Introduction to Cloud Computing

General Day to Day Services from Public service Providers

- Electricity
- Water
- Gas

As individual can not produce himself/ herself

Cloud Computing .

“ Cloud Computing is a service model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources like networks, servers, storage, applications, and services. The services can be rapidly provisioned and released with minimal management effort or service provider interaction.”

By the US National Institute of Standards and Technology (NIST)

Cloud computing offers scalable & elastic computing and storage services at low cost. The resources used for these services can be metered and the users can be charged only for the resources they have used.

Why Cloud Computing?

To start new Business organization/Company we need IT infrastructure

On Campus

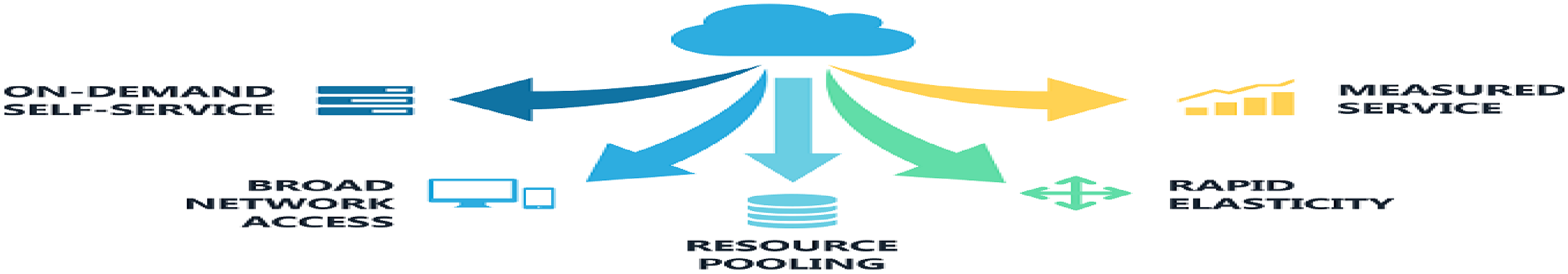
- High cost with less scalability
- Requires separate Server space with AC
- Needs hardware and software maintenance team.
- Poor data security
- High risk of data loss
- Lack of flexibility
- Data can not be accessed remotely

On Cloud

- Pay for what you utilize
- Requires no separate Server space with AC
- No expert is required for any maintenance.
- Better data security
- Low risk of data loss
- Higher flexibility
- Data can be accessed remotely

Characteristics of Cloud Computing

NIST has identified FIVE characteristics for Cloud Computing

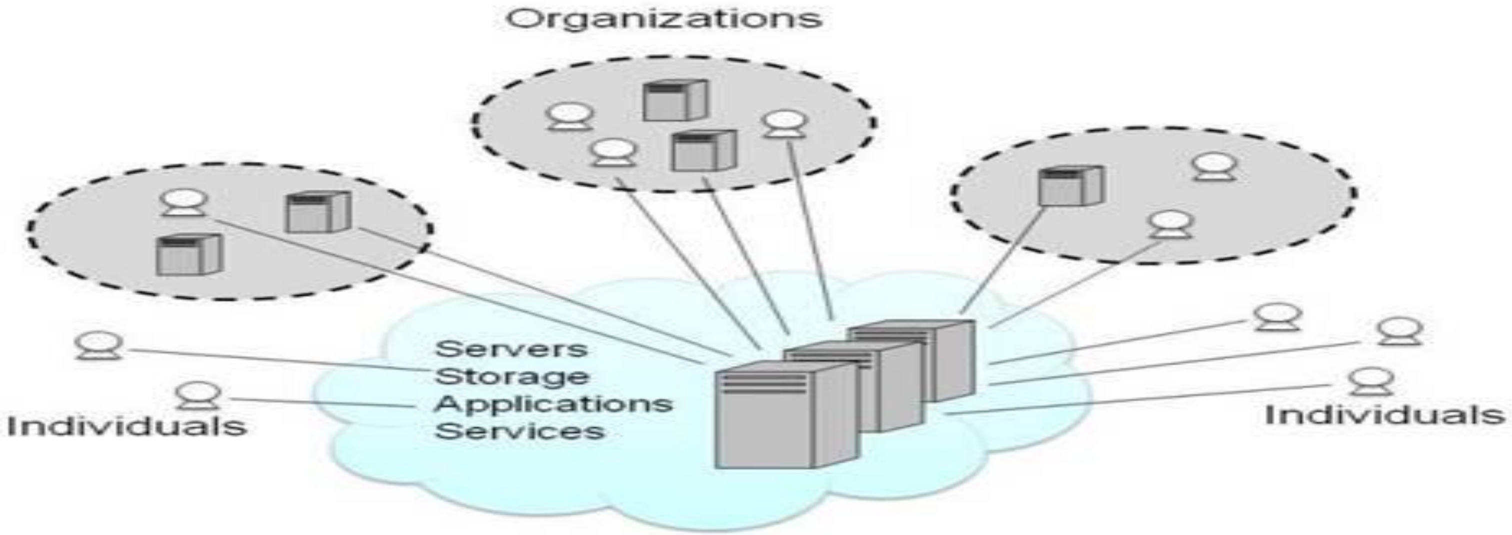


1. **On-demand self service:** User can request and get the service offered by the provider without interactions with the administrator of cloud service provider. The request and fulfillment process are automated which is advantageous to both service provider and consumer. User self-service reduces the administrative burden on the provider and also allows an organization's IT staff to focus on other, strategic, activities. (general example Ola/Uber service)

2. **Broad network access:** Cloud computing resources can be accessed from anywhere, any device and from any platform over the network using standard access mechanisms. Only thing is we need to consider proper authentication and authorization along with higher reliability and performance of connections

Characteristics of Cloud Computing (Contd.)

2. **Resource pooling** : The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. Multi-tenant aspects of cloud serves multiple users on the same physical hardware. The resource pooling may be **Processor and Memory Pooling, Storage Pooling and Network Infrastructure Pooling**



Characteristics of Cloud Computing (Contd.)

4. Rapid elasticity: Cloud resources can be rapidly scaled up or down based on the demand, it involves provisioning, allocating and releasing of computing resources. Rapid elasticity is adapting to workload changes in the service requirement.

- ***Horizontal Scaling(scale out):*** *This involves launching and provisioning of additional resources*
- ***Vertical Scaling(scaling up):*** *This involves changing the computing capacity assigned to the server resources by keeping the number of server resources same*

5. Measured service: The use of cloud resources is based on pay-per-use model. The usage of resources is measured and the user is charged based on the metrics such as amount of CPU cycles, amount of storage space used, number of network I/O requests etc.,

Characteristics of Cloud Computing (Contd.)

Some of the additional characteristics which leads to savings in cost

Performance: Provides better performance for applications by scaling up or down the resources based on the dynamic application workload.

Reduced Cost: Applications can run at reduced cost as the applications are provided with resources only as much as required. Cloud avoids the investment on the purchase of computing asset to cover worst case requirement.(Ex during weekends and holidays e-commerce site workload is high)

Outsourced Management: Cloud computing allows users/organization to outsource the IT infrastructure and infrastructure maintenance to external cloud provider and pay only for the operational expenses of the cloud resources used.

Characteristics of Cloud Computing (Contd.)

Reliability: As the cloud computing environment is managed by professionals, the deployed applications will have higher reliability and availability Cloud service providers promise 99.99% up time guarantee through Service Level Agreement(SLA)

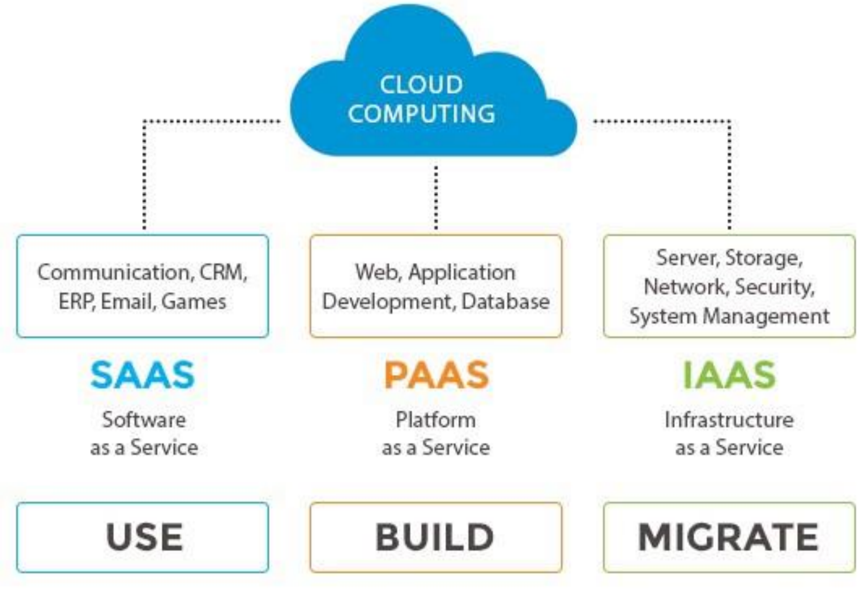
Multi-tenancy: Allows multiple users to same shared resources. B-to-B, Banking Social networking deployed in cloud are multi-tenanted applications.

- ***Virtual multi-tenancy:*** *Multiple tenants are served from VMs that execute concurrently on top of the same computing and storage resources.*
- ***Organic multi-tenancy:*** *Every component in the system architecture is shared among multi-tenants. Organic multi-tenants exits when explicit multi-tenants design patterns are coded into the application*

Cloud Models

Cloud services can be provided to the users in different forms, based on what service and how services are provided we can categories the cloud model as **Service Model** and **Deployment Model**

Service Models


CLOUD COMPUTING
Communication, CRM, ERP, Email, Games
SAAS
Software as a Service
USE
Web, Application Development, Database
PAAS
Platform as a Service
BUILD
Server, Storage, Network, Security, System Management
IAAS
Infrastructure as a Service
MIGRATE

1. Software-as-a-Service(SAAS):

End User (Appn. user)

- Application Software, Interface to the application are provide
- Applications are platform independent and accessed through various devices like thin clients, laptops, tab, smart phone etc
- Cloud infrastructure like servers, OS, network, storage etc. is hidden

Cloud Provider

- Manages cloud infrastructure
- Provider will either host the application and related data using its own servers, databases, networking and computing resources. Ex. SaaS for Sales Management, CRM, HRM, Microsoft Office 365 etc.

Cloud Models

Service Model (SaaS)

Advantages

- High Scalability
- Automatic Updates
- Accessibility and persistence

Disadvantages

- Relay on outside venders for software
- Little control
- Impose unwanted service changes
- Victim to security breach

2. Platform-as-a-Service(PAAS):

End User (Appn. Developer)

- Developing, Deploying, configuring and managing Application Software

Cloud Service Provider

- Provides software development tools(development frameworks, OS, deployment frameworks) APIs, libraries to the developer to develop applications
- Manages the cloud infrastructure
- Ex., Microsoft Azure, Heroku, AWS Lambda, Google App Engine, Dokku etc.

