Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.640 – 647

RESEARCH ARTICLE

The Multitenant Cloud Architecture

Understanding the Design of Salesforce.com's Internet Application Development Platform

JUEE DARYAPURKAR, Computer Science and Engineering, H.V.P.M. College of Engineering, Amravati

Email: jd22491@gmail.com

PROF. KARUNA BAGDE, Computer Science and Engineering, H.V.P.M. College of Engineering, Amravati

Email: karunabagde@rediffmail.com

ABSTRACT

Force.com is the preeminent on-demand application development platform in use today, supporting some 47,000+ organizations. Individual enterprises and commercial software-as-a service (SaaS) vendors trust the platform to deliver robust, reliable, Internet-scale applications. To meet the extreme demands of its large user population, Force.com's foundation is a metadata driven software architecture that enables multitenant applications. This paper explains the patented technology that makes the Force.com platform fast, scalable, and secure for any type of application.

Keywords: Force.com, multitenant, software-as-a service (SaaS)

INTRODUCTION

History has shown that every so often, incremental advances in technology and changes in business models create major paradigm shifts in the way software applications are designed, built, and delivered to end users. The invention of personal computers (PCs), computer networking and graphical user interfaces (UIs) gave rise to the adoption of client/server applications over expensive, inflexible, character-mode mainframe applications and today, reliable broadband Internet access, service-oriented architectures (SOAs), and the cost

inefficiencies of managing dedicated on-premises applications are driving a transition toward the delivery of decomposable.

Web-based services called software as a service (SaaS). Yet existing application frameworks are not designed to address the special needs of SaaS. This void has given rise to another new paradigm shift, namely platform as a service (PaaS). Hosted application platforms are managed environments specifically designed to meet the unique challenges of building SaaS applications and deliver them more cost efficiently than ever before.

Multitenant Applications

To decrease the cost of delivering the same application to many different sets of users, an increasing number of applications are multitenant rather than single-tenant. Whereas a traditional single-tenant application requires a dedicated set of resources to fulfil the needs of just one organization, a multitenant application can satisfy the needs of multiple tenants (companies or departments within a company, etc.) using the hardware resources and staff needed to manage just a single software instance (Figure 1).

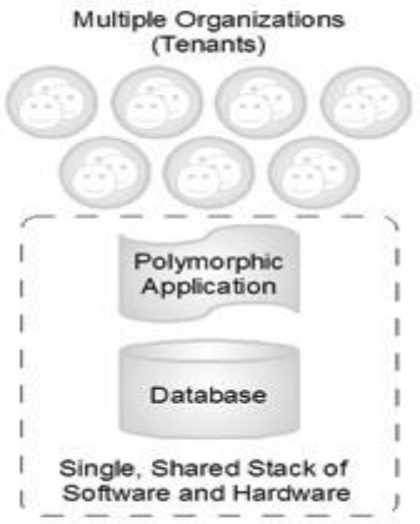

Figure 1: A multitenant application cost-efficiently shares a single stack of resources to satisfy the needs of multiple organizations. Tenants using a multitenant service operate in virtual isolation from one another: Organizations can use and customize an application as though they each have a separate instance, yet their data and customizations remain secure and insulated from the activity of all other tenants. The single application instance effectively morphs at runtime for any particular tenant at any given time.

Multitenancy is an architectural approach that pays dividends to both application providers and users. Operating just one application instance for multiple organizations yields

tremendous economy of scale for the provider. Only one set of hardware resources is necessary to meet the needs of all users, a relatively small, experienced administrative staff can efficiently manage only one stack of software and hardware, and developers can build and support a single code base on just one platform (operating system, database, etc.) rather than many. The economics afforded by multitenancy allow the application provider to, in turn, offer the service at a lower cost to customers.

Some interesting side benefits of multitenancy are improved quality, user satisfaction, and customer retention.

