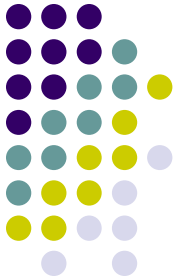
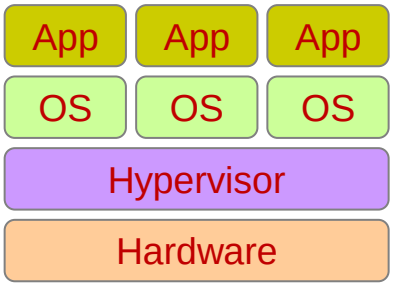


MODULE 2
VIRTUALIZATION

Virtualization

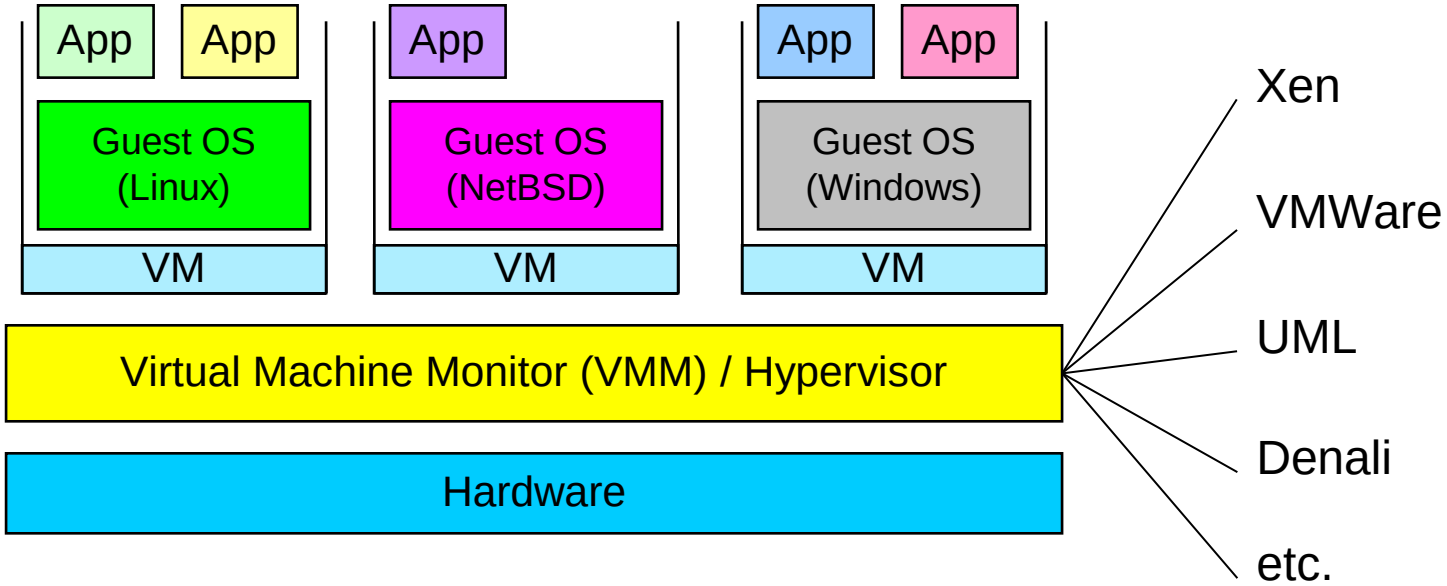
- Virtual workspaces:
  - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. O/S, provided services).
- Implement on Virtual Machines (VMs):
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
  - VMWare, Xen, etc.
- Provide infrastructure API:
  - Plug-ins to hardware/support structures



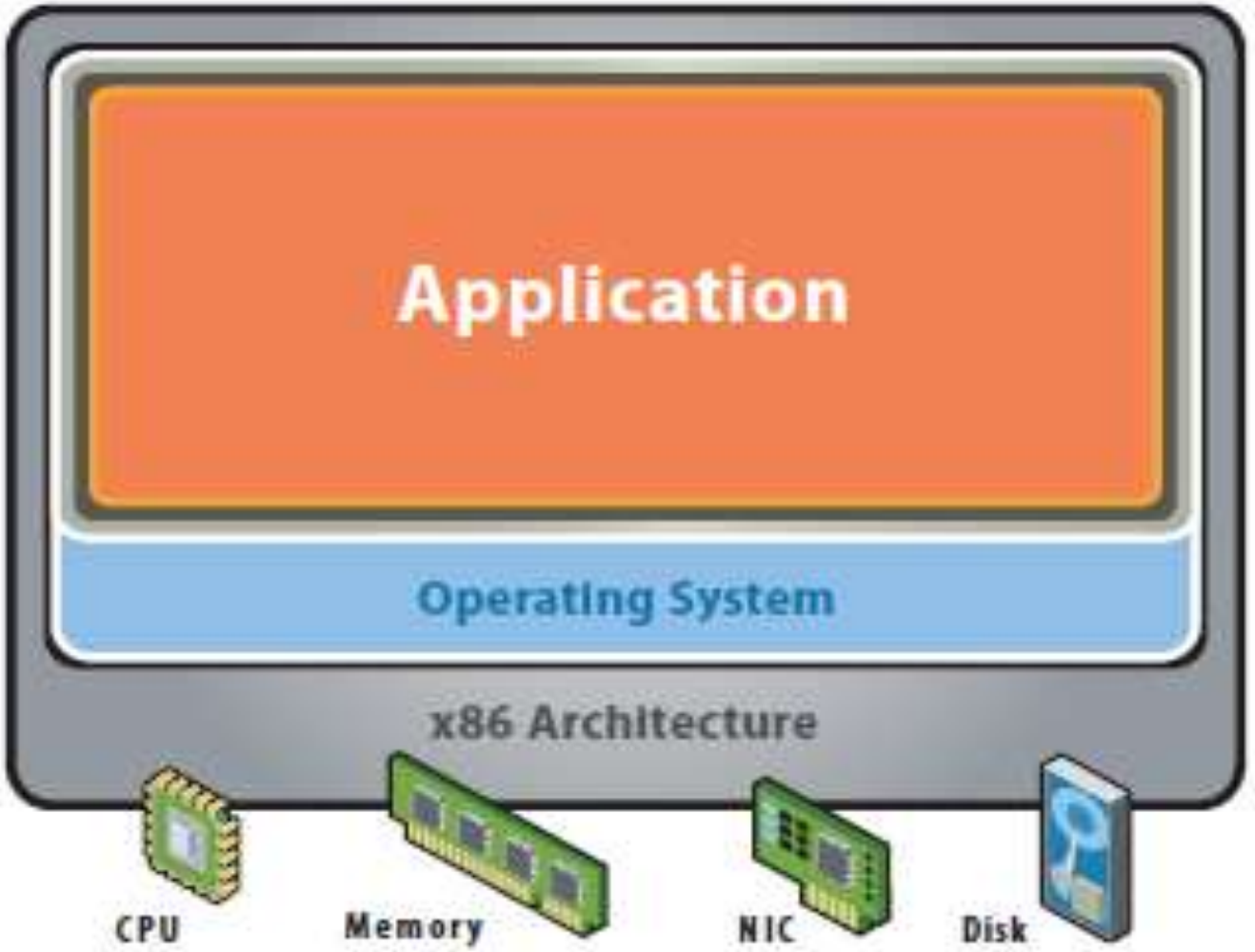
Virtualized Stack

Virtual Machines

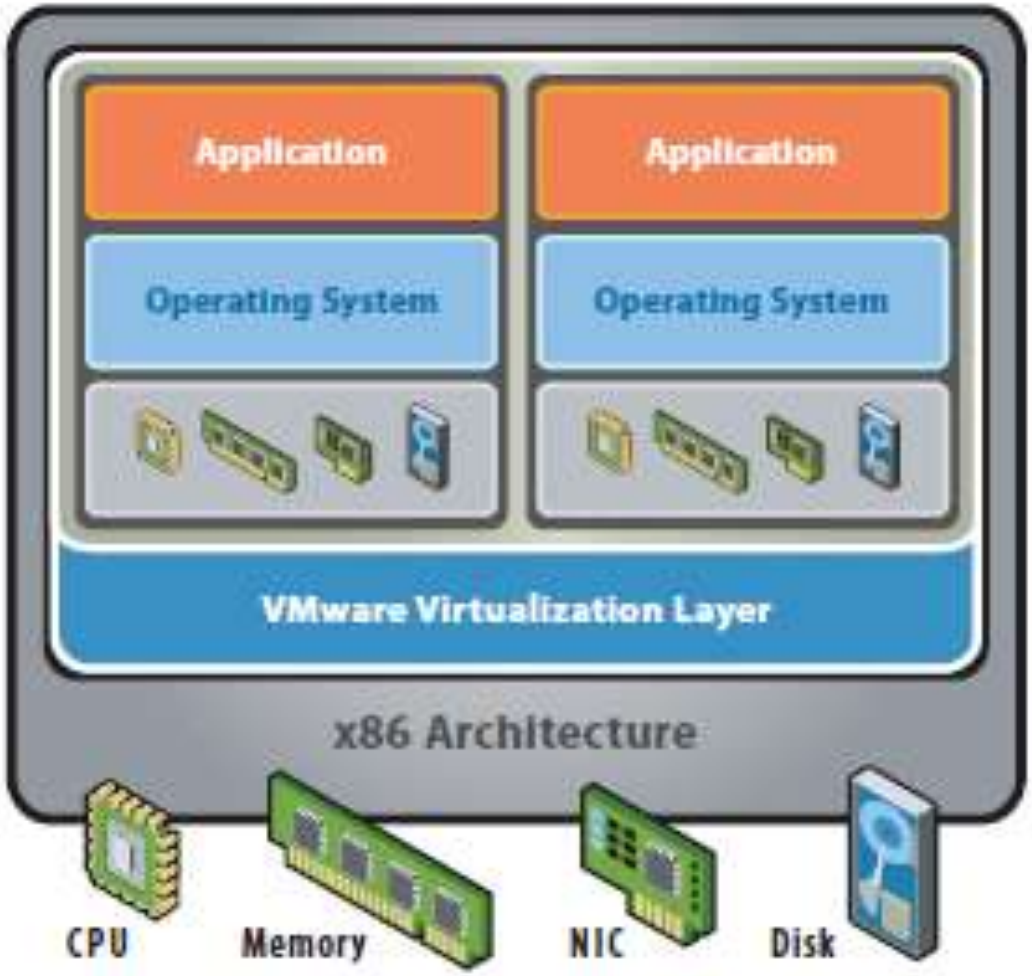
- VM technology allows multiple virtual machines to run on a single physical machine.



