RAMAIAH
Institute of Technology

Cloud Computing and Big Data

Subject Code: CS71 (Credits: 4:0:0)

Textbook:

1. Cloud Computing Theory and Practice – **DAN C**. Marinescu – Morgan Kaufmann Elsevier.
2. Cloud Computing A hands - on approach – Arshdeep Bahga & **Vijay madisetti** Universities press
3. Big Data Analytics, Seema Acharya and Subhashini Chellappan. 2nd edition, Wiley India Pvt. Ltd. 2019

NOTE: I declare that the PPT content is picked up from the prescribed course text books or reference material prescribed in the syllabus book and Online Portals.

Unit I

Introduction:

- Cloud Computing, delivery models & Services,
- Ethical issues, Cloud vulnerabilities, Challenges,
- Cloud Infrastructure: Amazon, Google, Azure & online services,
- Open source private clouds. Storage diversity and vendor lock-in, intercloud,
- Energy use & ecological impact of data centers,
- Service and Compliance level agreement, Responsibility sharing, User experience, Software licensing.

Applications & Paradigms:

- Challenges, existing and new application opportunities,
- Architectural styles of cloud applications;
- different cloud architectures,
- Applications: Healthcare, Energy systems, transportation, manufacturing, Education, Government, mobile communication, application development.

Introduction: Cloud Computing

Cloud Computing refers to:

- **Use of Remote Resources hosted on the Internet to share /manage/ process data.**
- **It provides ability to manage Applications and Services through a Network.**
- **It Refers to the delivery of On-Demand computing Services over the Internet on Pay-as-you-go model.**

Introduction: Cloud Computing

- Cloud Computing refers to **manipulating, configuring,** and **accessing** the hardware and software resources remotely. It offers online data storage, infrastructure, and application.
- Cloud Computing can be defined as **delivering computing power (CPU, RAM, Network Speeds, Storage ,OS software) a service over a network (usually on the internet)** rather than physically having the computing resources at the customer location.
- Cloud computing means **on demand delivery of IT resources** via the internet with **pay-as-you-go pricing**. It provides a solution of IT infrastructure in low cost.
    - **Cloud Computing** Uses **Internet technologies** to offer **scalable and elastic services**.
        - The term "elastic computing" refers to **the ability of dynamically acquiring computing resources and supporting a variable workload**.
    - The data storage strategy can increase **reliability, as well as security**, and can lower communication costs

Why Cloud Computing?

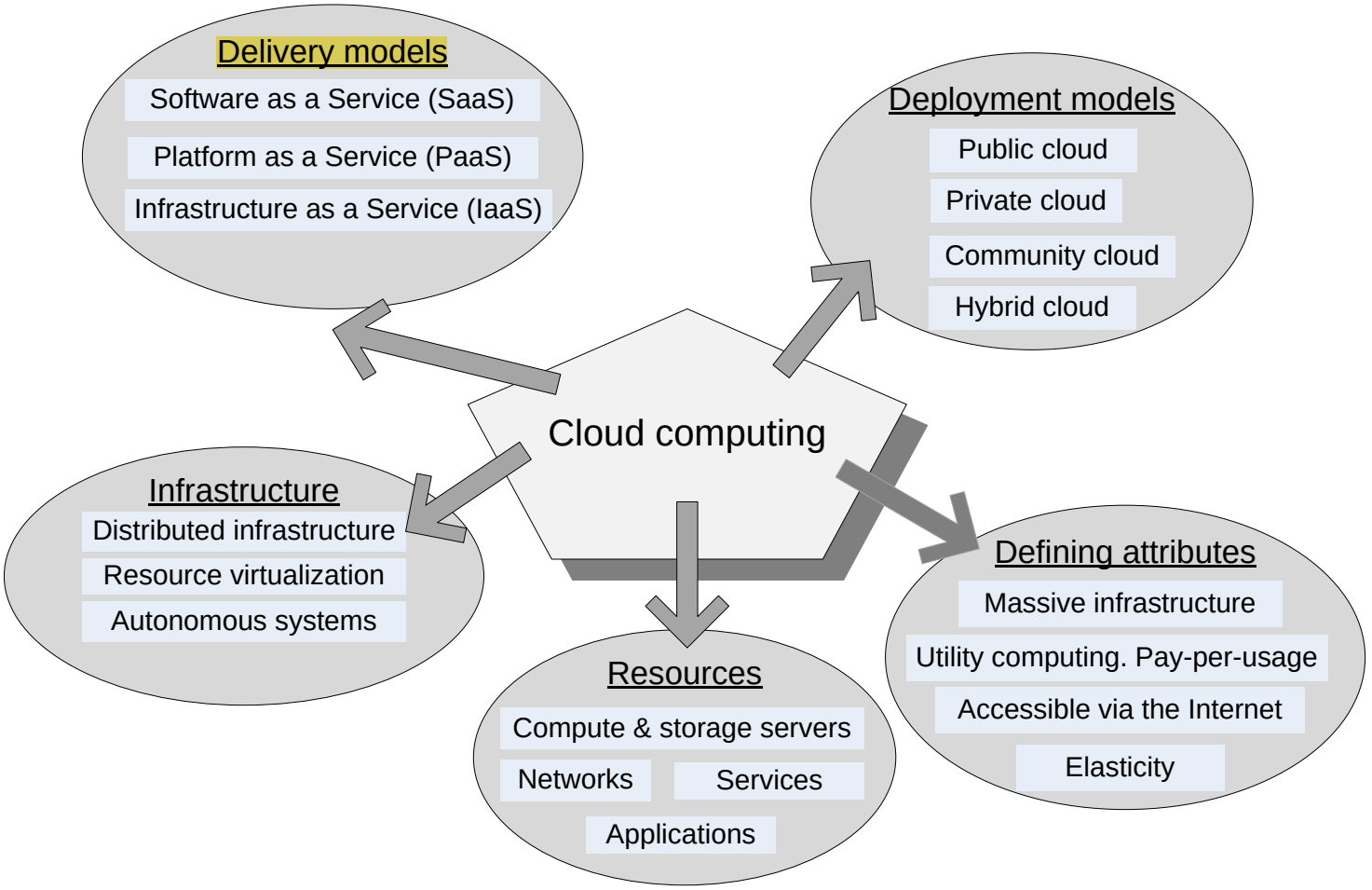
- With **increase in computer and Mobile user's**, **data storage has become a priority in all fields.** Large and small scale businesses today thrive on their data & they spent a huge amount of money to maintain this data. It **requires a strong IT support and a storage hub.**

    - **Not all businesses can afford high cost of in-house IT infrastructure** and **back up support services.** For them Cloud Computing is a cheaper solution.

- Cloud computing **decreases the hardware and software demand from the user's side**. The only thing that user must be able to run is the cloud computing systems interface software, which can be as simple as Web browser, and the Cloud network takes care of the rest.

- In an **in-house IT server, you have to pay a lot of attention and ensure that there are no flaws into the system** so that it runs smoothly. And in case of any technical glitch you are completely responsible; it will seek a lot of attention, time and money for repair. Whereas, in cloud computing, **the service provider takes the complete responsibility of the complication and the technical faults**

Cloud Computing: Delivery Models and Deployment Models



Types of Cloud Computing: Deployment Models

Public Cloud - the infrastructure is made available to the general public or a large industry group and is owned by the organization selling cloud services.

Private Cloud – the infrastructure is operated solely for an organization.

Community Cloud - the infrastructure is shared by several organizations and supports a community that has shared concerns.

Hybrid Cloud - composition of two or more clouds (public, private, or community) as unique entities but bound by standardized technology that enables data and application portability

Private Cloud

- The Private cloud allows **the accessibility of systems and services within the organization**. Private cloud is operated only within a particular organization. But it **will be managed internally or by third party.**

- Private Cloud also termed as **'Internal Cloud'; which allows the accessibility of systems and services within a specific boundary or organization**.

- The cloud platform is implemented in a cloud-based secure environment that is guarded by advanced firewalls under the surveillance of the IT department that belongs to a particular organization. **Private clouds permit only authorized users, providing the organizations greater control over data and its security.**

    - **More Control:** Private clouds have more control on its resources and hardware than public cloud because it is accessed only within the boundary of an organization.

    - **High security and privacy:** Private cloud resources are shared from distinct pool of resources and hence highly secured.

Public Cloud

- Public cloud allows the accessibility of systems and services easily to general public.

Eg: **Amazon, IBM, Microsoft, Google, Rackspace etc**

- **Reliable:** Public cloud provides large number of resources from different locations, **if any of the resource fail, public cloud can employ another one.**

- **High Scalability**: Cloud resources are available as per the demand from the pool of resources that means they can be **scaled up or down according to the requirement**.

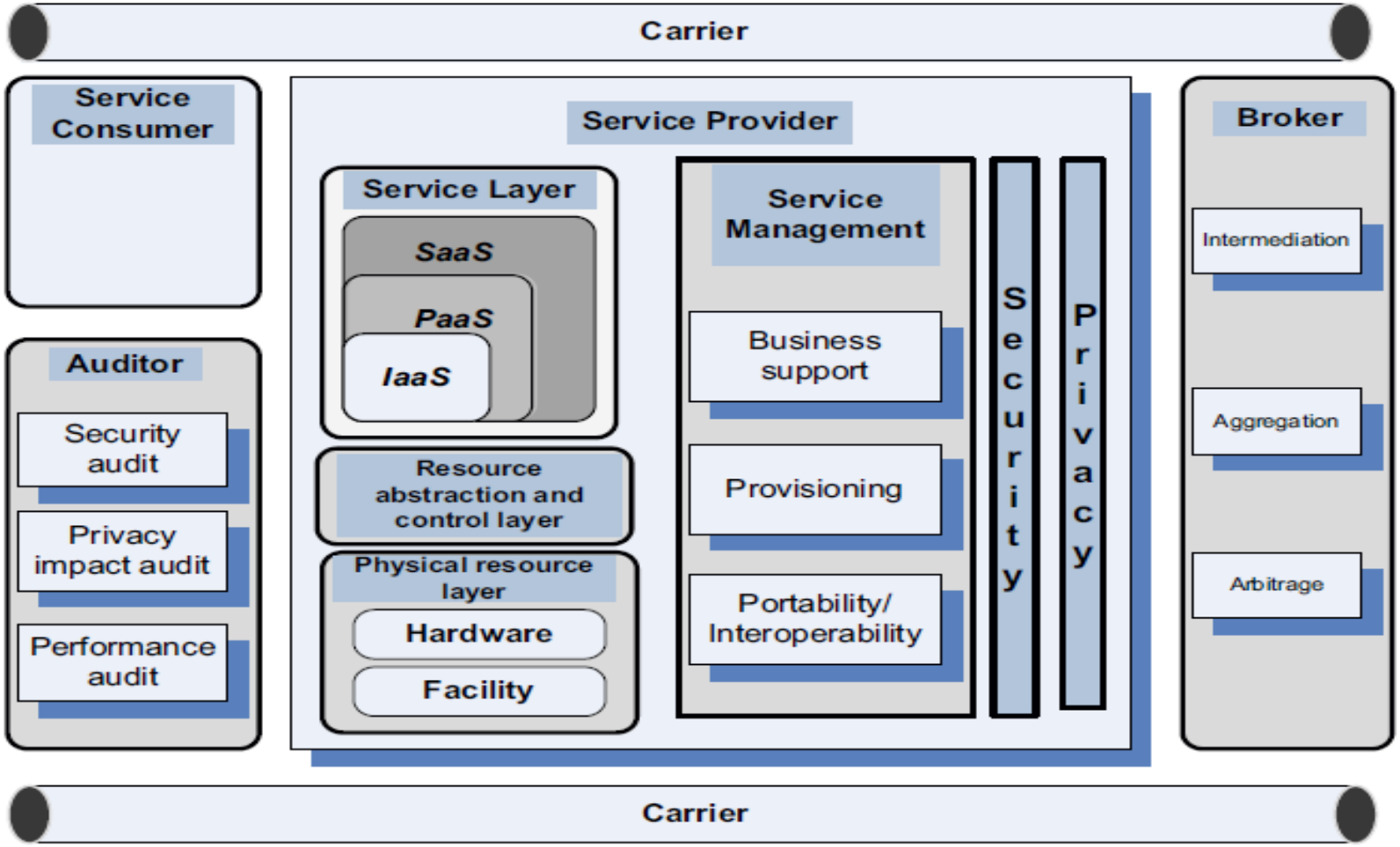
Community Cloud

It is another type of cloud computing in which the setup of the cloud is shared manually among different **organizations that belong to the same community or area**. Ex: BANK

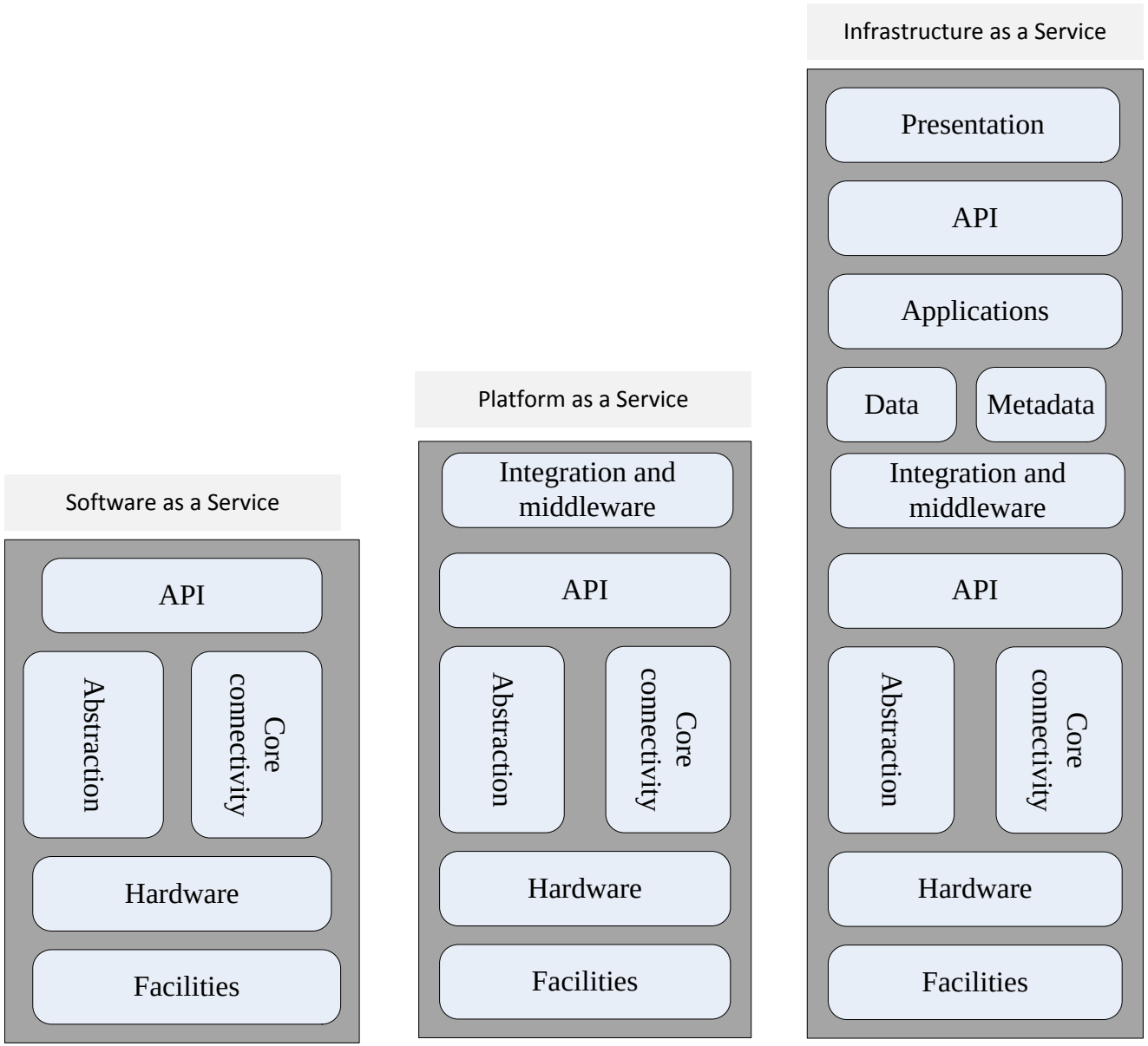
- Example of such a community is where organizations/firms are there along with the financial institutions/banks.
- A multi-tenant setup developed using cloud among different organizations that belong to a particular community or group having similar computing concern.

