An Exploration of Cloud Service Delivery Models in a Hybrid Environment – A New Depiction to Corporate Computing

C. Vijayalakshmi
Department of Computer Science & Engineering
Jubail University College – Female branch
Dammam, Kingdom of Saudi Arabia
chakravarthyv@ucj.edu.sa

ABSTRACT: *Cloud computing has evolved from a risky and confusing concept to a strategy that organizations large and small are beginning to adopt as part of their overall computing strategy. Cloud computing turns traditional siloed computing assets into shared pools of resources that are based on an underlying Internet foundation. Cloud computing makes these resources easier to use by providing standardization and automation. A private cloud provides a flexible environment with a higher level of security than would be available in a public cloud. Combining public services with private clouds and the data center as a hybrid is the new definition of corporate computing. To really comprehend the cloud, you need to understand the three foundational delivery models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In this paper, we examine the different models that are available to deliver cloud services – both public and private.*

Keywords: Cloud Service Delivery Models, IaaS, SaaS and PaaS, Delivery Models in Cloud Computing, Hybrid Computing Service Delivery Models, Different Models in Cloud Service Delivery

Received: 18 July 2013, Revised 21 August 2013, Accepted 25 August 2013

1. Introduction

Cloud computing is a method of providing a set of shared computing resources that includes applications, computing, storage, networking, development, and deployment platforms as well as business processes. The cloud provides a new economic model of computing. Instead of purchasing, managing, and maintaining a self-contained, traditional data center, a business is able to transform computing into a more streamlined computing environment that better serves changing computing requirements. Clouds come in different versions, depending on your needs. The *public cloud* is a set of hardware, networking, storage, services, applications and interfaces owned and operated by a third party for use by other companies or individuals. The *private cloud* is a set of hardware, networking, storage, services, applications and interfaces owned and operated by an organization for the use of its employees, partners, and customers. A *hybrid cloud* is a combination of a private cloud combined with the use of public cloud services where one or several touch points are between the environments.

Understanding the foundations of cloud computing calls for an understanding the different cloud delivery models. Service delivery in Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). *Infrastructure as a Service* (*IaaS*) is the delivery of services such as hardware,

Figure 1. Service models of Cloud computing

storage, networking, data center space, and various utility software elements on a request basis. There are both public and private versions of IaaS. In the public IaaS, the user simply needs a credit card to acquire these resources. When that user stops paying, the resource disappears.

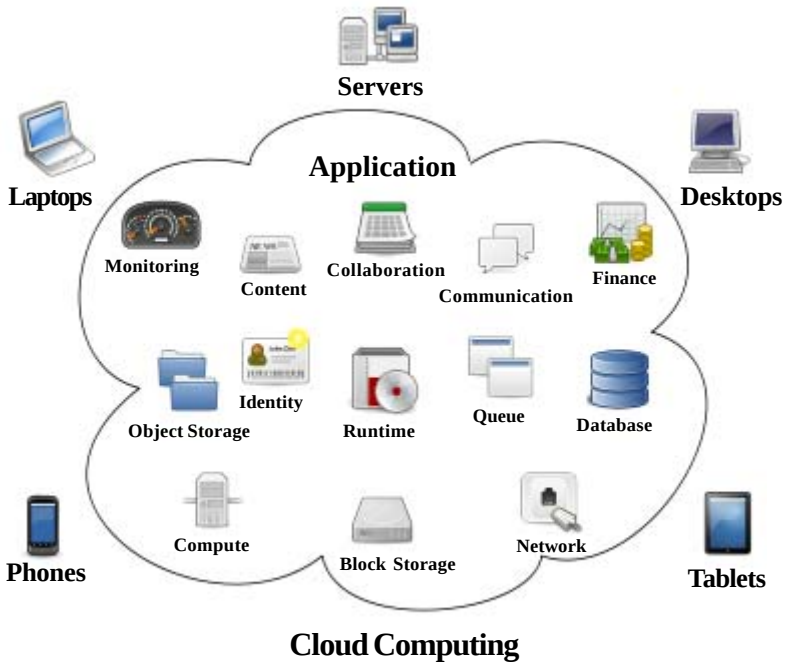

Figure 2. Cloud computing logical diagram

In a private IaaS service, it is usually the IT organization or an integrator who creates an infrastructure designed to provide resources on demand to internal users and sometimes business partners. *Platform as a Service* (*PaaS*) is a mechanism for combining IaaS with an abstracted set of middleware services, software development, and deployment tools that allow the organization to have a consistent way to create and deploy applications on a cloud or on-premises environment. *Software as a*

Service (*SaaS*) is a business application created and hosted by a provider in a multi-tenant model. Customers pay for the service either per user on a monthly or yearly contract model. The SaaS application sits on top of both a Platform as a Service and foundational Infrastructure services.