Traditional App/Server



Virtual Server Model



Why Use Virtual Machines?

Physical machine

Difficult to move or copy

Bound to a specific set of hardware components

Requires personal contact to upgrade hardware

Virtual machine

Easy to move and copy:

- Encapsulated into files
- Independent of physical hardware

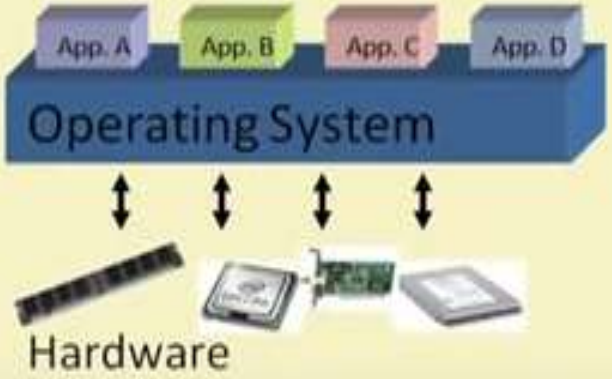
Easy to manage:

- Isolated from other virtual machines
- Insulated from hardware changes

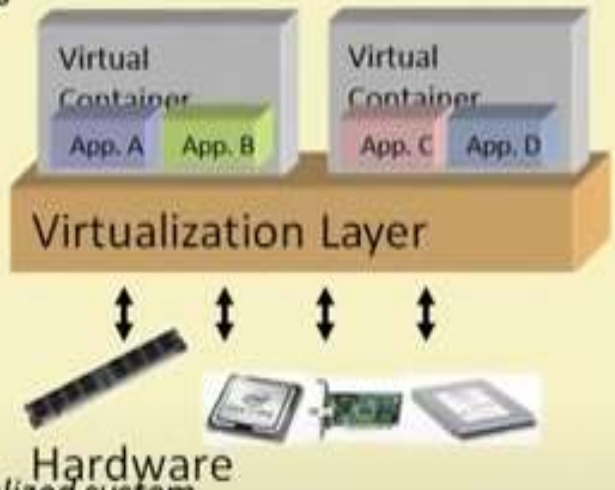
Provides the ability to support legacy applications

Virtualization

- Virtualization is a broad term (virtual memory, storage, network, etc)
- Focus: **Platform virtualization**
- Virtualization basically allows one computer to do the job of multiple computers, by sharing the resources of a single hardware across multiple environments



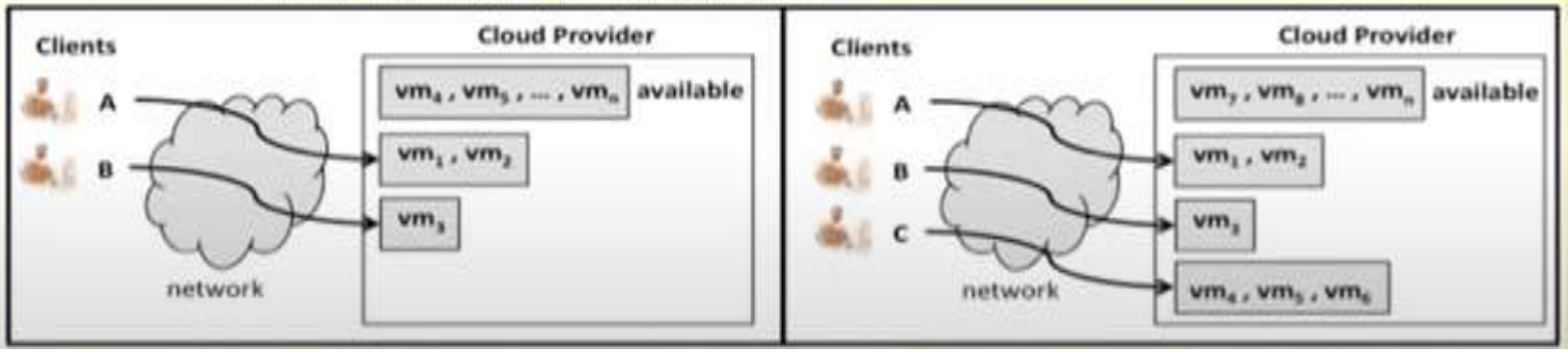
'Non-virtualized' system
A single OS controls all hardware platform resources



Virtualized system
It makes it possible to run multiple Virtual Containers on a single physical platform

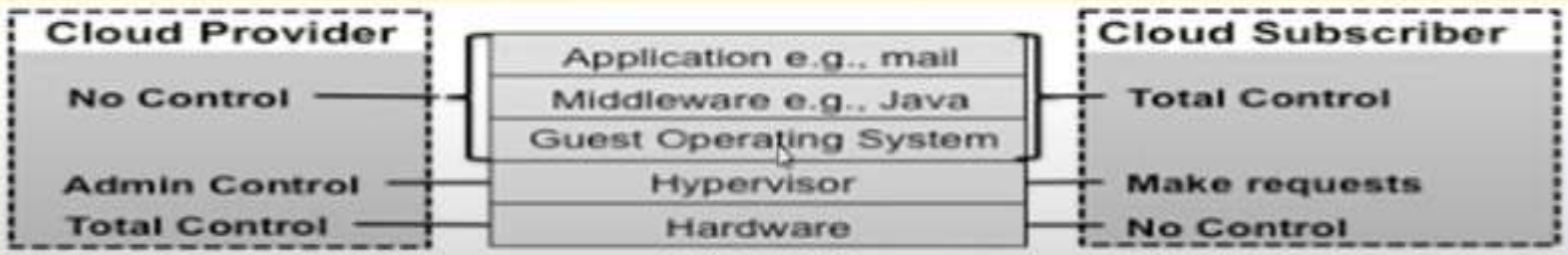
IaaS Provider/Subscriber Interaction Dynamics

- Provider has a number of available virtual machines (VMs) that it can allocate to clients.
  - Client A has access to vm1 and vm2, Client B has access to vm3 and Client C has access to vm4, vm5 and vm6
  - Provider retains only vm7 through vmN



IaaS Component Stack and Scope of Control

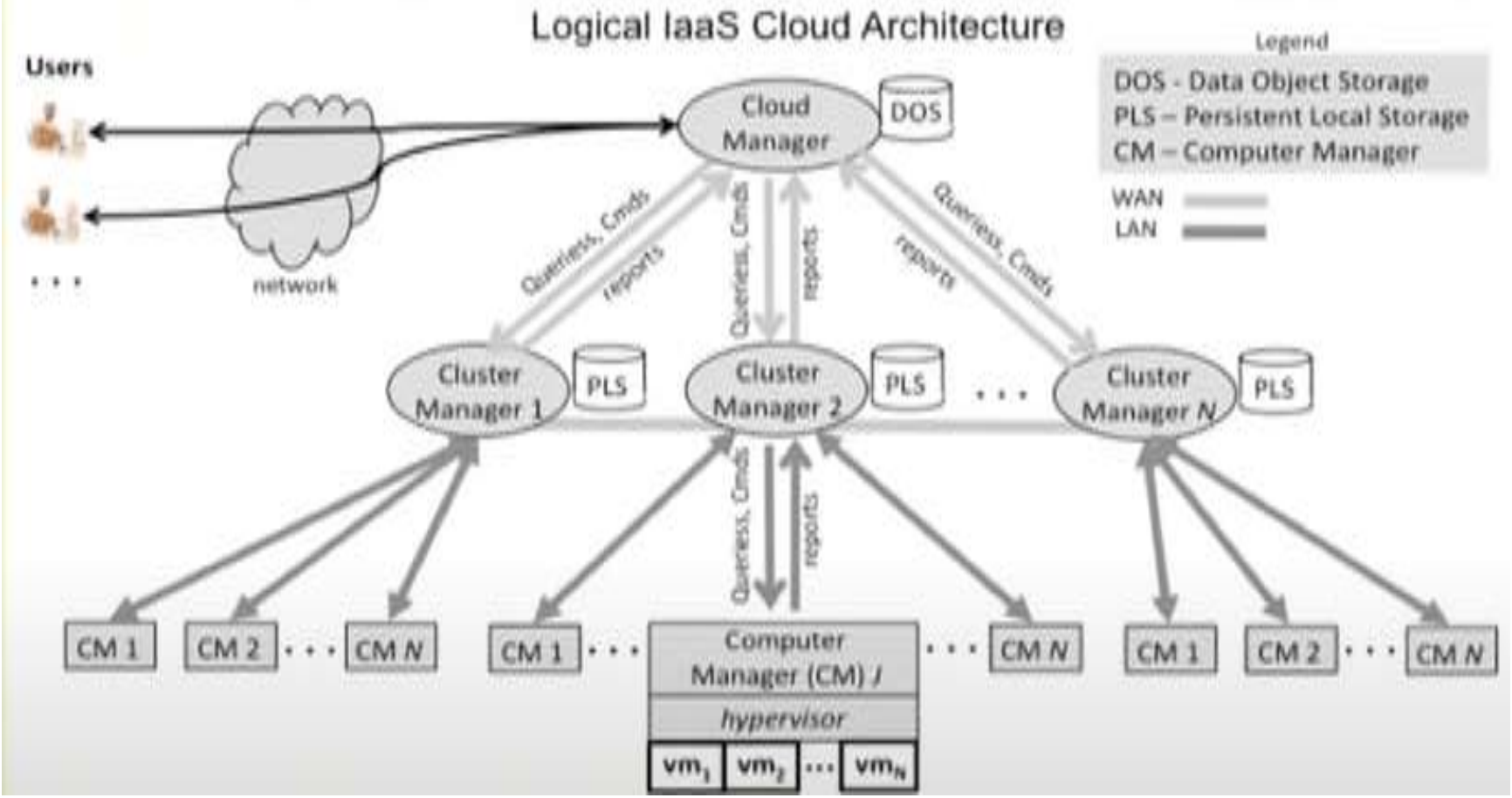
- IaaS component stack comprises of hardware, operating system, middleware, and applications layers.
- Operating system layer is split into two layers.
  - Lower (and more privileged) layer is occupied by the Virtual Machine Monitor (VMM), which is also called the Hypervisor
  - Higher layer is occupied by an operating system running within a VM called a guest operating system

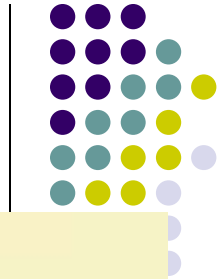


- In IaaS Cloud provider maintains total control over the physical hardware and administrative control over the hypervisor layer
- Subscriber controls the Guest OS, Middleware and Applications layers.
- Subscriber is free (using the provider's utilities) to load any supported operating system software desired into the VM.
- Subscriber typically maintains complete control over the operation of the guest operating system in each VM.

