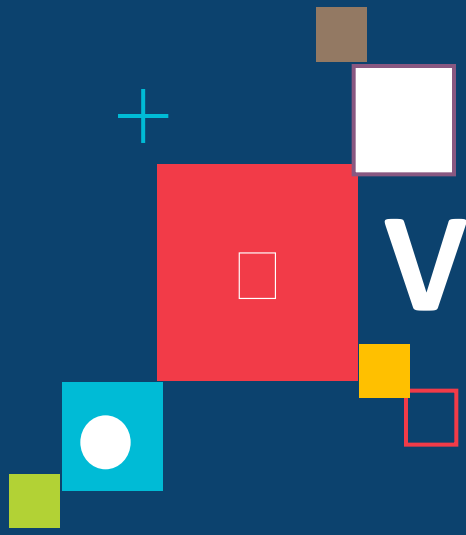


VIRTUALIZATION

Learning Outcome

This module offers you a detailed understanding of

- Virtualization Technologies and its Terminologies
- Approaches and the Models
- Virtualization types - Server, Storage, Network, Application and Desktop
- Working knowledge on the Virtualization Tools like KVM, Virtual Box and VMware

Agenda

- Façade
- Evolution
- Concepts
- Approaches
- Models
- Types
- Tools

Abbreviations/Acronyms

OS	Operating System
VM	Virtual Machine
VMM	Virtual Machine Monitor
QEMU	Quick Emulator
CPU	Central Processing Unit
RAM	Random Access Memory
NIC	Network Interface Card
VT-x	Virtualization Technology for x86 architecture
VT-i	Virtualization Technology for Itanium architecture
VT-d	Virtualization Technology for directed I/O
VT-c	Virtualization Technology for connectivity
VMX	Virtual Machine Extension

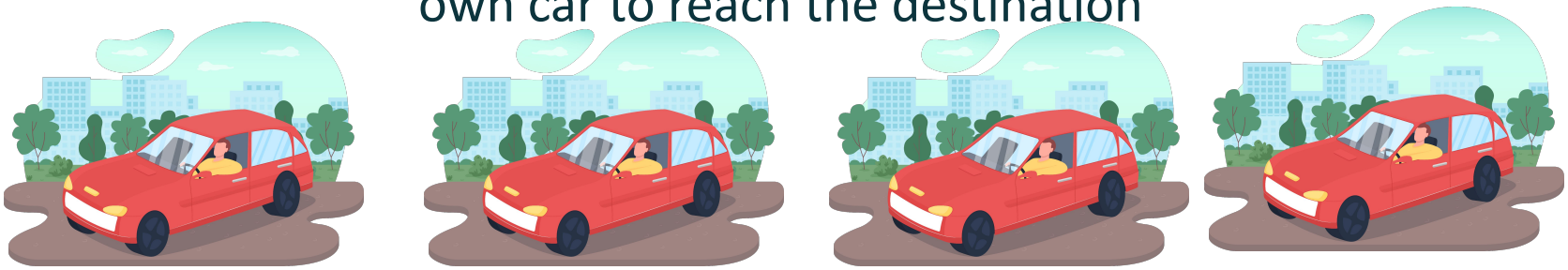
Abbreviations/Acronyms

MMU	Memory Management Unit
I/O	Input/Output
EPT	Extended Page Table
DMA	Direct Memory Access
SAN	Storage Area Network
PCI	Peripheral Component Interconnect
SR-IOV	Single Root I/O Virtualization
LXC	Linux Containers
KVM	Kernel Virtual Machine
RHEV	Red Hat Enterprise Virtualization
VMware ESXi	VMware Elastic Sky X (i integrated)
KSM	Kernel Same Page Merging

Faćade

Façade

Traditional Architecture is like every individual persons driving their own car to reach the destination



Virtualization is like the Public Transportation Authority, where the physical host is a suburban train and virtual machines are the passengers of this train