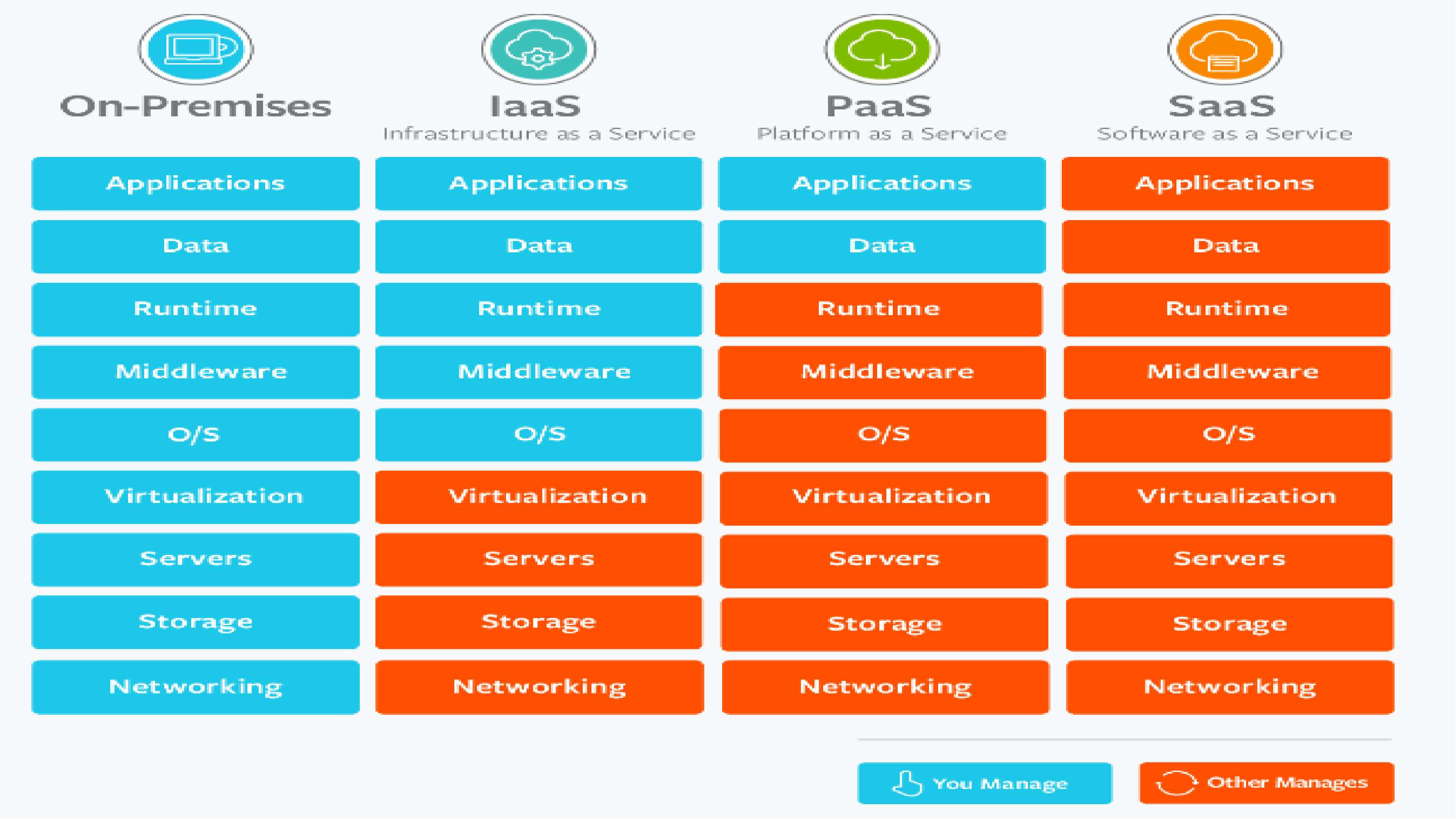
Cloud Models

3. Infrastructure-as-a-Service(IAAS):

End User

Cloud Service Provider

Cloud Models

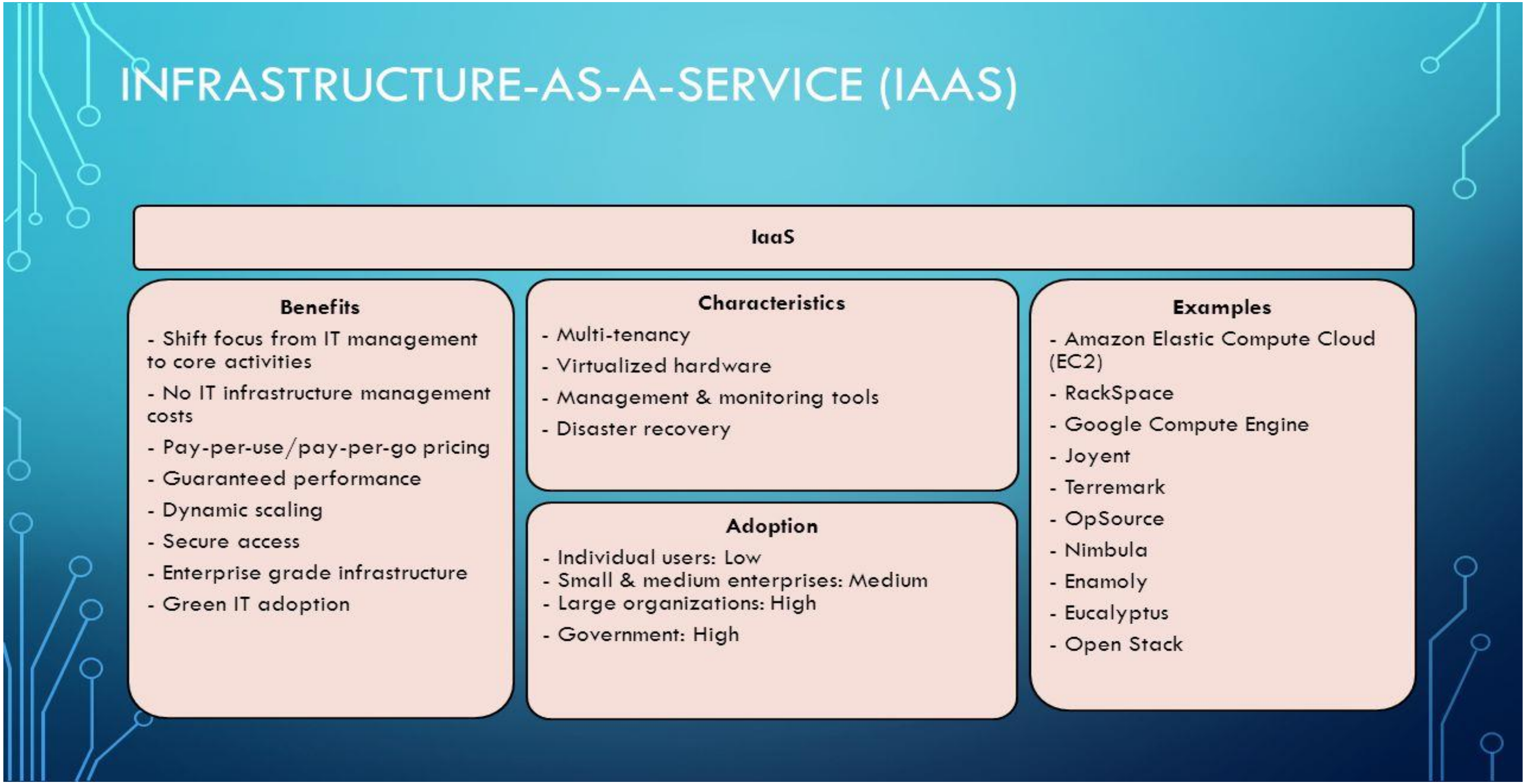


On-Premises	IaaS Infrastructure as a Service	PaaS Platform as a Service	SaaS Software as a Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

You Manage

Other Manages

Summary of Service Models-IaaS



INFRASTRUCTURE-AS-A-SERVICE (IAAS)

IaaS

Benefits

- Shift focus from IT management to core activities
- No IT infrastructure management costs
- Pay-per-use/pay-per-go pricing
- Guaranteed performance
- Dynamic scaling
- Secure access
- Enterprise grade infrastructure
- Green IT adoption

Characteristics

- Multi-tenancy
- Virtualized hardware
- Management & monitoring tools
- Disaster recovery

Adoption

- Individual users: Low
- Small & medium enterprises: Medium
- Large organizations: High
- Government: High

Examples

- Amazon Elastic Compute Cloud (EC2)
- RackSpace
- Google Compute Engine
- Joyent
- Terremark
- OpSource
- Nimbula
- Enamoly
- Eucalyptus
- Open Stack

PaaS

PaaS

Benefits

- Lower upfront & operation cost
- No IT infra management cost
- Improved scalability
- Higher performance
- Secured access
- Quick & easy deployment
- Seamless integration

Characteristics

- Multi-tenancy
- Open integration protocols
- App development tools &SDKs
- Analytics

Adoption

- Individual users: **Low**
- Small & Medium Enterprise: **Medium**
- Small organizations: **High**
- Government: **Medium**

Examples

- Google App Engine
- Windows Azure platform
- Force.com
- Right Scale
- Heroku
- Github
- Gigaspaces
- Appscale
- OpenStack
- LongJump

SaaS

SaaS

Benefits

- Lower cost
- No infrastructure required
- Seamless upgrades
- Guaranteed performance
- Automated backups
- Easy Data recovery
- Secure
- High adoption
- On-the move access

Characteristics

- Multi-tenancy
- On-demand software
- Open integration protocols
- Social network integration

Adoption

- Individual users: **High**
- Small & Medium Enterprise: **High**
- Small organizations: **High**
- Government: **Medium**

Examples

- Google Apps
- Salesforce.com
- Facebook
- Zoho
- Dropbox
- Taleo
- MS Office 365
- Linkedin
- Slideshare
- Carecloud

Cloud Deployment Models

Depending on how these service model can be implemented NIST defines FOUR cloud deployment models

1. Public Cloud : Cloud services are available to

- General Publics (individuals)
- Large Organizations
- Small & medium enterprises and Governments through internet

Best suited for the users who wants to use cloud infrastructure to develop, test and host applications in cloud to serve large workload

Operated on the **pay-as-per-use model** and administrated by the **third party**

The same storage is being used by multiple users at the same time.

Amazon Elastic Compute Cloud (EC2), Microsoft Azure, IBM's Blue Cloud, Sun Cloud, and Google Cloud are examples of the public cloud.

Deployment Models (Public)

Advantages

- *Low Cost*

Lower cost than private, or hybrid cloud, as it shares the same resources with a large number of consumers.

- *Location Independent*

its services are offered through the internet.

- *Quickly and easily set up*

Can easily buy public cloud on the internet and deployed and configured it remotely through the cloud service provider within a few hours.

- *Save Time*

Service provider is responsible for the manage and maintain data centers

- *Scalability and reliability*

Easy to add and remove) and reliable (24*7 available) services to the users at an affordable cost.

Disadvantages

- *Low Security*

Resources are shared publicly.

- *Performance*

performance depends upon the speed of internet connectivity.

- *Less customizable*

Public cloud is less customizable than the private cloud.

Deployment Models (Private)

2. Private (Internal/Corporate)Cloud : Cloud services are

- Available to Single Organization
- Infrastructure can be set up on premise or off-premise
- Managed internally or by third party

Best suited for the application where security is very important and organization needs tight control over their data

HP Data Centers, Microsoft, Elastra-private cloud, and Ubuntu are the example of a private cloud.

Deployment Models (Private)

Advantages

- *More Control*

Have more control over their resources and hardware than public clouds as only accessed by selected users. .

- *Security & privacy*

Security & privacy are one of the big advantages of cloud computing. Private cloud improved the security level as compared to the public cloud.

- *Improved performance*

Private cloud offers better performance with improved speed and space capacity.

Disadvantages

- *High cost*

The cost is higher than a public cloud because set up and maintain hardware resources are costly.

- *Restricted area of operations*

Private cloud is accessible within the organization, so the area of operations is limited.

- *Limited scalability*

Private clouds are scaled only within the capacity of internal hosted resources.

- *Requires skilled people*

Skilled people are required to manage and operate cloud services.

Deployment Models (Hybrid)

3. Hybrid Cloud :

- Combination of Private and Public cloud services
- **Non-critical activities** are performed by the public cloud and **critical activities** are performed by the private cloud.
- Hybrid cloud is used in finance, healthcare

Best suited for the organization where security is very important, application and data is hosted on a private cloud. Organization wants to take cost saving benefit by hosting shared application and data in public cloud.

The best hybrid cloud provider companies are Amazon, Microsoft, Google, Cisco, and NetApp.

Deployment Models (Hybrid)

Advantages

- *Flexible and secure*

It provides flexible resources because of the public cloud and secure resources because of the private cloud. .

- *Cost effective*

Hybrid cloud costs less than the private cloud. It helps organizations to save costs for both infrastructure and application support.

- *Security*

Hybrid cloud is secure because critical activities are performed by the private cloud.

Disadvantages

- *Networking issues*

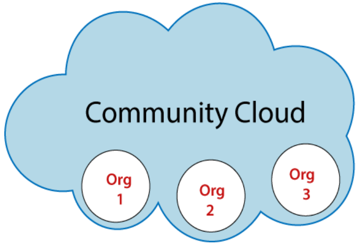
In the Hybrid Cloud, networking becomes complex because of the private and the public cloud.

- *Infrastructure Compatibility*

With dual-levels of infrastructure, a private cloud controls the company, and a public cloud does not, so there is a possibility that they are running in separate stacks.

Source: javapoints

Deployment Models (Community)



4. Community Cloud :

- Allows systems and services to be accessible by a group of several organizations to share the information
- Is a collaborative effort in which infrastructure is shared between several organizations from a specific community with common concerns.
- It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.

Deployment Models (Community)

Advantages

- *Security*

Community cloud is more secure than the public cloud but less secure than the private cloud.

- *Cost effective*

Community cloud is cost effective because the whole cloud is shared between several organizations or a community.

- *Sharing infrastructure*

Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organizations..

Disadvantages

- Community cloud is not a good choice for every organization.

- The fixed amount of data storage and bandwidth is shared among all community members.
- Community Cloud is costly than the public cloud.
- Sharing responsibilities among organizations is difficult.

Cloud Services Examples

IaaS: Elastic Compute Cloud (EC2) , Google Compute Engine, Azure VMs

EC2(TM) :

- Provides Computing capacity in the form of VM
- From Amazon.com using simple web based interfaces
- Provides pre-configured Amazon Machine Images(AMIs) with pre launched OS and also user custom OS, applications, libraries
- User can launch the applications easily and rapidly
- EC2 can provision hundreds or thousands of server instances simultaneously
- Provides the instances with varying computing capacity ranging from 1core with 1 EC2 CPU unit, 1.7GB RAM and 160 GB storage to large instances of 4 core with 2 EC2 CPU with each 15GB RAM and 1690 GB storage
- Pricing model is based on pay-per-use model
- Reserved instances, spot instances based on biding

Cloud Services Examples

PaaS: Google App Engine, Azure Platform, GitHub

Google App Engine (GAE): From Google providing PaaS

Cloud based web service for hosting web applications and storing data, provides SDK for developing web applications

Provides automatic scaling and load balancing capability

Features

- **Popular programming languages :** User can build applications using Node.js, Java, Ruby, C#, Go, Python, or PHP—or bring your own language runtime.
- **Open and flexible:** user can customize runtimes by having any library and framework through Docker
- **Application versioning:** Host and control the versions of user applications, easily create development, test, staging, and production environments.
- **Traffic splitting:** Route incoming requests to different version of app, 1.0/1.5 test, and do incremental feature rollouts.
- **Application security:** Protects application by defining access rules with App Engine firewall

Cloud Services Examples

SaaS: Salesforce.com Google Apps, Office 365

Salesforce Sales Cloud: Cloud based CRM allows to manage customer profiles, track opportunities, optimize campaigns, can access through laptops, tabs, smart phones

Service Cloud: Cloud based customer service management that allows to listen and respond to customers across multiple Social Platforms and automatically route cases to the appropriate agent. Since social customer service is integrated into the Salesforce Customer Success Platform, your Social Media team can gather a comprehensive picture of the customer to inform responses.

Marketing Cloud: platform to supporting various phases of marketing. Combines the digital marketing capabilities of Marketing Cloud with the data management, segmentation, and campaign management tools in Salesforce.