IaaS Cloud Architecture

- Logical view of IaaS cloud structure and operation





IaaS Cloud Architecture

- Three-level hierarchy of components in IaaS cloud systems
  - *Top level* is responsible for *central control*
  - *Middle level* is responsible for *management of possibly large computer clusters* that may be *geographically distant* from one another
  - *Bottom level* is responsible for *running the host computer systems* on which virtual machines are created.
- Subscriber queries and commands generally flow into the system at the top and are forwarded down through the layers that either answer the queries or execute the commands

Operation of the Cloud Manager

- Cloud Manager is the public access point to the cloud where subscribers sign up for accounts, manage the resources they rent from the cloud, and access data stored in the cloud.
- Cloud Manager has mechanism for:
  - Authenticating subscribers
  - Generating or validating access credentials that subscriber uses when communicating with VMs.
  - Top-level resource management.
- For a subscriber's request cloud manager determines if the cloud has enough free resources to satisfy the request

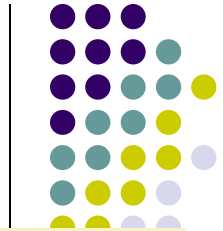
Data Object Storage (DOS)

- DOS generally stores the subscriber's metadata like user credentials, operating system images.
- DOS service is (usually) single for a cloud.

Operation of the Cluster Managers

- Each *Cluster Manager* is responsible for the operation of a collection of computers that are connected via high speed local area networks
- *Cluster Manager* receives resource allocation commands and queries from the *Cloud Manager*, and calculates whether part or all of a command can be satisfied using the resources of the computers in the cluster.
- *Cluster Manager* queries the *Computer Managers* for the computers in the cluster to determine resource availability, and returns messages to the *Cloud Manager*

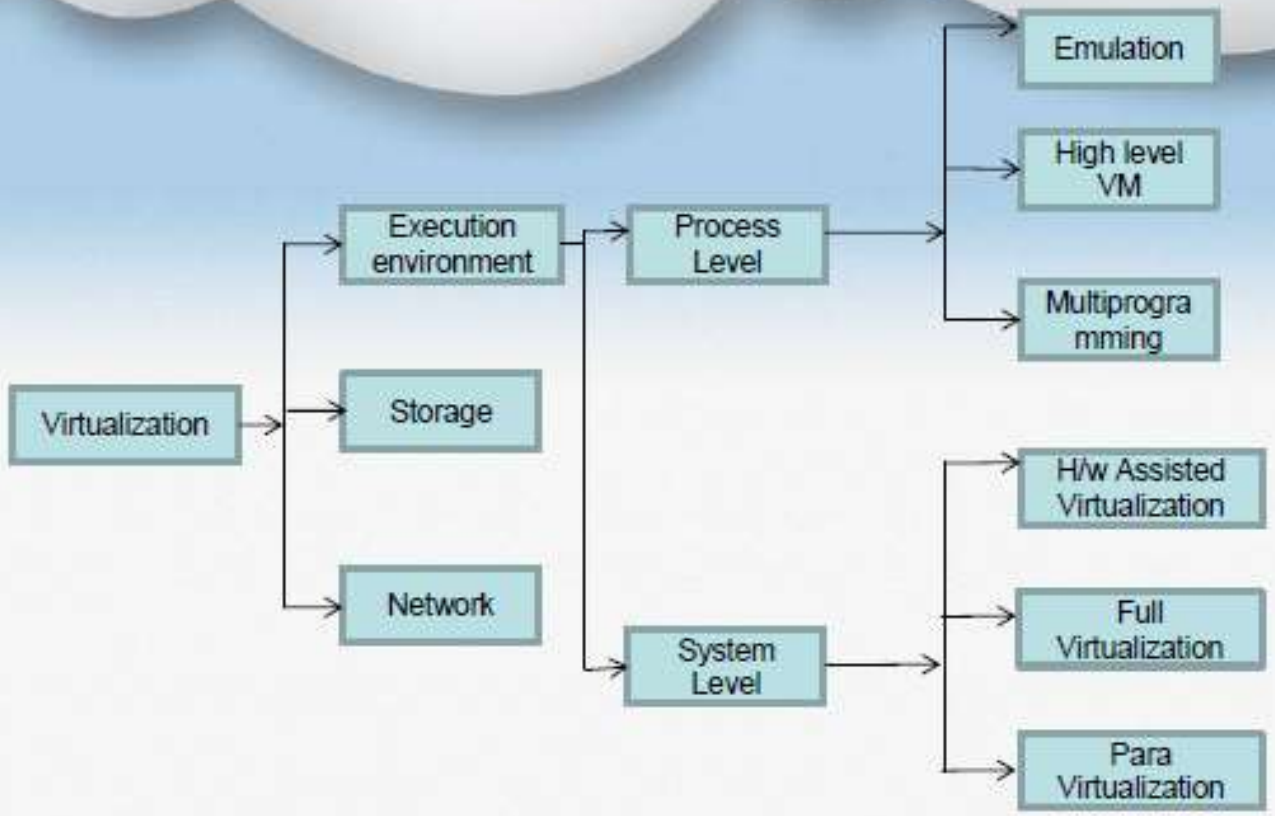
- Directed by the Cloud Manager, a Cluster Manager then instructs the Computer Managers to perform resource allocation, and reconfigures the virtual network infrastructure to give the subscriber uniform access.
- Each Cluster Manager is connected to Persistent Local Storage (PLS)
- PLS provide persistent disk-like storage to Virtual Machine



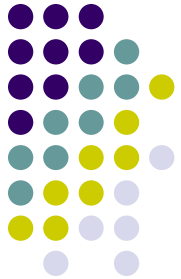
Operation of the Computer Managers

- At the lowest level in the hierarchy computer manger runs on each computer system and uses the concept of virtualization to provide Virtual Machines to subscribers
- Computer Manger maintains status information including how many virtual machines are running and how many can still be started
- Computer Manager uses the command interface of its hypervisor to start, stop, suspend, and reconfigure virtual machines

Classification of different Types of Virtualization



Hardware Virtualization



- Full virtualization – almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified.

  e.g. VMWare ESXi and Microsoft virtual server

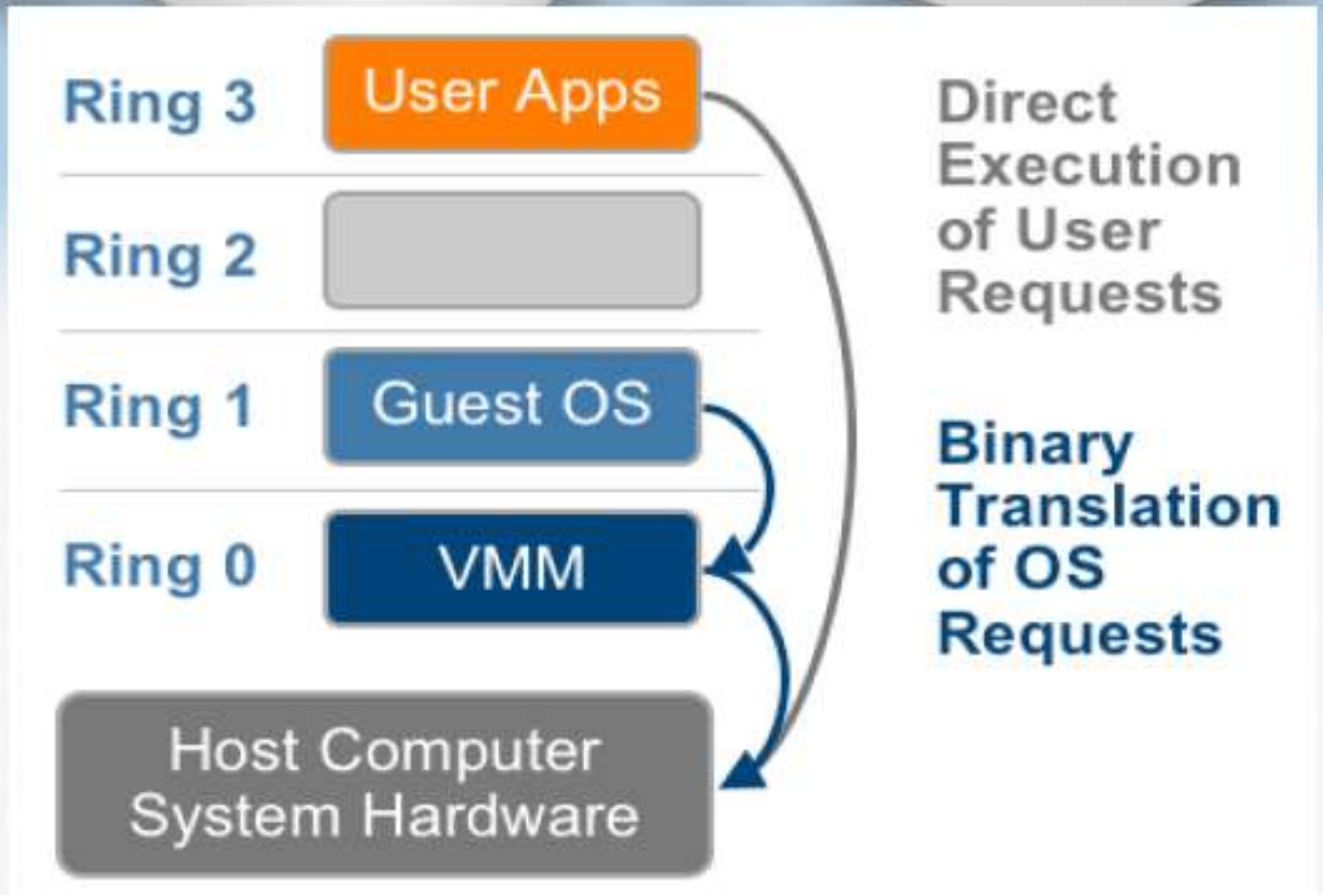
- Partial virtualization – some but not all of the target environment attributes are simulated. As a result, some guest programs may need modifications to run in such virtual environments.

