

Busting Myths of On-Demand: Integration (August 2007)

Peter Coffee
Director, Platform Research
pcoffee@salesforce.com

Contents

Myth: It's hard to integrate with on-demand systems 1

Busted: On-demand integration is easier, faster, and less risky 1

Integration and salesforce.com 1

Obstacles only imagined 2

Integrate what? how? why? 2

What's being integrated?..... 2

How will integration be performed? 3

What is the goal of the integration effort? 5

What are the relevant measures of integration success? 5

Myth:
It's hard to integrate with on-demand systems

The same Web browser interface that gives users “anywhere, anytime” access to on-demand convenience may seem to be a barrier to strategic applications.

It may seem that the advantages of on-demand delivery are only fully realized in applications that are self-contained, without reference to other enterprise data or logic, or in applications that can tolerate low-bandwidth integration with other IT assets.

Busted:
On-demand integration is easier, faster, and less risky

On-demand platforms exhibit growing capability to provide, not merely hooks for integration, but a superior foundation for enterprise-class integrated systems.

For every integration scenario—custom coding, native connectors to specific software products, middleware layers facilitating multi-way integration, coupling with personal productivity applications, or user interface integrations of the type often dubbed the “mash-up”—there are options available that let enterprise IT builders choose their own balance between flexibility, capability, and speed of solution deployment.

Integration and salesforce.com

Salesforce.com’s proven API is already handling more than half of a transaction volume whose peaks exceed 100 million transactions per day. In addition to that API, developer tools and partner integration products enable every mode of integration of likely interest to the enterprise, as well as enabling affordable and low-risk solutions for small- and medium-scale customers.

On-premise enterprise IT portfolios are complemented, not compromised, by the introduction of on-demand innovation

Read on for more

Obstacles only imagined

On-demand systems offer clear and compelling efficiencies compared to on-premise or hosted single-tenant systems. Application domains including customer relationship management provide abundant examples. Reduction of on-premise infrastructure, and streamlining of the code base and the software stack maintenance workload via multi-tenant architecture, create significant financial and operational incentives.

Early stages of on-demand adoption are often driven by business units seeking to solve specific problems, adopting cost-effective on-demand offerings with their rapid deployment timelines in an uncoordinated fashion. Forrester senior analyst Liz Herbert has noted¹ the difference between the early years of on-demand technology and the current trend toward enterprise incorporation of on-demand elements into a larger IT portfolio, saying, “In the early days of SaaS [software as a service], a lot of applications were standalone and running in silos... People just weren’t focused on the challenge of integration... Now we’re seeing SaaS used more broadly.”

A report from Saugatuck Technology describes² the resulting emergence of a growing enterprise IT role in setting the on-demand agenda: “While SaaS initially shifts the customer conversation from technology issues to business issues, the conversation changes when the need arises to integrate those SaaS applications with other enterprise applications. At this point the IT department usually becomes involved in assessing and managing how this will occur.”

At first glance, it may seem that on-demand advantages are only fully realized in applications that are self-contained, without reference to other enterprise data or logic, or in applications that can tolerate low-bandwidth integration with other IT assets. The same Web browser interface that gives users “anywhere, anytime” access to on-demand convenience may seem to be a barrier to the construction of strategic systems.

Enterprises evaluating on-demand options must look behind the browser (or other thin-client user interface), as noted³ by Fred Chong, a SaaS architecture strategist with Microsoft:

It is essential to test drive SaaS providers for integration hooks like programmable interfaces or messaging endpoints to enable the integration scenarios. Furthermore, as line-of-business applications commonly process long running transactions, it may also be necessary to look for SaaS provider interfaces that understand how to handle asynchronous application requests.

Similarly, enterprises may also need to consider exposing callback interfaces that are visible through the corporate firewalls so that processing results at the SaaS provider can be sent back and correlated with the original application request.

Integration capabilities are thus key differentiators between consumer- and enterprise-grade SaaS offerings – but this is not to say that on-demand integration is difficult *per se*. Any such perception is belied by a review of the extensive and powerful integration options available to developers on enterprise-oriented on-demand platforms.

Integrate what? how? why?

Like many terms used in IT, “integration” is an umbrella that spans a considerable range of specific requirements and solutions. A non-specific reference to integration goals or solutions may pertain to data, to business logic, or to the interface between a system and its interactive users or its interaction with cooperating systems and processes.

