CIO

Print Article Close Window

From: www.cio.com

Cloud Computing: Advice for Application Control Freaks

– David Taber, CIO

March 29, 2011

In the good old days that weren't so good, we suffered from DLL hell: the need to find and certify libraries that we didn't write but did depend on. Cloud computing presents an analogous challenge with services we want to use, but don't really control. You might not see it the short run, but if you plan to have clouds applications operational over years, this can present a very real issue.

What strategies can be used to cope with the inevitable problem of cloud control?

Private Clouds

In private cloud deployments, your organization has paid vendors for services or paid your own developers to create them. Consequently, standard contractual and configuration management tactics will work pretty well. So the basic function of the service — and the "API contract" — can be assured over time.

But that doesn't mean the service will really be useful: the security controls and user access privileges must be maintained, even as the services evolve and take on new user communities. For example, the CRM's sales cloud needs to be able to interact with the accounting system's commissions cloud, so that reps can plan for their boat payments and sales managers can design next quarter's sales contest. Since commissions are highly privileged compensation data, the accounting cloud will have careful controls on who can see what. As the accounting cloud is expanded by your team over time, its security policies may be modified in ways that are exactly correct within the domain of its cloud. But the ramifications of those policies on the CRM cloud can mean that users won't be able see what they're supposed to any more.

These security and access issues were probably handled when the clouds were initially integrated, but there needs to be an ongoing mechanism for accommodating the evolution that inevitably occurs across all the cloud services you depend upon. The classic security review / configuration control board needs to be extended up into the cloud.

Of course, there are tools and infrastructure that promise to make SOA governance a snap. But the reality I see with clients is that the solution is more about people, policies, and information sharing than it is about buying a product. Eighty percent of the problem is going to be solved with a well-cultivated wiki with people who know how to write and are empowered to manage the cross-cloud issues. Since these will be required as part of deploying an SOA governance product in any case, why not start with the basics and see how far you can get.

Slideshow: What is Cloud Computing?

Public Clouds

Private clouds give you at least the possibility of complete control. But that control comes at significant ongoing costs in both of operations and labor. The whole point of a public cloud is leveraging best-of-breed reliability, scalability, and operations infrastructure that you could not afford to replicate.

In contrast, leveraging public clouds means lower costs and simpler staffing...but you have to cede some control. The vendors you work with may be very accommodating, but in the end they must manage their service in the interests of their business. For example, you cannot really control the frequency and timing of bug fixes and feature releases that may affect both features and access control details. (True, Salesforce and some other vendors let you delay the updates to a time of your choosing, but you cannot stay with a "down rev" configuration for as long as you like.)

Since you don't have explicit control, the strategy here is preparation and advance testing. The best cloud vendors will have release notes and pre-release versions of their new functionality a few weeks in advance of the cutover. Your teams must analyze the impact of the service changes and do testing of the changed cloud and any clouds to which it is connected. Unfortunately, this pre-release testing can't cover everything, as your sandbox environment can be only so comprehensive and realistic. For significant cloud configurations, this means weekend regression testing and debug sessions immediately after the version cutover (just like it does for on-premises system upgrades). The good news is that for even the most complex cloud configurations, the need for these weekend parties will be limited to a few times a year.

If you are leveraging a public cloud service that you aren't paying for, there's even less leverage. For example, a geolocation mash-up may be way cool...but what happens if that free service is discontinued? Your team must identify second-source alternatives and understand the costs of making a transition if the cloud provider goes out of business or puts the free service behind a big pay-wall. Ideally, there will be several alternatives with varying features and transition costs. If you're lucky, a suitable alternative cloud service will have an open source code base. If so, you could contribute to an organization (such as a university or industry association) willing to host that service well into the future. Worse comes to worst, you could deploy that service in your own private cloud, to be self-sufficient.

Of course, the tradeoff for that full self-sufficiency and control is the cost of running your own cloud. There's no right answer for everyone: the key is making conscious choices based on both cost and business impact.

David Taber is the author of the new Prentice Hall book, "Salesforce.com Secrets of Success" and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline.

© 2010 CXO Media Inc.