  e.g. Address space virtualization used in time sharing systems

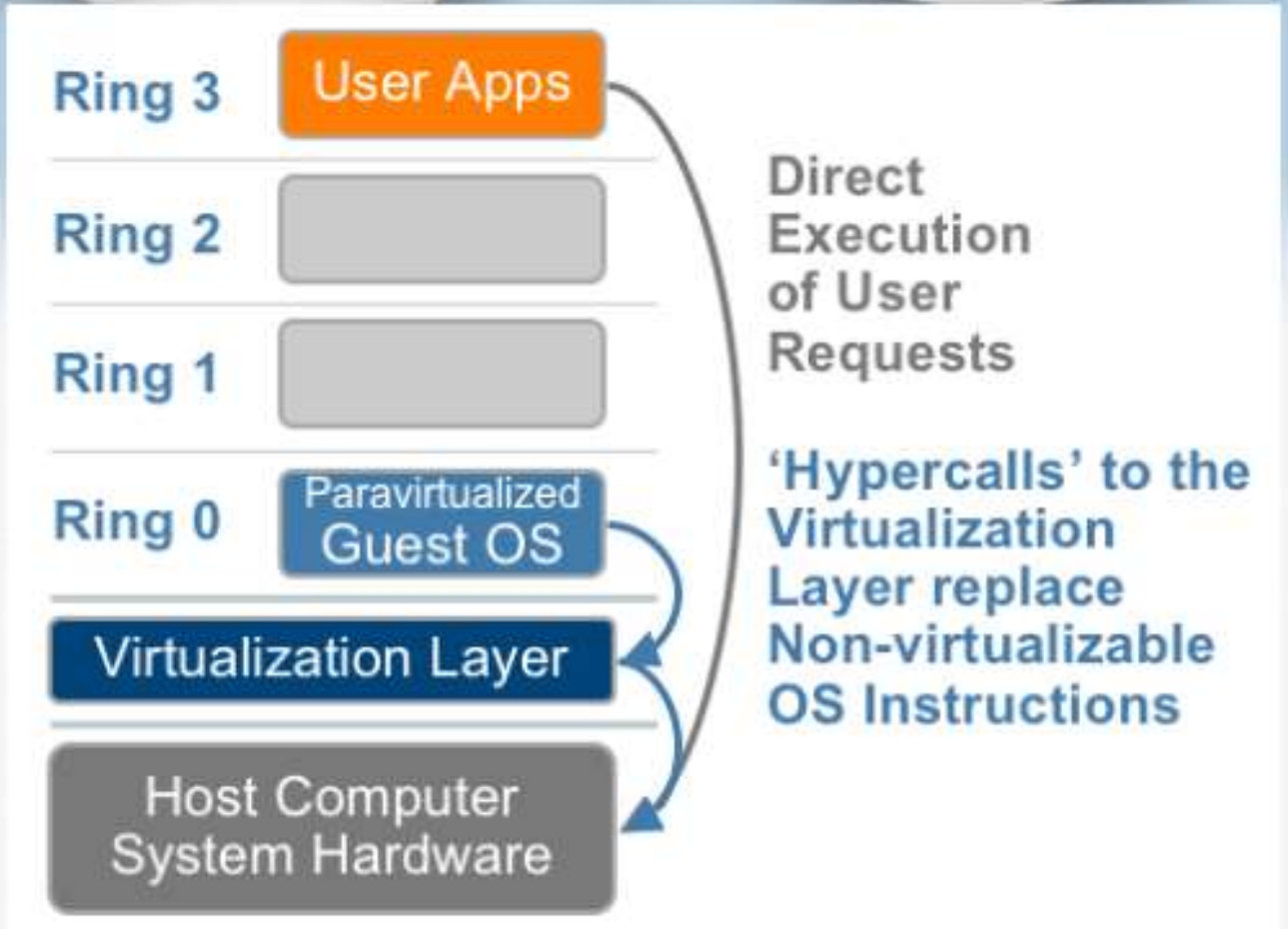
- Paravirtualization – a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system. Guest programs need to be specifically modified to run in this environment.

Full Virtualization	Para virtualization
In Full virtualization, virtual machines permit the execution of the instructions with the running of unmodified OS in an entirely isolated way	In Para virtualization, a virtual machine does not implement full isolation of OS but rather provide a different API which is utilized when OS is subjected to alteration
F. V. is less secure	P.V. is more secure
F. V. uses binary translation and a direct approach as a technique for operations	While P.V. uses hypercalls at compile time for operation
It is slow	Comparatively fast
More portable and compatible	Less portable and compatible
e.g. Microsoft	e.g. Microsoft Hyper-V, Xen etc.
It supports all guest operating systems without modification	Guest OS has to be modified and only a few OS supports it

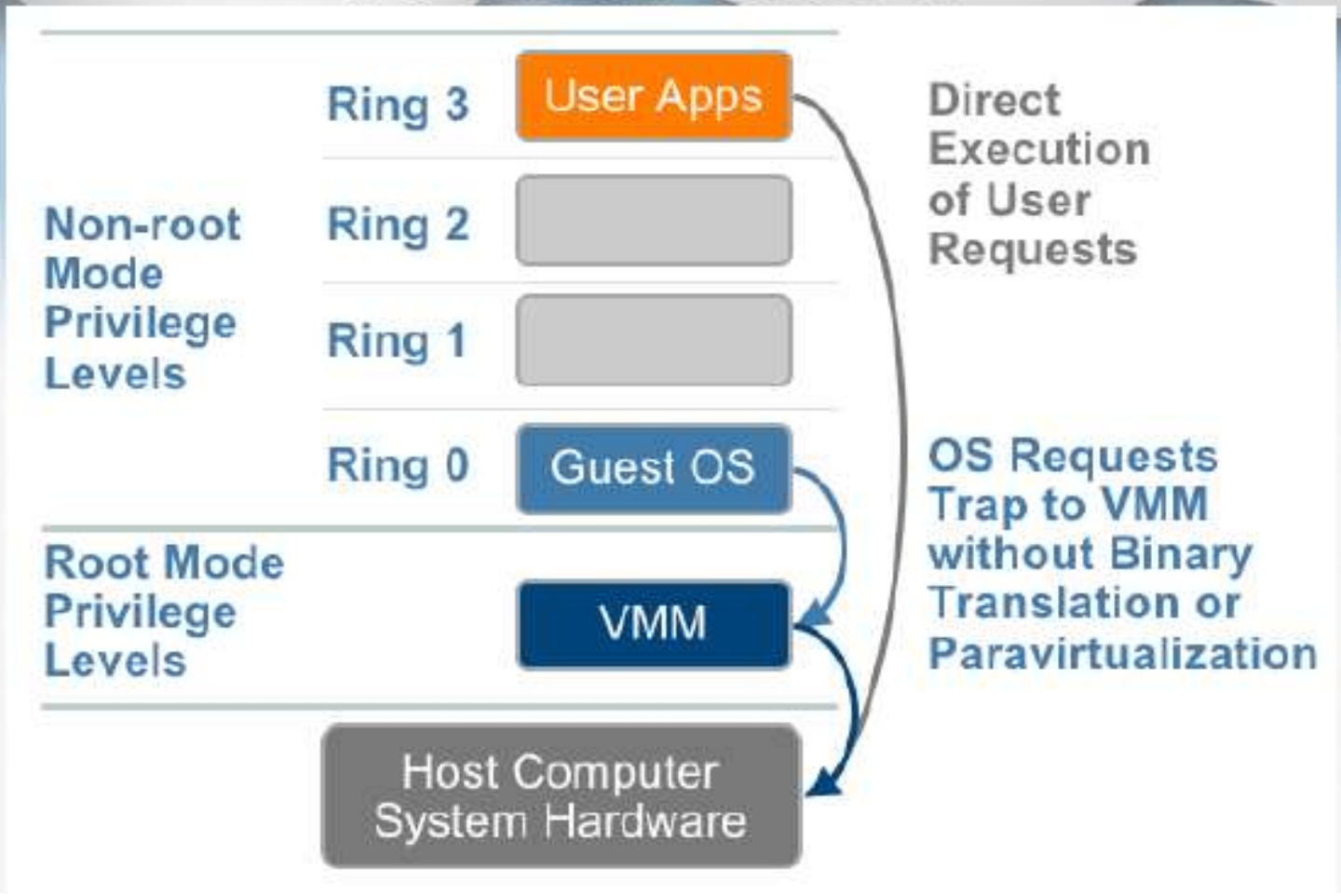
Full Virtualization Using Binary Translation



OS assisted Virtualization or Paravirtualization



Hardware Assisted Virtualization



Software Virtualization

- Operating system-level virtualization, hosting of multiple virtualized environments within a single OS instance.
- Application virtualization and workspace virtualization, the hosting of individual applications in an environment separated from the underlying OS. Application virtualization is closely associated with the concept of portable applications.
- Service virtualization, emulating the behavior of dependent (e.g., third-party, evolving, or not implemented) system components. It virtualizes only specific slices of dependent behavior critical to the execution of development and testing tasks.

Storage Virtualization

- Storage virtualization, the process of completely abstracting logical storage from physical storage
- Distributed file system, any file system that allows access to files from multiple hosts sharing via a computer network
- Virtual file system, an abstraction layer on top of a more concrete file system, allowing client applications to access different types of concrete file systems in a uniform way