What’s being integrated?

Within the domains of data, logic and interface, there’s further segmentation due to widely varying needs.

- **Data integration** may entail shared access to a structured database, or consolidation (actual or virtual) of unstructured collections such as libraries of documents.

The structure and timing of data sharing operations may be as simple and as flexible as running a periodic export/import script, or as complex and tightly coupled as real-time synchronization with transactional integrity.

¹ McGillicuddy, Shamus, “Integration tricky for SMBs using multiple SaaS apps,” SearchSMB.com, 19 June 2007

² West, Mike, “SaaS Integration Platforms: The Looming SaaS Deployment and Support Dilemma,” STR-278, Saugatuck Technology, 6 October 2006

³ Chong, Frederick, “Enterprise SaaS Customers: Are You Ready for the Two Headed Monster?,” Fred Chong’s WebBlog, 24 August 2006

Beyond the browser

- :: Thin-client convenience for end users may look like a barrier to integration for IT architects and developers
- :: Evaluation of enterprise-class on-demand options requires consideration of APIs and partner ecosystems
- :: Success in extending on-demand benefits beyond single functions depends on thoughtful analysis of present and future integration needs and strategies

- **Logic integration** may entail a federation of packaged applications that are not designed to expose their internal functions as services, or it may be eased by non-proprietary service APIs.

It should not be assumed that “logic integration” takes place purely in an IT domain: it may require the identification and management, with or without automation, of what are currently manual processes.

- **Interface integration** may mean something as simple as placing side-by-side windows on a screen, possibly with a simple cut-and-paste capability for data transfer.

At the other extreme, it may entail the creation of sophisticated dashboards – perhaps with their own capabilities for computation at that interface layer – to merge, compare, or add other value to existing data feeds.

How will integration be performed?

A decision to integrate one or more systems, especially packaged systems or legacy systems that were never conceived as parts of a larger whole, begins a process of choice among many technical options:

- **Build application-specific connectors:** superficially attractive, this custom-code solution may seem to be a way to pay for only the function that’s needed, and to guarantee prompt attention to top priorities.

In most cases, though, the resulting architecture is likely to prove inflexible and lacking in useful features – while taking more time and costing more money than anticipated.

- **Devise and deploy an enterprise application bus:** a longer view will envision the need to adapt and adopt – that is, to accommodate changes to existing applications and to incorporate new systems going forward.

Some companies⁴ have accurately foreseen the benefits of this approach, and have realized attractive returns on investments without waiting for industry consensus to emerge in areas such as Web services and their off-the-shelf management tools. Even in those cases, however, enterprise IT architects expect their future investments to shift toward the use of Web services standards that they expect to yield more rapid time to market for new applications.

- **Adopt an off-the-shelf enterprise service bus** and modify applications as needed to use it. No one should think that this is a rapid process: even a successful initiative of this kind can take years.⁵ The major apparent benefit may be to outsource the problems of tracking evolving standards, providing semantically complete vocabularies of operations on service portfolios, and enabling discovery and re-use of services, but that happy state of affairs may only last as long as the initial “honeymoon” stage of a service-oriented architecture initiative.

Over time, the issue of integration may merely prove to have risen by one level of abstraction, as noted in an eWEEK interview⁶ with SOA Software Executive VP Frank Martinez:

Often discussed as the abstract backbone of a service-oriented architecture, any real service bus becomes at some point a specific vendor’s product—and Martinez observed that real implementations of ESB are just as hard to decompose, and to mix and match in multivendor constellations, as are other application confederations such as office suites.

“If you start to decompose the capabilities of an ESB, there are capability dependencies. It’s hard to leverage the messaging capabilities of the ESB independently of the mediation capabilities; it requires some form of service container, which today is built primarily on top of the application container,” Martinez said.

In the long run, the enterprise that adopts any single vendor’s enterprise service bus will likely still find itself someday addressing *ad hoc* integration issues, whether to accommodate diverse ESB choices of acquired or acquiring firms or to enable interaction with the systems of supply-chain partners and customers.

