Homogeneous vs. Heterogeneous Clouds: Pros, Cons, and Differences

One of the many, many splits within the cloud camp is between homogeneous and heterogeneous clouds.

What is a homogeneous cloud?

Simply put, a homogeneous cloud is one where the entire software stack, from the hypervisor (or remote cloud provider), through various intermediate management layers, all the way to the end-user portal, is provided by one vendor.

What is a heterogeneous cloud?

A heterogeneous cloud, on the other hand, integrates components by many different vendors, either at different levels (a management tool from one vendor driving a hypervisor
from another) or even at the same level (multiple different hypervisors, all driven by the same management tool).

**How to choose between a homogeneous and heterogeneous cloud?**

The argument for homogeneous environments is that because everything comes pre-integrated they are easier to set up, and if something goes wrong there is only one responsible party – “one neck to wring”, as the saying has it. On the other hand, by giving so much power to one vendor, users place themselves at the mercy of that vendor’s commercial and technical strategy. In farming, this is known as a *monoculture*, that is, only a single crop is grown. Superficially, this is an attractive idea, as farmers can specialize and take advantage of economies of scale. The problem is that any disease, frost, drought, or other event that affects that crop can wipe out the entire harvest. To mitigate that risk, farmers try to sow several different types of crop, as insurance against losing the entire harvest.

The same arguments apply in IT. The advantage of an IT monoculture is that everyone can specialize in that one vendor’s tools and it is easy for admins to cover for each other. The downsides are a bit different: on the technical side, features will be available when – and if – the vendor chooses to develop them. The real pain often comes on the commercial side, because once users are “locked in” to a single-vendor strategy, they have no recourse if that vendor decided to change its pricing structure in a way that causes costs to increase.

Heterogeneous architectures attempt to bypass this lock-in effect by introducing components from many different vendors and allocating their use according to a common set of strategies. At some point, however, a single management component will need to be introduced. Defenders of homogeneous
approaches will counter charges of lock-in by pointing out that this convergence on a single management layer just moves the lock-in further up the stack, but still leaves users at the mercy of the provider of that one component.

The false equivalence between platform lock-in and supposed management lock-in is a neat rhetorical trick, but does not really hold up. Management vendors need to keep up with the development pace of the managed platforms, or risk falling behind the competition from other heterogeneous management vendors. Any attempt at predatory business practices will be nipped in the bud for the same reason.

Meanwhile, users can migrate away from management suites more easily than they can change hypervisors or cloud providers. The reason is that a platform change is almost guaranteed to cause disruption unless it is very carefully managed, while replacing a management platform, even at very short warning, will certainly be painful for IT and cause delays in delivery of new requests, but will not affect workloads that are already running on the underlying platforms.

BMC, not having a dog in the hypervisor fight, has been heterogeneous from day one, so we have been following VMware’s recent moves in this direction with interest, as have many others, notably James Staten of Forrester Research. The value in this rapidly evolving field is moving from the hypervisor into the management layer and the processes that it implements. In the next post on this series I will discuss that move further, and what it means for enterprises which are considering a move to cloud or have already begun one.

Last Updated: 11/1/2016