SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCS7023

CLOUD COMPUTING

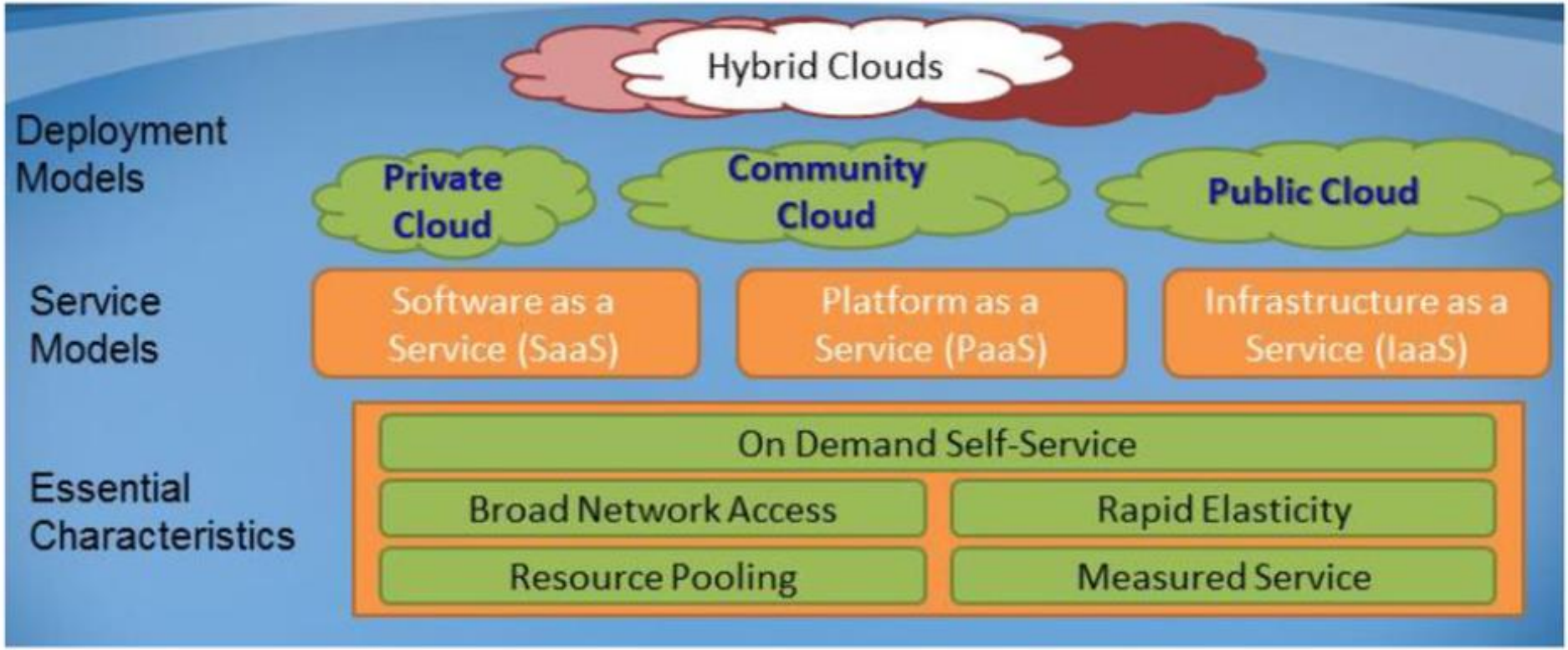
UNIT 3

CLOUD DEPLOYMENT MODELS

UNIT 3 CLOUD DEPLOYMENT MODELS AND VIRTUALIZATION 9 Hrs.
Deployment models: Public cloud – Private Cloud –Hybrid cloud – Community cloud - Need for virtualization – Types of Virtualization – Virtualization OS – VMware, KVM – System VM – Process VM - Virtual Machine Monitor – Properties - Xen, Hyper V, Virtual Box, Eucalyptus .

OVERVIEW OF CLOUD COMPUTING

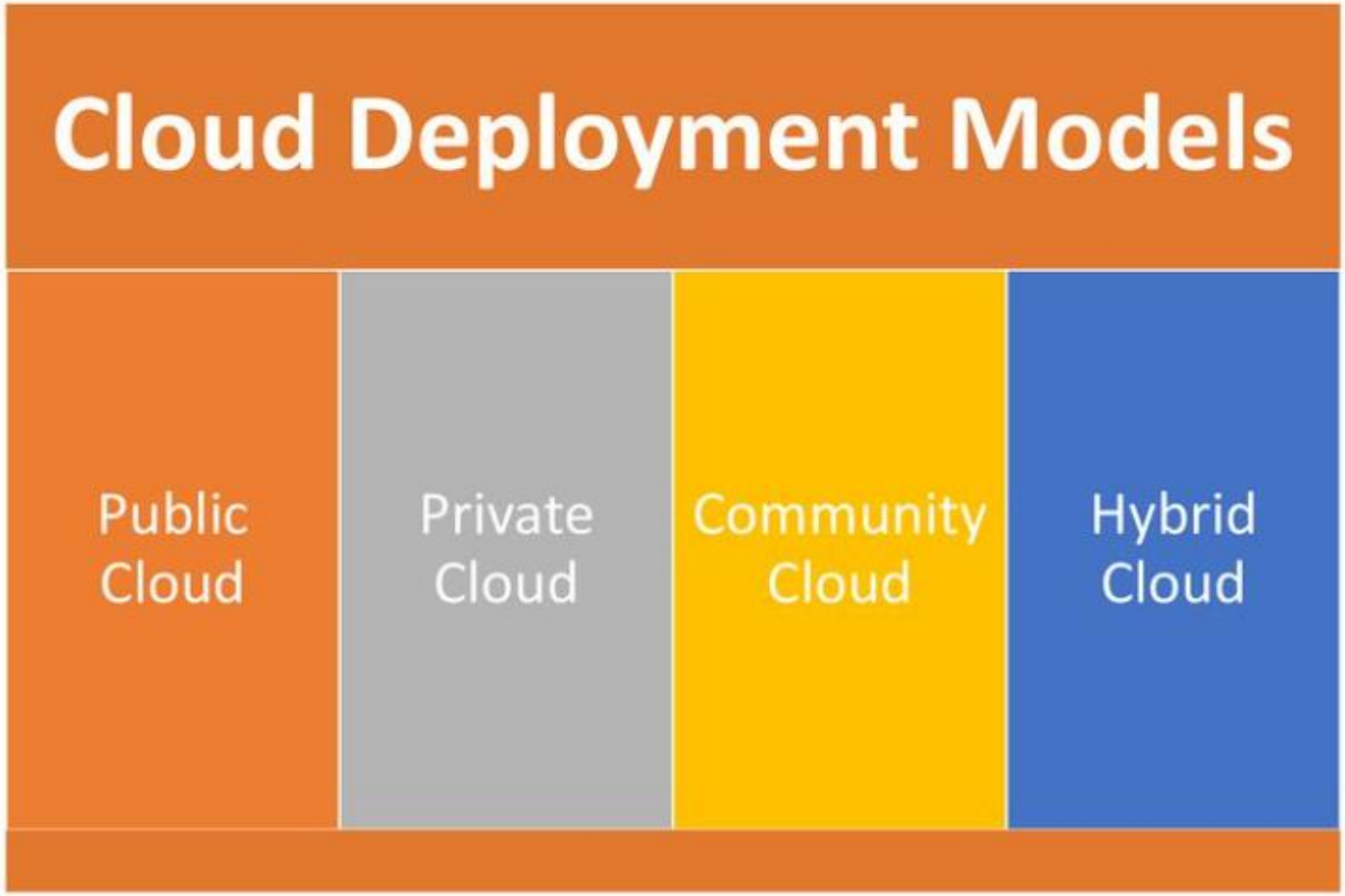
Architectural Overview of Cloud Computing

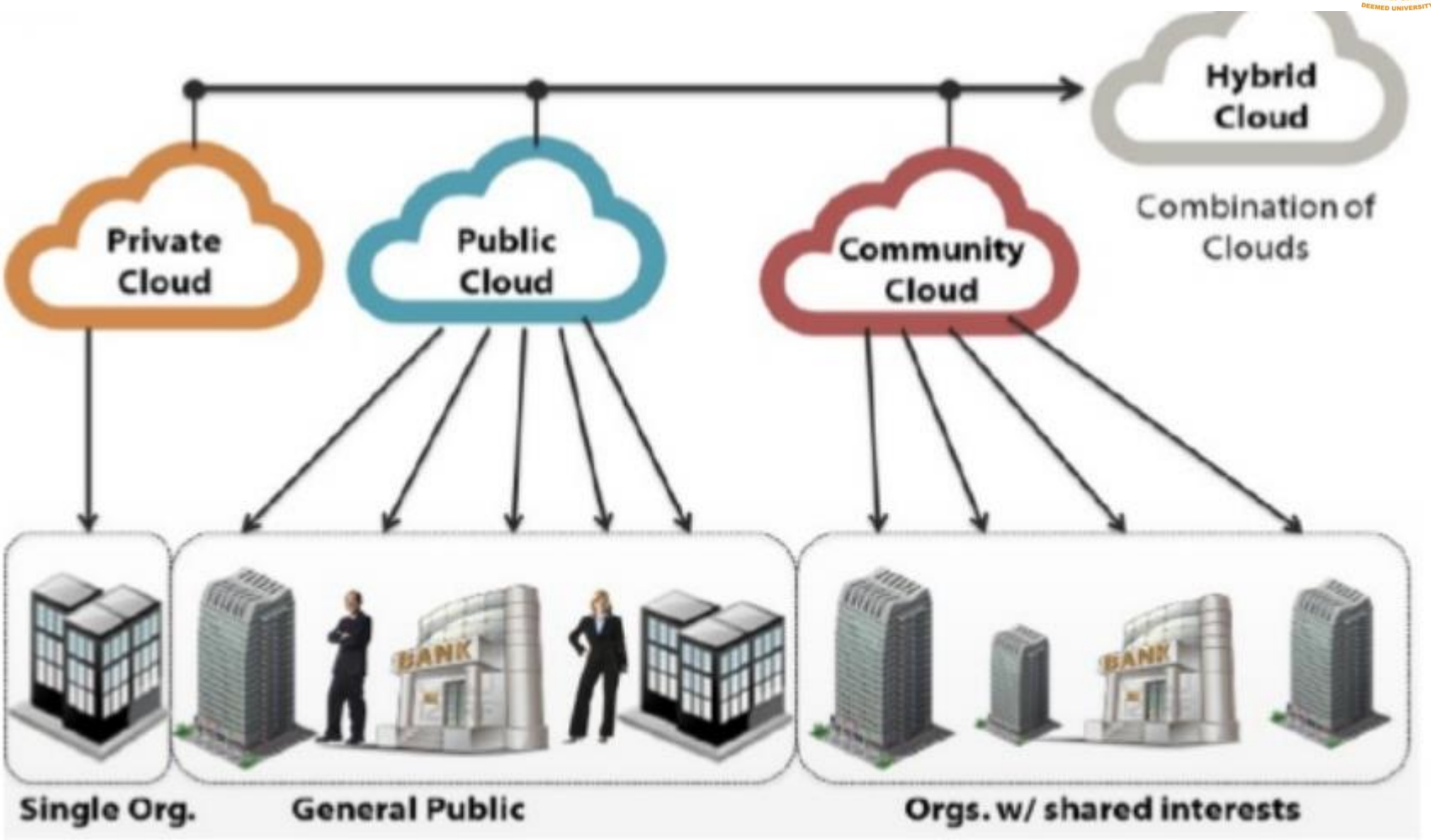


CLOUD DEPLOYMENT MODELS

- Cloud Deployment Models
  - ✓Public Cloud
  - ✓Private Cloud
  - ✓Community Cloud
  - ✓Hybrid Cloud
  - ✓ Architectural Overview of Cloud Computing
  - ✓References

CLOUD DEPLOYMENT MODELS



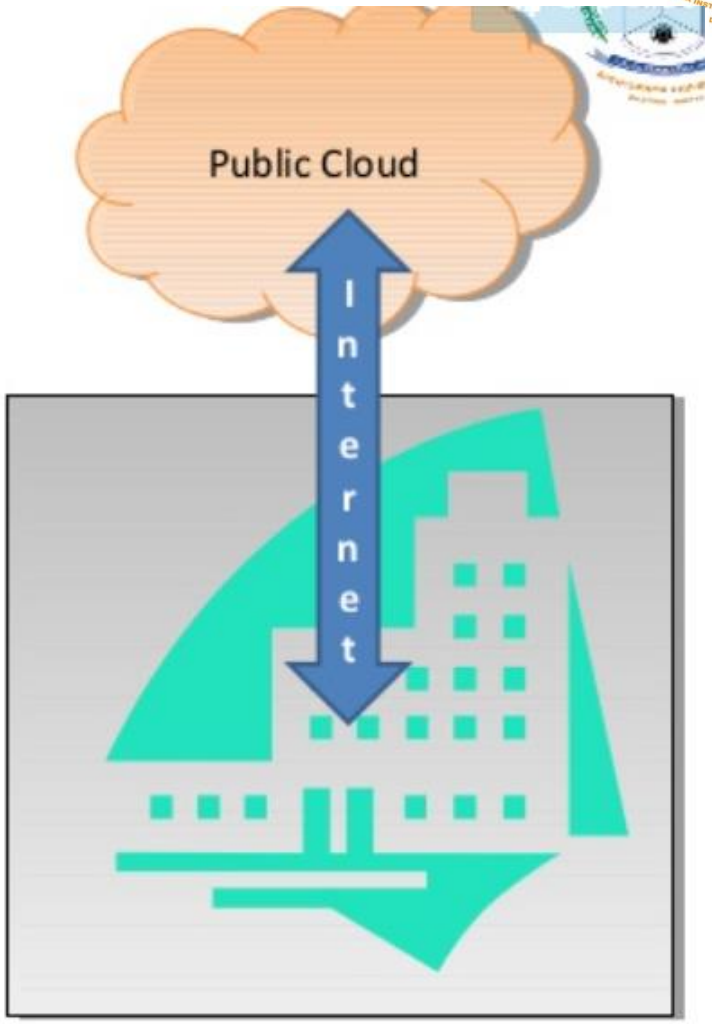

Hybrid Cloud
Combination of Clouds
Private Cloud
Public Cloud
Community Cloud
BANK
BANK
Single Org.
General Public
Orgs. w/ shared interests

PUBLIC CLOUD

- A *public cloud* is a publicly accessible cloud environment owned by a third-party cloud provide
- The IT resources on public clouds are usually provisioned via cloud delivery models .
- The cloud provider is responsible for the creation and on-going maintenance of the public cloud and its IT resources.

