

Introduction

- Cloud is not new name..it has been evolved...
- Before emerging the cloud computing, there was Client/Server computing which is basically a centralized storage in which all the software applications, all the data and all the controls are resided on the server side.
- If a single user wants to access specific data or run a program, he/she need to connect to the server and then gain appropriate access, and then he/she can do his/her business.
- Then after, distributed computing came into picture, where all the computers are networked together and share their resources when needed.
- On the basis of above computing, there was emerged of cloud computing concepts that later implemented.
- At around in 1961, John MacCharty suggested in a speech at MIT that computing can be sold like a utility, just like a water or electricity. It was a brilliant idea, but like all brilliant ideas, it was ahead if its time, as for the next few decades, despite interest in the model, the technology simply was not ready for it.
- But of course time has passed and the technology caught that idea and after few years we mentioned that:
- *In 1999, **Salesforce.com** started delivering of applications to users using a simple website.* The applications were delivered to enterprises over the Internet, and this way the dream of computing sold as utility were true.
- *In 2002, **Amazon** started Amazon Web Services, providing services like storage, computation and even human intelligence. However, only starting with the launch of the Elastic Compute Cloud in 2006 a truly commercial service open to everybody existed.*
- *In 2009, **Google Apps** also started to provide cloud computing enterprise applications.*

Distributed Computing

- Is a computing system consists of multiple computers or processor machines connected through a network, which can be homogeneous or heterogeneous, but run as a single system.
- The CPUs in a distributed system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network.
- The heterogeneity in a distributed system supports any number of possible configurations in the processor machines, such as mainframes, PCs, workstations, and minicomputers.
- The goal of distributed computing is to make such a network work as a single computer.
- Distributed computing systems are advantageous over centralized systems due to..
 - Scalability
 - Redundancy or replication

High Performance Computing(HPC)

- A pool of processors connected /networked with other resources like memory, storage, and input and output devices, and the deployed software is enabled to run in the entire system of connected components.
- Processor machines can be of homogeneous or heterogeneous type.
- The legacy meaning of HPC- Supercomputers; it is not true in present-day computing scenarios.
- Examples of HPC -a small cluster of desktop computers or personal computers (PCs) to the fastest supercomputers.
- HPC systems are normally found in those applications where it is required to use or solve scientific problems.

Parallel Computing

- Parallel computing is one of the facets of HPC.
- A set of processors work cooperatively to solve a computational problem.
- These processor machines or CPUs are mostly of homogeneous type.

- In serial or sequential computers, the following apply:
 - It runs on a single computer/processor machine having a single CPU.
 - A problem is broken down into a discrete series of instructions.
 - Instructions are executed one after another.
- In parallel computing, since there is simultaneous use of multiple processor machines, the following apply:
 - It is run using multiple processors (multiple CPUs).
 - A problem is broken down into discrete parts that can be solved concurrently.
 - Each part is further broken down into a series of instructions.
 - Instructions from each part are executed simultaneously on different processors.
 - An overall control/coordination mechanism is employed.

Cluster Computing

- Consists of a set of the same or similar type of processor machines connected using a dedicated network infrastructure.
- All processor machines share resources such as a common home directory and have a software such as a message passing interface (MPI) implementation installed to allow programs to be run across all nodes simultaneously.
- This is also a kind of HPC category. The individual computers in a cluster can be referred to as nodes.
- The reason to realize a cluster as HPC is due to the fact that the individual nodes can work together to solve a problem larger than any computer can easily solve. And, the nodes need to communicate with one another in order to work cooperatively and meaningfully together to solve the problem in hand.

Grid Computing

- The computing resources in most of the organizations are underutilized but are necessary for certain operations.
- The idea of grid computing is to make use of such non-utilized computing power by the needy organizations, and thereby the return on investment (ROI) on computing investments can be increased.
- It is a network of computing or processor machines managed with a kind of software such as middleware, in order to access and use the resources remotely.
- The managing activity of grid resources through the middleware is called grid services. Grid services provide access control, security, access to data including digital libraries and databases, and access to large-scale interactive and long-term storage facilities.

- Grid computing is more popular due to the following:
 - Its ability to make use of unused computing power, and thus, it is a cost-effective solution (reducing investments, only recurring costs)
 - As a way to solve problems in line with any HPC-based application
 - Enables heterogeneous resources of computers to work cooperatively and collaboratively to solve a scientific problem

Cloud Computing

- The computing trend moved toward cloud from the concept of grid computing, particularly when large computing resources are required to solve a single problem, using the ideas of computing power as a utility and other allied concepts.
- The potential difference between grid and cloud is that grid computing supports leveraging several computers in parallel to solve a particular application,
- while cloud computing supports leveraging multiple resources, including computing resources, to deliver a unified service to the end user.
- In cloud computing, the IT and business resources, such as servers, storage, network, applications, and processes, can be dynamically provisioned to the user needs and workload.
- In addition, while a cloud can provision and support a grid, a cloud can also support nongrid environments, such as a three-tier web architecture running on traditional or Web 2.0 applications.

Need for Cloud Computing

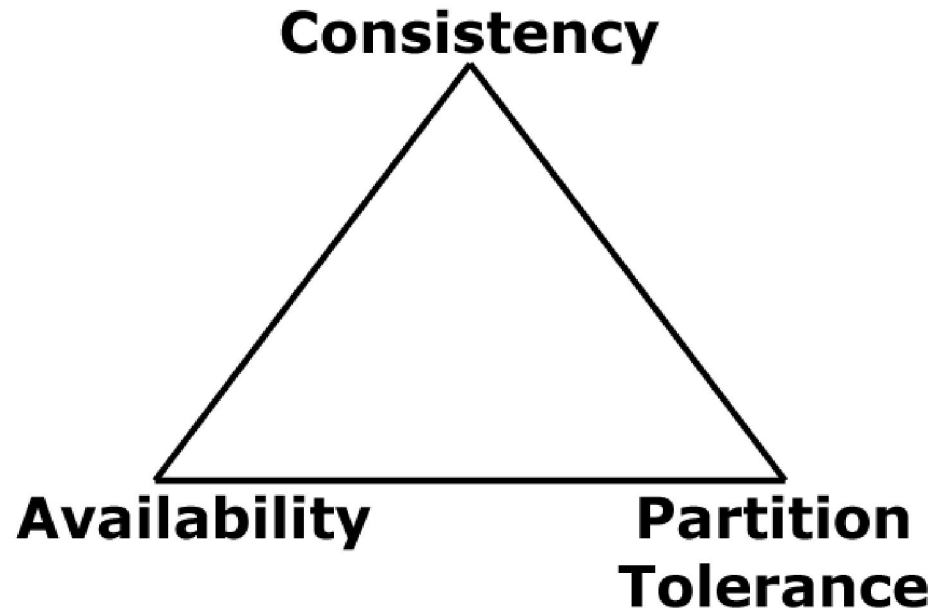
- The main reasons for the need and use of cloud computing are - convenience and reliability.
- In the past, we used to save files in USB flash drive, external hard drive, or CD and bring that device to a different place.
- Instead, saving a file to the cloud (e.g. Dropbox/OneDrive/Google Drive) ensures that we will be able to access it with any computer that has an Internet connection.
- The cloud also makes it much easier to share a file with friends, making it possible to collaborate over the web.
- There is always a risk that someone may try to gain access to our personal data, and therefore, it is important to choose an access control with a strong password and pay attention to any privacy settings for the cloud service.

CAP Theorem

- The CAP theorem was formulated by Prof. Eric A. Brewer in 2000 – long before the advent of the term *cloud computing*.
- It is also known as Brewer's Theorem.
- The acronym CAP stands for **Consistency**, **Availability**, and **Partition Tolerance**
- In a distributed system -
- **Consistency** means that all nodes see the same data at the same time.
- **Availability** means a guarantee that every request receives a response about whether it was successful or failed
- **Partition Tolerance** means the system continues to operate despite arbitrary message loss.

Cont..

- Brewer discovered that a distributed system can satisfy any two of these guarantees at the same time, but not all three.



Cont..

- The system properties C, A and P can be regarded as gradual quantities, i.e. the *availability* is high if the system has short response times, and low if the system has slow response times.
- With regard to *consistency*, the system has a consistent state in an instant (In ACID principle of relational database management systems) or after a certain time frame of inconsistency (In BASE principle of NoSQL datastores)
- Any given distributed system is located on one of the sides (CA), (CP) or (AP) of our triangle. In real world 24/7 applications high availability is always required. So it can be questioned if a (CP) system makes any sense and if our choice is only between (CA) and (AP).