Façade

Effective Utilization of the Living Space

Single House

Multiple houses in an apartment

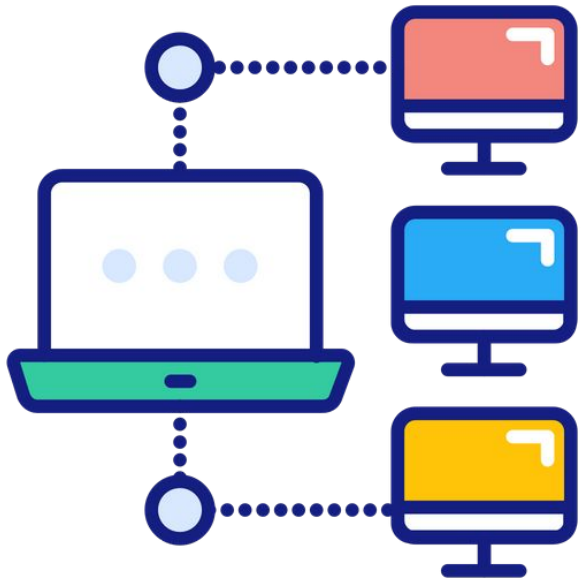
Façade

Definition

Virtualization is the process of running a virtual instance of a computer system, in a layer abstracted from the actual hardware.

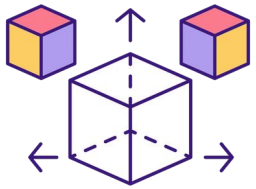
Virtualization is a technology that allows

- to create multiple emulated environments.
- dedicated resources from a single, physical hardware system.



Source: https://ieeexplore.ieee.org/document/6488669
https://opensource.com/resources/virtualization