PUBLIC CLOUD

- Available to everyone.
- Anyone can go and signup for the service.
- Economies of Scale due to Size.
- Some public cloud concerns
  - Ownership
  - Control
  - Regulatory compliance
  - Data/Application security
  - Liability for SLA breaches

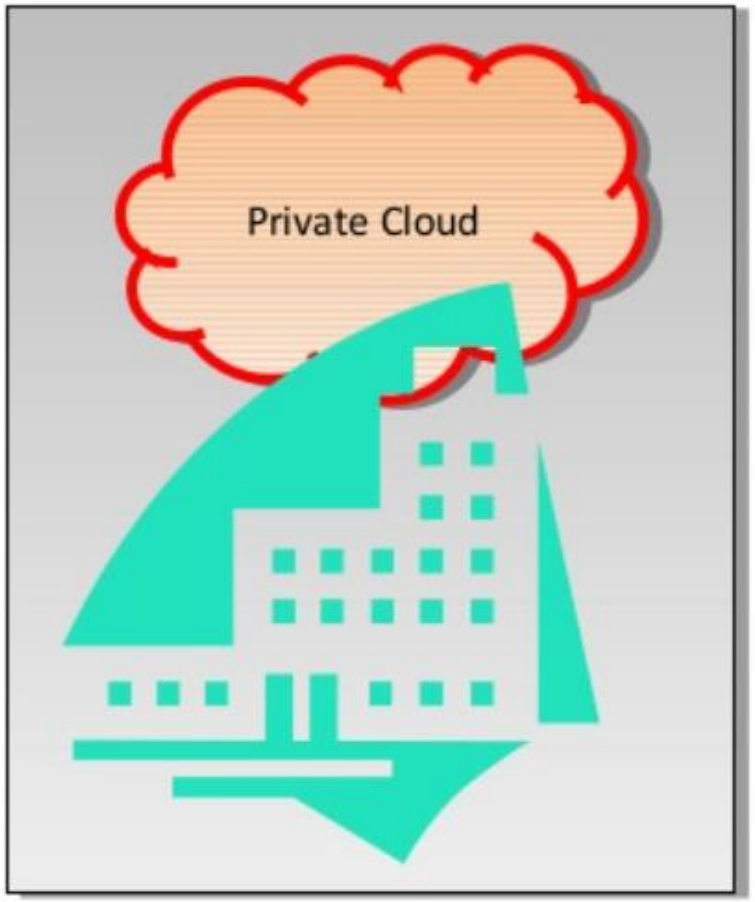


PRIVATE CLOUD

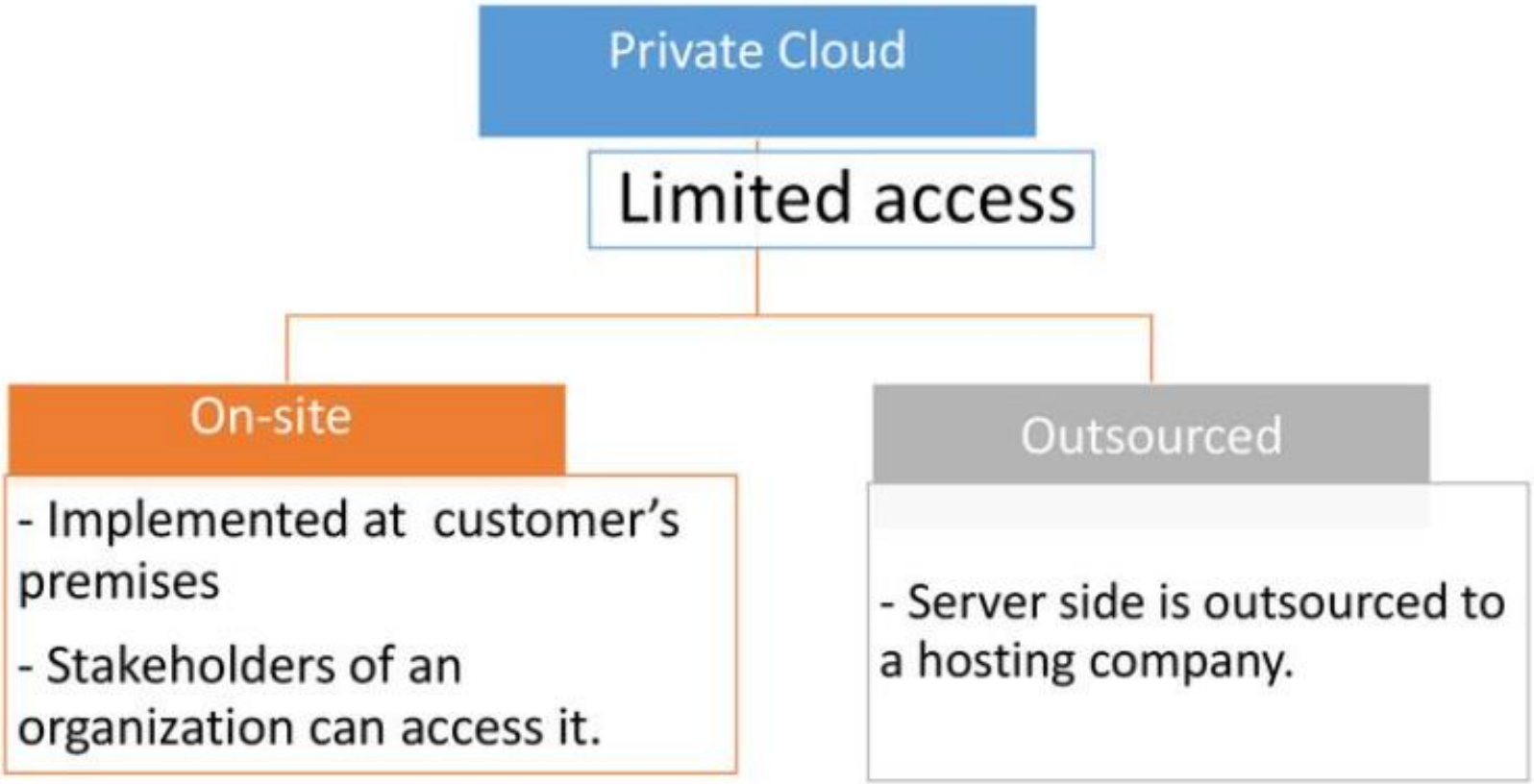
- A private cloud is owned by a single organization.
- Private clouds enable an organization to use cloud computing technology as a means of centralizing access to IT resources by different parts, locations, or departments of the organization.
- When a private cloud exists as a controlled environment, the problems described in the Risks and Challenges section do not tend to apply.

PRIVATE CLOUD

- Cloud infrastructure built in house
- Retains control of resources
- More security & privacy
- Can conform to regulatory requirement
- Needs capital investment
- Needs expertise to build and maintain

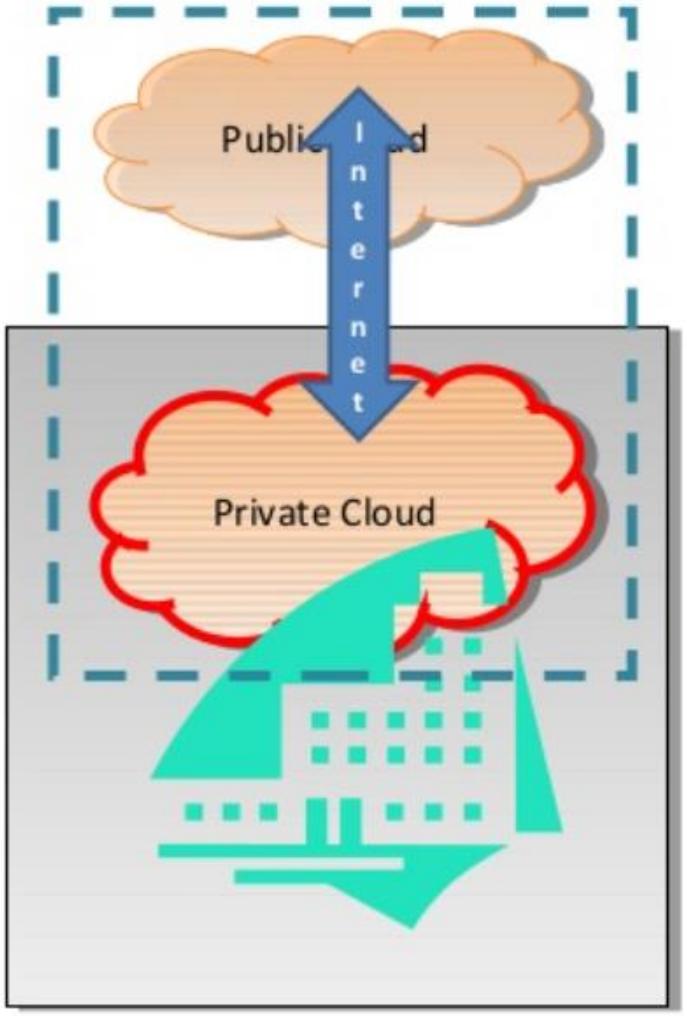


PRIVATE CLOUD



HYBRID CLOUD

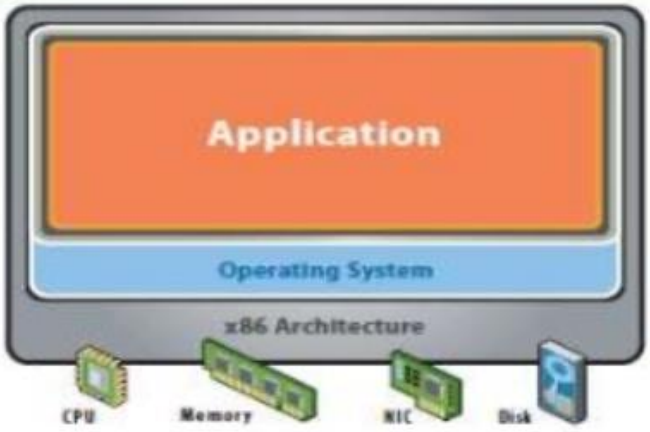
- A hybrid cloud is a cloud environment comprised of two or more different cloud deployment models.
- Best of Both World
- Workload is deployed mostly on private cloud
- Resources can be used from public cloud when there is a surge in peak load (Cloud Burst)



HYPERVISOR

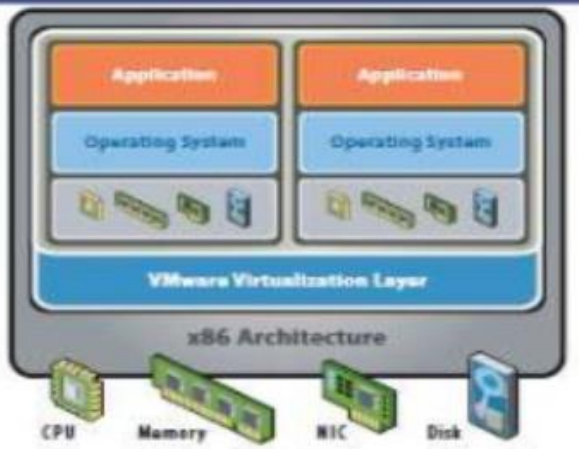
WHAT IS VIRTUALIZATION?

Virtualization allows multiple operating system instances to run concurrently on a single computer; it is a means of separating hardware from a single operating system.



Before Virtualization:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure



After Virtualization:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

WHAT IS VIRTUALIZATION?

- Broadly, the separation of a resource or request for a service from the underlying physical delivery of that service

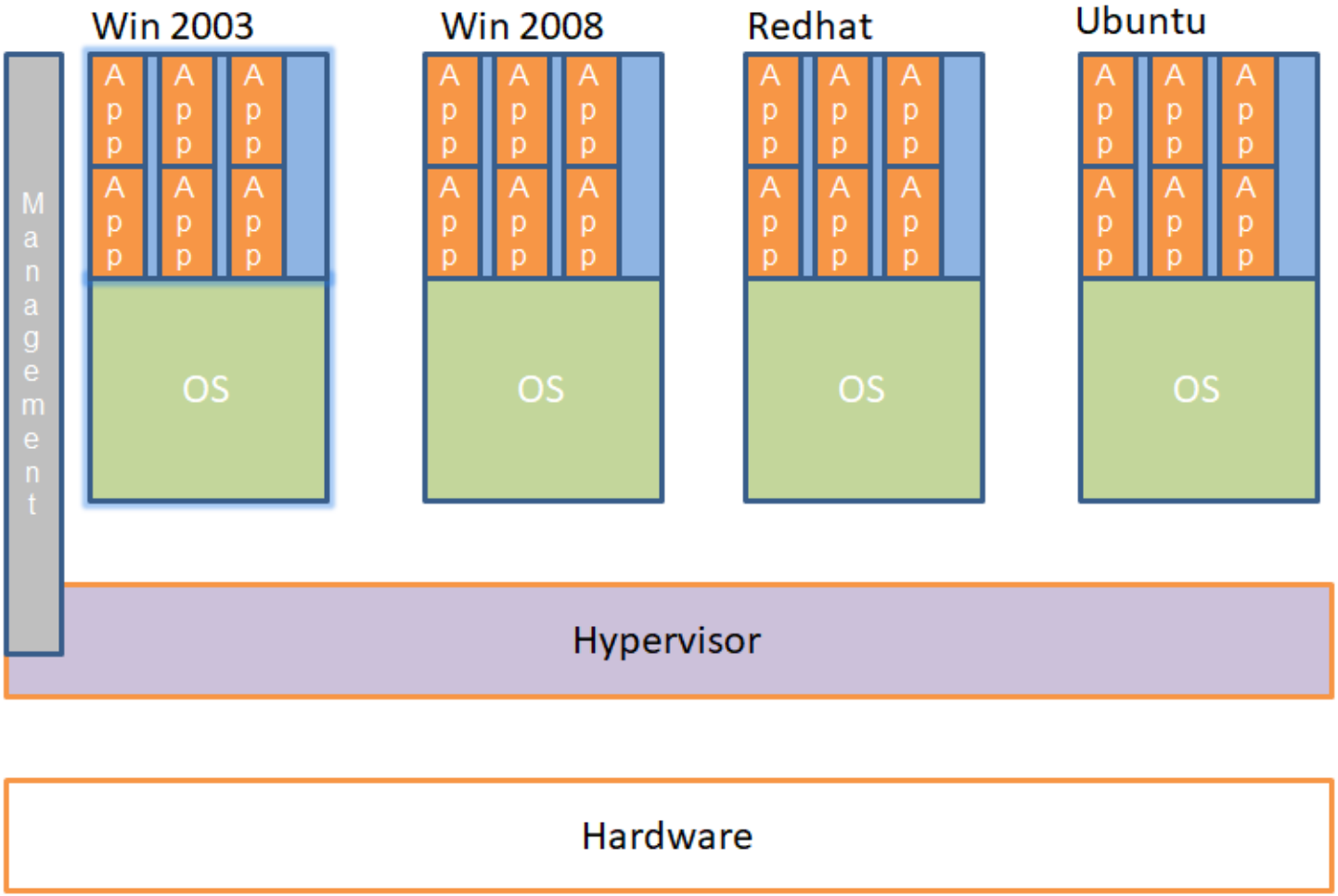
- *Concept in which access to a single underlying piece of hardware is coordinated so that multiple guest operating systems can share that single piece of hardware, with no guest operating system being aware that it is actually sharing anything at all*
  - Guest Operating System is the OS that is hosted by the virtualization software on the Host Operating System

