International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.1, No.1, Mar ch 2012, pp. 12~17
ISSN: 2089-3337

Cloud Computing-Software as Service

Gurudatt Kulkarni*, Jayant Gambhir*, Rajnikant Palwe*
* Lecturer, Marathwada Mitra Mandal's Polytechnic, Pune, India

Article Info

Article history:

Received Dec 20th, 2011
Revised Feb 10th, 2012
Accepted March 5th, 2012

Keyword:

Cloud Computing
Software as a services

ABSTRACT

Enterprises would do well to consider the flexibility and risk-management implications of adding SaaS to their portfolios of IT services. Integration and composition are critical components in your architecture strategies to incorporate SaaS successfully as a fully participating member of your service-centric IT infrastructure. Today, SaaS applications are expected to take advantage of the benefits of centralization through a single-instance, multi-tenant architecture, and to provide a feature-rich experience competitive with comparable on-premise applications. A typical SaaS application is offered either directly by the vendor or by an intermediary party called an aggregator, which bundles SaaS offerings from different vendors and offers them as part of a unified application platform.

Corresponding Author:

First Author,
Marathwada Mitra Mandal's Polytechnic, Pune, India.
Email: gurudatt.kulkarni@mmpolytechnic.com

1. INTRODUCTION

Cloud Computing," to put it simply, means "Internet Computing." The Internet is commonly visualized as clouds; hence the term "cloud computing" for computation done through the Internet. With Cloud Computing users can access database resources via the Internet from anywhere, for as long as they need, without worrying about any maintenance or management of actual resources. Besides, databases in cloud are very dynamic and scalable. Cloud computing is unlike grid computing, utility computing, or autonomic computing. In fact, it is a very independent platform in terms of computing. The best Example of cloud computing is Google Apps where any application can be accessed using a browser and it can be deployed on thousands of computer through the Internet.

2. WHAT IS CLOUD COMPUTING

Cloud computing provides the facility to access shared resources and common infrastructure, offering services on demand over the network to perform operations that meet changing business needs. The location of physical resources and devices being accessed are typically not known to the end user. It also provides facilities for users to develop, deploy and manage their applications 'on the cloud', which entails virtualization of resources that maintains and manages itself. Some generic examples include:

- Amazon's Elastic Computing Cloud (EC2) offering computational services that enable people to use CPU cycles without buying more computers
- Storage services such as those provided by Amazon's Simple Storage Service (S3) [1]
- Companies like Nirvanix allowing organizations to store data and documents without adding a single on-site server
- SaaS companies like Salesforce.com delivering CRM services, so clients can manage customer information without installing specialized software

Cloud computing can be categorized into three parts:

1. Platform as a Service (PaaS)
2. Infrastructure as a Service (IaaS)
3. Software as a Service (SaaS)

3. SOFTWARE AS A SERVICE (SAAS)

SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet. SaaS is generally used to refer to business software rather than consumer software, which falls under Web 2.0. By removing the need to install and run an application on a user's own computer it is seen as a way for businesses to get the same benefits as commercial software with smaller cost outlay. SaaS can alleviate the burden of software maintenance and support but users relinquish control over software versions and requirements. Other terms that are used in this sphere include Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). A public cloud sells services to anyone on the Internet. A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. All the major companies have come up with their own code based or non-code based cloud computing frameworks. Some of the most prominent code-based frame works are:

- Java Google web Toolkit (Google App Engine).
- Python Djangno (Google App Engine)
- Ruby on Rails
- Microsoft.NET (Azura Service Platform)

