Cloud Computing Across Different Platforms

Supervisor:

Hammad Ali

Prepared by:

Israt Karim (ID: 080201011)

Sumaiya Sarwar (ID:09301025)

BRAC University, Dhaka, Bangladesh

Table of Contents

Abstract:

In this project we have studied fundamentals of cloud computing. We have studied architectures of different Cloud solutions. We have covered major advantages and disadvantages of Cloud Computing. We have also covered challenges of setting up a cloud server. Finally we have discussed the advantages of implementing a Cloud solution in BRAC university.

1.Introduction:

Now a day's cloud computing is a very famous research topic. It involves networking in different layers. It has several interesting area to study and implement. Moreover there are different researches challenges involve in it. Day by day the demand of cloud server and its applications are increasing. There are some challenges which are not resolved yet making this area not growing in that way as it supposed to be. So we took our personal interest to look into these challenges like data security, data integrity, data locality, scalability which we think are the main hindrances in the growth of cloud computing.

In this report we give an overview of cloud computing. Then we discuss the categories of cloud computing. Then we describe the basic architecture for Cloud Computing solution. As an example we go through the basic architecture of Eucalyptus and Nimbus which are Cloud solutions. We discuss some security issues in Cloud Computing and go through possible solutions. In the end we discuss the advantages of using Cloud Computing in BRAC University.

2.Background:

2.1 Overview of cloud computing:

There are different definition of Cloud Computing but here we adopt the definition of cloud computing provided by The National Institute of Standards and Technology (NIST) , as it covers, in our opinion, all the essential aspects of cloud computing.

NIST definition of cloud computing: *"Cloud computing is a model for enabling convenient*, *on-demand network access to a shared pool of configurable computing* *resources* (*e.g*., *networks*, *servers*, *storage*, *applications*, *and services*) *that can be rapidly provisioned and released with minimal management effort or service provider interaction*."

The concept itself has been around since the 1960s and has been boosted in recent years. Various factors have contributed to this such as the increased availability of broadband internet, improved technologies such as virtualization and new models to deliver web-based services.

2.2 Characteristics:

Cloud computing has the following main characteristics:

- Multi-tenancy – IT resources are shared between different users and customers
- Rented service delivery model – customers pay for the service instead of buying software licenses and hardware
- On-demand usage/flexibility – cloud services can be used almost instantly and can easily be scaled up and down
- External data storage – a customers' data is usually stored externally at the location of the cloud computing vendor

2.3 Coming of cloud:

In a sense, what we're seeing now is the second coming of cloud computing. The history is something like this-

Almost 50 years ago a similar transformation came with the creation of service bureaus and time-sharing systems that provided access to computing machinery for users who lacked a mainframe in a glass-walled room down the hall. A typical time-sharing service had a hub-and-spoke configuration.

When personal computers arrived in the 1980s, part of their appeal was the promise of "liberating" programs and data from the central computing center.

Individuals were free to control their own computing environment, choosing software to suit their needs and customizing systems to their tastes.

The client-server model introduced in the 1980s offered a central repository for shared data while personal computers and workstations replaced terminals, allowing individuals to run programs locally.

In the current trend, the locus of computation is shifting again, with functions migrating outward to distant data centers reached through the Internet. Although the new model of Internet computing has neither hub nor spokes, it still has a core and a fringe. The aim is to concentrate computation and storage in the core, where high performance machines are linked by high-bandwidth connections and all of these resources are carefully managed. At the fringe are the end users making the requests that initiate computations and who receive the results.

Examples:

- The Google Docs programs are an example, including a word processor, a spreadsheet, and a tool for creating PowerPoint-like presentations.
- Another recent Adobe product is Photoshop Express, which has turned the well-known image-manipulation program into an online service.
- *Enterprise computing in the cloud.* Software for major business applications has generally been run on corporate servers, but several companies now provide it as an on-demand service.
- *Cloudy infrastructure.* It's all very well to outsource the chore of building and maintaining a data center, but someone must still supply that infrastructure.
- *The cloud OS.* For most cloud-computing applications, the entire user interface resides inside a single window in a Web browser.

3. System description:

For those deploying software out in the cloud, scalability is a major issue—the need to marshal resources in such a way that a program continues running smoothly even as the number of users grows.

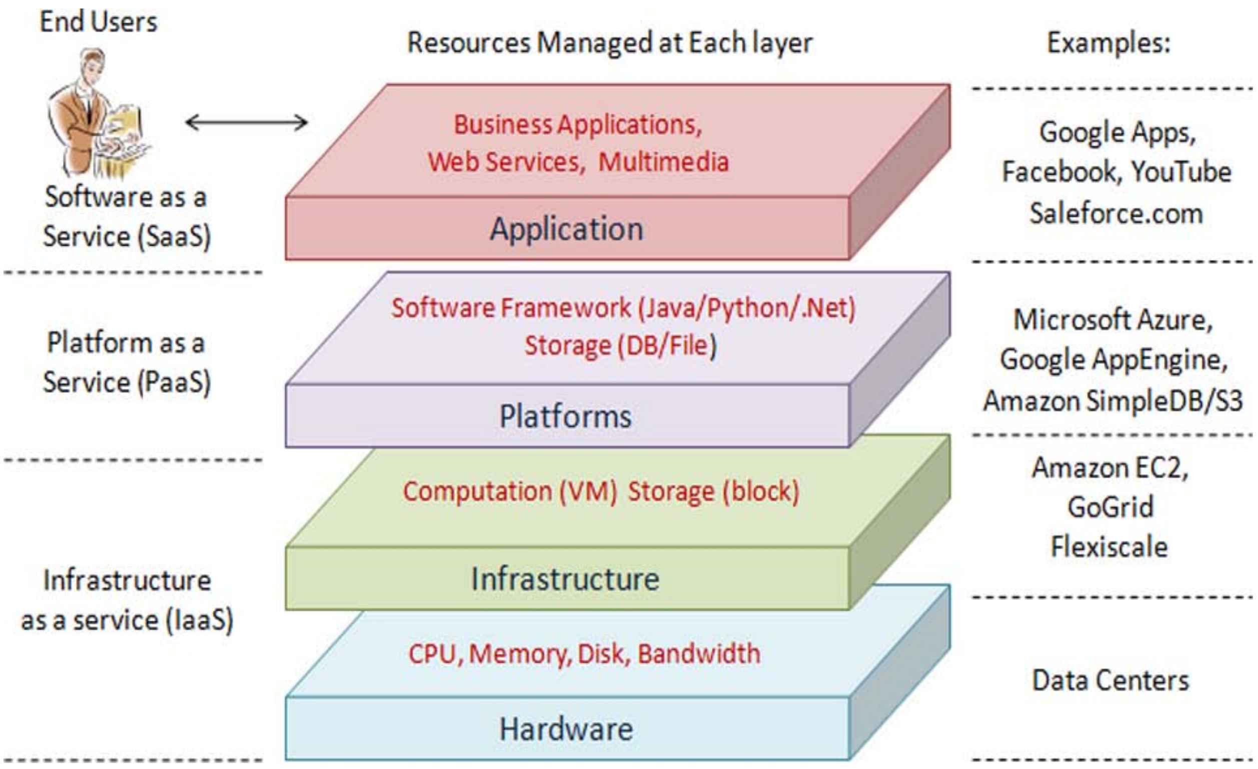

End Users
Resources Managed at Each layer
Examples:
Software as a Service (SaaS)
Business Applications, Web Services, Multimedia
Application
Google Apps, Facebook, YouTube Saleforce.com
Platform as a Service (PaaS)
Software Framework (Java/Python/.Net) Storage (DB/File)
Platforms
Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
Computation (VM) Storage (block)
Infrastructure
Amazon EC2, GoGrid Flexiscale
Infrastructure as a service (IaaS)
CPU, Memory, Disk, Bandwidth
Hardware
Data Centers

Fig1: cloud computing architecture.

Cloud computing employs a service-driven business model. Clouds offer services that can be grouped into three categories: software as a service (SaaS), platform as a service (PaaS), and infrastructures a service (IaaS).

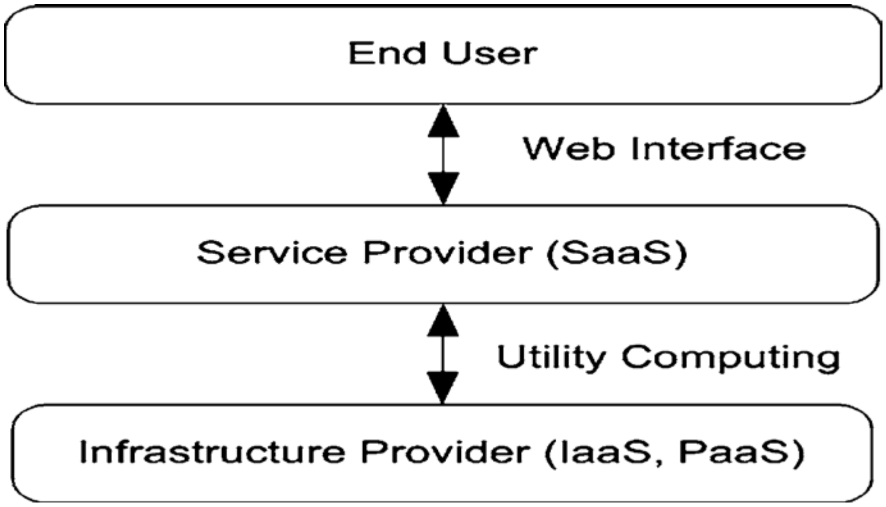


Fig2: Business model of cloud computing

4. Types of cloud:

There are different types of clouds, each with its own benefits and drawbacks:

Public clouds: A cloud in which service providers offer their resources as services to the general public.

Private clouds: Also known as internal clouds, private clouds are designed for exclusive use by a single organization.

Hybrid clouds: A hybrid cloud is a combination of public and private cloud models that tries to address the limitations of each approach.

Virtual Private Cloud: An alternative solution to addressing the limitations of both public and private clouds is called Virtual Private Cloud (VPC).

For most service providers, selecting the right cloud model is dependent on the business scenario. For example, Computation-intensive scientific applications are best deployed on public clouds for cost-effectiveness.

Data centers plays vital role in cloud computing-

5. Basic layer design of data center network infrastructure:

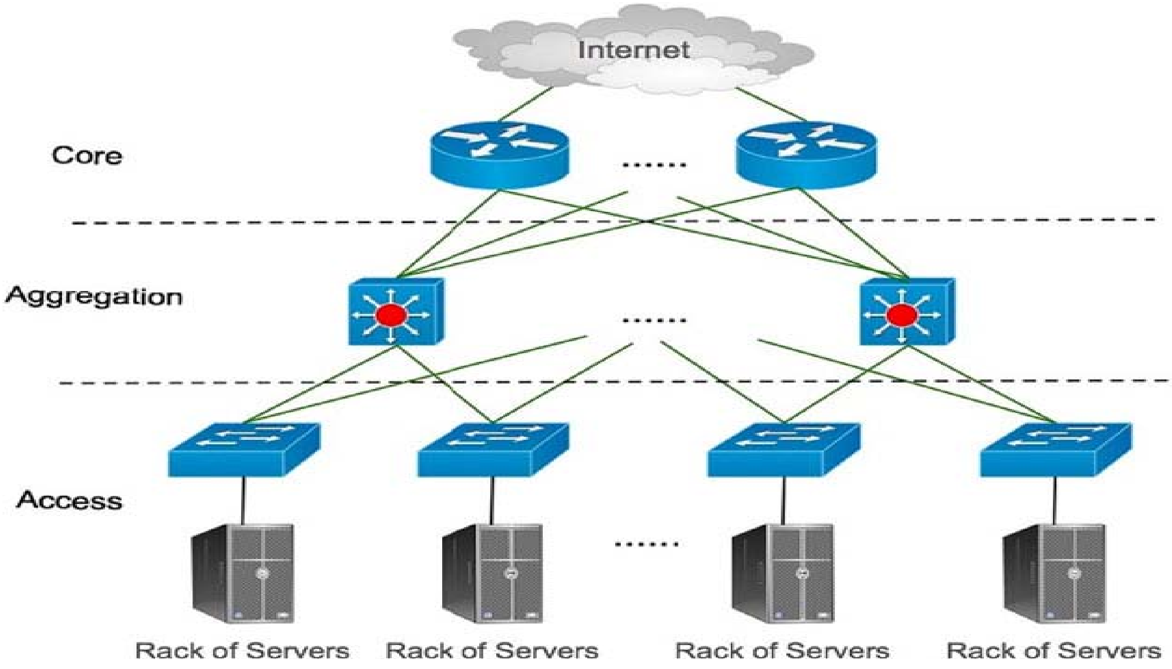


Fig3: Basic layered design of data center network infrastructure

Some of the dominant cloud computing products are Amazon EC2, Microsoft Windows Azure platform etc.