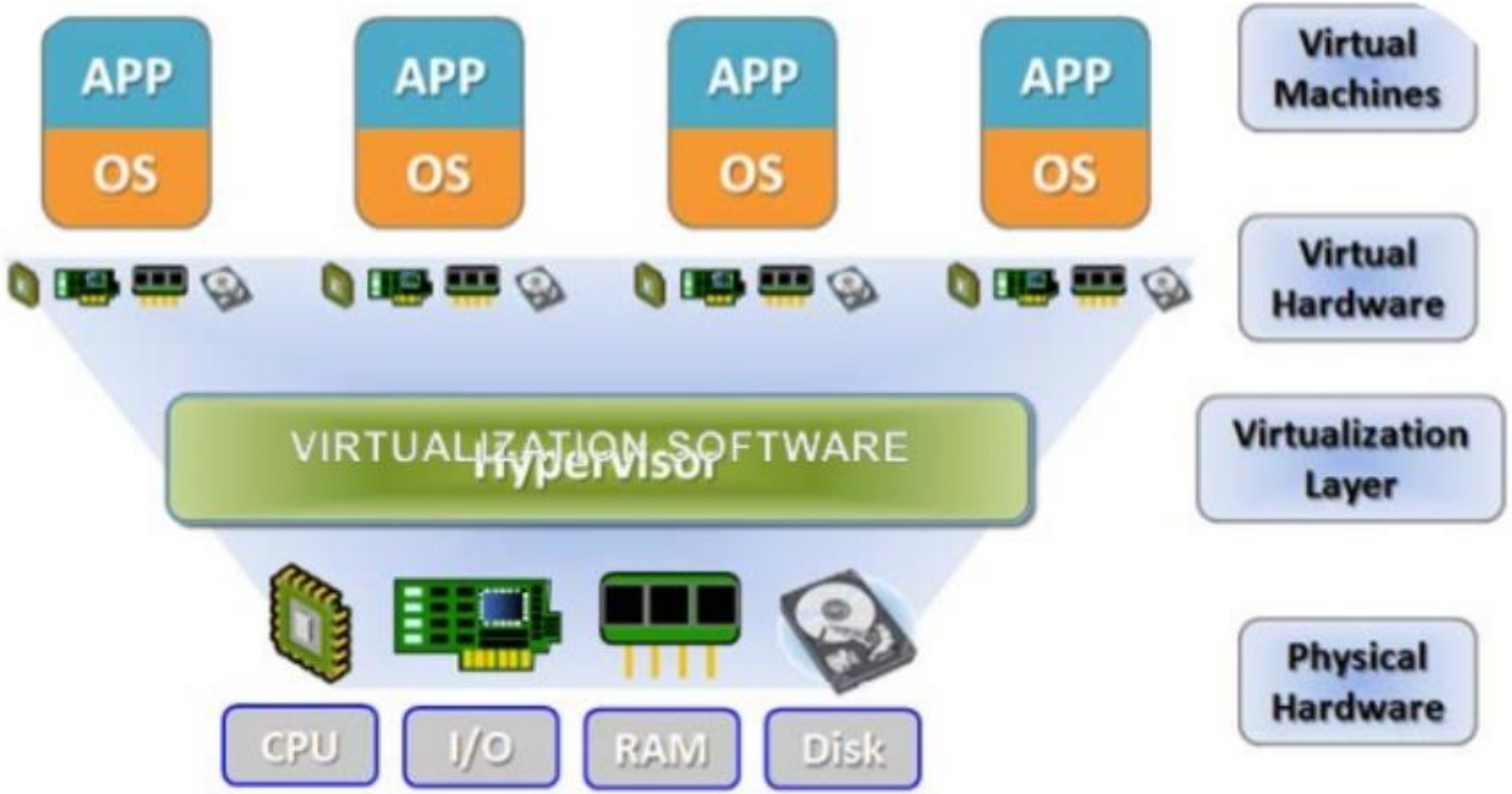
WHAT IS VIRTUALIZATION?

- Before using Virtualization, we had:
  - Single OS per machine
  - Software and hardware tightly coupled
  - Underutilized resources (idle time)
  - Inflexibility

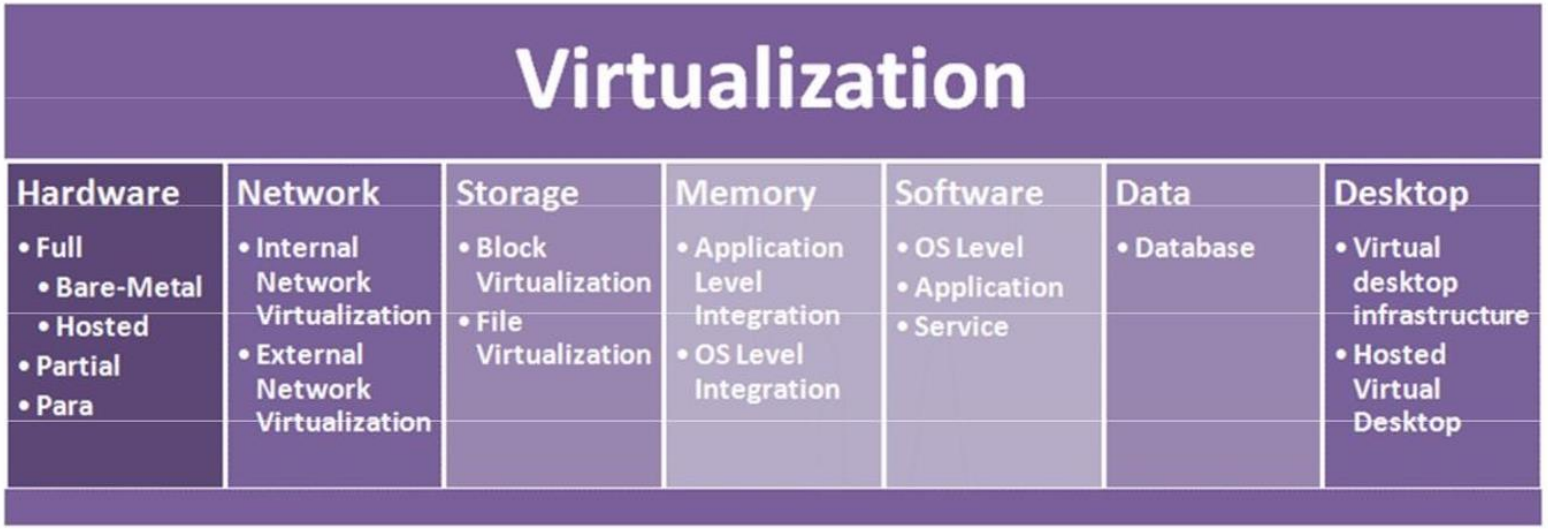
- Virtualization gives you:
  - Hardware independence of operation system and applications
  - Ability to encapsulate OS and applications in to virtual machines
  - Ability to provision virtual machines to any system

TYPICAL SERVER VIRTUALIZATION DEPLOYMENT




APP
OS
APP
OS
APP
OS
APP
OS
Virtual Machines
Virtual Hardware
VIRTUALIZATION SOFTWARE
Hypervisor
Virtualization Layer
CPU
I/O
RAM
Disk
Physical Hardware

Types of Virtualization



Approaches of ways to virtualizes cloud servers.

- Grid Approach
- OS - Level Virtualization
- Hypervisor-based Virtualization
- Software virtualization
- Hardware virtualization
- Server virtualization

Virtualization

- **Grid Approach -** where the processing workloads are distributed among different physical servers, and their results are then collected as one.
- **OS - Level Virtualization -**Here, multiple instances of an application can run in an isolated form on a single OS
- **Hypervisor-based Virtualization-**With hypervisor's virtualization, there are various sub-approaches to fulfill the goal to run multiple applications & other loads on a single physical host

Virtualization

- **Hardware Virtualization -**It is the abstraction of computing resources from the software that uses cloud resources. It involves embedding virtual machine software into the server's hardware components. That software is called the hypervisor.

Virtualization

Three types of **hardware virtualizations**

- **Full Virtualization -** Here the hardware architecture is completely simulated. Guest software doesn't need any modification to run any applications
- **Emulation Virtualization -** Here the virtual machine simulates the hardware & is independent. Furthermore, the guest OS doesn't require any modification
- **Para-Virtualization -** Here, the hardware is not simulated; instead the guest software runs its isolated system

Software virtualization

- It is also called application virtualization
- Software virtualization is similar to that of virtualization except that it is capable to abstract the software installation procedure and create virtual software installation. Many applications & their distributions became typical tasks for IT firms and departments. The mechanism for installing an application differs.
- So virtualized software is **introduced in which an application that will be installed into its self-contained unit and provide software virtualization**. Some of the examples are **Virtual Box, VMware**, etc
- Benefits are **Ease of Client Deployment , Software Migration, Easy to Manage**

Server virtualization:

- In this process, the server resources are kept hidden from the user. This partitioning of physical server into several virtual environments; result in the dedication of one server to perform a single application or task.
- This technique is mainly used in web-servers which reduces the cost of web-hosting services. Instead of having separate system for each web-server, multiple virtual servers can run on the same system/computer.

Primary uses of server virtualization are:

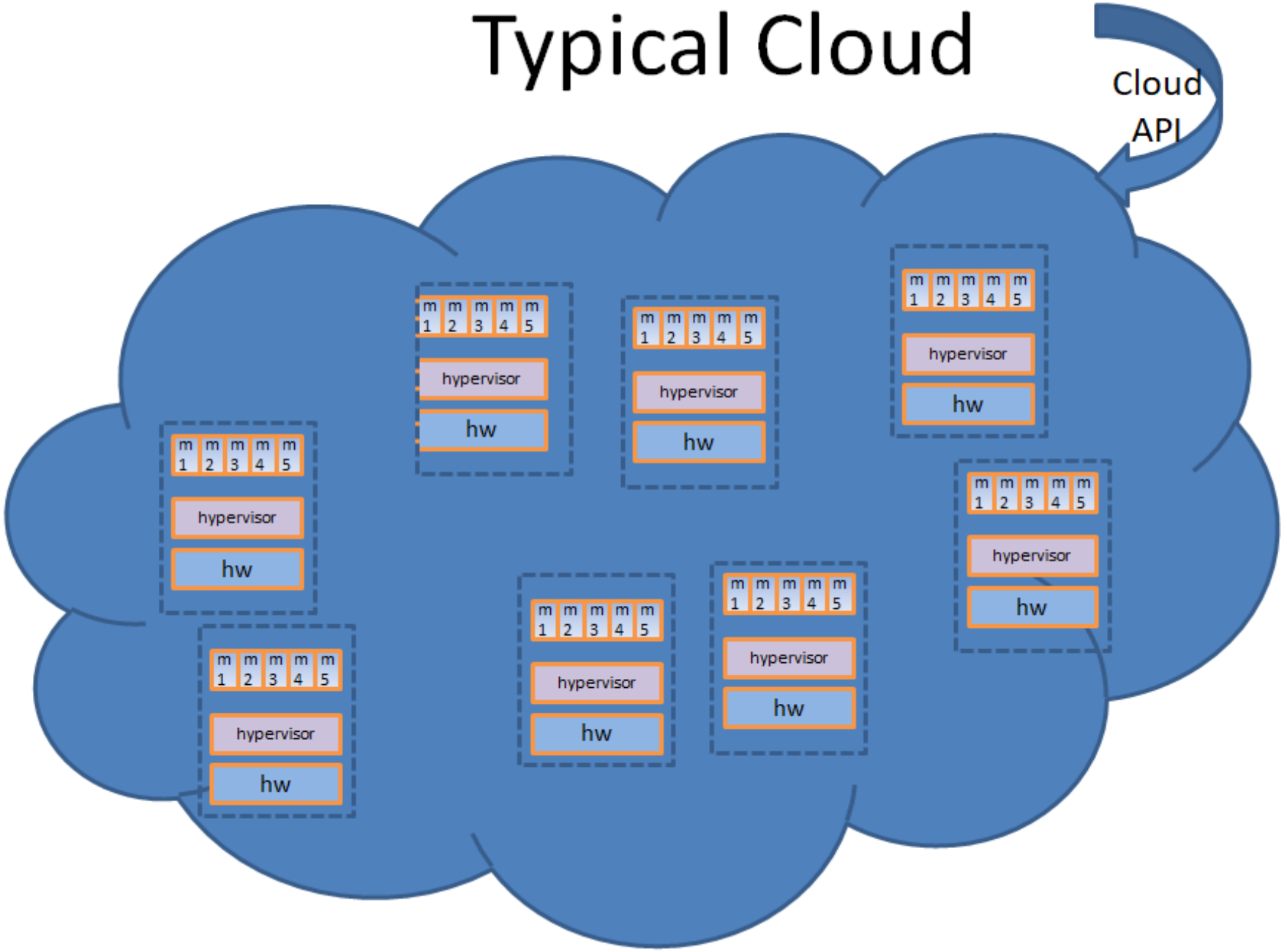
- To centralize the server administration
- Improve the availability of server
- Helps in disaster recovery
- Ease in development & testing
- Make efficient use of server resources.