Comparing Raw Cloud Computing and PaaS

Raw computing clouds are machine-centric services that provide on-demand infrastructure as a service (IaaS) for the deployment of applications. Such clouds provide little more than the computing power and storage capacity needed to execute virtual servers that comprise an application. Some SaaS vendors looking for a quick go-to-market strategy avoid the challenges of developing a true multitenant solution and choose to deliver single-tenant instances via IaaS. Platform as a service (PaaS) such as Force.com is an application-centric approach that abstracts the concept of servers altogether. PaaS lets developers focus on core application development from day one and to deploy an application with the push of a button. The provider never needs to worry about multitenancy, high availability, load balancing, scalability, system backups, operating system patches and security, and other similar infrastructure-related concerns—all these services are delivered as the "S" in PaaS.

Metadata-Driven Architectures

Multitenancy is practical only when it can support applications that are reliable, customizable, upgradeable, secure, and fast. But how can a multitenant application allow each tenant to create custom extensions to standard data objects and entirely new custom data objects? And how will the application's response time scale as tens of thousands of tenants subscribe to the service? It's difficult to create a statically compiled application executable that can meet these and other unique challenges of multitenancy. Inherently, a multitenant application must be dynamic in nature, or polymorphic, to fulfil the individual expectations of various tenants and their users. For these reasons, multitenant application designs have evolved to use a runtime engine that generates application components from metadata—data about the application itself. In a well-defined metadata-driven architecture (Figure 2), there is a clear separation of the compiled runtime engine (kernel), application data, the metadata that describes the base functionality of an application, and the metadata that corresponds to each

tenant's data and customizations. These distinct boundaries make it possible to independently update the system kernel, modify the core application, or customize tenant-specific components, with virtually no risk of one affecting the others.

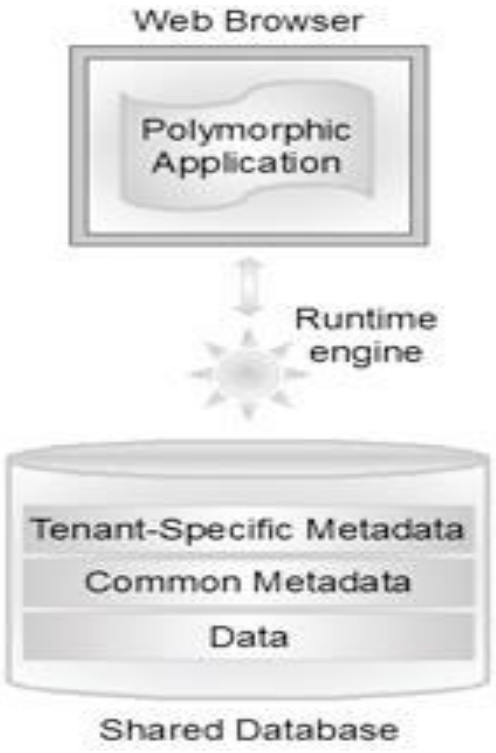

Figure 2: A metadata-driven application had clear separation between the runtime engine, data, common application metadata, and tenant-specific metadata.

New Challenges and Emerging Solutions

Attempting to weave multitenancy throughout the fabric of an application's core logic and its underlying infrastructure is a complex undertaking. Building metadata-driven, multitenant applications from scratch without any prior experience is destined to be a timeconsuming and error-prone effort. In the end, many would-be SaaS providers struggle to succeed in building multitenant applications and end up wasting valuable time that could have been spent focused on the innovation of core application functionality and features. One problem is that traditional application development frameworks and platforms are not equipped to handle the special needs of modern Internet applications. As a result, new types of platforms are emerging to help simplify the development and deployment of multitenant applications. Force.com is the first and most mature generalpurpose, multitenant, Internet application development platform available today. The remaining sections of this paper explain specific details about the technical design of Force.com so you can better understand its capabilities.

What is Force.com?

Force.com is an on-demand platform designed to build and run business process and data applications on Salesforce.com.