Deployment Models and Services

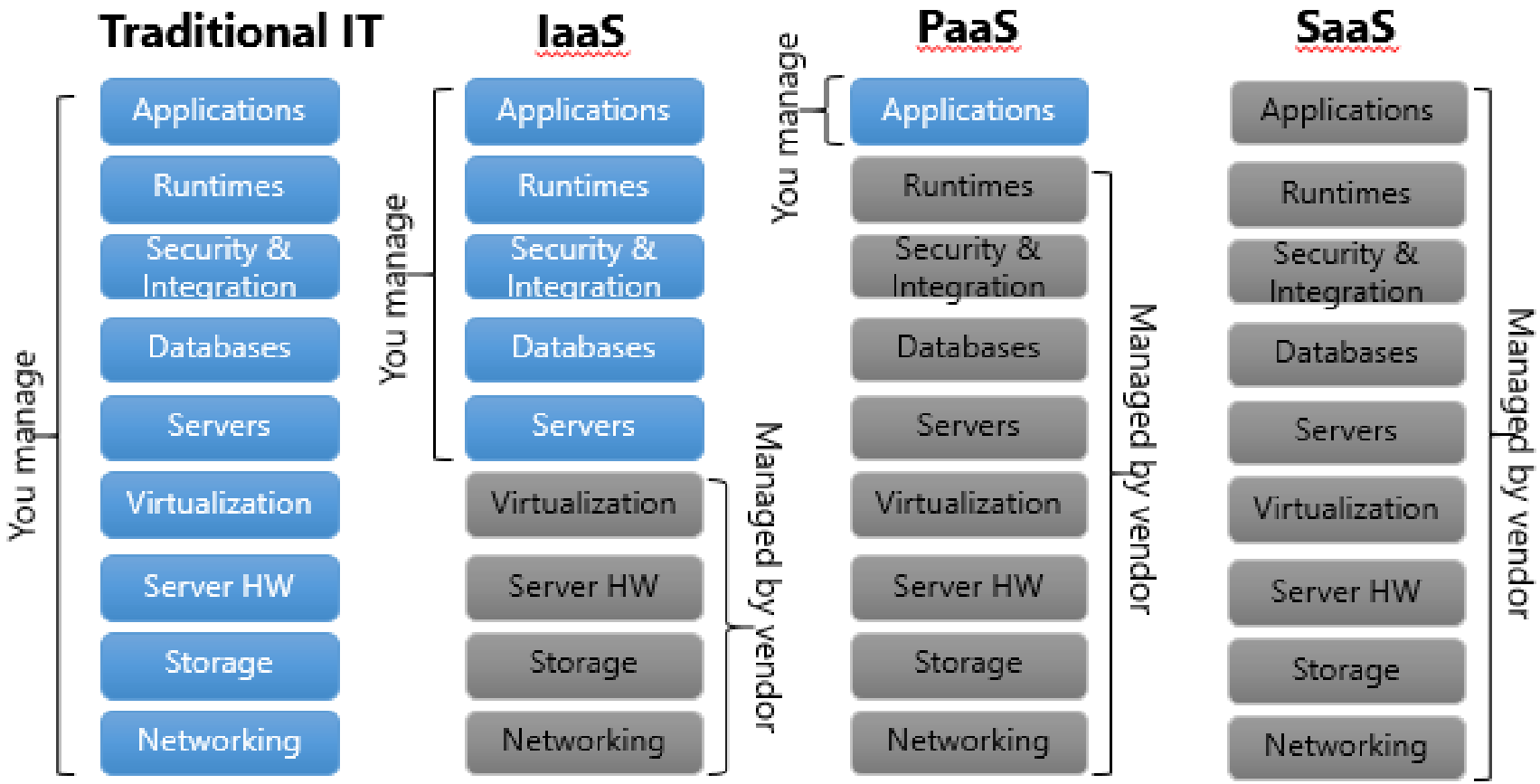
National Institute of Standard and Technology[NIST]



Types of Cloud Computing: Service Models



Types of Cloud Computing: Service Models



Software as a Service | SaaS

- SaaS gives the capability to **use applications supplied by Service Provider** in a Cloud Infrastructure.
- SaaS is defined as the **software distribution model** that is **deployed on the internet** in which a cloud service provider provides applications. **It is also known as "on-demand software" or "pay-as-you-go application".**

- In SaaS, the software & the applications associated with it are **centrally located on the cloud server,** and users can access them via a thin client connecting application, i.e., **using a web browser.**

- **The SaaS provides various applications such as:**
    - CRM applications
    - Solution to Human Resource (HR)
    - Pre-existing Billing & Invoicing systems
    - Workflow Management, Supply Chain Communication, financial Management.
    - Web 2.0 Applications such as Metadata managment, Social Networking, Blogs and Portal Services.

There are various advantages. (SaaS)

- **Easy to buy:** The cost of **SaaS is based on a monthly or yearly fees** allowing new organizations to access the world of business at a low-cost, at least lesser than licensed application.
- **Minimization of Hardware Requirement:** All SaaS **software is hosted remotely** & so there is no or lesser need of hardware for the organizations.
- **Special Software:** No special software versions are required as all the users will use the same software version. **SaaS reduces IT costs by outsourcing hardware & software maintenance**.
- **Low Maintenance:** SaaS **removes the daily problem of installing, maintaining and updating software**. The set-up cost of SaaS is also less in comparison to enterprise software.

Disadvantages while using the SaaS:

- **Latency factor:** comes due to a variable distance of data between the cloud & the end-user, and hence a possibility of latency may arise while interacting with applications.
- **Internet Connection:** is a major issue. Without internet connection, SaaS applications are unusable.
- **Switching between SaaS vendors** in case of any change is very difficult

Platform as a Service | PaaS

- Allows a cloud user **to deploy consumer-created or acquired applications** using programming languages and tools supported by the service provider

  - **The user:**
    - Has **control over the deployed applications** and, possibly, application hosting environment configurations.
    - **Does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage.**

- PaaS cloud computing platform is a developer programming platform which is created for the programmer **to develop, test, run and manage the applications.**

- PaaS has a feature of point-and-click tool that **allows non-programmers to develop** web applications.
- Examples: **App-Engine of Google & Force.com, Windows Azure, AppFog, Openshift and VMware Cloud Foundry.**

Advantages of PaaS:

- **Scalability:** of users ranges from hundreds to thousands.
- **Prebuilt Business Plan:** PaaS vendors provide pre-defined business functionality for users to directly start the project.
- **Low Cost:** Development via PaaS requires a computer & a good internet connection and hence less investment in hardware & software.
- **Instant Community:** PaaS providers facilitates user providing online communities where a developer can get new ideas & share their experience & advice.
- **Simple & easy to use**

Disadvantages of PaaS are as follows:

- **Vendor Migration:** Migration from one PaaS vendors' application to another PaaS vendor will create some problem.

Infrastructure as a Service | IaaS

- **The capability to provision processing, storage, networks and other fundamental computing resources;**
- IaaS, as the name suggests, is a way of providing **Cloud computing infrastructure such as virtual machines, storage drives, servers, operating systems & networks**, which is also an on-demand service like that of SaaS.

- **Services offered by this delivery model include:** server hosting, Web servers, storage, computing hardware, operating systems, virtual instances, load balancing, Internet access, and bandwidth provisioning.

- **Rather than purchasing servers or developing software, clients buy those resources as a fully outsourced service based on their requirement**.

- "Public cloud" is considered as an infrastructure that consists of shared resources, based on a self-service over the Internet.

- In one word, it is the only layer of the cloud where the customer gets the platform for their organization to outsource IT infrastructure on a pay-per-use basis.

Advantages of IaaS cloud computing layer

- You can **dynamically choose a CPU, memory and storage configuration as per your needs.**
- You **easily access** the vast computing power available on IaaS cloud platform
- You can **eliminate the need of investment** in rarely used IT hardware.
- IT infra will be handled by the IaaS **cloud computing platform vendors.**

Ethical issues in Cloud Computing :

- Implications
  - **Unauthorized access.**
  - **Data corruption.**
  - **Infrastructure failure, and service unavailability.**

- **De-perimeterization:** Systems can span **the boundaries of multiple organizations** and cross the security borders.
- The complex structure of cloud services can make it difficult to determine **who is responsible in case something undesirable happens.**

- **Identity fraud and theft are made possible by the unauthorized access to personal data** in circulation and by new forms of dissemination through social networks and they could also pose a danger to cloud computing.

- **Accountability is necessary ingredient of cloud computing** ;Adequate information about How data is handled within the cloud and about allocation of responsibility are key elements for enforcing ethics rules in cloud computing.
- **Unwanted dependency on cloud service provider** , the so called **vendor lock-in ,** is a serious concern, and the current standardization efforts at NIST attempt to address this problem.

Cloud Vulnerabilities:

- **Vulnerability**: is a weakness that can be **exploited by the attacker for his own personal gain or to perform unauthorized actions within a computer system**. A weakness can be present in software, environments, systems, network, etc.
- To exploit a **vulnerability**, an attacker must have at least one applicable tool or technique that can connect to a system weakness.

- **Internet Dependency:** By using the cloud services, we're dependent upon the Internet connection, so **if the Internet temporarily fails due to a lightning strike or ISP maintenance, the clients won't be able to connect to the cloud services**. Therefore, the business will slowly lose money, because the users won't be able to use the service that's required for the business operation.

- **Account Hijacking/Session Riding/Down Time/Failure of Infrastructure**
    - Insufficient identity, credential, and access management
    - Insecure interfaces and application programming interfaces (APIs)

- **System Vulnerabilities.** Some common system vulnerabilities include:
    - Lack of input validation on user input
    - Lack of sufficient logging mechanism
    - Fail-open error handling
    - Not closing the database connection properly

Challenges for cloud computing

Data Security and Privacy

- The data stored online is always possessed with **the risk of cyber-attack** and that is true when it comes to cloud storage. The size of the data stored in the cloud is vulnerable to cyber-attacks leading to **the risk of losing data.**

- The fact that **the valuable enterprise data** will reside outside the corporate firewall raises serious concerns**. Hacking to cloud infrastructure would affect multiple clients even if only one site is attacked**.

- These risks can be mitigated by using security applications, encrypted file systems, data loss software, and buying security hardware to track unusual behavior across servers.

- Entrepreneurs have to think on these issues before adopting cloud computing technology for their business. Since you are **transferring your company's important details to a third party so it is important to ensure yourself about the manageability and security system** of the cloud

Challenges for cloud computing

Data Security and Privacy

- **Highly sensitive applications** related to the management of the Acritical Infrastructure, Healthcare Applications and others will most likely be **hosted by Private clouds.**

- **Data in storage** is most vulnerable to attack , so special attention should be devoted to the protection of storage servers.

- **Data replication is necessary** may protect data in storage, but eventually data must be decrypted for processing and then it is exposed to attack.

- **Virtualization is a critical design option** .The **trusted computing base(TCB)** of virtual environment includes not only the hardware and the hypervisor but also the management of operating system.

Interoperability and Standardization

- **Availability of service;** what happens when the service provider cannot deliver?
- Diversity of services, data organization, user interfaces available at different service providers limit user mobility; once a customer is hooked to one provider it is hard to move to another. Standardization efforts at NIST! (Vendor Lock in)
- **Data confidentiality and auditability**, a serious problem.
- **Data transfer bottleneck**; many applications are data-intensive.

Resource management on the Cloud

- It include several class of policies
    - **Admission Control,**
    - **Capacity Allocation,**
    - **Load Balancing ,**
    - **Energy Optimization and QoS gurentees.**

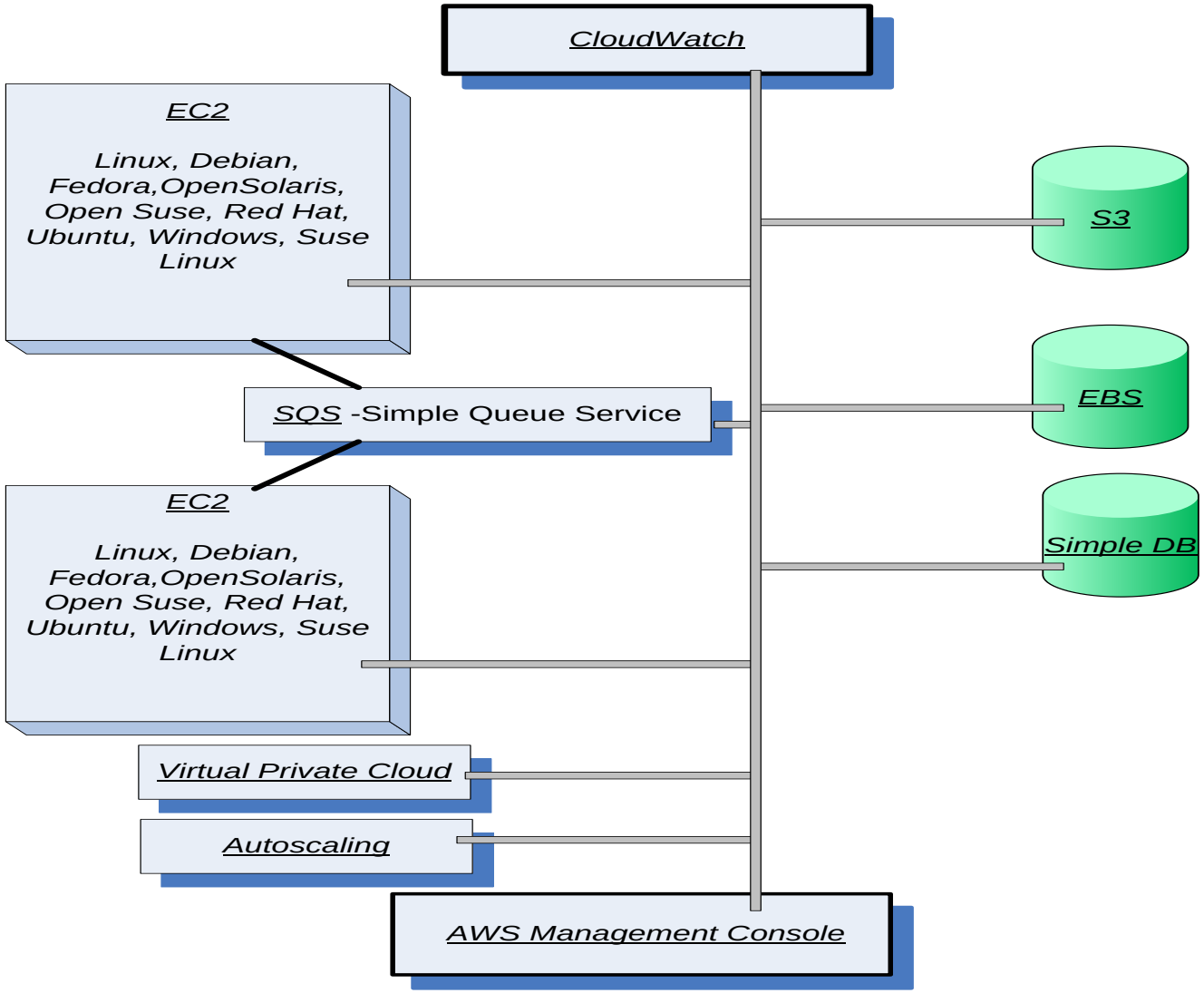
Amazon Web Services

- Amazon Web Services (AWS) is Amazon's cloud web hosting platform that offers **flexible, reliable, scalable, easy-to-use, and cost-effective solutions**.