HYPERVISOR

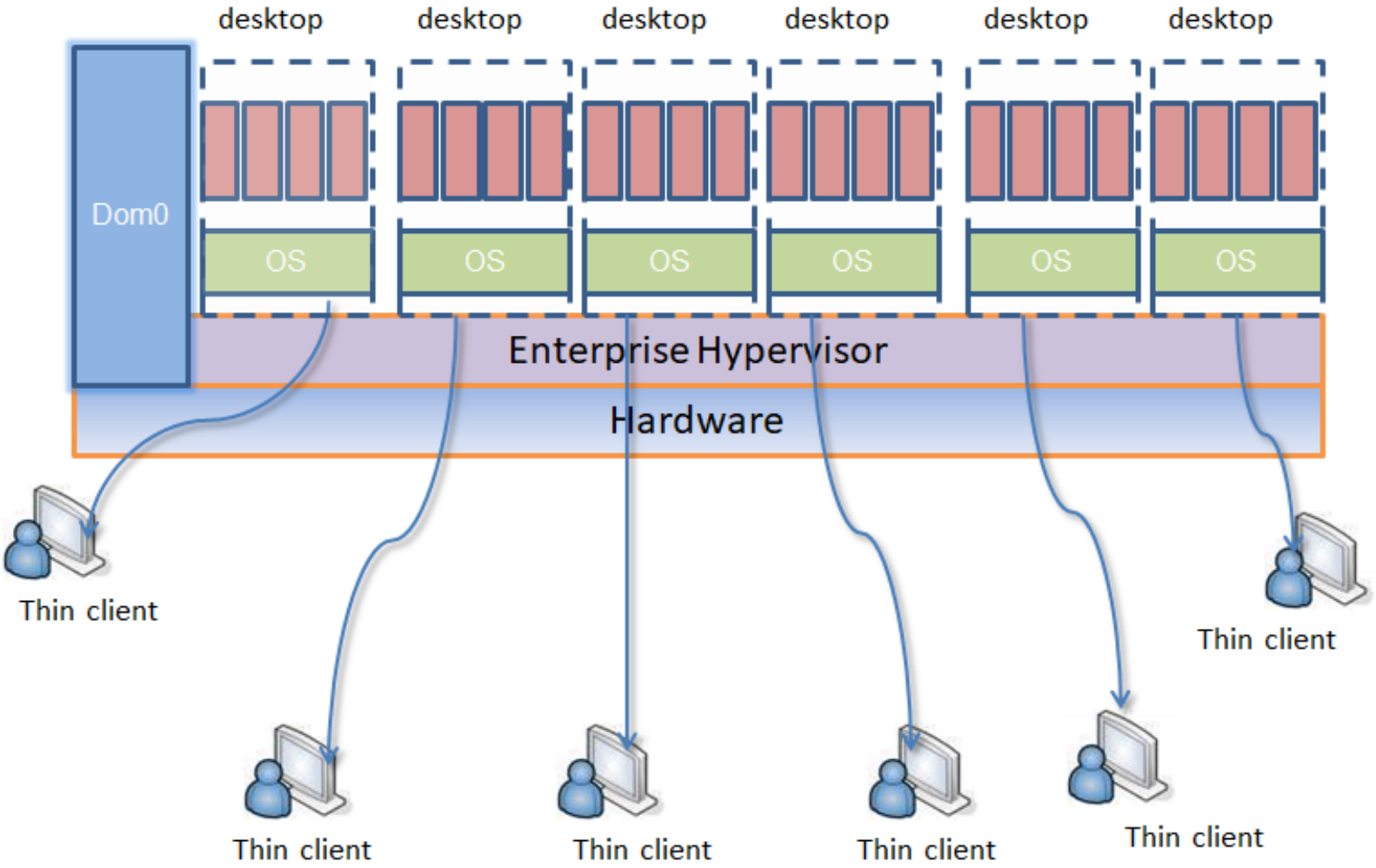
- **Hypervisors**: The approach to virtualization

- **Hypervisor** also known as the Virtual Machine Monitor
- Software that allows multiple operating systems to share a single hardware host
- Emulates hardware resources to guest operating systems
- Each operating system appears to have host's processor, memory, and other resources all to itself
- In reality, hypervisor controls the resources and allocates them as needed by each guest OS in a synchronized manner

Typical Cloud



TYPICAL VDI(Virtual Desktop Infrastructure) DEPLOYMENT



ORIGIN OF HYPERVISOR

- IBM first developed CP/CMS operating system in 1967, an attempt to build time-sharing systems for mainframe systems
- In 1972, IBM's zSeries line featured Virtualization
- Early acceptance and rapid development by developers all over
- In 1985, IBM introduced the PR/SM hypervisor to manage logical partitions
- Other companies, Sun Microsystems, HP, and SGI joined the race and started selling virtualized software around 2000

SECURITY ISSUES IN HYPERVISOR

- Hypervisor is the ***underlying component*** of all these architectures. It is a new layer which needs to be protected
- Scale of deployments – Just the sheer scale of deployments make this a security nightmare.
- Imagine 150 machines running a simultaneous scheduled AV scan on the same physical host. Chokes IO/Disk bandwidth.
- Isolation - Machines of a **company and its competitor** could be running on the same physical machine. Insufficient isolation could lead to disaster
- New API's to access Virtualization/Cloud services.
- Bugs in these could lead to compromise of entire infrastructure.

SECURITY SOLUTIONS

- Trusted hypervisor
  - Hyperguard – Phoenix Technologies – A hypervisor integrity scanner in SMM.
  - Deepwatch – Intel project – Virtualization rootkit scanner
- Domain 0 Hardening – Various security solutions to white-list and harden Dom0

SECURITY OPPORTUNITIES

- New breed of security products is now possible to protect guest OS's from being hijacked
- Hypervisor based security suites cannot be detected by malware running in the guest
- Hypervisors allow introspection of very early boot sequences of the guest, thereby making possible an entire need breed of BIOS rootkit and kernel rootkit scanners

CLASSIFICATIONS OF HYPERVISOR

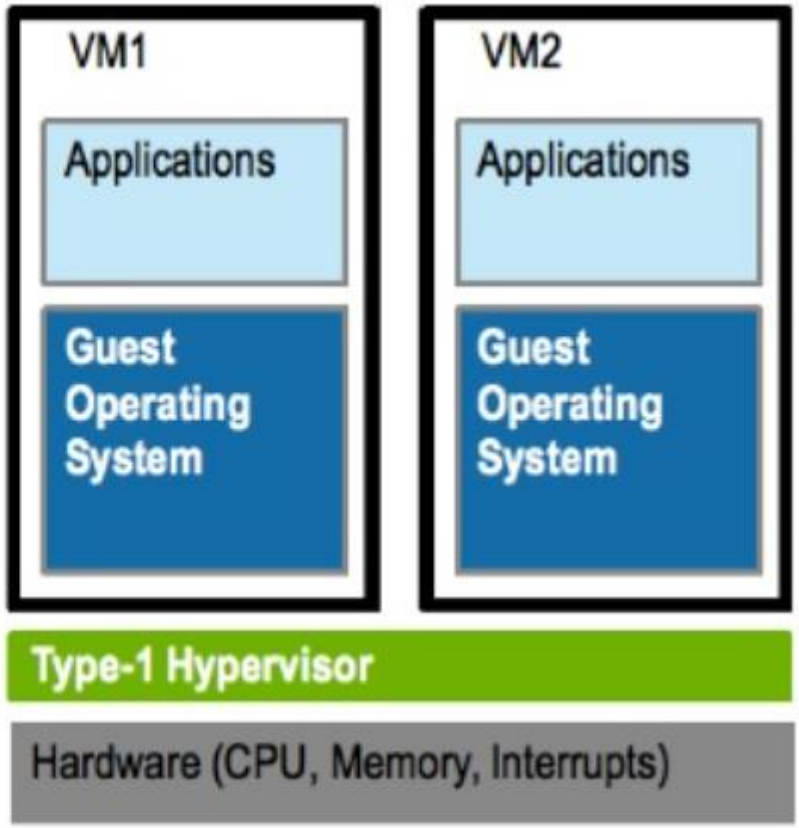
There are two types of hypervisor based on architecture:

- **Type 1 Hypervisor**
  - also known as native or bare metal hypervisor
- **Type 2 Hypervisor**
  - also known as hosted hypervisor

CLASSIFICATIONS OF HYPERVISOR

Type 1 Hypervisor

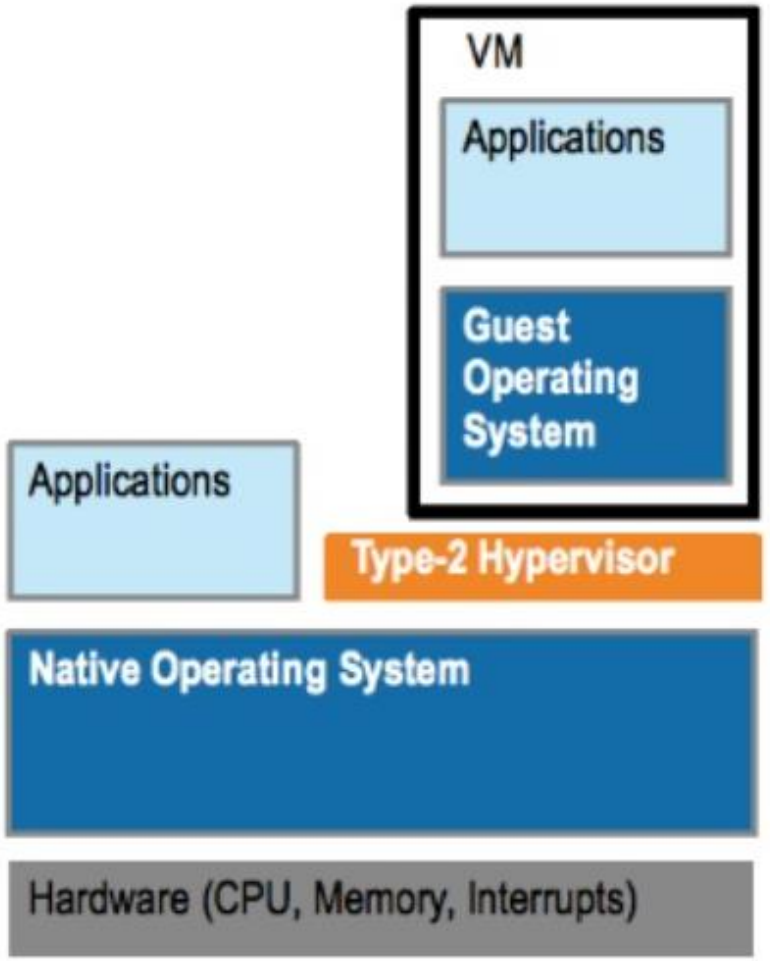
- Runs directly on the host's hardware to manage guest operating systems
- Does not require any base server operating system
- Direct access to hardware resources
- Better performance, scalability, and stability
- However, hardware support is limited



CLASSIFICATIONS OF HYPERVISOR

Type 2 Hypervisor

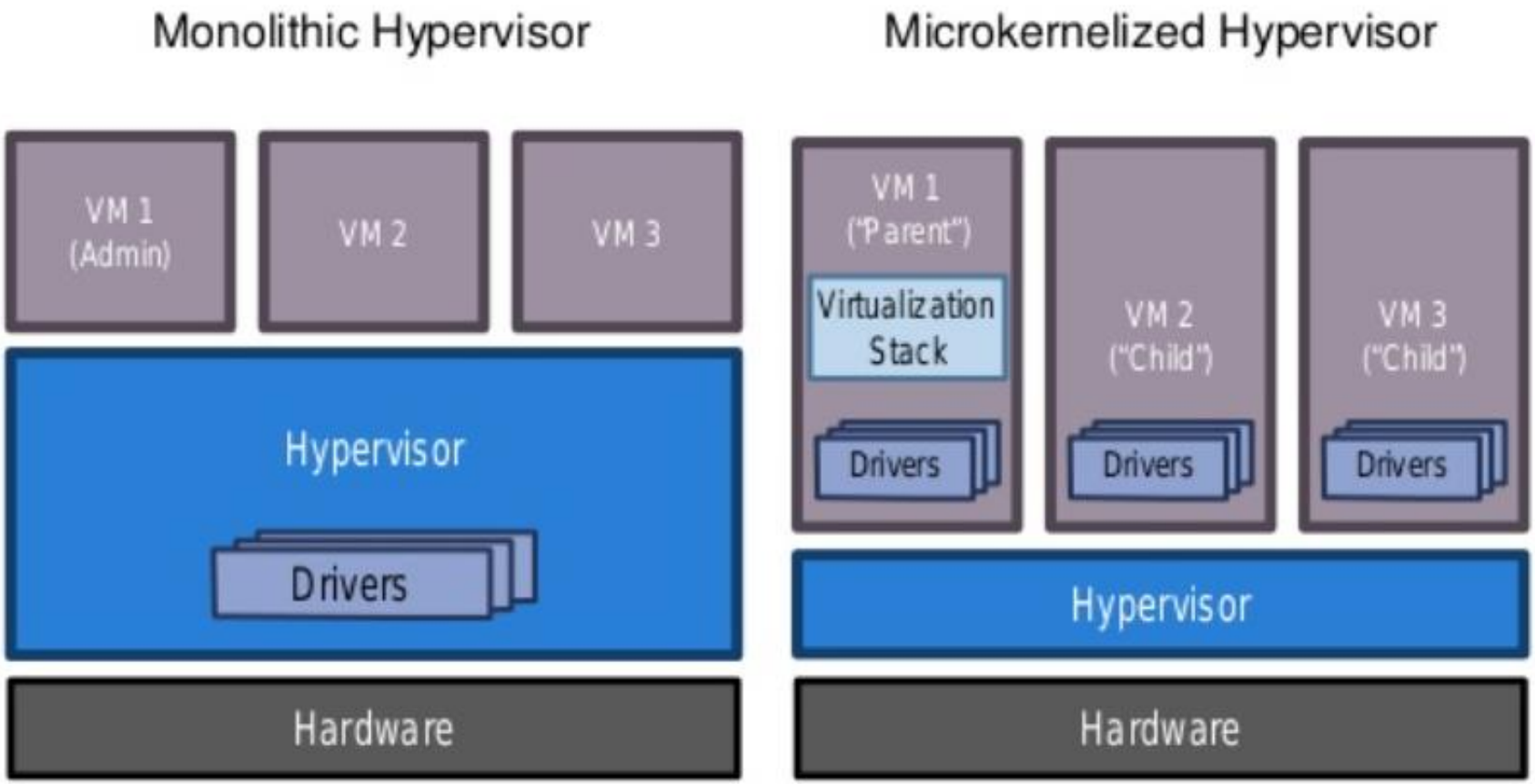
- Hosted on the main operating system
- Basically a software installed on an OS
- Hypervisor asks OS to make hardware calls
- Better compatibility with hardware
- Increased overhead affects performance



CLASSIFICATIONS OF HYPERVISOR

Type 1 Hypervisors can be further classification into two main ways to architect the hypervisor solutions:

- **Monolithic**
  - Hosts the hypervisor/VMM in a single layer that also includes most of the required components, such as the kernel, device drivers, and the I/O stack
- **Microkernelized**
  - Uses a very thin, specialized hypervisor that only performs the core tasks of ensuring partition isolation and memory management. This layer does not include the I/O stack or device drivers.
  - Virtualization stack and hardware-specific device drivers are located in a specialized partition called the parent partition.