All together provides:

- Accounts and contacts
- Opportunities
- Campaigns
- Analytics and Forecasts

Cloud Based Applications

Intelligent Transport System(ITM) : Is a data driven system that helps in transport system

- Real Time vehicles tracking
- Dynamic vehicles routing
- Anticipating customer demands for passenger pickup & drop
- Monitoring vehicle health
- E-ticketing management

Energy Systems:

- Collect real time data and condition monitoring, failure predictions
- Dynamic load-shifting program
- Smart Grid

Cloud Based Applications

Education System :

- Cloud based discussion forums
- Online and distance education programs
- Online exams
- Tracking student progress

Healthcare System:

- Securely accessing patient data
- E-prescribing
- Stored history and information can streamline admission and discharge process

Cloud Migration

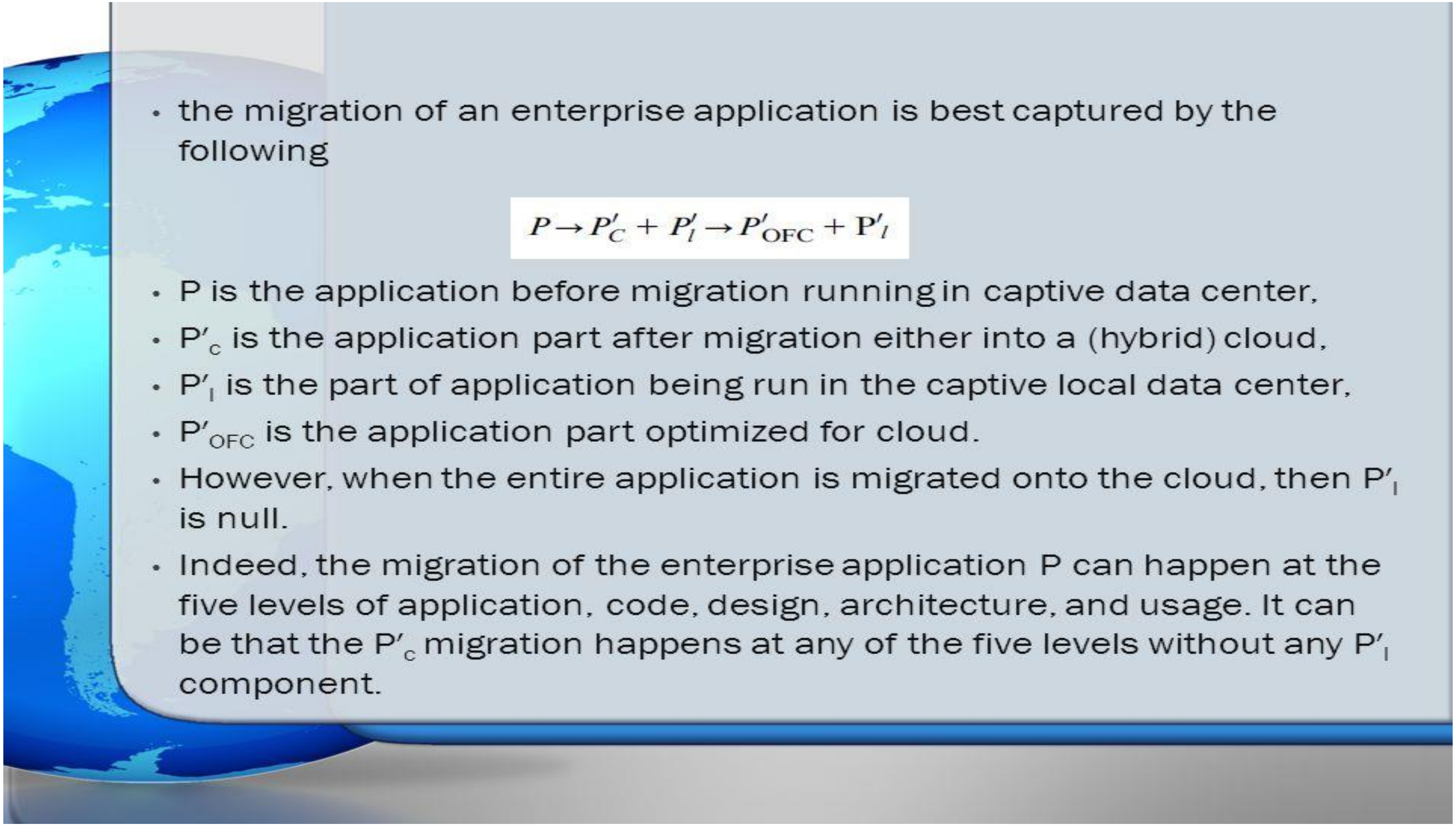
Cloud migration is the process of moving an application's schema or data from a local application to the cloud. The time it takes to migrate to the cloud will vary depending on the organizations current legacy

Migration of an application into the cloud can happen in one of several ways:

- Either the application is clean and independent, so it runs as is
- Some degree of code needs to be modified and adapted
- The design (and therefore the code) needs to be first migrated into the cloud computing service environment
- Migration results in the core architecture being migrated for a cloud computing service setting, this resulting in a new architecture being developed, along with the accompanying design and code implementation.
- The application is migrated as is, it is the usage of the application that needs to be migrated and therefore adapted and modified.

So the migration can happen at one of the FIVE levels of application, code, design, architecture, and usage.

Cloud Migration (Contd.)



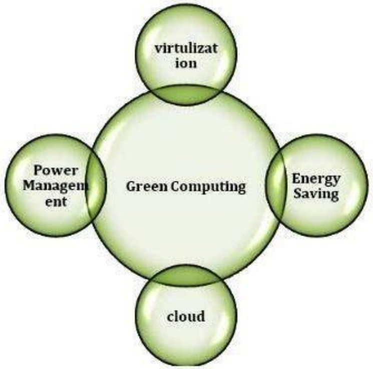
- the migration of an enterprise application is best captured by the following

$$P \rightarrow P'_C + P'_l \rightarrow P'_{\mathrm{OFC}} + \mathrm{P}'_l$$

- P is the application before migration running in captive data center,
- P'_c is the application part after migration either into a (hybrid) cloud,
- P'_l is the part of application being run in the captive local data center,
- P'_{OFC} is the application part optimized for cloud.
- However, when the entire application is migrated onto the cloud, then P'_l is null.
- Indeed, the migration of the enterprise application P can happen at the five levels of application, code, design, architecture, and usage. It can be that the P'_c migration happens at any of the five levels without any P'_l component.

Challenges in Cloud Technologies

Cloud service offerings simplistic view of IT in case of IaaS, or a simplistic view of programming in case PaaS or a simplistic view of resources usage in case of SaaS, the underlying systems level support challenges are **huge and highly complex.**



- Security
- Performance Monitoring
- Consistent & Robust Service abstractions
- Meta Scheduling
- Energy efficient load balancing
- Scale management
- SLA & QoS Architectures
- Interoperability & Portability
- Green IT

Cloudonomics

Migrating into the cloud is driven by economic reasons of cost cutting in both the IT capital expenses (Capex) as well as operational expenses (Opex). Migrating into cloud leads to both short term and long term benefits.

Cloudonomics is the Economics of Cloud Computing. It is the expression of when a migration can be economically feasible or tenable.

According to *Cloudonomics*, if the average costs of using an enterprise application on a cloud is substantially lower than the costs of using it in one’s captive data center and if the cost of migration does not add to the burden on ROI, then the case for migration into the cloud is strong.

The Seven-step Model Of Migration Into A Cloud

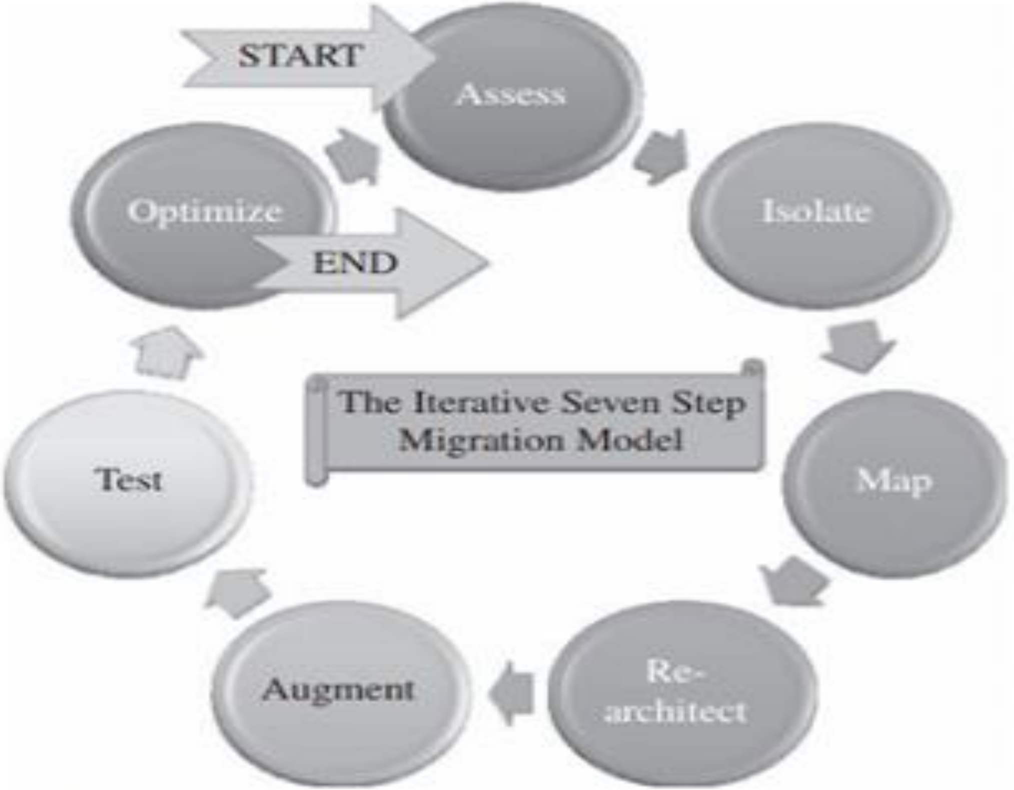
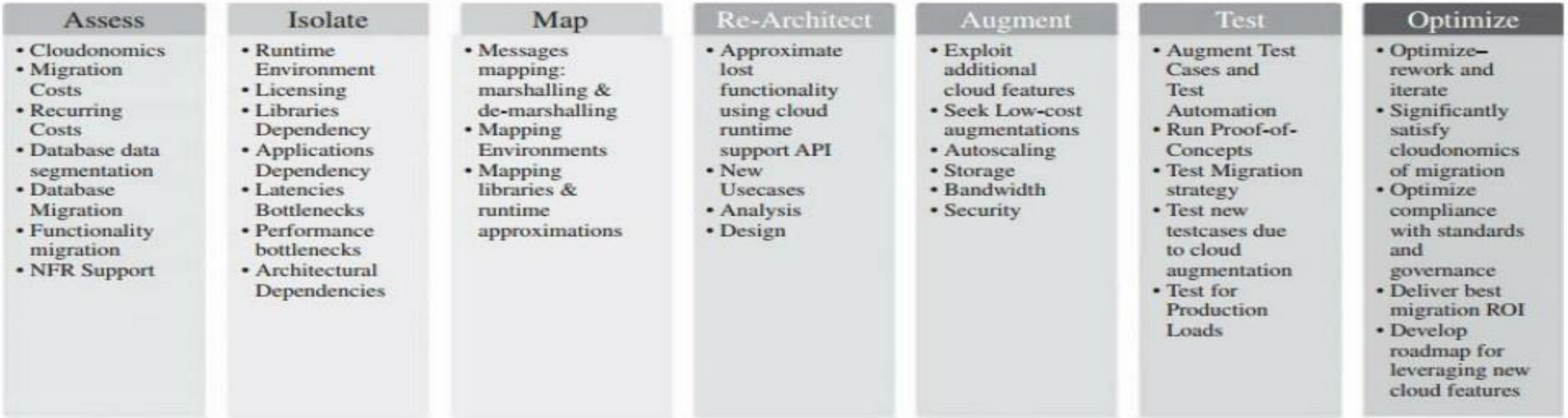


FIGURE 2.5. The iterative Seven-step Model of Migration into the Cloud. (*Source:* Infosys Research.)

The Seven-step Model Of Migration Into A Cloud

1. ***Assess :*** assessment of the issues relating to migration, at the application, code, design, and architecture levels. Moreover, assessments are also required for tools being used, functionality, test cases, and configuration of the application.
2. ***Isolate :*** isolation of all the environmental and systemic dependencies of the enterprise application within the captive data center. These include library, application, and architectural dependencies. This step results in a better understanding of the complexity of the migration.
3. ***Map :*** A mapping construct is generated to separate the components that should reside in the captive data center from the ones that will go into the cloud.
4. ***Re-architect :*** Some part of the application has to be re-architected and implemented in the cloud. This can affect the functionalities of the application and some of these might be lost. It is possible to approximate lost functionality using cloud runtime support API.
5. ***Augment :*** The features of cloud computing service are used to augment the application.
6. ***Test :*** Once augmented the application needs to be validated and tested. This is to be done using a test suite for the applications on the cloud. New test cases due to augmentation and proof-of-concepts are also tested at this stage.
7. ***Optimize :*** It may take several optimizing iterations for the migration to be successful. It is best to iterate through this seven step model as this will ensure the migration to be robust and comprehensive.

The Seven-Step Model of Migration into a Cloud

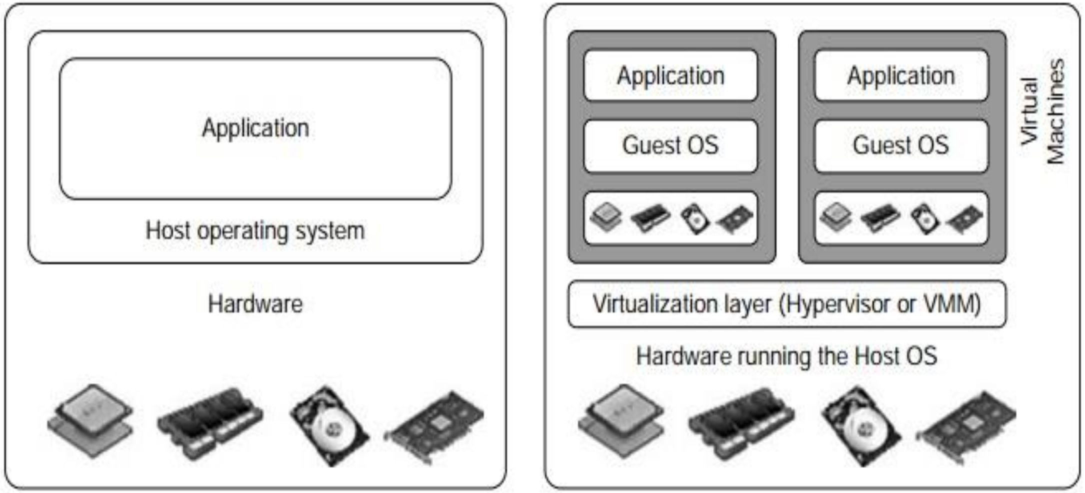


Assess	Isolate	Map	Re-Architect	Augment	Test	Optimize
• Cloudonomics • Migration Costs • Recurring Costs • Database data segmentation • Database Migration • Functionality migration • NFR Support	• Runtime Environment • Licensing • Libraries Dependency • Applications Dependency • Latencies Bottlenecks • Performance bottlenecks • Architectural Dependencies	• Messages mapping: marshalling & de-marshalling • Mapping Environments • Mapping libraries & runtime approximations	• Approximate lost functionality using cloud runtime support API • New Usecases • Analysis • Design	• Exploit additional cloud features • Seek Low-cost augmentations • Autoscaling • Storage • Bandwidth • Security	• Augment Test Cases and Test Automation • Run Proof-of-Concepts • Test Migration strategy • Test new testcases due to cloud augmentation • Test for Production Loads	• Optimize– rework and iterate • Significantly satisfy cloudonomics of migration • Optimize compliance with standards and governance • Deliver best migration ROI • Develop roadmap for leveraging new cloud features

FIGURE 2.6. Some details of the iterative Seven Step Model of Migration into the Cloud.