- Amazon Web Services (AWS) is the **market leader in IaaS (Infrastructure-as-a-Service)** and **PaaS (Platform-as-a-Service) for cloud ecosystems**,

    - which can be combined to create a scalable cloud application without worrying about delays related **to infrastructure provisioning (compute, storage, and network) and management.**

Cloud Infrastructure: Amazon Web Services



Examples of Amazon Web Services

- ***AWS Management Console*** - allows users to **access the services offered by AWS**.
- ***Elastic Cloud Computing (EC2)*** - allows a user to **launch a variety of operating systems.**
- ***Simple Queuing Service (SQS)*** - allows **multiple EC2 instances** to communicate with one another.
- ***Simple Storage Service (S3),*** *Simple DB*, and Elastic *Bloc Storage (EBS)* - **storage services.**
- ***Cloud Watch*** - **supports performance monitoring.**
- ***Auto Scaling*** - supports **elastic resource management**.
- ***Virtual Private Cloud*** - **allows direct migration** of parallel applications.

What Is Amazon EC2?

- Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the **Amazon Web Services (AWS)** cloud.

- You can use **Amazon EC2** to **launch as many or as few virtual servers** as you need, configure security and networking, and manage storage.

- Amazon EC2 enables you to **scale up or down** to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

- EC2 - web service for **launching instances of an application under several operating systems, such as:**
    - Several Linux distributions.
    - Microsoft Windows Server 2003 and 2008.
    - OpenSolaris.
    - FreeBSD.
    - NetBSD.

- **EC2 instances** boot from an **AMI (Amazon Machine Image)** digitally signed and stored in S3.
- A user can interact with EC2 using a set of **SOAP(Simple Object Access Protocol) Messages**

AWS instances

An instance is a virtual server with a well specified set of resources including: CPU cycles, main memory, secondary storage, communication and I/O bandwidth.

- **The user chooses:**
  - The **region and the availability zone** where this virtual server should be placed.
  - An instance type from a limited menu of **instance types**.
- When launched, **an instance is provided with a DNS name**; this name maps to a
  - **Private IP address** → for internal communication within the internal EC2 communication network.
  - **Public IP address** → for communication outside the internal Amazon network, e.g., for communication with the user that launched the instance.
- **Network Address Translation (NAT) maps external IP addresses to internal ones**.
- The public IP address is assigned for the lifetime of an instance.
- **An instance can request an *elastic IP address***, rather than a public IP address. The elastic IP address is a static public IP address allocated to an instance from the available pool of the availability zone.
- **An elastic IP address is not released when the instance is stopped or terminated and must be released when no longer needed.**

Instance types

- **Standard instances**: micro (StdM), small (StdS), large (StdL), extra large (StdXL); small is the default.
- **High memory instances**: high-memory extra large (HmXL), high-memory double extra large (Hm2XL), and high-memory quadruple extra large (Hm4XL).
- **High CPU instances**: high-CPU extra large (HcpuXL).
- **Cluster computing**: cluster computing quadruple extra large (Cl4XL).

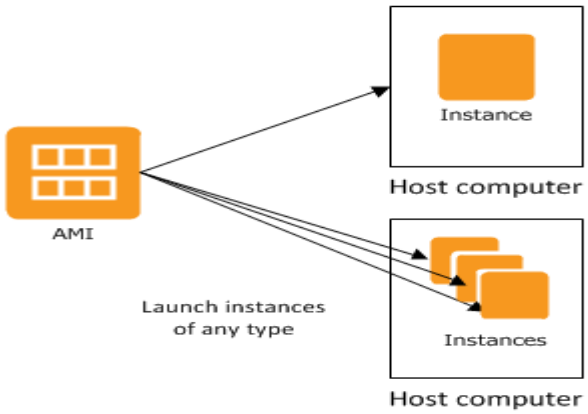
Instance name	API name	Platform (32/64-bit)	Memory (GB)	Max EC2 compute units	I-memory (GB)	I/O (M/H)
StdM		32 and 64	0. 633	1 VC; 2 CUs		
StdS	`m1.small`	32	1.7	1 VC; 1 CU	160	M
StdL	`m1.large`	64	7.5	2 VCs; 2 × 2 CUs	85	H
StdXL	`m1.xlarge`	64	15	4 VCs; 4 × 2 CUs	1,690	H
HmXL	`m2.xlarge`	64	17.1	2 VCs; 2 × 3.25 CUs	420	M
Hm2XL	`m2.2xlarge`	64	34.2	4 VCs; 4 × 3.25 CUs	850	H
Hm4XL	`m2.4xlarge`	64	68.4	8 VCs; 8 × 3.25 CUs	1,690	H
HcpuXL	`c1.xlarge`	64	7	8 VCs; 8 × 2.5 CUs	1,690	H
Cl4XL	`cc1.4xlarge`	64	18	33.5 CUs	1,690	H

Amazon EC2 provides the following features:

- **Virtual computing environments(Virtual Server**), known as **instances**
- **Preconfigured templates** for your instances, known as **Amazon Machine Images (AMIs),** that package the bits you need for your server (including the operating system and additional software)
- Various configurations of **CPU, memory, storage, and networking capacity for your instances**, known as **instance types**
- **Secure login information** for your instances using **key pairs** (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as **instance store volumes**
- Persistent storage volumes for your data using **Amazon Elastic Block Store (Amazon EBS),** known as Amazon EBS volumes
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as **regions** and **Availability Zones**
- A **firewall that enables you to specify the protocols, ports, and source IP ranges** that can reach your instances using **security groups**
- Static IPv4 addresses for dynamic cloud computing, known as **Elastic IP addresses**
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as **virtual private clouds (VPCs)**

- An **Amazon Machine Image (AMI)** is a template that contains **a software configuration** (for example, an operating system, an application server, and applications).

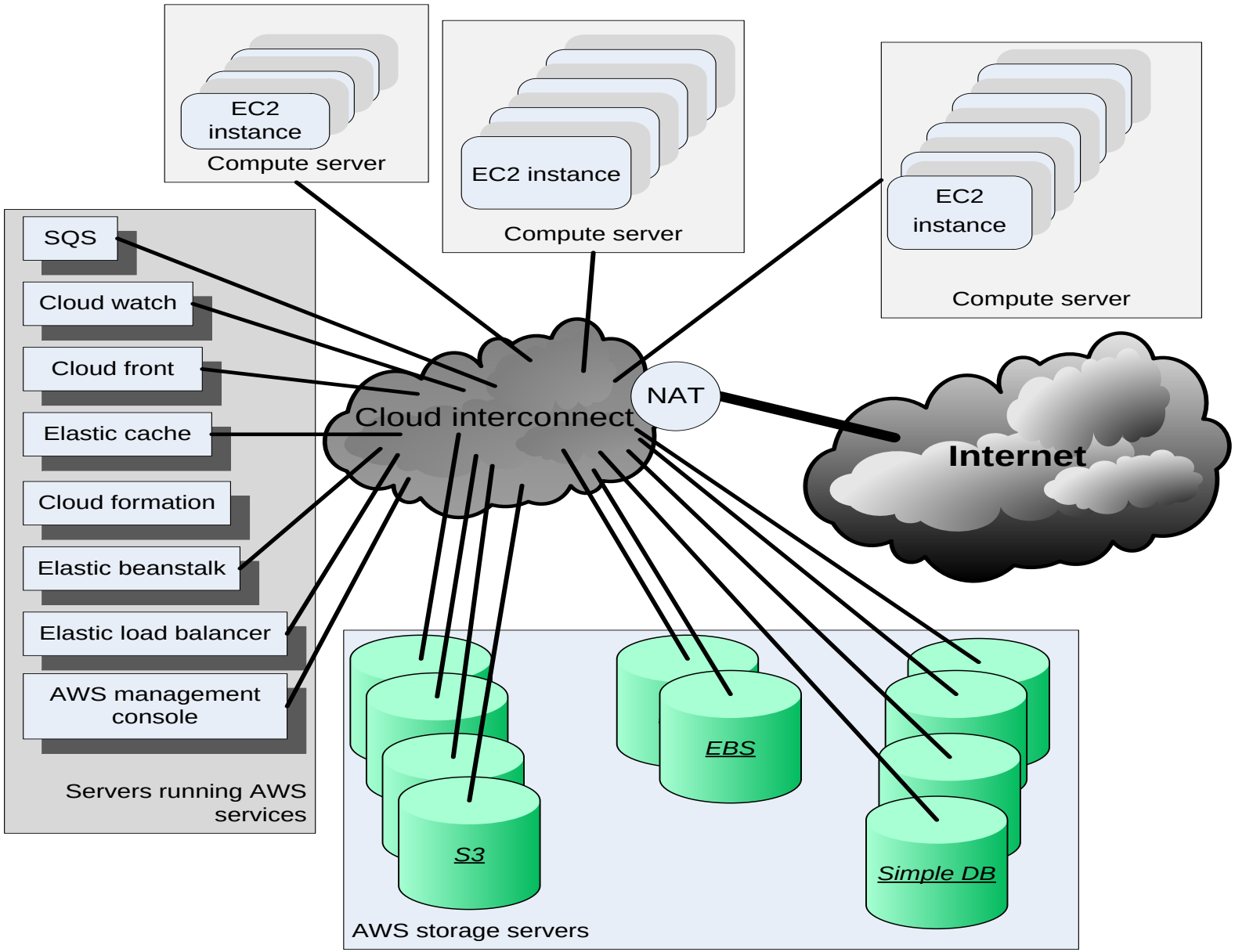
- From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud. **You can launch multiple instances of an AMI**, as shown in the following figure



- You can launch **different types of instances from a single AMI.**

- **Each instance type offers different compute and memory capabilities**.

    - Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance.


EC2 instance
Compute server
EC2 instance
Compute server
EC2 instance
Compute server
SQS
Cloud watch
Cloud front
Elastic cache
Cloud formation
Elastic beanstalk
Elastic load balancer
AWS management console
Servers running AWS services
Cloud interconnect
NAT
Internet
S3
EBS
Simple DB
AWS storage servers

S3 – Simple Storage System

- Amazon S3 is **a repository for Internet data**. Amazon S3 provides access to **reliable, fast, and inexpensive data storage infrastructure.**
- It is designed to make web-scale computing easy by **enabling you to store and retrieve any amount of data, at any time**, from within Amazon EC2 or anywhere on the web.
- Amazon S3 stores **data objects redundantly on multiple devices across multiple facilities** and allows **concurrent read or write access to these data objects** by many separate clients or application threads
- **Service designed to store large objects**; an application can handle **an unlimited number of objects ranging in size from 1 byte to 5 TB.**
- **An object is stored in a bucket** and retrieved **via a unique, developer-assigned key;** a bucket can be stored in a Region selected by the user.
- **S3 maintains for each object: the name, modification time, an access control list**, and up to 4 KB of user-defined metadata.
- Supports a minimal set of functions: **write, read, and delete**; it **does not support primitives to copy, to rename, or to move an object from one bucket to another.**

Elastic Block Store (EBS)

- **Amazon Elastic Block Store (Amazon EBS)** provides block level storage volumes for use with EC2 instances.

- EBS volumes are **highly available and reliable storage** volumes that can be attached to any running **instance that is in the same Availability Zone.**

- EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. **With Amazon EBS, you pay only for what you use.**

- Provides **persistent block level storage volumes for use with *EC2* instances**; suitable for database applications, file systems, and applications using raw data devices.
- A volume appears to **an application as a raw, unformatted and reliable physical disk**; the range 1 GB -1 TB
- An *EC2* instance may mount **multiple volumes**, but a volume cannot be shared among multiple instances.
- EBS supports the creation of **snapshots of the volumes attached to an instance** and then uses them to restart the instance.
- The **volumes are grouped together in Availability Zones and are automatically replicated in each zone**.

Simple DB

- Amazon SimpleDB is **a highly available NoSQL data store** that offloads the work of database administration.
    - Developers simply **store and query data items via web services requests** and Amazon SimpleDB does the rest.
- Amazon SimpleDB **creates and manages multiple geographically distributed replicas of your data** automatically to enable high availability and data durability.
- With Amazon SimpleDB, you can focus on application development without worrying about **infrastructure**
    - **provisioning, high availability, software maintenance, schema and index management, or performance tuning.**

SQS - Simple Queue Service

- **Amazon Simple Queue Service (SQS)** is a fully managed **message queuing service** that enables you to decouple and scale microservices, distributed systems, and serverless applications.

- SQS eliminates the complexity and overhead associated with **managing and operating message oriented middleware**, and empowers developers to focus on differentiating work.

- Using SQS, you **can send, store, and receive messages between software components** at any volume, without losing messages or requiring other services to be available.

  - Hosted message queues are accessed through **standard SOAP and Query interfaces**.

  - Supports **automated workflows - *EC2* instances can coordinate** by sending and receiving SQS messages.

  - Applications using **SQS can run independently and asynchronously**, and do not need to be developed with the same technologies.

CloudWatch

- Amazon CloudWatch is **a monitoring and management service** built for developers, system operators, site reliability engineers (SRE), and IT managers.

- CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to **system-wide performance changes, optimize resource utilization**, and get a unified view of operational health.

- CloudWatch collects **monitoring and operational data in the form of logs, metrics, and events,** providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers.

- You can use **CloudWatch to set high resolution alarms**, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to optimize your applications, and ensure they are running smoothly.

- When launching an Amazon Machine Image (AMI) the user can start the CloudWatch and specify the type of monitoring:

    - **Basic Monitoring** - free of charge; collects data at five-minute intervals for up to seven metrics.
    - **Detailed Monitoring** - subject to charge; collects data at one minute interval.