Emulation vs Virtualization



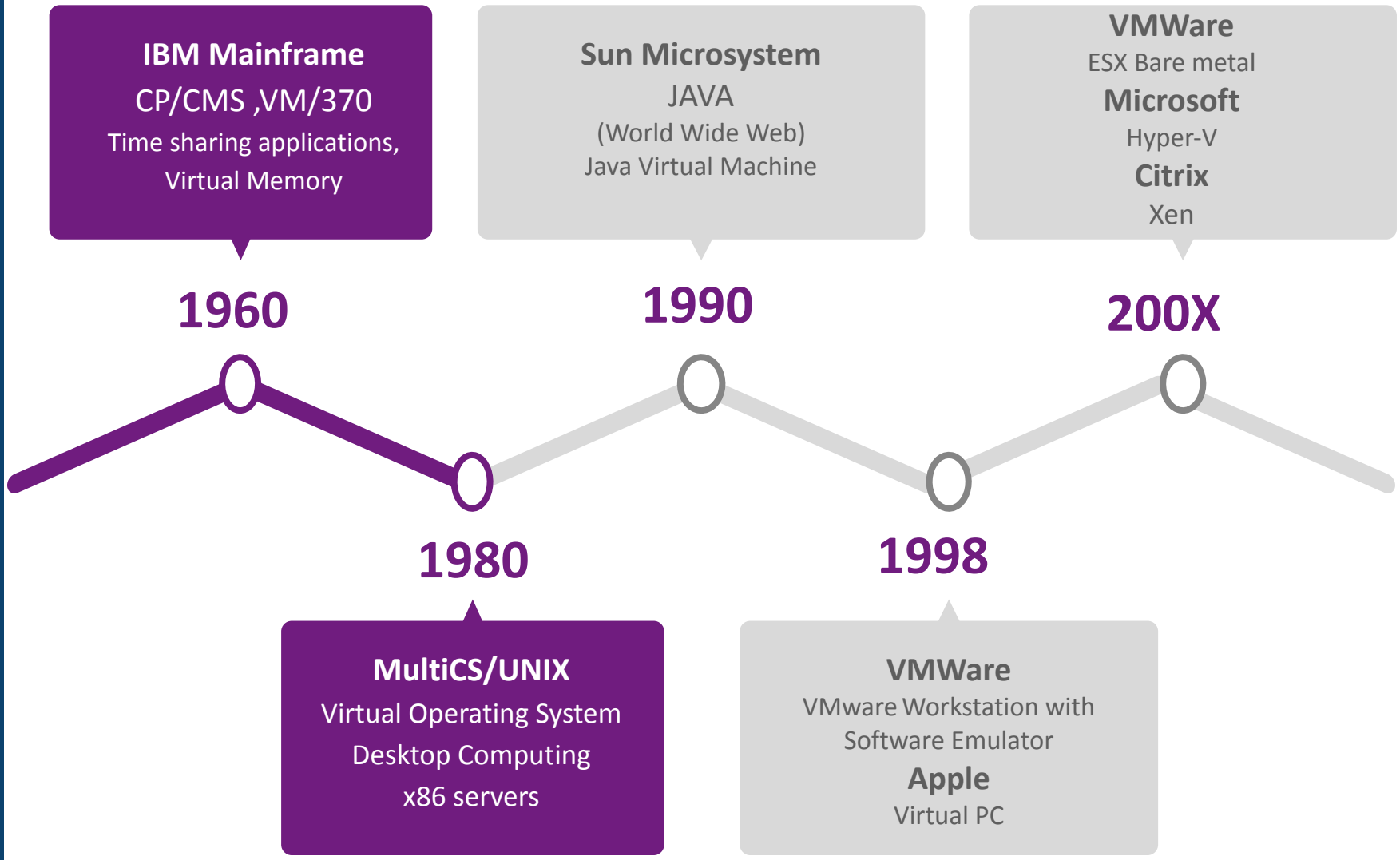
Emulation

Virtualization

Emulation	Virtualization
It is a working simulator of any real device.	It is a process of creating virtual version of operating systems, servers, storage and network devices.
It requires software connector to access hardware.	It accesses hardware directly without any need of software connector.
It is relatively slower.	It is relatively faster.
It provides less effective backup solutions	It provides effective and better backup solutions

EVOLUTION

The Journey



CP - Control Program
CMS - Console Monitor System(interactive OS)

Source : https://ieeexplore.ieee.org/document/6488669/
https://www.idkrtm.com/history-of-virtualization/

EVOLUTION

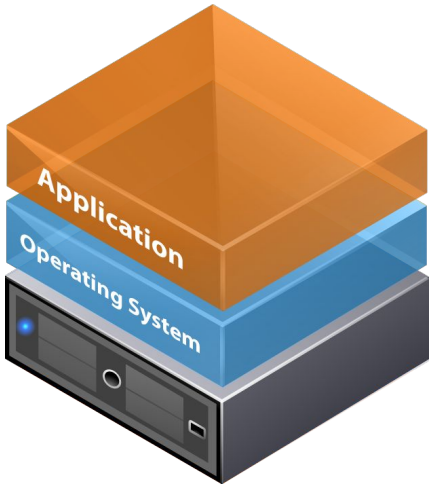
VIRTUALIZATION CONCEPTS

Virtualization (contd.)

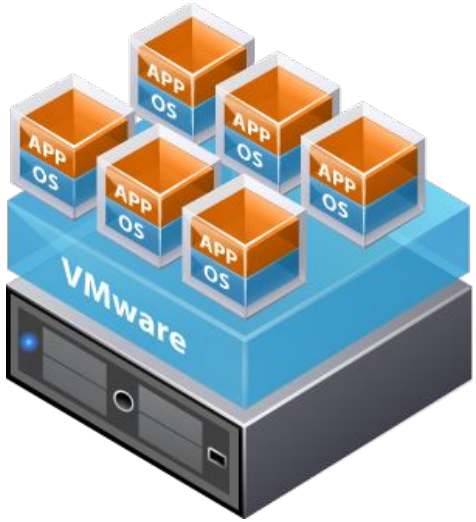
Virtualization has ability to allow

- single physical resource serve as multiple virtual resources
- multiple physical resources function as a single virtual resource.

