Governance: Service Catalogs and the Cloud

Lori MacVittie, 2009-02-07

Can the inherent abstraction of virtualization succeed where SOA did not?

My first read through a post on the Cloud Front Office led me to scoff disdainfully at the re-emergence of a concept central to a successful SOA implementation: the service catalog. Oh, we called it "registry" and then "registry/repository (reg/rep)" and finally "governance" but the concept behind it was exactly the same. Take a gander at the description of a cloud service catalog apparently growing out of discussions that began at Structure 09:

> *Last week I attended Structure 09, one of the better cloud computing conferences. Everyone I could hear the echo of the need for a service catalog; often it was implicit.*
>
> *For example, everyone spoke about the need for standardization and multi-tenancy, about the need to abstract computing resources from the application to enable moving workloads as needed. Which in my mind requires you to a) know what the workload is, b) know the SLA of that workload, c) have well defined underpinning technical services that compose that application service, and therefore d) Have a SERVICE CATALOG to track all of that!*

Now, interestingly enough SOA governance utilized standards-based metadata like WSDL and WSIL to describe services. These descriptions included the endpoint (the physical location) of the service as well as the operations available and how to invoke them. In a full SOA implementation, the registry was at the heart of the architecture; if you wanted to invoke a service you queried the registry (usually via UDDI), retrieved the description (WSDL, WSIL), and then gathered the information necessary to invoke the service. This abstraction allowed services, ostensibly, to ***move*** and ***change*** over time without, allegedly, negatively impacting the client (consumers).

So while I was ready to excoriate this idea at first, a quiet hour trying to get a restless toddler back to sleep in the dark o'night led me to reconsider what appears to have been a rather hasty judgment. Might not this notion of a service catalog be more apposite than I first believed?

Absolutely.

ABSTRACTION AND MANAGEMENT (GOVERNANCE)

Let us assume that we can, in some standard way, describe a virtual machine such as is often the foundational building block of an on-demand data center. Perhaps we could even assume that OVF might suffice for this purpose. Imagine then, if you will, that these descriptors are stored in a central repository. A repository that is used not necessarily for

developers but for ***administrators*** attempting to automate the provisioning and management processes that are the basis for so many benefits of on-demand computing. If these descriptors, this ***metadata***, contains the appropriate ***management*** endpoints rather than just the endpoints of the actual service (application), it would then be possible to easily automate the launch and tear-down of such virtual machines.

Furthermore, the inherent chaos that could be caused by virtual machine movement within an on-demand infrastructure could be soothed. When a virtual machine (service) was launched, the descriptor could be updated to include its new physical location. In fact, assuming such services were launched in multiple locations (dare I suggest even across multiple clouds, as might be the case with intercloud) the descriptor could be updated to include *all* locations, perhaps even prioritized.

One of the interesting things virtualization brings to the table that SOA did not is the ability to abstract ***management*** of services. While WSDL abstracted a service and its operations away from the underlying implementation, that underlying implementation was still very much relevant in that managing the service required an understanding of the platform (.NET, Java EE, etc...) on which the service was deployed. You could not use WSDL or WSIL, for example, to actually manage the service - that was the ***raison d'être*** for SOA management solutions.

But virtualization solutions - likely because of many cloud implementations heavy reliance - provide management endpoints (APIs) through which virtual machine instances can be managed. VMs can be started and stopped and monitored through these endpoints (APIs) and the APIs are almost universally ***web-based***, meaning the endpoint is the equivalent of a URI. That fits nicely into a service catalog and could easily be automated by any number of existing solutions (BPM systems come to mind immediately) or incorporated into custom solutions with relative ease.

IS THIS THE BEGINNING OF A CNS (Cloud Naming Service)?

When you start looking at the usefulness of a service catalog it immediately appears that this is more than just an internal governance solution. By stepping back and looking at the bigger picture and the potential implementation of cloudbursting and intercloud solutions in the future, you begin to suspect that perhaps the service catalog and DNS need to merge, somehow. Perhaps this is where global load balancing can really shine. On the innertubes today if we want to find a site we use DNS to look up the IP address. In a cloudbursting or intercloud scenario we can't be sure exactly where a service may be residing. Using a system similar to DNS we might (a) lookup the service, (b) retrieve its current location and management endpoints, and (c) manage/invoke at will.

Perhaps this is an extension to DNS? Perhaps it's something completely different. While many of the core concepts of cloud computing and virtualization are not new at all and, in fact, appear to be heavily borrowed from programmatic models of the past (distributed systems, SOA, object-oriented programming) some of the solutions that will be created to help manage cloud and virtualization ***will*** be new, because we have a completely new set of problems and issues to address caused by the convergence of a variety of programmatic and architectural models.

The use of SOA governance solutions never truly took the world by storm, and that may be in part that the metadata it carried wasn't "meta" enough given the level of abstraction used by SOA. Virtualization and cloud computing take that abstraction far enough to be useful both in invocation and management. SOA, too, was hampered by the fact that automation of processes - while nice - was not a necessary piece of the value equation. For cloud computing (on-demand) automation is one of ***the*** key variables in the benefit equation, making abstraction of management a necessity.

The more I think about this concept, the more I am liking where it may be going. Service catalogs may, in fact, be one of the "movers" of cloud in the enterprise, as it would offer a firm foundation on which automation and management solutions could be more easily built and deployed.

- Intercloud: The Evolution of Global Application Delivery
- Emergent Consensus: Private Cloud Needs a Front Office Part 1
- Building an elastic environment requires elastic infrastructure
- Building a Cloudbursting Capable Infrastructure

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com