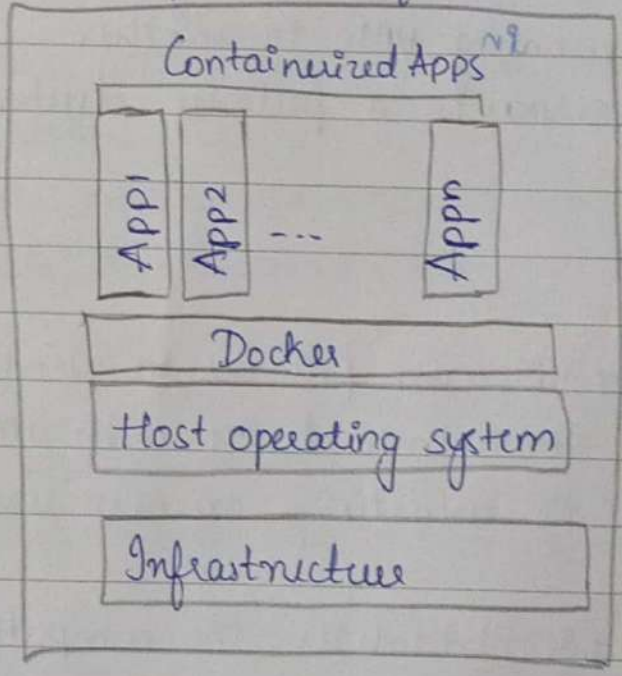
4)

Virtual machines (VM)	Containers
* More weight compaitively	* Lightweight
* VM's are measured in gigabytes	* Measured in megabytes
* It is hardware level virtualization	* They are OS level virtualization.
* It has its own OS	* Shared OS of host system kernel
* Less portable	* More portable
* More space is required	* Less space is required



* Runs complete OS including Kernel, thus requires more system resources	* Runs on the same OS type of the host and runs the user mode portion of the OS and can be tailored to contain just the needed services for the app, using fewer system resources
* Runs any OS in virtual machine	* Runs the the same os version as the host

* Contains OS images for each VM	* Shares host Kernel space
* Works at physical level	* Works at application level
* Provides complete isolation from the host OS and other VMs	* Typically provides lightweight isolation from host and other containers, but doesn't provide strong security boundary as VM.
* VM load balancing moves running VMs to another servers in a failover cluster	* Containers themselves don't move instead an orchestrator can automatically start or stop containers on cluster nodes to manage changes in load and availability
* VMs can failover to another server in a cluster, with VMs OS restarting on new server	* If a cluster node fails, any containers running on it are rapidly recreated by the orchestrator on another node
* It virtualizes the computer system.	* It virtualizes OS only.
* It is more secure	* It is less secure
* VM takes minutes to run	* It takes seconds to run
* they are useful when we require all of OS resources to run various applications	* Useful when we are required to maximize the running application using minal server.
Eg: VMware, KVM, Xen	Eg: Rancher OS, Containers by Docker

Benefits of containers

1) Less overhead: Containers requires less system resources than that of traditional or hardware virtual machines as they don't include operating system images.

2) Increased portability: Applications running on containers can be deployed easily to multiple different OS and hardware platforms.

3) More consistent operations: DevOps teams ~~the~~ knows that that application will run the same way, no matter where it is deployed.

4) Greater efficiency: Containers allow applications to be more rapidly deployed, scaled or patched

5) Better application development: Containers support agile and DevOps efforts to accelerate development, test and production cycles

Common ways in which containers can be used

1) "Lift and shift" existing applications into modern cloud architecture.

→ Some organizations use containers to migrate existing applications into more modern environments. While this practice delivers some of the basic benifits of operating system virtualization, it does not offer the full benefits of a modular, container-based application architecture.

2) Refactor existing applications for containers

→ Although refactoring is much more intensive than lift-and-shift migration, it enables the full

benefits of a container environment.

3) Develop new container-native applications
→ Much like refactoring, this approach unlocks the benefits of containers.

4) Provides better support for microservices architecture.
→ Distributed applications and microservices can be more easily isolated, deployed, and scaled during individual containes building blocks.

5) Provide DevOps support for continous integration and deployment (CI/CD).
→ Containe technology supports streamlined build, test and deployment from the same containes images.

6) Provides easier deployment of repetitive jobs and tasks
→ Containers are being deployed to support one or more similar processes, which often run in the background, such as ETL functions or batch jobs.

(c)

Public Cloud: Computing in which service provider makes all resources public over Internet. It is most pervasive and well-known computing model. All the resources needed to run the infrastructure - servers, storage, networking components and supporting software - are owned and managed by third-party provider, and accessed by users via Internet. It may be free-of-cost or with minimal pay-per-usage
Ex: Google Drive, Google Slides
Public Clouds are AWS, Google Cloud, MS Azure

Merits:
1) Low costs and no maintainenace: It is hazel-free as we need not maintain the infrastructure. Pricing is usually free or minimal having pay-as-you-go policy

2) Easy to use and flexible: We need not bother about back-end just use it and it's flexible to meet unpredictable demand

3) On-demand scalability and reliability: Providers maintain immense resources so whenever the need is high we can easily scale-up and scale-down when the demand is less. They are highly-reliable to ensure customers against outages and downtime.

Demerits: 1) Lack of security / privacy: It is least secured as all the data is on cloud. Hence not suitable for sensitive applications / data.
2) Limited infrastructure visibility as it is maintained by the CSP's
3) High expense: for large-scale use the total cost of ownership is high

Private Cloud: It is a computing infrastructure devoted to use by single organization. It has only authentic-users and single-occupant architecture. All the resources are run and maintained on private-network for use. It is customizable to meet unique business and security needs of organization

Merits:

1) Security: Only authentic users can acess
2) Suitable when data is sensitive.
3) Scalability without tradeoffs: High scalability and efficiency to meet unpredictable demands without compromising on security and performance
4) Flexibility: The infrastructure is transparent and is flexible as we can transform the infrastructure based on the needs
5) Exclusive environments: Dedicated and secure environments that cannot be accessed by other organizations

Demerits:

1) Cost: More capital is required
2) Maintainence and human resource: We needs specialite to create, build and maintain the cloud in organization. Maintainence cost can be a burden leading to lengthy development cycles
3) Scaling limitations: Scaling up can a be resource and time-intensive owing to neccessity to purchase and configure hardware and enabling software

Hybrid Cloud: They combine public and private cloud resources to yield advantage of both. Apps and data workloads can share the resources between public and private cloud deployment based on organizational needs. The critical process and data might be kept private and other information or process are made public.