Traditional Architecture



Virtual Architecture



Definition Source :https://ieeexplore.ieee.org/document/7570950/definitions?anchor=definitions
Image Source: https://docs.hol.vmware.com/HOL-2019/hol-1910-01-sdc_html_en/

Why Virtualization?

Near-complete resources utilization

Running Heterogeneous environments

Manageability

Application Isolation

Reduced power requirement

Reduced ownership cost

Virtualization Impact on a DataCenter

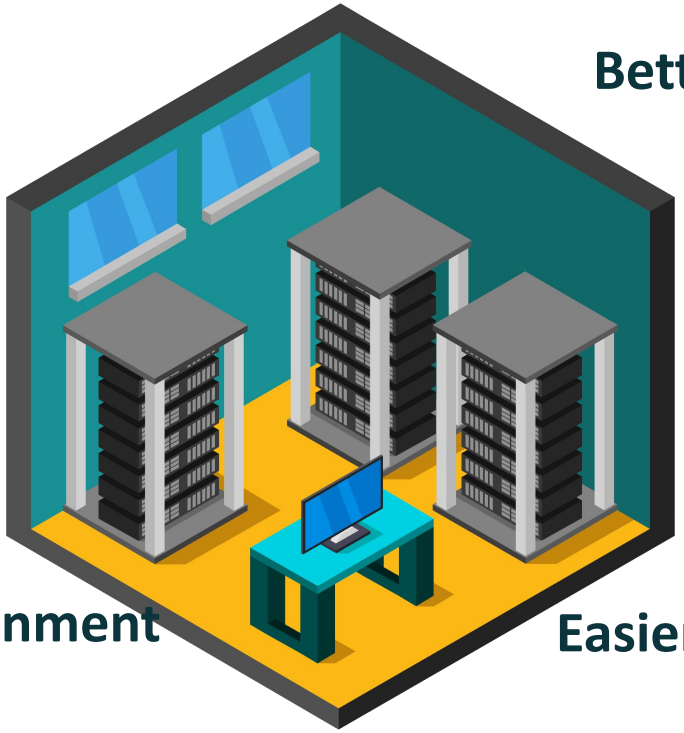
Less Heat Buildup

Reduced Cost

Faster Redeploy

Better Testing Environment

Easier Backups /Snapshots



Better Disaster Recovery

Application Isolation

No Vendor Lock-in

Easier Migration to Cloud

Green Pastures

Virtualization Terminologies

Hypervisor/ VMM	It is a software program that allows multiple operating systems(VMs) to share a single hardware host.
Virtual Machine	A virtual machine (VM) is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system.
Host OS	The physical server equipped with hypervisor like KVM, Hyper-V and VMware that hosts the VM.
Guest OS	It is an operating system that runs under the control of a VMM rather than directly on the hardware.
System calls	System calls are programs that request services from the kernel of the OS. It provides the interface between the application/ process and the OS It is a software trap from an application to the kernel.
System Trap	Exception or Fault that switches to kernel mode by invoking any system call
High Availability	It is a component of a technology system that eliminates single points of failure to ensure continuous operations or uptime for an extended period.

Virtualization Terminologies

Hypercalls	It is a software trap from a domain to the hypervisor whereas the domain is a privileged Linux host os to access the hardware
Kernel mode	In this mode the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address
User mode	In this mode, the executing code has no ability to directly access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory.
Privileged instructions	An instruction (usually in machine code) that can be executed only by the operating system in a specific mode. eg).Set value of timer, clear memory, turn off interrupts,modify entries in device-status table, access I/O device
Sensitive instructions	Control-sensitive instructions seek to change the configuration of resources. Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

Virtualization Terminologies

VM Cluster	Effective technique to ensure high availability of servers and network to protect resources from failure
Memory Management Unit(MMU)	an I/O memory management feature that remaps I/O DMA transfers and device interrupts
Direct Memory Access(DMA)	It is a feature of computer systems that allows certain hardware subsystems to access main system RAM independently of the CPU.
I/O Virtualization	It is a software technology that abstracts the upper-layer protocols from physical connections or physical transports
Digital Workspace	A digital workspace is an integrated technology framework designed to deliver and manage app, data, and desktop delivery.
Microkernel	A Kernel type that provides low-level address space, thread management and interprocess communication to implement OS
Monolithic kernel	A kernel type where the entire OS works in the kernel space.