Monolithic Hypervisor
VM 1
(Admin)
VM 2
VM 3
Hypervisor
Drivers
Hardware
Microkernelized Hypervisor
VM 1
("Parent")
Virtualization
Stack
Drivers
VM 2
("Child")
Drivers
VM 3
("Child")
Drivers
Hypervisor
Hardware

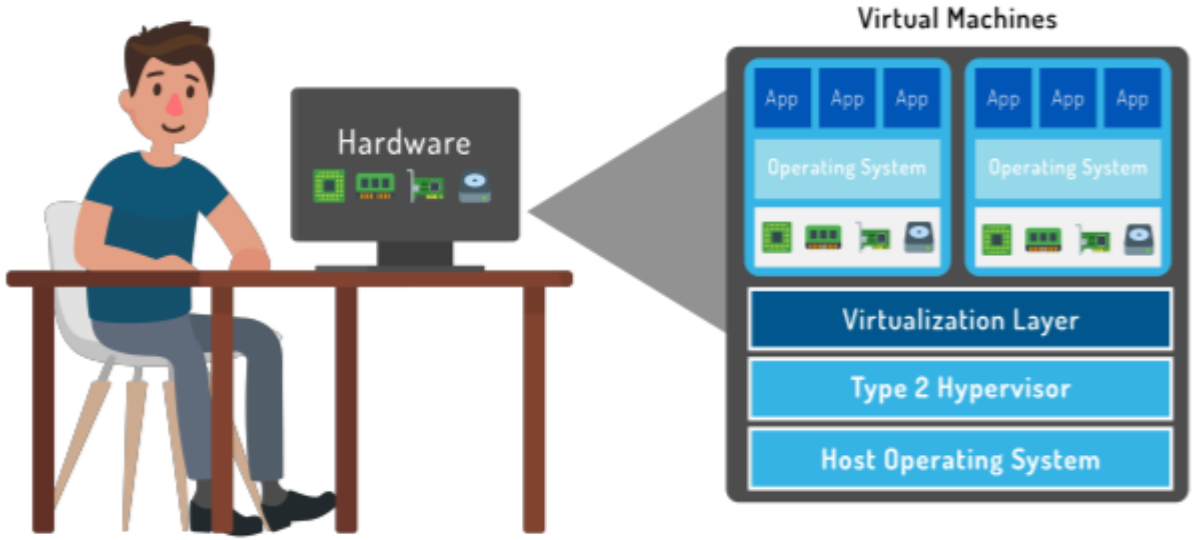
LEADING HYPERVISORS

- Type 1
  - VMware ESX and ESXi
  - Microsoft Hyper-V
  - Citrix Systems XenServer

- Type 2
  - VMware Workstation
  - Oracle VM VirtualBox
  - Microsoft Virtual PC
  - Parallels Desktop

What is a Hypervisor?

- Software installed on top of hardware that created virtualization layer
- Hosts VMs
- Type 1 Hypervisor – Bare metal hypervisor (VMware ESXi)
- Type 2 Hypervisor – Hosted hypervisor (VMware Workstation)



CASE STUDIES : VMware

What is a VM?

- Virtualization creates virtual hardware by cloning physical hardware
- The hypervisor uses virtual hardware to create a virtual machine (VM)
- A VM is a set of files
- With a hypervisor and VMs, one computer can run multiple OS simultaneously

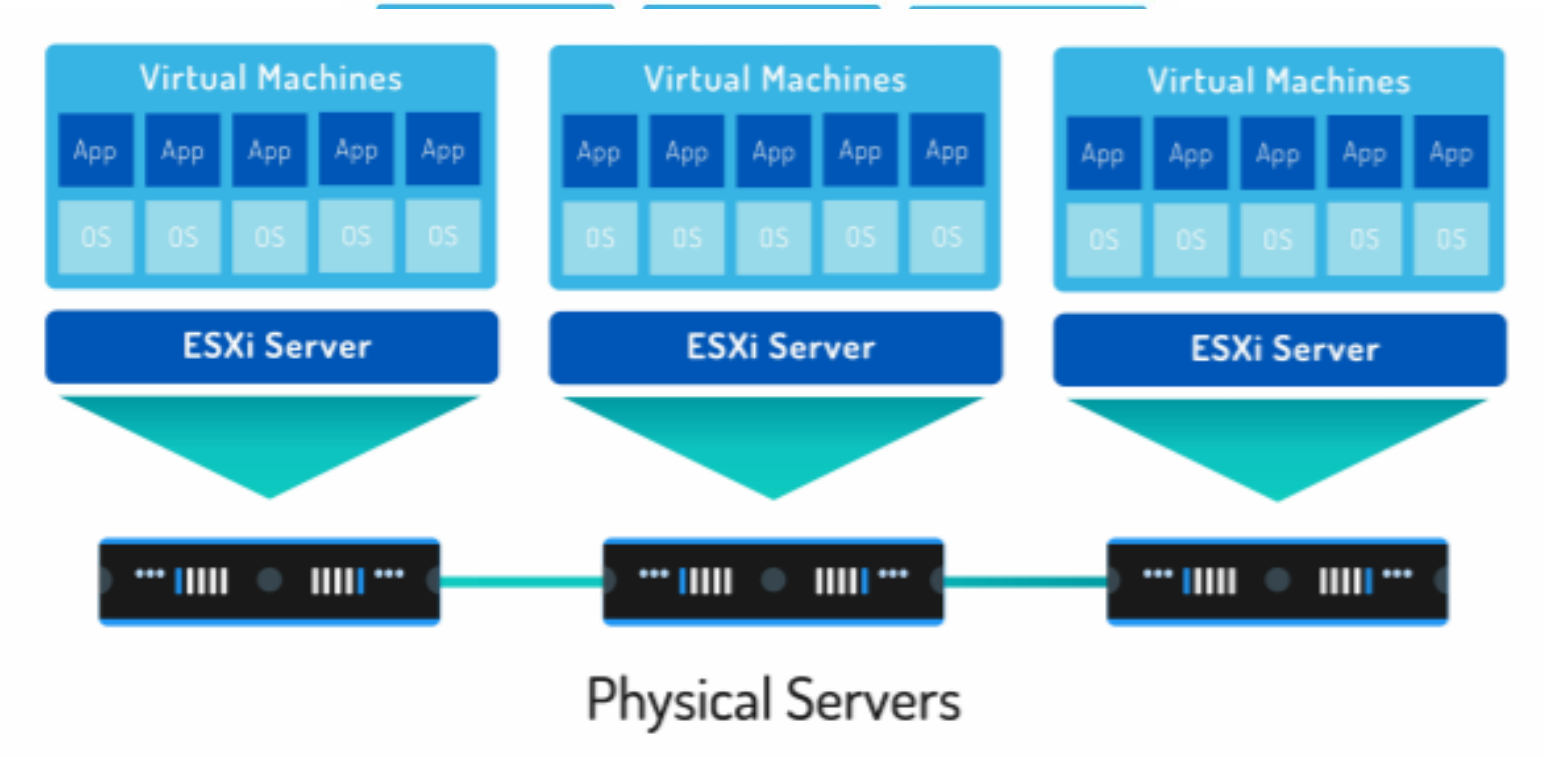
VMware ESX (Elastic Sky X) and ESX i (ESX Integrated)

- Enterprise virtualization platform offered by VMware
- ESX runs on *bare metal*, without an OS
- Monolithic architecture
- Includes own Linux kernel, which is started first and then loads VMware's vmkernel component
- ESXi, an upgrade from ESX, does not contain the Linux kernel. Directly loads from vmkernel
- Vmkernel has three interfaces:
  - Hardware
  - Guest systems
  - Service console (Linux based console operating system)

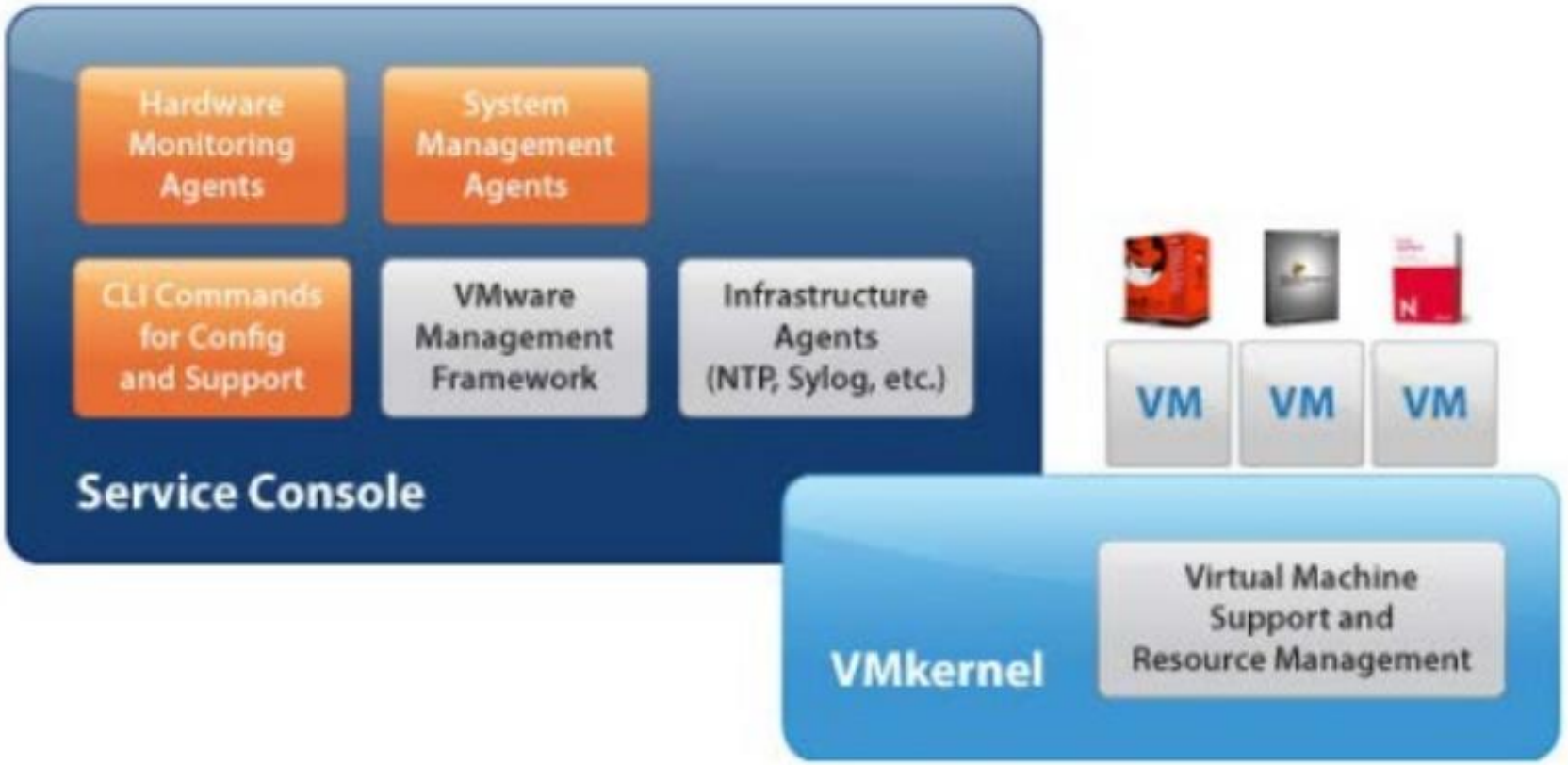
What is ESXi?