⁴ Coffee, Peter, “Con-Way, Guardian Cite SOA Success,” eWEEK, 18 April 2005

⁵ Coffee, Peter, “Bringing IT Challenges Down to Earth,” eWEEK, 12 December 2005

⁶ Coffee, Peter, “Developers Demand Disciplined Diversity,” eWEEK.com, 8 May 2006

More than one model

:: Connections may be easier to design with more limited function

:: Future costs of modification may eclipse short-term economies

:: Off-the-shelf enterprise service bus products may proliferate, creating new integration burdens

:: On-demand integration hubs may offer best prospects for a future-proof strategy

- **Adopt an on-demand platform as the integration hub:** from the perspective provided by the previous three scenarios, the use of an on-demand platform as integration hub seems not merely feasible but advantageous.

As noted⁷ by Laurie McCabe, vice president at Access Markets International Partners Inc., “SaaS applications are built on open standards platforms and open architectures. In that way, they’re almost more amenable to integration than proprietary on-premise applications.”

A similar sentiment comes from a report on salesforce.com integration offerings for Oracle and SAP products, with *The Register* analyst Gavin Clarke opining:⁸

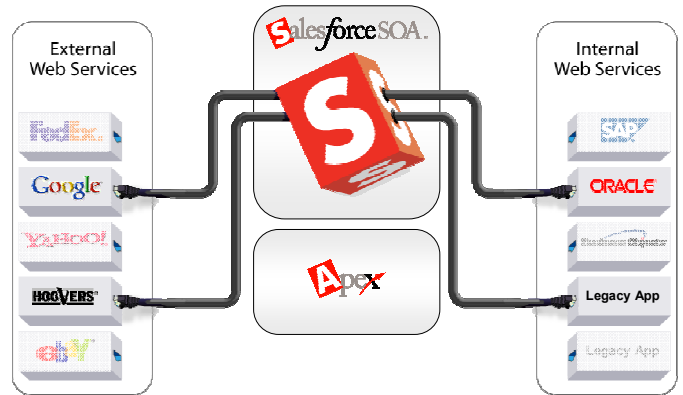
Single-tenancy systems traditionally have a tendency to be complex, thanks to their architecture and the number of modules plugged-in through customization, making upgrades and integration notoriously difficult.

[Salesforce.com] capitalizes on the multi-tenant model, where many people use the same platform, by making integration and customization easier. Connecting with non-salesforce.com systems occurs at the API layer making it possible to kick-start an outbound message from salesforce.com to another business application, such as an order fulfillment system.

Clarke’s comments, it will be noted, suggest that salesforce.com’s approach would be favorably appraised by Microsoft’s Fred Chong.

On-demand platforms exhibit growing capability to provide a foundation for integration, as seen in the May 2007 announcement⁹ of the salesforce SOA technology that enables both exposure and consumption of external Web services in Apex code modules running on salesforce.com systems (figure at right).

This is not to say, however, that a move to a Web services protocol strategy is a prerequisite to on-demand integration. For every one of the scenarios described above—custom coding, native connectors to specific software products, middleware layers facilitating multi-way integration, coupling with personal productivity applications, or user interface integrations of the type often dubbed the “mash-up”—there are options available for use with the salesforce.com platform (figure below).



⁷ McGillicuddy, *op.cit.*
⁸ Clarke, Gavin, “Salesforce.com integrates to get ahead,” *TheRegister.co.uk*, 27 November 2006
⁹ Ferguson, Renee, “Salesforce Announces On-Demand SOA Feature,” *eWEEK.com*, 21 May 2007

What is the goal of the integration effort?

In the same way that the word “integration” can describe many different IT domains and technologies, an integration project may be driven by any of several goals. When there is no shared understanding of why integration is being pursued, it’s unlikely that there will be consensus on whether the effort has been a success.

- Retain existing on-premise capabilities indefinitely

Continued use of legacy systems, in combination with on-demand elements, implies a need to accommodate patching, feature updates, and even major version replacements of the on-premise installations. Under these conditions, point-to-point connection among on-premise systems (with or without the inclusion of on-demand elements in the IT portfolio) is notoriously brittle, and a source of unpredictable costs and delays.

Movement to a service-oriented architecture appears to be a growing consensus strategy to minimize these concerns in general, with lowered barriers to on-demand integration as a collateral benefit.

- Minimize disruption during planned consolidation of redundant legacy applications into streamlined portfolio

When current on-premise systems are considered to be in their end-of-life phase, the feature specificity of point-to-point connections may have net advantages in reducing complexity compared to devising and deploying a more general and “future-proof” solution.