Amazon Virtual Private Cloud

- VPC provides **a bridge between the existing IT infrastructure of an organization and AWS Cloud**

- **Amazon Virtual Private Cloud (Amazon VPC)** lets you provision a logically isolated section of the AWS Cloud where you can **launch AWS resources in a virtual network** that you define.

- You have complete **control over your virtual networking environment**,

    - including selection of your own IP address range,
    - creation of subnets, and configuration of route tables and network gateways.

- You can **use both IPv4 and IPv6 in your VPC** for **secure and easy access to resources** and applications.

AWS services introduced in 2012

- ***Elastic MapReduce (EMR)*** - supports **processing of large amounts of data** using a hosted Hadoop running on *EC2*.

- ***Simple Workflow Service (SWF)*** - supports **workflow management;** allows scheduling, management of dependencies, and **coordination of multiple EC2 instances**.

- ***ElastiCache*** - enables web applications to retrieve data from a managed in-memory caching system rather than a much slower disk-based database.

- ***DynamoDB*** - scalable and low-latency fully managed **NoSQL database service**.

- ***CloudFront*** - web service for content delivery.
- ***Elastic Load Balancer*** - **automatically distributes the incoming requests** across multiple instances of the application.
- ***Elastic Beanstalk*** - handles automatically deployment, capacity provisioning, load balancing, auto-scaling, and application monitoring functions.
- ***CloudFormation*** - allows the creation of a stack describing the infrastructure for an application.

Elastic Beanstalk

- AWS Elastic Beanstalk is an easy-to-use service **for deploying and scaling web applications** and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

- You can simply upload your code and **Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring**.

- At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

- There is no additional charge for Elastic Beanstalk - you **pay only for the AWS resources needed to store and run your applications.**

Cloud computing the Google perspective:

- **Google Cloud Platform** is a suite of **public cloud computing services** offered by Google. The platform includes a range of hosted **services for compute, storage and application development that run on Google hardware.**
    - Google Cloud Platform services can be accessed by **software developers, cloud administrators and other enterprise IT professionals** over the public internet or through a **dedicated network connection**.
- Google ‘s effort is concerned in the area of **Software-as-a-Service(SaaS)**
    - Services such as **Gmail, Google Drive, Google Calendar, Picasa and Google Groups**

Cloud Infrastructure: Google

- ***Gmail*** - hosts Emails on Google servers and provides **a web interface to access the Email**.
- *Google docs* - a web-based software for **building text documents, spreadsheets and presentations.**
- ***Google Calendar*** - a browser-based scheduler; **supports multiple user calendars, calendar sharing, event search, display of daily/weekly/monthly** views, and so on.
- ***Google Groups*** - allows users to host discussion forums to create messages online or via Email.
- ***Picasa*** - a tool to **upload, share, and edit images**.
- ***Google Maps*** - web mapping service; **offers street maps, a route planner,** and an urban business locator for numerous countries around the world

- **Google Compute Engine**, which is an **infrastructure-as-a-service (IaaS)** offering that provides users with **virtual machine instances** for workload hosting.

- **Google App Engine**, which is a **platform-as-a-service (PaaS)** offering that gives software developers access to Google's scalable hosting.

    - **Developers can also use a software developer kit (SDK) to develop software products that run on App Engine**.

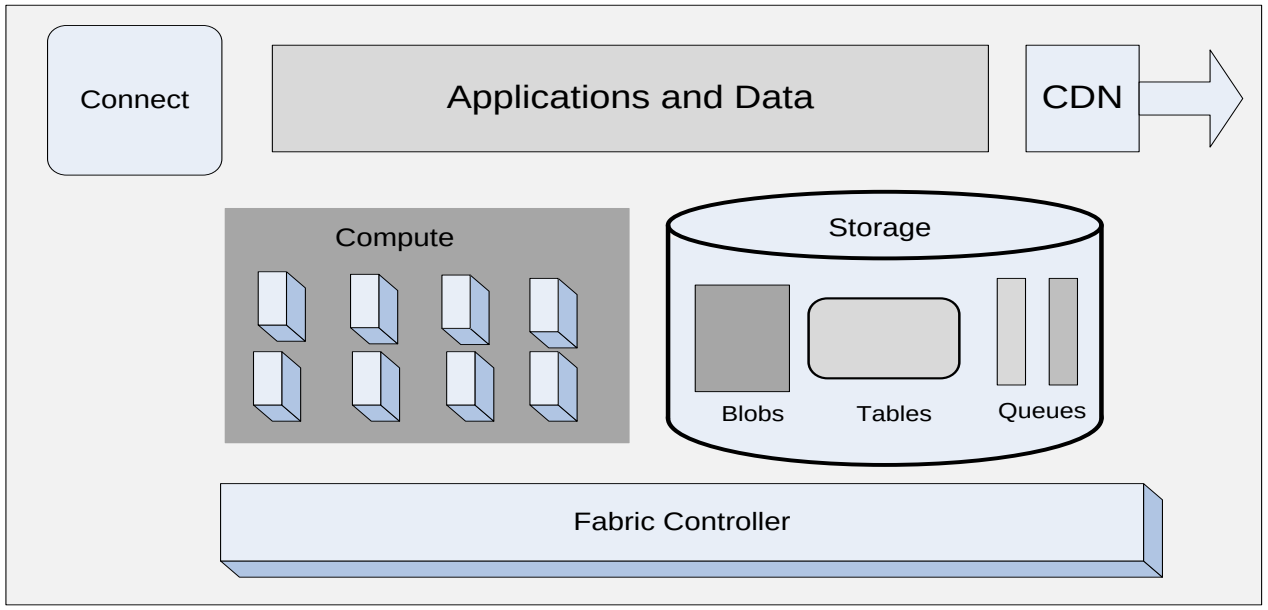
- ***AppEngine*** - a **developer platform** hosted on the cloud.
    - Initially **supported Python, Java** was added later.
    - The database for code development can be accessed with GQL (Google Query Language) with a SQL-like syntax.

- ***Google Co-op*** - allows users to create **customized search engines** based on a set of facets/categories.

- ***Google Drive*** - an online service **for data storage**.

- ***Google Base*** - allows users to load structured data from different sources to a central repository, **a very large, self-describing, semi-structured, heterogeneous database.**

Cloud Infrastructure: Azure & components of Windows Azure



- **Microsoft Azure** is a cloud computing service created by Microsoft for **building, testing, deploying, and managing applications and services** through a global network of Microsoft-managed data centers

- Microsoft Azure, formerly known as **Windows Azure**, is **Microsoft's public cloud computing platform**. It provides a range of cloud services, including those for **compute, analytics, storage and networking.**

Cloud Infrastructure: Azure & components of Windows Azure

- Azure OS is that part of Azure platform which runs applications and stores data. Windows Azure OS contains five services.

  - **Compute:** provides **computational Environment**
  - **Storage:** For **scalable Storage**
  - **Fabric Controller:** Which **deploys, manages and monitors applications**. (It ensures **Scaling, load balancing, memory management**, and reliability)
  - **CDN(Content Delivery Networks):** Maintains cache copies of data to speed up computations.
  - **Connect:** supports IP connections between the users and their applications running on Azure.

Open-source platforms for private clouds

- **Eucalyptus** is a paid and **open-source computer software for building Amazon Web Services-compatible private and hybrid cloud computing environments**, originally developed by the company Eucalyptus Systems.
    - **Eucalyptus** - can be regarded as an open-source **counterpart of Amazon's EC2**
    - Eucalyptus is an acronym for **Elastic Utility Computing Architecture for Linking Your Programs** To Useful Systems.(www.eucalyptus.com)

- ***Virtual Machines*** - run under several VMMs including Xen, KVM, and VMware.
- ***Node Controller*** - runs on server nodes hosting a VM and controls the activities of the node.
- ***Cluster Controller*** - controls a number of servers.
- ***Cloud Controller*** - provides the cloud access to end-users, developers, and administrators.
- ***Storage Controller*** - provides persistent virtual hard drives to applications. It is the correspondent of EBS.
- ***Storage Service (Walrus)*** - provides persistent storage; similar to S3, it allows users to store objects in buckets.

SEARCH

LEARN | EUCALYPTUS CLOUD | PARTICIPATE | SERVICES | PARTNERS

- EUCALYPTUS
- RIGHTSCALE MYCLOUD
- ECOSYSTEM TOOLS
- SECURITY
- DOCUMENTATION
- GET EUCALYPTUS
  - FastStart
  - Free Trial
  - Eucalyptus
  - Euca2ools
  - Community Cloud
  - Source

Home > Eucalyptus Cloud > Get Eucalyptus

DOWNLOAD EUCALYPTUS

First time using Eucalyptus? Try Eucalyptus FastStart.

1. Download and Install Eucalyptus

Choose a distribution:

- CentOS 5
- CentOS 6
- RHEL 5
- RHEL 6
- Ubuntu 10.04 LTS
- Ubuntu 12.04 LTS
- Source
- Versions prior to Eucalytpus 3.1

Nightlies
Release Notes

Looking for Euca2ools?

2. Configure Your Cloud

Documentation | Engage (Q&A) | Consulting
Education | Support

3. Use Your Cloud

To help get you started, we have prepared pre-packaged virtual machines ready to run in your Eucalyptus cloud.
Download images

Or check out a variety of use cases.

Learn About Eucalyptus For

- MANAGERS
- ARCHITECTS
- APPLICATION ARCHITECTS
- ADMINISTRATORS
- DEVELOPERS
- USERS

Euca2ools

Eucalyptus supported command-line tools.
Get Euca2ools

Ecosystem Tools

Find tools developed for Amazon EC2 and S3 which are compatible with Eucalyptus.
Get tools

Open-source platforms for private clouds

- **Open-Nebula** is a cloud computing platform for **managing heterogeneous distributed data center infrastructures.**
  - The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service
  - a private cloud with users actually logging into the head node to access cloud functions. The system is centralized and its **default configuration uses the NFS file** system.(www.opennebula.org)

- **Nimbus** - a cloud solution **for scientific applications** based on Globus software; inherits from Globus:(www.nimbusproject.org)
  - The image storage.
  - The credentials for user authentication.
  - The requirement that a running Nimbus process can **ssh** into all compute nodes.

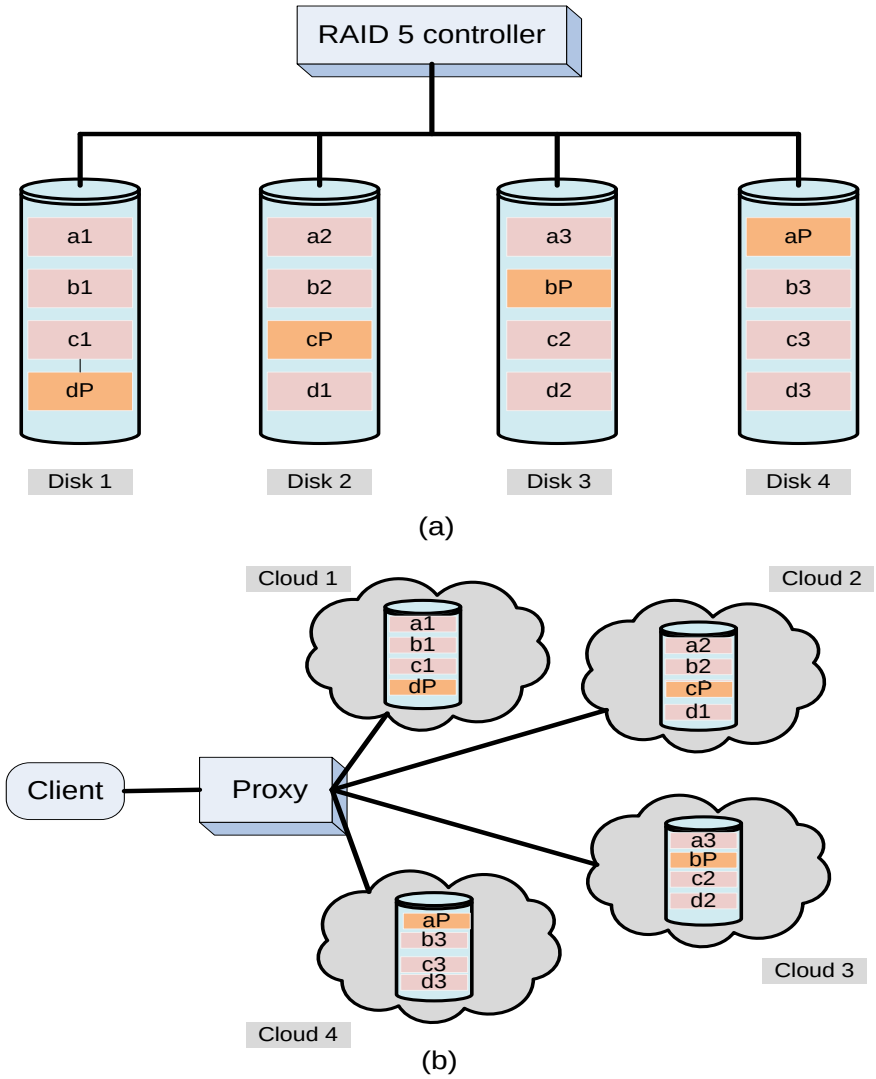
Comparison of Eucalyptus, OpenNebula, and Nimbus.

Table 3.5 A side-by-side comparison of *Eucalyptus*, *OpenNebula*, and *Nimbus*.

	Eucalyptus	***OpenNebula***	***Nimbus***
Design	Emulate *EC2*	Customizable	Based on Globus
Cloud type	Private	Private	Public/Private
User population	Large	Small	Large
Applications	All	All	Scientific
Customizability	Administrators and limited users	Administrators and users	All but image storage and credentials
Internal security	Strict	Loose	Strict
User access	User credentials	User credentials	x509 credentials
Network access	To cluster controller	—	To each compute node

Cloud storage diversity and vendor lock-in

- Risks when a large organization relies on a single cloud service provider:
  - Cloud **services may be unavailable** for a short or an extended period of time.
  - **Permanent data loss** in case of a catastrophic system failure.
  - The **provider may increase the prices for service.**
- **Switching to another provider could be very costly due to the large volume of data to be transferred from the old to the new provider.**
- A solution is to **replicate the data to multiple cloud service providers, similar to data replication in RAID**.



- **vendor lock-in**, also known as proprietary **lock-in** or customer **lock-in**, makes a customer dependent on a **vendor** for products and services, unable to use another **vendor** **without substantial switching costs.**

    - **Lock-in** costs that create barriers to market entry may result in antitrust action against a monopoly.

A solution to guarding against the problems posed by the vendor lock-in is to **replicate the data to multiple cloud service providers.**

- Straightforward replication is very costly and, at the same time, poses technical challenges.

- The overhead to maintain data consistency could drastically affect the performance of the virtual storage system consisting of multiple full replicas of the organization's data spread over multiple vendors.

Another solution could be based on **an extension of the design principle of a RAID-5** system used for reliable data storage.

