efforts platform engineering, they embody the mindset, skills, and approach necessary for solving the problem of ever-growing complexity (the fly) without swallowing ever-larger animals in the process.

To set the stage, in this chapter we'll cover:

- What we mean by platforms, and a few other important terms we'll use throughout the book
- How system complexity has gotten worse in the era of cloud computing and OSS, leaving us in an "over-general swamp" of exposed complexity
- How platform engineering manages this complexity and so frees us from the swamp

This chapter has a slight emphasis on infrastructure and developer tooling, but don't worry, this book isn't just for people working on infrastructure or developer platforms! We'll use systems common to all developers to provide a tangible illustration of the current state of affairs, but the underlying challenge of managing complexity is common to all kinds of internal platform development.

Defining "Platform" and Other Important Terms

Before we get started, let's define several important terms we'll be using throughout this book, so we all have the same frame of reference:

Platform

We use Evan Bottcher's definition from 2018, with a couple of terms updated. A platform is a foundation of self-service APIs, tools, services, knowledge, and support that are arranged as a compelling internal product. Autonomous application teams[1] can make use of the platform to deliver product features at a higher pace, with reduced coordination.

A corollary here is to ask: what, then, isn't a platform? Well, for the purposes of this book, a platform requires you to be doing platform engineering. So, a wiki page isn't a platform, because there's no engineering to be done. "The cloud" also is not a platform by itself; you can bring cloud products together to create an internal platform, but on its own the cloud is an overwhelming array of offerings that is too large to be seen as a coherent platform.

1 We'll sometimes call these teams your "users" or "customers," if it makes more sense in the context.