PERCONA

WHITEPAPER

The High Cost of Vendor Lock-in

PERCONA

What is Vendor Lock-In?

Business often moves at a lightning pace. To meet internal and external expectations, there may be pressure to sign-up to a vendor that solves your immediate issues. You then become dependent upon that provider for proprietary services.

When you choose a proprietary software, database, or platform, you are often committed to solely using that technology going forward and are unable to move to an alternative vendor without the substantial financial expense or business disruption. This is known as vendor lock-in.

Vendor lock-in requires you to relinquish control and give up your ability to choose alternatives. You become closely tied to your vendor as you create business and IT processes around the tools and software they provide. You merge the vendor's technology into your business, often having to integrate new tools and processes, even if they are not an ideal fit. Switching to an alternative provider quickly becomes costly and complicated. No wonder it becomes so difficult to leave.

Many (if not most) technology vendors will actively try and lock you into a relationship. It is in their interest to make leaving as difficult and expensive as possible.

A vendors' solution to a problem is often to use more of their software, or scale-up, even if that software is not a good fit for your evolving business needs. A big risk posed by vendor lock-in is the cost it will bring down the road in terms of lost agility, as new technologies, which you are unable to access, enter the market and become popular.

Although prices often start out reasonably, you usually have no control over increases which might occur due to business growth, additional required features, or changing vendor pricing structures.

Being tied-in to proprietary database software brings its own restrictions, you often have no choice but to pay-up in order to keep your database running. You might have to settle for sub-standard features, availability, or performance, which would be improved with a different set-up.

The Benefits of Vendor Lock-In

Not all lock-in is 'bad'. If a vendor provides a genuinely innovative way of solving a specific problem, and there are no alternatives on the market, there will inevitably be an element of lock-in, as you benefit from something unavailable elsewhere.

Having a single vendor relationship can also be beneficial for the following reasons:

- A single contact for a variety of needs
- Clear support processes from a trusted partner
- Potential cost reductions (normally upfront) as the more you buy, the bigger the discount

Relying on a single technology (or just a few):

- Simplifies your environment
- Helps you build critical expertise, as it is easier to master fewer technologies
- Ensures all components work well together (in theory at least)
- Makes decision-making easier, as fewer options are available.

If you actively choose vendor lock-in, you should be aware of your choices and the trade-offs that occur as a result. It is still desirable to have an exit plan in place, just in case.

PERCONA

The Pitfalls of Vendor Lock-In

Most companies want to find a trusted advisor who will guide them to the best possible technology option for their business.

Most vendors want to be this trusted advisor, but only offer solutions that use *their* technology and *their* platforms.

Technology solutions vary drastically, as do the needs of individual businesses. There is rarely a one-size-fits-all solution. Every application and workload has a unique fingerprint, which requires you to build and utilize an optimum set of tools, software, etc. to get the best outcome.

When you rely solely on a single vendor your fate becomes tied to theirs, this makes you susceptible to:

- Price hikes
- Paying for bundled technology you don't need
- Being unable to adopt industry standards and technological advances if not advantageous to your vendor, or if your vendor does not implement them
- A roadmap you can't control where:
    - A feature you relied on could be dropped without warning
    - New features may require you to completely rewrite portions of your application

You also have to meet your vendor's terms. This might involve them arbitrarily changing licenses or support fees; or even how they count a CPU (one day its X, the next Y). Oracle, for example, has gained a negative reputation for arbitrary price hikes, high support fees, and ever-compounding ULA renewals.

The fundamental problem with vendor lock-in is that you're stuck. If you have sunk millions of dollars and engrained this software in your infrastructure, it simply becomes too costly to move. To extract yourself can take years of hard work, and potentially cost millions more.

Being stuck limits your options. Companies using dead or dying technologies (sometimes just versions), can stagnate. This impacts on the people who work for you, who stop learning new skills and fail to innovate.

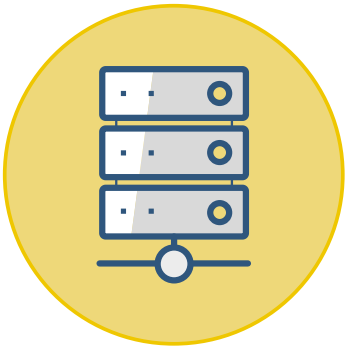
As the progression of your company's technical base stalls, your people-assets become progressively out-of-touch with current and emerging technologies. As a result you naturally follow a path which inhibits your organization from bringing in new technologies, as well as failing to bring in people who know about that technology. This becomes a spiral which stifles agility and can make an organization out of touch, and unable to respond to change.

Being stuck with your vendor means you miss out on transformational features or trends. Having portability enables you to decide for yourself if other technologies/features from other vendors are compelling enough to switch.

In the end, switching will always incur some level of cost (money, time, skills, etc), so you need to weigh the pros and cons when selecting something new.