Merits:

1) Security: Confidential applications can be operated privately and less-sensitive workloads can be deployed to a public cloud. Like this it creates flexible environment
2) Resource optimization: flexibly deploy applications to maximize utilization of both on-premises resources and cost-saving public infrastructure.
3) Reliability: Distributing services across multiple data centers, some public, some private, results in maximum reliability.
4) Scalability: Scale it to meet unpredictable needs.

Demerits:

1) Cost: Running, maintaining and optimizing the on-premise segment of a hybrid cloud is an expensive proposition, especially for smaller organization. Human resource is required for building and maintaining.
2) Management costs are high: Strong compatibility and integration is required between cloud infrastructure spanning different locations.
3) Complex to build.

Private Cloud	Public Cloud	Hybrid Cloud
→ Deployment model solely works for single organization	Public cloud deployment model that renders services over a network for public use	A composition of private and public clouds that offer benefits of multiple deployment models.
→ Highly Secured	Less Secure	More secure than public cloud
→ High maintainence cost	Low maintainence cost	High maintainence cost
→ Accessed via private network	accessed via public network	accessed via public or private network
→ More expensive	Requires a minimum cost	Cost-effective than private cloud
→ Suitable for confidential workloads	Not suitable for confidential workloads	has provision of private computing for confidential work loads
→ The computing resources remain behind organization's firewall	The infrastructure is managed by a third party service provider	Provides a mixed-service environment using both public and private cloud services
Large enterprises and public-sector bodies are targets Ex: KLE private Cloud	Start-up's and end-users Ex: Google sheets, docs	Large enterprises and public-sector bodies Ex: Amazon

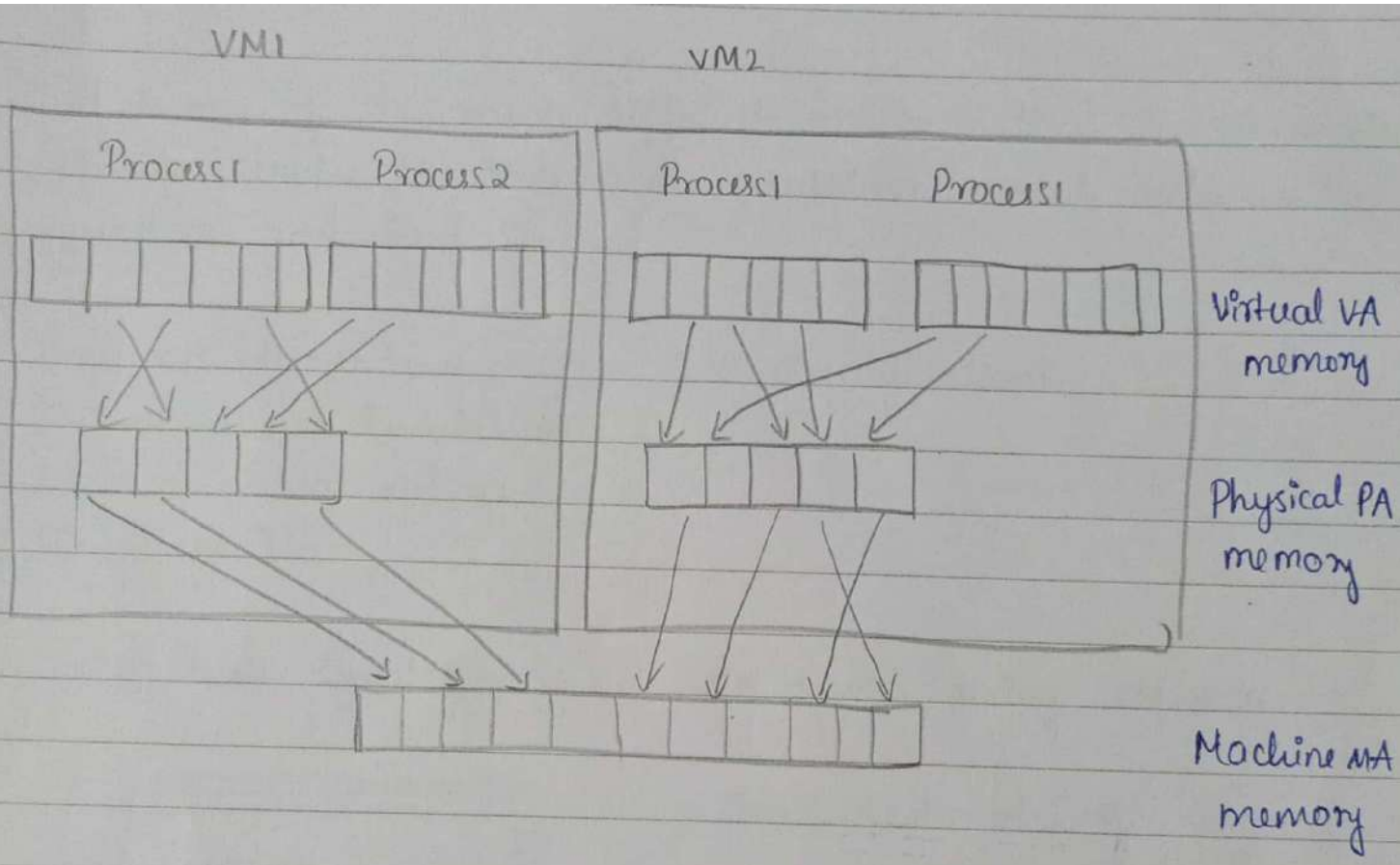
Considering the difference between all of them.
Hybrid cloud is best as it takes benefits from both public and private. Secured tasks can be in private cloud and others in public. Maintainence is minimal compared to full private cloud as here some part is made public. But compared to public its more. But its cost effective than private cloud.