- ESXi is VMware's Type 1 hypervisor software installed directly on the physical server and creates the virtual layer

- Components of ESXi:
  - Unix Microkernel
  - VMware Kernel (VMkernel)



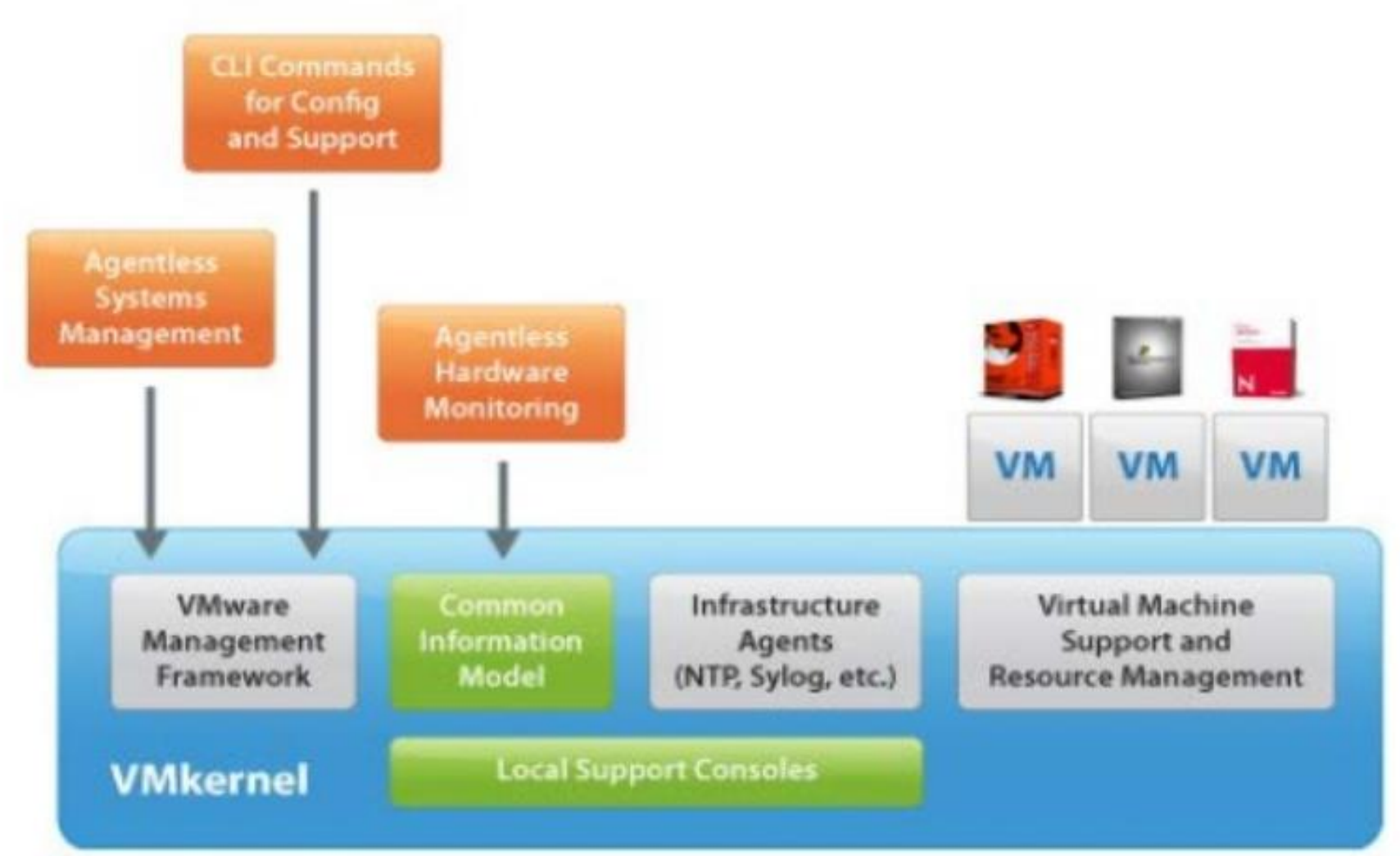
VMware ESX Architecture



VMware ESXi

- ESXi is a smaller footprint version of ESX
- Does not include the ESX Service Console
- All VMware related agents and 3rd party agents run directly on vmkernel
- Ultra-thin architecture, highly reliable, small code-base
- More secure as there is less patching
- Uses Direct Console User Interface (DCUI) for management

VMware ESXi Architecture



ESX and ESXi comparison

CAPABILITY	ESX	ESXI
Service Console	Present	Removed
Troubleshooting performed via	Service Console	ESXi Shell
Secure Syslog	Not Supported	Supported
Management Network	Service Console Interface	VMKernel Interface
Hardware Monitoring	3 rd Party agents installed in Service console	Via CIM Providers
Software patches and updates	Needed as similar to Linux operation system	Few pacthes because of small footprint and more secure
vSphere web Access	Only experimental	Full managenet capability via vSPhere web client
Locked Down Mode	Not present	Present . Lockdown mode prevents remote users to login to the host
Rapid deployment via Auto Deploy	Not supported	Supported
Custom Image creation	Not supported	Supported
VMkernel Network Used for	vMotion, Fault Tolarance, Stoarge Connectivity	Management Network , vMotion, Fault Tolerance, Storage Connectivity, ISCSI port binding

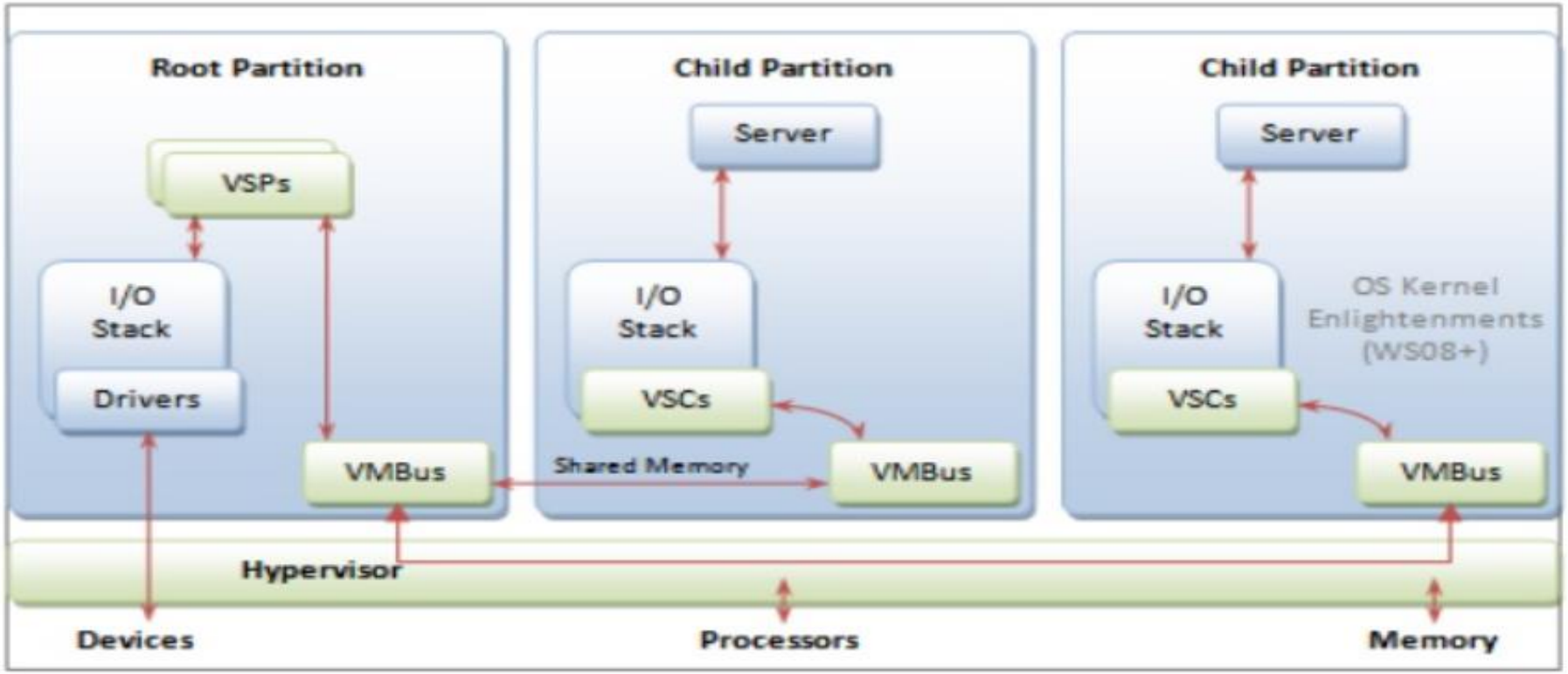
CASE STUDIES : HYPER V

- Introduced in 2008, it is Microsoft's Type 1 Hypervisor for x86-64 systems
- Has a microkernel design
- Hyper-V comes in two variants:
  - **Hyper-V Server**: A stand-alone version
    - Variant of core installation of Windows Server 2008
    - Includes full Hyper-V functionality
    - Mostly CLI for configuration but MMC can be installed for administration
  - An installable role version for Windows Server
    - Installed as a role on Windows Server 2008/2012
    - Configuration and administration done by Remote Desktop or management consoles for much easier graphical control

HYPER V ARCHITECTURE

- Implements virtual machines in terms of partitions
  - Partition is a logical unit of isolation in which an OS is running
- Virtualization layer runs in parent partition giving direct access to hardware resources
- Parent partition creates child partitions for guest OSs
- Compared to VMware ESX/ESXi, all hardware calls go through root or parent partition
- Access to hardware is controlled via root partition though VMBus. This is indirect driver model. Paravirtualized.
- Hyper-V installs on hardware and places original OS in the root partition

HYPER V ARCHITECTURE



Virtualization Service Client (VSC)
Virtualization Service Provider(VSP)

CASE STUDIES : XEN

Xen is a virtualization system supporting both paravirtualization and hardware-assistant full virtualization

Name from ne**X**t g**EN**eration virtualization

Initially created by University of Cambridge Computer Laboratory

Open source (Licensed under GPL2)

VIRTUALIZATION IN XEN

Paravirtualization:

- Uses a modified Linux Kernel (e.g. Linux-2.6.18-xen)
- Guest loads dom0's pygrub or dom0's kernel
- Front-end and back-end virtual device model
- Cannot run windows
- Guest "knows" it's a VM and tells the hypervisor

Hardware-assisted full virtualization:

- Uses the same, normal, OS Kernel
- Guest contains grub and Kernel
- Normal device drivers
- Can run windows
- Guest doesn't "know" it's a VM, so the hardware manages it

WHY TO USE XEN?

Paravirtualization (PV)

- High performance (claim to fame)
- High scalability
- Uses a *modified* Operating System

Hardware-assisted full virtualization (HVM)

- Leading hardware vendors to enhance virtualization in x86 architecture
- Uses an *unmodified* Operating System

Xen is powered by a growing and active community and a diverse range of products and services

Xen offers high performance and secure architecture

CASE STUDIES : XEN

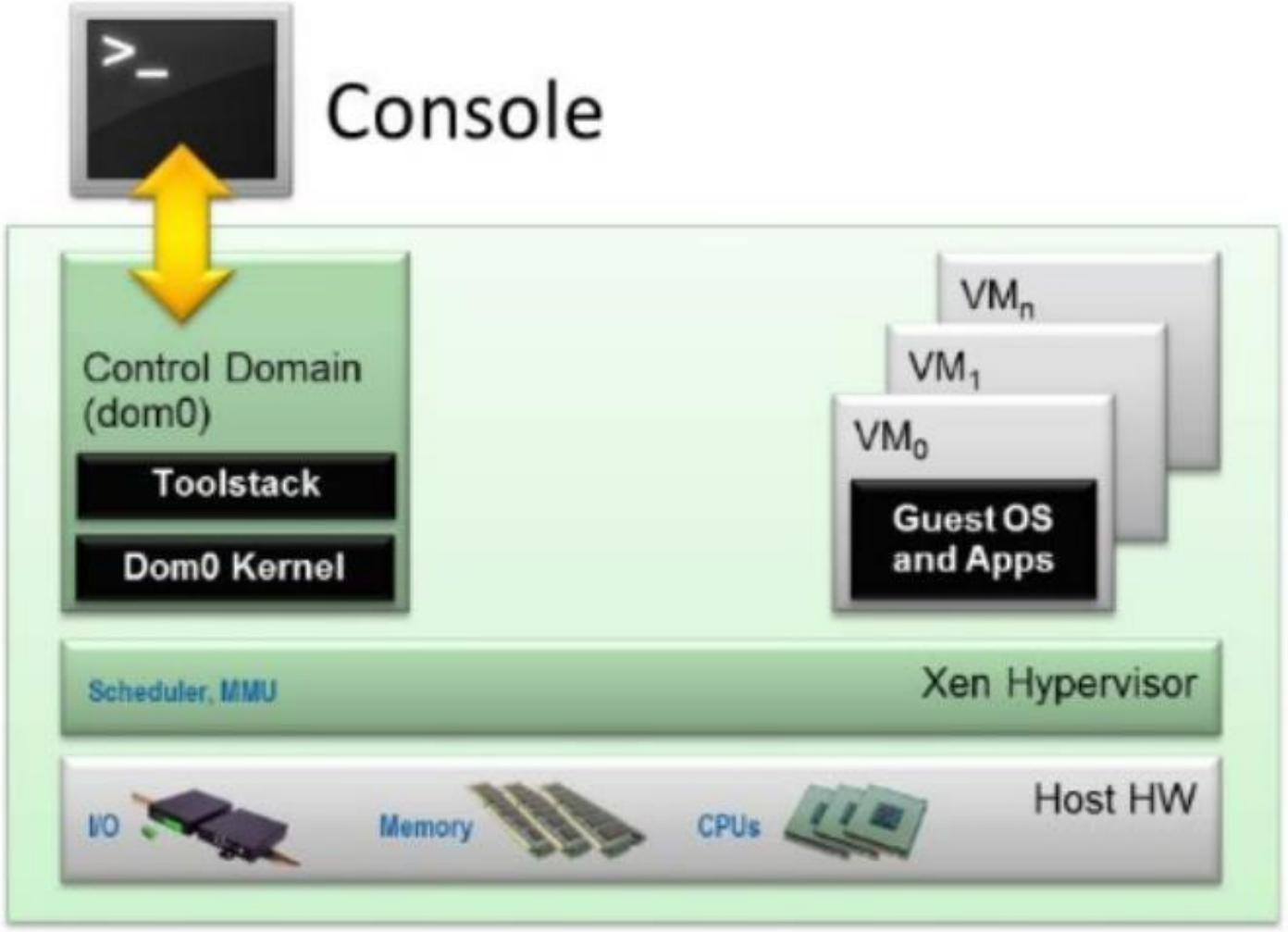
Citrix Systems XenServer

- An open-source Type-1 or bare-metal hypervisor
- Originated as a research project at the University of Cambridge in 2003
  - Commercialize by XenSource Inc.
  - Developed and made available as open-source software under GNU General Public Licence (GPL), version 2
  - Acquired by Citrix in 2007
  - Commercial versions: Citrix XenServer, Oracle VM
- Small footprint and interface; uses a microkernel design
- Supports Paravirtualization
- Supports IA-32, x86-64 and ARM instruction sets.

