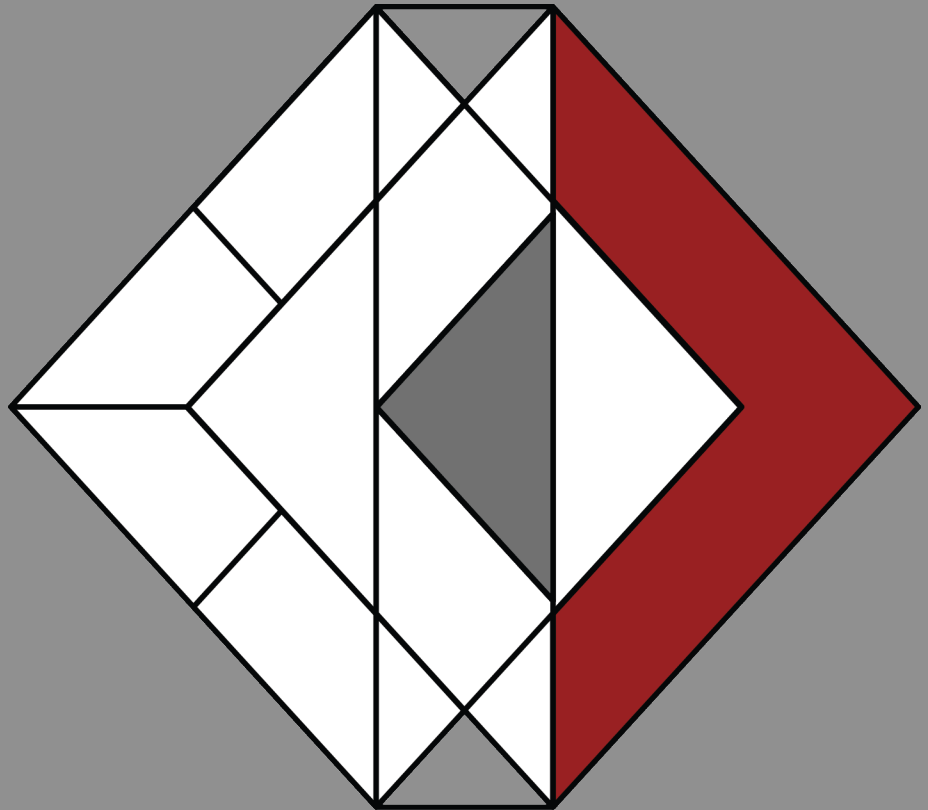


# How to avoid Vendor Lock-in?

Sriram Subramanian

Vendor Lock-in commonly refers to getting limited to a specific product or services vendor due to technology, business, legal, compliance or any other reason, thereby limiting the flexibility to choose a different provider or vendor. This position paper provides prescriptive ways to avoid vendor lock-in in an enterprise IT environment. Or does it?



CloudDon

<https://www.clouddon.com>



## How to avoid Vendor Lock-in?

*Prescriptive ways to avoid vendor lock-in<sup>1</sup>*

---

<sup>1</sup> Currently it is impossible to achieve a true lock-in less nirvana in an enterprise IT environment given how many ways lock-in [Appendix] could happen. This paper employs *reductio ad absurdum* to explain this inevitability and is inspired by (Upper, 1974). We also expect open source alternatives to avoid lock-in to be developed eventually as abstractions at levels higher than underlying infrastructure are getting developed.

## Appendix

### Types of Lock-in

This section enumerates most common types of lock-ins that could happen in an enterprise IT environment. While this list covers the types of lock-ins that impact the most, it is by no means complete.

#### *Architecture Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of software/ application architecture (monolithic, distributed, client-server, mobile, cloud-native, micro-services, type of availability, etc.).

#### *Business Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the line of business an enterprise is in. For example, brick-and-mortar retailers have known to have apprehensions on having dependency on an online retailer for their infrastructure needs.

#### *Database Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of databases used (for example, relational or non-relational databases, schemas, database sharding, database licenses, hosted databases, as-a-service databases, etc.).

#### *Data Gravity Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the location, amount and type of data consumed being stored.

#### *Geo Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to geographical location, geo-politics, proximity etc. This could limit the choice of service providers, data storage policies, data movement etc. This could also limit the end user experience that a solution offers.

#### *Infrastructure Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of underlying infrastructure (server types, manufacturer, storage backends, network routers, switches, virtualization platforms, physical firewalls, type of isolation: hosted, dedicated, co-located, or managed, etc.).



### *Interface Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of APIs (SOAP or REST, end points, URLs, etc.).

### *Legal Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the international, national, local, business, compliance or any other governing laws, IP, compete, trademarks, or any other legal restrictions, etc.

### *Process Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of process and practices followed (SDLC, security practices, ITSM, ITIL, communication, etc.).

### *Runtime Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of application runtimes (.NET, Java, container run times, etc.). Though technically a subset of software lock-in, given its importance, and the impact it can have, it is listed as a separate lock-in. It is also to be noted that runtimes like Java are not impacted by limitations like operating systems.