If someone needs no maintainence costs then public cloud is best. But for organizations where data is crucial no huge human resources are available hybrid cloud is best suited.

But in view for CSP's maintainence is more for private cloud.

Users can take best benefits from public cloud as its free and no maintainence is involved.

2a) I/o virtualization is most difficult ones as memory mappings could be done easily. Direct access in case I/o virtualization is more prone to security risks as user directly uses the hardware could modify it and make it corrupt and create a deadlock situation as there is no governing entity to tell him/her to release it.



OS maintains mappings of virtual memory to machine memory using page tables, which is one-stage mapping from virtual memory to machine memory. CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. Two stage mapping process should be maintained by the guest OS and the VMM virtual memory to physical memory and physical memory to machine memory.

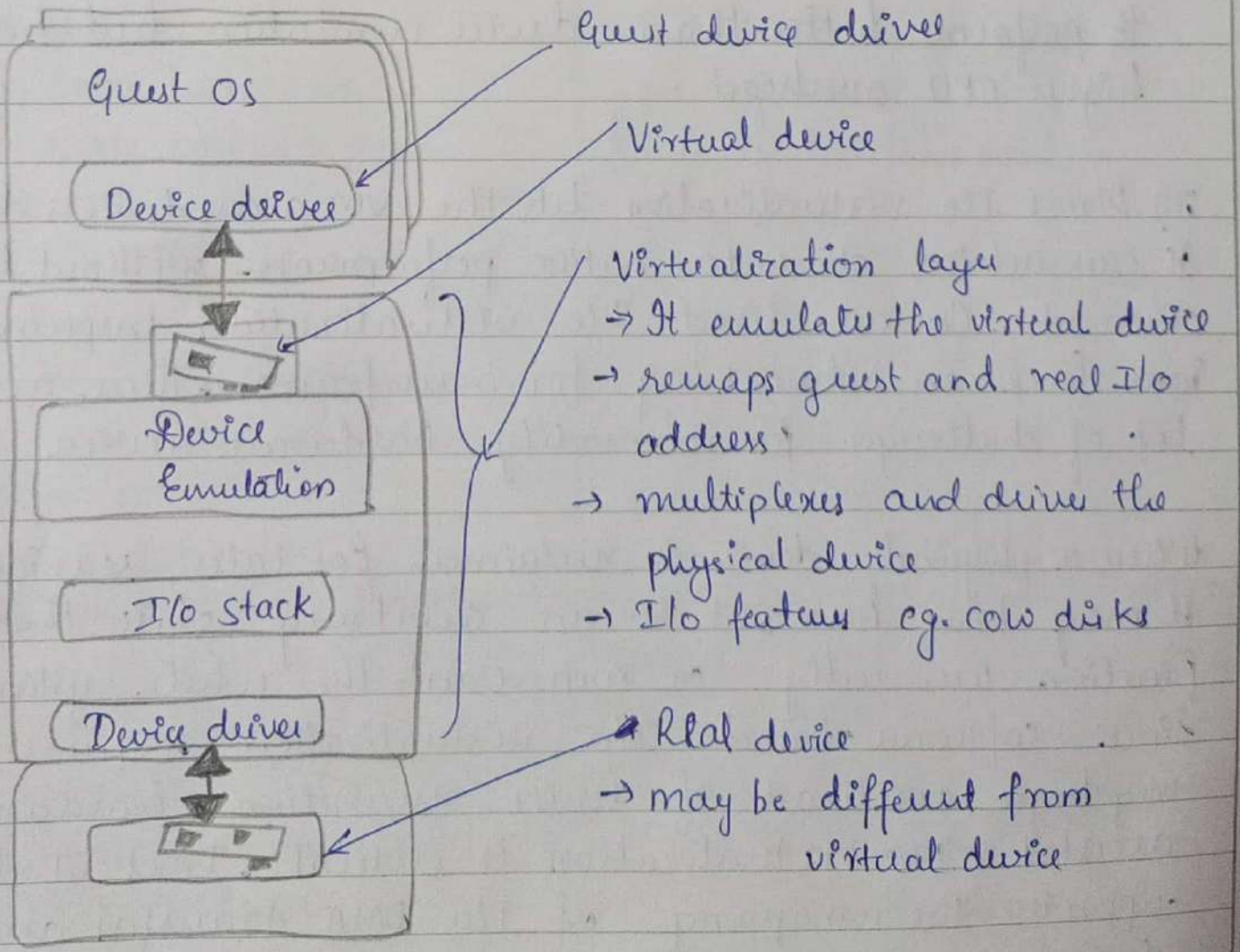
Guest OS VMM
Virtual memory ⟶ Physical memory ⟶ Machine memory

Guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. VMM is responsible for mapping the guest physical memory to actual machine memory

It is one of the most difficult one to realize due its complexity if I/O service routines and emulation needed between the guest OS and host OS.

There are three ways for I/O virtualization:
i) Full device Emulation
ii) Para-virtualization
iii) direct I/O

Full-Device Emulation - Emulates device using software



ii) Para - Virtualization

The para virtualization method of I/O virtualization is typically used in Xen. It is also known as the split driver model consisting of a frontend and a backend driver

The frontend driver is running in Domain U and the backend driver is running in Domain 0. They interact with each other via a block of shared memory. The frontend driver manages the I/O requests of the guest OSes and the backend driver manages the real I/O devices and multiplexing the I/O data of different VMs.

It performs better than device emulation, but has higher CPU overhead

iii) Direct I/O virtualization lets the VM access devices directly It can achieve close-to-native performance without high CPU costs. Current direct I/O virtualization implementations focus on networking for mainframes. There are a lot of challenges for commodity hardware devices.