6. Features of Cloud Computing:

Cloud computing provides several salient features that are different from traditional service computing, which we summarize below:

Multi-tenancy: In a cloud environment, services owned by multiple providers are co-located in a single data center.

Shared resource pooling: The infrastructure provider offers a pool of computing resources that can be dynamically assigned to multiple resource consumers.

Geo-distribution and ubiquitous network access: Clouds are generally accessible through the Internet and use the Internet as a service delivery network.

Dynamic resource provisioning: One of the key features of cloud computing is that computing resources can be obtained and released on the fly.

7. Compounded Challenges

Cloud computing have many open issues. We wonder if the reciprocal definitions of these two paradigms also suggest that the challenges of one might serve as an opportunity for the other, like Maintaining High Service Availability, Providing End-to-End Secure Solutions, Managing Longer-Standing Service Workflows

Now that we've looked at some of the challenges, what are some of the ways we could combine each paradigm's strengths to help neutralize the other's weaknesses? We can see at least three opportunities: service discovery through Federated Clouds, rapid service deployment, agent-Mediated Ontology Generation from Co-Located Information.

8. Clouds, Grids, and Distributed Systems:

For an overview of the relationship between Clouds and other domains that it overlaps with. Web 2.0 covers almost the whole spectrum of service-oriented applications, where Cloud Computing lies at the large-scale side. Supercomputing and Cluster Computing have been more focused on traditional non-service applications.

Grid Computing overlaps with all these fields where it is generally considered of lesser scale than supercomputers and Clouds.

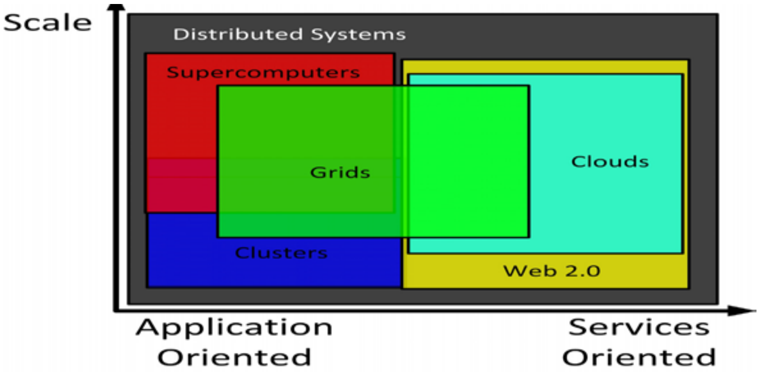


Fig4: **Grids and Clouds Overview**

8.1 Resource Management:

This section describes the resource management found in Grids and Clouds, covering topics such as the compute model, data model, virtualization, monitoring, and provenance. These topics are extremely important to understand the main challenges that both Grids and Clouds face today, and will have to overcome in the future.

8.1.1 Compute Model: Most Grids use a batch-scheduled compute model, in which a local resource manager (LRM), such as PBS, Condor, SGE manages the compute resources for a Grid site, and users submit batch jobs (via GRAM) to request some resources for some time.

8.1.2 Data Model: While some people boldly predicate that future Internet Computing will be towards Cloud Computing centralized, in which storage, computing, and all kind of other resources will mainly be provisioned by the Cloud.

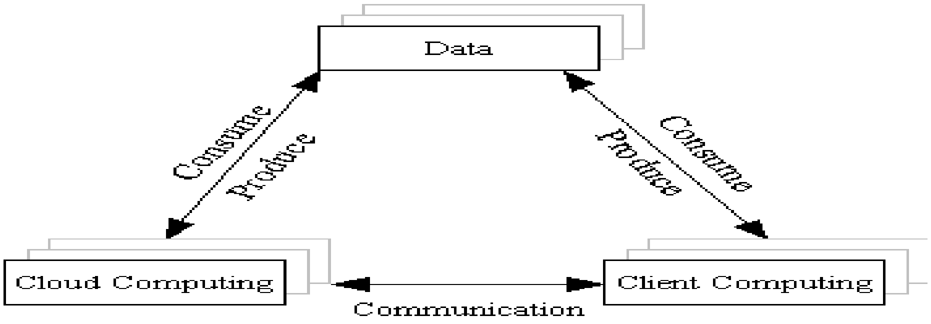


Fig5: The triangle model of next-generation Internet Computing.

8.1.3 Data Locality: As CPU cycles become cheaper and data sets double in size every year, the main challenge for efficient scaling of applications is the location of the data relative to the available computational resources – moving the data repeatedly to distant CPUs is becoming the bottleneck.

8.1.4 Combining compute and data management: Even more critical is the combination of the compute and data resource management, which leverages data locality in access patterns to minimize the amount of data movement and improve end application performance and scalability.

8.1.5 Virtualization: Virtualization has become an indispensable ingredient for almost every Cloud; the most obvious reasons are for abstraction and encapsulation.

8.1.6 Application Model: Grids generally support many different kinds of applications, ranging from high performance computing (HPC) to high throughput computing (HTC).

8.1.7 Security Model: Clouds mostly comprise dedicated data centers belonging to the same organization, and within each data center, hardware and software configurations, and supporting platforms are in general more homogeneous as compared with those in Grid environments.

Here, we can see that Clouds and Grids share a lot commonality in their vision, architecture and technology, but they also differ in various aspects such as security, programming model, business model, compute model, data model, applications, and abstractions. We also identify challenges and opportunities in both fields.

9. Challenges:

The development of cloud computing solutions brings several technical challenges to cloud developers. These challenges can be grouped in three main areas: negotiation, decision, and operation. In the negotiation area, these are the challenges relative to how application developers interface with the cloud as well as the description of the cloud offerings. It includes also the definition of the programmability level that the cloud
solution will offer.

9.1 Negotiation: The negotiation area concerns itself with challenges relative to the interface between the application developers and the cloud. Generally, the interface between the cloud and application developers assumes the form of an Application Programming

Interface (API), but, depending on the programmability level offered by the cloud, this API can be implemented in several ways ranging from a web-service based toolkit to control virtual machines in the cloud to a set of programming primitives used to develop distributed applications in the cloud.

9.2 Decision: The main target of any cloud operator is to schedule developer applications aiming for the maximum utilization of cloud resources. A developer´s application covers, beyond the actual code, some additional information about application´s needs and services negotiated previously.

9.3 Operation: Metaphorically, one can say that while in the decision area the cloud operator must identify solutions for the "brain" of the cloud, in the operation area it must attack the

problem of the "limbs" of the cloud, i.e., they must provide some form to enforce

decisions. The enforcement here covers the communication protocols and the configuration of cloud elements.

9.4 Standardization efforts: A considerable challenge present in many of the raised discussions around the cloud is related to the need for standardization.

10. Open-source solutions for Cloud Computing:

Due to the large growth of cloud computing, there are several solutions in this area. This article is focused on open source solutions, highlighting their main characteristics and architectures proposed.

10.1 Xen Cloud Platform (XCP):

The Xen hypervisor [Citrix Systems 2010b] is a solution for infrastructure virtualizationthat provides an abstraction layer between servers' hardware and the operating system.

A Xen hypervisor allows each physical server to run several "virtual servers" handlingthe operating system and its applications from the underlying physical server. The Xensolution is used by many cloud solutions such as Amazon EC2, Nimbus and Eucalyptus.

10.2 Nimbus:

Nimbus [Keahey 2009] is an open source solution (licensed under the terms of the Apache License) to turn clusters into an Infrastructure as a Service (IaaS) for Cloud Computing focusing mainly on scientific applications.

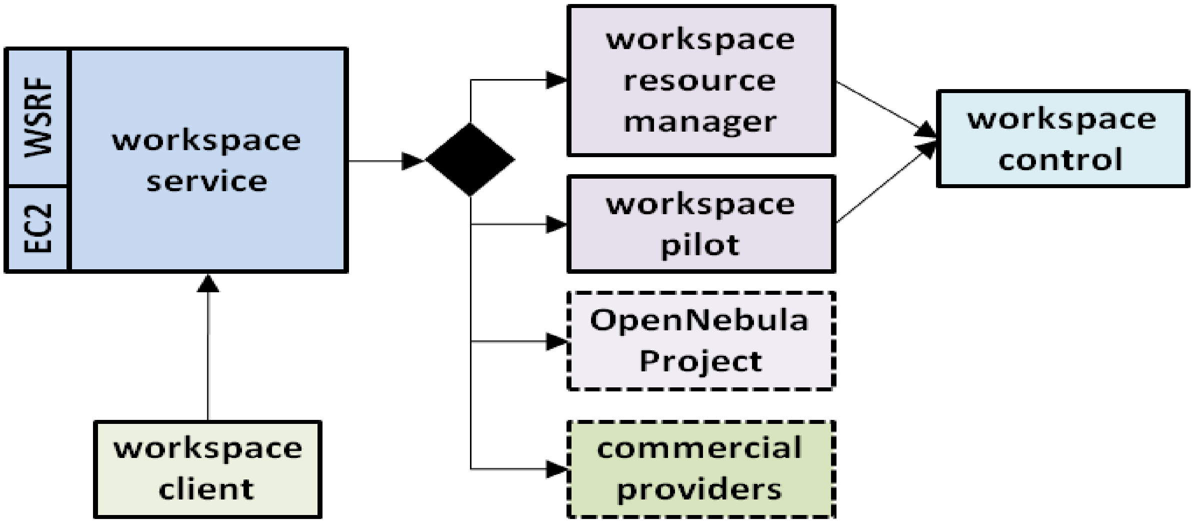


Fig6: Nimbus workspace components

10.3 OpenNebula:

OpenNebula [OpenNebula Project 2010] is an open-source toolkit used to build private, public and hybrid clouds. It has been designed to be integrated with networking and storage solutions and to fit into existing data centers.

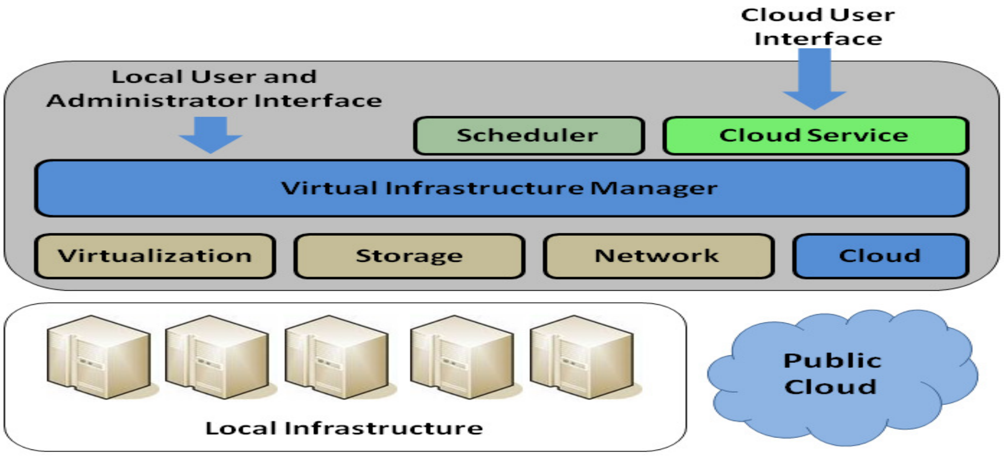


Fig7: Open Nebula architecture

The Cumulus design is a layered architecture with three main entities:

Cumulus frontend, Open Nebula frontend, and OS Farm. This proposal focuses on reaching scalability and autonomy of data centers.

