

Busting Myths of On-Demand: Why Multi-Tenancy Matters (Nov. 2007)

Peter Coffee
Director, Platform Research
pcoffee@salesforce.com

Contents

Myth: Multi-tenancy is not a core technology of IT as a service 1

Busted: Multi-tenant architecture is the foundation of on-demand advantage 1

Multi-tenancy and Salesforce.com 1

Software as What Kind of Service? 2

Single-Tenant Offsite Operations (Managed Service Providers) 2

Isolated-Tenant Application Services (Application Service Providers) 2

Multi-Tenant, On-Demand Software (& Platform) as a Service 2

Separating Business Logic from Data 3

Separating Provider Incentive from Customer Benefit 4

Myth:**Multi-tenancy is not a core technology of IT as a service**

During the first few decades of enterprise IT, the user organization typically owned and operated a single-tenant stack: mainframe or server hardware, systems software, application software and client-device infrastructure dedicated to that one organization's needs. Many enterprise technology providers have years of experience and billions of dollars invested in designing, deploying and supporting single-tenant software.

The relative costs of processing power, network bandwidth, and maintenance costs have changed. These changes have encouraged emergence of a multi-tenant model, in which a single high-capacity stack can support many user organizations while maintaining confidentiality and integrity of data and operations. Multi-tenancy offers many operational and economic benefits, but it represents challenge and disruption to a provider with only single-tenancy expertise.

A provider with a portfolio of single-tenancy skills and assets will find it tempting to go on the offensive: to suggest that the goals of the Software-as-a-Service model can be just as readily met by other means such as the use of server virtualization. Many enterprise software providers seek to blur the picture of multi-tenancy's benefits that has been clearly drawn by industry analysts as well as by the experience of many customers.

Busted:**Multi-tenant architecture is the foundation of on-demand advantage**

A multi-tenant foundation provides clear separation between common functions of the environment (e.g., preservation of database integrity) and unique behaviors defined by a single user organization (e.g., execution of specific business logic at the time of inserting a new database record). In a single-tenant stack, user customizations may become scattered throughout the stack in a way that makes it difficult to upgrade the performance of the stack without risking disruption of custom logic's functions. A multi-tenant environment, in contrast, can safely be upgraded without disrupting each separate user's well-demarcated customizations. Upgrades can therefore be made more often, with less customer risk and much lower adoption cost.

Multi-tenant platforms like salesforce.com's Force.com are similar to consumer Web platforms like Yahoo, Amazon, and eBay. Each represents a single code base that is shared by all users, with new capabilities accessible to all users simultaneously without the delays and labor-intensive processes of conventional software upgrades.

It's crucial to understand, though, that enterprise multi-tenancy is distinguished from consumer offerings in that an enterprise upgrade represents availability rather than imposition of new features and behaviors. An enterprise multi-tenant platform can continue to expose and support many past generations of capability, accessible by choice of each customer's administrative staff. Within a properly factored code base, this need not involve either redundancy or inconsistency: it lets each customer write its own timetable for adopting new features.

A major differentiator among various multi-tenant platforms is the degree of customization that can be represented in a form that appears to the customer as native to the platform—while being represented and managed within the platform as governable and upgrade-safe metadata. Broad statements about the limitations of multi-tenant architecture are often inaccurate generalizations, describing shallow designs in which metadata can only represent superficial aspects of a customer's preferences. There is no inherent reason why metadata representation can not extend to the deepest levels of database definition, custom business logic, and/or fundamental redesign of user interface appearance and behavior.

Multi-tenancy and Salesforce.com

Salesforce.com has pioneered new depth and power for multi-tenant platforms, setting a new standard for enterprise IT capability that has made tens of thousands of users wildly successful. Lower initial cost, faster deployment, zero-touch upgrade, lower administrative cost and greater user satisfaction are among the major benefits that flow to customers directly from the salesforce.com commitment to a multi-tenant model.

The future of software begins with the end of software. The end of software begins with Force.com, the multi-tenant Platform-as-a-Service**Read on for more**

Software as What Kind of Service?

Companies have traditionally owned and administered their own on-premise IT facilities. Service-based offerings began as both niche and novel: the issues of how to manage a service relationship tended to dominate discussions, and the niceties of how the service was being supported may have seemed to be low-level details.

As it turns out, there are many ways to take the functions of enterprise IT and package them as services: some of these service delivery modes merely relocate and reassign long-standing problems, while other models confront and mitigate—or even eliminate—some of the most vexing elements of IT operations.

Single-Tenant Offsite Operations (Managed Service Providers)

The simplest way to turn IT from product to service is to have a contractor purchase, configure, install, operate and maintain hardware and software in a contractor facility that would otherwise be located on enterprise premises and managed by enterprise employees. Such a facility might house many customers’ single-tenant stacks, each configured and operated largely independently of the others.