### *Service Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of external services an enterprise typically depends on (outsourced software development, data center management, reseller services, support services, hardware support, network connectivity, utilities etc.).

### *Service Provider Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of the service providers an enterprise has relationship with and the nature of such relationships. This includes, but not limited to, preferred hardware vendor, regional reseller, ISV partner, managed service provider etc., cloud service provider, etc.

### *Software Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of software stack (operating systems, platforms, orchestration engines, programming languages, compiles, frameworks, operating systems, IDEs, SDKs, etc.).

### *Toolset Lock-in*

This type of lock-in commonly refers to limitations in flexibility and choices due to the choice of the toolset employed (tools used for development tools, testing, CI/CD, bug tracking, incident management, monitoring, logging, communication, ITSM, etc.).

## Bibliography

*Amazon Web Services (AWS) - Cloud Computing Services.* (n.d.). Retrieved from Amazon Web Services: <https://aws.amazon.com/>

Bloom, G. (n.d.). *Why tech companies should beware of 'cloud handcuffs'.* Retrieved from <https://www.forbes.com/sites/forbestechcouncil/2017/07/25/why-tech-companies-should-beware-of-cloud-handcuffs/#3bf233192710>

Chan, M. (n.d.). *6 things you can do to avoid cloud vendor lock-in.* Retrieved from <https://www.thorntech.com/2017/09/avoidingcloudvendorlockin/>

*Cloud Price Index, 451 Research.* (n.d.). Retrieved from 451 Research: <https://451research.com/services/price-indexing-benchmarking/cloud-price-index>

Cockcroft, A. (2016, April 22). *Part 2: Who Doesn't Like Lock-In?* Retrieved from Battery Ventures: <https://www.battery.com/powered/doesnt-like-lock-in-part-2/>

Cockcroft, A. (2016, April 20). *Who Doesn't Like Lock-In?* Retrieved from Battery Ventures: <https://www.battery.com/powered/who-doesnt-like-lock-in/>

Cote, M. (n.d.). *Moving beyond "Lock-in" with mutli-cloud.* Retrieved from <https://content.pivotal.io/blog/moving-beyond-lock-in-with-multi-cloud>

*Data Gravity.* (n.d.). Retrieved from <https://datagravity.org>

Giorgio Nebuloni, E. F. (n.d.). *European Multicloud Infrastructure.* Retrieved from [https://www.idc.com/getdoc.jsp?containerId=IDC\\_P1400](https://www.idc.com/getdoc.jsp?containerId=IDC_P1400)

Golden, B. (n.d.). *Don't avoid cloud vendor lock-in. Embrace it.* Retrieved from <https://techbeacon.com/dont-avoid-cloud-vendor-lock-embrace-it>

*Google Cloud Computing, Hosting Services & APIs | Google Cloud Platform.* (n.d.). Retrieved from Google Cloud Platform: <https://cloud.google.com/>

Justice Opara-Matins, R. S. (n.d.). *Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective.* Retrieved from <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-016-0054-z>

Lydia Leong, M. D. (n.d.). *Addressing Lock-in concerns with public cloud infrastructure as a service.* Retrieved from <https://www.gartner.com/doc/3712817/addressing-lockin-concerns-public-cloud>

## How to avoid Vendor Lock-in?



McLuckie, C. (n.d.). *Four ways to avoid vendor lock-in when moving to public cloud*. Retrieved from Forbes: <https://www.forbes.com/sites/forbestechcouncil/2017/12/14/four-ways-to-avoid-vendor-lock-in-when-moving-to-the-public-cloud/#71e9ec1161f9>

*Microsoft Azure Cloud Computing Platform & Services*. (n.d.). Retrieved from Microsoft Azure: <https://azure.microsoft.com/en-us/>

*Microsoft Hosting and Cloud Study 2017*. (n.d.). Retrieved from Microsoft: <http://partner-11.microsoft.com/hosting-cloud-research-report-2017.html>

*Open source software for creating private and public clouds*. (n.d.). Retrieved from OpenStack: <https://www.openstack.org/>

*Production-Grade Container Orchestration*. (n.d.). Retrieved from Kubernetes: <https://kubernetes.io/>

Stewart, N. (n.d.). *How to avoid vendor lock-in with multi-cloud deployments*. Retrieved from <http://superuser.openstack.org/articles/avoid-vendor-lock-multi-cloud-deployments>

Subramanian, S. (n.d.). *Comparative Analysis of OpenStack Distributions*. Retrieved from CloudDon: <https://clouddon.com/download/openstack-distributions-a-comparative-analysis/>

Upper, D. (1974). The unsuccessful self-treatment of a case of "writer's block". *Journal of Applied Behavior Analysis*, 3 (Fall).

*What is a Container*. (n.d.). Retrieved from Docker: <https://www.docker.com/what-container>