2. Exploring Infrastructure as a Service

Infrastructure as a Service (IaaS) is the most straightforward of the three models for delivering cloud services. IaaS is the virtual delivery of computing resources in the form of hardware, networking, and storage devices. It may also include the delivery of operating systems and virtualization technology to manage the resources. Rather than buying and installing the required resources in their own data center, companies are renting these required resources on an as-needed basis.

Two types of IaaS are *public* and *private*. A public cloud IaaS service is designed so consumers in any size business can acquire services in a rental model. In contrast, private IaaS services are provided inside a company's own data center, enabling IT management to provide a self-service portal for employees and partners to easily access approved services.

Figure 3. Deployment models of Cloud computing

2.1 Understanding the key characteristics of IaaS in a Public Cloud

The key characteristics of IaaS delivery model includes dynamic scaling, agreed-upon service levels, renting, licensing, metering and self-service. All the characteristics are the same whether the environment is a public or a private IaaS environment.

2.2 Dynamic Scaling

Some level of uncertainty always exists when planning for IT resources. One of the major benefits of IaaS for companies faced with this type of uncertainty is the fact that resources can be automatically scaled up or down based on the requirements of the application. This important characteristic of IaaS is called *dynamic scaling*. If customers wind up needing more resources than expected, they can get them immediately. A provider or creator of IaaS typically optimizes the environment so that the hardware, the operating system, and automation can support a huge number of workloads.

2.3 Service levels

Additionally, consumers acquire IaaS services in different ways. Many consumers rent capacity based on an on-demand model with no contract. In other situations, the consumer signs a contract for a specific amount of storage or compute. A typical IaaS contract has some level of service guarantee. At the low end, a provider might simply state that the company will do its best to provide good service. If the consumers are willing to pay a premium price, they might be able to leverage a mirrored service so

that there is almost no change of service interruptions. A typical service level agreement states what the provider has agreed to deliver in terms of availability and response to demand. The service level might, for example, specify that the resources will be available 99.999 percent of the time and that more resources will be provided dynamically if greater than 80 percent of any given resource is being used.

2.4 The rental model

When companies use IaaS, it is often said that the servers, storage, or other IT infrastructure components are rented for a fee based on the quantity of resources used and how long they're in use. Although this is true, there are some important differences between this rental arrangement and the traditional rental models you may be familiar with. For example, when you purchase server and storage resources using IaaS services, you gain immediate virtual access to the resources you need. You are not, however, renting the actual physical servers or other infrastructure. The physical components stay put in the infrastructure service provider's data center. This concept of renting is an essential element of cloud computing, and it provides the foundation for the cost and scalability benefits of the various cloud models.

Within a private IaaS model, renting takes on a different focus. Although you might not charge each user to access a resource, in the *charge-back* model you can allocate usage fees to an individual department based on usage over a week, month, or year. Because of the flexibility of the IaaS model, you can charge more of the budget to heavy users.

2.5 Licensing

The use of public IaaS has led to innovation in licensing and payment models for software you want to run in these cloud environments. Note that this licensing is for the software you want to run in your cloud environment, not the license between you and the cloud provider. For example, some IaaS and software providers have created a "*bring your own license*" (BYOL) plan so you have a way to use your software license in both traditional and cloud environments. Another option is called "*pay as you go*" (PAYG), which generally integrates the software licenses with the on-demand infrastructure services. For example, say that you are running Microsoft Windows Server and using the PAYG route. If you are paying ten cents an hour for cloud access, a few cents of that fee might be going to Microsoft.

