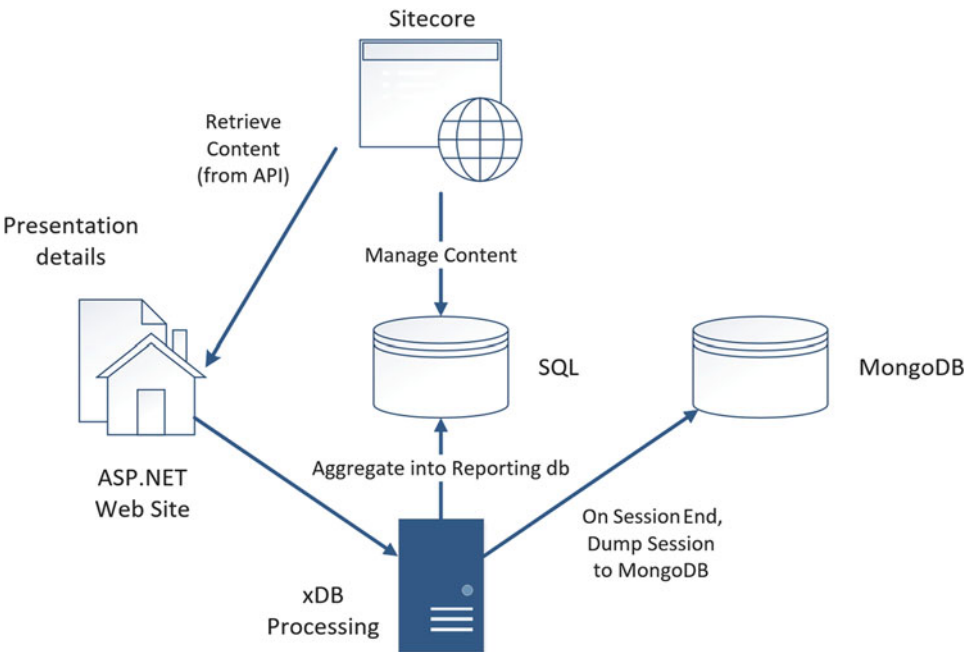
## Sitecore Database Layer

The Database layer is the heart of Sitecore. Data is the lifeblood of your experience marketing solutions. It is otherwise impossible to deliver a highly relevant, personalized, channel optimized customer experience. Sitecore’s xDB architecture leverages MongoDB for massive data capture and retention capabilities. This allows high volume, high velocity, unstructured data to be captured and become actionable in real time.

Key features and capabilities of the database layer include:

- *xDB (MongoDB)*: Mongo is a NoSQL database that is highly denormalized, which enables massive scaling of flexible, unstructured data. Out-of-the-box, *tons* of data about a user is captured, but you can easily extend the xDB schema to track all kinds of custom properties and events, against which you can report or personalize.
- *Scalability*: Sitecore is built on the need to scale to support millions of visits a day. You can scale the web site to run on as many Content Delivery (CD) servers as you need (more in the next section on infrastructure), and you can scale the database servers as well with SQL availability groups and the like.
- *High availability (HA)*: Like scalability, Sitecore supports HA scenarios. SQL availability groups, load balanced CD servers, geographically-distributed delivery clusters, and cloud-based deployments can provide the 99.95% uptime you’re looking for.

Figure 1-4 shows more elements you didn’t see in Figures 1-1 and 1-2, namely MongoDB and the corresponding xDB processing server. As was previously mentioned, Sitecore logs all interactions for a current user’s browsing session to SessionState. Once the session ends (when Session.Abandon is called manually or when the timeout value configured on SessionState in web.config is exceeded), data from SessionState is flushed to the collections database in MongoDB. A separate server, namely the xDB processing server, crawls through all the new data in MongoDB, aggregates that data, and writes it into a reporting database for analytics purposes.



**Figure 1-4.** MongoDB adds a few extra pieces to the puzzle, namely a MongoDB database and an xDB processing server

## Integration Layer

The Integration layer provides connectivity into backend systems such as your CRM or ERP systems. Tying these systems together forms the basis of a single view of your customer, as well as provides the data you need for more complex personalization rules (such as past purchases).

Key features and capabilities of the integration layer include:

- *CRM connectors*: Sitecore has connectors for Microsoft Dynamics CRM as well as for Salesforce to support two-way contact synchronization and submit actions for Web Forms (such as creating a lead in CRM).
- *Sitecore Commerce Connect*: This is an interface that allows any commerce system that implements the interface to natively talk to Sitecore. Your web site or app may need to submit an order, get the number of items in a shopping cart, or check the inventory of a product: All these actions and more can be developed through this interface, thus making the backend commerce solution interchangeable.
- *SharePoint connectors*: Sitecore has a connector to SharePoint to help publish content in SharePoint to Sitecore after it's approved.
- *Social connections*: Provides integration into Facebook, Twitter, etc. for your digital properties.

## Infrastructure and Services Architecture Options

The goal of this section is to give developers a high-level understanding of Sitecore's infrastructure services and deployment options. Many developers are also asked to set up these configurations, but it's really a systems administrator job. You have to wear multiple hats, which is a common ask of many Sitecore professionals.

