

## Demystifying SaaS vs. Cloud

I was asked to participate on a panel a few months ago on the topic of SaaS versus the Cloud and explore the question of whether SaaS would ultimately be “subsumed” by the cloud. To be honest, I had to study up for this one because I didn’t even realize it was a question. What I found was this was much more a marketing issue than a technical one.

The technical distinction in my mind is clear: cloud delivers computing as a utility, SaaS delivers an application (such as CRM) as a utility. (A much more fun topic is the difference between grid computing and cloud computing but there are a plethora of posts dedicated to that one!) A SaaS application can be delivered in a range of models from the vendor’s own datacenter, to a third-party, “hosting” vendor, to a true cloud computing environment (ala AWS) which takes advantage of the latest technologies such as virtualization to maximize resource utilization (again the distinctions and pros/cons of each of these models has been well covered so I won’t go into them here).

However it is provisioned, SaaS (application delivery model) is quite distinct from Cloud (compute delivery model). So why all the confusion? As I said, I think it’s largely marketing. Cloud has simply become a very handy, very trendy way of describing all things that occur “outside the firewall” as it were. If it’s not happening on premise, it’s happening in the cloud – whether you’re talking apps or compute.

There is however one HUGELY important distinction. Just because an app is hosted in the cloud does NOT necessarily make it SaaS. Why is this important? As a consumer, if the application is not truly multi-tenant, you’re not getting the great benefits of SaaS, you are getting an ASP delivery model. Someone is simply running your copy of their application somewhere in a datacenter. Even if it’s “virtualized.” That’s not multi-tenant. That’s not SaaS.

But, I’ve heard it argued, what does it matter to the end user? Why do they care how the functionality is being delivered? True, the typical end user probably does not care. However, end users generally aren’t making the purchasing decisions and paying the bills or responsible for the information

technology strategy for the entire enterprise. Those folks tend to care a lot more.

There is a reason why SaaS eclipsed ASP for application delivery. It is hands down a superior delivery model from cost to technology to scalability. Multi-tenancy is what makes all the benefits of SaaS possible: rapid time to deployment/value, faster innovation cycles, infinite scalability etc. And it has a dramatically reduced cost structure because it leverages one platform. Vendors who deliver their application via SaaS have a competitive advantage over those who do not.

There are some simple tests you can do to see if your vendor is using SaaS. For one, sign on to their website and start using their app immediately. Doing something like provisioning a trial account, which is a trivial matter with a true SaaS application, becomes a scaling and maintenance nightmare for ASP vendors – even using new technology such as virtualization. There are vendors who have had to withdraw their “free trial” because it simply cost them too much to actually provide it. Another simple test is to see if you can buy and provision your account without speaking to the vendor’s sales personnel. Again, trivial for SaaS vendors, not so much for ASP vendors.

If you’re an end customer, I encourage you to ask questions when selecting your “SaaS” applications. Go beyond “we’re in the cloud” to understand exactly how your vendor delivers its application. Presumably, you’ve come to the cloud because you want to exploit the economics and advantages of utility computing and applications. Caveat Emptor – if it’s not multi-tenant, it’s not SaaS.

Reprinted with Permission from Boomi, Inc.  
The Boomi Blog  
Posted by Bob Moul on March 16, 2009