Cont..

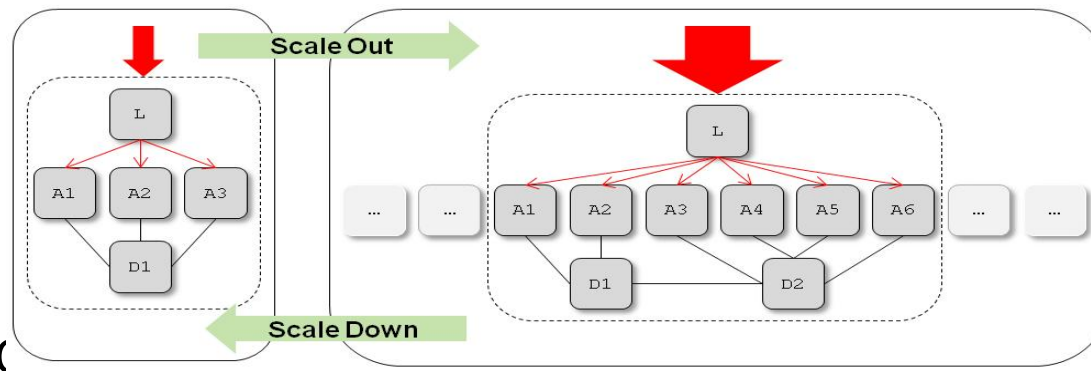
- DNS(Domain Name System) – It is an (AP) system.
- Availability is really high (do you remember the last downtime of your DNS server?)
- The same is true for the partition tolerance.
- But the consistency is not always given at an instant. It can sometimes take *days* for a DNS entry to travel its way to the root hierarchy and before being visible by all other nodes.

Cont..

- RDMS – It is a (CA) system.
- Availability and consistency of a single node is very high.
- Building clustered systems with data replication leads to a decrease of availability because a consistent transaction takes up more time. The more nodes you have to synchronize their data, the longer such transactions will take.

Cont..

- Cloud platforms rely on horizontal scaling (scale-out), i.e. the load is distributed over a lot of single nodes, which may run on cheap commodity hardware.
- Cloud infrastructure has to cope with regular outages of single nodes (partition tolerance). High availability is a must have, since the end user will not tolerate response times above a certain level any more these days before trying the next competitor.
- So a cloud platform (or at least large parts of it) is an (AP) system.



- Because of the (AP) system.
- Being not strictly consistent does not mean your data will be corrupt all the time.
- There are weaker concepts of consistency than ACID. But these concepts are often acceptable in your application.

Definition – Cloud Computing

- The formal definition of cloud computing comes from the National Institute of Standards and Technology (NIST):

“Cloud computing is a model for enabling ubiquitous, 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”.

5-4-3 Principles of Cloud Computing

- The 5-4-3 principles put forth by NIST describe
 - (a) the five essential characteristic features that promote cloud computing,
 - (b) the four deployment models that are used to narrate the cloud computing opportunities for customers while looking at architectural models, and
 - (c) the three important and basic service offering models of cloud computing.

Essential Characteristics

1. On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
2. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and personal digital assistants [PDAs]).

3. Elastic resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify the location at a higher level of abstraction (e.g., country, state, or data center).

4. Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
5. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Deployment models

1. Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
2. Public cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
3. Community cloud: The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.

4. Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Private Cloud

- In the business perspective, making the correct decision regarding the deployment model is very important and should be selected based on the needs, requirements, budget, and security.
- According to NIST, private cloud can be defined as the cloud infrastructure that is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units).
- It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
- Private cloud can be deployed using open source tools such as Openstack, Eucalyptus.
- **Characteristics**: Secure, Central control, Weak SLAs
- **Suitability**: Organizations that - require a separate cloud for their personal or official use, have a sufficient fund to manage/maintain cloud, consider data security to be important, want autonomy and complete control over the cloud, have a less number of users, have prebuilt infrastructure
- **Not suitable**: Organizations that - have high user base, have financial constraints, do not have prebuilt infrastructure, do not have sufficient manpower to maintain and manage the cloud
- **Advantages**: Small in size and is easy to maintain, High level of security and privacy to the user, Controlled by the organization
- **Disadvantages**: Budget is a constraint, Loose SLAs.

Public Cloud

- According to NIST, the public cloud is the cloud infrastructure that is provisioned for open use by the general public.
- It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
- Well known providers – Amazon AWS, MS Azure, GCP, IBM, etc.
- **Characteristics**: Highly scalable, Affordable, Less secure, Highly available, Stringent SLAs
- **Suitability**: Organizations that – have large user base, varying resource requirements, no prebuilt environment, have financial constraint.
- **Not suitable**: Where - security is important, autonomy is expected, third party reliability is not preferred
- **Advantages**: No need of establishing/maintaining infrastructure, Less costly, Strict SLAs, No user limit, highly scalable
- **Disadvantages**: Security is an issue, Privacy and organizational autonomy are not possible

Community cloud

- According to NIST, the community cloud is the cloud infrastructure that is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- Either the organizations or a single organization may collectively maintain the cloud.
- **Characteristics:** partially secure, cost effective
- **Suitability:** Want to establish a private cloud but have financial constraint, Do not want to complete maintenance responsibility of the cloud, Want to establish the cloud in order to collaborate with other clouds, Want to have a collaborative cloud with more security features than the public cloud
- **Advantages:** low-cost private cloud, collaborative work on the cloud, sharing of responsibilities among the organization, better security than the public cloud.
- **Disadvantages:** Autonomy of an organization is lost, Security features are not as good as the private cloud

Hybrid Cloud

- According to NIST, the hybrid cloud can be defined as the cloud infrastructure that is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.
- This is aimed at combining the advantages of private and public clouds.
- The usual method of using the hybrid cloud is to have a private cloud initially, and then for additional resources, the public cloud is used.
- The hybrid cloud can be regarded as a private cloud extended to the public cloud to utilize the power of the public cloud by retaining the properties of the private cloud.
- One of the popular examples for the hybrid cloud is Eucalyptus.
- **Characteristics:** Scalable, Partially secure, Stringent SLAs, Complex cloud management
- **Suitability:** Organizations that - want the private cloud environment with the scalability of the public cloud, require more security than the public cloud
- **Not Suitable:** Organizations that - consider security as a prime objective, will not be able to handle hybrid cloud management
- **Advantages:** power of both the private and public clouds, highly scalable, provides better security than the public cloud.
- **Disadvantages:** The security features are not as good as the public cloud, Managing a hybrid cloud is complex, It has stringent SLAs.

Service Offering Models

- IaaS
- PaaS
- SaaS

IaaS: Infrastructure as a Service

- cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.

When to use IaaS:

- Startups and small companies may prefer IaaS to avoid spending time and money on purchasing and creating hardware and software.
- Larger companies may prefer to retain complete control over their applications and infrastructure, but they want to purchase only what they actually consume or need.
- Companies experiencing rapid growth like the scalability of IaaS, and they can change out specific hardware and software easily as their needs evolve.
- Anytime you are unsure of a new application's demands, IaaS offers plenty of flexibility and scalability.

Cont.

- IaaS is fully self-service for accessing and monitoring computers, networking, storage, and other services.
- IaaS allows businesses to purchase resources on-demand and as-needed instead of having to buy hardware.
- These cloud servers are typically provided to the organization through a dashboard or an API, giving IaaS clients complete control over the entire infrastructure.
- As opposed to SaaS or PaaS, IaaS clients are responsible for managing aspects such as applications, runtime, OSes, middleware, and data.

IaaS Advantages

IaaS offers many advantages, including:

- The most flexible cloud computing model
- Easy to automate deployment of storage, networking, servers, and processing power
- Hardware purchases can be based on consumption
- Clients retain complete control of their infrastructure
- Resources can be purchased as-needed
- Highly scalable

IaaS Characteristics

- Resources are available as a service
- Cost varies depending on consumption
- Services are highly scalable
- Multiple users on a single piece of hardware
- Organization retain complete control of the infrastructure
- Dynamic and flexible

IaaS Limitations/Concerns

- Security - While the customer is in control of the apps, data, middleware, and the OS platform, security threats can still be sourced from the host or other virtual machines (VMs)
- Legacy systems operating in the cloud - While customers can run legacy apps in the cloud, the infrastructure may not be designed to deliver specific controls to secure the legacy apps.
- Internal resources and training
- Multi-tenant security - the vendor is required to ensure that other customers cannot access data deposited to storage assets by previous customers

PaaS: Platform as a Service

- PaaS delivers a framework for developers that they can build upon and use to create customized applications.