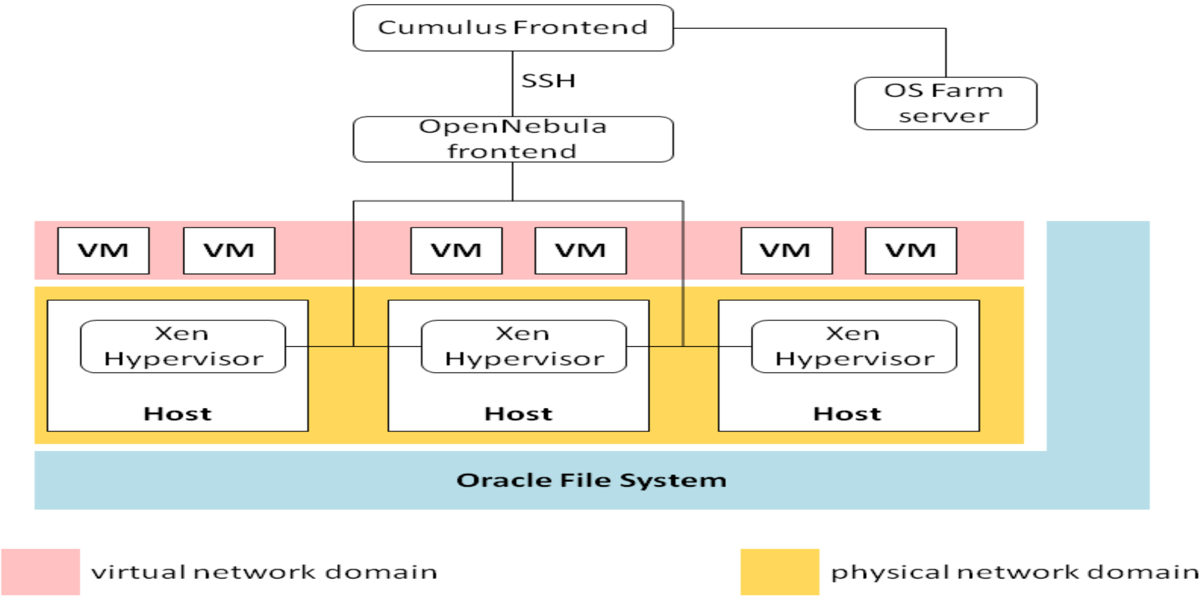


Fig8: Cumulus architecture

10.4 Eucalyptus:

Eucalyptus [Nurmi et al 2009] is an open source cloud computing framework focused on academic research. It provides resources for experimental instrumentation and study. Eucalyptus users are able to start, control, access and terminate entire virtual machines. In its current version, Eucalyptus supports VMs that run atop the Xen supervisor.

10.4.1 EUCALYPTUS Design:

The architecture of the EUCALYPTUS system is simple, flexible and modular with a hierarchical design reflecting common resource environments found in many academic settings. In essence, the system allows users to start, control, access, and terminate entire virtual machines using an emulation of Amazon EC2’s SOAP and “Query” interfaces.

There are four high-level components, each with its own Web-service interface, that comprise a EUCALYPTUS installation:

•***Node Controller*** controls the execution, inspection, and terminating of VM instances on the host where it runs. A Node Controller (NC) executes on every node that is designated for hosting VM instances. An NC queries and controls the system software on its node (i.e., the host operating system and the hypervisor) in response to queries and control requests from its Cluster Controller.

• ***Cluster Controller(CC)*** :gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.

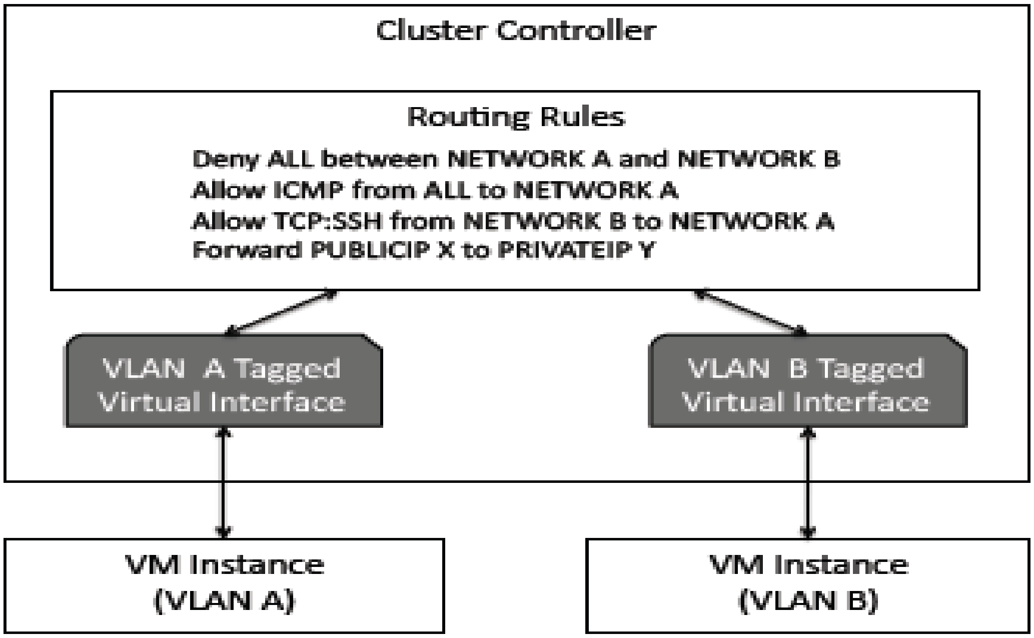

Cluster Controller
Routing Rules
Deny ALL between NETWORK A and NETWORK B
Allow ICMP from ALL to NETWORK A
Allow TCP:SSH from NETWORK B to NETWORK A
Forward PUBLICIP X to PRIVATEIP Y
VLAN A Tagged Virtual Interface
VLAN B Tagged Virtual Interface
VM Instance (VLAN A)
VM Instance (VLAN B)

Fig9: The CC uses the Linux iptables packet filtering system to allow users to define inter-VM network ingress rules, and to assign public IP addresses dynamically at boot or run-time.

• ***Storage Controller (Walrus)*** is a put/get storage service that implements Amazon’s S3 interface, providing a mechanism for storing and accessing virtual machine images and user data.

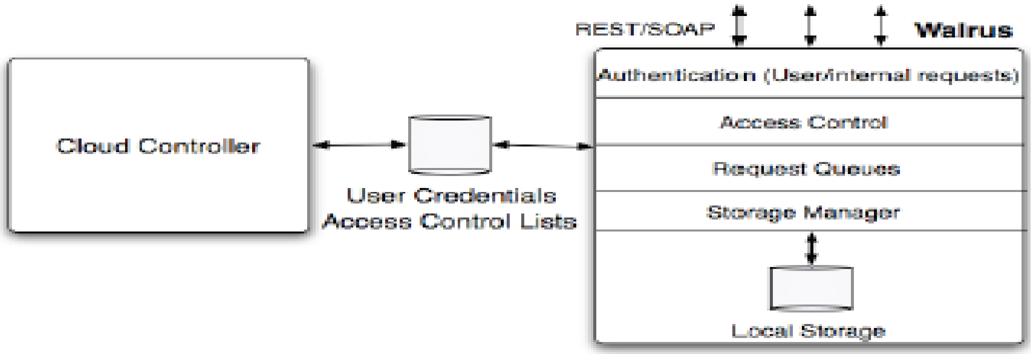


Fig10 : EUCALYPTUS *includes Walrus, a S3 compatible storage management service for storing and accessing user data as well as images.*

• ***Cloud Controller*** is the entry-point into the cloud for users and administrators. It queries node managers for information about resources, makes high level scheduling decisions, and implements them by making requests to cluster controllers.

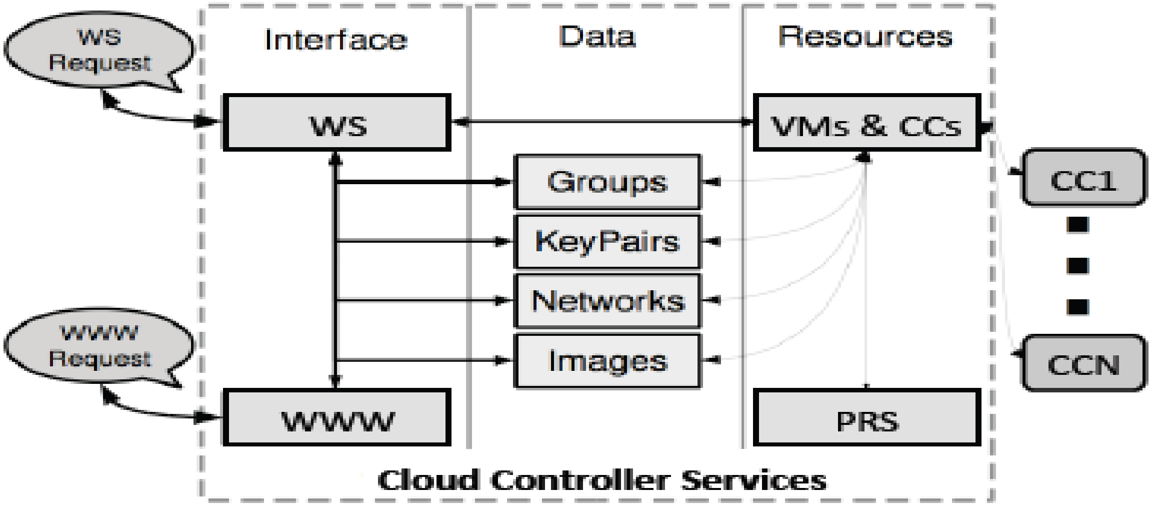


Fig11 : *Overview of Cloud Controller services. Dark lines indicate the flow of user requests while light lines correspond to inter-service system messages.*

The relationships and deployment locations of each component within a typical small cluster setting are shown in the Figure below-.

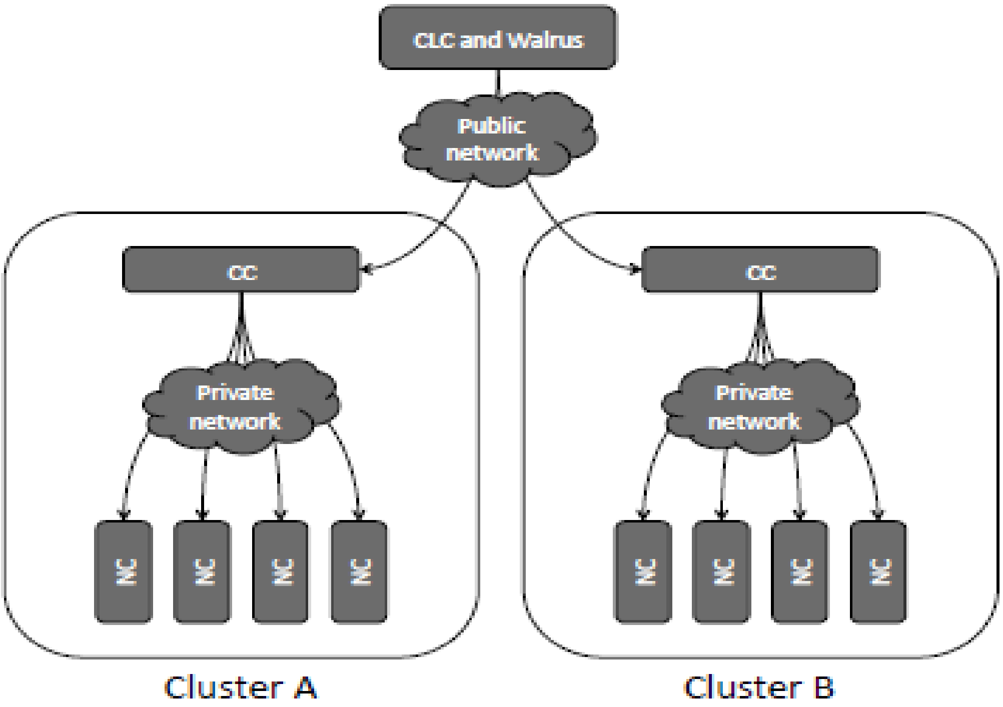


Fig12 : EUCALYPTUS *employs a hierarchical design to reflect underlying resource topologies.*

10.5 TPlatform:

TPlatform [Peng et al 2009] is a cloud solution that provides a development platform for web mining applications, which is inspired in Google cloud technologies, and which acts as a Platform as a Service (PaaS) solution. Their infrastructure is supported by three technologies: a scalable file system called Tianwang File System (TFS) what is similar to the Google File System (GFS), the BigTable data storage mechanism, and the MapReduce programming model. The TPlatform framework is composed by three layers.