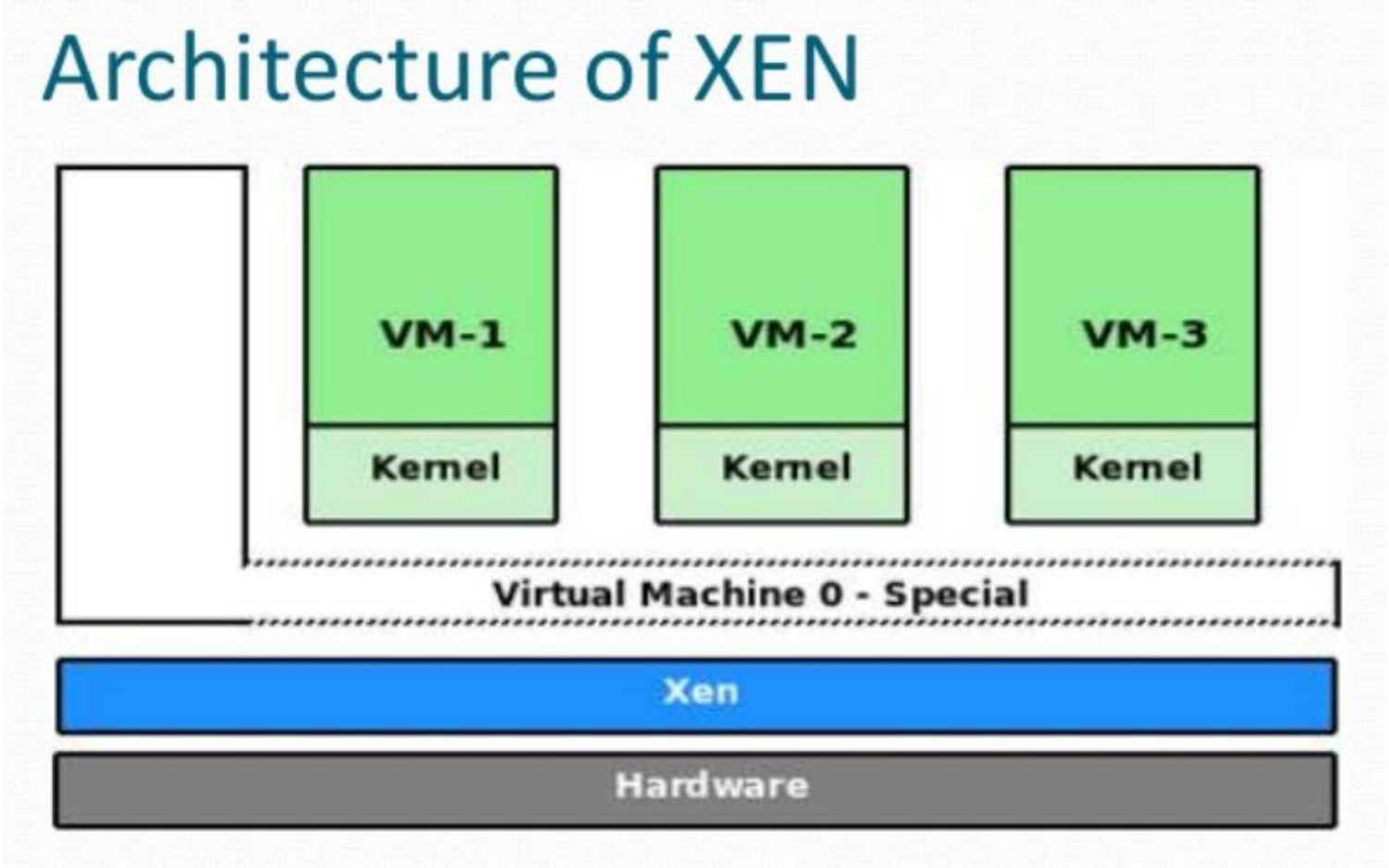
XEN ARCHITECTURE

- Guest types: The hypervisor can run fully virtualized guests, or paravirtualized guests.
- A running instance of a virtual machine is called a **domain** or **guest**.
- Employs a special domain called **domain 0** which contains:
  - Drivers for the hardware
  - Toolstack to control VMs.
  - A control stack to:
    - manage virtual machine creation,
    - destruction, and
    - configuration.

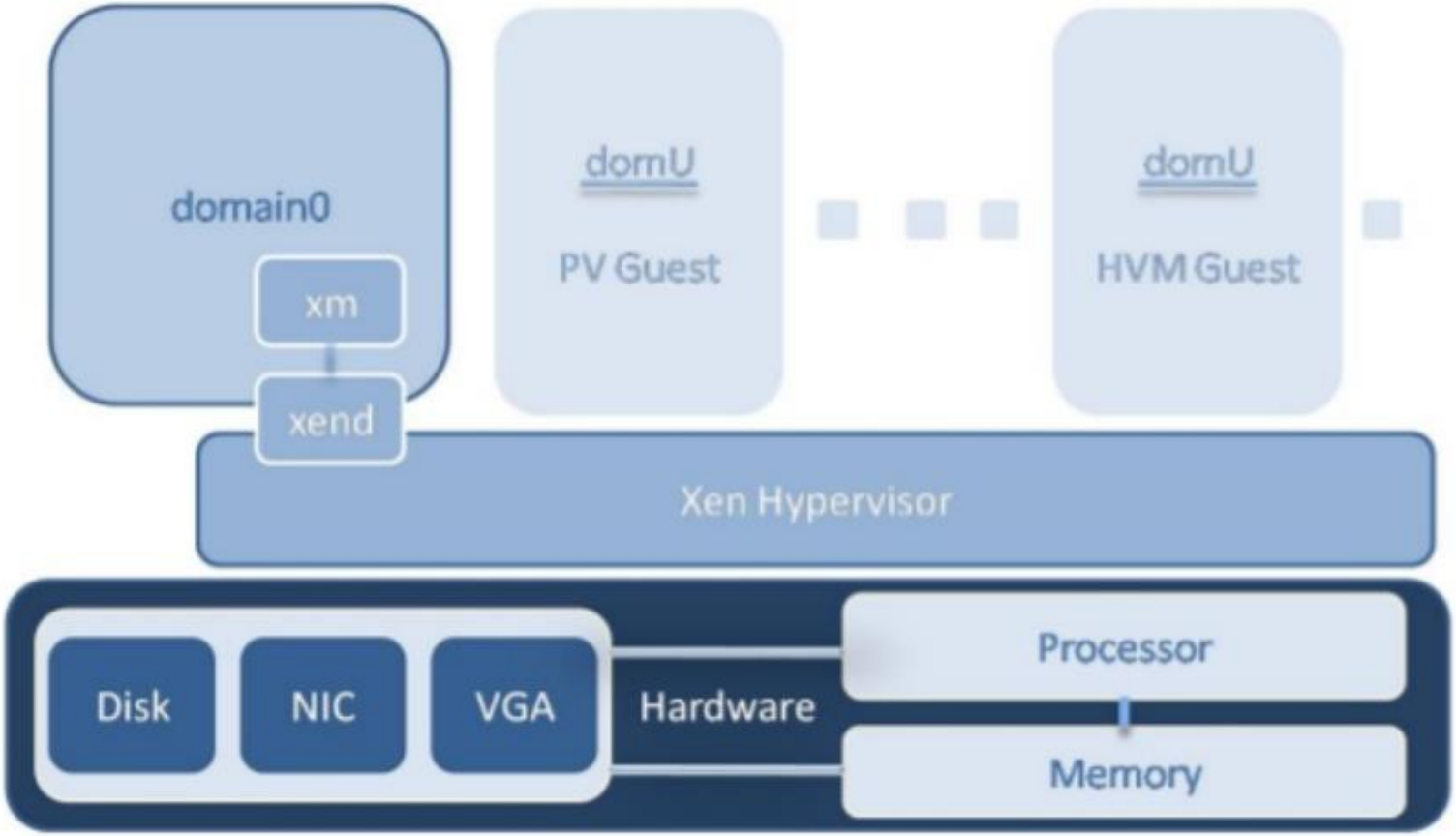
XEN ARCHITECTURE



ARCHITECTURE OF XEN



ARCHITECTURE OF XEN



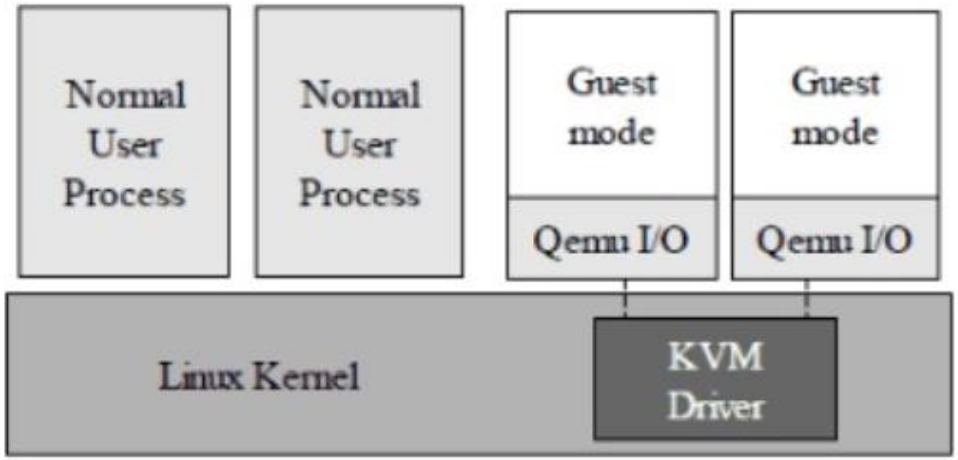
CASE STUDIES : KVM

Kernel-based Virtual Machine (KVM)

- Initially developed by Qumranet later acquired by Redhat in 2008
- Full virtualized solution with a small code base for Linux on x86 hardware with virtualization extensions (Intel VT or AMD-V)
- Designed as small, light kernel module to leverage the facilities provided by hardware support of virtualization
- Originally supported x86 processors but now has been ported to IBM S/390, Intel IA-64
- Free, open source virtualization architecture. Kernel component comes standard in vanilla Linux (2.6.20)

KVM

- KVM hypervisor Type 1 or Type 2: Still up for discussion
- Implemented as a kernel module, allowing Linux to become a hypervisor simply by loading it
- Device appears in /dev/kvm. Allows control by ioctl() system calls to create new VMs, assign memory, etc
- Hardware emulation or platform virtualization controlled by QEMU-k



Virtual Private Servers: Cloud services

- Virtual Private Server (VPS):
  - A virtual machine sold as a service by an Internet hosting service.
  - **Amazon Elastic Compute Cloud (EC2)**
    - Amazon.com's cloud computing platform, Amazon Web Services (AWS)
    - Uses Xen as its hypervisor
  - **Microsoft Azure**
    - Microsoft's cloud application platform.
    - Powered by Microsoft Hyper-V
  - **Google Compute Engine (GCE)**
    - Infrastructure as a Service (IaaS) component of Google Cloud Platform
    - Uses KVM as the hypervisor
  - **Rackspace Cloud** uses VMware
- Used in enterprise IT

CASE STUDIES : VIRTUAL BOX

Virtual Box

- Owned By Oracle Organization
- Free ware
- 3D Virtualization Supports
- Multi Screen Resolution Support
- Support large number of Guest & Host Operating Systems

- VirtualBox is a cross-platform virtualization application.
- It installs on your existing Intel or AMD-based computers.
- Free Ware
- Developed By Oracle Organization
- Supports Large number of guest operating systems

- **Running multiple operating systems simultaneously.**
  VirtualBox allows you to run more than one operating system at a time.
- **Easier software installations.**
  Software vendors can use virtual machines to ship entire software configurations.
- **Testing and disaster recovery.**
  Once installed, a virtual machine and its virtual hard disks can be considered a "container" that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.
- **VirtualBox SnapShot feature.**
  one can save a particular state of a virtual machine and revert back to that state, if necessary.
- **Infrastructure consolidation.**
  Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads.

KEY FEATURES OF VIRTUAL BOX

- ✓ **Multigenerational branched snapshots**
  - VirtualBox can save arbitrary snapshots of the state of the virtual machine.
- ✓ **VM groups**
  - VirtualBox provides a groups feature that enables the user to organize and control virtual machines collectively, as well as individually.
- ✓ **Clean architecture**
  - unprecedented modularity.
- ✓ **Remote machine display**
  - The Virtual Box Remote Desktop Extension (VRDE) allows for high-performance remote access to any running virtual machine.

KEY FEATURES OF VIRTUAL BOX

- ✓ **Portability**
  - VirtualBox runs on a large number of 32-bit and 64-bit host operating systems
- ✓ **No hardware virtualization required**
  - For many scenarios, VirtualBox does not require the processor features built into newer hardware like Intel VT-x or AMD-V.
- ✓ **Guest Additions**
  - shared folders, seamless windows, 3D virtualization.
- ✓ **Great hardware support**
  - Guest multiprocessing (SMP).
  - USB device support.
  - Full ACPI support. The Advanced Configuration and Power Interface (ACPI) is fully supported by VirtualBox.
  - Multiscreen resolutions. VirtualBox virtual machines support screen resolutions many times that of a physical screen.

CASE STUDIES : EUCALYPTUS

What's in a name?

Elastic Utility Computing Architecture Linking Your Programs To Useful Systems

- Eucalyptus is a simple open architecture for implementing cloud functionality at the IaaS level.
- It is specifically designed to be easy to install and maintain in a research setting, *and* that it is easy to modify, instrument, and extend.
- Eucalyptus can be deployed and executed without modification to the underlying infrastructure.
- Eucalyptus components have well defined interfaces (described by WSDL documents), support secure communication (using WS-Security policies), and rely upon industry-standard Web-services software packages (Axis2, Apache, and Rampart).

WHY EUCALYPTUS?

Why Eucalyptus?

"Elastic Utility Computing Architecture Linking Your ***Programs*** To Useful System"

Eucalyptus is the world's most widely deployed software platform for on-premise (private) Infrastructure as a Service (IaaS) clouds.

It uses existing infrastructure to create a scalable, secure web services layer that abstracts compute, network and storage to offer IaaS.

Eucalyptus can be dynamically scaled up or down depending on application workloads.

WHY EUCALYPTUS?

- **Open Source**

  you can download it and have the source code at your fingertips.
- **Modular**

  The Eucalyptus components have well-defined interfaces (via WSDL, since they are web services) and thus can be easily swapped out for custom components.
- **Distributed**

  Eucalyptus allows its components to be installed strategically close to the needed/used resources. For example Walrus can be installed close to the storage, while the Cluster Controller can be installed close to the cluster it will manage.
- **Designed to Perform**

  Eucalyptus was designed from the ground up to be scalable and to achieve optimal performance in diverse environments (designed to overlay an existing infrastructure).