PERCONA

History Repeating Itself



There has been a significant push-back against proprietary enterprise software in recent years, something that also occurred years ago when mainframes dominated the landscape.

At that time you had no choice but to buy from a small selection of companies, and to use their software. You were locked-in to companies such as IBM and you paid a high price for that privilege.

As there was little competition and few options, prices were extremely high. No matter the size of the application, or your need, you had to buy the whole package. You generally got your operating system, database, tools, and development environments, etc. from a single vendor.

Just as people are now trying to free themselves from modern-day vendor lock-in, over the last 30 years much time, expense, and resources went into extracting companies from mainframe.

The industry was eventually disrupted by smaller, agile, more open vendors as Digital, Sun, Hewlett Packard, and others, who started developing and bringing to market, “mid-range” systems.

These systems were cheaper, easier to run, and did not require as much “proprietary” locked-in software. You were still locked-in, but not to the entire ecosystem:

- You could bring your own database
- You could choose different vendor tools and services

This ability to bring your own software environment completely disrupted the mainframe industry. It led to a vibrant software ecosystem with a huge array of options; however:

- Operating systems and hardware were often still locked into one vendor
- If you needed more memory, new hard drives, or a new CPU, you had to pay high fees to your original equipment manufacturer
- Upgrade paths were often limited and expensive
- These big, expensive machines could run dozens of applications and often used shared resources

Then another change came as the mini-computer, especially the Linux movement, completely disrupted that mid-range business:

- Instead of hardware locked into a specific manufacturer, you had commoditized components
- You could run Linux on almost any platform
- Microsoft and Intel ‘unlocked’ the market, allowing hundreds of hardware manufacturers to build plug and play components
- Commodity hardware could replace mid-range systems en mass. You could start to consider having a server for each application

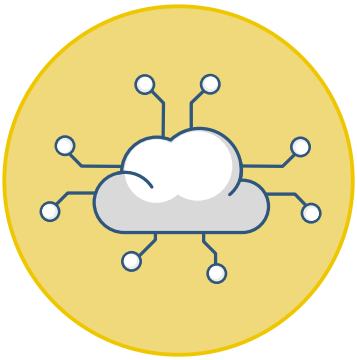
Commodity hardware grew as consolidation in the software space started to accelerate. Through many

acquisitions, Oracle was no longer just a “database” company as it offered enterprise resource planning (ERP), databases, tools, etc. Redhat was not just a Linux distro. Many software suites started to develop features/ benefits that worked ‘better’ with the owners’ other software.

This represented a seismic shift, as power moved from hardware manufacturers who controlled the whole stack, to software conglomerates who started consolidating and gaining power. The shift reached a high point when software giant Oracle themselves became a hardware company through their acquisition of Sun Microsystems.

So, originally we started out with heavy centralized control over everything — where you went to a single vendor for all your software and hardware needs. We then moved to central control over hardware and low-level operating systems, with a lot of choices around software vendors. Then we reached centralized control of software, and completely open hardware/operating systems.

Enter the Cloud, and Everything as a Service (XaaS)



Hardware has now turned into such a commodity that it rarely impacts companies. Instead, businesses are turning to XaaS. Vendors run the software, infrastructure, etc. for you. You may relinquish some overall control, but in exchange you can shift your focus back to your core business competencies as the vendor provides simplicity, cost savings, and (theoretically) less stress.

Once again, the push is to let just one provider manage your software, hardware, and services.

Ironically, this takes us back to the old mainframe days with the associated risks they presented:

- Lack of flexibility and vendor lock-in
- Arbitrary price hikes
- Prices so high they are unsustainable

Unsurprisingly, there is a back-lash. Over the last 10 years, an increasing number of enterprises have chosen open source solutions and are choosing open source software. Open source offers lower costs, more visibility, and a lower barrier for entry (as you can start using it for free).

The Case for Open Source

The database landscape has evolved rapidly in a short space of time. Open source database software and the open source community continues to innovate to meet user and enterprise needs.

Open source software is attractive for a number of reasons. For many companies, cost is the primary consideration; open source software is free to download and isn’t subject to subscription fees. Due to the open nature of the development model with its focus on open standards, vendor lock-in is highly improbable. Additionally, in many cases, choosing the right open source software allows you to access enterprise-level features, without the associated costs.