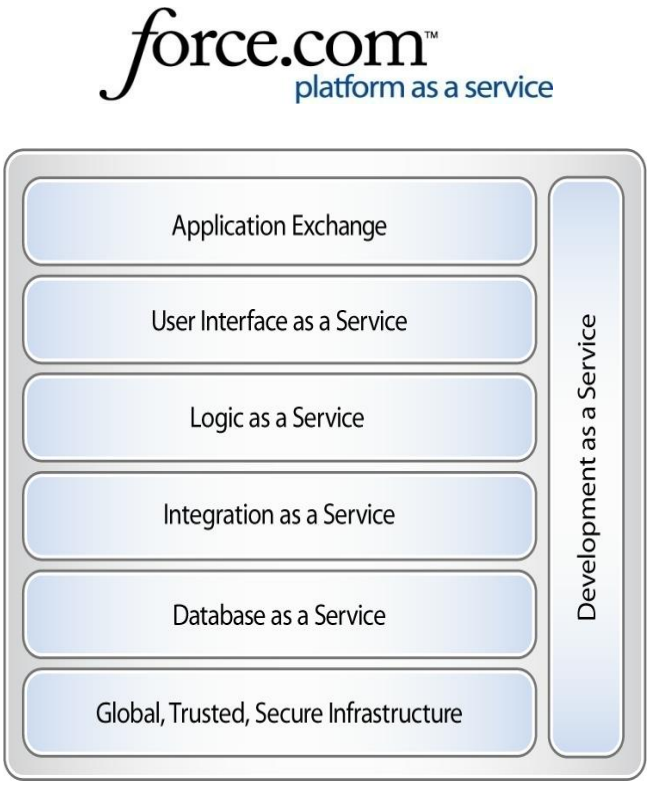

Figure 3: The Force.com platform

The Force.com platform is different from more traditional platforms, as shown in the diagram above. A traditional platform supplies only the core services of database access and containers for logic and presentation. One has to use other software to create entire application, as well as the additional components to run the data center which supports the platform. Of course, additional components means additional expenditure, for acquisition, maintenance, and integration.

The Force.com platform provides a complete stack—including a complete user interface to the data, full reporting and analytic capabilities, a flexible security and sharing model, and other services not displayed in the diagram, such as workflow and approvals—fully available and integrated from the outset. There are many extended features such as built-in internationalization, full support for mobile devices, and integration with existing systems, all of which are beyond the scope of this paper.

Companies and ISVs are using the Force.com platform to build everything: supply chain management, audit, tax calculation, event management, compliance tracking, brand management, pricing, accounts receivable, accounts payable, billing, HR, payment processing, employee on-boarding, claims processing, and many others.

Force.com Platform Architecture

Force.com's optimized metadata-driven architecture delivers extraordinary performance, scalability, and customization for on-demand, multitenant applications (Figure 3).