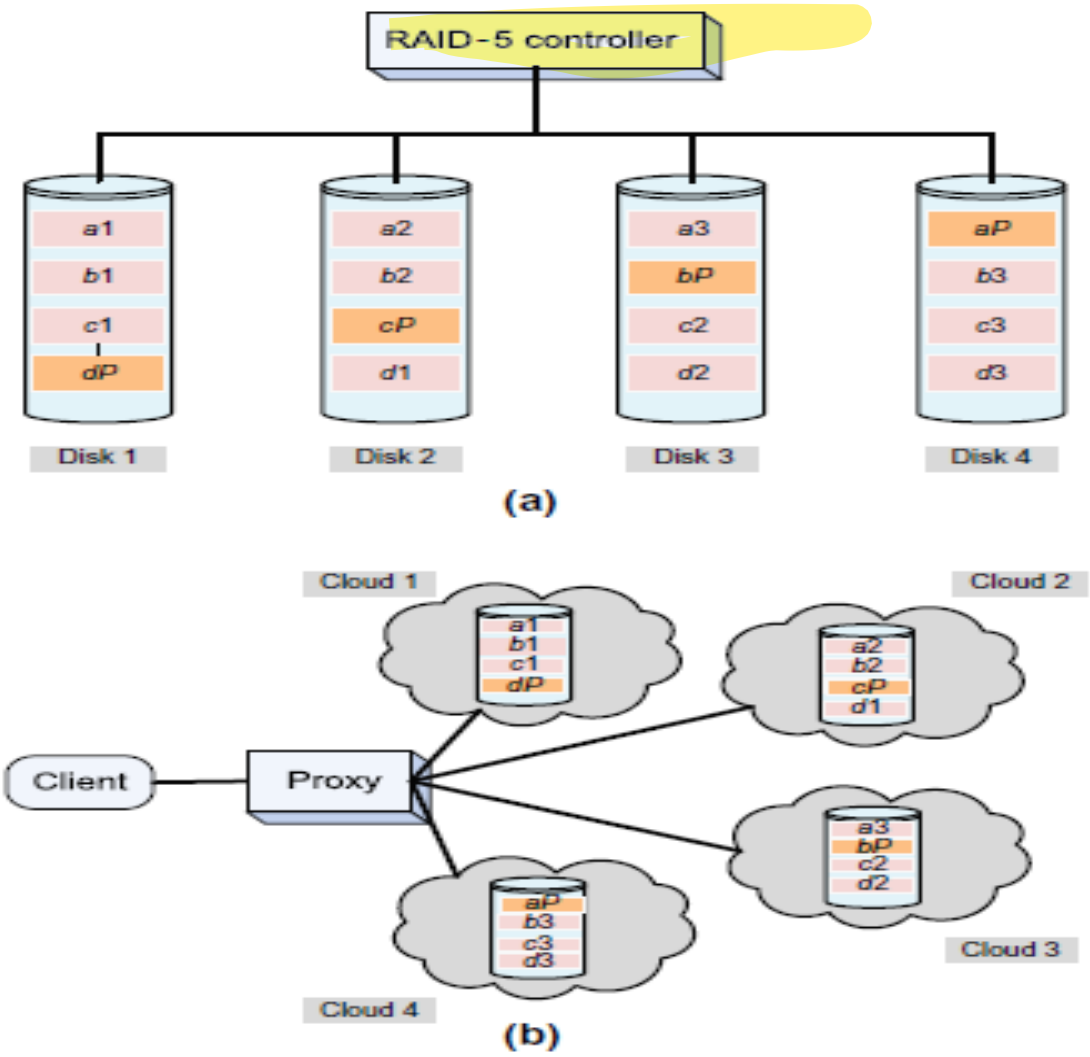


FIGURE 3.5

(a) A (3, 4) RAID-5 configuration in which individual blocks are stripped over three disks and a parity block is added; the parity block is constructed by XOR-ing the data blocks (e.g., $aP = a1 \text{XOR} a2 \text{XOR} a3$). The parity blocks are distributed among the four disks: aP is on disk 4, bP on disk 3, cP on disk 2, and dP on disk 1. (b) A system that strips data across four clouds; the proxy provides transparent access to data.

- A RAID-5 system **uses block-level stripping** with **distributed parity over a disk array**, as shown in Figure 3.5(a); the disk controller distributes the sequential blocks of data to the physical disks and computes **a parity block by bit-wise XOR-ing of the data blocks.**

- The **parity block is written on a different disk for each file** to avoid the bottleneck possible when all parity blocks are written to a dedicated disk, as is done in the case of RAID-4 systems. **This technique allows us to recover the data after a single disk loss.**

- For example, **if Disk 2** in Figure 3.5 **is lost,** we still have all the blocks of the third file, c1, c2, and c3, and we can recover the missing blocks for the others as follows:

$$\begin{aligned} a2 &= (a1)\ \text{XOR}\ (aP)\ \text{XOR}\ (a3) \\ b2 &= (b1)\ \text{XOR}\ (bP)\ \text{XOR}\ (b3)\ . \\ d1 &= (dP)\ \text{XOR}\ (d2)\ \text{XOR}\ (d3) \end{aligned}$$

- The system in **Figure 3.5(b) strips the data across four clusters**. The access to data is controlled by a proxy that carries out some of the functions of a RAID controller, as well as authentication and other security-related functions.

- The proxy ensures before-and-after atomicity as well as all-or-nothing atomicity for data access

Cloud interoperability; the Intercloud

- Could Interoperability, the concern that users could become hopelessly dependent on a single cloud service provider, so called **Vendor lock in.**

- **An Intercloud** ➔ a federation of clouds that cooperate to provide a better user experience.

- **Is an Intercloud feasible?**

- **Not likely at this time:**
    - There are no standards for either storage or processing.
    - The clouds are based on different delivery models.
    - The set of services supported by these delivery models is large and open; new services are offered every few months.
    - **CSPs (Cloud Service Providers) believe that they have a competitive advantage due to the uniqueness of the added value of their services.**
    - **Security is a major concern** for cloud users and an Intercloud could only create new threats.

Energy use and ecological impact

- **The energy consumption of large-scale data centers and their costs for energy and for cooling are significant.**
    - In 2006, the 6,000 data centers in the U.S consumed 61×10^9 KWh of energy, 1.5% of all electricity consumption, at a cost of $4.5 billion.

- The **energy consumed by the data centers was expected to double** from 2006 to 2011 and peak instantaneous demand to increase from 7 GW to 12 GW.
    - The greenhouse gas emission due to the data centers is estimated to increase from 116 $\times 10^9$ tones of CO_2 in 2007 to 257 tones in 2020 due to **increased consumer demand.**

- **The effort to reduce energy use is focused on computing, networking, and storage activities of a data center.** Operating efficiency of a system is captured by the performance per Watt of power.
    - The **performance of supercomputers has increased** 3.5 times faster than their operating efficiency – 7,000% versus 2,000% during the period 1998 – 2007.

- A typical **Google cluster spends most of its time** within the 10-50% CPU utilization range; there is a mismatch between server workload profile and server energy efficiency.

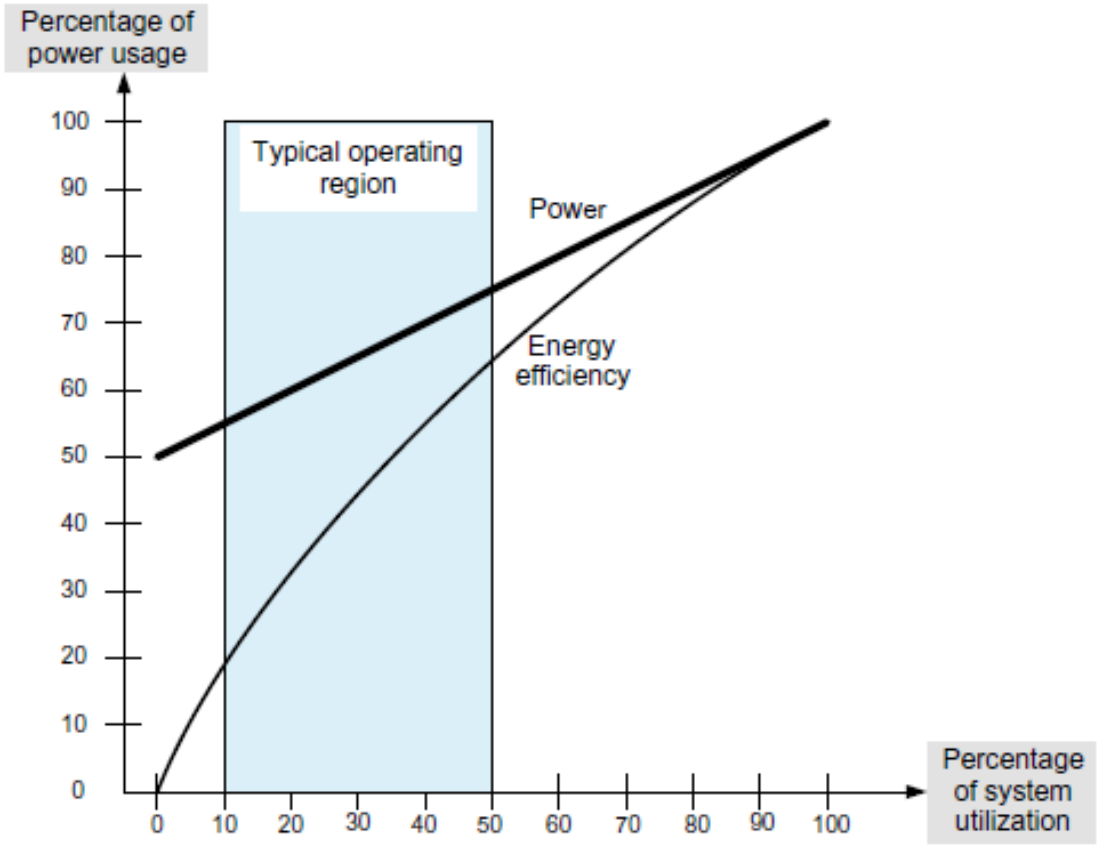


FIGURE 3.6

Even when power requirements scale linearly with the load, the energy efficiency of a computing system is not a linear function of the load; even when idle, a system may use 50% of the power corresponding to the full load. Data collected over a long period of time shows that the typical operating region for the servers at a data center is from about 10% to 50% of the load.

Service Level Agreement (SLA)

- SLA - a **negotiated contract between the customer and CSP**; can be legally binding or informal. **Objectives:**
    - Identify and define the **customer's needs and constraints** including the level of resources, security, timing, and QoS.
    - Provide a **framework for understanding**; a critical aspect of this framework is a clear definition of classes of service and the costs.
    - Simplify complex issues; **clarify the boundaries between the responsibilities of clients and CSP in case of failures.**
    - Reduce areas of conflict.
    - Encourage dialog in the event of disputes/disagreement.
    - **Eliminate unrealistic expectations**.

- Specifies the services that the customer receives, rather than how the cloud service provider delivers the services.

- **SLA records a common understanding in several areas:**
    - Services
    - Priorities
    - Responsibilities
    - Guarantees
    - Warranties

An SLA agreement usually covers :

- Services to be delivered
- Performance
- Tracking and reporting
- Problem Management
- Legal Compliance
- Customer duties and responsibilities
- Security
- Handling Confidential Information and termination

- **The common metrics specified by an SLA are service-specific**. For example, the metrics used by a call center usually are:

  - **(i) abandonment rate:** percentage of calls abandoned while waiting to be answered;
  - **(ii) average speed to answer:** average time before the service desk answers a call;
  - (**iii) time service factor:** percentage of calls answered within a definite time frame;
  - **(iv) first-call resolution**: percentage of incoming calls that can be resolved without a callback; and
  - **(v) turnaround time:** time to complete a certain task.

There are two well-differentiated phases in SLA management: the negotiation of the contract and the monitoring of its fulfillment in real time. In turn, automated negotiation has three main components:

(i) the object of negotiation, which defines the attributes and constraints under negotiation;

(ii) the negotiation protocols, which describe the interaction between negotiating parties; and

(iii) the decision models responsible for processing proposals and generating counterproposals

Responsibility sharing: between user and cloud service provider

After reviewing cloud services provided by Amazon, Google, and Microsoft, we are in a better position to understand the differences among SaaS, IaaS, and PaaS.

There is no confusion about SaaS;

- The **service provider supplies both the hardware and the application software**, and the user has direct access to these services through a Web interface and has no control over cloud resources.
- **Typical examples are** Google with Gmail, Google Docs, Google Calendar, Google Groups, and Picasa and Microsoft with the Online Services.

In the case of IaaS, the **service provider supplies the hardware (servers, storage, networks) and system software (operating systems, databases);** in addition, the provider ensures system attributes such as security, fault tolerance, and load balancing.

PaaS provides only a platform, including the hardware and system software, such as operating systems and databases;

- the **service provider is responsible for system updates, patches, and software maintenance.**
- PaaS does not allow any user control of the operating system, security features, or the ability to install applications.
- **Typical examples are** Google App Engine, Microsoft Azure, and Force.com, provided by Salesforce.com.

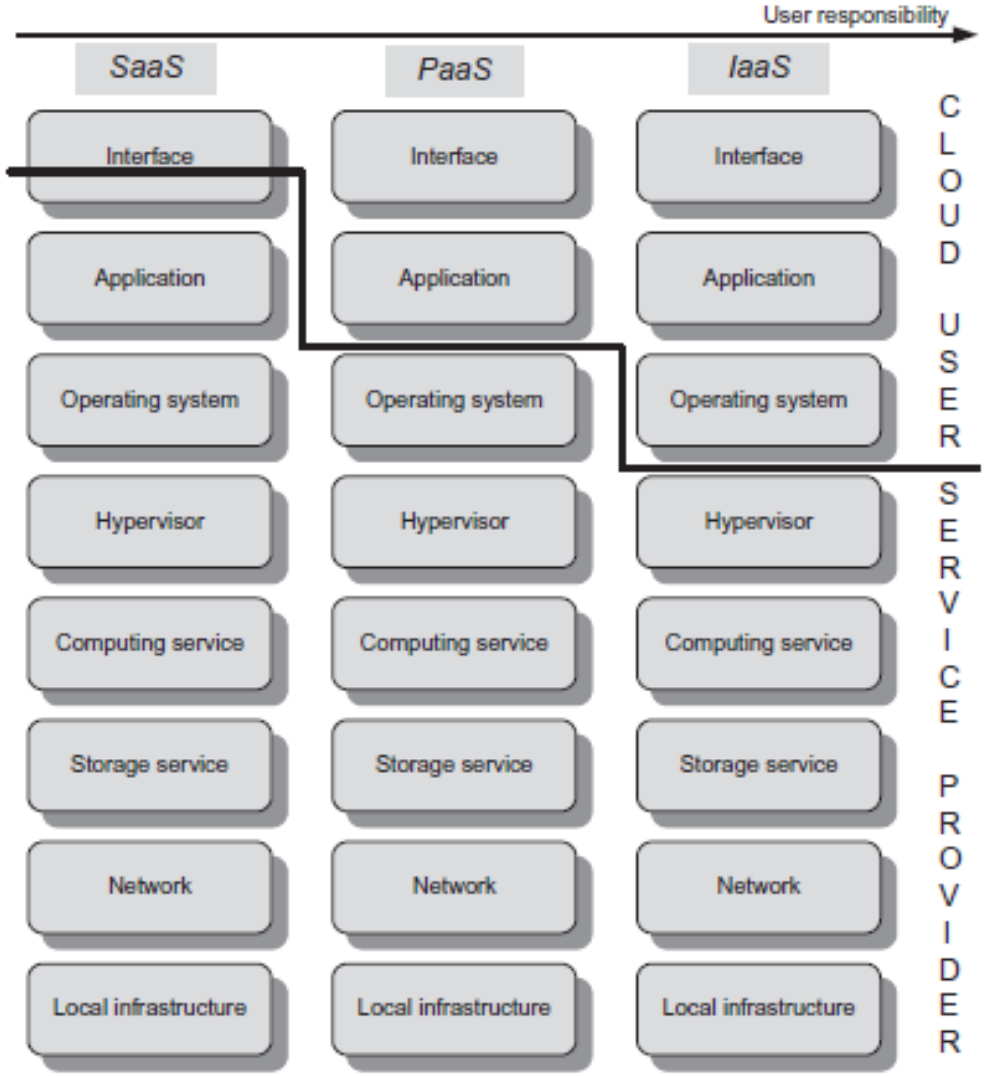


FIGURE 3.7

The limits of responsibility between a cloud user and the cloud service provider.

