

TECHNOLOGY

Privatized Open Source

by Evan J. Zimmerman



Open source software is everywhere, but maintenance is becoming an intractable challenge.

✓ **INSIGHT** | NOTE 24 Jun 2020

“In a future that includes competition from open source, we can expect that the eventual destiny of any software technology will be to either die or become part of the open infrastructure itself.”

-Eric S. Raymond, Cathedral and the Bazaar

Open Source’s Sustainability Problem

For years, the word in the open source community has been **sustainability**. The problem is this: open source is built on contributors, who are not paid by the community—after all, code commits are called contributions—and which enables anyone to use and edit the software without any remuneration. This has allowed open-source to be widely used, both by **scientists** and **big enterprises** alike, and today significant infrastructure on the Internet is **sometimes maintained by a single person working for free** — for example, OpenSSL, which secures two-thirds of all websites, has never been maintained by more than four people.

This results in distribution for open-source, but it also results in questions of maintenance and support, which has become a grand challenge. *The Cathedral and the Bazaar*, the spiritual bible of open source, posited that open source would be sustained by social rewards, or “egoboo.” But now it’s broken. Open-source is infrastructure and it’s straining under the weight of its own scale, most visibly with the Heartbleed bug. Why?

Infrastructure is primarily about maintenance, **which isn’t what most developers signed up for**, especially since it’s **financially so precarious**. There’s no passion in that—as Kurt Vonnegut observed, humans love to build but not maintain—so all that’s left is ecosystem dependence. The question is: how do you give entities the benefits of open source, such as a guarantee of codebase visibility and the creative power of the hivemind, while mitigating the challenges, like maintenance and support? In recent years, a solution has emerged: privatized open source.

The core idea is simple. Take an open source project, mix it with the computing paradigm of the day in what is essentially a wrapper, and sell it. The big successes include Redhat, Github, Databricks, Confluent, Hashicorp, and MongoDB, and more. Why did it take so long

for open source projects to be properly commercialized? It's quite likely that open source software needed SaaS to be privatized. Of the successful privatized open source companies, almost all are from the cloud and post-cloud era. The only notable pre-cloud exception, Redhat, is at heart a consulting company (in the same sense as IBM, which acquired it). This makes sense—prior to cloud, the value-add opportunities a company could provide beyond the software itself were extremely limited.

There are a ton of advantages to this model. And, amazingly, it seems like it is a new, repeatable business model.

Why Privatize Open Source?

With privatized open source, the market and technology have already been proven. In order for this model to work, you must choose an open source project that is already in wide use. That significantly reduces the technology risk. Plus, there are automatic stabilizers, as the open source community will support the project as it expands. In fact, it becomes a flywheel, as the more popular the project is the more interest developers will have in the community, thus reducing the acquisition cost and churn in the market.

The first leads are also already there. If your project is sufficiently popular on Github, you probably have enough blue-chip engineers publicly associated with the project that if you can convert even 10% of them you would have a legitimate unicorn. And it's open source, which means that potential customers don't have as much risk in being your early adopters—*they've already adopted you*—and the open source community gives some stability to the core technology.

Lastly, privatized open source mixes the benefits of privatization with the strengths of open source. By hiring core developers and paying them a salary, private open source companies both provide something to pay for and an incentive to maintain good projects. At the same time, by keeping open source open, private open source companies get the benefit of a community, which both reassures potential customers and provides a font of creativity for the development of the platform.

These moats appear to be quite strong. Yes, a successful enterprise company builds customer relationships, monopolizes key developers, and embeds its wrapper in enterprise stacks. These are moats for every enterprise business, and they won't go away anytime soon. But it is also extremely difficult to create a legitimate competitor to the leader within the same project. How will your competitors differentiate themselves from you, exactly? You've already built the deployment standard and the core technology is out there for everyone to use. The proof is in the pudding: there are very few meaningful competitors to companies that have successfully privatized an open source project with the notable exception of git, which has spawned Github and Gitlab. Even big companies don't seem able to use their muscle to kill privatized open source startups. Most notably, AWS, which is perhaps the strongest arm in tech today, didn't beat MongoDB but rather angered the open source community with its **attempted shiv**.

Instead, the biggest risk privatized open source faces is obsolescence. Unlike with closed source software, everyone can see the source code and anyone can clone or fork the repo. As a result, developers can find the shortfalls in existing projects and create a competing library. As an example, the biggest threat to Hortonworks was never another Hadoop competitor. It was a threat to Hadoop itself, which came in the form of Spark and its privatizer, Databricks. Those competitors, though, will also be open source, so a privatized open source company can always compete on the usurper's own roadmap to minimize the wedge and even adopt those projects themselves as part of a broader enterprise solution.

Indeed, the biggest fail in privatized open source, Docker, ironically demonstrates just how durable privatized open source software truly is. In truth, no one ever came up with a great competitor to Docker's version of containers. Instead, Docker the company was crushed by Kubernetes, a container orchestration system that competed with Docker Swarm and which, ultimately, consumed the primary enterprise value of the product because orchestration rather than containerization itself is the main service companies need. It wasn't enough that Docker faced the full force of Google. It wasn't enough that it faced a new technology. No, Docker had to face all of these headwinds while **bungling the opportunity itself through horrid mismanagement**. The case of Docker shows not how quickly privatized open source companies can fail but how many things need to go wrong for failure to occur.

What's Next

When starting a company, one of the most difficult parts of getting financing is creating a convincing signal of credibility for investors. Leading a popular open-source project is a verifiable way to jumpstart that process, so it seems likely that this will be a route for key employees running corporate open source projects to start companies, thus creating a new vein for mafia-style entrepreneurship networks. Just this month, Greylock led a round for Chronosphere, which is privatizing Uber's M3. A little while ago, they also co-led a round with Sequoia for Rockset, which is commercializing Facebook's RocksDB. There will be more.

Nonetheless, it feels like this model is still being systematized as a method of starting and operating businesses, like the lean startup method. We need this—yes, big tech companies now support infrastructure projects the Linux Core Infrastructure Initiative, but it hasn't been enough and **open-source bugs continue to proliferate**. To survive, projects need dedicated developers to roll up their sleeves, which requires payroll, and that's only possible if the projects are themselves self-sustaining. As experimentation continues, the pace of privatized open-source startup creation will increase and the quality of those startups will only go up. It will be an exciting time for the entire community and, for at least some kinds of projects, the market could be the salvation open source was looking for.



Evan J. Zimmerman [Follow](#)

Evan J. Zimmerman is an entrepreneur, investor, and writer. He is the founder of Jovono, a venture capital firm, and Chairman of the Clinton Health Access Initiative technology council, which advises the technology of global public health in dozens of partner countries.