Virtualization

- Partitioning the resources of physical system (CPU, memory, storage and network) into multiple virtual resources.
- Allows resource pooling and multi-tenancy. Users are assigned virtual resources that run on top of physical resource

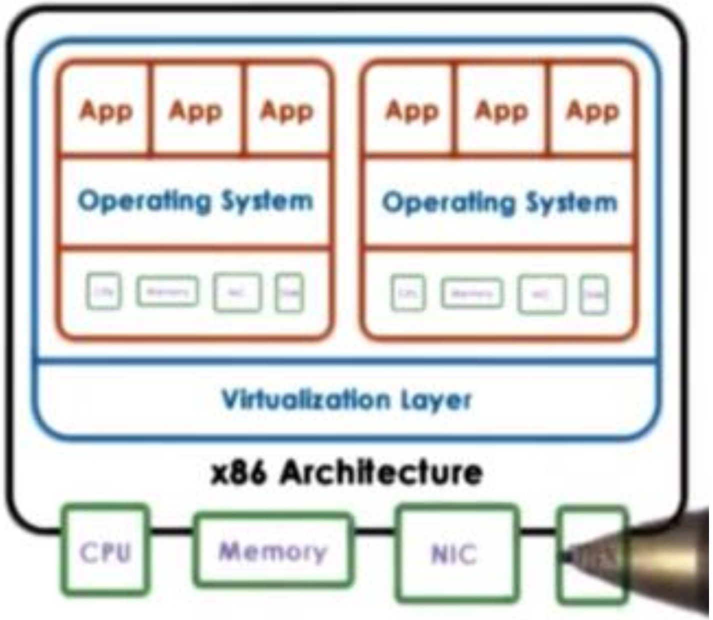


(a) Traditional computer

(b) After virtualization

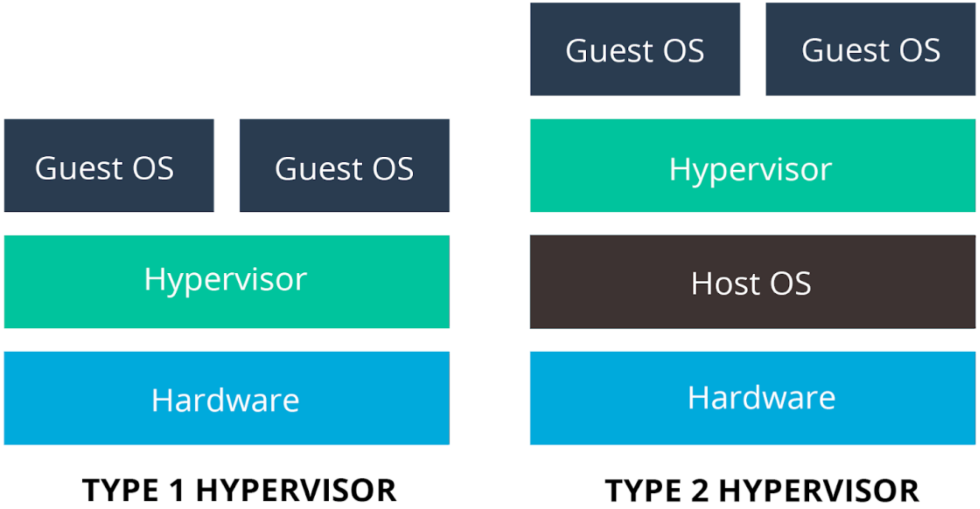
FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.



Virtualization (Contd.)

Hypervisor: A hypervisor (or virtual machine monitor, VMM, virtualizer) is a kind of emulator; it is computer software, firmware or hardware that creates and runs virtual machines

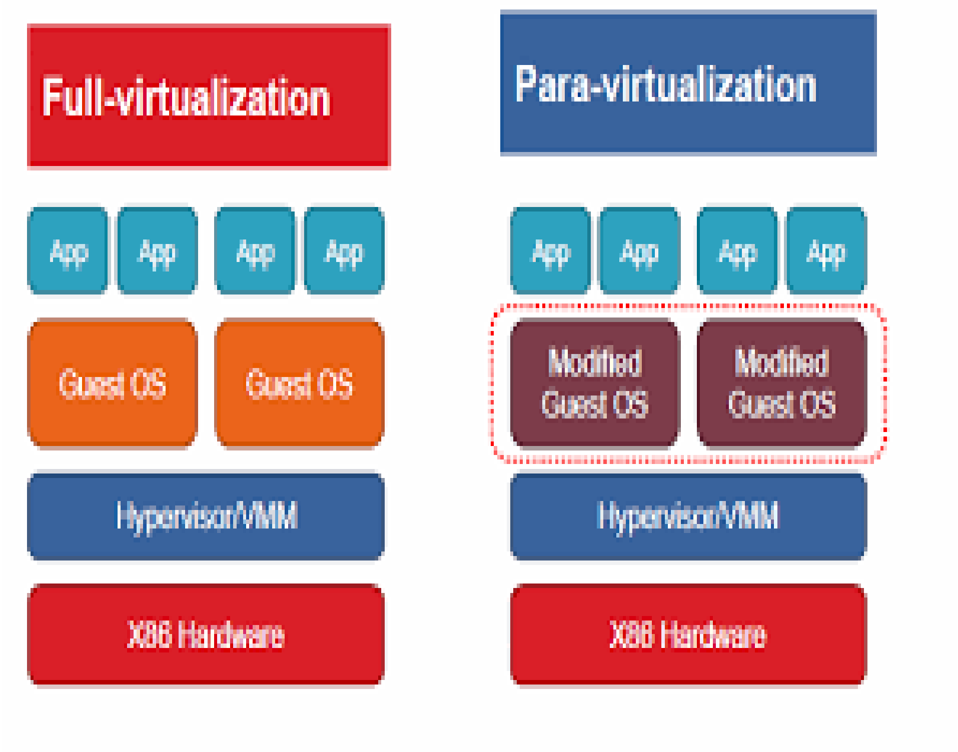


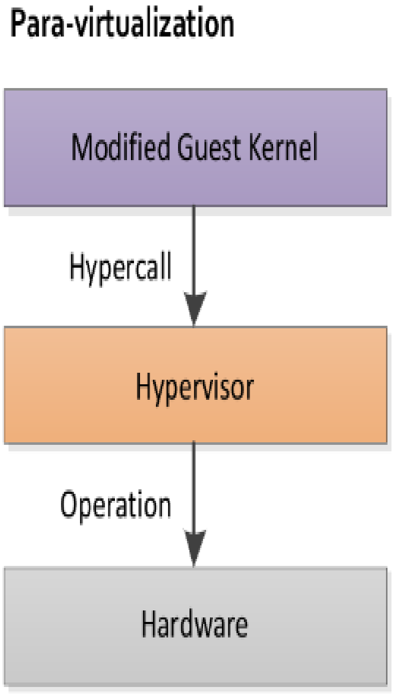
An enterprise or **large** organization and must deploy hundreds of VMs, a Type 1 hypervisor will suit their needs. A **smaller** deployment or require a testing environment, Type 2 hypervisors

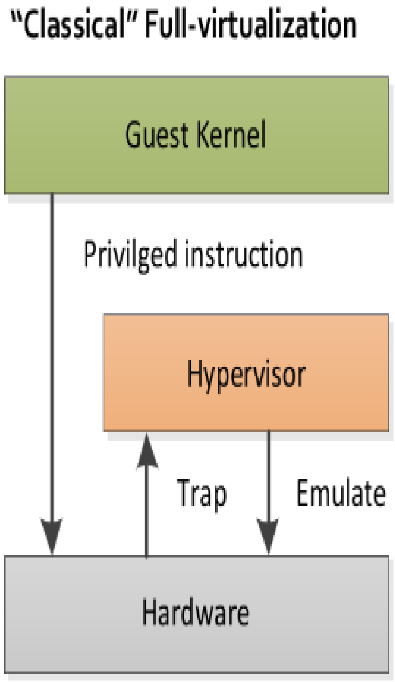
- **A Type 1** hypervisor runs directly on the host machine's physical hardware, and it's referred to as a bare-metal hypervisor.
- Type 1 hypervisors are regarded as the most efficient and best-performing hypervisors available for enterprise computing.
- Much higher initial cost and greater support contract requirements.

- **A Type 2** hypervisor is typically installed on top of an existing OS. It is sometimes called a hosted hypervisor
- it relies on the host machine's preexisting OS to manage calls to CPU, memory, storage and network resources.

Virtualization (Contd.)







Virtualization (Contd.)

Full virtualization : In full virtualization, guest OS is completely isolated virtualization layer and hardware. Microsoft and Parallels systems are examples of full virtualization.

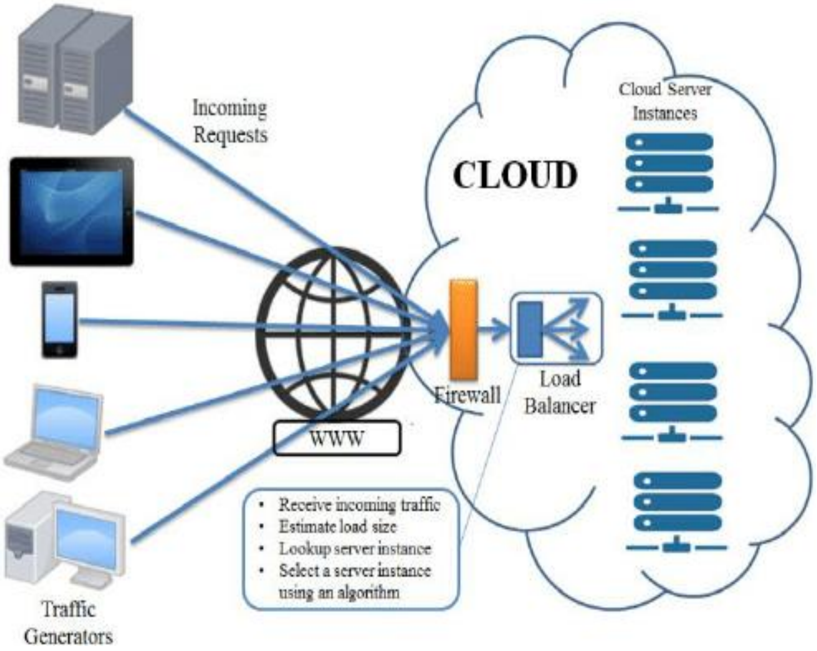
Para virtualization: In para virtualization, guest OS is not completely isolated but it is partially isolated by the virtual machine from the virtualization layer and hardware. VMware and Xen provides para virtualization.

Virtualization (Contd.)

S.No.	Full Virtualization	Para virtualization
1.	In Full virtualization, virtual machine permit the execution of the instructions with running of unmodified OS in an entire isolated way.	In para virtualization, virtual machine does not implement full isolation of OS but rather provides a different API which is utilized when OS is subjected to alteration.
2.	Full Virtualization is less secure.	While the Para virtualization is more secure than the Full Virtualization.
3.	Full Virtualization uses binary translation and direct approach as a technique for operations.	While Para virtualization uses hyper calls at compile time for operations.
4.	Full Virtualization is slow than para virtualization in operation.	Para virtualization is faster in operation as compared to full virtualization.
5.	Full Virtualization is more portable and compatible.	Para virtualization is less portable and compatible.
6.	Examples of full virtualization are Microsoft and Parallels systems.	Examples of para virtualization are VMware and Xen.

Load Balancing

- Distributing workloads across multiple servers
- Minimizing response time
- Maximizing throughput and maximum resource utilization
- Cloud based apps can achieve high availability and reliability
- Load balancer automatically reroute the user traffic to the healthy resources in the event of resource failure
- Under the load balancer pool of servers appears as a single high computing server for the end user
- Load balancing is beneficial with almost any type of service, like HTTP, SMTP, DNS, FTP, and POP/IMAP.
- Nginx, HAproxy, Pound are S/W load balancer
- Cisco System Catalyst 6500, Barracuda LoadB H/W

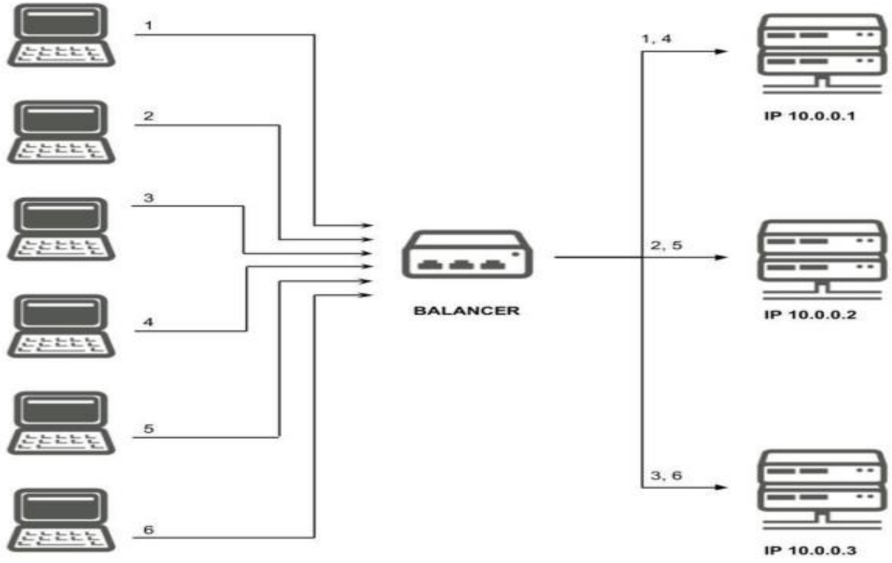


- Receive incoming traffic
- Estimate load size
- Lookup server instance
- Select a server instance using an algorithm