User experience

There have been a few studies of user experience based on a large population of cloud computing users.

- The main user concerns are **security threats**, the dependence on **fast Internet connections** that forced version updates, data ownership, and user behavior monitoring.
- **All users reported that trust in the cloud services is important,**
  - two-thirds raised the point of fuzzy boundaries of liability between cloud user and provider,
  - about half did not fully comprehend the cloud functions and its behavior, and
  - about one-third were concerned about security threats

The security threats perceived by this group of users are:

(i) abuse and villainous use of the cloud;
(ii) APIs that are not fully secure;
(iii) malicious insiders;
(iv) account hijacking;
(iv) data leaks; and
(v) issues related to shared resources.

Identity theft and privacy were major concerns for about half of the users questioned; **availability, liability, and data ownership and copyright were raised** by a third of respondents.

The suggested solutions to these problems are as follows:

- **SLAs and tools to monitor** usage should be deployed to prevent abuse of the cloud;
- **data encryption and security testing** should enhance the API security;
- **an independent security layer** should be added to prevent threats caused by malicious insiders;
- **strong authentication and authorization should be enforced** to prevent account hijacking;
- **data decryption in a secure environment should be implemented** to prevent data leakage; and
- **Compartmentalization of components and firewalls should be deployed** to limit the negative effect of resource sharing.

A broad set of concerns identified by the NIST working group on cloud security includes:

- Potential loss of control/ownership of data.
- Data integration, privacy enforcement, data encryption.
- Data remanence after deprovisioning.
- Multitenant data isolation.
- Data location requirements within national borders.
- Hypervisor security.
- Audit data integrity protection.
- Verification of subscriber policies through provider controls.
- Certification/accreditation requirements for a given cloud service.

Software Licensing

- Software licensing for cloud computing is an enduring problem without a universally accepted solution at this time.
- The license management technology is based on the old model of computing centers with licenses given on the basis of named users or as site licenses.
- This licensing technology, developed for a centrally managed environment, cannot accommodate the distributed service infrastructure of cloud computing or of grid computing
- Only very recently IBM reached an agreement allowing some of its software products to be used on EC2. Furthermore,MathWorks developed a business model for the use ofMATLAB in grid environments
- The Software-as-a-Service (SaaS) deployment model is gaining acceptance because it allows users to pay only for the services they use.
    - **There is significant pressure to change the traditional software licensing model and find nonhardware-based solutions for cloud computing.**

Applications & Paradigms:

Cloud computing is very attractive to the users:

- **Economic reasons.**
  - low infrastructure investment.
  - low cost - customers are only billed for resources used.
- **Convenience and performance.**
  - Application developers enjoy the advantages of **a just-in-time infrastructure**; they are free to design an application without being concerned with the system where the application will run.
  - The execution time of **compute-intensive and data-intensive applications** can, potentially, be reduced through parallelization.
  - Cloud computing is also beneficial for the providers of computing cycles - it typically leads to **a higher level of resource utilization.**

Challenges for Cloud Computing:

- **Performance isolation -** nearly impossible to reach in a real system, especially **when the system is heavily loaded.**

- **Reliability -** major concern; server **failures expected when a large number of servers cooperate for the computations.**

- Many applications consist of multiple stages; in turn, each stage may involve multiple instances running in parallel on the systems of the cloud and communicating among them. Thus, **efficiency, consistency, and communication scalability are major concerns for an application developer.**

- **Data storage plays a critical role in the performance of any data-intensive application**; the organization of the storage, the storage location, and the storage bandwidth must be carefully analyzed to lead to optimal application performance

- Cloud infrastructure exhibits **latency and bandwidth fluctuations** which affect the application performance.

- **Performance considerations limit the amount of data logging**; the ability to identify the source of **unexpected results and errors** is helped by frequent logging.

Existing and new application opportunities:

- Three broad categories of existing applications:
    - **Processing pipelines.**
    - **Batch processing systems.**
    - **Web applications.**

Data Processing Applications:

- **Indexing** large datasets created by web crawler engines.
- **Data mining** - searching large collections of records to locate items of interests.
- **Image processing** .
    - Image conversion, e.g., enlarge an image or create thumbnails.
    - Compress or encrypt images.
- **Video transcoding** from one video format to another, e.g., from AVI to MPEG.
- **Document processing**.
    - Convert large collections of documents from one format to another, e.g., from Word to PDF.
    - Encrypt documents.
    - Use Optical Character Recognition to produce digital images of documents.

Batch processing applications:

- **Generation of daily, weekly, monthly, and annual activity reports** for retail, manufacturing, other economical sectors.

- **Processing, aggregation, and summaries of daily transactions** for financial institutions, insurance companies, and healthcare organizations.

- **Processing billing and payroll records.**

- **Management of the software development**, e.g., nightly updates of software repositories.

- **Automatic testing and verification of software and hardware systems**.

Web applications:

- Sites for **online commerce**.
- Sites with **a periodic or temporary presence**.
    - Conferences or other events.
    - Active during a particular season (e.g., the Holidays Season) or income tax reporting.

Architectural styles for Cloud applications

Cloud Computing is based on the **client-server paradigm**.

- **Stateless servers** - view a client request as **an independent transaction and respond to it**; the client is not required to first establish a connection to the server.
  - **A stateless server keeps no state information.**
  - **A stateless system is simple, more robust and scalable.**
  - For example, **a basic Web server is stateless**;
  - It responds to an HTTP request **without maintaining a history of past interactions with the client.**

- The client, a browser, is also stateless since it sends requests and waits for responses. The **Hypertext Transfer Protocol (HTTP)** used by a browser to communicate with the Web server is a request/response application protocol.

- The use of **TCP ensures reliable delivery of large objects** but exposes the Web servers to denial-of-service attacks when malicious clients fake attempts to establish a TCP connection and force the server to allocate space for the connection

- **Stateful servers, on the other hand, do store session state**. They may, therefore, keep track of which clients have opened which files, current read and write pointers for files, which files have been locked by which clients, etc.)

- The **Common Object Request Broker Architecture (CORBA**) was developed in the early 1990s to allow networked applications developed in **different programming languages and running on systems with different architectures** and system software to work with one another.

- The **Simple Object Access Protocol (SOAP)** is an application protocol developed in 1998 for Web applications; its message format is based on the Extensible Markup Language (XML). S

- The **Web Services Description Language (WSDL**) was introduced in 2001 as an XML-based grammar to describe communication between endpoints of a networked application.

- **Representational State Transfer (REST)** is a style of software architecture for distributed hypermedia systems. REST supports client communication with stateless servers. It is platform- and language independent, supports data caching.

Different Cloud Architectures

Overview:

- In this, we will find **several sample diagrams of cloud-based solution architectures** that you can build with the RightScale platform using both **public and/or private cloud infrastructures.**
    - Most of these architectures can be **built using existing Server Templates** that are available in the **Multi Cloud Marketplace.**
- Each application is **unique and will have a custom set of requirements**.
- The purpose of the system architecture diagrams is to provide you with real-world examples that you can **use as base reference architectures when you design your own custom system architectures in the cloud.**

https://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.html

Things to Consider:

- There are several factors that you need to take into consideration before **designing your own cloud-based systems architecture**, particularly if you're considering **a multi-cloud/region architecture.**

COST - Before you architect your site/application and start launching servers, you should clearly understand the SLA and pricing models associated with your cloud infrastructure(s).

- For example, in AWS, data transferred between servers **inside of the same datacenter (Availability Zone) is free**,

- whereas **communication between servers in different datacenters within the same cloud (EC2 Region) is cheaper** than communication between servers in different clouds or on-premise datacenters.

COMPLEXITY - Before you construct a highly customized hybrid cloud solution architecture, make sure you properly understand the actual requirements of your application, SLA, etc. Simplified architectures will always be easier to design and manage.

- For example, a **system architecture that is distributed across multiple clouds (regions) introduces complexity at the architecture level** and may require changes at the application level to be more latency-tolerant and/or be able to communicate with a database that's migrated to a different cloud for failover purposes.

SPEED - The cloud gives you more **flexibility to control the speed or latency of your site/application**. For example, you could launch different instance types based on your application's needs.

CLOUD PORTABILITY–

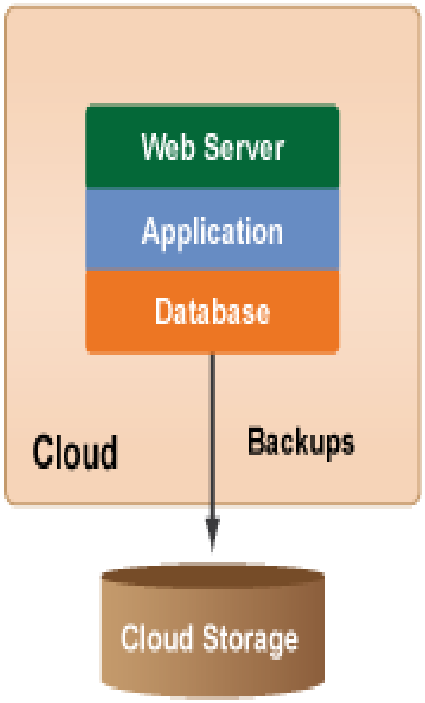
- Although it might be easier to use one of the cloud provider's tools or services, such as a load balancing or database service,

- it's important to realize that if and when you need to move that particular tier of your architecture to another cloud provider, you will need to modify your architecture accordingly.

SECURITY–

For MultiCloud system architectures, it's important to realize that cross-cloud/region communication is performed over the public Internet and may introduce security concerns that will need to be addressed **using some type of data encryption or VPN technology.**

Single "All-in-one" Server:

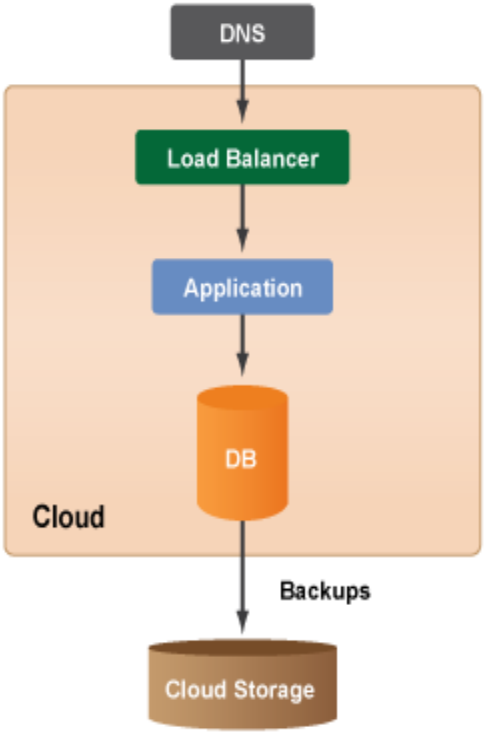


- Use one of the "All-in-one" Server Templates, such as the
- **LAMP (Linux, Apache, MySQL, PHP) Server Template**
    - To launch a single server that contains a web server (Apache), as well as your application (PHP) and database (MySQL).
- You'll find a **collection of simple "All-in-one" Server Templates in the Multi Cloud Marketplace**, which are useful for new Right Scale users and basic demos.

Single Cloud Site Architectures:

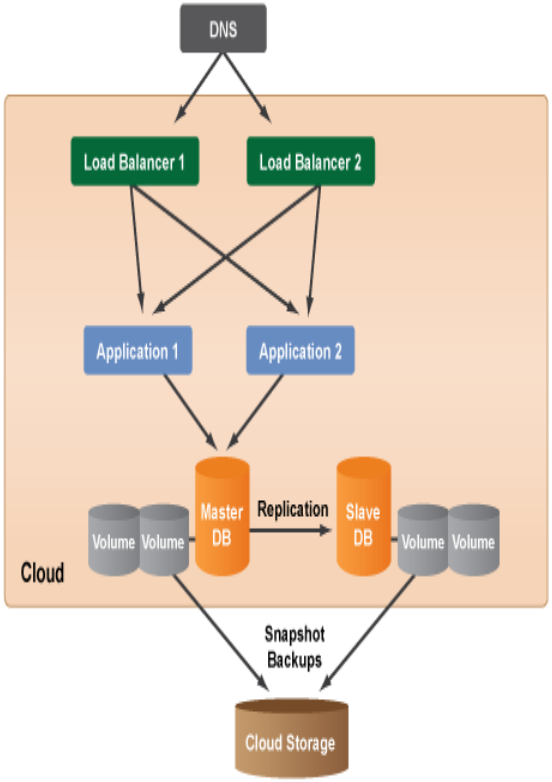
Non-Redundant 3-Tier Architecture

In a standard **three-tier website architecture**, there is at least one dedicated server in each tier of the system architecture. (Load Balancing Server, Application Server, Database Server)



- If you are only testing the interactivity between each tier of your architecture, you may want to **use a non-redundant system architecture** to save on costs and resources.
- Since it is a non-redundant system architecture it is primarily used for basic test and development purposes.
- In the example diagram , **there are dedicated servers for each tier of the application/site.**
- A non-redundant architecture is **not recommended for production environments**.