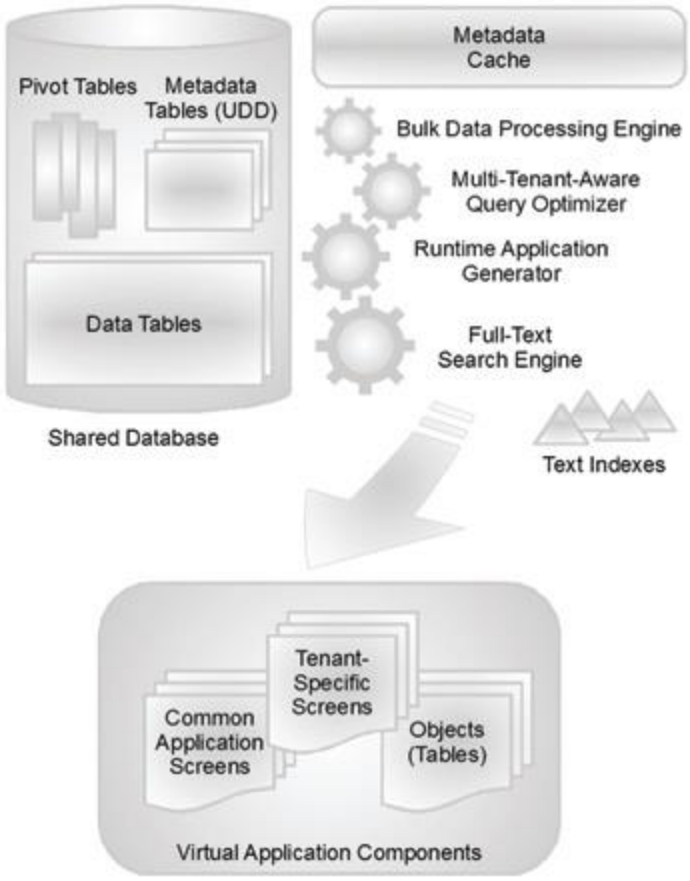

Figure 4: Force.com's metadata-driven architecture optimally generates virtual application components at runtime.

In Force.com, everything exposed to developers and application users is internally represented as metadata. Forms, reports, work flows, user access privileges, tenant-specific customizations and business logic, even the definitions of underlying data tables and indexes, are all abstract constructs that exist merely as metadata in Force.com's Universal Data Dictionary (UDD). For example, when a developer is building a new custom application and defines a custom table, lays out a form, or writes some procedural code, Force.com does not

create an "actual" table in a database or compile any code. Instead,Force.com simply stores metadata that the platform's engine can use to generate the "virtual" application components at runtime. When someone wants to modify or customize something about the application, all that's required is a simple non-blocking update to the corresponding metadata.

To optimize access to data in the system's large tables, Force.com's engine relies on a set of specialized pivot tables that maintain denormalized data for various purposes such as indexing, uniqueness, relationships, etc.

Force.com's data processing engine helps streamline the overhead of large data loads and online transaction processing applications by transparently performing data modification operations in bulk. The engine has built-in fault recovery mechanisms that automatically retry bulk save operations after factoring out records that cause errors.

As Force.com's runtime application generator dynamically builds applications in response to specific user requests, the engine relies heavily on its "multitenant-aware" query optimizer to execute internal operations as efficiently as possible. The query optimizer considers which user is executing a given application function, and then, using related tenant-specific metadata maintained in the UDD along with internal system pivot tables, builds and executes data access operations as optimized database queries.

CONCLUSIONS

The platform's metadata driven architecture enables anyone to efficiently build and deliver sophisticated, customizable, internet-scale multitenant applications. Using standards-based Web service APIs and native platform development tools, Force.com developers can easily build all components of a Web-based application, including the application's data model (tables, relationships, etc.), user interface (data entry forms, reports, etc.), business logic (workflows, validations, etc.), integrations with other applications, and more. Platform features such as the bulk data processing API help make multitenant platform applications highly efficient and scalable with little or no thought from developers. Salesforce.com's managed approach for the deployment of production applications ensures top-notch performance, scalability, and reliability for all dependent applications.

REFERENCES

[1] https://wiki.developerforce.com/page/Multi_Tenant_Architecture

[2] http://www.developerforce.com/media/Forcedotcom_Whitepaper/WP_Forcedotcom-InDepth_040709_WEB.pdf

[3] http://www.spinnakerlabs.com/CloudComputing.pdf "Welcome to the new era of cloud computing PPT"

[4] J. Gray, "Distributed computing economics," ACM Queue, vol. 6, pp. 63-68, 2008.

[5] M. Nelson, "Building an Open Cloud," vol. 324, p. 1656, 2009.

[6] http://www.infoworld.com/article/08/04/07/15FE-cloud-computing-reality_1.html.

[7] Dr. Werner Fritsch, Article name: Cloud computing als IT-Architecture und Outsourcing-Option, Information week, May 19, 2008

[8] (http://informationweek.de/showArticle.jhtml?articleID=207800804&pgno=1) Herrmann 2008 Cloud Computing – das Buzzword des Jahres?Wolfgang Herrmann, Computerwoche, April 9, 2008

[9](http://www.computerwoche.de/knowledge_center/software_infrastruktur/1860108/)

[10] Chris Howard und Kathrin Schmitt, Atrticle name: Die Wolke und Du – was bringt Cloud Computing? Burton Group, October 8, 2008