WHY EUCALYPTUS?

- **Flexible**

  Eucalyptus is flexible and can be installed on a very minimal setup. Yet it can be installed on thousands of cores and terabytes of storage. And it can do so as an overlay on top of an existing infrastructure.
- **Compatible**

  Eucalyptus is compatible with the most popular and widely used Cloud API currently available: Amazon EC2 and S3.
- **Hypervisor Agnostic**

  Currently Eucalyptus fully supports KVM and Xen. Additionally, the Enterprise Edition supports the proprietary VMware hypervisor.
- **Hybrid Cloud**

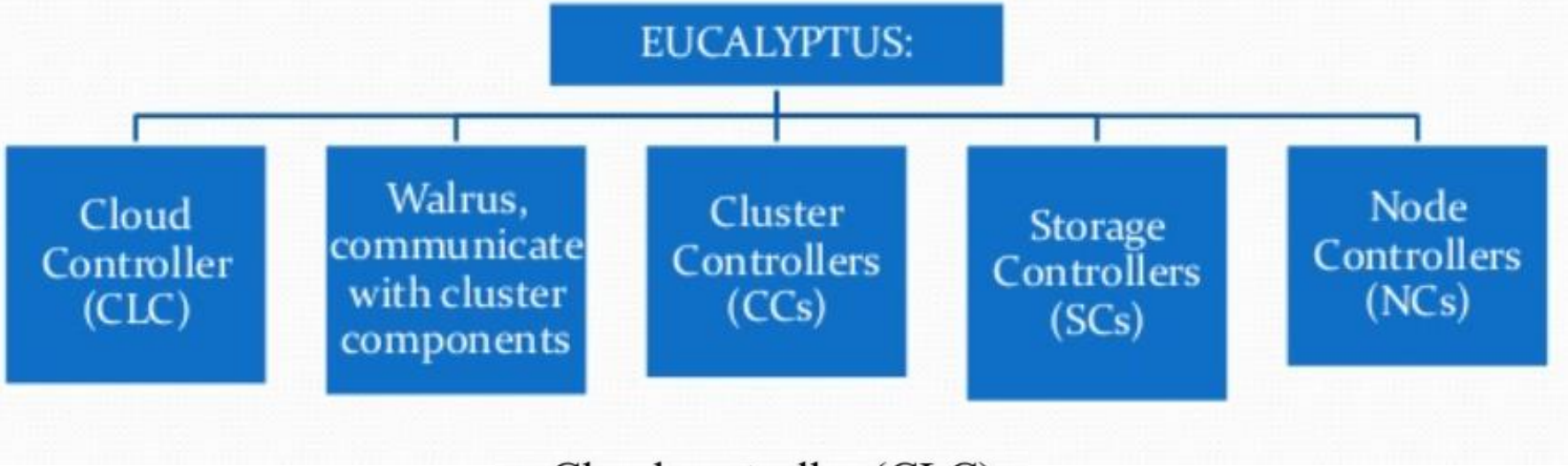
  The above characteristics makes Eucalyptus easy to deploy as an hybrid cloud. An hybrid cloud combines resources drawn from multiple clouds, typically one private and one public.

SECURITY BASED EUCALYPTUS

A Secure Cloud

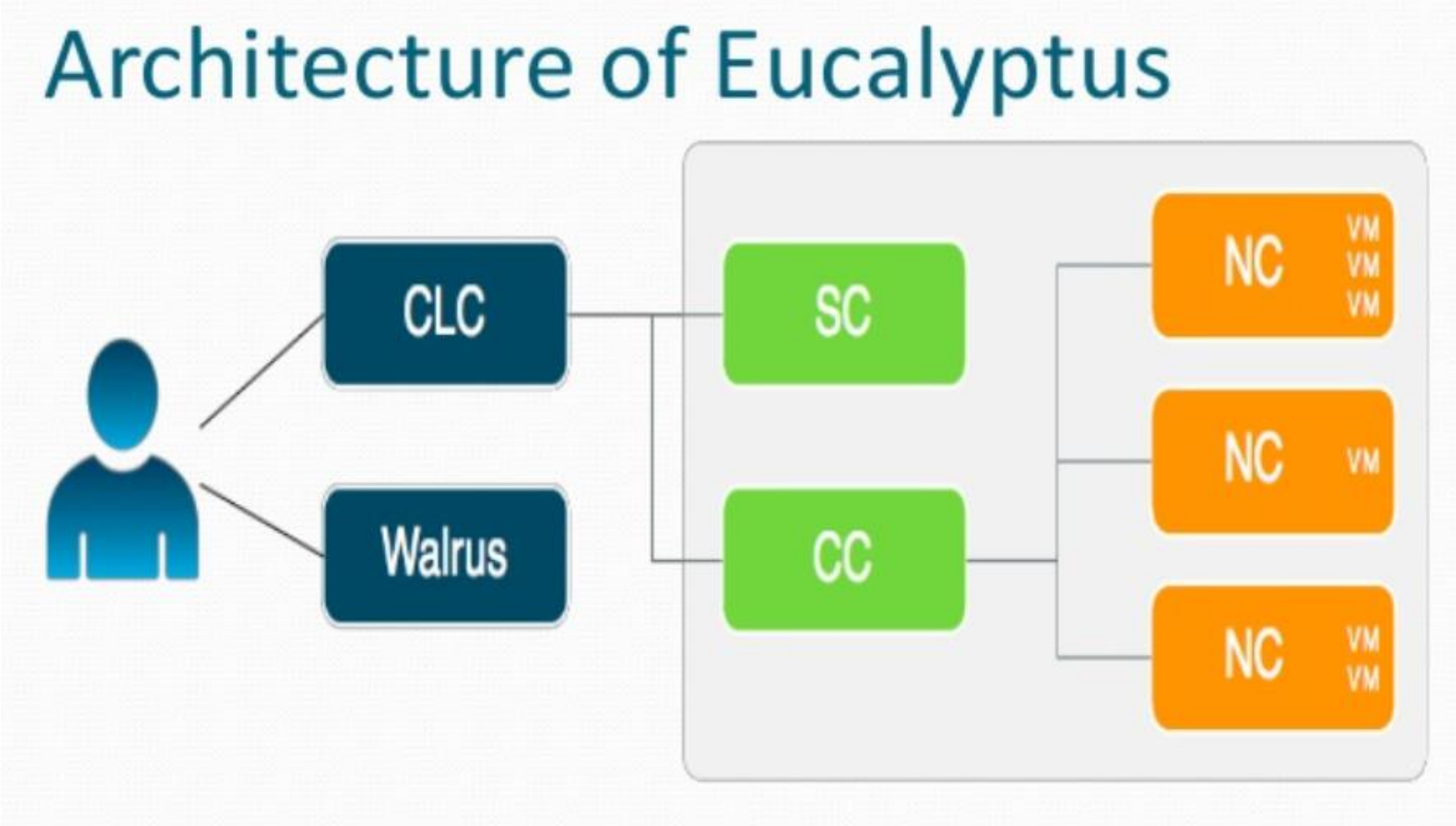
- Eucalyptus is a Linux-based open source software architecture that implements efficiency-enhancing private and hybrid clouds within an enterprise's existing IT Infrastructure.
- A Eucalyptus private cloud is deployed across an enterprise's "on--premise" data center infrastructure and is accessed by users over enterprise intranet. Thus sensitive data remains Entirely secure from external intrusion behind the enterprise firewall.

COMPONENTS OF EUCALYPTUS



- Cloud controller (CLC)
- Warlus
- Storage controller
- Cluster controller
- VMBroker (optional)
- Node controller

ARCHITECTURE OF EUCALYPTUS



COMPONENTS OF EUCALYPTUS

Cloud Controller (CLC)

The Cloud Controller (CLC) is the entry-point into the cloud for administrators, developers, project managers, and end-users.

Functions:

- Monitor the availability of resources on various components of the cloud infrastructure, including hypervisor nodes that are used to actually provision the instances and the cluster controllers that manage the hypervisor nodes
- Resource arbitration { Deciding which clusters will be used for provisioning the instances }
- Monitoring the running instances

COMPONENTS OF EUCALYPTUS

Cluster Controller(CC)

The Cluster Controller (CC) generally executes on a cluster front---end machine, or any machine that has network

- Connectivity to both the nodes running NCs and to the machine running the CLC. CCs gather information about a set of VMs and schedules VM execution on specific NCs. The CC also manages the virtual instance network and participates in the enforcement of
- All nodes served by a single CC must be in the same broadcast domain (Ethernet).

Functions:

- To receive requests from CLC to deploy instances
- To decide which NCs to use for deploying the instances on
- To control the virtual network available to the instances
- To collect information about the NCs registered with it and report it

  to the CLC

Node Controller (NC)

- The Node Controller (NC) is executed on every node that is designated for hosting VM instances.
- NCs control the execution, inspection, and termination of VM instances on the host where it runs, fetches and cleans up local copies of instance images (the kernel, the root file system, and the ramdisk image), and queries and controls the system software on its node (host OS and the hypervisor) in response to queries and control requests from the cluster controller. The Node controller is also responsible for the management of the virtual network endpoint.

Functions:

- Collection of data related to the resource availability and utilization
- on the node and reporting the data to CC
- Instance life cycle management

COMPONENTS OF EUCALYPTUS

VMware Broker

- VMware Broker (Broker or VB) is an optional Eucalyptus component, which is available if you are a Eucalyptus Subscriber.
- VMware Broker enables Eucalyptus to deploy virtual machines (VMs) on VMware infrastructure elements. VMware Broker mediates all interactions between the CC and VMware hypervisors (ESX/ESXi) either directly or through VMware vCenter.

Storage Controller

- The Storage Controller (SC) provides functionality similar to the Amazon Elastic Block Store (Amazon EBS). The SC is capable of interfacing with various storage systems (NFS, iSCSI, SAN devices, etc.).
- Elastic block storage exports storage volumes that can be attached by a VM and mounted or accessed as a raw block device

Walrus

- Walrus allows users to store persistent data, organized as buckets and objects. You can use Walrus to create, delete, and list buckets, or to put, get, and delete objects, or to set access control policies.
- Walrus is interface compatible with Amazon's Simple Storage Service (S3), providing a mechanism for storing and accessing virtual machine images and user data

BENEFITS OF EUCALYPTUS

Benefits of The Eucalyptus

- Scalable data center infrastructure. Eucalyptus clouds are highly scalable, which enables an organization to efficiently scale-up or scale-down data center resources according to the needs of the enterprise.
- Elastic resource configuration. The elasticity of a Eucalyptus cloud allows users to flexibly reconfigure computing resources as requirements change. This helps the enterprise workforce remain adaptable to sudden changes in business needs.
- Open source innovation. Highly transparent and extensible, Eucalyptus' open source core architecture remains entirely open and available for value- adding customizations and innovations provided by the open source development community. The Eucalyptus open source software core is available for free download at www.eucalyptus.com.

ADVANTAGES OF HYPERVISOR

- Consolidation of hardware resources
- Ease of administration
  - Centralization of resources
- Significant cost savings
  - Less hardware
  - Energy and power savings
- Fault tolerance through clustering
- Ease of deployment and management
- Better use from hardware
  - Virtualization enables higher utilization rates of hardware because each server supports enough virtual machines to increase its utilization from the typical 15% to as much as 80%.

ADVANTAGES OF HYPERVISOR

- Scalability
  - Additional processing power, network bandwidth, and storage capacity can be accomplished quickly and easily by apportioning additional available resources from the host to the guest VM.
- High availability
  - Mitigating downtime of VMs by dynamically allocating more resources to a guest as needed, migrating VMs between different hosts with zero downtime, fault tolerance, and backing up snapshots of the running systems state.
- Software installation made easy
  - Software vendors can deliver their products preinstalled in VMs (also known as ***virtual appliances***), much of the traditional installation and configuration work associated with software will disappear.

DISADVANTAGES

- Due to the demands of server consolidation, VMs tend to consume more CPU and memory, and require greater disk I/O bandwidth than physical servers with comparable computing loads.

OS Virtualization

- A virtual machine (VM) is a software-defined computer with its own operating system that runs on a host server with a different underlying operating system.
- Virtual hosts are able to share resources between multiple guests, or virtual machines, each with their own operating system instance.

Basic types of virtual machines

- process VM
- system VMs.

- A **process virtual machine** allows you to run a single process as an application on a host machine. An example of a process virtual machine is the Java Virtual Machine (JVM) which allows any system to run Java applications as if they were native to the system.
- A **system virtual machine** is a fully virtualized VM designed to be a substitute for a physical machine. It runs on a different host machine by utilizing a hypervisor such as VMware ESXi to access the underlying machine's resources.

Server virtualization

- When it comes to servers, a single host can potentially run hundreds of virtual machines, each with independent operating systems. Keeping operating systems separate means that one host can support both Linux and Windows servers simultaneously.
- A virtual host can be configured to use thin provisioning as well, which means virtual machines only use resources they absolutely need at that time. This allows administrators to optimize resource allocation and reduce the amount of hardware that a business needs to own.

How a Virtual Machine Works

- Virtual machine architecture can be separated into four different components, listed from the bottom up:
- An underlying system which includes the physical machine and its operating system. Bare metal hypervisors do not require an underlying operating system at this layer.
- A hypervisor which acts as a communication and translation layer.
- Multiple virtual machines that use the host's resources by communicating with the hypervisor.
- Applications and processes that run on each guest's operating system.

Hypervisor configuration

- The hypervisor needs to be appropriately configured before deploying any virtual machines. Using KVM, an open source virtualization technology built into Linux, administrators can create virtual machines from a command line interface.

Summary on Virtual Machines

- Virtual machines reduce the amount of physical hardware organizations have to purchase and allow them to run multiple operating systems on one underlying host.
- Running multiple virtual machines on a physical host can optimize the use of system resources.
- Testing applications in different environments can be done quickly and cost-free by spinning up and decommissioning virtual machines.

References:

- Windows Azure: https://www.tutorialspoint.com/microsoft_azure/microsoft_azure_windows.htm
- Windows Azure: https://en.wikipedia.org/wiki/Microsoft_Azure
- Disaster Recovery: https://www.nakivo.com/blog/disaster-recovery-in-cloud-computing/
- VM:https://blog.stackpath.com/virtual-machine/