Redundant 3-Tier Architecture



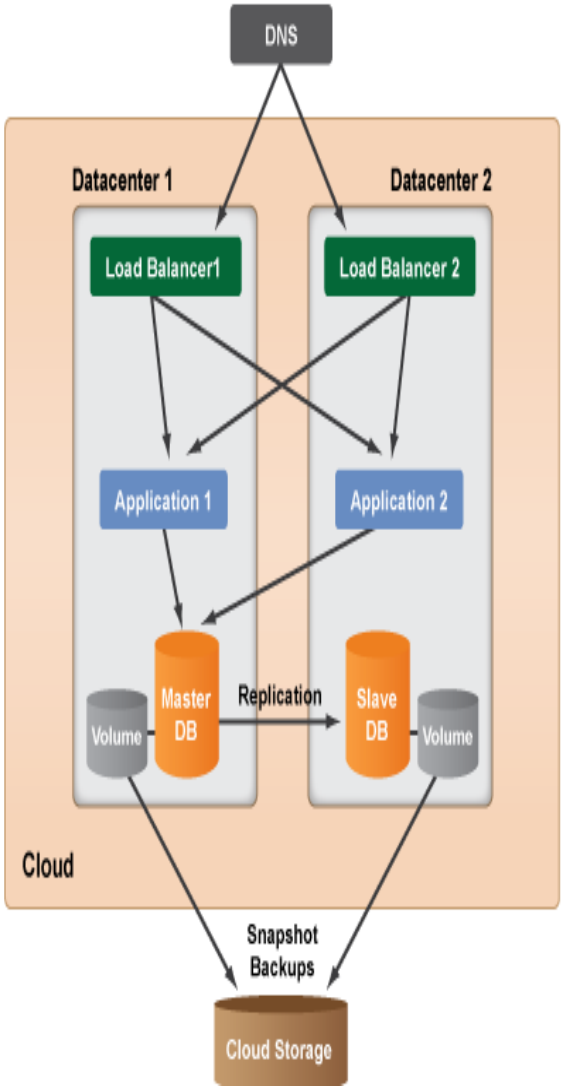
- Any production environment that is launched in the cloud should also have a redundant architecture **for failover and recovery purposes.**

- Typically, you will use a **Server Array for your application tier to take advantage of auto scaling** in the cloud,
- However there may be some scenarios where your application is not designed to auto scale.
- In such cases, you can still create **a redundant multi-tier architecture** where you have redundancy at each tier of your reference architecture.

- In the example below, there are **two load balancer servers**, **two application servers**, as well as **master and slave database servers**.

- A redundant architecture will help **protect your site/application from system downtime.**

Multi-Datacenter Architecture

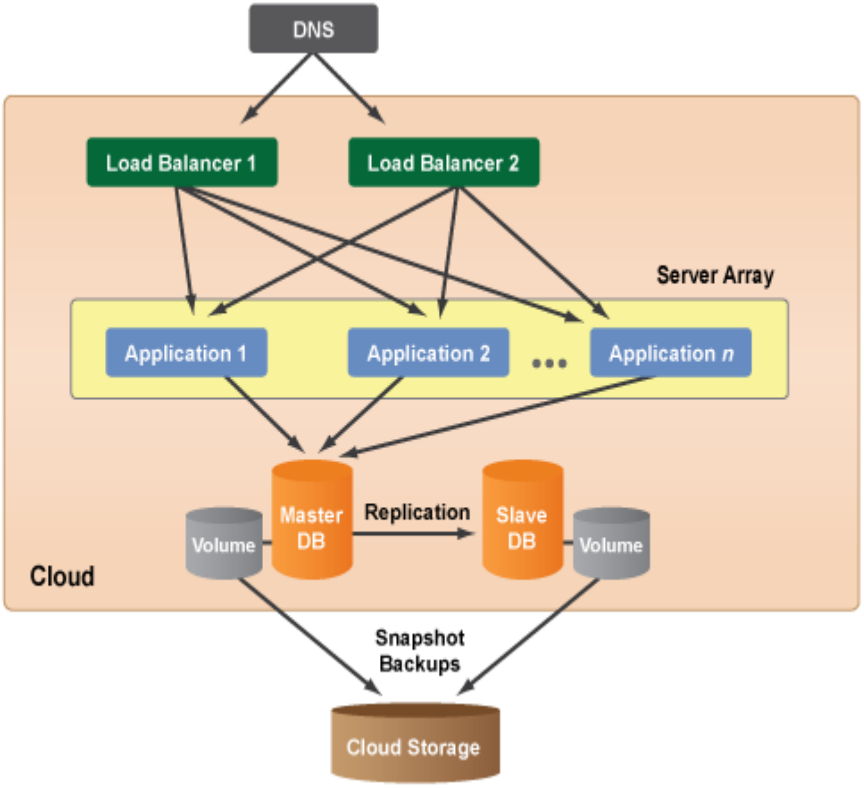


- If your cloud infrastructure supports **multiple** datacenters (or zones), it's recommended that you spread your system architecture across multiple datacenters **to add another layer of redundancy and protection.**

- Each datacenter in a cloud is **designed to be an isolated segment inside the same geographical cloud.** So if a power failure occurs in one datacenter, the other datacenters will be unaffected.
  - For example, **within a cloud/region there may be several resource pools** called availability zones and datacenters.

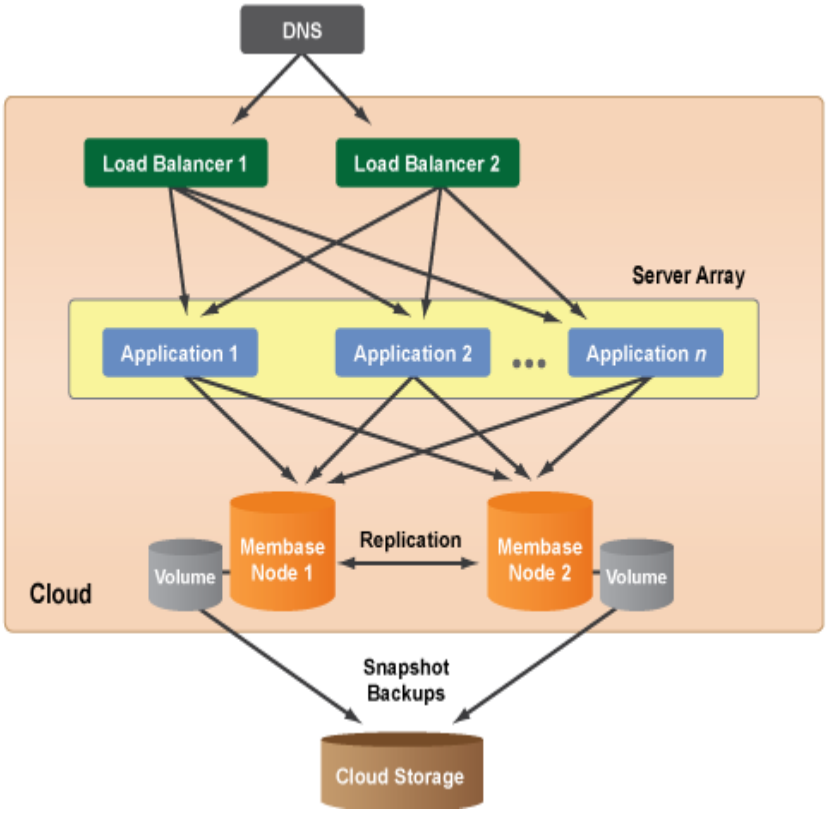
- The benefit of using multiple datacenters is to **protect your entire site/application from being negatively affected by some type of**
  - network/power failure,
  - lack of available resources, or service outage that's specific to a particular datacenter.

Autoscaling Architecture



- One of the key benefits of the cloud is the **ability to horizontally scale (i.e. grow or shrink the number of running server resources) as the demands of your application/site change** over time.
- With RightScale, you can use Server Arrays to set up a particular tier of your architecture to **autoscale based on predefined alert conditions.**
- Autoscaling is most commonly used for the application tier of your cloud reference architecture.

Scalable Architecture with Membase



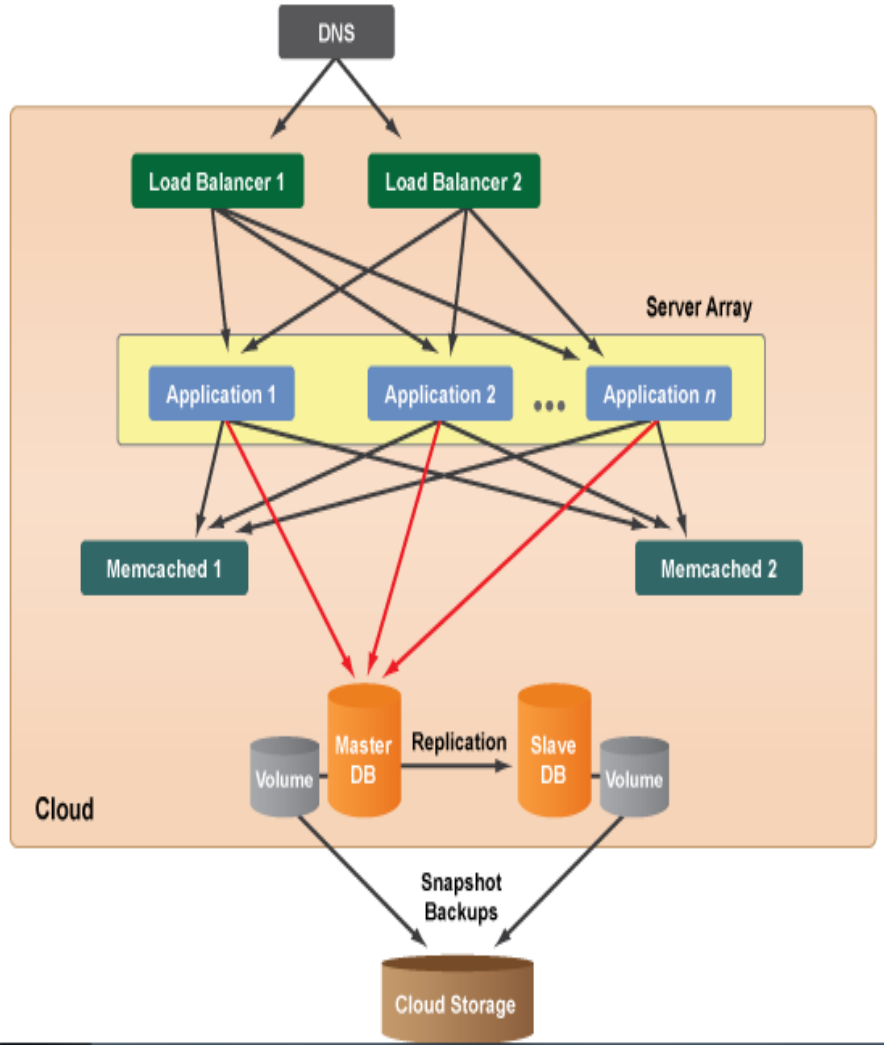
If you do not want to use a **Master-Slave MySQL setup,**

you could also use **Membase (Couchbase)** nodes for your database tier,

- which is a distributed NoSQL database, **which replicates data across all of the Membase nodes.**

- If you are using the Enterprise edition you can attach volumes to each node (shown below), but the Community Edition doesn't support the use of volumes.

Scalable Multi-Tier Architecture with Memcached

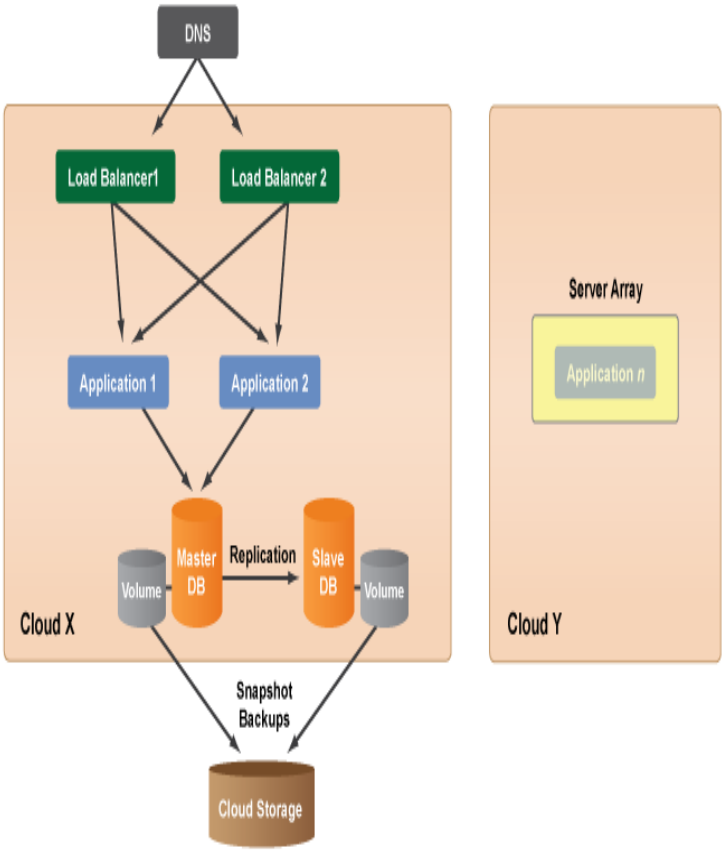


- For **applications/sites that require lots of reads from the database and serve a lot of static content**, you might want to add a Memcached layer to your cloud system architecture to offload a read-heavy database.

Memcached is an open source distributed memory object caching system that's ideal for speeding up dynamic web applications by alleviating(reduce) database load.

In the example diagram below, the application servers can still make writes to the database, but many commonly used objects will be retrieved from one of the Memcached servers instead of the Master-DB server.

Hybrid Cloud Site Architectures



- Another way that you can protect your site/application in the cloud is to design a hybrid cloud site architecture that leverages **multiple public/private cloud infrastructures** or dedicated hosted servers.

- One of the key benefits of the RightScale platform is **cloud portability**, where you **can use the same assets (ServerTemplates, RightScripts, etc.)** to launch identically functioning servers into multiple public/private clouds.

- **Avoid cloud lock-in** and design a solution architecture that takes advantage of multiple cloud resource pools instead of just a single cloud.

- Similarly, you can also **design a hybrid cloud architecture where servers in a cloud can communicate with dedicated servers that are hosted in an internal/external datacenter.**

Cloud Applications

- **Cloud computing for Healthcare**
- **Cloud Computing for Education**
- **Cloud Computing for Government**
- **Cloud Computing for Transpiration Systems**
- **Cloud Computing for Mobile Communication**
- **Cloud Computing for Manufacturing Industry**
- **Cloud Computing for Energy Systems**

Cloud computing for Healthcare

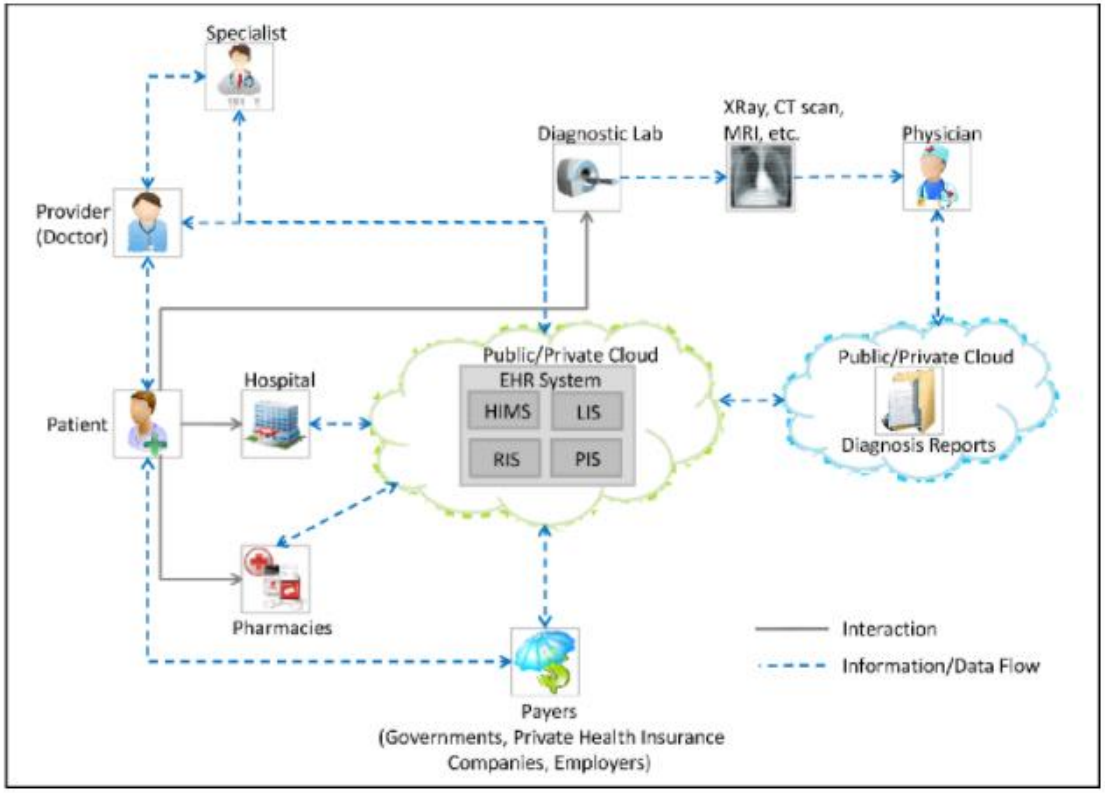


Fig. 1. Application of cloud computing environments to the healthcare ecosystem.