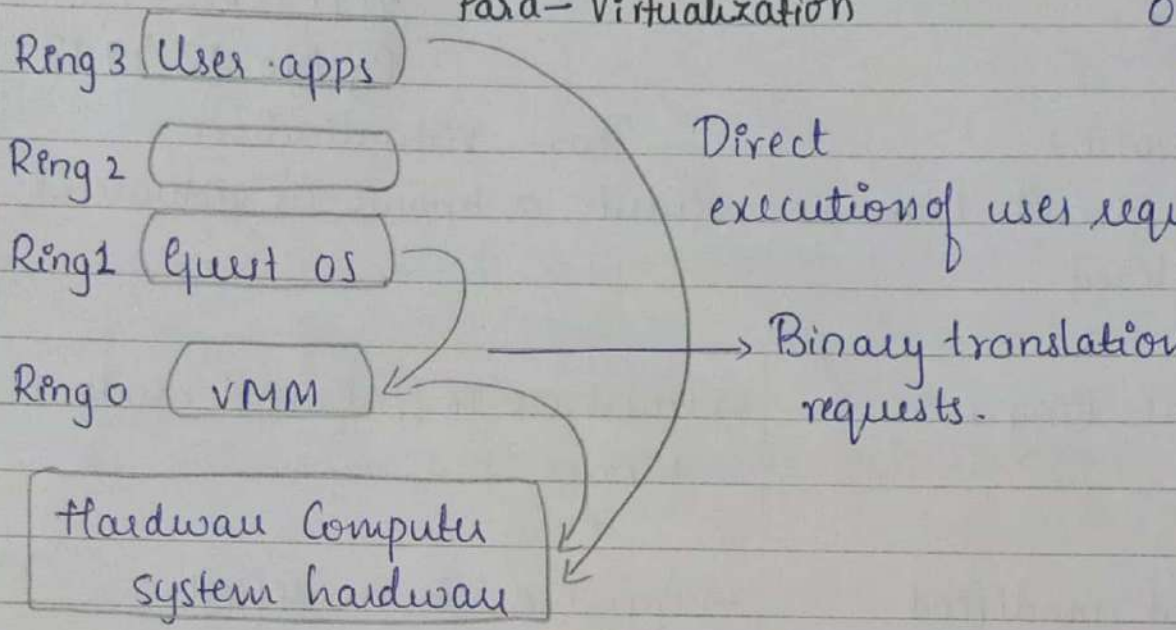
When a physical device is reclaimed for later reassignment, it may have been set to an arbitrary state that can function incorrectly or even crash the whole system. Since software-based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical. Intel VT-d supports the remapping of I/O DMA transfers and device-generated interrupts

1b)

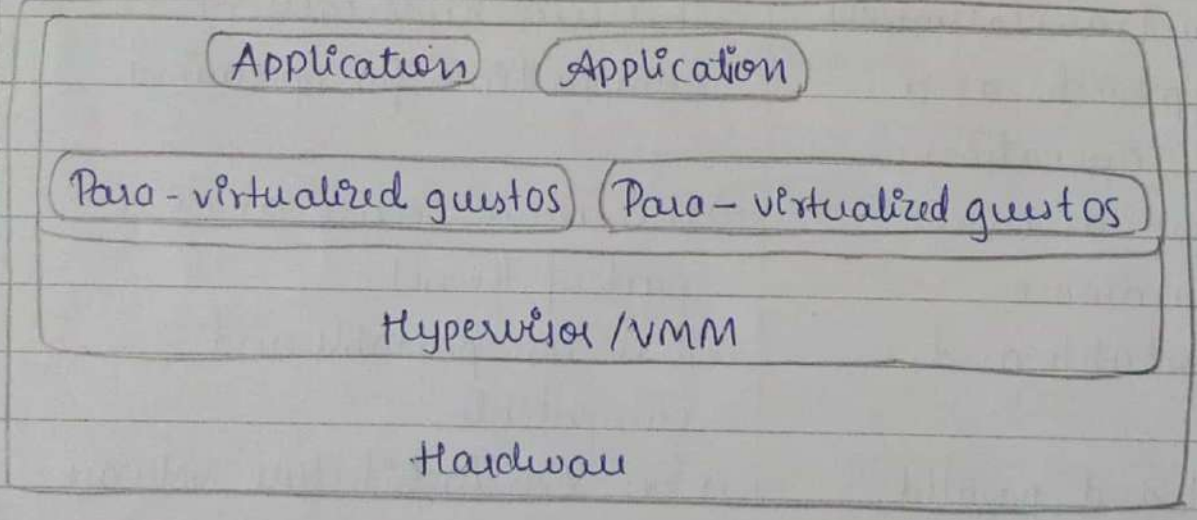
Full virtualization	Para-Virtualization
→ the guest OS doesn't know it is virtualized	→ Guest OS knows it's virtualized
→ Guest OS is at Ring 1	→ Guest OS is in kernel mode at Ring 0
→ Guest OS is not modified	→ Guest OS is modified
→ It is slower than para-virtualization	→ It is faster in operations
→ It uses binary translation (BT) and direct approach as a technique for operation	→ This uses hypercalls at compile time for operations
→ It is less secure as OS is away from hardware	→ It is more secure as OS is part of kernel
→ It is more portable and compatible	→ It is less portable and compatible
→ Ex: Microsoft and parallel System	→ Ex: Xen Architecture, VMware
→ Privileged instructions are binary translated and then executed	→ Priveledged instruction are trapped as hypercalls and converted to system calls