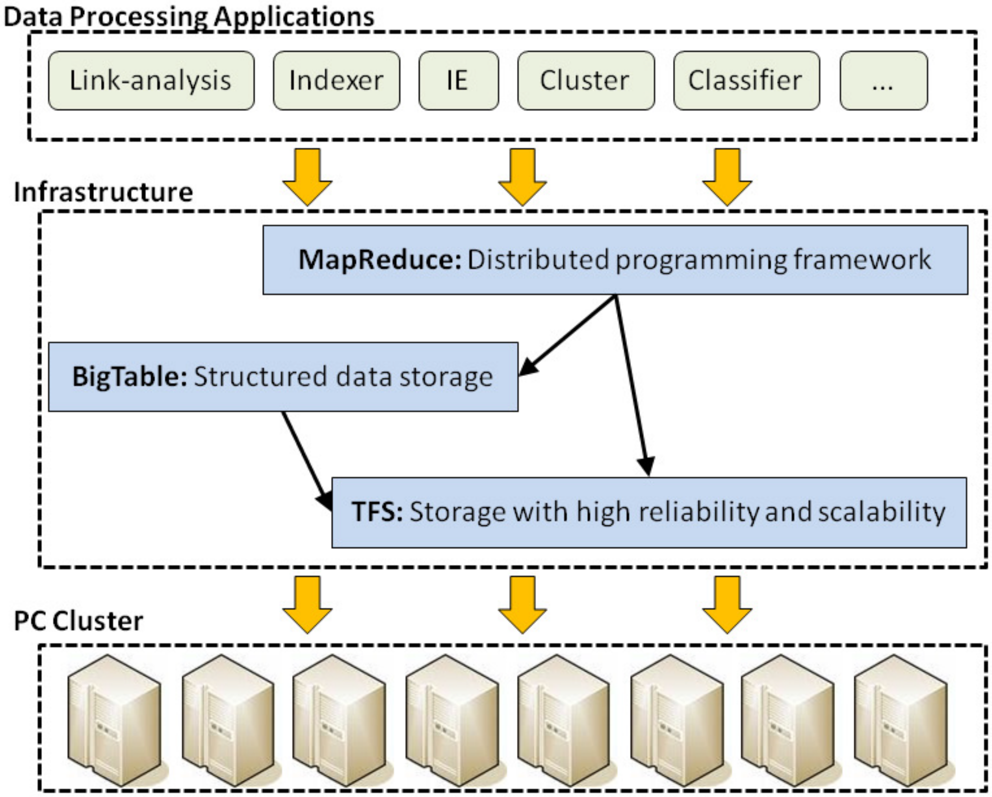

Data Processing Applications
Link-analysis
Indexer
IE
Cluster
Classifier
...
Infrastructure
MapReduce: Distributed programming framework
BigTable: Structured data storage
TFS: Storage with high reliability and scalability
PC Cluster

Fig12: TPlatform framework

10.6: Apache Virtual Computing Lab (VCL):

Apache VCL [VCL 2010] is an open-source solution for the remote access over the Internet to dynamically provision and reserve computational resources for diverse applications, acting as Software as a Service (SaaS) solution. VCL has a simple architecture formed by three tiers:

•**Web server:** represents the VCL portal and uses Linux/Apache/PHP solution. This portal provides an user interface that enable the requesting andmanagement of VCL resources;

•**Database server:** storages information about VCL reservations, access controls,machine and environment inventory. It uses Linux/SQL solution;

•**Management nodes:** is the processing engine. A management node controls asubset of VCL resources, which may be physical blade servers, traditional rack ,or virtual machines. It uses Linux/VCLD (perl)/image library solution. VCLD is a middleware responsible to process reservations or jobs assigned by the VCL web portal.

11. SECURITY ON DEMAND:

Though cloud computing is targeted to provide better utilization of resources using virtualization techniques and to take up much of the work load from the client, it is fraught with security risks.

The complexity of security risks in a complete cloud environment is illustrated in the figure below:

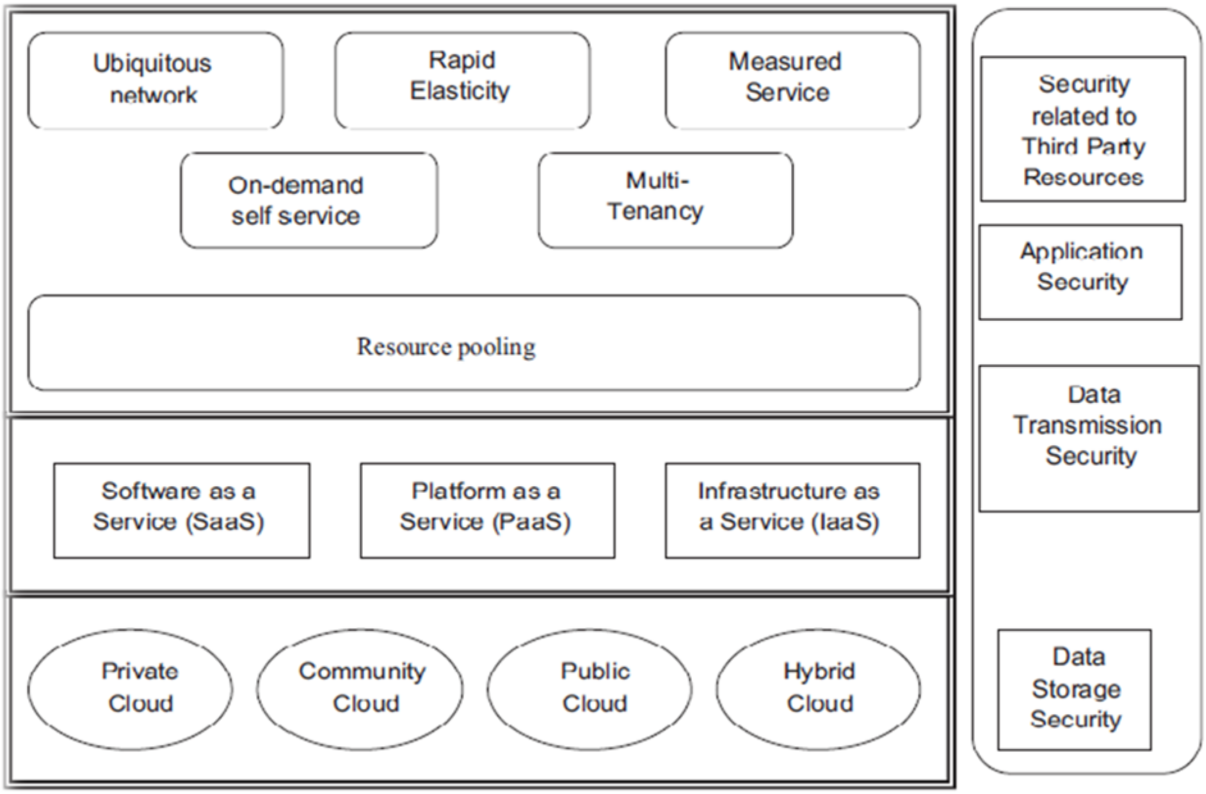


Fig. 1. Complexity of security in cloud environment.

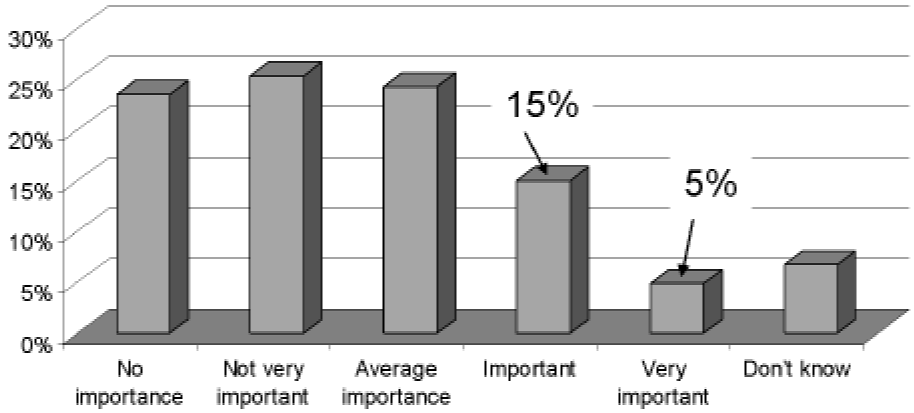
Fig13: Complexity of security in cloud environment.

In the figure, the lower layer represents the different deployment models of the cloud namely private, community, public and hybrid cloud.

The layer just above the deployment layer represents the **different delivery models** that are utilized within a particular deployment model. These delivery models are the SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service) delivery models.

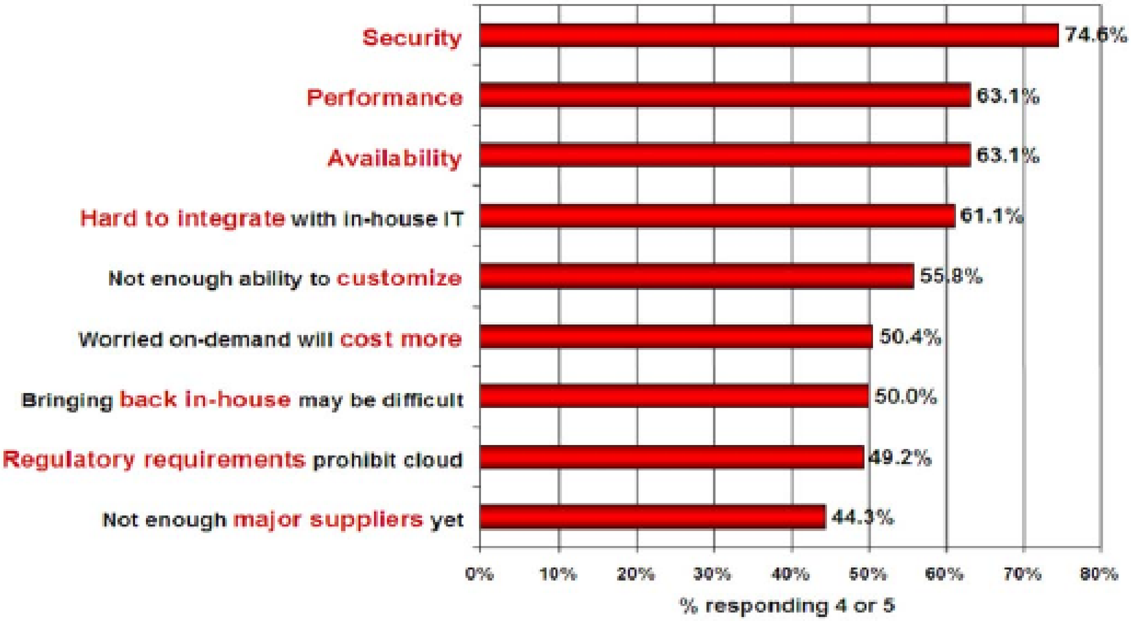
These delivery models form the core of the cloud and they exhibit certain characteristics like on-demand self-service, multi-tenancy, ubiquitous network, measured service and rapid elasticity which are shown in the **top layer**. These fundamental elements of the cloud require security which depends and varies with respect to the deployment model that is used, the way by which it is delivered and the character it exhibits.

more and more users are considering Cloud Computing is important and start to setup applications in the Cloud Computing system or adopt the services provided by it. According to a survey over a large number of firms, which evaluate the importance of using Software as a Service (SaaS) in terms of their points of view, more and more firms are thinking it is important. graph 1 gives it in detail. 15% of the firms view it is important and another 5% of the firms consider it is very important. In the survey [1], it also claims that a typical organization today might have 5 to 15 applications in the Cloud. As Cloud Computing has advantages for both providers and users, it is developing in an amazing pace and predicted to grow and be adopted by a large amount of users in the near future.