- Pave way for adoption of on-demand platform as future default mode of developing and deploying new applications

Migration of data to on-demand systems can be handled in phases, beginning with gateways and/or synchronization engines and gradually transitioning to an on-premise backup capability (if desired).

Existing on-premise business logic can continue to be executed using on-premise infrastructure, via API calls or other integration aids, until migration of those functions to “in the cloud” logic becomes feasible.

Are we there yet?

- :: Integration goals should be architectural, not just technical
- :: Success criteria should be defined by business need at least as much as by “speeds and feeds”
- :: Future mandates include security and governance as well as business function and partner interaction

What are the relevant measures of integration success?

- Costs, initial and ongoing
 - Licenses and administrative costs of on-premise software versus on-demand subscription fees
 - Facility space and upkeep cost for on-premise installations versus shifting of these costs to an external service provider
- Performance
 - Server price/performance trends

Capital expenditure per unit of computing capacity is improving most rapidly in multi-core CPU designs—for example, Sun Microsystems’ Niagara CPUs, expected to appear in August 2007 in a “Niagara 2” configuration of eight cores per processor.

With each core supporting eight threads of execution, this CPU will run 64 threads at half the power consumption per unit of work of the Niagara 1. Such designs are most fully utilized in shared arrangements such as a multi-tenant on-demand facility. Moore’s Law has not been repealed, but its jurisdiction is shifting from desktop and small servers back toward the largest data centers.

- Peak-to-average ratios in SOA environments

Successful creation of a service-oriented culture can be measured by the degree to which services are used in unexpected ways. Workloads may exhibit bursts of demand with far higher peak-to-average ratios than those of the traditional data center.

Moving new deployments to service provider infrastructure can therefore yield capital expenditure reductions even greater than those that would result from merely shifting currently projected capacity growth requirements.

- Uptime

In addition to higher peak-to-average ratios, a successful SOA initiative will be marked by more diverse use of services rather than mere modularity benefits. This can reduce the availability of safe maintenance windows, requiring higher uptime percentages.

- Transaction speed

In addition to improving processor price/performance ratios in shared systems, an on-demand installation can stay abreast of the state of the art in massively scalable long-distance bandwidth. Sustained trends at salesforce.com have steadily improved transaction response times, with typical measured results on the order of 250 milliseconds despite the growing transaction volumes mentioned above.

- Security

- Monitoring and detection of new vulnerabilities and actual attacks

Every point of integration is also a potential point of attack. Migrating systems to a single on-demand infrastructure minimizes this concern as the complexity of an application portfolio grows.

- Testing and deployment of patches

When on-premise products or hosted single-tenant stacks are individually maintained, the number of potential breakage modes quickly grows beyond acceptable bounds. Multi-tenant systems pose far less risk.

- Certifications and audits

Every unique interface represents a burden of identification, analysis and certification. The API-level integration of on-demand systems with on-premise systems, rather than the brittle and error-prone customization schemes in common use by on-premise products today, provides short-term gains in transparency and resilience; the shared integration model of salesforce.com's AppExchange is the future of application integration.

- Granularity of privileges

Integrated systems can create vulnerabilities for privilege-escalation attacks, as authorized access to one system becomes a pathway to exploitation of another. The strong granularity of privileges and the ease of privilege management in on-demand systems make them arguably superior as hubs for secure administration of complex integration arrangements.

The Americas

The Landmark @ One Market
Suite 300
San Francisco, CA 94105
United States of America
1-800-NO-SOFTWARE
www.salesforce.com

Latin America

Alfonso Napoles Gandara 50
4th floor
Col. Santa Fe
Mexico City
Mexico 01012
+52-55-9171-1882
www.salesforce.com

Japan

Ebisu Business Tower 18F
1-19-19 Ebisu, Shibuya-ku
Tokyo, 150-0013
Japan
+81-3-5793-8301
www.salesforce.com/jp

Asia/Pacific

9 Temasek Boulevard
#40-01 Suntec Tower 2
Singapore 038989
+65-6302-5700
www.salesforce.com/au

Europe, Middle East & Africa

Ch. de la Dent d'Oche 1B
1024 Ecublens
Switzerland
+353-1-2723-500
www.salesforce.com