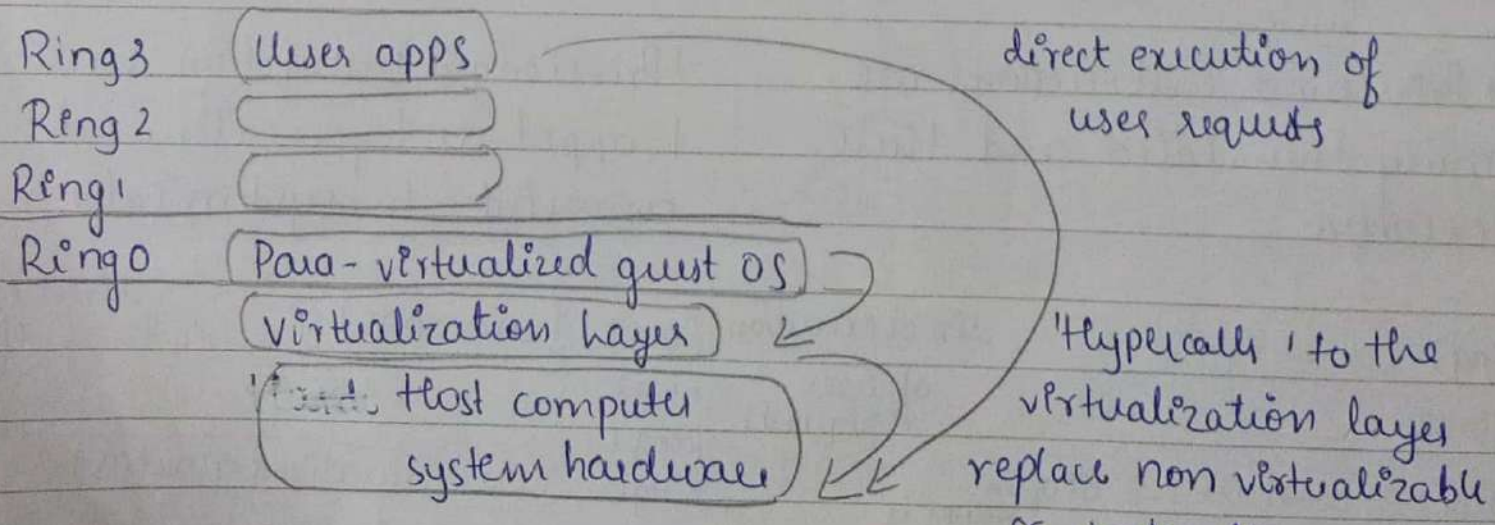
Para-Virtualization

01fe18bcs241
Pg No 1. 4



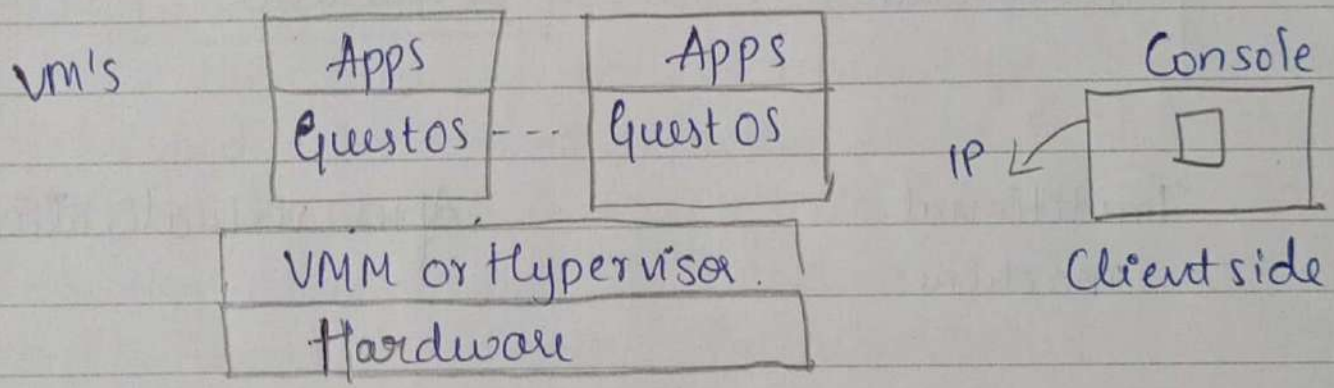
Full virtualization





(a) Virtualization is a technology that allows sharing of the physical instance of the single resource instance among multiple VM's
It leads to better resource utilization.

Type-1 or Native Hypervisor:



→ Hypervisor is a form of virtualization software used in cloud hosting to divide and allocate the resources on various pieces of hardware.
→ It is a piece of hardware used to create, delete, manage and monitor VM's.
→ In Native there is hardware level virtualization
→ Hypervisor is called as virtual machine monitor.
→ In type-1 the hypervisor runs directly on the bare hardware. So sometimes it is also called as bare metal VM.
→ This does not require any server operating System

→ It has direct access to hardware resources whereas in hosted VM or fully virtualized VM the priveleged instructions needs to undergo binary translation after getting trapped. Hence it is very slower. An in dual mode OS needs to be modified to acomodate some part of VMM into it this is not the case in Type-1 hypervisor.

→ In type-1 security risk of os isn't there but it is in hosted type

→ Hypervisor sits on the bare metal computer hardware like CPU memory etc. It manages all resources and VMs. No need for special software or OS

→ Type1 hypervisor is very efficient because they are having direct access to hardware which boosts there performance. No extra translation as in hosted or conversion from hypercalls to system calls is needed. VMM is on host OS in Type-2 or hosted. So OS doesn't know it's virtualized

→ The guest OS are layer above the hypervisor.

→ Type-2 for client also called client hypervisor.

→ This causes the empowerment of security because there is nothing any kind of third party resource that could attack hardware. whereas in Hosted VM security is less as OS is not in kernel mode and in dual mode OS is in kernel mode but is modified to certain extent.

→ Type-1 used at production level in enterprises. It can scale very significantly. It is robust. It is expensive.