2.6 Metering and Costs

To ensure that users are charged for the resources they request and use, IaaS providers need a consistent and predictable way to measure usage. This process of controlling, measuring, and monitoring the use of compute, network, and storage resources is called *metering*. Ideally, the IaaS provider will have a transparent process for identifying charges incurred by the user. With multiple users accessing resources from the same environment, the IaaS provider needs an accurate method for measuring the physical use of resources to make sure each customer is charged the right amount.

IaaS providers often use the metering process to charge users based on the instance of computing consumed. An instance is defined as the CPU power and the memory and storage space consumed in an hour. When an instance is initiated, charges begin to accumulate on an hourly basis until the instance is terminated. Typically, users can make multiple choices during set up, including the amount of compute power, memory and storage requirements of an instance. Some vendors offer instances that come in various sizes, ranging from small to large. The charge for a very small instance may be as little as two cents an hour; the hourly fee could increase to $2.60 for a large resource-intensive instance running Windows.

Metering to assess the charges for the IaaS services you request begins when the instance is initiated and ends when the instance is terminated. At this point, the virtual machine provisioned for you is removed, and you no longer receive charges. Until this point, the charges apply whether the resources are fully used or are lying idle. In addition to the basic per-instance charge, your IaaS provider may include charges such as Storage, Data transfer, Optional services, Pay as you go, Reserved pricing and Trial use.

2.7 Self-service provisioning

The banking ATM service is a great example of the business value of self-service. Without the availability of the self-service ATM, banks would be required to use costly resources to manage activities of all their customers – even for the most repetitive tasks. With an ATM, repetitive tasks can be handled easily with a self-service interface. The customer makes a direct request to perform routine transactions that conform to predefined business rules.

For example, a customer must have an account to withdraw money. In addition, the customer can't take out more money than is in her account. There may be rules dictating how much money an individual can withdraw from the ATM at one time or in one

day. This process is precisely how self-service works in the IaaS public cloud environment.

In a private cloud environment, management can enable users to provision resources when they need them based on a set of predefined rules and business priorities. In this way, everyone is satisfied. The business also gets to control expenses and reduce capital expenditures. The business units have the freedom to avoid time-consuming processes that slow down the ability to get the job done.

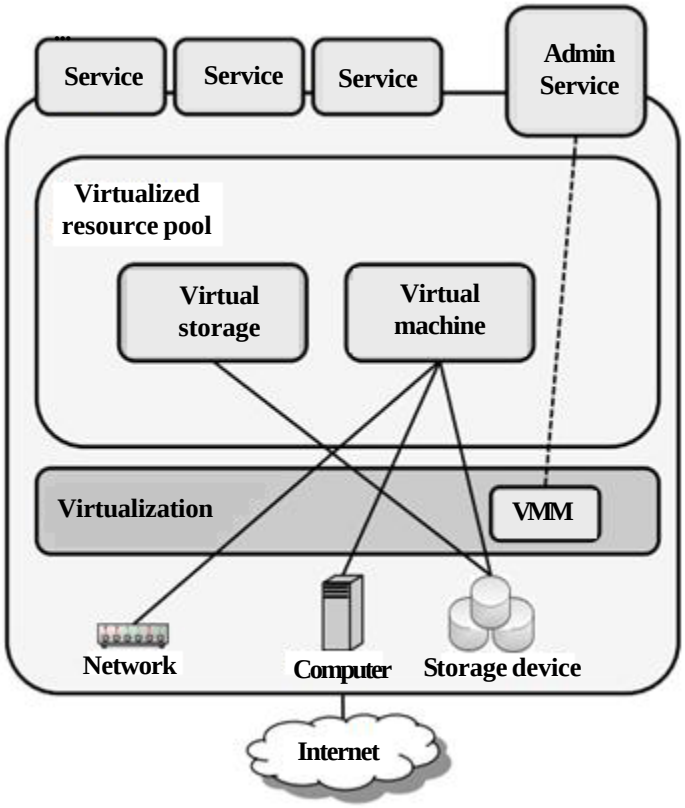

Figure 4. Infrastructure-as-a-Service (IaaS) Stack

Many organizations that leverage IaaS opt for a hybrid approach – using private services in combination with public IaaS services. This approach is attractive because a company can leverage its private cloud resources but use trusted public cloud services to manage peak loads. When used in a controlled way, this hybrid approach is effective. Control means that a company establishes rules for when and how business units can use an outside cloud service; thus, the company is better able to control costs. In addition, by implementing distinct usage rules, users can be prevented from storing sensitive data on a public cloud.