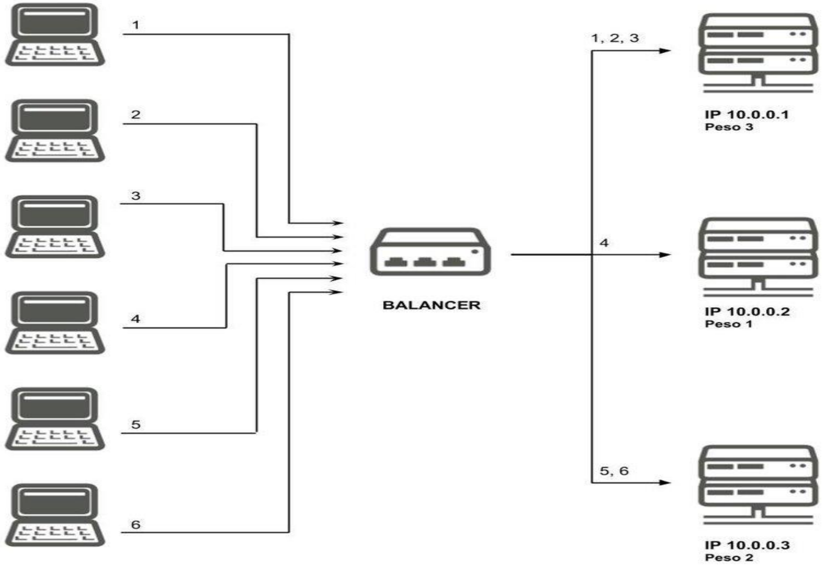
Source : *ResearchGate*

Load Balancing Algo

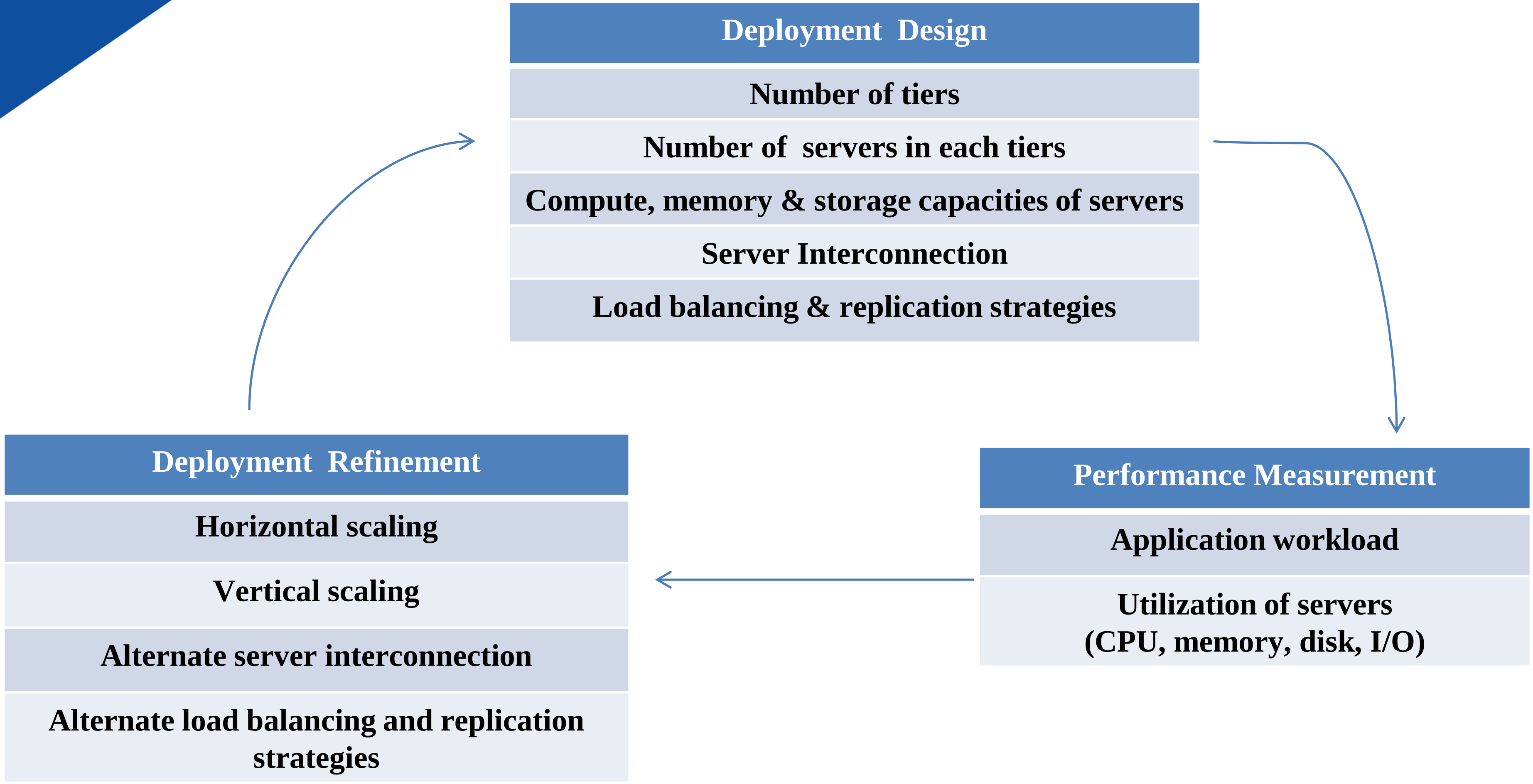
Round Robin



Weighted Round Robin



Deployment



Deployment Design

Number of tiers

Number of servers in each tiers

Compute, memory & storage capacities of servers

Server Interconnection

Load balancing & replication strategies

Performance Measurement

Application workload

Utilization of servers (CPU, memory, disk, I/O)

Deployment Refinement

Horizontal scaling

Vertical scaling

Alternate server interconnection

Alternate load balancing and replication strategies

Cloud application deployment lifecycle

Replication

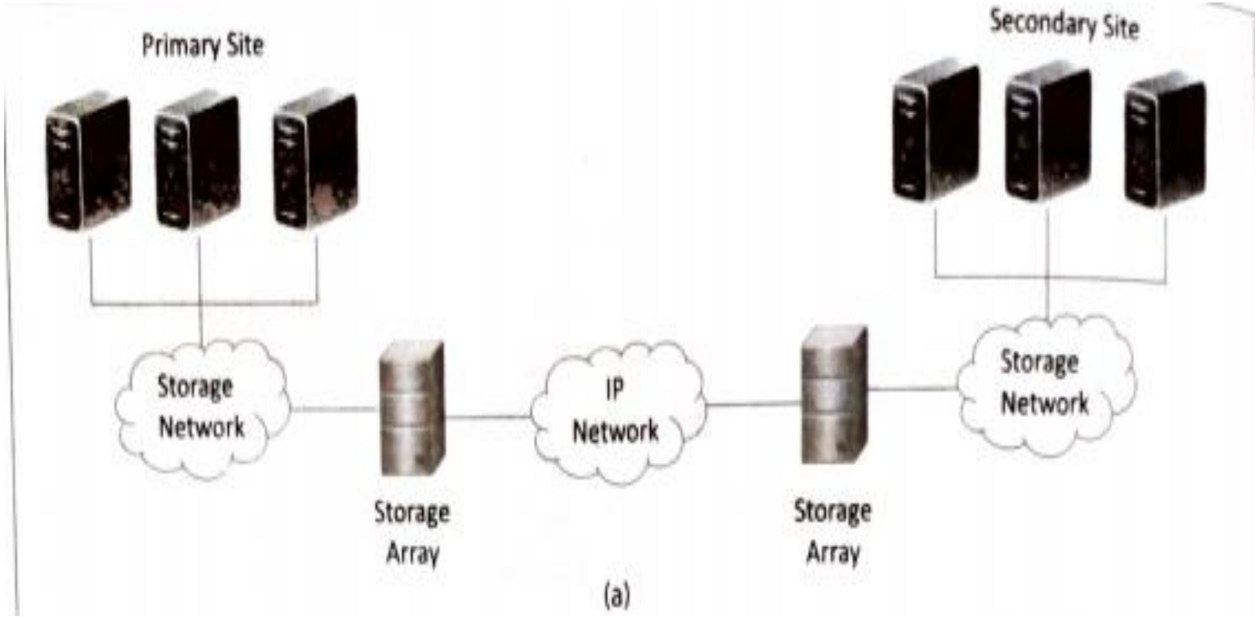
- Create & maintain multiple copies of data in cloud
- Helps in business continuity and disaster recovery
- Organizations can continue to operate their applications from secondary data source in the event of data loss at primary location
- Cloud based replication approaches provides replication in multiple locations, automated recovery, Low Recovery point Objective (LPO) and Low Recovery Time Objective (LTO)

LPO: Maximum amount of data – as measured by time – that can be lost after a recovery from a disaster, failure, or comparable event before data loss will exceed, RPO) determines

- How much data will be lost after a disaster or event
- How frequently you need to backup your data for disaster recovery

RTO: is the amount of real time a business has to restore its processes at an acceptable service level after a disaster to avoid intolerable consequences associated with the disruption.

Array-based Replication

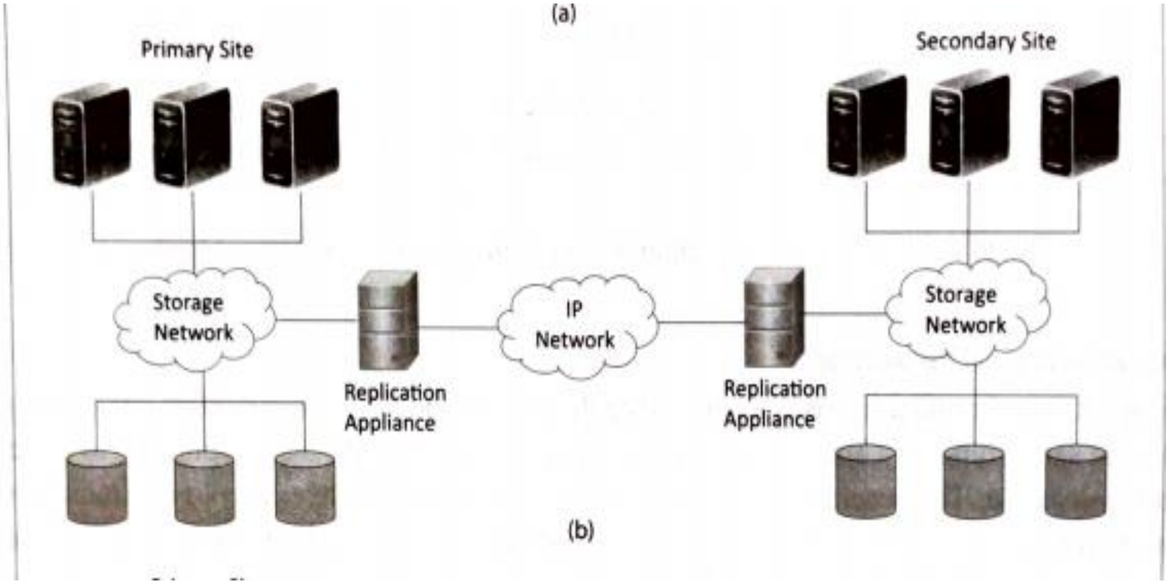


An array-based data replication strategy uses built-in software to automatically replicate data. With this type of data replication, the software is used in compatible storage arrays to copy data between each.

Advantages: More robust, Requires less coordination when deployed, The work gets offloaded from the servers to the storage device

Disadvantages: Requires homogenous storage environments: the source and target array have to be similar, It is costly to implement

Network-based Replication

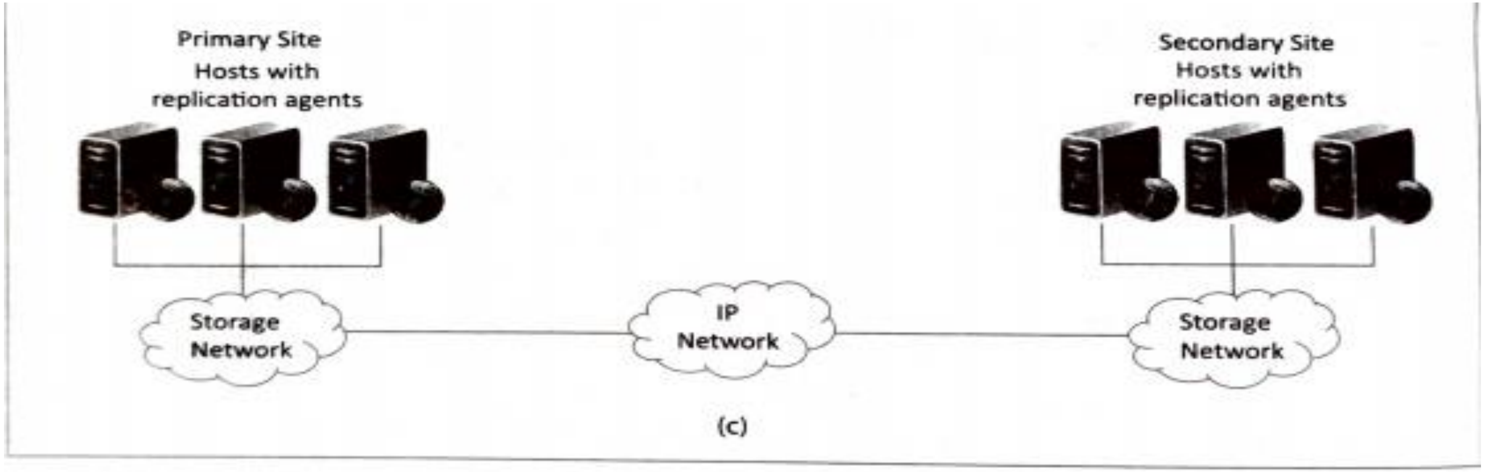


Uses a device or appliance that sits on the network in the path of the data to manage replication. The data is then copied to a second device. These devices usually have proprietary replication technology but can be used with any host server and storage hardware.

Advantages: 1. Effective in large, heterogeneous storage and server environments 2.Supports any host platform and works with any array. 3.Works separately from the servers and the storage devices 4. Allows replication between multi-vendor products

Disadvantages: Higher initial set-up cost because it requires proprietary hardware, as well as ongoing operational and management costs

Host-Based Replication



uses the servers to copy data from one site to another site. Host-based replication software usually includes options like compression, encryption and, throttling, as well as failover.

Advantages: 1.Flexible: It can leverage existing IP networks. 2. Can be customized to your business' needs: You can choose what data to replicate 3. Can create a schedule for sending data: allows you to throttle bandwidth 4.Can use any combination of storage devices on each end.

Disadvantages: 1.Both storage devices on each end need to be active, which means you will need to purchase dedicated hardware and OS. 2. Not all applications can support this type of data replication 3.Can be affected by viruses or application failure

Software Defined Networks

Software-defined networking (SDN): It is an architectural approach that optimizes and simplifies network operations by binding the interaction among applications, network services and network devices. SDN improves network performance and monitoring system by enabling dynamic and programatically efficient network configuration. It is achieved by adopting logically centralised network controlwhich is called as SDN controller

The network devices like Router, Swith are comprised of

Data plane: All the data packet moving activities like packet forwarding, Segmentation & reassembly of data, Replication of packets for multicasting belongs to Data plane.