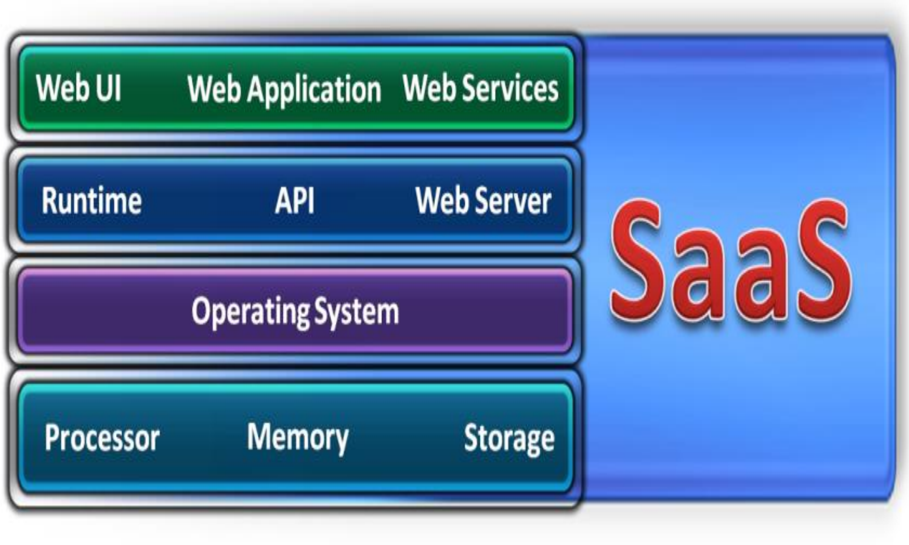

Figure1. SaaS Structure

When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services. SaaS is one of the methodologies of Cloud Computing, which is based on a "one-to-many" model whereby an application is shared across multiple clients. The exact definition of software as a service (SaaS) is open to debate, and asking different people would probably result in different definitions. Everyone believe that SaaS is going to have a major impact on the software industry, because software as a service will change the way people build, sell, buy, and use software.

For this to happen, though, software vendors need resources and information about developing SaaS applications effectively. Still, most experts would probably agree on a few fundamental principles that distinguish SaaS from traditional packaged software on the one hand, and simple websites on the other. Expressed most simply, software as a service can be characterized as "Software deployed as a hosted service and accessed over the Internet." Software as a service (or SaaS) is a way of delivering applications over the Internet-as a service. Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management. SaaS applications are sometimes called Web-based software, on-demand software, or hosted software. Whatever the name, SaaS applications run on a SaaS provider's servers.

The provider manages access to the application, including security, availability, and performance. SaaS customers have no hardware or software to buy, install, maintain, or update. Access to applications is easy: you just need an Internet connection. This types of cloud computing delivers a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one app to maintain, costs are low compared to conventional hosting.Salesforce.com is by far the best-known example among enterprise applications which provide CRM solutions as SaaS, but SaaS is also common for HR apps and has even worked its way up the food chain to ERP, with players such as Workday. Beside these, some of the desktop applications like Google Apps and Zoho Office had made their mark in the market.

3.1. The Aplication Arcitecture

Much like any other software, Software as a Service can also take advantage of Service Oriented Architecture to enable software applications to communicate with each other. Each software service can act as a service provider, exposing its functionality to other applications via public brokers, and can also act as a service requester, incorporating data and functionality from other services. It is important to understand that the SaaS methodology requires system architecture capable of supporting peak usage demands and the ability to process large numbers of transactions in a secure and reliable environment.

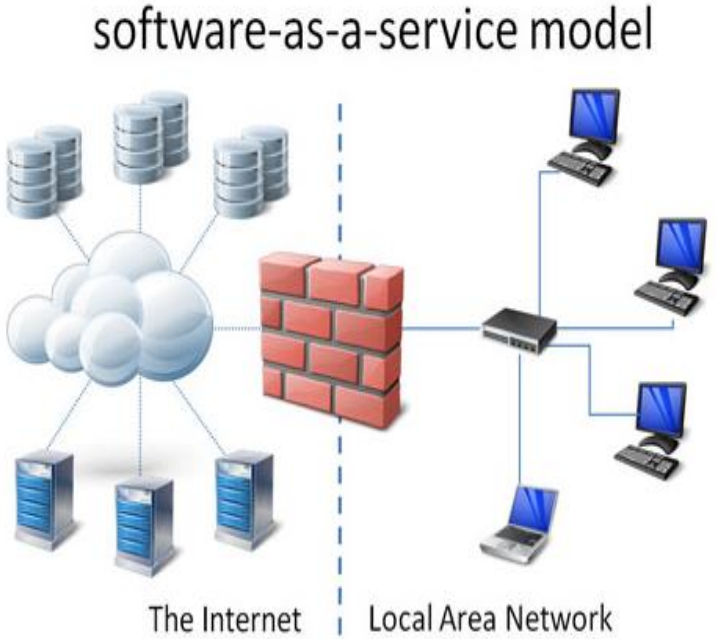

Figure 2. Software as Service Model

The software would need to meet certain criteria's to work on a model such as this. The application would need to be well architected to sustain and provide the scalability, ease of use of the traditional desktop applications. There are three key points which would differentiate a successful SaaS application from an un-successful SaaS application:

- Scalability: Scaling the application means maximizing concurrency and using application resources more efficiently-for example, optimizing locking duration, statelessness, sharing pooled resources such as threads and network connections, caching reference data, and partitioning large databases.
- Multi-tenant efficient: Multi-tenancy may be the most significant paradigm shift that an architect accustomed to designing isolated, single-tenant applications has to make. For example, when a user at one company accesses customer information by using a CRM application service, the application instance that the user connects to may be accommodating users from dozens, or even hundreds, of other companies-all completely abstracted to any of the users. This requires an architecture that maximizes the sharing of resources across tenants, but that is still able to differentiate data belonging to different customers.

- Configurable: if a single application instance on a single server has to accommodate users from several different companies at once, you can't simply write custom code to customize the end-user experience-anything you do to customize the application for one customer will change the application for other customers as well. Instead of customizing the application in the traditional sense, then, each customer uses metadata to configure the way the application appears and behaves for its users. The challenge for the SaaS architect is to ensure that the task of configuring applications is simple and easy for the customers, without incurring extra development or operation costs for each configuration There can be four ways of hosting an application on the SaaS architecture. These are also called as the maturity models of SaaS:
- Ad-hoc/Custom: It is similar to the traditional application service provider (ASP) model of software delivery, dating back to the 1990s. Each customer has its own customized version of the hosted application, and runs its own instance of the application on the host's servers.
- Configurable: The vendor hosts a separate instance of the application for each customer (or tenant). Unlike the previous one, each instance is individually customized for the tenant, at this level, all instances use the same code implementation, and the vendor meets customers' needs by providing detailed configuration options that allow the customer to change how the application looks and behaves to its users. Despite being identical to one another at the code level, each instance remains wholly isolated from all the others.

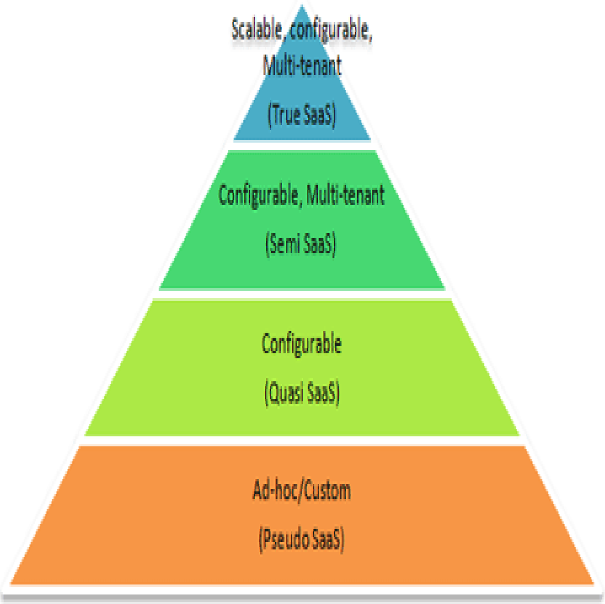

Figure 3. SaaS Various Type

- Configurable, Multi-tenant-efficient: The vendor runs a single instance that serves every customer, with configurable metadata providing a unique user experience and feature set for each one. Authorization and security policies ensure that each customer's data is kept separate from that of other customers; and, from the end user's perspective, there is no indication that the application instance is being shared among multiple tenants. This approach eliminates the need to provide server space for as many instances as the vendor has customers, allowing for much more efficient use of computing resources than the second level, which translates directly to lower costs. A significant disadvantage of this approach is that the scalability of the application is limited.
- Scalable, configurable, Multi-tenant-efficient: The vendor hosts multiple customers on a load-balanced farm of identical instances, with each customer's data kept separate, and with configurable metadata providing a unique user experience and feature set for each customer.

3.2. Characteristic in Software as a Service

Although the basic definition of cloud computing could also be used in Software as a Service, there are basic differences SaaS have when compared to other forms of cloud computing.