2.8 Considering the Role of a Private IaaS

Why would a company choose to implement a private version of IaaS? There are two compelling reasons:

- The company sees itself as a service provider to its employees and partners.
- The company needs to control access because of security concerns.

Another benefit of a private cloud service is overall manageability. One of the overarching benefits of IaaS is the ability to gain access to an image or copy of a set of resources via a self-service process.

Although a public cloud vendor uses the self-service portal to ensure that the customer pays for the service, IaaS can be used differently in the private cloud. When a developer provisions a resource from a private cloud, IT management makes sure that when a project is completed, the resource is returned to the pool of resources. This way, IT can better control not only the resources available for projects, but also costs, utilization, and security.

2.9 Using IaaS

Organizations use IaaS for a variety of needs. Some companies want to leverage IaaS to quickly and economically provision

additional compute power, whereas others want to avoid costly and complex maintenance of servers. Some companies find that using IaaS for disaster recovery is highly cost-effective and enables the company to mirror its entire site.

3. Exploring Software as a Service

Software as a Service is the way many business users are first introduced to the cloud, because SaaS applications are designed to implement a specific business process – ranging from accounting to customer relationship management (CRM), collaboration, and human resource management. These business users have increasingly found that SaaS applications represent a more cost-effective, flexible, and secure alternative to traditional on-premises applications.

SaaS is not a stand-alone environment. Instead, these applications and services are frequently used in combination with lots of other cloud and on-premises models. Companies need their SaaS applications to couple with other applications and platforms on their own data center and with other cloud platforms.

3.1 Characteristics of SaaS

• Generalized applications that incorporate the right mix of common services so the applications meet the needs of different companies across multiple industries.

• True multi-tenancy means that all the users are using the same codebase. However, their configurations and data are stored in separate containers.

• Highly elastic SaaS applications can easily scale up or down to support changing needs of a business.

• Self-service puts the power to acquire more resources and generally manage an application's deployment directly in the hands of users. Customers can go to their portal and add more licenses, renew their contract, or reduce their number of licenses in a matter of minutes. Self-service includes built-in billing, monitoring, and usage information that give customers a unified view of what they are paying and what they are receiving.

• SaaS applications are modular and service-oriented.

• SaaS applications provide sophisticated business process configurations for customers. Each customer can change the process within the standardized SaaS application.

• SaaS applications need to constantly provide fast releases of new features and new capabilities.

3.2 Multi-Tenancy and Its Benefits

Multi-tenancy means that the SaaS vendor provides a single version of its software for all its customers. This differs from a single-tenant *hosted* solution where the application is housed on a vendor's server but the codebase is unique for each customer.

Although all users of the software access the same foundational components, the data and configurations that are specific to a customer are stored in a separate and secure container. Users can access all the capabilities of the software, but what is not shared is their data. The advantages of a multi-tenancy SaaS over a third-party-hosted, single-tenancy application includes,

• Lower costs through economies of scale „h Shared infrastructure leads to lower costs

• Ongoing maintenance and updates

• Configuration can be done while leaving the underlying codebase unchanged

• Vendors have a vested interest in making sure everything runs smoothly

The SaaS environment too runs on various cloud deployment models (such as private or public) that has various cloud delivery models (such as IaaS or PaaS) supporting it. SaaS on its own is a truly integrated environment that needs to be able to live cooperatively within its own computing environment as well as with other elements in a company's cloud ecosystem. That's what makes a hybrid cloud so complex.