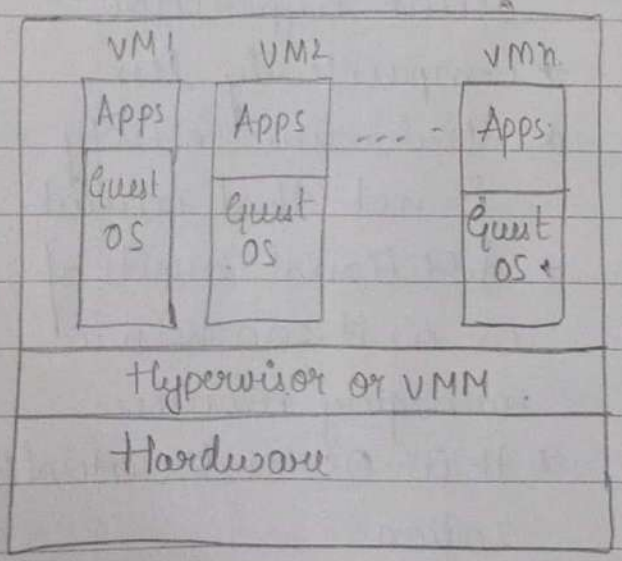
→ Type-2 or hosted has more delay as it has to go through OS So there is latency. Low cost. Software testing is done

Type-1	Type-2
* Expensive	* Cheaper compairtively
* Less latency	* More latency
* No risk of security flaws from host OS (as there is none)	* Risks of security flaws in host OS as it can compromise with all VMs
* More efficient as it has direct access to hardware and sits directly on top of it	* Less efficient as it sits on host OS and all its actions should pass through host OS.
* Used in production level in enterprises	* Used for client or SDE in company for especially testing pupose
* It is also called bare-mital VM	* It is also called client hypervisor.
* Highly scalable and robust	* Compairtively less
* Used when security is of higher concern	* Used when security is not that crucial
* No additional support.	* Additional merits of OS as it can help in managing resource
* It is HAL or Hardware Level Abstraction	* It is OS level virtualization
* No guest OS	* Guest OS must be similar to container OS
* OS images of all VM's should be in memory	* Single OS template is shared for all containers

Type-1

* Better scalability
* Simple, as long as we have necessary hardware support
* Faster
* Higher performance