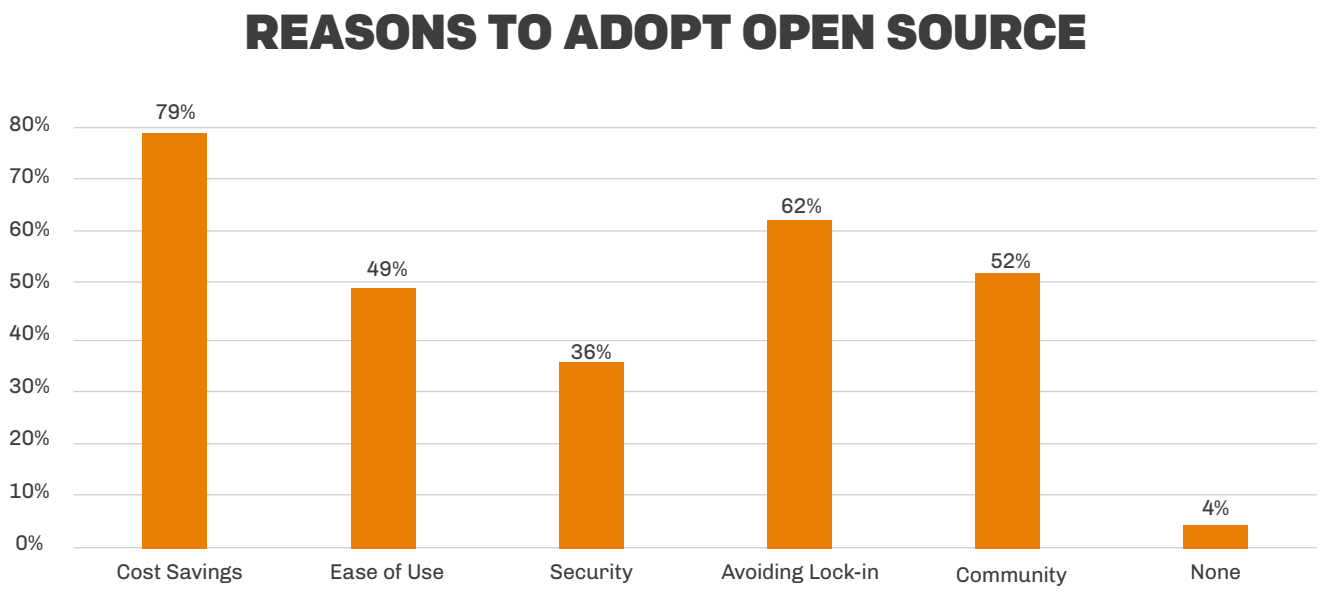
If you are hedging your bets on the future direction of your business technologies, open source software gives you usage flexibility and the freedom to change your software in the future, with less restrictions.

Percona recently conducted a survey of open source database users to find out how they deploy open source software databases.

As expected, **one of the strongest motivations to move to open source is cost savings**, with 79% of survey respondents citing it as a major factor. Rather than being locked into a proprietary vendor and trapped into paying high license fees, many companies discover they can get the same results, response, and reliability with open source solutions.

62% of the survey respondents use open source software to avoid vendor lock-in. This helps businesses remain agile and respond quickly to changing customer needs or their own development plans. Most respondents were also actively averse to adopting software with restrictive licensing. Only 8.5% were **very likely** to adopt "not open" software, compared to 68% being **very likely** to adopt "totally open" software.

The benefit of having a community scored highly (52%). Open source software has strong, passionate, online communities which can be a useful resource if you have problems or need advice. These communities also often set and enforce standards, which inhibits vendor lock-in by ensuring capabilities are portable and adaptable across the industry.



Through conversations with our customers and our research, it is clear that an increasing number of companies are adopting open source software, including open source databases, as part of their general business IT architecture.

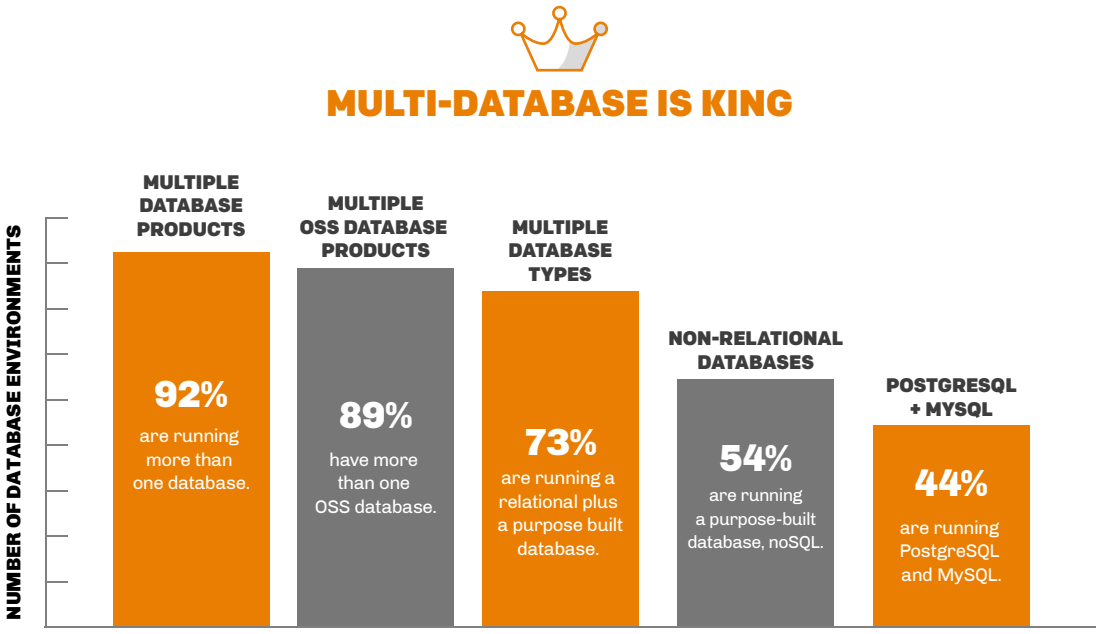
When it comes to database deployment models, businesses are often mixing and matching different types of technology: different databases on-premises and in the cloud, using new and emerging deployment strategies that include things like containers and Kubernetes. The database landscape has become increasingly complex and there is no one-size-fits-all solution.