Graph 1: Rating the Importance of Using SaaS in Terms of firms' Points of view

Graph 2 shows the nine challenges in detail. Where security is the top one concern.



Source: IDC Enterprise Panel, August 2008 n=244

Graph 2. Rate the Challenges/Issues Ascribed to the Cloud On-demand Model

12. Security issues in service models:

Cloud computing utilizes three delivery models by which different types of services are delivered to the end user. The three delivery models are the SaaS, PaaS and IaaS which provide infrastructure resources, application platform and software as services to the consumer. These service models also place a different level of security requirement in the cloud environment. IaaS is the foundation of all cloud services, with PaaS built upon it and SaaS in turn built upon it.

12.1 Security issues in SaaS:

In SaaS, the client has to depend on the provider for proper security measures. The provider must do the work to keep multiple users’ from seeing each other’s data. So it becomes difficult to the user to ensure that right security measures are in place and also difficult to get assurance that the application will be available when needed.

The layered stack for a typical SaaS vendor and critical aspects that must be covered across layers in order to ensure security of the enterprise data is illustrated in Fig. 2.

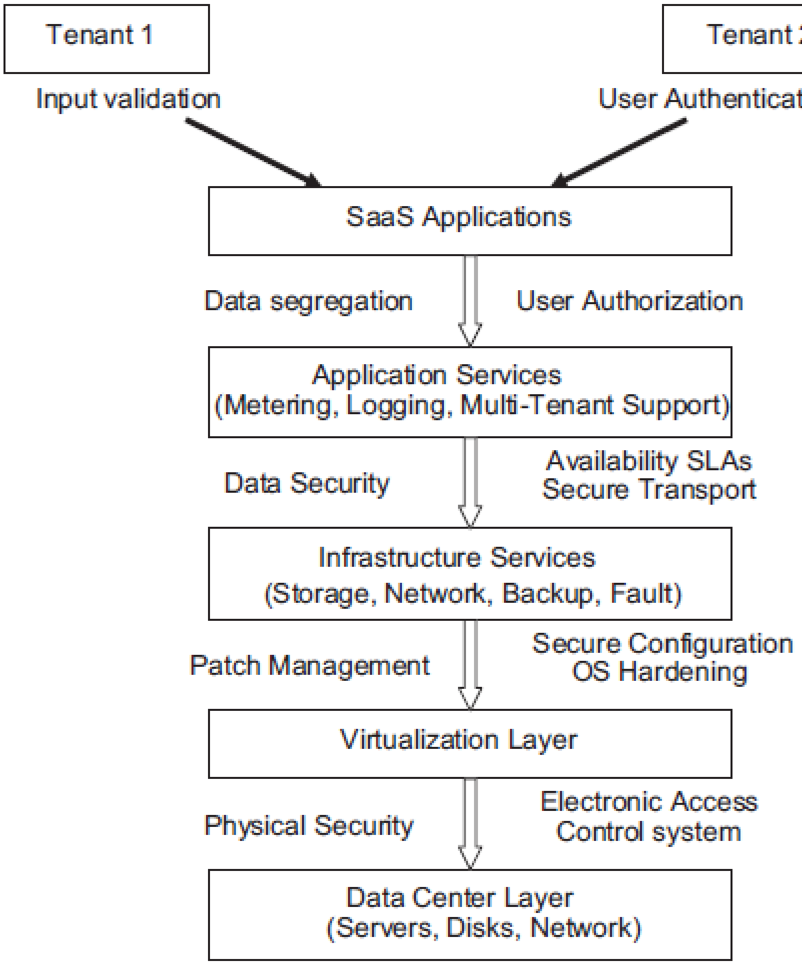


Fig 14: Security for the SaaS stack

The following key security elements should be carefully considered as an integral part of the SaaS application development and deployment process:

- **Data security**: In a traditional on-premise application deployment model, the sensitive data of each enterprise continues to reside within the enterprise boundary and is subject to its physical, logical and personnel security and access control policies. However, in the SaaS model, the enterprise data is stored outside the enterprise boundary, at the SaaS vendor end. Consequently, the SaaS vendor must adopt additional security checks to ensure data security and prevent breaches due to security vulnerabilities in the application or through malicious employees. This involves the use of strong encryption techniques for data security and fine-grained authorization to control access to data.

The following assessments test and validate the security of the enterprise data stored at the SaaS vendor:

1. Cross-site scripting[XSS]
2. Access control weaknesses
3. OS and SQL injection flaws
4. Cross-site request forgery[CSRF]
5. Cookie manipulation
6. Hidden field manipulation
7. Insecure storage
8. Insecure configuration.

- **Network security**: In a SaaS deployment model, sensitive data is obtained from the enterprises, processed by the SaaS application and stored at the SaaS vendor end. All data flow over the network needs to be secured in order to prevent leakage of sensitive information. This involves the use of strong network traffic encryption techniques such as Secure Socket Layer (SSL) and the Transport Layer Security (TLS) for security.

- **Data locality**: In a SaaS model of a cloud environment, the consumers use the applications provided by the SaaS and process their business data. But in this scenario, the customer does not know where the data is getting stored. In many a cases, this can be an issue. Due to compliance and data privacy laws in various countries, locality of data is of utmost importance in many enterprise architecture

- **Data integrity**: Data integrity is one of the most critical elements in any system. Data integrity is easily achieved in a standalone system with a single database. Data integrity in such a system is maintained via database constraints and transactions. Transactions should follow ACID (atomicity, consistency, isolation and durability) properties to ensure data integrity. Most databases support ACID transactions and can preserve data integrity.

- **Data segregation**: Multi-tenancy is one of the major characteristics of cloud computing. As a result of multi-tenancy multiple users can store their data using the applications provided by SaaS. In such a situation, data of various users will reside at the same location. Intrusion of data of one user by another becomes possible in this environment. This intrusion can be done either by hacking through the loop holes in the application or by injecting client code into the SaaS system.

- **Data access:** Data access issue is mainly related to security policies provided to the users while accessing the data. In a typical scenario, a small business organization can use a cloud provided by some other provider for carrying out its business processes. This organization will have its own security policies based on which each employee can have access to a particular set of data.

- **Authentication and authorization**: Most companies, if not all, are storing their employee information in some type of Lightweight Directory Access Protocol (LDAP) servers. In the case of SMB companies, a segment that has the highest SaaS adoption rate, Active Directory (AD) seems to be the most popular tool for managing users.

- **Data confidentiality:** Some of the findings related to the confidentiality issues are:
  1. Cloud computing has significant implications for the privacy of personal information as well as for the confidentiality of business and governmental information.

2. A user's privacy and confidentiality risks vary significantly with the terms of service and privacy policy established by the cloud provider.

3. For some types of information and some categories of cloud computing users, privacy and confidentiality rights, obligations, and status may change when a user discloses information to a cloud provider.

4. Disclosure and remote storage may have adverse consequences for the legal status of protections for personal or business information.

5. The location of information in the cloud may have significant effects on the privacy and confidentiality protections of information and on the privacy obligations of those who process or store the information.

6. Information in the cloud may have more than one legal location at the same time with differing legal consequences.

7. Laws could oblige a cloud provider to examine user records for evidence of criminal activity and other matters.

8. Legal uncertainties make it difficult to assess the status of information in the cloud as well as the privacy and confidentiality protections available to users.

- **Web application security:** Since the web applications and SaaS are tightly coupled in providing services to the cloud users, most of the security threats of web application are also posed by the SaaS model of the cloud. The Open Web Application Security Project has identified Top 10 security risks faced by web applications. Those threats are:

1. Injection flaws like SQL, OS and LDAP injection

2. Cross-site scripting

3. Broken authentication and session management

4. Insecure direct object references

5. Cross-site request forgery

6. Security misconfiguration

7. Insecure cryptographic storage

8. Failure to restrict URL access

9. Insufficient transport layer protection

10. Invalidated redirects and forwards.

- **Data breaches:** Since data from various users and business organizations lie together in a cloud environment, breaching into the cloud environment will potentially attack the data of all the users. Thus the cloud becomes a high value target .

- **Virtualization vulnerability:** Virtualization is one of the main components of a cloud. But this poses major security risks. Ensuring that different instances running on the same physical machine are isolated from each other is a major task of virtualization which is not met completely in today's scenario. The other issue is the control of administrator on host and guest operating systems. Current VMMs (Virtual Machine Monitor) do not offer perfect isolation.

- **Availability:** The SaaS application needs to ensure that enterprises are provided with service around the clock. A multi-tier architecture needs to be adopted, supported by a load-balanced farm of application instances, running on a variable number of servers. At the same time, an appropriate action plan for business continuity (BC) and disaster recovery (DR) needs to be considered for any unplanned emergencies. This is essential to ensure the safety of the enterprise data and minimal downtime for enterprises.

- **Backup:** The SaaS vendor needs to ensure that all sensitive enterprise data is regularly backed up to facilitate quick recovery in case of disasters. Also the use of strong encryption schemes to protect the backup data is recommended to prevent accidental leakage of sensitive information.

- **Identity management and sign-on process**.: Identity management(IdM) or ID management is a broad administrative area that deals with identifying individuals in a system (such as a country, a network or an organization) and controlling the access to the resources in that system by placing restrictions on the established id entities. Identity management can involve three perspectives

1. The pure id entity paradigm: Creation, management and deletion of identities without regard to access or entitlements.

2. The user access (log-on) paradigm: For example: a smartcard and its associated data used by a customer to logon to a service or services (a traditional view).

3. The service paradigm: A system that delivers personalized role- based, online, on-demand, multimedia (content), presence- based services to users and their devices.

12.2 Security issues in PaaS:

In PaaS, the provider might give some control to the people to build applications on top of the platform. But any security below the application level such as host and network intrusion prevention will still be in the scope of the provider and the provider has to offer strong assurances that the data remains inaccessible between applications. PaaS is intended to enable developers to build their own applications on top of the platform. As a result it tends to be more extensible than SaaS, at the expense of customer-ready features. This trade off extends to security features And capabilities, where the built-in capabilities are less complete, but there is more flexibility to layer on additional security.

12.3 Security issues in IaaS:

With IaaS the developer has better control over the security as long as there is no security hole in the virtualization manager. Also, though in theory virtual machines might be able to address these issues but in practice there are plenty of security problems.

The other factor is the reliability of the data that is stored within the provider’s hardware.

The security responsibilities of both the provider and the consumer greatly differ between cloud service models.

13 Current security solutions:

There are several research works happening in the area of cloud security. Several groups and organization are interested in developing security solutions and standards for the cloud. The Cloud Standards web site is collecting and coordinating information about cloud-related standards under development by the groups. The Open Web Application Security Project (OWASP) maintains list of top vulnerabilities to cloud-based or SaaS models which is updated as the threat landscape changes. The best security solution for web applications is to develop a development framework that has tough security architecture.

though there are extreme advantages in using a cloud-based system, there are yet many practical problems which have to be solved. Cloud computing is a disruptive technology with profound implications not only for Internet services but also for the IT sector as a whole. Still, several outstanding issues exist, particularly related to service-level agreements (SLA), security and privacy, and power efficiency.

14. Top 10 Obstacles and Opportunities for Cloud Computing:

Number 1. Business Continuity and Service Availability

Organizations worry about whether utility computing services will have adequate availability, and this makes some wary of cloud computing. Ironically, existing SaaS products have set a high standard in this regard.

Google Search has a reputation for being highly available, to the point that even a small disruption is picked up by major news sources.

Number 2. Data Lock-In

Software stacks have improved interoperability among platforms, but the storage

APIs for cloud computing are still essentially proprietary, or at least have not been the subject of active standardization. Thus, customers cannot easily extract their data and programs from one site to run on another. Concern about the difficulty of extracting data from the cloud is preventing some organizations from adopting cloud computing.

For example, an online storage service called The Linkup shut down on Aug. 8, 2008 after losing access as much as 45% of customer data.6 The Linkup, in turn, had relied on the online storage service Nirvanix to store customer data, which led to finger pointing between the two organizations as to why customer data was lost. Meanwhile, The Linkup's 20,000 users were told the service was no longer available and were urged to try out another storage site.