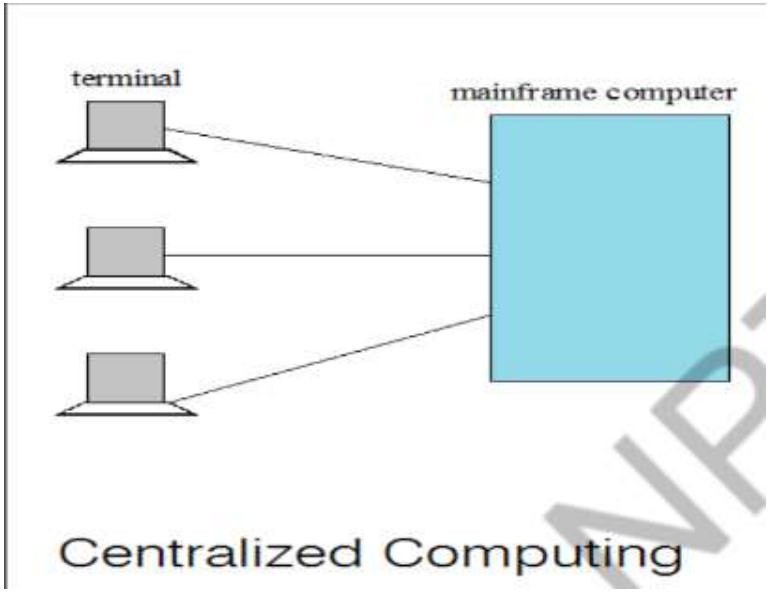
Storage Virtualization

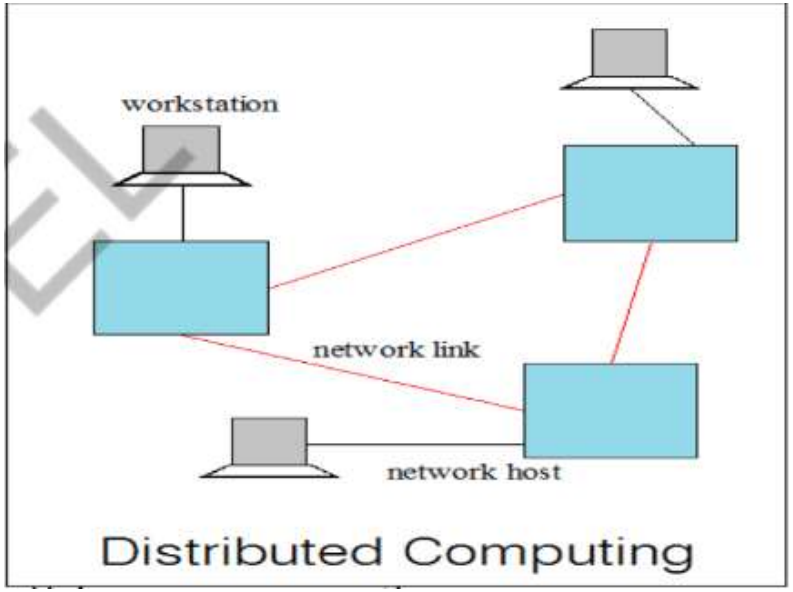
- Storage hypervisor, the software that manages storage virtualization and combines physical storage resources into one or more flexible pools of logical storage
- Virtual disk drive, a computer program that emulates a disk drive such as a hard disk drive or optical disk drive

Network Virtualization

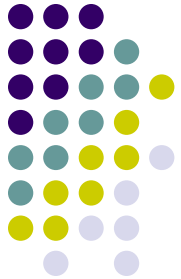
- Network virtualization, creation of a virtualized network addressing space within or across network subnets
- Virtual private network (VPN), a network protocol that replaces the actual wire or other physical media in a network with an abstract layer, allowing a network to be created over the Internet

Centralized Vs Distributed Computing





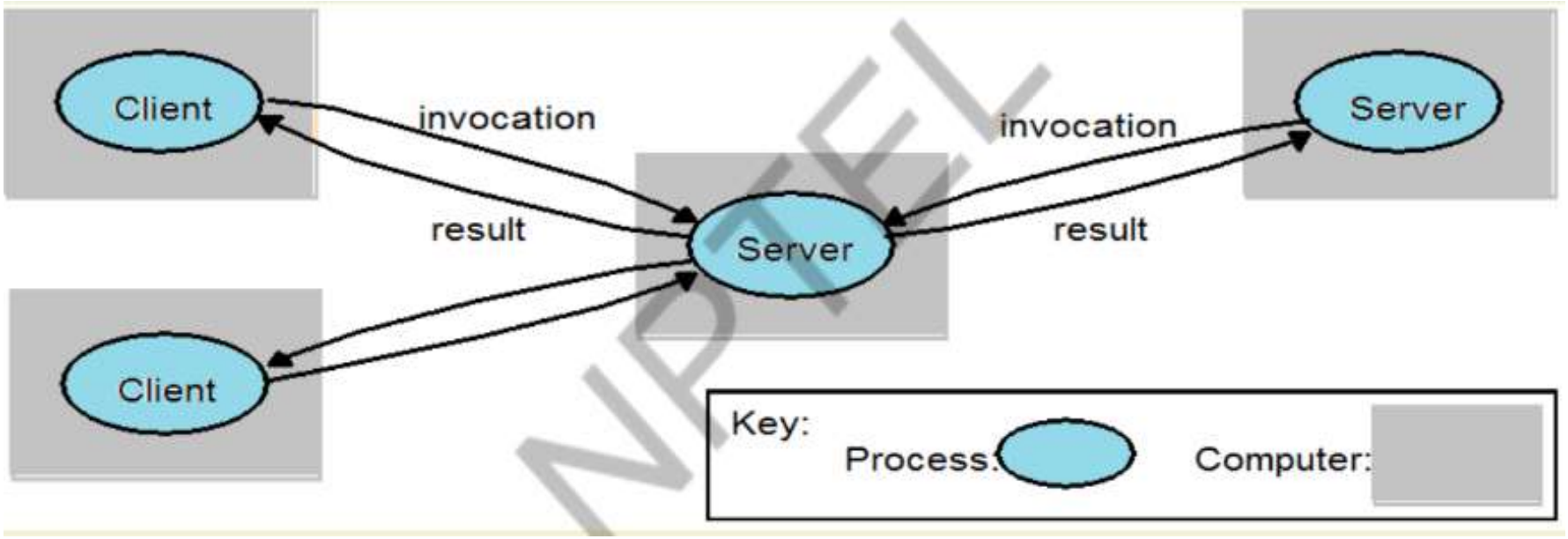
Distributed computing



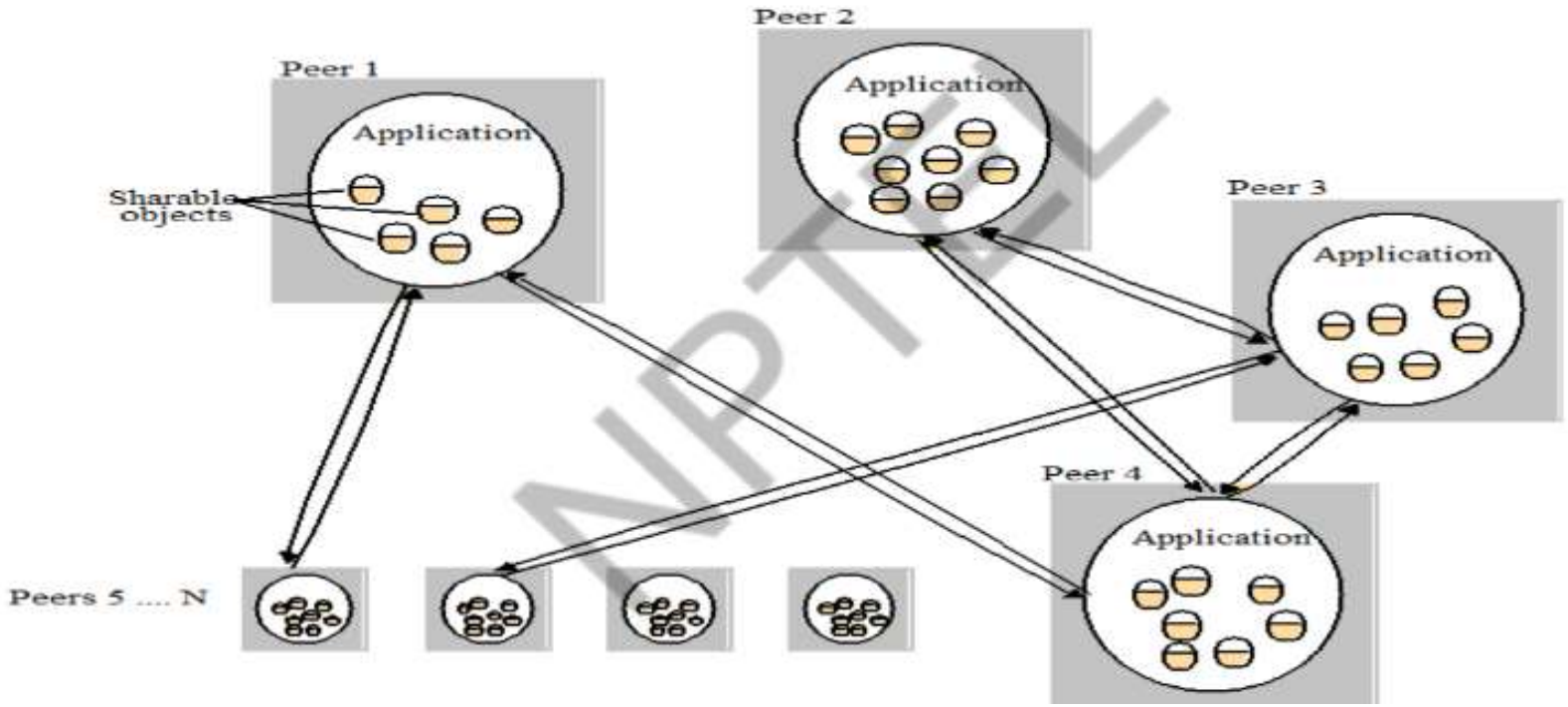
- Controlling and managing is distributed
- Uni processor is not a concept
- Distributed Processor carries its own local memory and storage
- It provides computing services in distributed manner
- Each processor communicates with another processor through high speed lines
- Components: Workstations, Servers, Personal assistant devices
- Examples: Internet, ATM machines, Intranet

Distributed applications

1. Clients invoke individual servers



- 2. A typical distributed application based on peer processes



Grid Computing

- Grid computing harnesses unused processing cycles of all computers in a

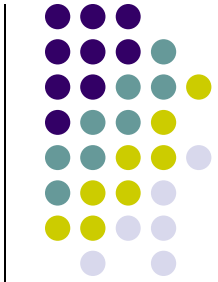
Electrical Power Grid

- Users (or electrical appliances) get access to electricity through wall sockets with no care or consideration for where or how the electricity is actually generated.
- **"The power grid"** links together power plants of many different kinds

Grid

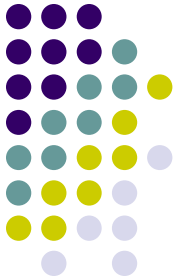
- Users (or client applications) gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on are
- **"The Grid"** links together computing resources (PCs, workstations, servers, storage elements) and provides the mechanism needed to access them.