This offers the potential of immediate but superficial economies of moving IT installations to less costly office space, perhaps with sharing of capital-intensive hardware such as backup electrical power, and the hiring of personnel in lower-cost labor markets—along with improved utilization of expert skills that may be needed on short notice, but that are rarely required by any single user organization. Those savings are offset, though, by any given customer’s loss of ability to prioritize operations and maintenance efforts, perhaps resulting in delays to critical system updates or problem resolution processes.

Isolated-Tenant Application Services (Application Service Providers)

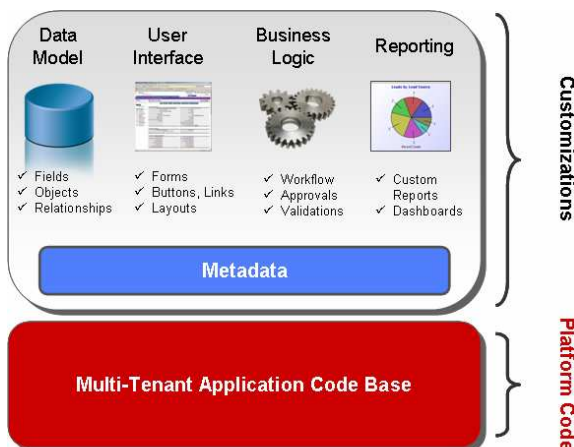
An enterprise might instead draw on multiple service providers, each supporting a single application or perhaps a multi-application portfolio. An application software provider might offer customers a choice between licensing the application code for installation and use on the customer’s premises, or subscribing to the use of an application stack maintained by the provider at the provider’s own site.

An application service provider arrangement offers some gains and some losses, compared to the offsite data center of an MSP as described above. An ASP setup has the potential for efficiencies resulting from a provider maintaining essentially identical copies of isomorphic application stacks, each using the same components but perhaps with differing capacities. Growing use of virtualization technology may seem to be a logical match for this model: it may be useful to define a single virtualized environment, and to copy and activate as many instances of that “virtual appliance” as are needed to provide adequate capacity under varying loads.

These potential benefits are offset, though, by difficulties of integrating application function, or sharing data, across both the physical and the architectural boundaries of single-tenant stacks that can not be expected to share common principles of design—and which will likely be operated at separate locations by uncoordinated teams.

Multi-Tenant, On-Demand Software (& Platform) as a Service

Instead of designing a conventional application for either on-premise deployment or isolated-tenancy service delivery, an application provider can begin with a plan to design an application that will support many separate organizations on a single instance of the application. Partitioning of data, rather than being a side effect of deploying multiple instances, becomes instead an element of the application’s design. Customization of application behavior, rather than being done by modifying application code, is done instead by configuring application metadata that’s invoked on a user-by-user basis as suggested by the figure below left. This multi-tenant architecture is the heart of the Force.com platform, and is one of the key innovations of salesforce.com.



A multi-tenant platform shares a common infrastructure and code base that is centrally maintained. Customer deployments are unique, separate, and secure within the multi-tenant environment, without the burdens on capital and labor that arise from separate stacks.

Multi-tenant application portfolios like those of salesforce.com, and general-purpose multi-tenant platforms like Force.com, are similar to consumer Web platforms like Yahoo, Amazon, and eBay. Both salesforce.com applications and Force.com projects bring their customizations and integrations forward without disruption whenever the service is upgraded: in time, the latter term may even lose the ironic edge that it has acquired in traditional IT arrangements.

Service Models Matter

- :: Service delivery should not merely relocate core technology problems to a contractor’s site
- :: Movement to a service-based model enables elimination of costly and risky elements of traditional IT architecture
- :: Multi-tenancy represents sharing of everything that can be shared, while proper metadata-based design continues to enable deep and rich customization for each user organization

Sources of economy in the multi-tenant model go far beyond those mentioned for MSP and ASP models described above. Operations and development teams need not divide their attention among multiple stacks, in proportion to the number of customers, but only in proportion to the number of physical instances of the platform and applications that are needed to provide acceptable aggregate capacity and fail-over reserves.

Customizations, being represented as metadata, become inherently safe—that is, unable to corrupt system function. It’s certainly the case that in early stages of development, a customization in progress they may not implement the precise behavior that the user intended, but this is readily addressed by a “sandbox” capability for testing code during development. In the multi-tenant environment, it is trivial to provide a test environment that is faithful to every detail of the production systems: the sandbox space (with its test data sets) merely looks like one more user, running in the same environment that’s designed to partition the data and the actions of each user from every other.

Separating Business Logic from Data

The clean separation between the logic of an application, and the data on which that logic operates, is a goal of many approaches to application development. Even the earliest COBOL applications defined such separation, but did not prevent developers from “hard-coding” critical values in ways that impeded code maintenance or reusability. Object techniques invited developers to conceive of an instance of a class as a bundle of data, surrounded and protected by a finite vocabulary of method and privilege sets for altering those values—but that invitation could still be ignored.