3.3 Understanding the Hybrid SaaS Ecosystem

In order to create a more feature-rich application, some SaaS vendors have created an *ecosystem*. This is a set of partners that works directly with the vendor, both in technical and go-to-market terms. A SaaS vendor with thousands of paying customers

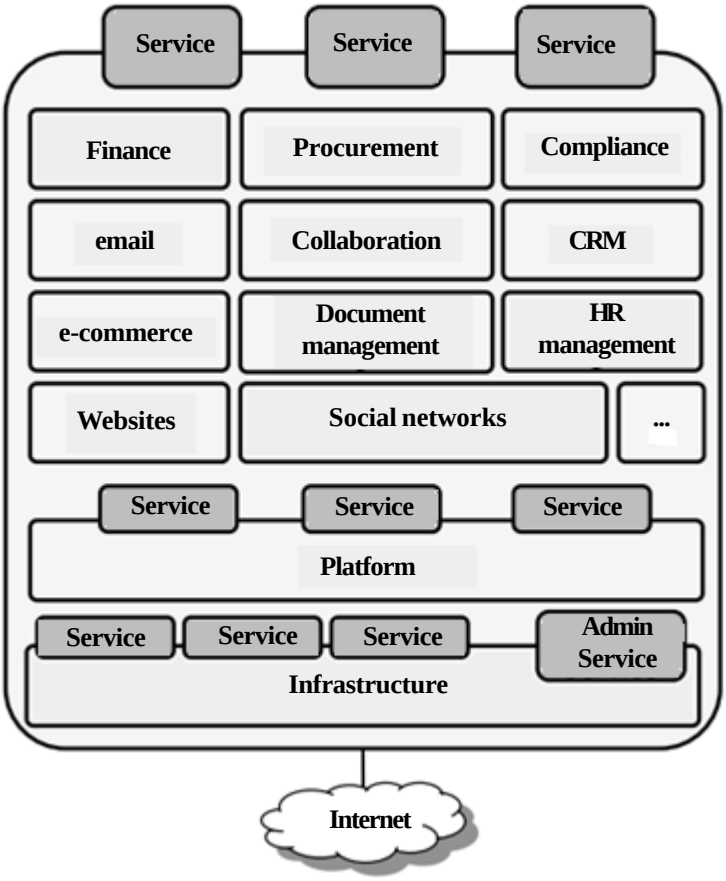

Figure 5. Software-as-a-Service (SaaS) Stack

opens its programming interfaces to other ISVs. These ISVs can then build on top of the SaaS vendor's infrastructure. Therefore, they don't need to write and deploy an entire application, but can focus on their industry-specific code. Messaging middleware, business process services, and other complex programming have already been taken care of by the SaaS vendor that created the ecosystem.

3.4 Using SaaS

SaaS is popular for collaborative applications.This area is dominated by software that focuses on all sorts of collaborative efforts, including web conferencing, document collaboration, project planning, instant messaging and even e-mail. In a sense, it was inevitable that these platforms would move to the cloud: These tasks occur throughout the organization and need to be easily accessed from many locations.

4. Exploring Platform as a Service

Platform as a Service (*PaaS*) is an environment that abstracts the details and complexity of the development process. PaaS offers the potential for much more flexibility that enables developers to more quickly and efficiently meet customer demands.

4.1 Understanding Platform as a Service

PaaS is an abstracted and integrated cloud-based environment that supports the development, running, and management of applications. Application components may exist in a cloud environment or may integrate with applications managed in private clouds or in data centers. A primary value of PaaS environment is that developers don't have to be concerned with some of the lower-level details of the environment. We expect that most organizations will use a combination of some PaaS with traditional on-premises development environments.

Services integrated in PaaS include middleware, operating systems, and development and deployment services to support

software development and delivery organizations. There are also some large enterprises that become a PaaS provider to their own internal developers. In all situations, the goal of the PaaS provider is to create an abstracted environment that supports an efficient, cost-effective, and repeatable process for the creation and deployment of high-quality applications. These applications are designed to be implemented in public or private cloud environments. PaaS enables an organization to do the following:

• Leverage key middleware services without having to deal with the complexities of managing individual hardware and software elements.

• Access a complete stack of development and deployment tools via a web browser, a middleware environment where APIs can be used to plug into selected development and deployment tools. A developer might also leverage a full desktop development environment.

• Overcome the challenges of managing lots of individual development and deployment tools by providing a suite of integrated and standardized tools – operating systems, security products, and the like – that meet company requirements.