Solution one would be to standardize the APIsd in such a way that a SaaS developer could deploy services and data across multiple cloud computing providers so that the failure of a single company would not take all copies of customer data with it.

Second, in addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in an internal data center and in a public cloud.

Number 3. Data Confidentiality/Auditability

Despite most companies outsourcing payroll and many companies using external email services to hold sensitive information, security is one of the most often-cited objections to cloud computing.

Cloud users face security threats both from outside and inside the cloud. Many of the security issues involved in protecting clouds from outside threats are similar to those already facing large data centers. In the cloud, however, this responsibility is divided among potentially many parties, including the cloud user, the cloud vendor, and any third-party vendors that users rely on for security-sensitive software or configurations.

The cloud user is responsible for application-level security. The cloud provider is responsible for physical security, and likely for enforcing external firewall policies.

Number 4.Data Transfer Bottlenecks

Applications continue to become more data-intensive. If we assume applications may be “pulled apart” across the boundaries of clouds, this may complicate data placement and transport. At $100 to $150 per terabyte transferred, these costs can quickly add up, making data transfer costs an important issue. Cloud users and cloud providers have to think about the implications of placement and traffic at every level of the system if they want to minimize costs. This kind of reasoning can be seen in Amazon’s development of its new cloud front service.

One opportunity to overcome the high cost of Internet transfers is to ship disks. Jim Gray found the cheapest way to send a lot of data is to ship disks or even whole computers. While this does not address every use case, it effectively handles the case of large delay-tolerant point-to-point transfers, such as importing large data sets.

To quantify the argument, assume that we want to ship 10TB from U.C. Berkeley to Amazon in Seattle, WA. Garfinkel9 measured bandwidth to S3 from three sites and found an average write bandwidth of 5Mbits/sec to 18Mbits/ sec. Suppose we get 20Mbits/sec over a WAN link. It would take

10 * 1012 Bytes / (20×106 bits/second)
= (8×1013)/(2×107) seconds
= 4,000,000seconds,

which is more than 45 days. If we instead sent 10 1TB disks via overnight shipping, it would take less than a day to transfer 10TB, yielding an effective bandwidth of about 1,500Mbit/sec. For example, AWS8 recently started offering such a service, called import/Export.

Number 5. Performance Unpredictability

Our experience is that multiple virtual machines (VMs) can share CPUs and main memory surprisingly well in cloud computing, but that network and disk I/O sharing is more problematic. As a result, different EC2 instances vary more in their I/O performance than in main memory performance.

One opportunity is to improve architectures and operating systems to efficiently virtualize interrupts and I/O channels.

Another possibility is that flash memory will decrease I/O interference. Flash is semiconductor memory that preserves information when powered off like mechanical hard disks, but

since it has no moving parts, it is much faster to access and uses less energy.

Number 6: Scalable Storage

There are three properties whose combination gives cloud computing its appeal: short-term usage ,no upfront cost, and infinite capacity on demand. While it’s straightforward what this means when applied to computation, it’s less clear how to apply it to persistent storage. There have been many attempts to answer this question, varying in the richness of the query and storage API’s, the performance guarantees offered, and the resulting consistency semantics. The opportunity, which is still an open research problem, is to create a storage system that would not only meet existing programmer expectations in regard to durability, high availability, and the ability to manage and query data, but combine them with the cloud advantages of scaling arbitrarily up and down on demand.

Number 7: Bugs in Large-Scale Distributed Systems

One of the difficult challenges in cloud computing is removing errors in these very large-scale distributed systems. A common occurrence is that these bugs cannot be reproduced in smaller configurations, so the debugging must occur at scale in the production data centers.

One opportunity may be the reliance on virtual machines in cloud computing. Many traditional SaaS providers developed their infrastructure without using VMs, either because they preceded the recent popularity of VMs or because they felt they could not afford the performance hit of VMs. Since VMs are de rigueur in utility computing, that level of virtualization may make it possible to capture valuable information in ways that are implausible without VMs.

number 8: Scaling Quickly

Pay-as-you-go certainly applies to storage and to network bandwidth, both of which count bytes used. Computation is slightly different, depending on the virtualization level. Google AppEngine automatically scales in response to load increases and decreases, and users are charged by the cycles used. AWS charges by the hour for the number of instances you occupy, even if your machine is idle.

The opportunity is then to automatically scale quickly up and down in response to load in order to save money, but without violating service level agreements. Another reason for scaling is to conserve resources as well as money. Since an idle computer uses about two-thirds of the power of a busy computer, careful use of resources could reduce the impact of data centers on the environment, which is currently receiving a great deal of negative attention. conservation, but configuration hassles make it tempting to leave machines idle overnight so that startup time is zero when developers return to work the next day. A fast and easy-to-use snapshot/restart Tool might further encourage conservation of computing resources.

Number 9: Reputation Fate Sharing

One customer’s bad behavior can affect the reputation of others using the same cloud. For instance, blacklisting of EC2 IP addresses13 by spam prevention services may limit which applications can be effectively hosted. An opportunity would be to create reputation-guarding services similar to the “trusted email” services currently offered (for a fee) to services hosted on smaller ISP’s, which experience a microcosm of this problem.

Another legal issue is the question of transfer of legal liability—cloud computing providers would want customers to be liable and not them (such as,
the company sending the spam should be held liable, not Amazon).

Number 10: Software Licensing

Current software licenses commonly restrict the computers on which the software can run. Users pay for the software and then pay an annual maintenance fee. The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing.

For example, Microsoft and Amazon now offer pay-as-you-go software licensing for Windows Server and Windows SQL Server on EC2.

15. Some security principles are:

- Install and maintain a firewall configuration to protect cardholder data.
- Do not use vendor-supplied defaults for system passwords and other security parameters.
- Protect stored cardholder data.
- Encrypt transmission of cardholder data across open, public networks.
- Use and regularly update anti-virus software.
- Develop and maintain secure systems and applications.
- Restrict access to cardholder data by business need to know.
- Assign a unique ID to each person with computer access.
- Restrict physical access to cardholder data.
- Track and monitor all access to network resources and cardholder data.
- Regularly test security systems and processes.
- Maintain a policy that addresses information security.

16. Protection type:

If an organization does wish to press ahead with a cloud computing contract and if the organization does have concerns about the security of data, what types of protections should it seek? The following is a non-exhaustive list of suggestions:

- **Confidentiality** — Most cloud agreements will contain some form of confidentiality clause whereby the provider promises to maintain the confidentiality of the data and the user promises to maintain the confidentiality of the system itself.

- **Acknowledgement of customer's data ownership/No data mining** — Related to confidentiality above is the issue of data ownership. As between the customer and the service provider, it is industry standard that the customer would own the data that it supplies to the cloud. However, issues can arise about derivatives of this data or even meta data about the customer's use of the service. The customer should strive to ensure that all derivatives, all usage data and all meta data are vested in the customer and that the provider is not entitled to use them. An assignment of ownership back to the customer of the derivatives and meta data should be included to evidence this ownership interest.
- **Auditing controls** — ideally any type of cloud arrangement should permit the user to audit the provider's security and control systems, including user authentication, its processing and storage of information procedures, its disaster recovery and backup procedures, and its physical and organizational safeguards in relation to user data. In reality, particularly for an "as is" type of cloud application, the provider will not permit this because, understandably, it cannot have hundreds of customers descending upon it to audit its operations.
- **User authentication and management** — again, contractually it's not really practical to require compliance with a specific type of user authentication technology that is frozen at a point in time. It is more important to describe the desired outcomes such as:

❖ only users with the "need to know" and appropriate user authorization should be able to access or modify customer data;

❖ restrictions on archiving and backup so that the user knows where the data reside and who has access;

❖ restrictions on accessing and moving data to and from a specific location.

- **Strong passwords** — The user could require that the cloud provider only accept strong passwords from all co-tenants and all provider administrators touching the system. Strong passwords would have a combination of letters, numbers and symbols and would not be easily discoverable.
- **Encryption** — many providers already provide for encryption of data while in transit. However, data at rest is a more difficult issue. Encryption of data at rest is not yet a widely accepted standard. For sensitive data this is a must and any user thinking of the cloud to store or process sensitive data should be considering this issue. It may not be achievable, but if nothing else, the user will make an informed decision about the service that it is procuring.
- **Incident Responses and Escalation Procedures** —these issues should form part of any service level commitment that the user is able to extract from the cloud provider. Ideally the cloud contract would provide that, when a security incident comes to the attention of the cloud provider, the incident is reported to the user, along with the description of the potential effects and the provider's remediation efforts.Some negotiation will be required over what a "security incident" actually is.
- **Change Management** — Most cloud solution contracts do not speak to change management. This is more of an issue for customized situations.
- **Subcontracting** — many service providers take the position that "it's our service, so why do you care how we deliver it as long as you get what you bargained for"? That's true on some level, but it's a bit trite to say that, especially when dealing with a large cloud computing contract involving vast amounts of sensitive information. The customer needs to manage its own risk and part of that risk management is having some level of comfort that the entity actually performing the work is bound by the contractual commitments that the cloud provider is giving to the customer.
- **Rules for co-tenants** — most cloud providers have an "acceptable use policy". This should be read carefully to determine the types of activities that are prohibited.

- **Business Continuity** — One of the benefits that cloud providers tout is that the service is “always available”. Google recently announced that it has removed its exclusion for scheduled downtime from its service level agreement for Google Apps.From a security perspective, the customer needs to ensure that any provision of the service through any backup provider is subject to the same level of security requirements as the primary service provider.

17. Comparison of Performance of Different Providers of Cloud Computing:

We need to know the performance of different providers to get the best service. In this section, we discuss about the performance of different providers.

17.1 Sourcing Models:

Sourcing models (shared or dedicated and internally or externally hosted) are defined by the ownership and control of architectural design and the degree of available customization. The different sourcing models can be evaluated against the three standards - cost, control, and scalability.

- **Shared Public Cloud:** The Shared Public Cloud provides the benefit of rapid implementation, massive scalability, and low cost of entry. It is delivered in a shared environment where the architecture, customization, and degree of security are designed and managed by the provider according to market-driven specifications.

- **Dedicated Public Cloud:** The Dedicated Public Cloud provides functionality similar to a Shared Public Cloud except that it is delivered on a dedicated infrastructure. Security, performance, and sometimes customization are better in the Dedicated Public Cloud than in the Shared Public Cloud. Its architecture and service levels are defined by the provider and the cost may be higher than that of the Shared Public Cloud, depending on the volume.

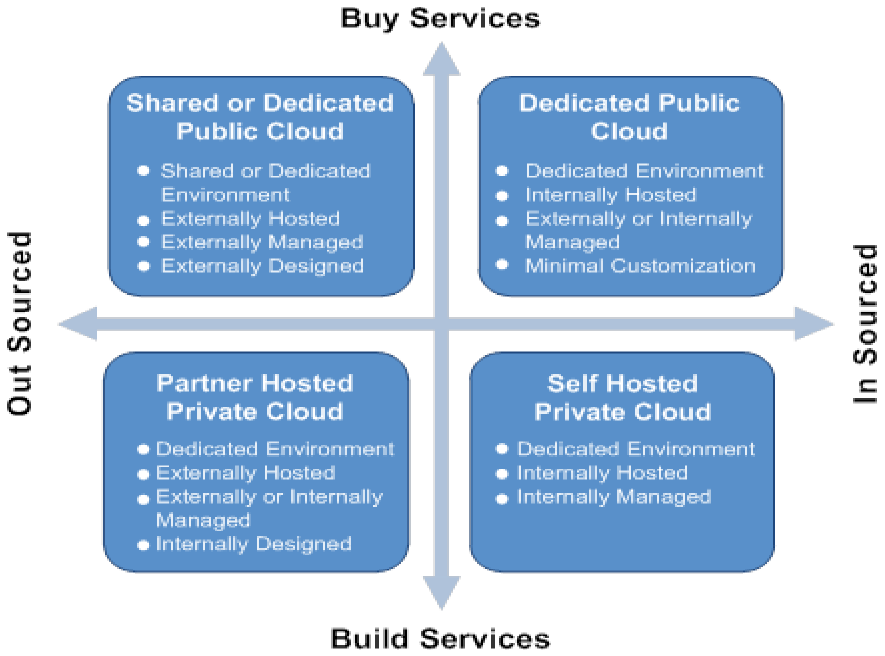


Fig15: sourcing models.