When to use PaaS:

- To streamline workflows when multiple developers are working on the same development project.
- PaaS is particularly beneficial if you need to create customized applications. This cloud service also can greatly reduce costs and it can simplify some challenges that come up if you are rapidly developing or deploying an app.

PaaS Advantages

- Simple, cost-effective development and deployment of apps
- Scalable
- Highly available
- Developers can customize apps without the headache of maintaining the software
- Significant reduction in the amount of coding needed
- Automation of business policy
- Easy migration to the hybrid model

PaaS Characteristics

- Builds on virtualization technology, so resources can easily be scaled up or down as your business changes
- Provides a variety of services to assist with the development, testing, and deployment of apps
- Accessible to numerous users via the same development application
- Integrates web services and databases

PaaS Limitations/Concerns

- Data Security
- Integrations
- Vendor lock-in
- Customization of legacy systems
- Runtime issues
- Operational limitations

SaaS: Software as a Service

- Utilizes the internet to deliver applications, which are managed by a third-party vendor, to its users.
- A majority of SaaS applications run directly through your web browser, which means they do not require any downloads or installations on the client side.

When to use SaaS:

- Start-ups or small companies that need to launch ecommerce quickly and don't have time for server issues or software
- Short-term projects that require quick, easy, and affordable collaboration
- Applications that aren't needed too often, such as tax software
- Applications that need both web and mobile access

SaaS Advantages

- Reduce the time and money spent on tedious tasks such as installing, managing, and upgrading software.

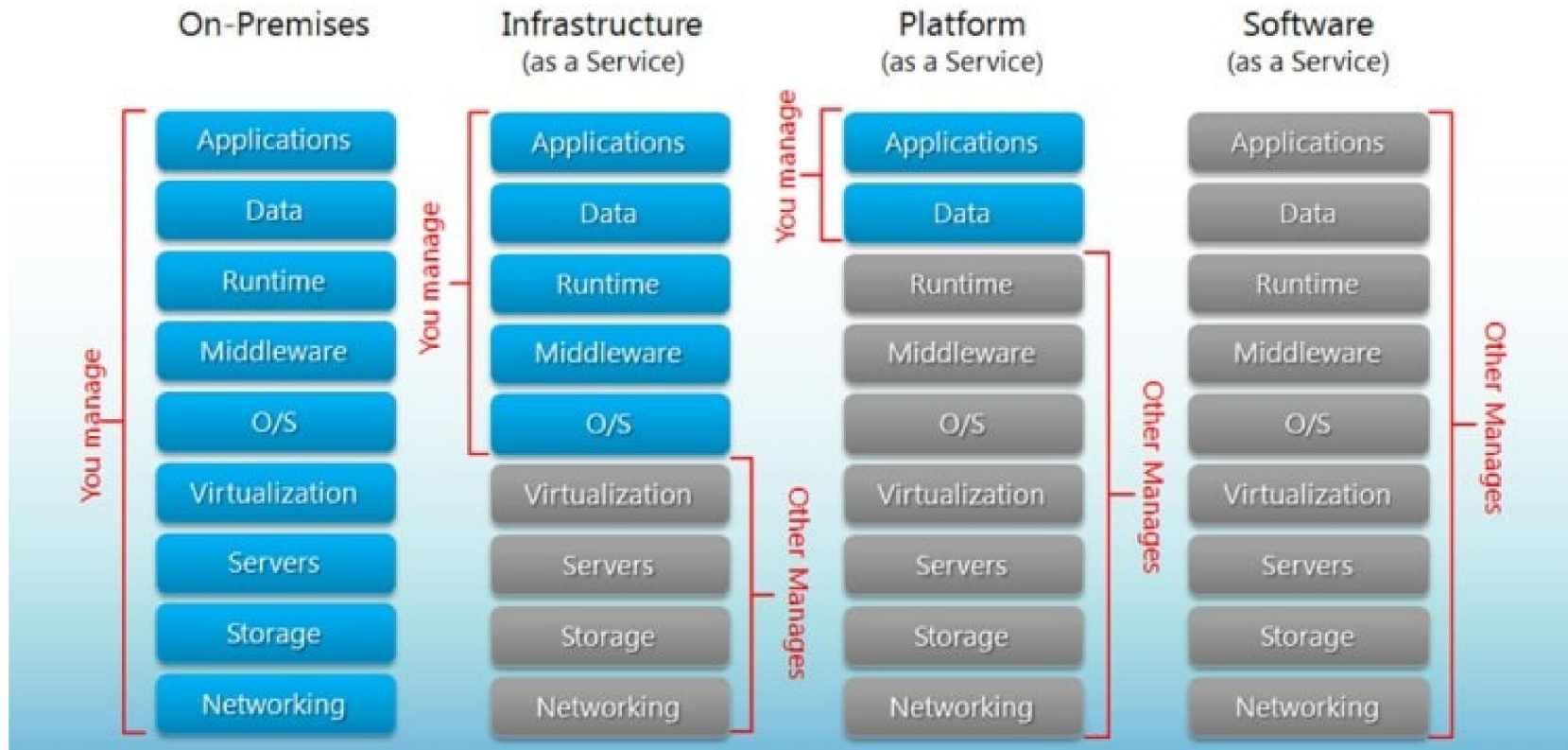
SaaS Characteristics

- Managed from a central location
- Hosted on a remote server
- Accessible over the internet
- Users not responsible for hardware or software updates

SaaS Limitations/Concerns

- Interoperability
- Vendor lock-in
- Lack of integration support
- Data security
- Customization
- Lack of control
- Feature limitations
- Performance and downtime