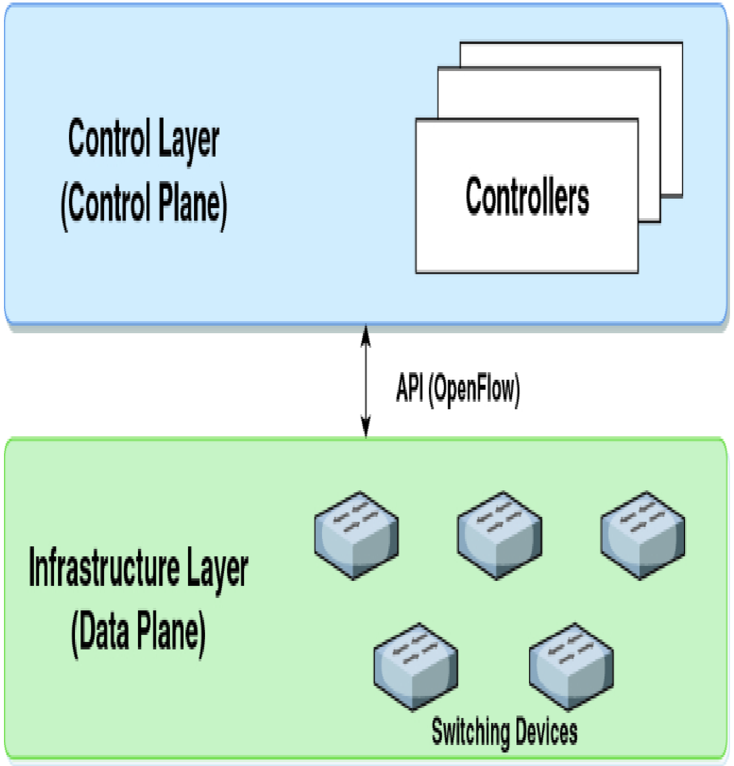
Control Plane: All the activities that controls the data plane activities belongs to this plane. Control plane do not involve end user data packets. It is the brain of the network. Construction of routing table, building packet handling policies are some of control plane activities.

Network device come with interfaces like CLI, GUI, Netconf etc., that allow a network operator to configure and manage these devices. Interfaces provide options that allows an operator to a access the device's capabilities, but they still often hide the lowest levels of details from the operator. To program these network devices the operator has to use the syntax or semantics of functionality that exists in a device.

Software Defined Networks

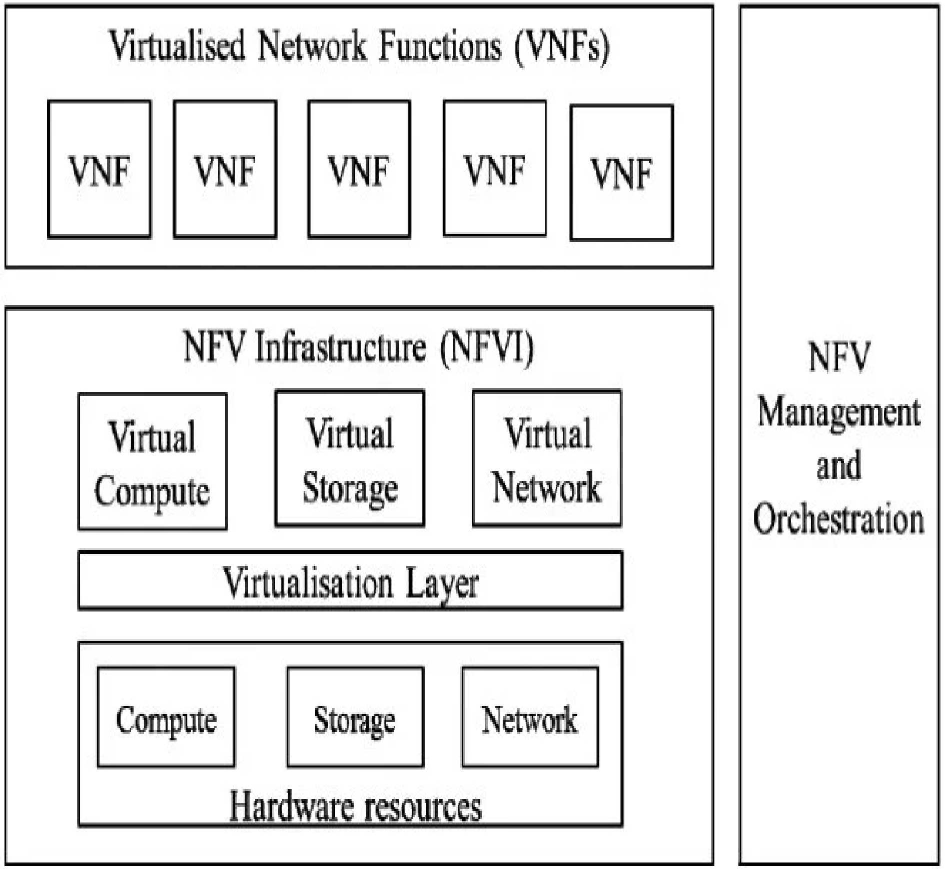
Distributed Control Plane: In conventional networks, every switch has its own data plane and control plane. The control plane of various switches exchange topology information and constructs a routing table to decides where an incoming data packet has to be forwarded via the data plane. As routing table is constructed individually this is called Distributed Control Plane

Centralized Control Plane: One single control plane/network brain would push commands to each device, thus commanding it to manipulate its physical switching and routing hardware. In CCP a network administrator can shape traffic via a centralised console without having to touch the individual switches. The forwarding activity is decided based on the entries in flow tables, which are pre-assigned by the controller.



Networks Function Virtualization

- Network Functions Virtualization (NFV) is the decoupling of network functions from proprietary hardware appliances and running them as software in virtual machines (VMs).
- NFV uses virtualized networking components to support an infrastructure totally independent of hardware.
- The data plane and control plane can also be virtualized with NFV.
- NFV virtualizes network infrastructure and SDN centralizes network control.
- Combined, SDN and NFV create a network that is built, operated, and managed by software.



NFV architecture standardized by ETSI

Identify and Access Management(IDAM)

IAM is one of the effective method to ensure cloud security

Identity : Proving who you are

Access : Opportunity to use resources

Management : Process of controlling or dealing with resources

Access Management: means access must fulfill the security policies (it is not that it must be hard to access)

Identity Access Management (IAM) : Deals with process (steps) and policies use to manage the users in accessing the critical resources
IAM allows only the authorized (identified) users to access the resources

Identify and Access Management(IDAM)

IAM goal: Grant Access to Right individual at Right time with right reason

Tasks of IAM: Authentication & Authorization

Authentication: The act of proving who you are

Common methods are

Form Based authentication(user name & password)

Multi Factor Authentication (MFA)

Certificates

Authorization: The act of granting access to specific resources or functions

Ex: Role-based access control
RBAC or Role-based access control technique is given to users as per their role or profile in the organization.

Identify and Access Management(IDAM)

IAM Tools :

1) User Identity management tools
2) User password management tools
3) User security policies
4) Login monitoring tools
5) Access management tools
6) Tools for Modification, Creation and Deletion of access

Service Level Agreement (SLA)

A service-level agreement (SLA) is a commitment between a service provider and a client. Particular aspects of the service, such as ***quality, availability, responsibilities*** are agreed upon between the service provider and the service user. It defines:

- The metrics used to measure the level of service provided.
- Remedies or penalties resulting from failure to meet the promised service level expectations.

Key Components of a Service-Level Agreement

Service-Level Parameter: Describes an observable property of a service whose value is measurable Availability, Response time, Latency, Throughput, Reliability

Metrics: Metrics are the key instrument to describe exactly what SLA parameters mean by specifying how to measure or compute the parameter values.

Measurement directives: These specify how to measure a metric.

Function: Functions are central to describing exactly how SLA parameters are computed from resource metrics.

Service Level Agreement (SLA)

In the perspective of application hosting **two types of SLAs**

- **Infrastructure SLA**: The infrastructure provider manages and offers guarantees on availability of the infrastructure, namely, server machine, power, network connectivity, and so on. Enterprises manage themselves, their applications that are deployed on these server machines.

- **Application SLA**: In the application co-location hosting model, the server capacity is available to the applications based solely on their resource demands. Hence, the service providers are flexible in allocating and de-allocating computing resources among the co-located applications.

Service Level Agreement (SLA)

Key Contractual Elements of an Infrastructural SLA

***Hardware availability*:** 99% uptime in a calendar month

Power availability : 99.99% of the time in a calendar month

***Data center network availability*:** 99.99% of the time in a calendar month

***Backbone network availability*:** 99.999% of the time in a calendar month

***Service credit for unavailability*:** Refund of service credit prorated on downtime period ***Outage notification guarantee* :** Notification of customer within 1 hr of complete downtime ***Internet latency guarantee*:** When latency is measured at 5-min intervals to an upstream provider, the average doesn't exceed 60 msec

Packet loss guarantee :Shall not exceed 1% in a calendar month

Service Level Agreement (SLA)

Key contractual components of an application SLA

Service-level parameter metric : Web site response time (e.g., max of 3.5 sec per user request)

Latency of web server (WS) (e.g., max of 0.2 sec per request)

Latency of DB (e.g., max of 0.5 sec per query)

Function : Average latency of WS= (latency of web server 1+ latency of web server 2) /2

Web site response time = Average latency of web server + latency of database

Measurement directive: DB latency available via http://mgmtserver/em/latency

WS latency available via http://mgmtserver/ws/instanceno/ latency

Service-level objective :Service assurance

web site latency , 1 sec when concurrent connection , 1000

Penalty : 1000 USD for every minute while the SLO was breached

Service Level Agreement (SLA)

LIFE CYCLE OF SLA

Each SLA goes through a sequence of steps starting from identification of terms and conditions, activation and monitoring of the stated terms and conditions, and eventual termination of contract once the hosting relationship ceases to exist. Such a sequence of steps is called SLA life cycle and consists of the following five phases:

1. Contract definition
2. Publishing and discovery
3. Negotiation
4. Operationalization
5. De-commissioning

Service Level Agreement (SLA)

1. **Contract definition** : service providers define a set of service offerings and corresponding SLAs using standard templates. These service offerings form a catalog. Individual SLAs for enterprises can be derived by customizing these base SLA templates

2. **Publishing and discovery**: Service provider advertises these base service offerings through standard publication media, and the customers should be able to locate the service provider by searching the catalog. The customers can search different competitive offerings and shortlist a few that fulfill their requirements for further negotiation.

3. **Negotiation:** Once the customer has discovered a service provider who can meet their application hosting need, the SLA terms and conditions needs to be mutually agreed upon before signing the agreement for hosting the application. (standard package /Customized)

4. **Operationalization:** SLA operation consists of SLA monitoring, SLA accounting, and SLA enforcement. SLA monitoring involves measuring parameter values and calculating the metrics defined as a part of SLA and determining the deviations. On identifying the deviations, the concerned parties are notified. SLA accounting involves capturing and archiving the SLA adherence for compliance.

5. **De-commissioning:** involves termination of all activities performed under a particular SLA when the hosting relationship between the service provider and the service consumer has ended.

Billing

Cloudonomics

Migrating into the cloud is driven by economic reasons of cost cutting in both the IT capital expenses (Capex) as well as operational expenses (Opex). Migrating into cloud leads to both short term and long term benefits.

Cloudonomics is the Economics of Cloud Computing. It is the expression of when a migration can be economically feasible or tenable.

According to *Cloudonomics*, if the average costs of using an enterprise application on a cloud is substantially lower than the costs of using it in one’s captive data center and if the cost of migration does not add to the burden on ROI, then the case for migration into the cloud is strong.

Billing

- Billing is **the process of attributing resource usage to cloud tenants and creating appropriate invoices**.
- Pay-per-use is a core principle of cloud computing
- Economic appeal of Cloud Computing: “converting capital expenses to operating expenses (CapEx to OpEx)” or “pay as you go”. Hours purchased can be distributed non-uniformly in time.
- The cost of a cloud computing deployment is roughly estimated to be

$$\text{Cost}_{\text{CLOUD}} = \Sigma(\text{UnitCost}_{\text{CLOUD}} \text{ x } (\text{Revenue} - \text{Cost}_{\text{CLOUD}}))$$

Where the unit cost is usually defined as the cost of a machine instance per hour or another resource. Depending upon the deployment type, other resources add additional unit costs: storage quantity consumed, number of transactions, incoming or outgoing amounts of data, and so forth.

Billing

In theory, therefore, the CostCLOUD is better represented by the equation:

CostCLOUD = ${}_1{}^{n}\Sigma(\text{UnitCost}_{\text{CLOUD}} \text{ x } (\text{Revenue} – \text{Cost}_{\text{CLOUD}}))\text{INSTANCEn} +$
${}_1{}^{n}\Sigma(\text{UnitCost}_{\text{CLOUD}} \text{ x } (\text{Revenue} – \text{Cost}_{\text{CLOUD}}))\text{STORAGE\_UNITn} +.$
${}_1{}^{n}\Sigma(\text{UnitCostCLOUD x (Revenue – CostCLOUD))NETWORK\_UNITn} + \ldots$

https://calculator.s3.amazonaws.com/index.html

Reference Architecture for Cloud Applications

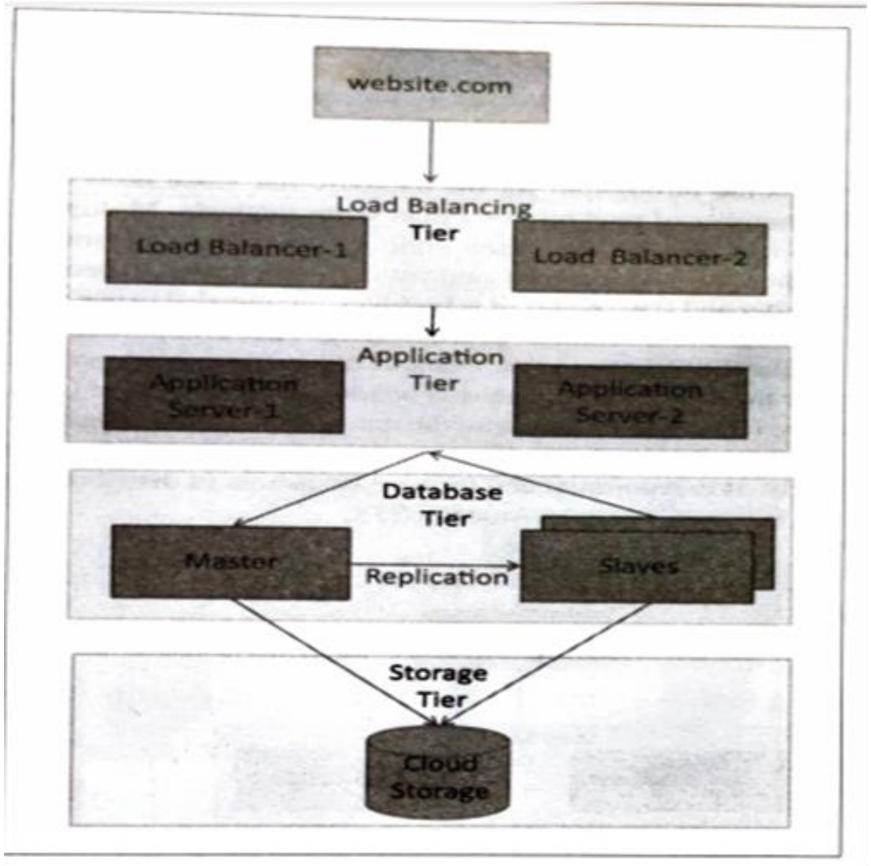
Deployment architecture for e-comm, B-2-B, Banking Applications