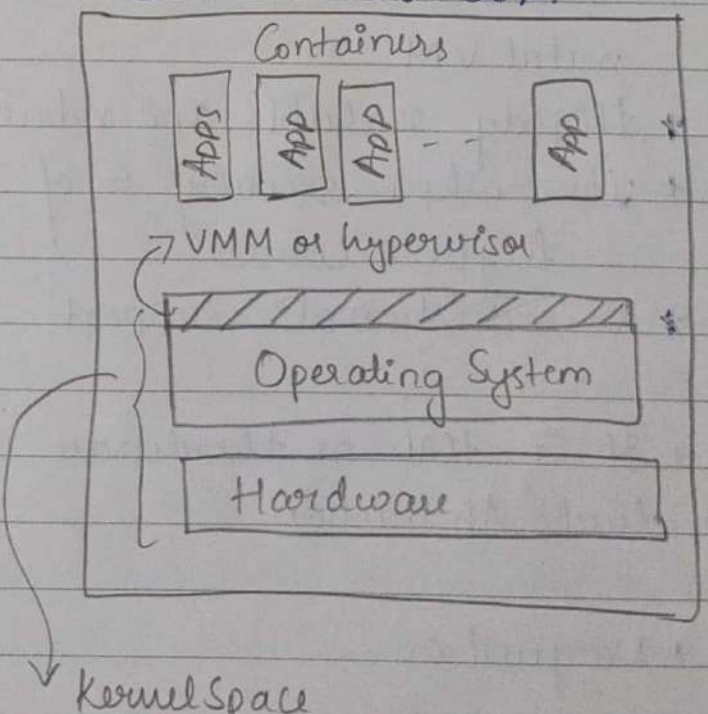
Eg: VMware ESXI
Microsoft Hyper-V



* More startup time

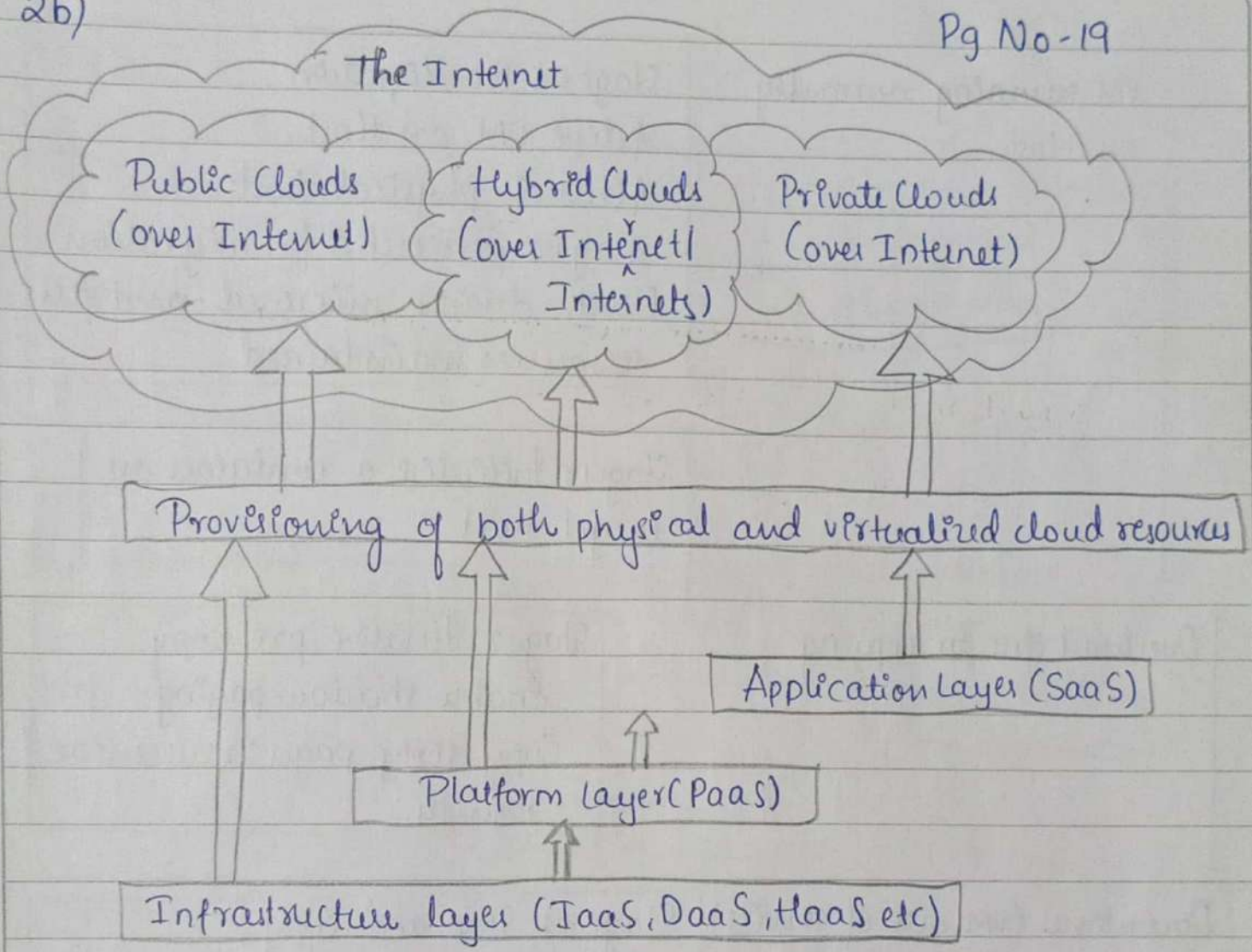
Type-2

* Not so much, because of its reliance on the underlying OS
* Simpler to setup, as already there is an OS
* Slower because of system's dependency
* Comparitively lower
* Eg: VMware Workstation Player
Sun's Virtual box.



* less startup time.

2b)



It is built in 3 layers as shown. The three layers are also implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The public, private and hybrid clouds are conveyed to users via Internet. The infrastructure layer is built with virtualized compute, storage and network resources. The abstraction of these hardware resources is meant to provide flexibility demanded by users. Internally, virtualization realizes automated provisioning of resources and optimization of the infrastructure management process. The virtualized platform layer serves as "system middleware" between infrastructure and application layer.

The application layer is formed with collection of all needed software modules for SaaS applications. Service applications in this layer include daily office management work such as information retrieval, document processing. The application layer is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transaction and supply chain management. Many applications may apply resources at mixed layer. These layers have dependency relation

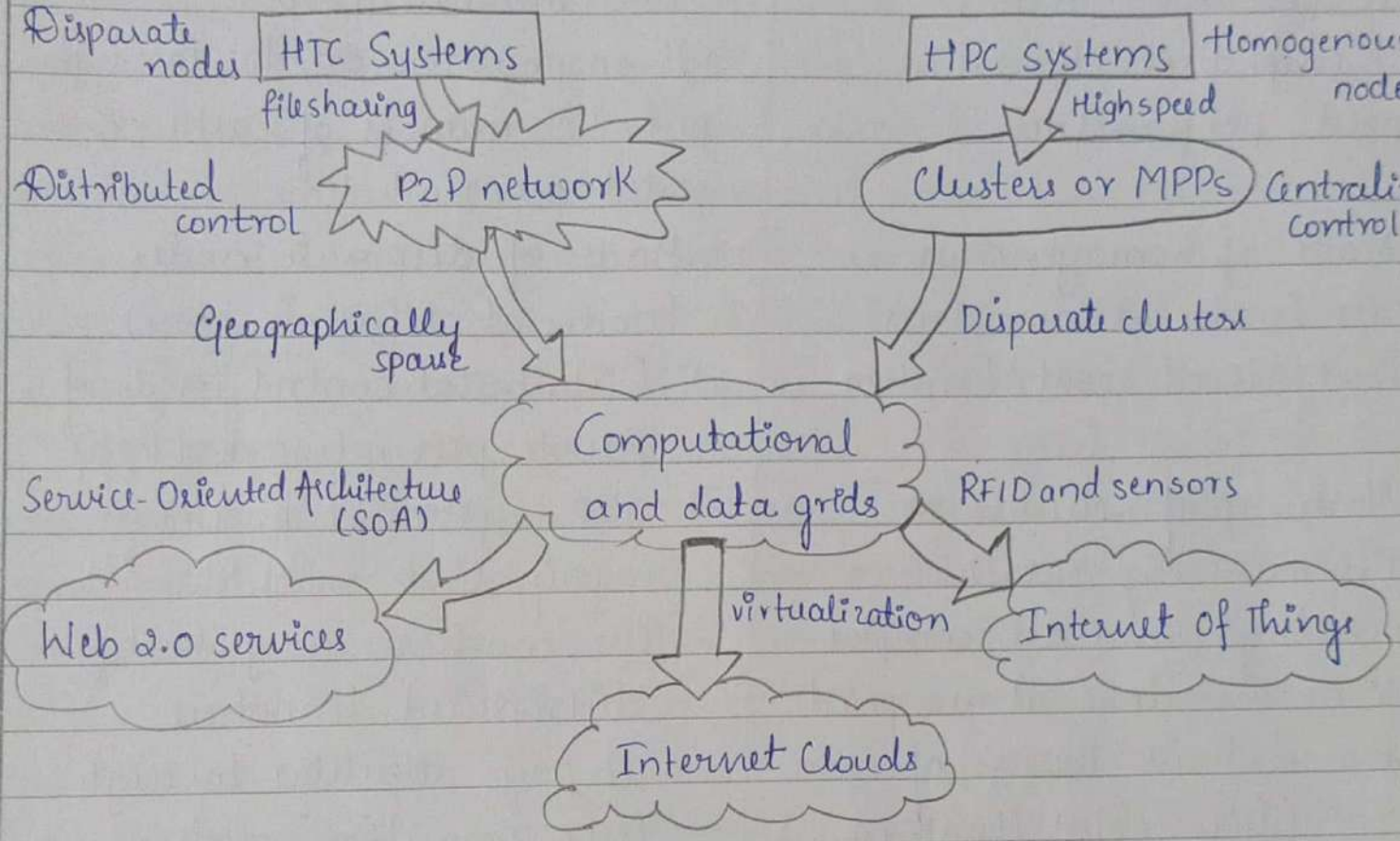
Design Challenges:-

i) Services at various layers demand different amounts of fundionality support and resource management by provider

ii) Integration of 3 layer for interdependencies between them

iii) SaaS demands most work from provider, PaaS in middle, IaaS demands least.

iv) Secuit is always a challenge. CSP should look into both insiders and outsiders threats.