- **Self-hosted Private Cloud:** A Self-hosted Private Cloud provides the benefit of architectural and operational control, utilizes the existing investment in people and equipment, and provides a dedicated on-premises environment that is internally designed, hosted, and managed.
- **Partner-hosted Private Cloud:** A Partner-hosted Private Cloud is a dedicated environment that is internally designed, externally hosted, and externally managed. It blends the benefits of controlling the service and architectural design with the benefits of outsourcing.
- **Private Cloud Appliance:** A Private Cloud Appliance is a dedicated environment procured from a vendor, that is designed by the vendor with provider/market driven features and architectural control, is internally hosted, and externally or internally managed. It blends the benefits of using pre-defined functional architecture and lower deployment risk with the benefits of internal security and control.

Cloud Sourcing Type	**Hosting Location**	**Shared or Dedicated**	**Architectural Control**	**Scalability**	**Investments**
Shared Public Cloud	External	Shared	Provider or market	Minimal constrains	Pay as you go
Dedicated Public Cloud	External	Partially or fully dedicated	Provider or market	Constrained by contract	Pay as you go
Self-Hosted Private Cloud	Internal	Fully dedicated	*Self Comparison of Cloud Sourcing Models*	Constrained by capital investment	Build is Cloud, share resources
Partner-hosted Private Cloud	External	Fully dedicated	Self	Constrained by capital investment or contract	Varies by contract, may or may not have capital impact
Private Cloud Applance	Internal	Fully dedicated	Provider	Constrained by offering	Varies by contract, may or may not have capital impact

Table1: cloud sourcing types

18: Comparing public clouds and private data centers:

Advantage	*Public Cloud*	*Conventional Data Center*
Appearance of infinite computing resources on demand	*Yes*	*No*
Elimination of an up-front commitment by Cloud users	*Yes*	*No*
Ability to pay for use of computing resources *on a short-term basis as needed*	*Yes*	*No*
Economies of scale due to very large data centers	*yes*	*Usually not*
Higher utilization by multiplexing of workloads *from different organizations*	*Yes*	*Depends on company size*
Simplify operation and increase utilization via resource virtualization	*yes*	*No*

Table2: comparing public and conventional data centers.

Table3: cloud computing characteristics:

Cloud computing characteristics	**Description**
On-demand self-service	IT is used as service and is readily available on demand without requiring manual intervention.
Broad network access	The service is made available via a network independently of the user end device. The network connection must be of sufficiently high performance and available for that particular service.
Resource pooling	The provider makes the necessary resources available to multiple consumers using technologies such as virtualization and multi-tenancy.
Rapid elasticity	The resources necessary can be provisioned rapidly and released without manual intervention when no longer needed.

Measured Service	A service consumed must be measurable in terms of the re-sources used. In this way, consumption-based billing becomes possible. Also known as “pay as you go” or “pay-per-use.”

Table 4 – Advantages and disadvantages of IaaS over owning the infrastructure

Advantages	**Disadvantages**
• High scalability of the systems required based on actual needs • Redundant data storage • Physical separation of data use and data storage • No maintenance for setting up and running the infrastructure • OPEX instead of CAPEX OPEX - Operational Expenditures: expenses incurred in the operation of the business infrastructure. CAPEX capital Expenditure: investment expenses for long-term fixed assets. • Pay as you go	• Data location not always identifiable (trans-parent) in public and private clouds • Strong dependence on the availability of infra-structure and networks • No or insufficient distinction between or isolation of data processing (for the various users) • Unauthorized access to data possible in case of misconfiguration • Guarantee of confidentiality, security or integrity of the data; liability in case of a breach thereof .

Table 5 – Advantages and disadvantages of PaaS over owner operation

Advantages	**Disadvantages**
• Less administrative effort as there is no need to implement/run the infrastructure in-house • Development by (geographically distributed) teams possible • Single platform with minimal costs (standardization) • No maintenance in setting up and run-ning platform and its tools • OPEX instead of CAPEX • Pay as you go	• Vendor lock-in o Lack of portability o Lack of interoperability o No standardized technologies • Insufficient flexibility • Special requirements in case of proprietary applications or development environments

Table 6 – Advantages and disadvantages of SaaS over software ownership

Advantages	**Disadvantages**
• Separability/multitenancy of the application • Rapid deployment; faster project introduction (time to market) • No maintenance needed to run the business functionalities • OPEX instead of CAPEX • Pay as you go • Lower TCO • Mobility, location independence	• Choice of right provider • Lack of portability • Lower integrability into existing application environment • Lower adaptation possibilities as standardization is given • Potentially longer response times • Security vulnerabilities when using shared SaaS solutions • Cannot be used without access to internet

19: Different cloud service provider companies:

19.1 OpenStack:

❖ **History**

1. Joint project with Rackspace & NASA
2. Launched in June 2010
3. Enable anyone to create and offer cloud computing services
4. Many corporations joined

❖ **Components**

1. Nova (compute)
2. Swift(object storage)
3. Glance (image service)
4. Keystone(identity management)
5. Horizon (gui interface)

19.2 Eucalyptus:

❖ **History**

1. Started as a research project at UC Santa Barbara
2. Company founded in 2009 to commercialize the project

3. Split into two editions:
    - Open source
    - Open core

   June 2012 back to fully open source

❖ **Components**

1. Cloud controller(CLC)
   - Manages the virtualization resources and APIs
   - Provides web interface
2. Walrus(S3 storage)
3. Cluster controller(CC)
   - Controls execution of VMs and their networking
4. Storage controller(SC)
   - Provides block-level storage to VMs(EBS)
5. Node Controller(NC)
   - Controls VMs via hypervisors

19.3 CloudStack :

❖ **History**

1.Originally developed by Cloud.com

2.Open sourced in may 2010(GPLv3)

3.Citrix purchased Cloud.com in Aug 2011

4.Donated to ASF in Feb 2012

❖ Components

1. management server
2. hypervisor Nodes
3. Storage Nodes
4. Layers: zone, Pod, Cluster, host, primary storage, secondary storage

19.4 Ganeti

- **History:**

1. Started as internal google
2. Open sources in august 2007
3. Used primarily for back office servers for google
4. Focus on hardware fault-tolerance
5. Local block level storage
6. Cheap commidity hardware

- **Components**

1. Master daemon: Controls overall cluster coordination
   - Node daemon: Controls node functions(storage,VMs etc)
   - Conf daemon: Provide a fast way to query configuration
2. API daemon – provide a remote API

   Htools – Auto-allocation & rebalancing tools

20. How to choose platform:

To choose the platform we need to think about our needs. For example---

1. **Data that are going to be storing and sharing via your cloud solution:**

If you’re in an organization that does not handle mass amounts of confidential data, you may be less concerned about security, and more concerned about flexibility and cost-effectiveness. At the opposite end of the spectrum, organizations that handle large amounts of data should put security first on their list of needs, which often means moving data into a private cloud.

2. **Risk tolerance for risk:**

Risk tolerance can be determined by a number of things, from the type of data an organization works with, to the personal feelings of an executive team about data security. Any organization considering a cloud computing solution should sit down and determine how they would cope internally, and in the market, if there were a data leak, and make cloud-deployment decisions based on their risk requirements.

3. **Does your data need to stay housed within a certain region:**

For instance, in European countries, regulations mandate that customer data be stored within the country where the customers live. Therefore, companies that operate in numerous countries need to choose how they will store data specific to that country, and how that solution will integrate with the broader company infrastructure. Often the only viable options are private or hybrid cloud solutions.

4. **Can the Platform Support Multiple Languages, Databases and Middleware:**

You may need to use multiple languages or databases as you create your applications. Each application will have different needs as it is developed, and those needs also may change over time. By finding a cloud provider that can support multiple languages and databases, you'll avoid having to select a different cloud for each type of application. It's important not just to look at the service a cloud provider is offering, but whether that platform provides the depth and breadth you need.

For example, a SaaS company that helps developers build visual prototypes in the cloud leverages several PHP frameworks, as well as a variety of databases and queueing technologies to meet its clients' varying needs. And a sport merchandise company uses multiple languages and related frameworks and middleware to power its site to ensure customers can shop anytime.

5. **Flexible Pricing Options:**

- Pay-as-you-go pricing: For applications with short-term, spiky, or otherwise unpredictable usage, a pay-as-you-go pricing model, where you pay only for what you use, and with no upfront commitment, may be the most appropriate option.

  Amazon EC2 On-Demand Instances let you pay for compute capacity by the hour with no long-term commitments,

- Make an up-front payment to obtain a discounted hourly rate: For applications with steady state or otherwise predictable usage, you should be able to make a low, one-time, upfront payment and get a significant discount to the hourly rate and a capacity reservation in the cloud.

  Amazon EC2 Reserved Instances let you make a low, one-time upfront payment and pay a significantly lower hourly rate.

- Market-based pricing for significant discounts on excess capacity: For applications with flexible starts and stops (i.e. able to be interrupted), or for applications that are economical only at very low compute prices, market-based pricing can enable you to obtain significant discounts to standard on-demand pricing at a bid price that you specify.

  Amazon EC2 Spot Instances allow you to specify the price you are willing to pay to obtain AWS excess capacity. AWS makes excess capacity it has available to customers via the Spot market. If your max bid exceeds the Spot price at any juncture in your bid window, you receive those instances at your low Spot bid.

6. **Automatic Scaling:**

Automatically scale up or down to meet customer demand. Your cloud infrastructure should automatically scale up or down, adding or removing capacity based on the policies and metrics that you define. This helps you meet the demands of your customers, while paying only for what you need and use.Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to triggers you define.

7. **Load Balancing:**

Automatically balance variable request loads.To ensure that demand for your applications is evenly balanced across your cloud infrastructure, you should be able to take advantage of a load balancing service that automatically scales and manages itself, freeing you from having to deploy and manage a separate service. The load balancing service should also check the health of your application so that failures do not impact your users.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It seamlessly provides the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing can be used with Amazon VPC to provide internal and external load balancing.

8. High Performance Computing:

- Powerful multi-core processors: High Performance Computing (HPC) workloads often require multiple, high speed cores. The availability of these processors in an on-demand utility computing environment puts supercomputer class power in the hands of every developer.

  Amazon EC2 Cluster Compute Instances feature the latest Intel Xeon processor E5 family processors, with advanced vector extensions, MUNA, turbo mode and hardware virtualization to provide an extremely high performance environment for your codes.

- High speed interconnects : Many HPC codes exchange information between nodes of a cluster over the network. A fast, interconnected network ensures low latency delivery of this exchange, and can significantly accelerate large-scale computational workloads.

  Amazon EC2 Cluster Compute instances are deployed on a high performance, low latency, fully bisectional, 10Gb ethernet network.

- Physical proximity between instances: Placing instances such that their underlying hardware that is physically close together reduces communication latency between those instances, improving computational performance.

  Amazon EC2 cluster placement groups ensure that applications benefit from full-bisection bandwidth and low-latency network performance.

21. How to Safely Store your Data in the Cloud

Once you have found the service that best fits your needs, it is important to make your data as safe as possible. Here are some general rules that you should follow for all your internet habits, but particularly for your data storage:

- **Pick a good password.** All Cloud services require a master password to get into your files, so make it a good one, something that is pretty long. When it comes to passwords, longer is better. True, it can be a hassle to remember a strong password but it's an even bigger hassle to have your information stolen.
- **Don't reuse your passwords.** The password you choose to access the Cloud should be unlike any other password you use. If a hacker gets access to your Facebook password which also happens to be your email password, they will not only have a clear view of where you hold financial accounts, but they will be able to reset all of your passwords without your knowledge.
- **Don't share your passwords.** Even with a trusted friend, sharing your password is never a good idea. The more people who know your password, the more likely it is to be spread around. Your password is the lock to your information, don't let more people in than need be there.
- **Back up your data.** The same way you back up your computer's hard drive,back up your Cloud data. There are some companies that offer a small amount of storage free of cost. Take advantage of this and make sure you have your most important data backed up in case of an unexpected loss.