Cloud Applications are multi-tier and have various deployment architectures

Load Balancing Tier: Minimum 2 instances of load balancer to avoid single point of failure, also preferred to provision balancer instances in separate zone of service provider for better reliability and availability

Application Tier: consist of one or more application servers, triggered auto scaling of servers is required when, CPU/memory/ storage exceeds defined threshold. Min 2 server running at all times to avoid single point of failure

Database Tier: contains a master database instance which servers write request and multiple slave instances which serves read requests and also acts as backup for server. This improves the performance as there will be few write request and more read request. Recommended to use disk sub systems (AWS S3) rather than storage instance

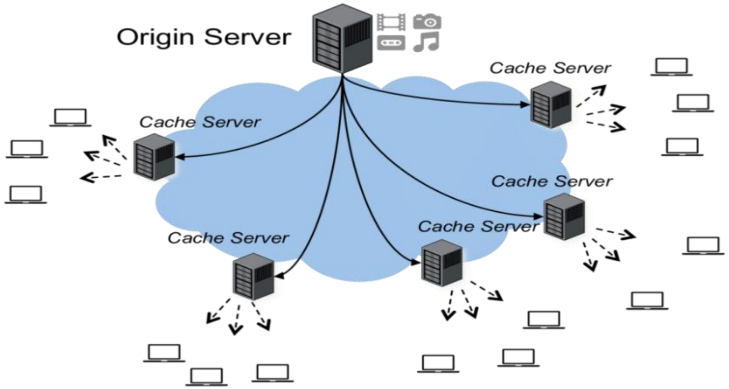


Deployment architecture for e-comm, B-2-B, Banking Applications

Reference Architecture for Content delivery Applications

Online photo albums, video webcasting, both relational and non-relational data stores are used. Content Delivery Network (CDN)

A content delivery network (CDN) is **a group of geographically distributed servers that speed up the delivery of web content by bringing it closer to where users are**. ... CDNs cache content like web pages, images, and video in proxy servers near to your physical location.



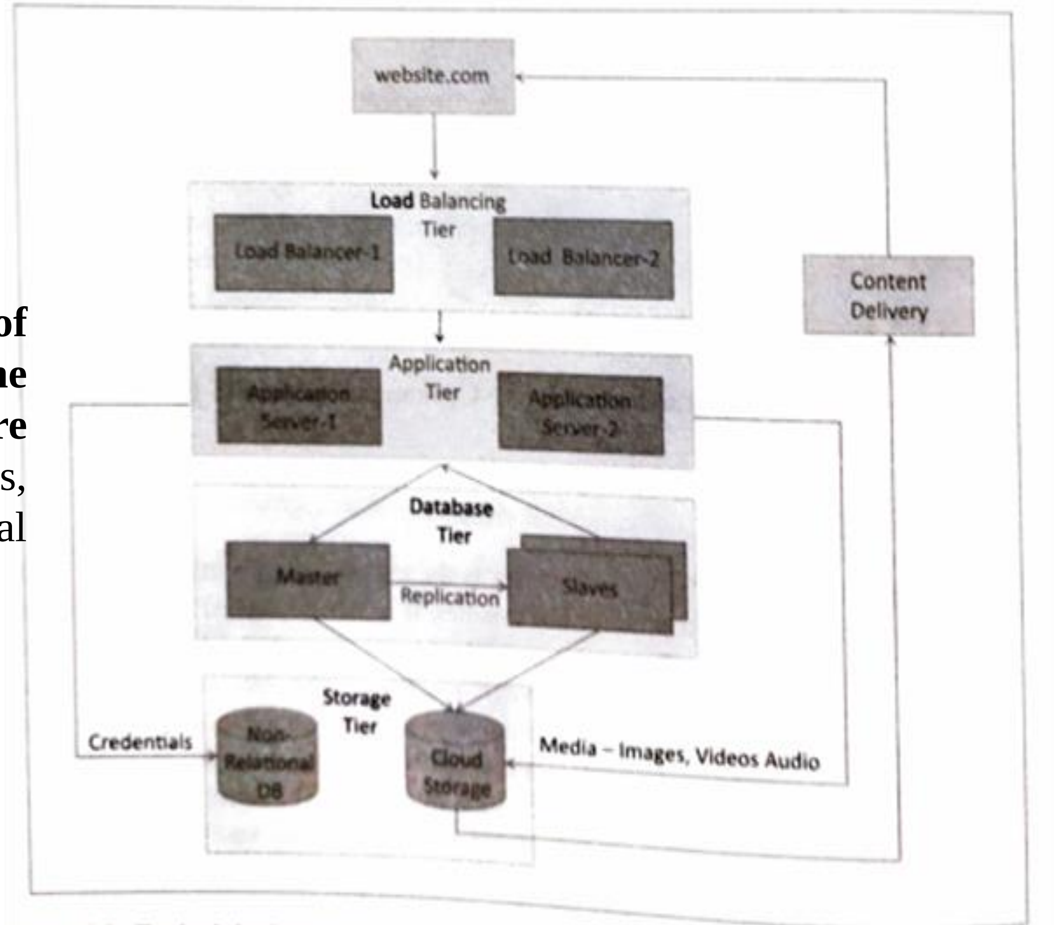
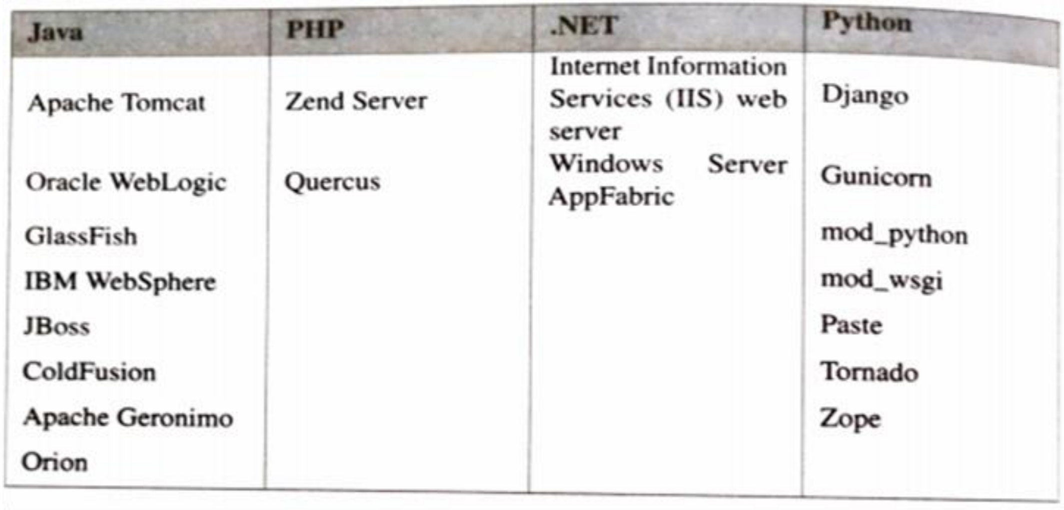


Figure 5.2: Typical deployment

Reference Architecture for Compute intensive Applications

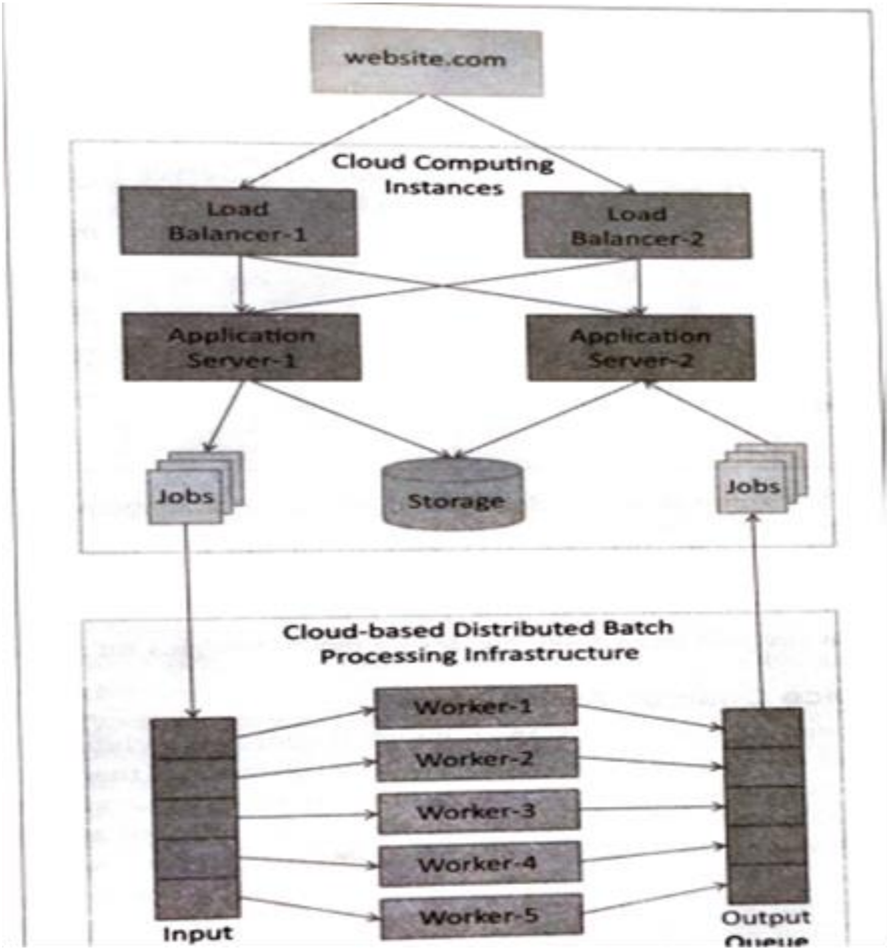
Data Analytics, Media Transcoding are compute intensive

Analytics tier consists of cloud based distributed batch processing frameworks such as Hadoop (big data analytics), jobs are submitted to analytics tier from application server



Java	PHP	.NET	Python
Apache Tomcat	Zend Server	Internet Information Services (IIS) web server	Django
Oracle WebLogic	Quercus	Windows Server AppFabric	Gunicorn
GlassFish			mod_python
IBM WebSphere			mod_wsgi
JBoss			Paste
ColdFusion			Tornado
Apache Geronimo			Zope
Orion			

Cloud deployment management tools



Cloud Application Design Methodologies

The design methodologies for cloud applications

- Service Oriented Architecture
- Cloud Component Model
- IaaS, PaaS and SaaS services for Cloud Application
- Model View Controller
- RESTfull Web Services

Service Oriented Architecture

Approach for designing and developing applications in the form of services that can be shared and reused.
SOA is a collection of discrete S/W modules or services that forms a part of a business application.

SOA: makes software components reusable via service interfaces using common communication standards in such a way that they can be rapidly incorporated into new applications without having to perform deep integration each time.

Advantages: Easy to integrate, Loose coupling, platform independent.

Service: service in an SOA embodies the code and data integrations required to execute a complete, discrete business function. e.g., checking a customer’s credit, calculating a monthly loan payment, or processing a mortgage application. Services communicates through SOAP which allows the exchange of information between web services.
WSDL and SOAP provides web services over internet.

Web Service Description Language

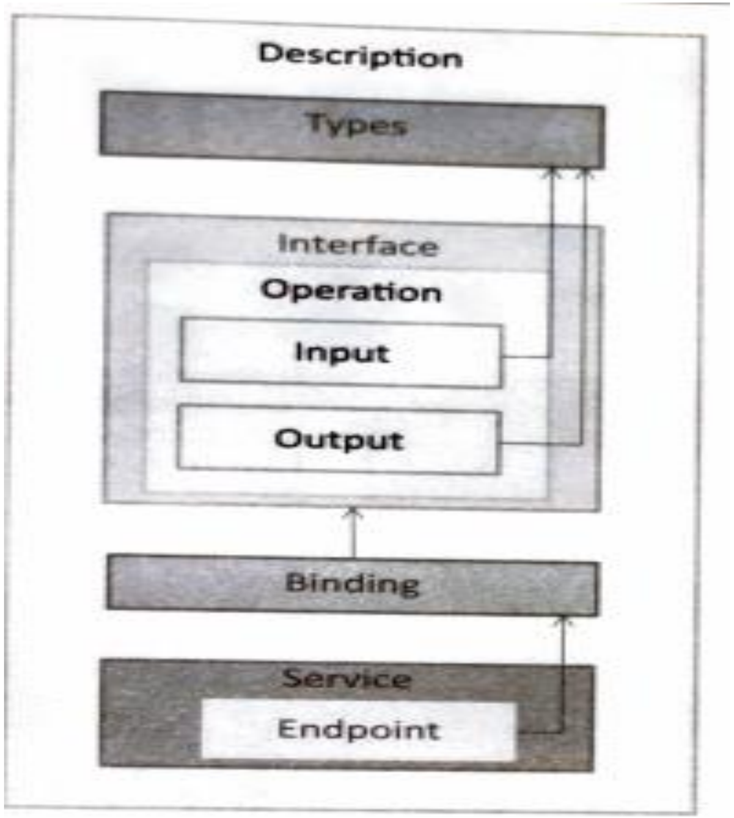
Web Service Description Language, is an XML based definition language. It’s used for describing the functionality of a SOAP based web service.

Binding: describes how the service is bound to a messaging protocol, particularly the SOAP messaging protocol. A WSDL SOAP binding can be either a Remote Procedure Call (RPC) style binding or a document style binding.

Interface : describes the generic web service, including the port type, messages, parts of the messages, and bindings

Operation − defines how the message is decoded and the actions that can be performed.

Data types − The data types to be used in the messages are in the form of XML schemas.