Application of Grid computing



- Today's Science/Research is based on computations, data analysis, data visualization & collaborations
- Computer Simulations & Modelling are more cost effective than experimental methods
- Scientific and Engineering problems are becoming more complex & users need more accurate, precise solutions to their problems in shortest possible time
- Data Visualization is becoming very important
- Exploiting under utilized resources

Cluster Computing



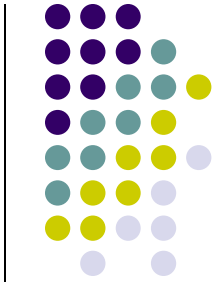
- A cluster is a type of parallel or distributed computer system, which consists of a collection of inter-connected stand-alone computers working together as a single integrated computing resource
- Key components of a cluster include multiple standalone computers (PCs, Workstations, or SMPs), operating systems, high-performance interconnects, middleware, parallel programming environments, and applications.
- Clusters are usually deployed to improve speed and/or reliability over that provided by a single computer
- Basic building blocks of clusters are broken down into multiple categories:

• Cluster Nodes

• Cluster Network

• Network Characterization

Utility Computing



- Utility Computing is purely a concept which cloud computing practically implements
- Utility computing is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate.
- Some highlights:

a) Pay-for-use Pricing Business Model

b) Data Center Virtualization and Provisioning

c) Solves Resource Utilization Problem

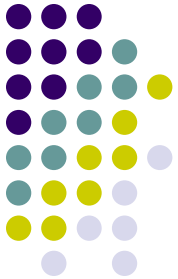
d) Outsourcing

e) Web Services Delivery

f) Automation

- **Drawbacks: Data Backup , Data Security, Partner Competency, Defining SLA**

Advantages



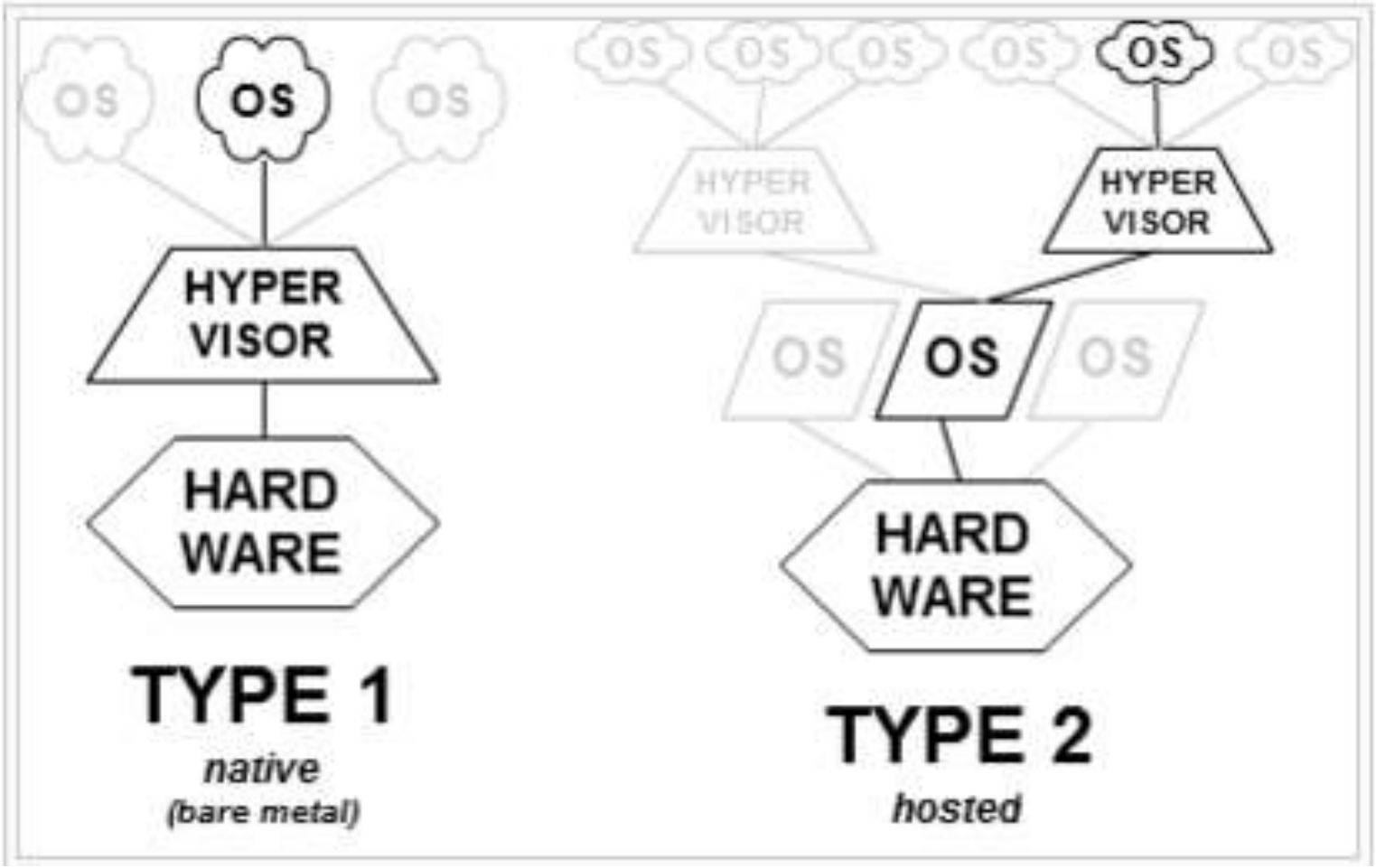
- Lower Computer cost
- Improved software updates
- Improved document format capabilities
- Unlimited storage capacity
- Increased data reliability
- Universal information access
- Latest version availability
- Easier group collaboration
- Device independance

Drawback

- Requires constant internet connection
- Does not work well with low speed
- Features might be limited
- Can be slow
- Security
- Stored data can be lost

Hypervisor

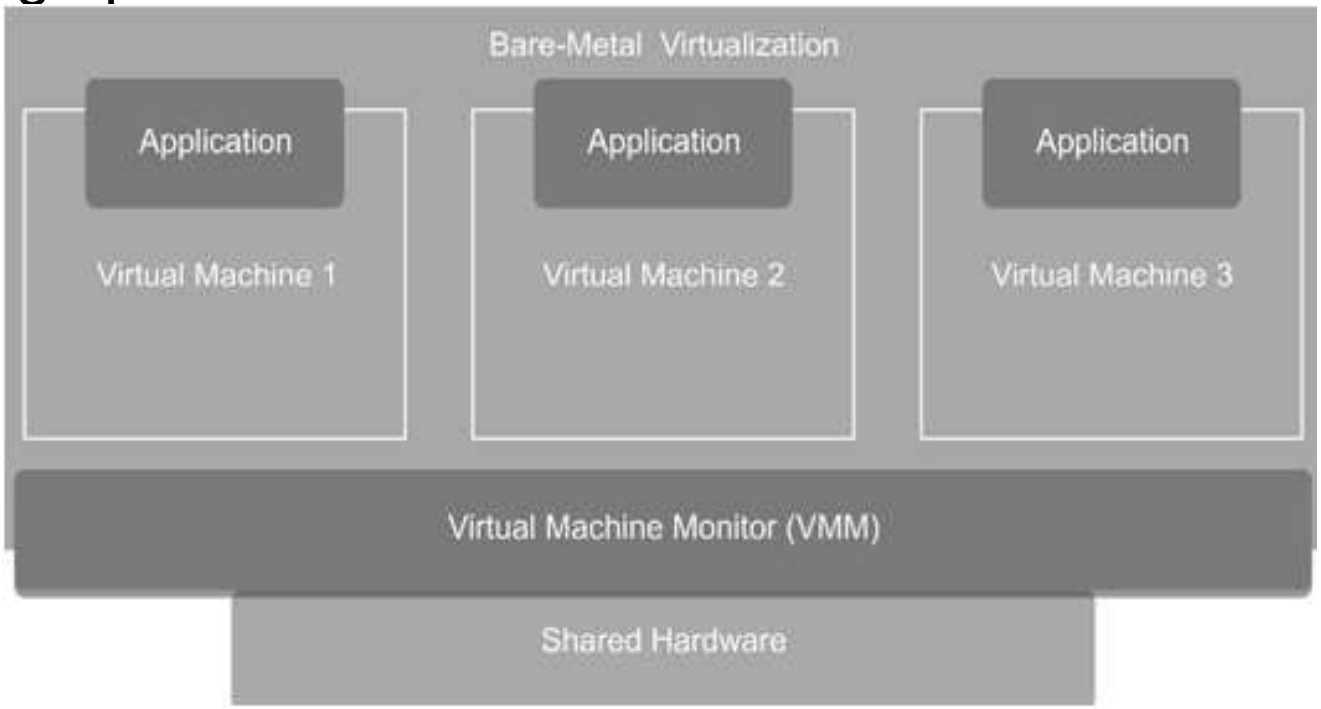
- A **hypervisor** or **virtual machine monitor** (**VMM**) is a piece of computer software, firmware or hardware that creates and runs virtual machines.
- A computer on which a hypervisor is running one or more virtual machines is defined as a *host machine*.
- Each virtual machine is called a *guest machine*.


OS
OS
OS
HYPER VISOR
HARD WARE
TYPE 1
native
(bare metal)
OS
OS
OS
OS
OS
OS
HYPER VISOR
HYPER VISOR
OS
OS
OS
HARD WARE
TYPE 2
hosted

Type-1, native or bare-metal hypervisors

- These hypervisors run directly on the host's hardware to control the hardware and to manage guest operating systems.
- Examples:
  - Oracle VM Server for SPARC,
  - Oracle VM Server for x86
  - Citrix XenServer
  - VMware ESX/ESXi
  - MicrosoftHyper-V 2008/2012.

