Taking Enterprise Application Integration Into The Cloud

By: Bob Moul, General Manager, Dell Boomi

If you’re like a lot of CIOs, you’ve been in this business a long time and have pretty much seen it all. From mainframe to mid-range to client-server to ASP and everything in between. It’s hard not to look at SaaS and cloud computing as just another fad – another name for things you’ve been doing for years. But as was the case with the advent of the internet, there are some fundamental transformations occurring in cloud computing that would be detrimental to ignore. The emergence of cloud-based integration platforms is one of those shifts.

The reputation that integration has garnered over the years as an ugly, unwieldy problem is almost entirely due to the way integration software was previously built and delivered as opposed to the actual integration projects themselves. Ironically, the integration industry has created much of its own complexity. Fortunately, the SaaS model has led to a fundamentally different way of thinking about application design and delivery – one that makes possible a new and vastly more efficient model for delivering integration – even if you are not engaged with cloud computing or using SaaS applications.

Genesis of Complexity

Integration by definition is a distributed function. Applications run on multiple systems in multiple locations using multiple databases and files. Data are distributed. Infrastructure is distributed. Networks are distributed. Users are distributed. Therefore, integration processes also need to be distributed – executing in any number of places where integration of systems is required.

In the traditional model of developing software, the approach was to build a standalone application (or suite) and then commercialize it by selling copies of that application to as many customers as possible. In today’s cloud parlance, this is called a multi-instance, single tenant application – multiple copies of the same application with no shared resources.

In this model, as the number of copies and versions of the application increase, the support costs increase dramatically while the innovation cycles come to a near standstill. Vendors spend more and more of their time and resources maintaining existing application code as opposed to delivering new features and functionality and yet continue to charge 20% or more in annual maintenance fees. At the same time, significant burden is placed on the end customers who have to maintain their copy of the application with new releases, updates and code patches. It is this maintenance challenge and fear of breaking hard-coded integrations that often keeps customers from upgrading. Enterprises are forced to spend 80% of their IT budget on maintaining current applications systems, and only 20% on new innovation.

Implications for Integration

When the multi-instance, single tenant model is applied to the world of data and application integration it inherently creates a number of complexities, such as security and governance challenges, that have plagued the industry for decades. Here’s why. The functions involved with integration include designing and building the integration process, deploying the integration, executing the integration, and then ongoing monitoring and management. While the execution of the integration needs to be distributed at multiple points throughout the enterprise, ideally all other functions, e.g. administrative functions, should be centralized.

However, vendors of integration products, using the traditional model of developing software, built all administrative and execution functionality into one product. Customers would then purchase a copy of an integration product (or even multiple products) for every instance where an integration process needed to run within the enterprise. It is typical for a large enterprise to be running 20 copies or more of these integration stacks in order to meet their business requirements.

The impact of the above scenario was that not only was the necessary runtime distributed across the enterprise, but the functionality to build, deploy and manage the integration processes was also replicated across the enterprise. Having multiple instances of what would ideally be centralized administrative and management functions created a maintenance nightmare and weakened governance and controls. Nearly every major integration challenge – including version control, data visibility, auditability, policy enforcement and governance are the result of this legacy style of software development and deployment.
A New Approach

The SaaS model has led to a fundamentally different way of thinking about application design and delivery. In the SaaS model, considerable focus is placed on designing an application whereby there is only one copy but that any number of customers can use (and customize for their unique needs). In cloud parlance this is referred to as single-instance, multi-tenant – one shared application with shared resources for multiple customers. Multi-tenancy makes possible all the great values of SaaS such as rapid deployment times, faster innovation cycles and ultra-low cost structure.

It also dramatically shifts the maintenance / development ratio as there is only one copy of the application to maintain (customizations of each tenant are maintained as extensions and automatically re-applied when the application is updated). Now when a change is made to the application and deployed — all customers are updated in lock step with little to no effort required by customer personnel. There is one single code base across all customers to be maintained freeing up significant developer time to focus on new features and functions. The resulting productivity boost has given rise to what some have called the “continuous innovation cycle” of SaaS.

An Old Dog Learns a New Trick

When SaaS concepts are applied to the domain of data and application integration, an entirely new and vastly more efficient delivery model is made possible – even if you are not currently using SaaS applications. Cloud-based integration platforms allow you to design, build, monitor, and manage integrations centrally (from the cloud) yet deploy just the runtime to where the integration needs to occur – either in the cloud or on premise.

Separating the runtime while unifying the administrative and management functions eliminates unneeded complexity and dramatically increases the quality of:

1. Collaboration
2. Component Re-use
3. Data visibility and control
4. Policy standardization and enforcement
5. Configuration management
6. Monitoring
7. Auditing/validation
8. Reporting

And because cloud integration platforms are built with SaaS technology and concepts, they deliver the same benefits as SaaS applications themselves including:

1. Significantly reduced cost structure
2. No integration products to install and maintain
3. Automatic upgrades — no “rev lock” issues
4. Usage based pricing — start small & expand over time
5. Global connectivity and access

Positioned for the Future

Using a cloud-based integration platform to address your current on premise integration requirements has the added benefit of positioning the enterprise to support the adoption of cloud and SaaS-based applications in the future. SaaS applications have introduced a new set of integration complexities which legacy integration products were not built to support.

Complexities such as enabling cloud to on-premise work flows without the need for special firewall configuration (and thereby introducing security vulnerabilities) and handling the dynamic nature of today’s complex APIs. With a cloud integration platform, applications and data sources outside your firewalls can easily be added to your portfolio using the same distributed runtime, centralized management framework.

Conclusion

SaaS technology is changing the way applications are designed, developed and delivered. When these concepts are applied to the world of application and data integration, an entirely new and vastly more efficient model for delivering integration is made possible – one that greatly simplifies decades old challenges such as configuration management and governance. We have the opportunity with cloud computing to not perpetuate the complexity of the past with hardcoded, brittle, essentially dead integration. We have the opportunity with cloud-based integration platforms to implement living, breathing, scalable interfaces that greatly enhance the agility of the enterprise.

About Bob Moul

Bob joined Boomi in late 2005 as a member of the board of directors and was appointed president and CEO in August 2006. He has led the company’s successful transformation from a traditional software product company to a leading infrastructure provider in the SaaS and cloud computing industry.