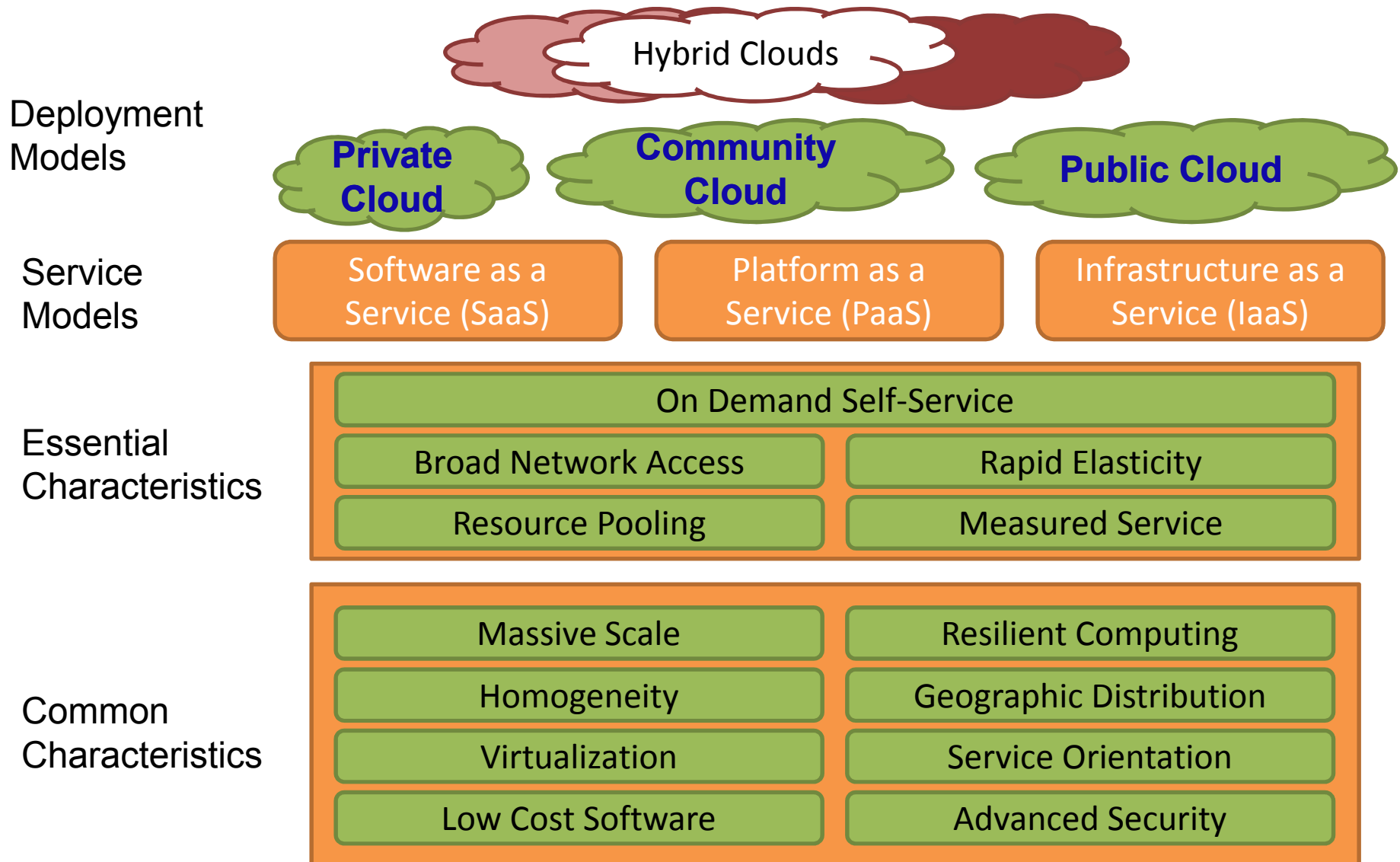
Service Offering Models



Examples

- SaaS examples: Google Apps(Gmail, Google Talk, Google Calendar, Google Docs, Google Videos and Google Cloud Connect), Salesforce(CRM), Dropbox, DocuSign, Slack(messages), GoToMeeting.
- PaaS examples: AWS Elastic Beanstalk, Windows Azure (mostly used as PaaS), Google App Engine, OpenShift – container platform.
- IaaS examples: AWS EC2, Rackspace Cloud, Google Compute Engine (GCE).

The NIST Cloud Def. Framework



NIST reference architecture

- A template description of the architecture, probably defined at different levels of abstraction
 - Highly abstract showing different functionalities
 - Lower level showing methods performing specific task
- Vendor-neutral description
- A conceptual model for discussing the technical requirements and operations of cloud computing
- A blueprint to guide developers in the design of (cloud) services and applications
 - Blueprint: compositions of interconnected services implementing reusable logic for building applications), list of functions and their interfaces (APIs), descriptions of their interactions

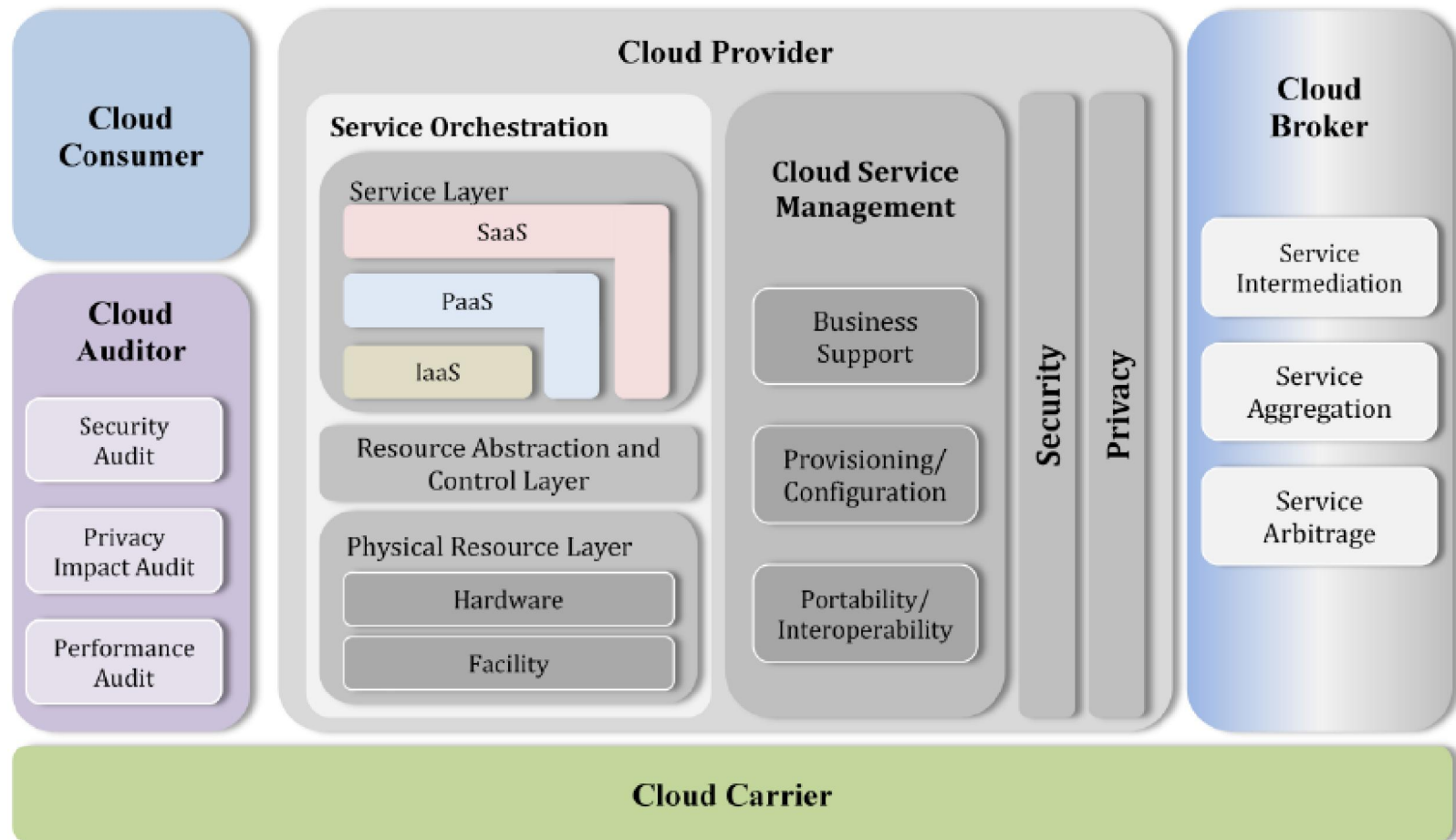
Cont.

- Actors and roles: core individuals or users with key responsibilities in system function
- Architectural components for managing and providing cloud services for
 - Deployment
 - Orchestration
 - Management
 - Security
 - Privacy

Actors and Roles

- Individuals or organizations with key roles
 - Consumer: acquires and uses services
 - Provider: the purveyor of services
 - Broker: intermediate between consumer – provider, they hide complexity of services or create new services
 - Auditor: independent performance, security monitoring and assessment of cloud services
 - Carrier: provides connectivity and transport of data and services between providers and consumers

Conceptual Reference Model



Cloud Consumer

- Browses the service catalogue of the provider
- Requests services depending on activities, usage scenarios
- Sets up service contracts with the providers
- May be billed for the service
 - SaaS consumers may be billed based on number of users, time of use, net bandwidth, storage volume
 - IaaS, PaaS consumers may be billed according to processing, storage, network resources, number of VMs, http calls, number of IPs used, net bandwidth, storage volume
- Consumers need SLAs to specify their performance requirements to be fulfilled by the provider (however SLAs are offered by cloud producers and in most cases aren't negotiable)

