AI will help you operate better, but only if and where you have the data.
: Machine learning requires data to make decisions, and specifically for your own platforms, the only way it will help you is if you have the instrumentation that can produce the needed data. For many modern systems this might be easy enough, but for those of you with legacy applications and platforms, if you want to be able to apply machine learning to improve your operations you'll need to get on top of the telemetry.

Platforms will be built that curate the ecosystem of large language model (LLM) tooling for your company and ensure all of the components work together well.
: This is, if anything, an opportunity for platform engineers to expand their expertise and value to companies. To date, the world of tooling to support research and model generation has largely been the domain of either massive companies like Google or understaffed data engineering teams that haven't properly invested in the usability and operability of their platforms due to a lack of funding and/or attention. Many of the companies building out in-house AI infrastructure are hitting the same problems that we have always seen when dealing with large-scale distributed infrastructure: handling software and hardware failures, orchestrating the work efficiently, and debugging when things go wrong.

Wrapping Up

In this chapter, we described the four pillars of successful platform engineering and gave some examples of what they mean in practice. To summarize, platform engineering teams should:

1. Take a curated product approach, building paved paths and railways as they identify common customer needs.
2. Develop software-based abstractions. These may include services, APIs, libraries, OSS customizations, and metadata integrations.
3. Serve a broad base of application developers. The platform should support multiple tenants, and each tenant should have self-service capabilities, with appropriate guardrails to prevent the mistakes that come at scale.
4. Operate foundational offerings, meaning applying high operational discipline and providing support for the full platform.

If you're doing all of these things, then you're doing platform engineering. There are cases where it will make sense to do less, but beware of the long-term consequences. Without opinions about what's in scope, you'll fail to manage the overall complexity, and without a customer-centric product approach, you'll probably build the wrong systems. If you're not doing any software engineering, you're just doing operations with a high level of customer empathy. If you are building for only a small set of