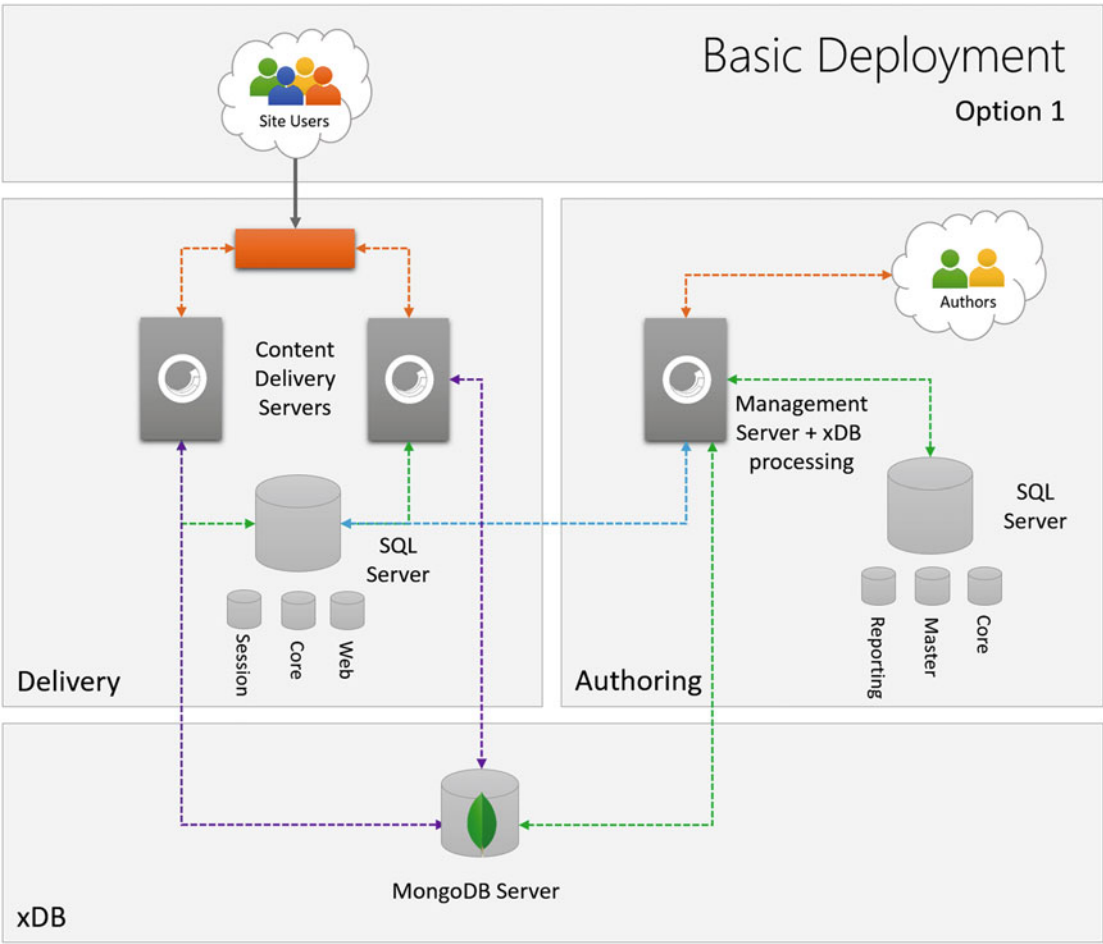
There are innumerable options when considering a deployment and each company should go through an exercise to determine which architecture best meets their needs; however, there are a few core concepts that apply to everyone or are common points to extend your unique needs. First we'll discuss the six types of servers/roles that Sitecore requires, and then we'll discuss three common deployment options worth considering.

- *Content management server*: This server is where authors and marketers go to edit content and view the Launchpad. It is common to put this server out of content-delivery rotation (even behind a firewall) to ensure that external users cannot access the Sitecore administrative interface.
- *Content delivery server*: This server serves up approved and published content to your customers.
- *SQL Server database server*: You need one or more SQL servers for the web, master, core, and reporting databases.
- *MongoDB database server*: Every interaction is dumped into a MongoDB database to be reported on later. Given the high velocity of data, MongoDB is used for performance and scalability needs.
- *xDB processing server*: This server pulls data out of the MongoDB database and stores it in a relational database optimized for reporting.
- *Indexing/search servers*: You may need one or more search and indexing servers depending on your chosen platform (Lucene, SOLR, Coveo, etc.) and architecture.

**Note** There are many other infrastructure considerations, such as cloud versus on-premises, physical versus virtual, IaaS versus PaaS, capacity needs, networking and security, etc., that you should consider as well, but they are far beyond the scope of this book.

## Basic Deployment

Figure 1-5 shows a common deployment for a small-to-medium-sized Sitecore installation. It features two content delivery servers and a SQL cluster to ensure high availability of the web site, one content management server for authors which doubles as the xDB processor, and a MongoDB server for the database.



**Figure 1-5.** A basic deployment minimizes licensing costs but doesn't sacrifice high availability for the more critical services