Platform as a Service (PaaS) can be viewed as having two fundamental parts: the platform (the actual software that is delivered to your organization) and the service (what really sets PaaS apart). One of the decisions you need to make when beginning to use a PaaS is whether you want to maintain the software or if you want the vendor to be the administrator. If you choose to maintain the software yourself, you must set up, configure, maintain, and administer the PaaS yourself (either on a public or private cloud). Alternatively, you can allow the vendor to provide services that handle these tasks. The result is reduced friction between the development and deployment teams.

4.2 Managing middleware complexity

Middleware is the software that allows independent software components to communicate with each other. Middleware enables application services to exchange data, implement business rules and processes, and manage transactions in an orderly and secure manner. Without middleware, each program would sit in isolation and not be able to share data or other messages.

4.3 Discovering what changes with PaaS

As complexity increases and as applications are becoming increasingly dynamic, PaaS offers the potential to simplify the world

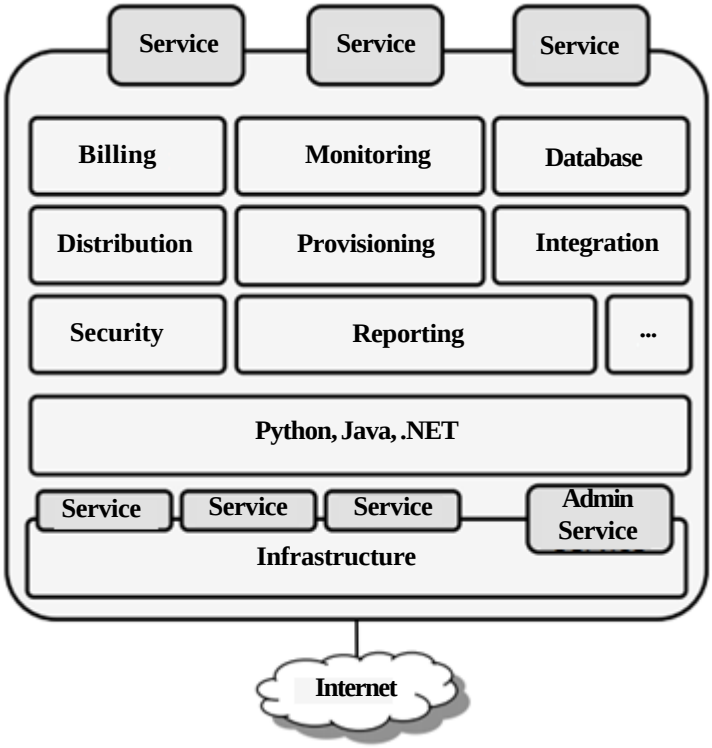

Figure 6. Platform-as-a-Service (PaaS) Stack

of software development and deployment. PaaS helps reduce the complexity of the development process by

• Delivering and managing a set of abstracted middleware services that are either hosted in a public PaaS environment or managed in a private cloud environment

• Ensuring that one standardized set of services is used by those developing and deploying the applications

• Eliminating the installation and operational burden from an organization

4.4 Easing service provisioning

A PaaS platform provides a simpler way to provision development services, including build, test and repository services to help eliminate bottlenecks associated with nonstandard environments. These development services improve efficiency, reduce errors, and ensure consistency in the management of the development lifecycle. Additionally, PaaS makes provisioning of run-time services including application runtime, containers for staging, running, and scaling applications.

4.5 Minimizing friction with IT

Traditionally, when a new application server or other middleware is introduced into an organization, IT must make sure that the middleware can access other services that are required to run that application. This requirement can create friction between the software development and operations teams.

With PaaS, these conflicts are minimized. Because the PaaS environment is designed in a modular, service-oriented manner, components can be easily and automatically updated. When PaaS services are provided by a third-party organization, those changes are automatically handled without the user having to deal with the details. When PaaS is implemented in a private cloud, the IT organization has the responsibility of using a self-service interface to provision the new services to the IT organization.

4.6 Improving the development lifecycle

PaaS changes the way that development and operations interact with resources. Although development and deployment personnel in a traditional development environment work in isolation from each other, the PaaS environment creates a single unified environment. This unification promotes visibility and accountability across the whole IT organization.

With a PaaS platform, the team gets visibility into the *entire* development and deployment lifecycle. The team can see whether software is working, broken, or ready to be released to manufacturing, staged, and so on across the entire application lifecycle.