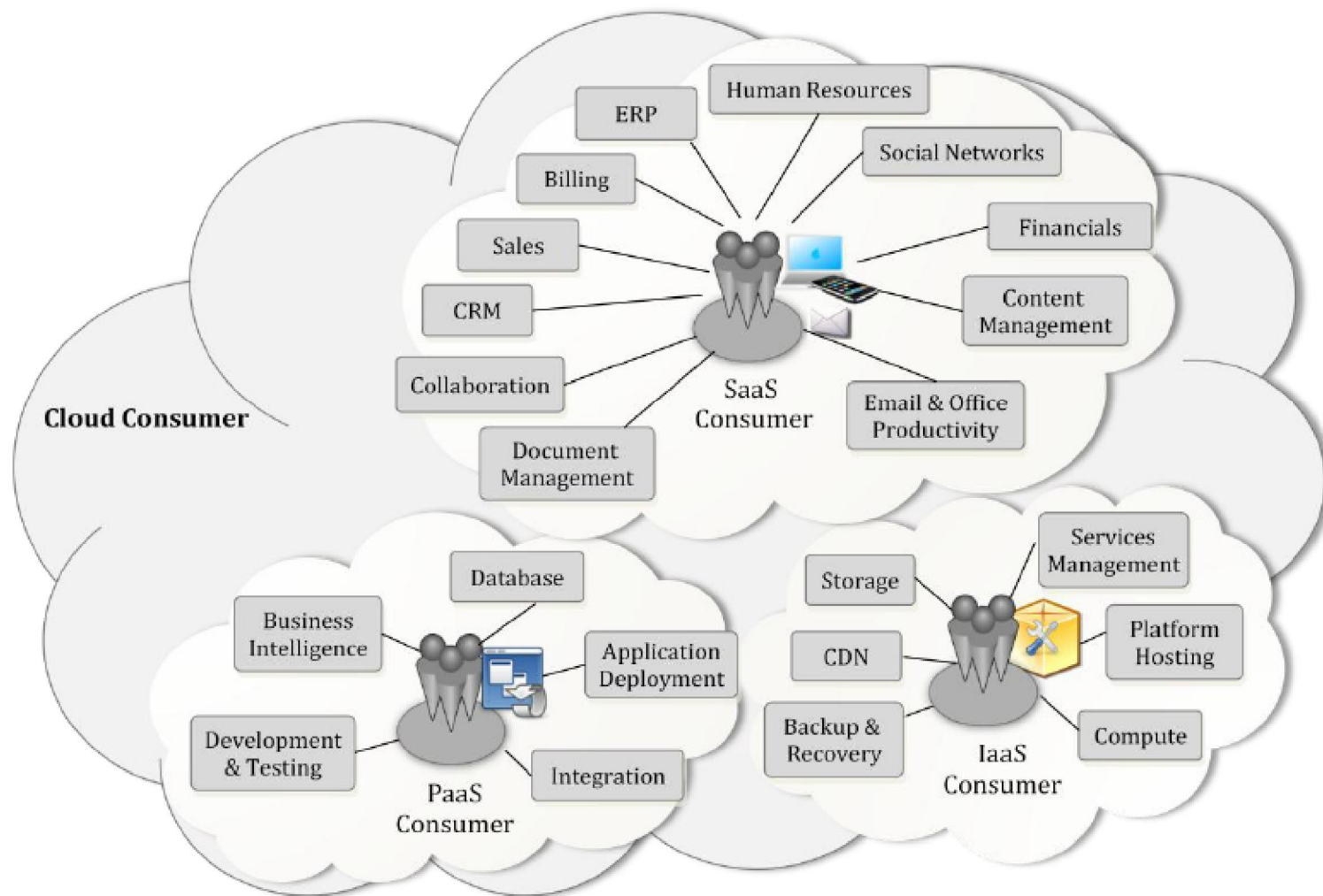
Service Level Agreements (SLAs)

- Contracts that are negotiated and agreed between provider and customers so to locate/reserve resources to satisfy consumers' requirements with efficiency and optimally resource and service usage
- To guarantee an agreed SLA, the auditor must be capable of measuring and monitoring relevant metrics (e.g., service availability, network metrics, storage metrics)
- Different SLA models must be considered for IaaS, PaaS and SaaS as each model sets different requirements
 - SLAs can be defined clearly for IaaS;
 - for PaaS and SaaS SLAs are still vague and difficult to be defined as these refer to higher levels of functionality but, can be agreed between providers / customers based on application requirements (business case) and business level plan

SLAs for IaaS

Parameter	Description
CPU capacity	CPU speed for Virtual Machines (VMs)
Memory size	Cash memory size for VM
Boot time	Time for VM to be ready for use
Storage	Storage size of data
Scale up	Max of VMs for one user
Scale down	Min number of VMs for one user
Scale up time	Time to increase number of VMs
Scale down time	Time to decrease number of VMs
Availability	Uptime of service in specific time

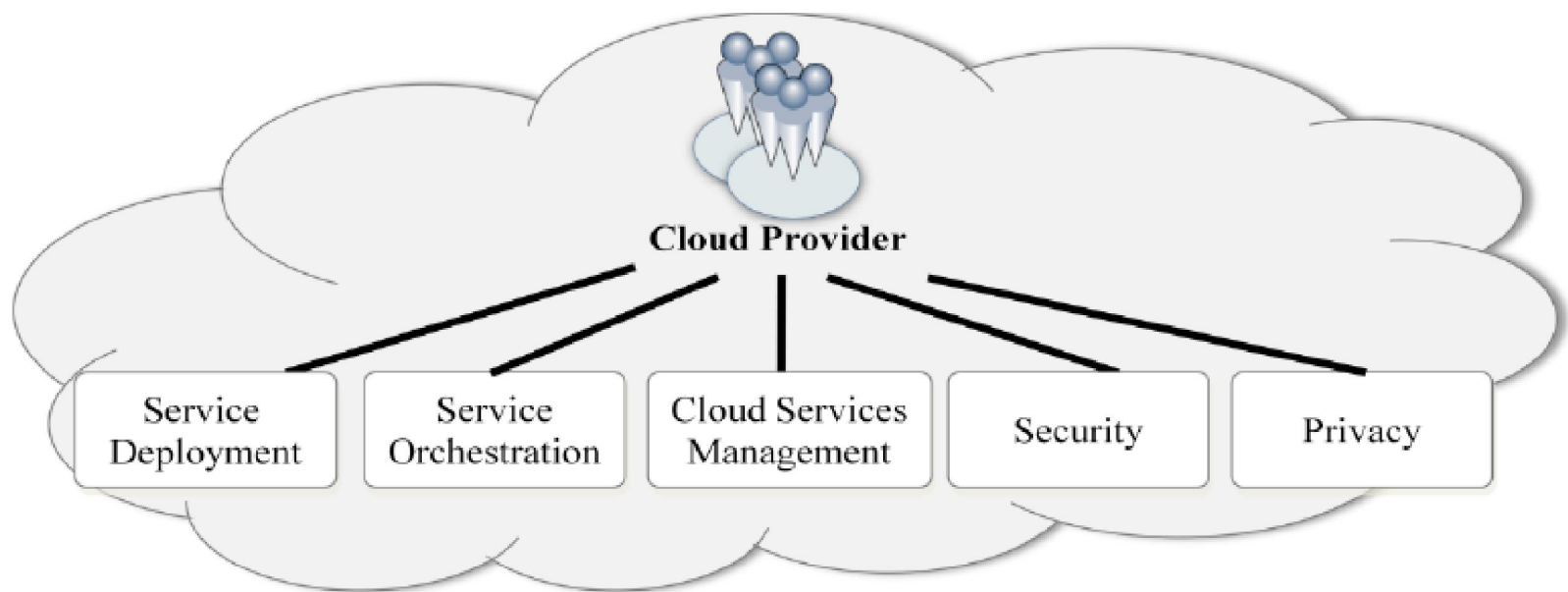
Examples of Cloud Services



Cloud Provider

- Acquires and manages the computing infrastructure
- Runs the cloud software, makes services available to interested parties
- Makes arrangements / contracts with consumers
- May also list SLAs i.e. Promises to consumers or limitations and obligations that consumers must accept
- Provider's pricing policy and SLAs are not negotiable in most cases

Responsibilities of Cloud Provider



Scope of Control (Provider)

