

# Is Open Source a True Technical Requirement?

What does it really mean to have “must be open source” as a requirement? It’s actually both a requirement and a constraint if you think about it. More than that, is it really a true “requirement”?

Let me lead with the fact that I advocate strongly for the use of open source platforms for a number of reasons. This is how I believe we should look beyond just making a blind choice only on the basis that something is open source.

## **Honing in on Real Requirements when Choosing Open Source**

OpenStack seems like a good place to start. Imagine being handed a set of requirements which begins with this:

R1 - Solution must use OpenStack because it is open source

One of the most commonly questioned requirements that comes up in a design proposal is “environment must meet N+1 resiliency”. We have to question it because the real requirement is buried further inside that statement. N+1 resiliency is a technical architecture and physical design decision that services a real requirement around resiliency and workload protection in the event of partial cluster failure.

The same holds true when we see someone say that the solution “Must use open source” or target a specific platform because it is open source. Let’s narrow down the real underlying reasons that allow us to get closer to what’s actually happening. This is what that process looks like.

We must use OpenStack. Why? OpenStack is open source, which is good. Yes, it is good, but why? Open source means a large collaborative development and support community. Great! What does that mean to us as a consumer of open source? OpenStack is being designed to meet the broad needs of many customers and personas. Again, this is great, but what does it really mean? Open development communities can increase the velocity of features and capabilities. How does this occur? The wide group of OpenStack consumers include a lot of contributors back into the ecosystem. Bingo!

That narrows it down to a more specific requirement which is that we must use OpenStack because it has a strong development velocity with strong commitments to growing the feature set. That requirement still should be open for debate, but the way that we solve that is to list the associated risks:

RI1 - OpenStack ecosystem may not maintain the velocity of growth and feature updates

Is it really a risk? Yes. Not much more of a risk than depending on a small to medium sized vendor who may go out of business one day. The point of the exercise here is that we have to identify the real requirement, identify any associated risks, measure any decisions against existing constraints, and then, and only then, do we truly understand that the open source requirement is valid.

It is entirely possible that open source is listed as a requirement because it is a personal decision on behalf of the organization's management team to use open source as a moral choice. It's really something that comes down to a near-religious decision if that is the case. This is the reason that I like to bring the questions forward that help us find the logical and physical requirements that are being satisfied by the use of open source.

When it all comes down to it, sometimes we just say "because I like it". Hey, it's a reason, but not a technical one ☐