- Network or Online Access – SaaS is an online application or at least, a network based application. Users will never need any installation in their local gadgets which is connected to the local network or the internet. Usually, the application is launched through a browser which could provide access not only to the application but additional services from the vendor.
- Centralized Management – control, monitoring and update could be done in a single location. The businesses that maintain the application will never need to manually make some changes in the local gadget but would provide improvement instead on the online application.
- Powerful Communication Features – Software as a Service is not only based on the fact that it provides functions for online processing, it also has powerful communication features. The mere fact that SaaS is often used online provides a strong backbone for Instant Messaging (Chat) or even voice calls (VOIP).

3.3. Advantage/ Disadvantages of SaaS

Software as a Service is geared towards specific type of business. Although they can easily work in most enterprise settings, there are certain requirements SaaS would have that make it undesirable for some businesses.

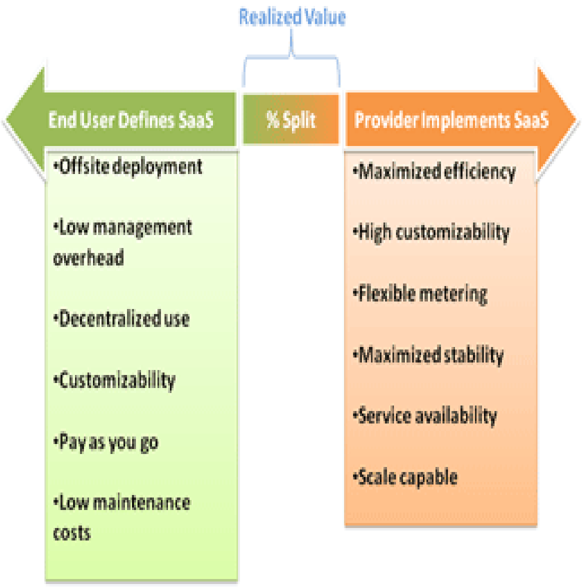

Figure 4. Advantage of SaaS in Software Developement

- **Powerful Internet Connection required** – although connection online is available almost everywhere, the rate of connection is never the same. Some areas can't provide strong internet connection and SaaS (as an online application) will have to load everything in the browser. The expected function might not even move forward without strong internet connectivity.
- **Increased Security Risk** – attacks are highly likely if everything is launched online. This is probably the most challenging part in SaaS and in Cloud Computing industry. SaaS has increase security concerns compared to other platforms because of its consistent interaction with different users.
- **Load Balancing Feature** – one of the challenges the business would face in cloud computing and all SaaS applications is load balancing. Although industry giants offer load balancing, it will still require consistent monitoring from businesses.

3.4. API and MASH-UPS in SaaS

SaaS is getting better and better as new trends in the industry are slowly being implemented. Among the trends in cloud computing is the powerful integration of API or Application Programming Interface. Although SaaS could provide the functionality the business needs, upgrades are important to keep up with the demands. Instead of changing the application, businesses will just add an API in their application. The integration is easy and maximum efficiency of the additional function is expected. Mash-up is another trend seen in SaaS which is a technique in combining two powerful applications.

4. CONCLUSION

Enterprises would do well to consider the flexibility and risk-management implications of adding SaaS to their portfolios of IT services. Integration and composition are critical components in your architecture strategies to incorporate SaaS successfully as a fully participating member of your service-centric IT infrastructure. Today, SaaS applications are expected to take advantage of the benefits of centralization through a single-instance, multi-tenant architecture, and to provide a feature-rich experience competitive with comparable on-premise applications. A typical SaaS application is offered either directly by the vendor or by an intermediary party called an aggregator, which bundles SaaS offerings from different vendors and offers them as part of a unified application platform.

REFERENCES

[1] “Cloud Computing: An Overview”, Srinivasa Rao V, Nageswara Rao N K, E Kusuma Kumari, Journal Of Theoretical And Applied Information Technology.

[2] http://www.c-sharpcorner/cloud-computing-saas, Cloud Computing – SaaS By Bhaskardeep January 18, 2010.

[3] http://www.exforsys.com/tutorials/cloud-computing/saas-model-cloud-computing.html

[4] “Cloud Computing: A New Phase in Information Technology Management”, Al Bento, Journal Of Information Technology Management

[5] Cloud Computing: A Study Of Infrastructure As A Service (Iaas), Sushil Bhardwaj, Leena Jain, Sandeep Jain, International Journal Of Engineering And Information Technology