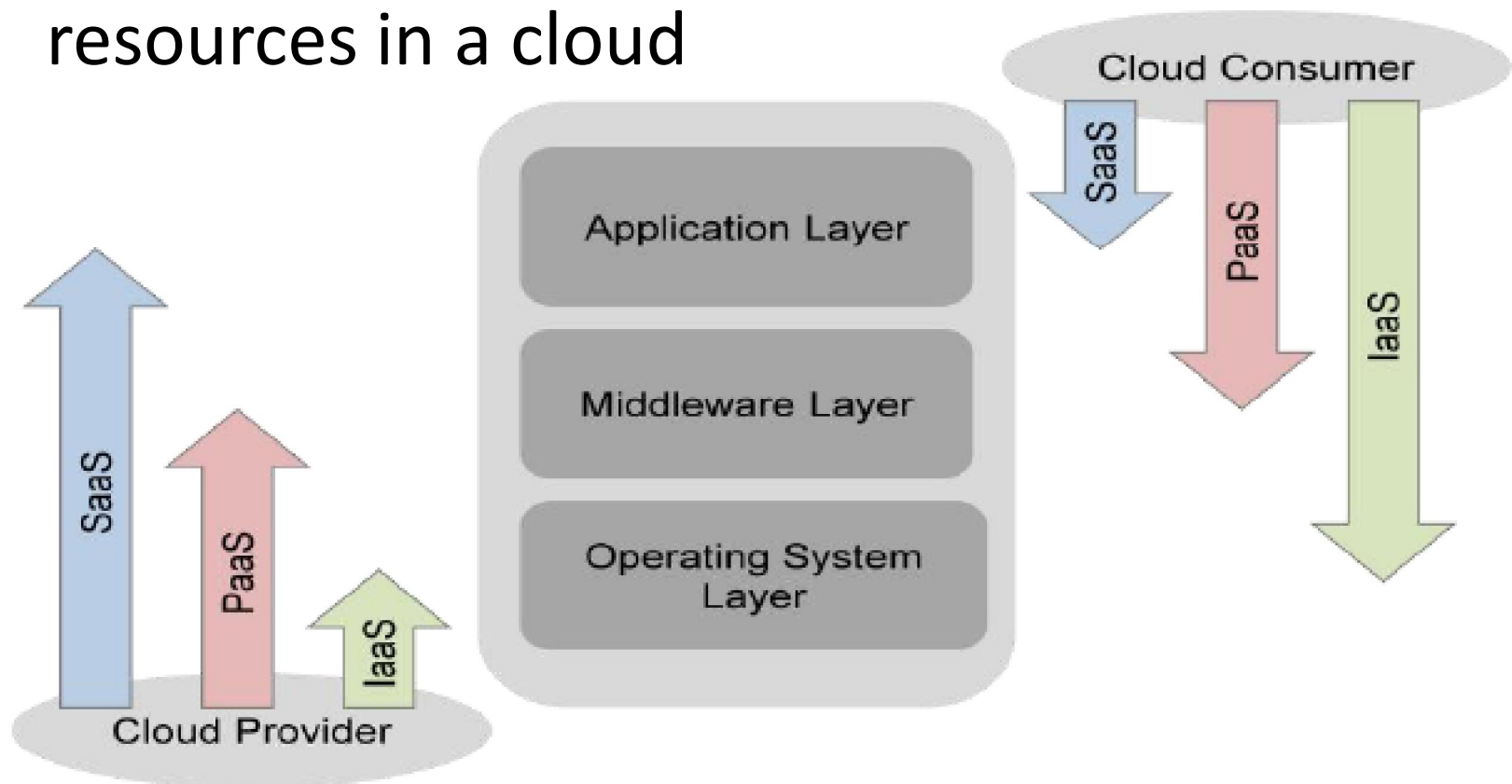
- Application layer: end-user apps and services used by SaaS consumers, installed/managed by PaaS consumers and SaaS providers
- Middleware layer (VM layer): provides building blocks for app development (libraries, dbms, Java VMs), used by PaaS consumers, installed/maintained/managed by PaaS providers, hidden from SaaS consumers
- OS layer: operating system VMs and drivers, hidden from SaaS /PaaS consumers, controlled by IaaS providers, used by IaaS consumers. An IaaS provider may allow multiple OS's as VMs

Scope of Control

- SaaS: Consumers have only limited administrative control of the applications and services
- PaaS: The provider manages infrastructure and provides tools of deployment of applications; the consumer has control over the application but limited / no access to the infrastructure (e.g. OS, servers, storage, drivers)
- IaaS: The provider acquires physical resources (servers, network, storage) and runs the software to make these available to IaaS, PaaS consumers through VMs; consumers have control over virtual software components (OS, network)

Scope of Control

- Provider and consumer share the control of resources in a cloud



Cloud Auditor

- Performs independent examination of cloud service controls and express opinion / issues evaluation
 - Ideally, have a contractual clause enabling 3rd parties to assess cloud operations
 - To determine the extend to which cloud operations are implemented/executed as planned and agreed
- Auditors objective is to verify conformance to standards (e.g. OCCl) or to security, privacy controls, performance, conformance to SLAs etc.
 - Issue security, privacy, performance audits

Cloud Broker

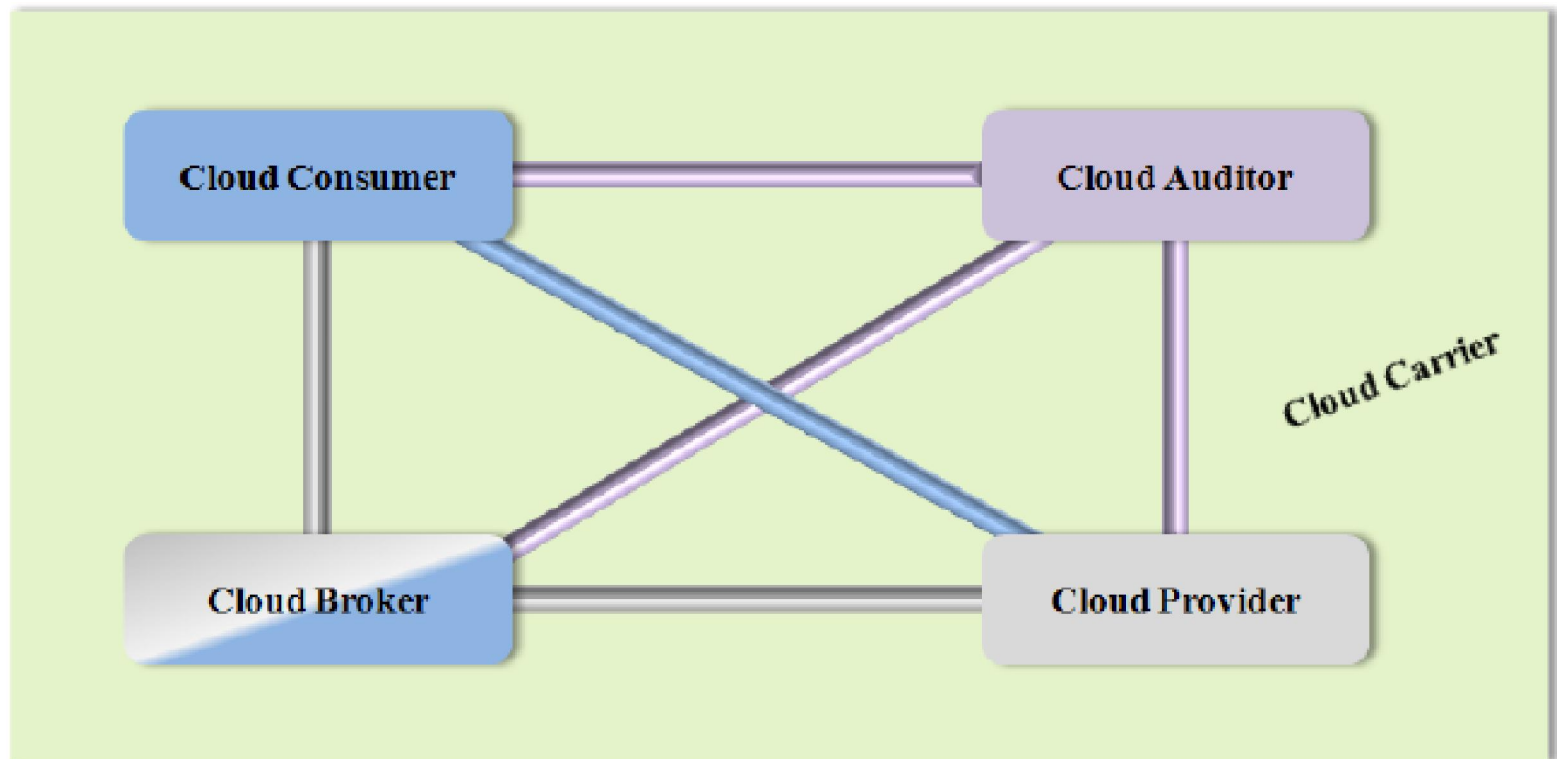
- Integration of cloud services by consumers can be too complex and can be requested from a cloud broker rather than from a provider directly
 - An entity/service operated by the provider or third party
- Provides services in three forms
 - Intermediation: presents the service to consumers (e.g. In catalogue), provides/enhances/improves a given service (e.g. by adding identity management, performance reporting, enhanced security)
 - Aggregation: combines and integrates multiple services into one
 - Arbitrage: the services being aggregated may change or come from different providers




Example Usage Scenario for Broker

- A consumer requests a service from a broker instead of contacting the provider directly
- The broker creates a new service by combining multiple services



Interactions between Actors



-  The communication path between a cloud provider and a cloud consumer
-  The communication paths for a cloud auditor to collect auditing information
-  The communication paths for a cloud broker to provide service to a cloud consumer