Multi-tenant architecture clarifies and strengthens the boundary between the platform and the applications that run on it, as well as between the applications and their data sets, making it possible to create applications that can readily be recomposed to meet different needs—while application logic remains reliably independent of the data it controls.

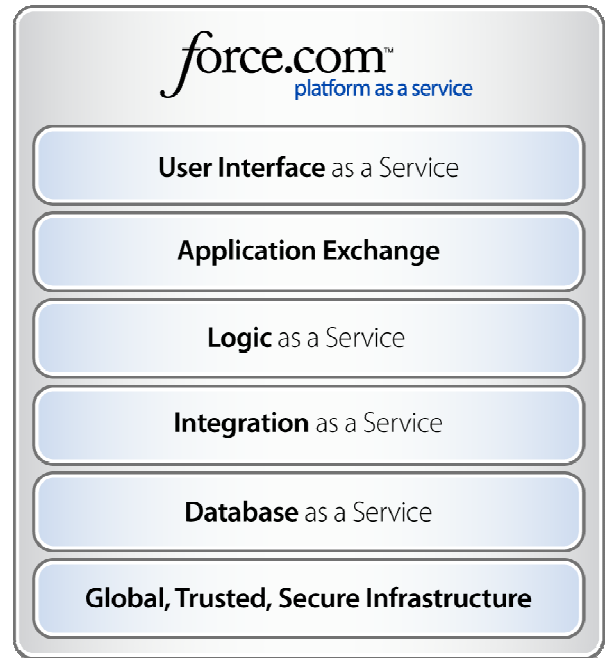
Instead of hard-coding data tables and page layouts, developers on a multi-tenant platform define attributes and behaviors as metadata, which functions as the application’s logical “blueprint.” This approach makes it possible for business users—those users most likely to be intimately familiar with the business processes the application is designed to support—to actually create the application by using simple point-and-click tools.

Application metadata can also include representation of custom logic—or so it can be made to appear to the developer. On the Force.com platform, for example, developers can write what looks to them like procedural code in the salesforce.com Apex programming language, while the multi-tenant platform can govern the process of turning those Apex expressions into run-time behaviors.

Problems such as infinite loops, which might ordinarily consume unreasonable fractions of a shared computing resource before detection and correction, can instead be identified by platform governance logic and can be safely aborted—with issuance of appropriate alerts to the code’s maintainer.

Not Just Theory, but Practice

- :: Multi-tenant architecture makes good design the downhill direction: clean separation of logic from data becomes a natural outcome, rather than demanding draconian discipline
- :: What looks to the developer like full-strength custom code appears to the platform as well-governed metadata: developer productivity and application robustness are both improved



Separating Provider Incentive from Customer Benefit

Given these fundamental differences between different modes of delivering IT as a service, it's blatantly self-serving for a software executive to declare—for example—that “Multi-tenancy has nothing to do with on-demand. [It is] a convenience for the vendor. Whether they put all customer data onto one database or onto multiple databases is of no value to the customer. In the enterprise it is the opposite. They do not want to put all their data [into the same database] as their competitors.”¹

The statement just cited is precisely as disingenuous as saying that companies “do not want to put all their money into the same bank account as their competitors,” when what those companies are actually doing is putting their money into different accounts at the same bank. When customers of a financial institution share what does not need to be partitioned—for example, the transactional logic and the database maintenance tools of a major financial institution—then they enjoy advantages of capacity, consistency and reliability that would not be likely to be affordably deliverable in isolated parallel systems. Multi-tenancy likewise offers a compelling combination of efficiency and capability without loss of security or flexibility.



Technology providers seek to blur the distinctions between one technology and another for one reason only: because they are unable to offer the superior technology to their customers, and hope to persuade their customers that real differences are not relevant to their needs. Multi-tenant platforms for enterprise on-demand applications represent genuine opportunities for customer advantage.

The reality of multi-tenant differentiation is acknowledged by authoritative industry analysts such as Gartner, whose March 2007 announcement² of its Outsourcing Summit that month included this definition of Software as a Service: “Hosted software based on a single set of common code and data definitions that are consumed in a one-to-many model.” In other words, hosting models that do not offer the leverage of multi-tenancy don't belong in the same discussion as the value proposition implied by the term, “SaaS.”

Multi-tenancy is a difference that makes a difference.

For More Information

Contact your account executive to learn how we can help you accelerate your CRM success.

¹ Wainwright, Phil, quoting Oracle Corp. President Charles Phillips, in “Oracle's misconceived SaaS strategy,” Software as Services (ZDNet.com), 27 August 2007

² Gartner Inc., “SaaS will have significant impact on IT services and outsourcing providers,” Tekrati, 7 March 2007

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 +001-415-536-4606 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 Suntec Tower Two 9 Tamasek Tower 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
---	--	---	---	--