- The VMM does not rely on the host system for pass-through permissions



- In the bare-metal virtualization technique, you have several options to access I/O devices from the guest systems
- VMM can have direct communication with the I/O devices
- partitioning is another method through which I/O devices can be approached by the hypervisor

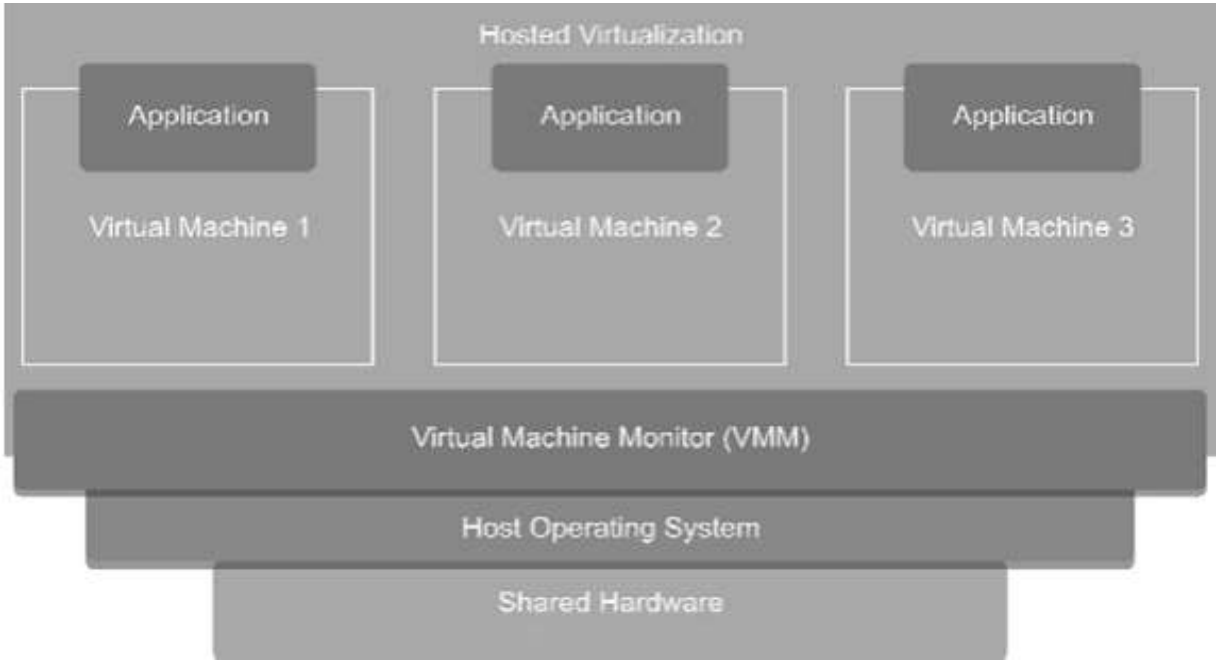
- Benefits and Drawbacks:

- VMMs of the bare-metal type may be used for binding the interrupt latency and enabling deterministic performance
- A single hardware platform can be used to run real-time and general-purpose OSs in parallel
- The hypervisor must include supporting drivers for hardware platforms, apart from including the drivers required for sharing the I/O devices amongst the guest systems
- It is harder to install the VMMs in a bare-metal structure rather than in the hosted structure

Type-2 or hosted hypervisors

- These hypervisors run on a conventional operating system just as other computer programs do.
- Type-2 hypervisors abstract guest operating systems from the host operating system.
- Examples:
  - VMware Workstation
  - VMware Player
  - VirtualBox

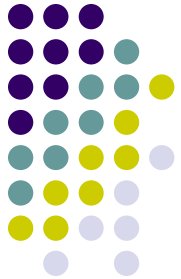
- Enables you to run various guest application windows of your own on top of a base OS with the help of the VMM
- Guest OSs in this virtualization structure have limited access to the I/O devices
- The I/O connections to a given physical system are owned by the host system only while their emulated view is presented (when possible) by the VMM to every single guest machine running on the same base system



- Benefits and Drawbacks:

- With the hosted virtualization structure, multiple guest systems are easily installed, configured, and run
- Setting VMWare Workstation on a Windows-based system through the basic Windows installer is a matter of only a few minutes
- After the VMM is installed, you can run several guest systems on various platforms without any extra physical resource requirement
- The hosted structure is incapable of providing a pass-through to many I/O devices
- Performance of the hosted system may be downgraded, because the I/O requests made by the guest systems must be passed through a host OS
- A real-time OS is not supported in this structure either

Benefits of Virtualization



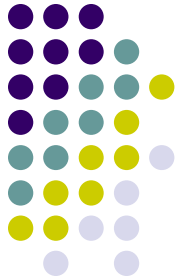
- Maximizing Resources—The pay-as-you-go facility of virtualization helps organizations utilize the maximum amount of required resources.
- Reducing Hardware Costs—When you have no requirements for infrastructure maintenance, the cost for hardware reduces automatically. You do not require installing large servers, huge disk space, or expensive databases, because you can avail these services virtually, anytime

- Minimizing Maintenance Requirements—The lesser is the hardware with you, the lesser is the requirement for maintenance. Virtualization helps you run multiple OSs on a single hardware, which reduces the hardware cost, as well as the need for maintaining the hardware
- Enjoying Benefits of OS Services—Virtualization helps you take advantage of the facilities offered by different Oss
- Using Multiple Systems—Use of multiple systems is made easy with the help of virtualization.

- Testing Beta Software and Maintaining Legacy Applications—

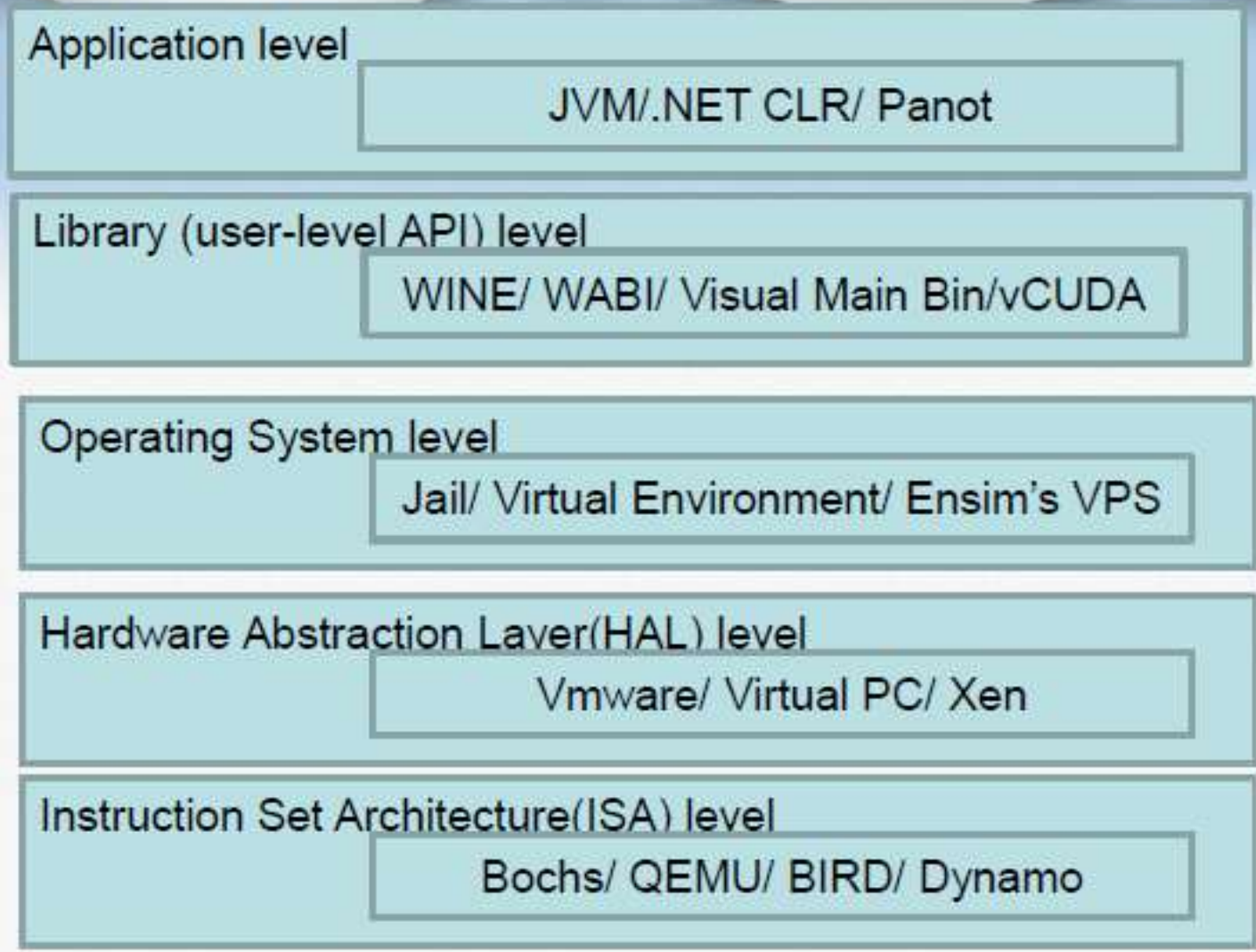
If the OS you use for testing software releases gets corrupted, you can still continue your work uninterrupted with the other system running on the same machine.

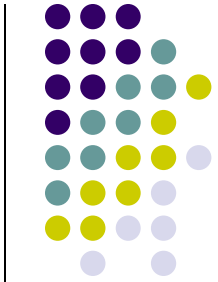
Likewise, if you have a legacy system on which certain applications are run and supported, you can continue with that without requiring to port programs to a different OS.



- Increasing System Security—You can increase the security of your systems through virtualization. Individual systems that are run on virtual machines can be separated from each other. This helps avoid the requirement for different computers to be run on different levels of security without being utilized to their full capacity

Implementation Levels of Virtualization





1. Virtualization at the Instruction Set Architecture (ISA) Level:

- Transforming the physical architecture of the system's instruction set completely into software
- Guest systems issue instructions for the emulator to process and execute
- The instructions are received by the emulator, which transforms them into a native instruction set
- These native instructions are run on the host machine's hardware
- Instructions include both the processor-oriented instructions and the I/O-specific ones

2. Virtualization at the Hardware Abstraction Layer (HAL):

- Time spent in interpreting the instructions issued by the guest platform into the instructions of the host platform is reduced
- It finds similarities that exist between the architectures of the systems
- Virtualization utilizes the native hardware for all its computation and processing by mapping the virtual resources into physical resources
- It increases the efficiency of the virtual machine in handling various tasks
- Cannot fully virtualize all the platforms through this technique

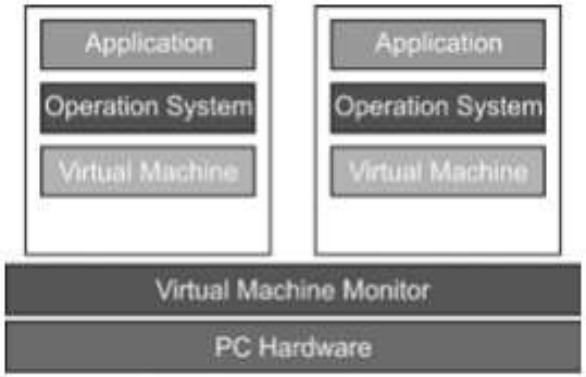


Figure 3: Virtualization at HAL

3. Virtualization at the OS Level:

- To overcome the issues of redundancy and time consumption (to avoid duplicity OS in native & virtual)
- Higher level virtualization
- Sharing both hardware and OS
- The virtualization layer replicates the operating environment, which is established on the physical machine to provide a VE for the application by creating partitions for each virtual system, whenever demanded

