**Part 2. Cloud-native patterns**

This is where the patterns are. Now, if you’re expecting a *Gang of Four* style of patterns, I’m afraid you might be disappointed, though I hope not. The *Design Patterns* book by Erich Gamma, John Vlissides, Ralph Johnson, and Richard Helm is fantastic and is arguably the book most responsible for raising awareness of reusable patterns in a whole era of software developers. But instead of taking that reference-book-like approach, my coverage of patterns is all in the context of the problems they’re designed to solve.

Virtually every chapter begins with a discussion of certain challenges, sometimes talking about design approaches that predate the cloud era, and then derives solutions—and those solutions are the patterns. It’s not accidental that the solutions I present are some of the ones you’ve undoubtedly heard about—sidecars and event-driven architectures, for example—but again, I hope that presenting them in this way will deepen your understanding and help you learn when and how to best apply them.

I start this part by introducing *event-driven design* in [chapter 4](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_016.html#ch04). Most of the patterns discussed in the context of cloud-native architectures implicitly assume a request/response approach at the core. Truth be told, that’s how most of us naturally think about our software. I want to plant the seed of event-driven thinking right at the onset so that you at least have it in the back of your mind as you consume the rest of the chapters. And then I close this part of the book in [chapter 12](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_024.html#ch12) with event-driven systems again, this time focusing on the important role that they play in enabling cloud-native data. It’s an admittedly all-too-brief coverage of cloud-native data, but I hope enough to complete the cloud-native picture for you, at least at a high level.

Between these bookends of [chapters 4](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_016.html#ch04) and [12](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_024.html#ch12), I cover a whole host of patterns. [Chapters 5](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_017.html#ch05), [6](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_018.html#ch06), and [7](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_019.html#ch07) focus on the cloud-native app, covering statelessness, configuration, and the application lifecycle. Starting with [chapter 8](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_020.html#ch08), I turn our focus more toward cloud-native interactions, first talking about service discovery and dynamic routing. Then in [chapters 9](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_021.html#ch09) and [10](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_022.html#ch10), I focus on the patterns you’ll apply to each side of an interaction after it’s established—the client and service sides. I’ll point out that [chapter 4](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_016.html#ch04) on event-driven design is also fundamentally about cloud-native interactions. The highly distributed architecture that’s constantly changing characterizes cloud-native software and poses new challenges for troubleshooting, and that’s what I cover in [chapter 11](https://learning.oreilly.com/library/view/cloud-native-patterns/9781617294297/kindle_split_023.html#ch11). It’s also interesting that the solutions in this chapter themselves exercise many of the patterns covered in the earlier chapters. And, as I mentioned, the book closes out by introducing the fundamental patterns of cloud-native data.