Web Service Description Language

PortType

```
<portType name="EightBall">
   <operation name="getAnswer">
      <input message="ebs:IngetAnswerRequest"/>
      <output
message="ebs:OutgetAnswerResponse"/>
   <operation/>
<portType>
```

Type

```
<types>
   <xsd:schema targetNamespace="...">
      <xsd:complexType name="questionType">
         <xsd:element name="question" type="string"/>
      </xsd:complexType>
      <xsd:complexType name="answerType">
```

Message

```
<message name="IngetAnswerRequest">
   <part name="meth1_inType"
type="ebs:questionType"/>
</message>
<message name="OutgetAnswerResponse">
   <part name="meth1_outType"
type="ebs:answerType"/>
</message>
```

Binding

```
<binding name="EightBallBinding" type="ebs:EightBall">
   <soap:binding style="rpc"
transport="schemas.xmlsoap.org/soap/http">
   <operation name="ebs:getAnswer">
   <soap:operation soapAction="urn:EightBall"/>
      <input>
         <soap:body namespace="urn:EightBall" ... />
```

SOA Layered Architecture

Business System: Contains custom built applications like ERP, CRM, SCM etc.

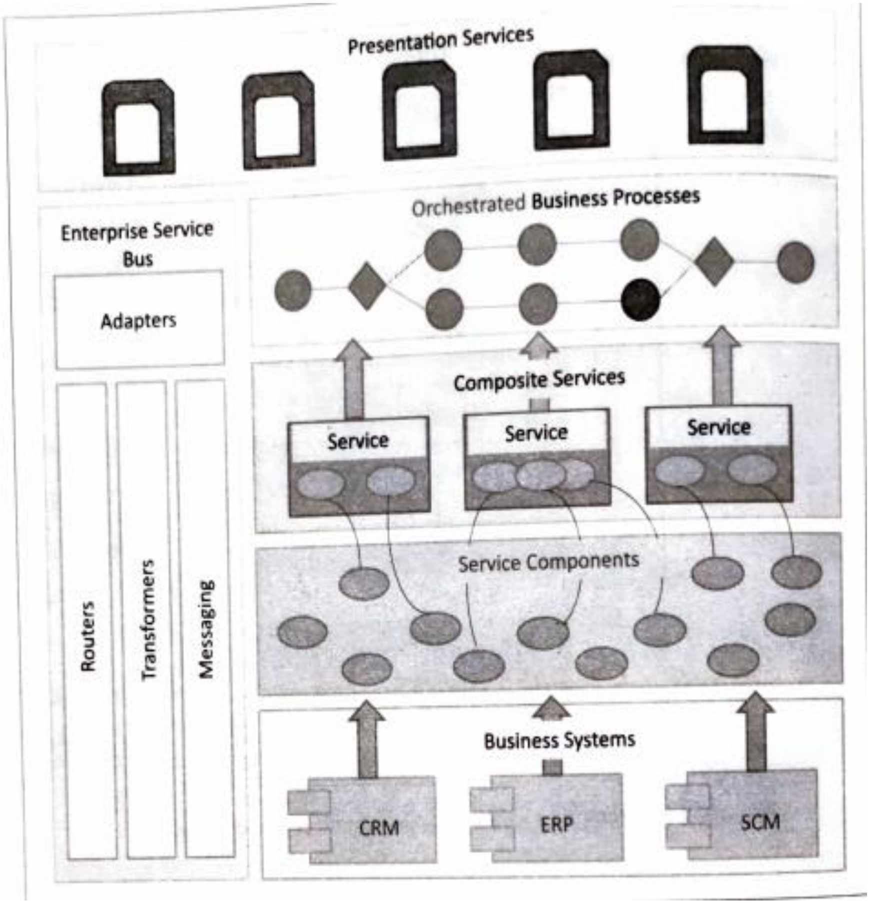
Service Components: Allows the layers above to interact with business systems and responsible for realizing the functionality.

Composite Services: composed of two or more service components to create business-unit specific components

Business Process: composite services are planned and arranged to create higher level business process

Presentation Services: Top most layer which provides interfaces to expose the services

Enterprise service bus (ESB) is an architecture that allows communication between different environments, such as software applications.



INTEGRATION AS A SERVICE' PARADIGM FOR THE CLOUD ERA

Clouds guarantee most of the non-function requirements (Quality of Service (QoS) attributes) such as availability, high performance, on-demand scalability/elasticity, affordability, global-scale accessibility and usability, energy efficiency etc.

On-premise and local applications are becoming online, remote, hosted, on-demand and off-premise applications.

Fresh applications are being implemented and deployed on clouds to be delivered to millions of global users simultaneously affordably are increasing.

All these portend and predict that there is a new dimension to the integration scenario.

Business-to-business (B2B) integration is being attended via special data formats, message templates, and networks and even via the Internet. Enterprises consistently expand their operations to several parts of the world as they establish special partnerships with their partners or buy other companies in different geographies for enhancing the product and service portfolios.

Hence it is logical to take the integration middleware to clouds to simplify and streamline the enterprise-to-enterprise (E2E), enterprise-to-cloud (E2C) and cloud-to-cloud (C2C) integration.

The Evolution of SaaS

IT as a Service (ITaaS) : IT as a service (ITaaS) is an operational model where the information technology (IT) service provider delivers an information technology service to a business.

The IT service provider can be internal IT organization or an external IT services company, Under an ITaaS model, the IT service provider will place great emphasis on the needs and the outcomes required by the business to improve employee productivity and improving the top line (revenue) and bottom line (profitability).

‘IT as a service (ITaaS)’. is highlighted due to the pervasive Internet. Clouds, is the most visible and viable infrastructure for realizing ITaaS.

Integration as a service (IaaS): Integration as a Service (IaaS) is **a cloud-based delivery model that strives to connect on-premise data with data located in cloud-based applications**. ... In business-to-business (B2B) integration, IaaS allows partners to develop, maintain and manage custom integrations for diverse systems and applications in the cloud. IaaS just imitates this established communication and collaboration model to create reliable and durable linkage for ensuring smooth data passage between traditional and cloud systems over the Web infrastructure

APPROACHING THE SaaS INTEGRATION

Integration as a Service (IaaS) is all about the migration of the functionality of a typical enterprise application integration (EAI) hub / enterprise service bus (ESB) into the cloud for providing for smooth data transport between any enterprise and SaaS applications.

Due to varying integration requirements and scenarios, there are a number of middleware technologies
Middleware is software that enables one or more kinds of communication or connectivity between two or more applications or application components in a distributed network

SRM Domain Within Reference Architecture of CSA

The Security and Risk Management domain provides the core components of an organization's Information Security Program to safeguard assets and detect, assess, and monitor risks inherent in operating activities. Capabilities include Identity and Access Management, GRC (Governance, Risk Management, and Compliance), Policies and Standards, Threat and Vulnerability Management, and Infrastructure and Data Protection.

- Security and Risk Management is the passwords, firewalls, and encryption that protect computer systems and data.
- It is the processes that define policies and audit systems against those policies.
- It uses ethical hackers and tools to test for weak spots in the systems. These services are what most people think of when they think of cyber security.

SRM Domain Within Reference Architecture of CSA

The Sub-domains of SRM

- **Governance Risk Management and Compliance:** This component is responsible for

  Compliance management assures compliance with all internal information security policies and standards

  Vendor Management: to ensure that service providers and outsourcers adhere to intended and contractual information security policies applying concepts of ownership and custody

  Audit Management to highlight areas for improvement

  IT risk management to ensure that risk of all types are identified, understood, communicated, and either accepted, remediated, transferred, or avoided

  Policy management to maintain an organizational structure and process that supports the creation, implementation, exception handling and management of policy that represent business requirements

  Technical awareness and training to increase the ability to select and implement effective technical security mechanisms, products, process and tools.

SRM Domain Within Reference Architecture of CSA

The Sub-domains of SRM

- **Information Security Management:** The main objective of Information Security Management is to implement the appropriate measurements in order to minimize or eliminate the impact that security-related threats and vulnerabilities might have on an organization. Measurements include **Capability Maturity Models**(from an immature state through several levels of maturity), **Capability Mapping Models**, **Risk Portfolios** (where identified risks are registered, monitored, and reported). **Dashboards** for security management and risk management to improve Analysis and plans for remediating residual risks are also part of the overall risk management framework.

- **Privilege Management Infrastructure:** Privilege Management Infrastructure ensures users have access and privileges required to execute their duties and responsibilities with Identity and Access Management (IAM) functions such as identity management, authentication services, authorization services, and privilege usage management. The technical controls of Privilege Management Infrastructure focus on identity provisioning, password, multi-factor authentication, and policy management.

SRM Domain Within Reference Architecture of CSA

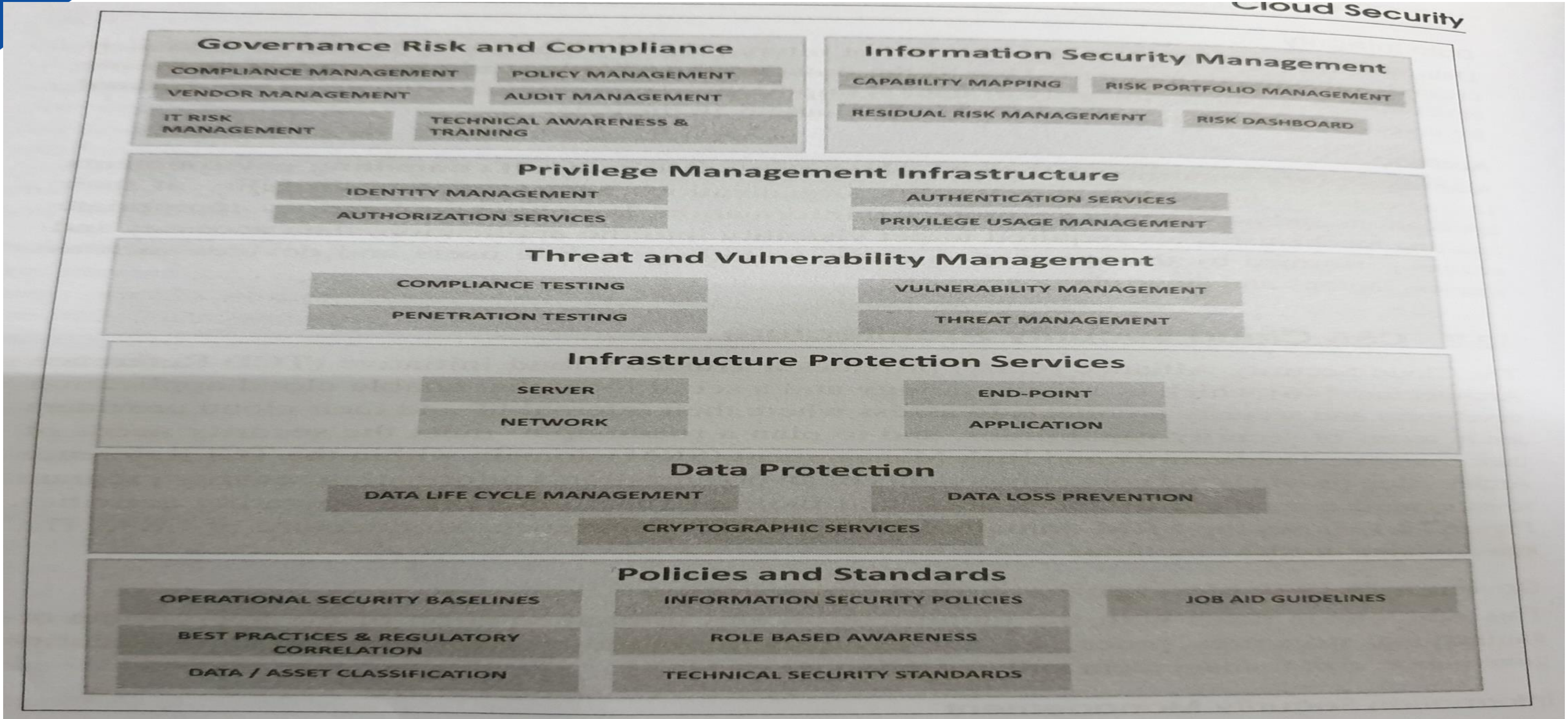
The Sub-domains of SRM

- **Threat and Vulnerability Management:** This discipline deals with core security, such as vulnerability management, threat management, compliance testing, and penetration testing. Vulnerability management is a complex endeavor in which enterprises track their assets, monitor and scan for known vulnerabilities, and take action by patching the software, changing configurations, or deploying other controls in an attempt to reduce the attack surface at the resource layer. Threat modeling and security testing are also part of activities in order to identify the vulnerabilities effectively.

- **Infrastructure Protection Services:** Infrastructure Protection Services secure Server, End-Point, Network and Application layers. The controls of Infrastructure Protection Services are usually considered as preventive technical controls such as IDS/IPS, Firewall, Anti-Malware, White/Black Listing and more. They are relatively cost-effective in defending against the majority of traditional or non-advanced attacks.

- **Data Protection:** Data is an asset. Data protection needs to cover all data lifecycle stages, data types, and data states. Data stages include create, store, access, roam, share, and retire. Data types include unstructured data, such as word processing documents, structured data, such as data within databases, and semi-structured data, such as emails. Data states include data at rest (DAR), data in transit (DIT) (also known as “data in motion” or “data in flight”), and data in use (DIU). The controls of Data Protection are data lifecycle management, data leakage prevention, intellectual property protection with digital rights management, and cryptographic services such as key management and PKI/symmetric encryption.

SRM Domain Within Reference Architecture of CSA



Cloud Security

Governance Risk and Compliance
COMPLIANCE MANAGEMENT
POLICY MANAGEMENT
VENDOR MANAGEMENT
AUDIT MANAGEMENT
IT RISK MANAGEMENT
TECHNICAL AWARENESS & TRAINING

Information Security Management
CAPABILITY MAPPING
RISK PORTFOLIO MANAGEMENT
RESIDUAL RISK MANAGEMENT
RISK DASHBOARD

Privilege Management Infrastructure
IDENTITY MANAGEMENT
AUTHENTICATION SERVICES
AUTHORIZATION SERVICES
PRIVILEGE USAGE MANAGEMENT

Threat and Vulnerability Management
COMPLIANCE TESTING
VULNERABILITY MANAGEMENT
PENETRATION TESTING
THREAT MANAGEMENT

Infrastructure Protection Services
SERVER
END-POINT
NETWORK
APPLICATION

Data Protection
DATA LIFE CYCLE MANAGEMENT
DATA LOSS PREVENTION
CRYPTOGRAPHIC SERVICES

Policies and Standards
OPERATIONAL SECURITY BASELINES
INFORMATION SECURITY POLICIES
JOB AID GUIDELINES
BEST PRACTICES & REGULATORY CORRELATION
ROLE BASED AWARENESS
DATA / ASSET CLASSIFICATION
TECHNICAL SECURITY STANDARDS

Figure 12.1: Security and Risk Management (SRM) domain within the TCI Reference