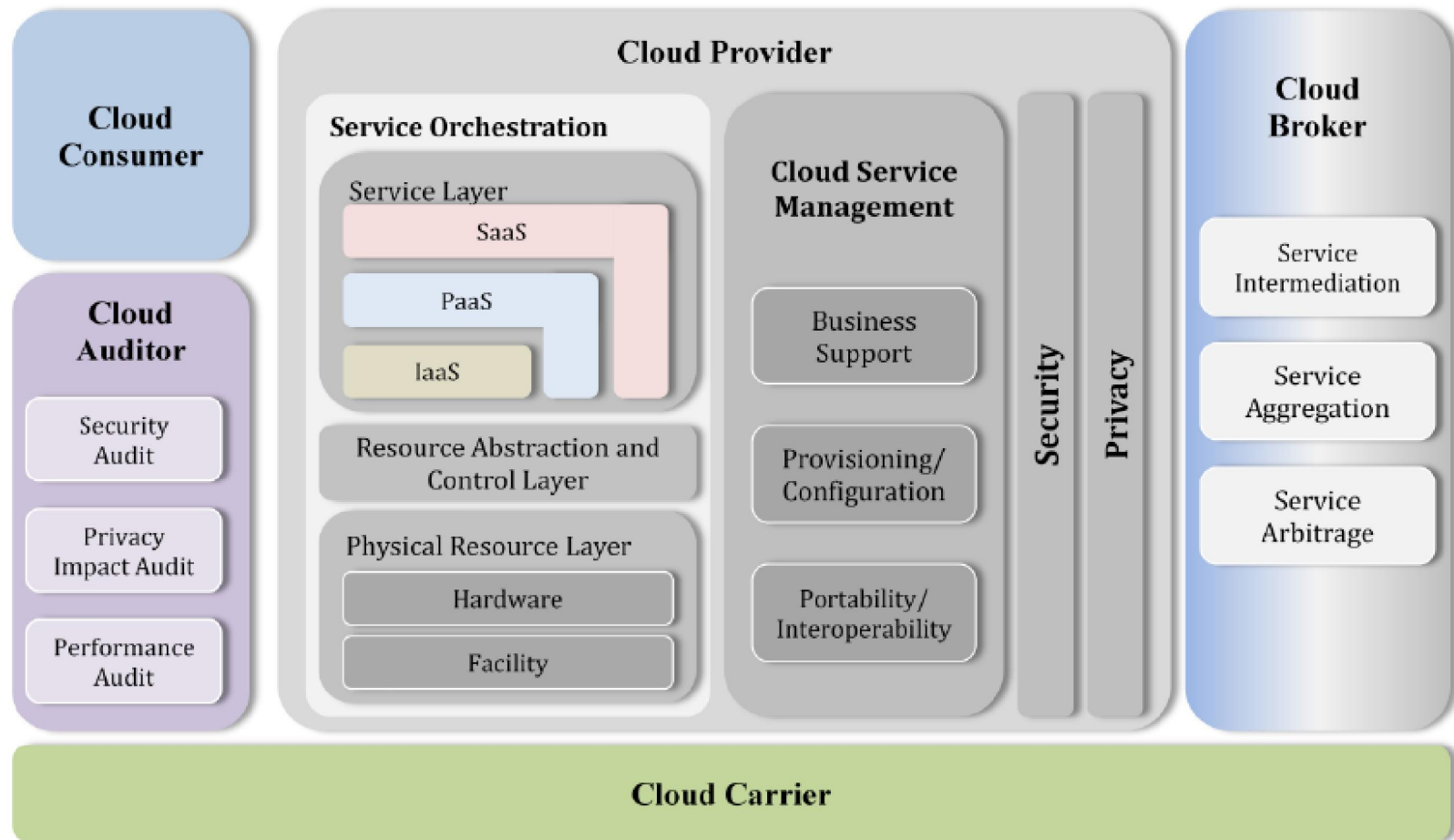
Cloud Carrier

- Acts as an intermediary that provides connectivity and transport of cloud services between cloud consumers and cloud providers
- Provides access to consumers through a public/private network or telecom provider
- A provider may set-up SLAs with cloud carriers in order to provide services with the level of SLAs offered to consumers (e.g. may require dedicated or secure connections)

Architectural Components

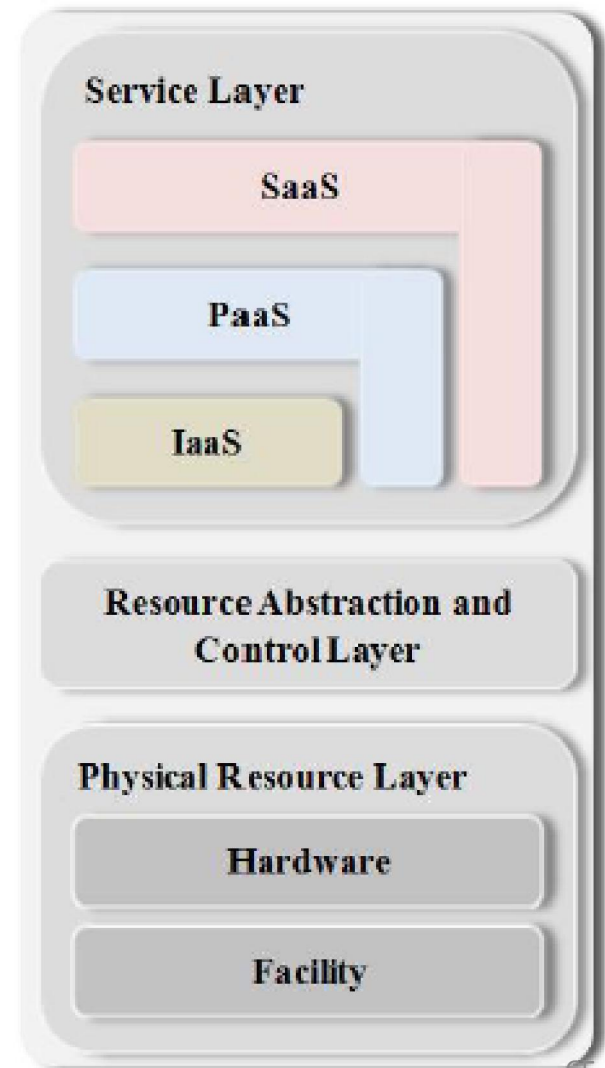
- Architectural Components for managing and providing cloud services, describe the important aspects of
 - Service deployment, orchestration, management, security and privacy
 - Portability and interoperability issues for data and services are also crucial factors as consumers need confidence and moving data and services across clouds
 - Security and privacy build trust and acceptance in clouds ability to provide a trustworthy and reliable system
 - Business support: implementation of specific business model

Conceptual Reference Model



Service Orchestration

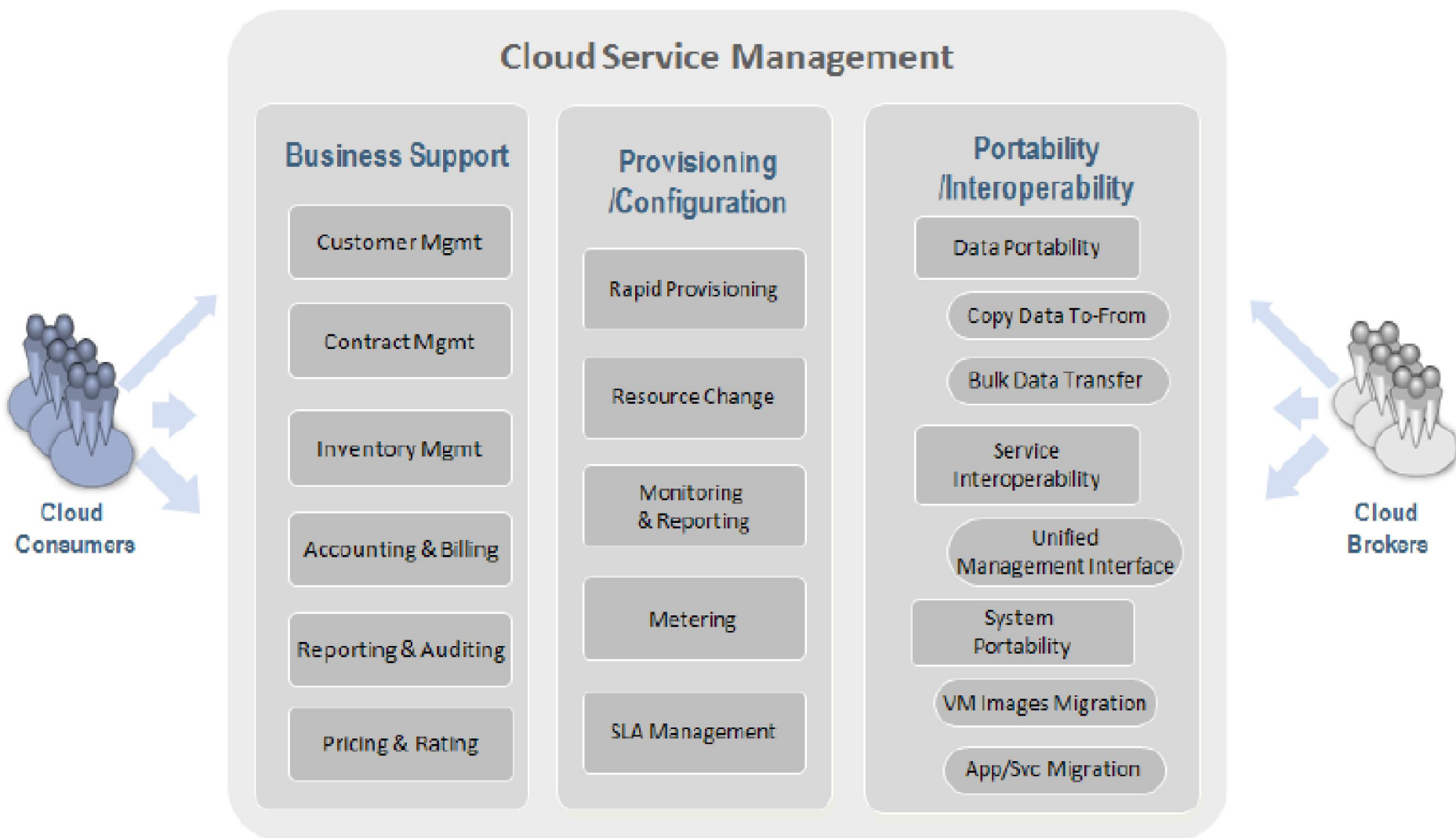
- Composition of service components to support cloud providers activities (in coordination with management of resources) in order to provide cloud services
- **Service Layer**: interfaces for accessing services (typically for IaaS, PaaS, SaaS)
- **Resource Abstraction / Control Layer**: interfaces for accessing virtualized resources e.g. hypervisors, VMs, virtual storage
- **Physical Resource Layer**: interfaces for accessing to physical resources (computers, disks, routers, firewalls, etc.)