Cloud Vendors — Swapping One Lock-in For Another?

The cloud has had a huge impact on all sectors of technology, transforming the way we store, manage, analyze, and retrive data. Vendors actively promote the cloud as the easiest way to deploy, operate, and scale databases. There are huge benefits in harnessing the power of the various cloud-based solutions available.

Many companies are looking at DBaaS options to save money, and reduce lock-in and outage risks. They believe their databases will be fully managed. Unfortunately, this is generally not the case. Many companies use multi-cloud and hybrid options as part of their strategy to avoid cloud vendor lock-in.

Our survey respondents are well-informed about using open source technology in the cloud. Interestingly, however, many respondents who champion cost-effectiveness, flexibility, and freedom from vendor lock-in, can find themselves tied to cloud vendors via a single solution, incurring large monthly costs.



As companies grow, our data shows that they are more likely to host their database infrastructure both on-premises and in the cloud. The larger the organization, the more complex the hosting environment.

To avoid being stuck with another vendor, just in a different environment, you need to look at best practices and ask searching questions on the impact to your business, and the availability of your data.

Why Percona is Different

We believe maintaining control of your data and applications is essential, and that you should have data portability if you decide a different database software is a better fit. It is crucial that you do not lose control of your ability to choose the right database for your business.

While we can (and will) help you solve issues with any open source database software you use, we also create and distribute our own versions of the most popular open source database software: Percona Server for MySQL and Percona Server for MongoDB.

PERCONA Server for MySQL

PERCONA Server for MongoDB

These free, open source database software products are completely compatible, drop-in replacements for the Community software versions. Our software includes features usually only available in paid-for enterprise versions.

We believe everyone should have access to those cutting-edge features that protect their businesses and enhance performance. Any improvements or fixes that we make are given back to the community and are available for MySQL and MongoDB to include in future releases.

We recently launched Percona Distribution for PostgreSQL. We took the best, most important, enterprise-level components from the open-source community, and designed and tested them to work together in a single source.

Installing Percona Distribution for PostgreSQL provides a complete package for PostgreSQL that meets the needs of both large and small companies.

We have also developed a range of complementary cutting-edge high availability, back-up, and monitoring software, to ensure your open source databases perform at the highest level.

There is no fee or commercial license requirement for downloading and using Percona software.

How can Percona help you avoid vendor lock-in?

- We support the most popular open source databases, on all the most popular platforms — we have no personal agenda when it comes to your choice of technology.
- We are technology agnostic — we help you pick the database, platform, and solution that best suits your needs, without bias.
- We don't believe you should be paying for features that you don't need or want — our software is 100% free and open source.
- We are against vendor lock-in — we work hard to ensure our software is compatible with upstream open source versions and includes free enterprise features. If you want to move, we give you an easy pathway back to the Community software version.
- Our dedicated consulting team has extensive experience assisting companies with complex database set-ups migrate from proprietary vendors to open source software — contact us to find out more.

Conclusion

Few companies actively chose vendor lock-in. It happens in increments, to make life easier in the short-term, to avoid making a tough choice, to keep the status quo. As time goes by it seems too difficult and too expensive to make a change.

For many companies already deeply embedded with a vendor, extraction is a lengthy and expensive process. You need to think carefully whether the (relatively) short-term pain of migration and the costs this will incur, is better than staying with your existing vendor. If prices are increasing and your technology is stagnating, it might well be worth the effort in the long-term.

If you are considering appointing a new vendor in the near future, it is essential that you establish mutual agreement around costs, control, and data portability. Having a clear idea of what you need and want from your technology vendor is crucial to avoiding the pitfalls of vendor lock-in.

Contact Us

For more information, contact us at +1-888-316-9775 (USA), +44 203 608 6727 (Europe) or have us reach out to you.