22. Comparison:

Among OpenStack, Eucalyptus, CloudStack, and Ganeti:

Table7: Storage comparison:

Type	OpenStack	Eucalyptus	CloudStack	Ganeti
Disk images	Yes	yes	Yes	Yes but disk image support

				has limitations.
Block devices	Yes. Via an elastic block storage service	Yes. Via an elastic block storage service	Yes like iSCSI, OCFS2, CLVM	Yes but primary storage method, also has sharedfs support.
Fault tolerance	Yes and users rsync in the backend	Yes but not added until version 3.0.uses DRBD	Yes but storage is on your own and parts are built in.	Yes

Table8: VM Image Comparison:

Type	OpenStack	Eucalyptus	CloudStack	Ganeti
Image Service	Yes	yes	Yes	No
Self service [1]	Yes	Yes	yes	No but third party applications can offer this
Amazon API	Yes but not all support	Yes	yes	No

1. Ability for users to create and manage their own VM images.

Table9: Self Service Comparison:

Type	OpenStack	Eucalyptus	CloudStack	Ganeti
Web Interface	Yes	Yes	Yes	Yes via third party application Ganeti Web Manager
Users & Quotas	Yes	Yes	Yes	Yes via third party application

				Ganeti Web Manager
Console access	Yes	Yes	Yes	Yes via third party application Ganeti Web Manager
User management	Yes	Yes	Yes	Yes via third party application Ganeti Web Manager

Table10:   Networking Comparison:

Type	OpenStack	Eucalyptus	CloudStack	Ganeti
Auto-allocation	Yes	yes	Yes	No but proposal submitted but not yet implemented
Floating IPs	Yes	Yes	yes	No
User defined	Yes	Yes	Yes	No
Layer 2	Yes	yes	Yes	No

Tabel11:   Other Factors:

	OpenStack	Eucalyptus	CloudStack	Ganeti
Codebase	Python	Java, C	Java	Python, Haskell, Shell
Hypervisors	Xen ,KVM,	Xen	Xen	Xen

	UML,LXC,VMware	,KVM,VMwarw	,KVM,VMware,citrix XenServer	,KVM,LXC
Installation Requirements	Medium	Large	Medium or large	Low
Maintenance	Many components to maintain	Depends on size	Medium	Easy

Table12:   Ease of installation:

OpenStack	Eucalyptus	CloudStack	Ganeti
Included in Ubuntu. Lots of configuration required. Puppet Labs Module	Excellent Install Guide Yum/APT repos Few commands for initialization	Provide their own repos Excellent install guide Minimal configuration needed	Included in Debian/ Ubuntu Good Docs. Simple initialization

Table13:   Strengths / weaknesses:

	OpenStack	Eucalyptus	CloudStack	Ganeti
Strengths	Single codebase Growing community Corporate support	Excellent commercial support Fault-tolerance Offers a hybrid-cloud solution	Well-rounded GUI Stack is fairly simple Customization of the storage	Fault- tolerance built in Customizable Very simple to manage and maintain

		with AWs	backend	
Weaknesses	Young codebase Uncertain future Initial configuration	Install requirements Configurable but not very customizable Community inclusion	Very GUI centric Single java core AWS integration weak	Admin centric VM Deployment No AWS integration

Comparison among XCP, Nimbus, OpenNebula, Eucalyptus, TPlatform, Apache VCL and Enomaly

Table14: Comparison between open-source Cloud Computing solutions:

Solutions	Service	Main characteristic	infrastructure	*Used by*
XCP	IaaS	Only a tool for automatic maintenance of clouds	Xen	XCP community
Nimbus	IaaS	Aims to turn legacy clusters into IaaS Clouds	Xen hypervisor and KVM	Brookhaven National Labs
OpenNebula	IaaS	Policy-driven resource allocation	Xen hypervisor	Cumulus Project
Eucalyptus	IaaS	Hierarchical Architecture	Xen hypervisor and KVM	UEC
TPlatform	PaaS	Focus on web text	TFS, BigTable	TPlatform Project

		mining applications	and MapReduce	
Apache VCL	SaaS	Internet access for several applications	VMware	Educational and Governmentusers
Enomaly	IaaS	Open version is focused in small clouds	Xen, KVM and VirtualBox	Several companies

23. Advantages of Using Cloud Computing in BRAC University:

There are many good reasons for which, Cloud computing is getting popular day by day. It has less overhead in terms of resource management and operation, time efficient etc. In this section, we discuss some of the advantages of using cloud computing in BRAC University.

- **Resource Management:**

We can easily use any software or run some simulation oriented software using cloud computing. In that case, we do not have to run the software in our server. The third party server will be responsible for the software running and maintenance. So we do not need any server and maintenance over head for these kinds of application. The same analogy is applicable for using any kinds of platform to use. With the help of cloud computing, we can efficiently use our resources.

- **Time Efficient:**

Traditionally, in server-client based system the server has to be in on-mode all the time for the clients regardless how many requests it gets over time. The system is not very time and energy efficient. For cloud computing, we will get a service on particular time basis. For an example, we are going to use a software for 2 hours. After 2 hours the session will be expire and service will be terminated. With a specific time frame of the service, the probability of getting a server idle is very less. As we do not have to run the server, we can use the computing more time efficiently.

- **Efficiently Resource Handle during Advising Session in the University:**

During the advising session, the university servers are on full utilization and sometimes get crashed due to high utilization. Cloud computing can be really useful in this scenario. It can allow many users at a time to get a service. As all the high performance server run and maintenance by third party, the utilization rate is always manageable and the probability of getting server down is almost zero. On the other hand, many user can use the service at the same time. In this particular scenario, cloud computing will ease the pressure of advising in the university very efficiently.

- **Using Different Software:**

Some of the software can be really expensive for university to buy for the whole year. Some software may use for one semester and it gets idle for the rest of the year. We can get these type of software from cloud and use for a specific time considering user demand. In terms of maintenance, it will be useful as third party is responsible for updating and renewal of license and maintenance of the software.

- **Running Intensive Simulation:**

For intensive simulation, we need high performance servers and huge storage. We know that it is very expensive to maintain these heavy duty severs. In this case, we can use cloud computing to run different simulation software from cloud and store all the results in different location.

- **Huge Storage for University Records**:

Cloud computing offers huge storage for data. We can store all the old documents and data in the storage and efficiently manage the data. We do not have to delete any data on the time basis as the storage is huge enough.

24. CONCLUSION:

Cloud Computing becomes a buzzword nowadays. More and more companies step into Cloud and provide services above on it. However, security and privacy issues impose strong barrier for users’ adoption of Cloud systems and Cloud services. We observed the security and privacy concerns presented by an amount of Cloud Computing system providers in this paper. Nevertheless, those concerns are not adequate. More security strategies should be deployed in the Cloud environment to achieve the 5 goals (i.e. availability, confidentiality, data integrity, control and audit) as well as privacy acts should be changed to adapt a new relationship between users and providers in the Cloud literature. We claim that prosperity in Cloud Computing literature is to be coming after those securities and privacy issues resolved.

REFERENCES

1. Amazon Web Services. TC3 Health Case Study; http://aws.amazon.com/solutions/case-studies/tc3-health/.
2. Armbrust, M., et al. Above the clouds: A Berkeley view of cloud computing. Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, Feb 2009.
3. Foster et al., "Cloud Computing and Grid Computing 360-Degree Compared," *Proc. IEEE Grid Computing Environments Workshop*, IEEE Press, 2008, pp. 1–10.
4. G. Rains, "Cloud Computing and SOA," MITRE, white paper, Oct. 2009; www. mitre .org/work / tech _ papers / tech _ papers_09/09_0743/09_0743.pdf.
5. M. Armburst et al., "Above the Clouds: A Berkeley View of Cloud Computing," tech. report UCB/EECS-2009-28, Electrical Eng. and Computer Science Dept., Univ. of California, Berkeley, 2009.
6. R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-Clouds: Managing Performance Interference Effects for QOS-Aware Clouds," *Proc. 5th European Conf. Computer Systems*, ACM Press, 2010, pp. 237–25
7. G arfinkel, S. An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS. Tech. Rep. TR-08-07,Harvard University, Aug. 2007.
8. McCalpin, J. Memory bandwidth and machine balance in current high performance computers. IEEE Technical Committee on Computer Architecture Newsletter (1995), 19–25
9. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster,C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke."Data Management and Transfer in High Performance Computational Grid Environments", Parallel ComputingJournal, Vol. 28 (5), May 2002, pp. 749-771.
10. Amazon Elastic Compute Cloud (Amazon EC2),http://aws.amazon.com/ec2, 2008.
11. Amazon Simple Storage Service (Amazon S3),http://aws.amazon.com/s3, 2008.
12. R. Buyya, D. Abramson, J. Giddy. "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid", IEEE Int. Conf. on High Performance Computing in Asia-Pacific Region (HPC ASIA) 2000.
13. R. Buyya, K. Bubendorfer. "Market Oriented Grid and Utility Computing", Wiley Press, New York, USA, 2008.
14. I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.
15. I. Foster, C. Kesselman, S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organization. The Intl. Jrnl. of High Performance Computing Applications, 15(3):200--222, 2001.

16. Ananthanarayanan R, Gupta K et al (2009) Cloud analytics: do we really need to reinvent the storage stack? In: Proc of HotCloud
17. Chandra A et al (2009) Nebulas: using distributed voluntary resources to build clouds. In: Proc of HotCloud
18. Chang F, Dean J et al (2006) Bigtable: a distributed storage system for structured data. In: Proc of OSDI
19. C.Wang, "Forrester: A close look at cloud computing security issues,"http://www.forrester.com/securityforum2009, 2009.
20. IDC,"It cloud services user survey, pt.2: Top benefits & challenges,"http://blogs.idc.com/ie/?p=210, 2008.
21. J.Bardin, "Security Guidance for Critical Areas of Focus in Cloud Computing,"www.cloudsecurityalliance.org/guidance/csaguide.pdf, 2009.
22. R. Gellman, "Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud Computing,"www.worldprivacyforum.org/pdf/WPF Cloud Privacy Report.pdf,2009
23. End user privacy in Human Interaction system http://www.cs.cmu.edu/~jasonh/publications/fnt-end-user-privacy-in-human-computer-interaction-final.pdf.
24. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I. (2009) "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: Future Generation Computer Systems, Elsevier B. V.
25. Peter Mell & Tim Grance, The NIST Definition of Cloud Computing (2009), online: National Institute of Standards and Technolog http://csrc.nist.gov/groups/SNS/cloud-computing/clouddef-v15.doc
26. Peter Mell & Tim Grance, "Effectively and Securely Using the Cloud Computing Paradigm" (Presentation delivered at the National Institute of Standards and Technology, October 7, 2009),online: NIST <http://csrc.nist.gov/groups/SNS/cloudcomputing/ cloud-computing-v26.ppt#313,81>.
27. Barry Reingold & Ryan Mrazik, "Cloud Computing: The Intersection of Massive Scalability, Data Security and Privacy (Part 1)" (2009) 14:5 Cyberspace Law 1 at 2-3, online: Perkins Coie http://www.perkinscoie.com/files/upload/PS_09-06_Cloud_Computing_Article.pdf
28. F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber.Bigtable: A Distributed Storage System for Structured Data. Proceedings of 7th Symposium on Operating System Design and Implementation (OSDI), page 205218, 2006.
29. J. Chase, D. Irwin, L. Grit, J. Moore, and S. Sprenkle. Dynamic virtual clusters in a grid site manager. High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on, pages 90–100, 2003.

30. J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Proceedings of 6th Symposium on Operating System Design and Implementation(OSDI), pages 137–150, 2004
31. W. Huang, M. Koop, Q. Gao, and D. Panda. Virtual machine aware communication libraries for high performance computing. In Proceedings of Supercomputing 2007.
32. Sun Microsystems, Inc., “Building customer trust in cloud computing with transparent security,” Online at https://www.sun.com/offers/details/sun transparency.xml, November 2009.