Service Management

- Includes all of service-related functions that are necessary for the management and operation of services available to consumers
- Can be described from different perspectives
 - Business support
 - Provisioning and configuration
 - Portability and interoperability

Cloud Service Management



Management: Business Support

- Business related services
 - **Customer management**: manage customer accounts, open/close accounts, manage user profiles, manage provider-customer relationships
 - **Contract management**: setup/negotiate/terminated contract and SLAs
 - **Pricing/Rating**: evaluate cloud services, handle promotions and pricing rules by user profile
 - **Accounting and Billing**: collect billing information, send billing statements, manage payments
 - **Reporting/auditing**: monitor user operations, generate reports

Management: Provisioning/Configuration

- Responsibilities included
 - **Rapid provisioning**: automatically deploy cloud services based on user demands
 - **Resource changing**: adjust service configurations or, resource assignment for repairs/upgrades
 - **Metering**: Provide metering capability per service type
 - **SLA management**: define SLAs, monitor SLAs, enforce SLAs

Management:

Portability/Interoperability

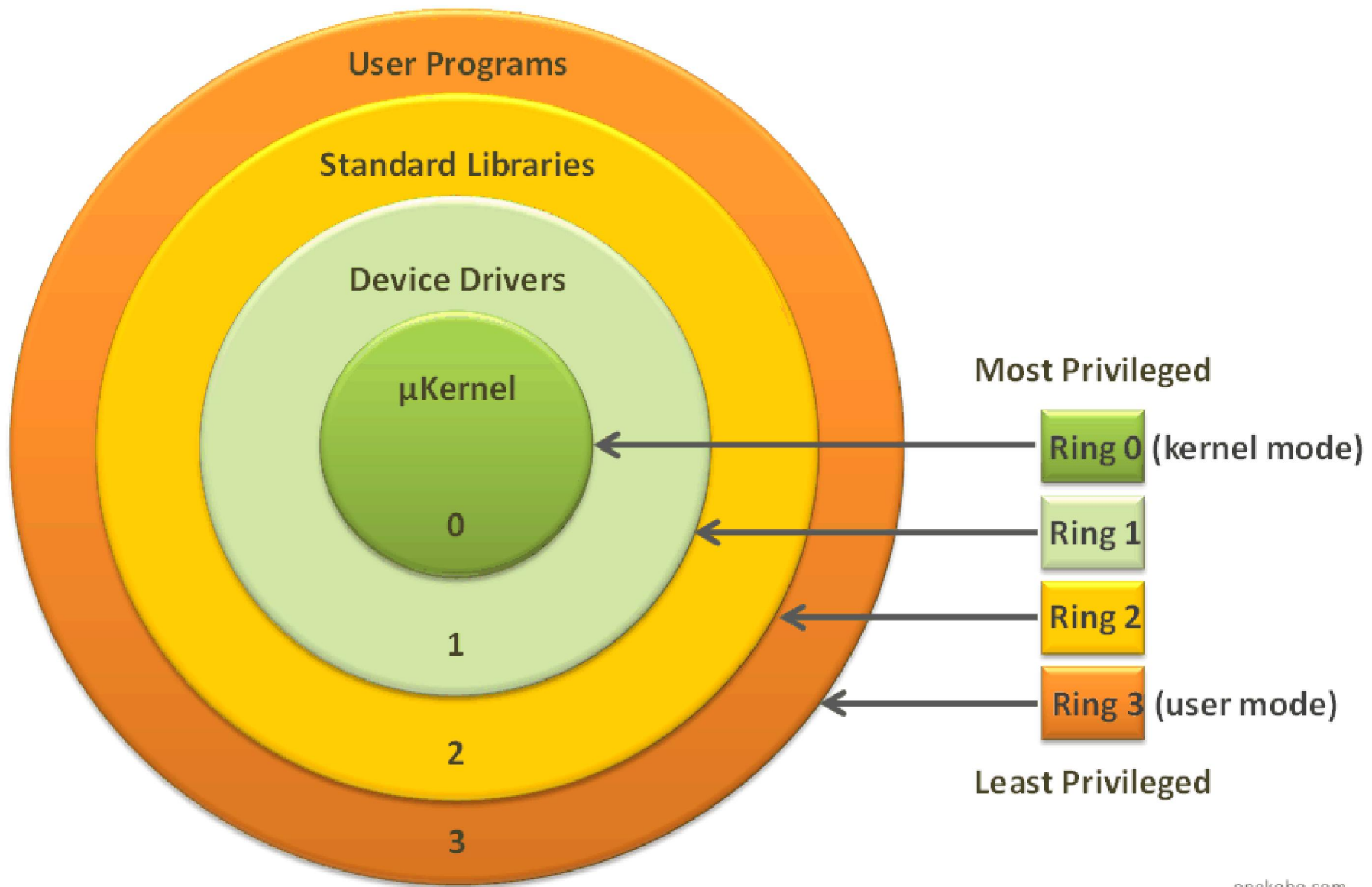
- Cloud adoption depends also how the cloud can address security, privacy, portability and interoperability concerns
- **Portability**: ability to move applications and data across clouds and cloud providers
 - **Data portability**: copy/move objects across clouds
 - **System portability**: move / migrate a stopped VMs or applications with their contents
 - **Service Interoperability**: use data and services across multiple cloud providers using common interface (RESTful APIs)
- Different requirements for different service models: IaaS, SaaS focus on data portability, IaaS, PaaS on compatibilities between different virtualization technologies, PaaS focus also on service interoperability

Privacy

- Ensure privacy of collected **personal identifiable information** that can be used to distinguish, trace user's identity based on
 - user habits (e.g. Buying patterns)
 - personal data: user id's, financial, health data, usage data
 - Also related to data security as application data encompass user related information
- Mainly a responsibility of cloud providers

Security

- Cloud systems need to address security requirements such as **authentication**, **authorization**, confidentiality, identity management, security monitoring, security policy management, incident response
- Responsibility shared between provider and consumer
- Consider impacts per service model:
 - **SaaS**: manage accessibility of cloud offerings using network connection and through Web browser (Web browser security is an issue)
 - **IaaS**: hypervisor security for VM isolation
 - **PaaS**: user authorization to use services
- Impacts per deployment model: private cloud is dedicated to one customer, public is not



onekobo.com

Software defined cloud

- approach for automating the process of optimal cloud configuration by extending virtualization concept to all resources in a data center. An SDC enables easy reconfiguration and adaptation of physical resources in a cloud infrastructure, to better accommodate the demand on QoS through a software that can describe and manage various aspects comprising the cloud environment