4. Virtualization at the Application Level :

The user level programs and OSs are executed on applications that behave like real machines

- I/O mapped input/output processing (in which special I/O instructions are issued for hardware manipulation) or a memory mapped input/output processing technique (in which a small part of memory is mapped to the I/O and then the memory is manipulated) is used to deal with the hardware.
- The set of instructions for an application is defined by the machine specifically for itself
- You can run your applications

on these virtual machines as if you are running your applications on a physical machine

- Less Secure

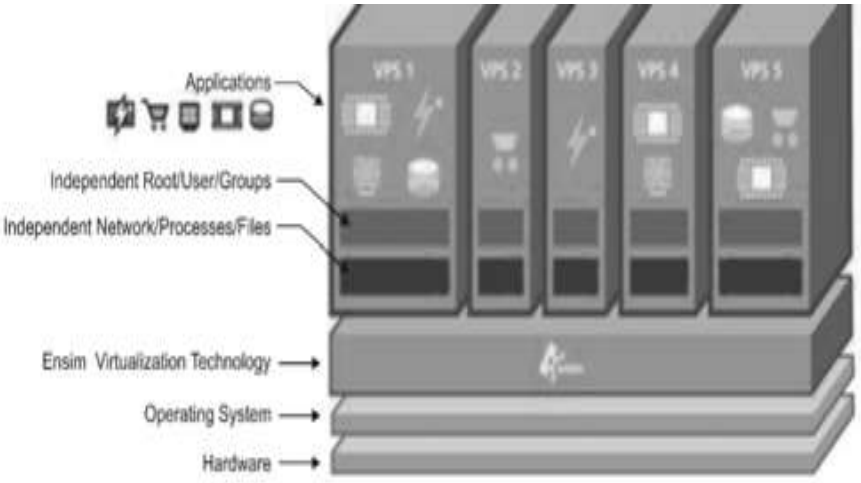


Figure 4: Application-level Virtualization

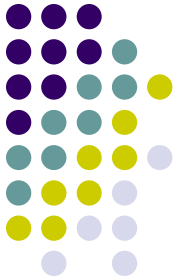
5. Virtualization at the Programming Language Level or Library Level :

- Programming the applications in most systems requires an extensive list of Application Program Interfaces (APIs) to be exported by implementing various libraries at the user-level
- At the user-level library implementation, a different VE is provided in this kind of abstraction
- VE is created above the OS layer, which can expose a different class of binary interfaces altogether
- Implementation of a different set of Application Binary Interfaces (ABIs) and/or APIs being implemented through the base system and performing the function of ABI/API emulation

Comparison between the Implementation Levels of Virtualization

Implementation Level	Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	Very Poor Performance	Excellent	Medium	Medium
HAL	Excellent Performance	Medium	High	Very Good
OS-Level	Excellent Performance	Low	Medium	Very Poor
Library Level	Medium Performance	Low	Low	Very Poor
Application Level	Poor Performance	Low	High	Excellent

Virtualization Design Requirements

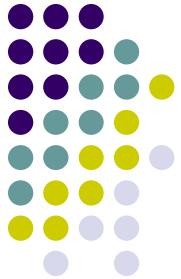


i. Equivalence Requirement—

- A machine that is developed through virtualization must have a logical equivalence with the real machines
- The emulated system must be able to execute all the applications and programs that are designed to execute on the real machines with the only considerable exception of timing

ii. Efficiency Requirement—

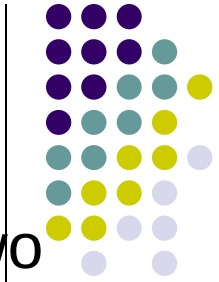
- The virtual machine must be as efficient in its performance as a real system
- Virtualization is primarily done with a purpose of getting efficient software without the physical hardware



iii. Resource Control Requirement—

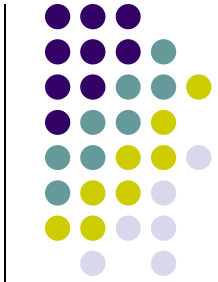
- A typical computer system is a combination of various resources, including processors, memory, and I/O devices. All these resources must be managed and controlled effectively by the VMM
- VMM must be in a state of enforcing isolation between the virtualized systems
- The virtual machines or the VMM should not face any interference in their operations due to other machines in any manner, barring a case where interference is entitled to the requirements for efficiency

Open Source Virtualization Technology

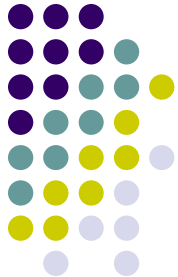


- Kernel-based Virtual Machine (KVM) and Xen are two open-source technologies that provide virtualization support for the Linux operating system
- **KVM** provides virtualization support for Operating Systems (OSs) that are based on x86 hardware coupled with virtualization extensions
- The infrastructure for virtualization, which provided by the kernel module in KVM technology, requires a modified Quick EMUlator (QEMU) for the implementation of virtualization
- KVM is used to host multiple VMs that run Linux OS images or Windows OS images without modification
- Each of the VMs has been provided with its own set of virtualized hardware components that include a network card, disk, graphic adapter, etc.

Xen Hypervisor



- Xen hypervisor is the only bare-metal hypervisor available as open source
- Through Xen, a VM (or a host) can run a number of OS images or multiple different OSs in parallel
- For example, the Xen hypervisor provides server virtualization, desktop virtualization, security applications, IaaS, and embedded and hardware appliances
- The Xen hypervisor is the most widely used virtualization technique in the production environment at present



The key features of the Xen hypervisor include the following:

- Robustness and Security: The technique follows the microkernel design approach, offering a higher level of robustness and security to the applications than other hypervisors.
- Scope for Other Operating Systems: Not only can the Xen hypervisor be run on the Linux OS working as the main control stack but it can also be adjusted to other systems as well
- Isolation of Drivers from the Rest of the System: The main device drivers can be allowed by the Xen hypervisor to run inside a VM, and in case the driver suffers a crash or is compromised, it can be restarted by rebooting the VM that contains the driver without causing any effect on the other parts of the system.

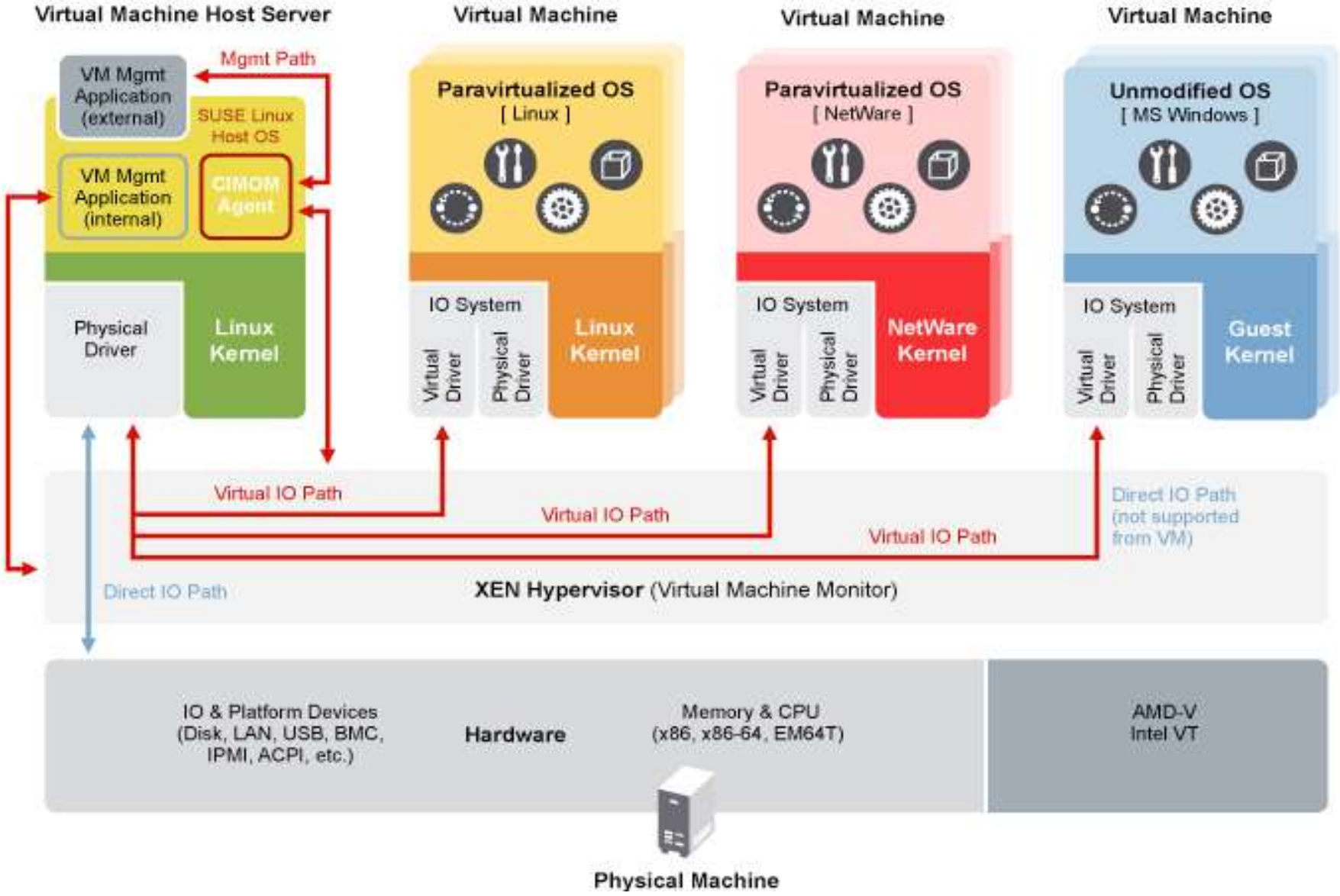
Continue…

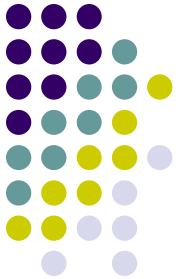
- Support for Paravirtualization:

The Xen hypervisor provides optimization support for paravirtualized guests so that they can be run as VMs. This feature helps guests run faster than the hypervisors providing the hardware extension. Hardware having no support for virtualization extension can also be used with the Xen hypervisor

* Note: KVM is Type 2 hypervisor where Xen is Type 1 hypervisor

Xen Virtualization Architecture





Thank You!