4.7 Streamlining development and deployment : changing the application lifecycle

In PaaS, support for the application lifecycle, from development through testing, staging, and deployment, is integrated into the platform. The development and the runtime platform are provided as services, so the formal lifecycle of an application is more directly supported.

Paas integration services generally include connectors, adaptors, and templates for integrating applications in cloud environments. These integration services need to cover various integration scenarios, including Connectivity to clouds, Connectivity between clouds, and Connectivity in clouds.

4.8 Using PaaS

One important benefit of PaaS is that it enables individuals developing applications and individuals deploying those applications to use the same services on the same platform. This approach takes away much of the misunderstanding that happens when two teams with different responsibilities are not in sync. By encompassing both development and runtime services, PaaS can streamline the application lifecycle.

5. Exploring IaaS, SaaS and PaaS in Hybrid Environment

In hybrid environments, companies will find a myriad of uses for IaaS (raw compute resources). Examples: Amazon EC2, and Rackspace Cloud. A wise user will evaluate each opportunity to use IaaS against the risk level associated with the IaaS provider and the overall economic risks and benefits.

SaaS provides incredible value to customers, but in practice, SaaS is another element in a hybrid cloud world (applications). Examples: Salesforce.com, QuickBooks online and WorkDay. Even if you plan to simply use a SaaS service to fill one business

need, we expect that you will have more and more of a need to connect systems. SaaS can be complex element in a hybrid cloud. As the market evolves, more SaaS providers will want their applications to work together. Companies will need their SaaS applications to be able to communicate with the applications they have on-premises.

In a hybrid environment, when the same PaaS environment can support both public and private services, organizations can benefit by this level of flexibility and agility (middleware, scaling and operating system). Examples: Google App Engine, Force.com, Microsoft Azure and Cloudbees. By providing a homogeneous platform, workloads can easily be moved from a private cloud to a public cloud for deployment and efficient scaling. This allows organizations to have a high degree of control on where a particular application is running. Some of the business benefits of PaaS include reduced costs and increased speed of development and deployment.

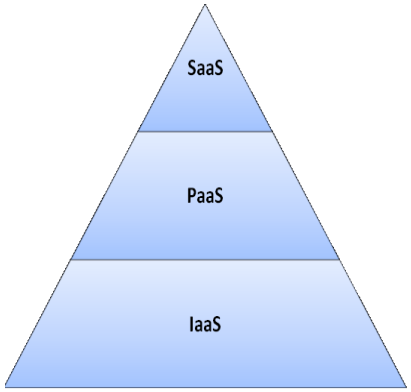

Figure 7. The software stack as a pyramid

6. Conclusion

SaaS provides complete applications to a cloud's end user. It is mainly accessed through a web portal and service oriented architectures based on web service technologies. PaaS comprises the environment for developing and provisioning cloud applications. The principal users of this layer are developers seeking to develop and run a cloud application for a particular platform. The services on the infrastructure layer are used to access essential IT resources that are combined under the heading IaaS. These essential IT resources include services linked to computing resources, data storage resources, and the communications channel. Thus, an exploration of the cloud service delivery models in hybrid approach outcomes the benefits such as High availability, automatic failover & resource balancing, network security, hybridize with physical servers and cost effective.

7. Acknowledgement

I would like to take this privilege to thank all those who have contributed a lot to the success of this paper presentation. First of all, my sincere salute to The Almighty who awarded me such a wonderful opportunity in my life. And this presentation is lovingly dedicated to my beloved husband – Mr. G. Sivakumar, M.B.A., for his invaluable help, guidance, encouragement and remarkable patience.

References

[1] Judith Hurwitz, Marcia Kaufman, Fern Halper and Daniel Kirsch. (2012). Hybrid Cloud for Dummies, John Willey & Sons, Inc., New Jersey.

[2] http://www.cloud-competence-center.com/understanding/cloud-computing-service-models/

[3] http://www.clouditeration.com/what-are-cloud-service-delivery-models

[4] http://www.firstnet-solutions.com/what-is-cloud-computing/index.php

[5] http://www.firstnet-solutions.com/cloud-service-delivery-models-overview/index.php

[6] http://en.wikipedia.org/wiki/Cloud_computing#Network_as_a_service_.28NaaS.29