- **Electronic health record (EHR)**
- **Personal Health Record(PHR)**
- **Hospital Information Management System (HIMS)**
- **Laboratory information system(LIS)**
- **Pharmacy Information System (PIS)**

- **Health Care Ecosystem :**

- Patients can access their own health information, stored in **Personal Health Record(PHR)**

- An integrated record that may even be a family Health Record.

- History and Information stored in the **Cloud using SaaS Application.**

- SaaS application can streamline the **Admissions, Care and Discharge** process by eliminating redundant data collection and entry.

Cloud Computing for Education

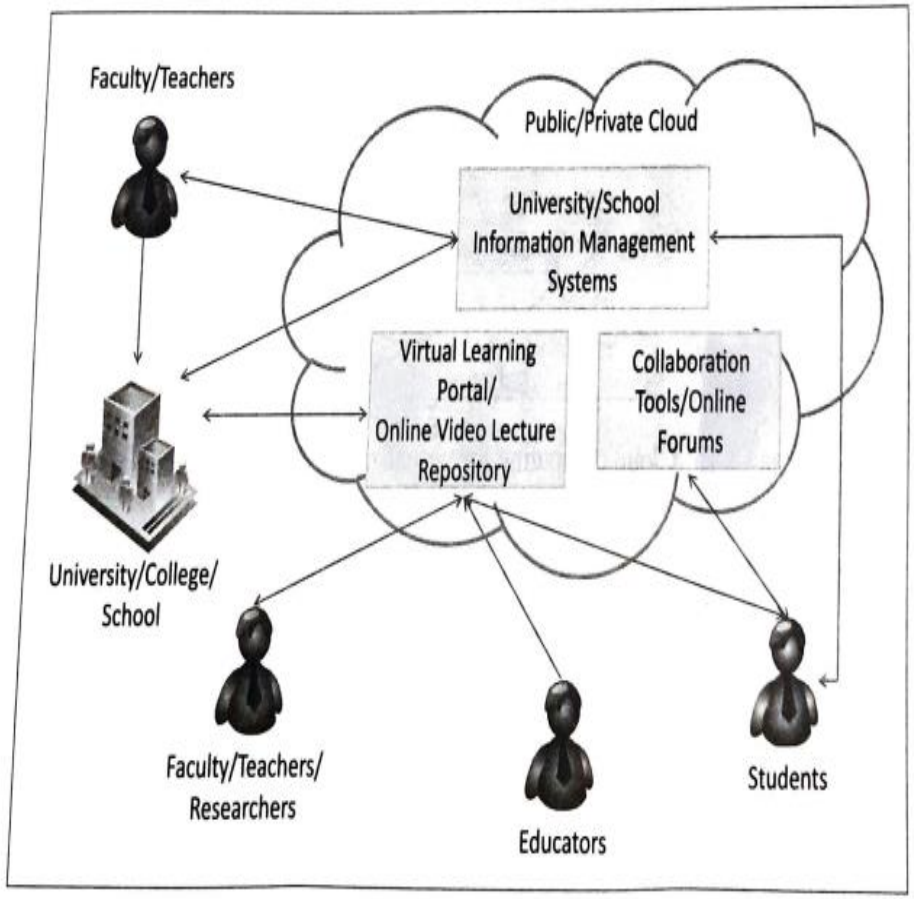


Figure 1.15: Cloud computing for education

- To **improve the quality of Online Education System.**
- Online forums, can help the student **discuss common problems and seek guidance** from experts.
- **University, College and School**s can use **Cloud Based Information Management System**.
- To improve administrative efficiency, Online/Distance Education programs to **track the progress of the students, collect feedback.**
- **MOOC, LMS, E-Learning** provide high quality education to the learner community using Cloud Technology.

Cloud Computing for Government

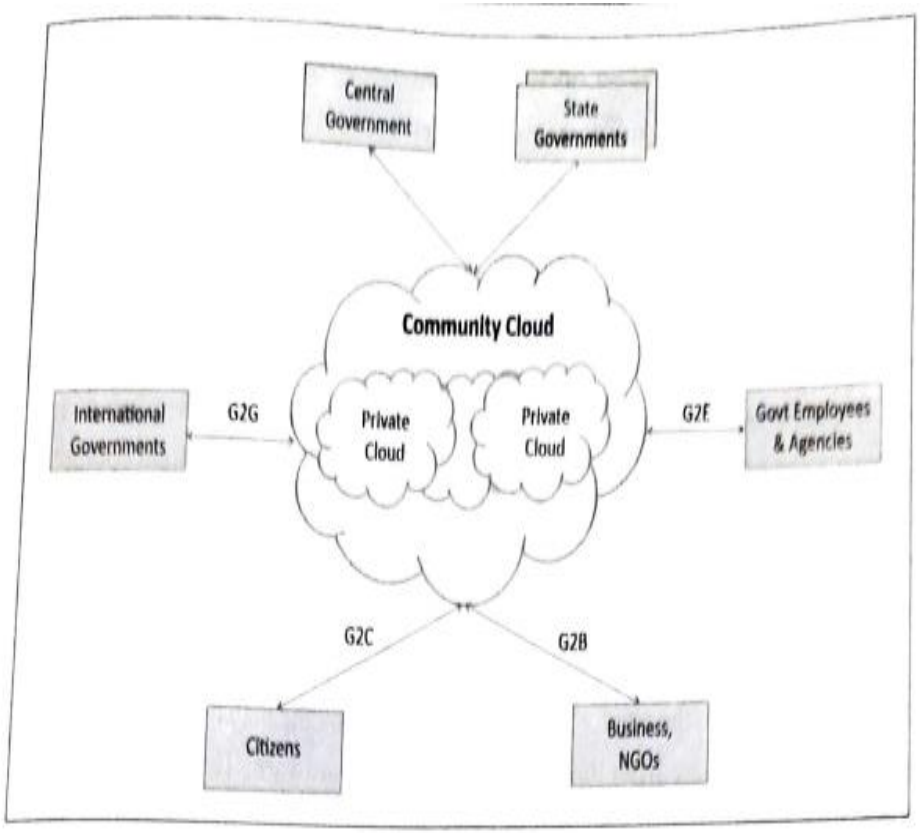


Figure 1.14: Cloud computing for government

- Cloud Computing can play a significant role for **improving the Efficiency and Transparency** of government operations.

- **E-Government System** can improve delivery of services to Citizens, Government Employees and Agencies etc.
- Also to improve various **Government Schemes** and **Policy Information Process.**

- **Public Services such as:** Public Transport Reservation System, Vehicle Registration, Income tax Filling, Electricity/Water Bill ,Birth/Death Certificate etc..

Cloud Computing for Transpiration Systems

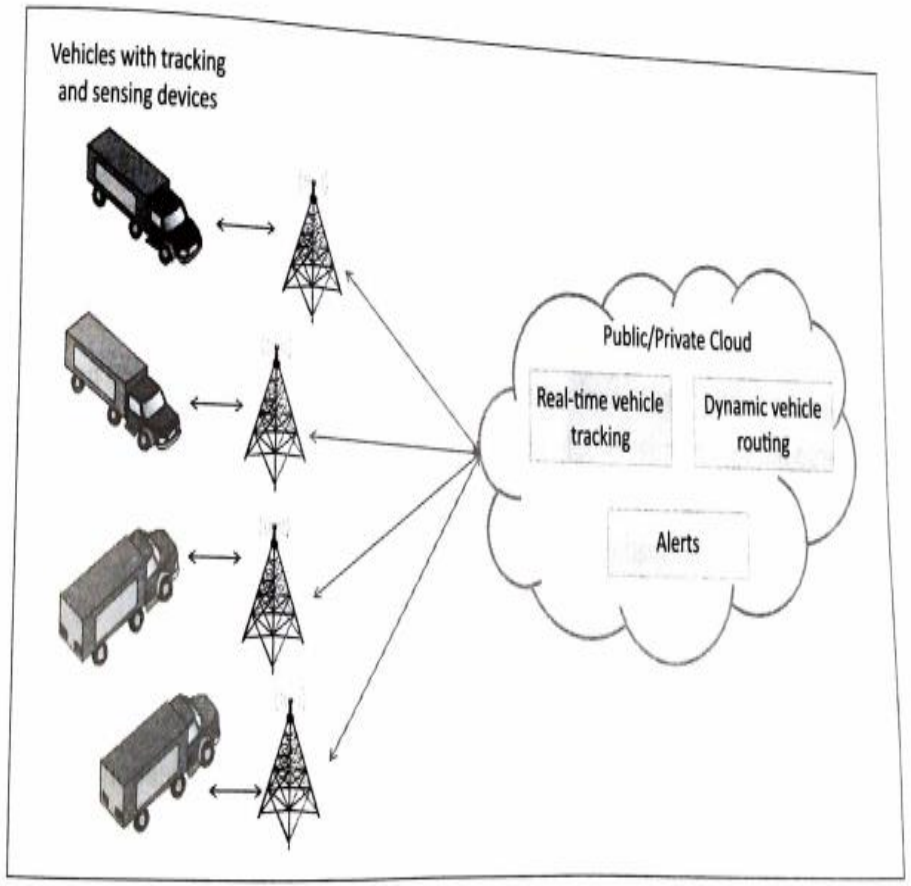


Figure 1.12: Cloud computing for transportation systems

- Intelligence Transportation System(ITS) can provide useful information , Route guidance, **Dynamic Vehicle routing**, customer pickup and delivery etc.
- Cloud base framework, for real time fresh food/Fruits/Vegitable supply **tracking and monitoring system.**
- Its also provides Environmental information like, **Temperature and Humidity .**
- Analysis and Interpretation of the data on the environmental conditions.

Cloud Computing for Mobile Communication

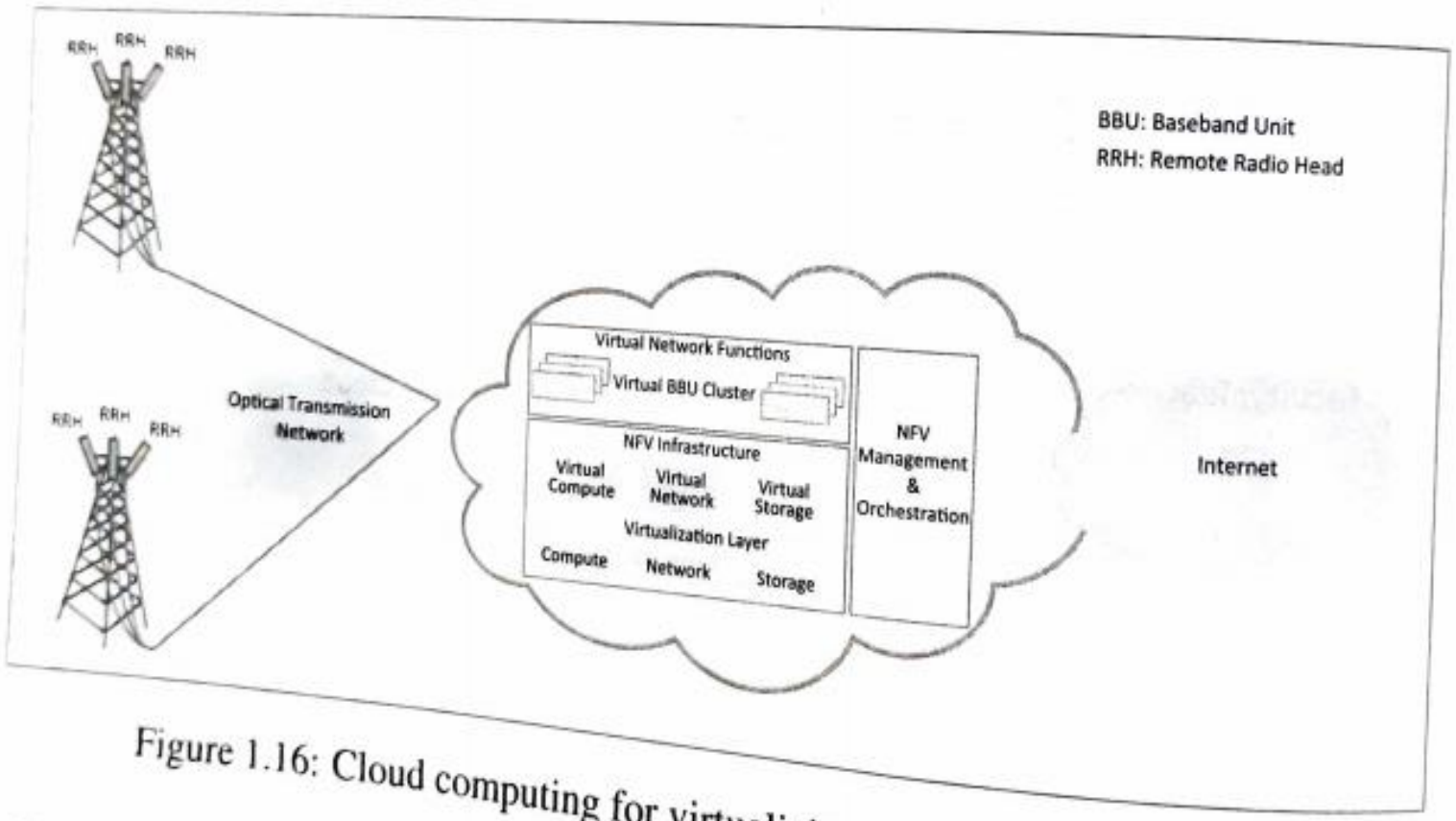


Figure 1.16: Cloud computing for virtualizing radio access network

Figure 1.16 shows a use case of cloud
access network