In 1950's to 1970's there were mainframes that were built to satisfy large business and government organization needs.
From 1970's to 1990's use of personal computers increased with VLSI processors and integrated chips. The hardware costs started decreasing.
Then in later 1995-2000 Emerged WWW and Internet.
Today we exchange huge data between the systems via Internet.

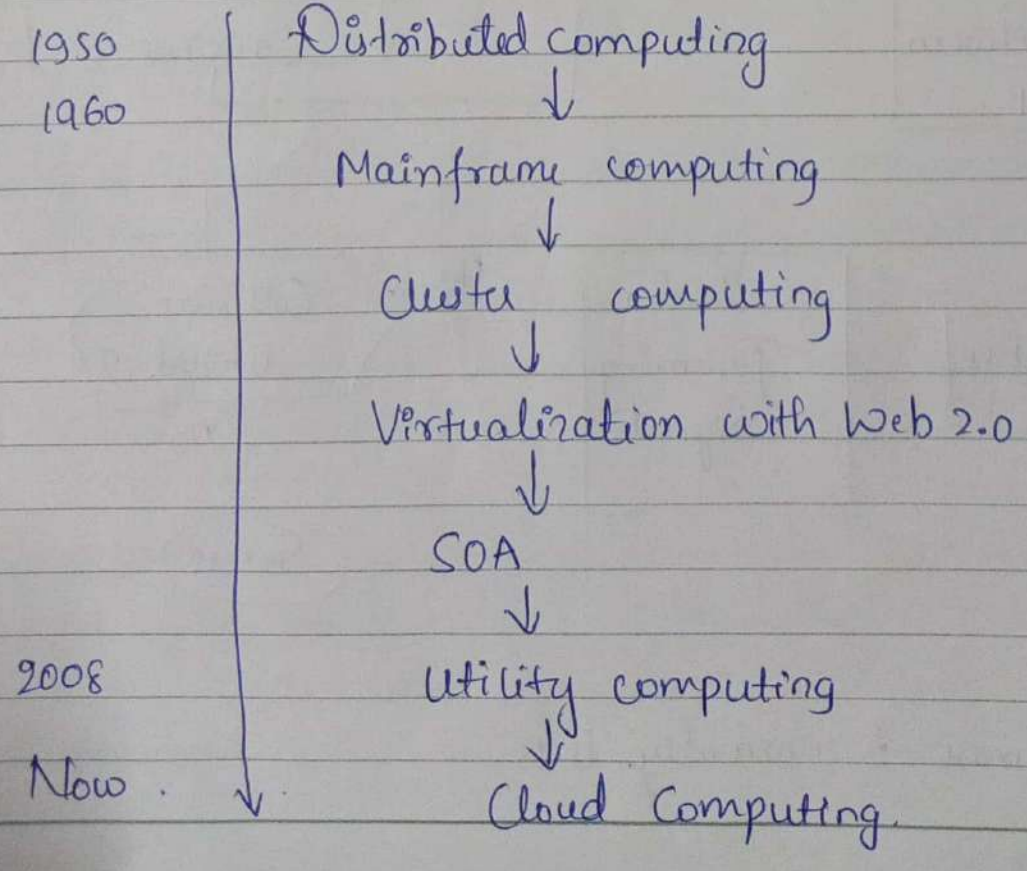
Earlier HPC and HTC systems were used.
HPC stands for high performance computing
HTC stands for high throughput computing.

HPC	HTC
→ It is used for scientific and research purpose	→ It is mainly used for market and business purpose.
→ It emphasize the raw speed performance	→ It emphasizes on the throughput i.e number of tasks completed per unit time.
→ Made of homogeneous nodes (nodes at same place)	→ Made of disparate nodes. (nodes at different place)
→ Centralized control system	→ Distributed control mode of Peer-to-peer networks (P2P)
→ Made up of clusters or MPPs (massively parallel processing) and all computations are done at one point	→ P2P system is built over many client machines. Peer machines are globally distributed in nature
→ Measured in terms of GFLOP (Gigo point floating point operations)	→ It pays attention to high flux computing such as Internet searches and web Services.

With SOA, Web 2.0 services are available. Advances in virtualization led to growth of Internet Clouds that proliferated a new computing paradigm. The maturity of radio-frequency identification (RFID), Global Positioning System (GPS) and sensors triggered to the development of Internet of things (IoT)

Grid computing is envisioned to allow close interaction among applications running on distant computers simultaneously (i.e. distributed computing). Internet services such as Telnet command enables a local computer to connect to remote computer.

Computing Grids offer an infrastructure that couples computers, software / middlewares, special instruments, and people and sensors together. The grid is often constructed across LAN, WAN or Internet backbone networks at a regional, national, or global scale. Enterprises or organizations present grids as integrated computing resources. The computers used in grids are primarily workstations, servers, clusters and supercomputers. Personal computers, PDAs and laptops can be used to access the devices in grid system



Some of the important technologies like:-

i) utility computing
ii) Service orientation
iii) Peripherals
iv) Virtualization
v) Web 2.0
vi) HPC and HTC
vii) Grids

} played very important role in the evolution of cloud computing.

Virtualization where a single instance of physical resource can be logically divided to provide better resource utilization is a key technique. It deals how pool of resources can be efficiently shared with different users. Starting from 1950 the evolution from distributed computing started.

~~1950~~

Cloud computing is enabled by convergence of technolgies in four areas (i) hardware virtualization and multi-core chips (ii) utility and grid computing (iii) SOA (Service Oriented Architecture), Web 2.0 and WS mashups (iv) Atonomic computing and data center automation.

i) Utility and grid computing lay the necessary foundation for cloud computing. Recent advances in SOA, Web 2.0 and mashups are pushing cloud one step forward