Virtualization Terminologies

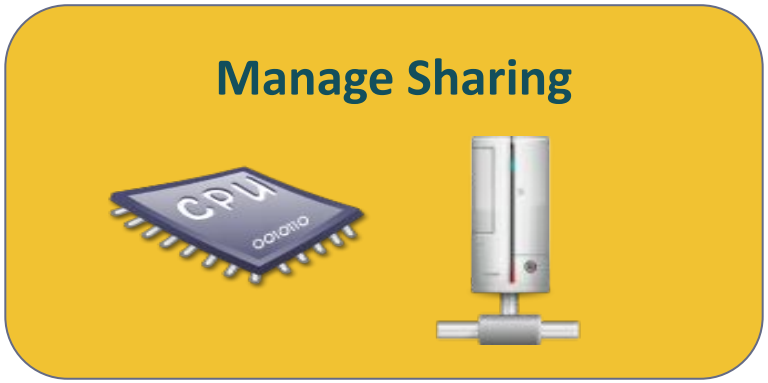
VirtIO	A virtualization standard for network and disk device drivers aware of being virtualized. It is an abstraction layer over paravirtualized hypervisor
Kernel Same-page merging(KSM)	It enables KVM guests to share identical memory pages. These shared pages are usually common libraries or other identical, high-use data. KSM avoids memory duplication.
Machine emulator	QEMU uses Binary translation for Guest OS
Virtualizer	QEMU achieves near native performance by executing the guest code directly on the host CPU like in KVM and Xen

Source: https://www.sciencedirect.com/topics/computer-science/paravirtualization
https://www.sciencedirect.com/topics/computer-science/virtual-machine-monitor
https://en.wikipedia.org/wiki/High_availability

Virtualization Components

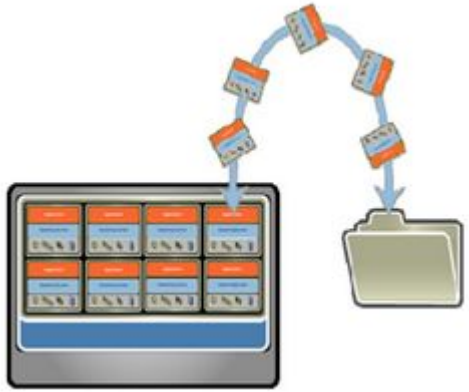
CPU and NIC resources of the host servers are shared with the VMs.

RAM and Storage resources of the host servers are mapped to the VMs.

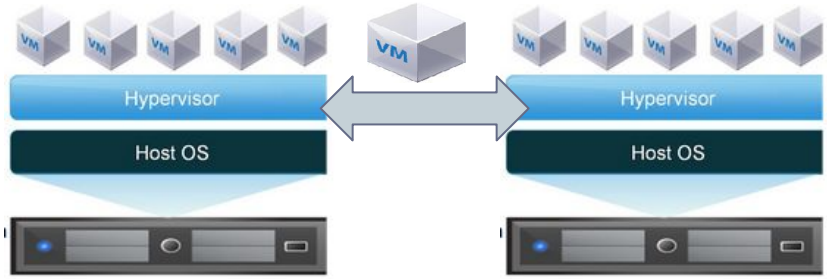


Virtualization Key Features

Partitioning



Encapsulation



Hardware Independent

Isolation

Virtualization Key Feature - Partitioning

Multiple VMs with either same or heterogenous OSs running on a single physical server.

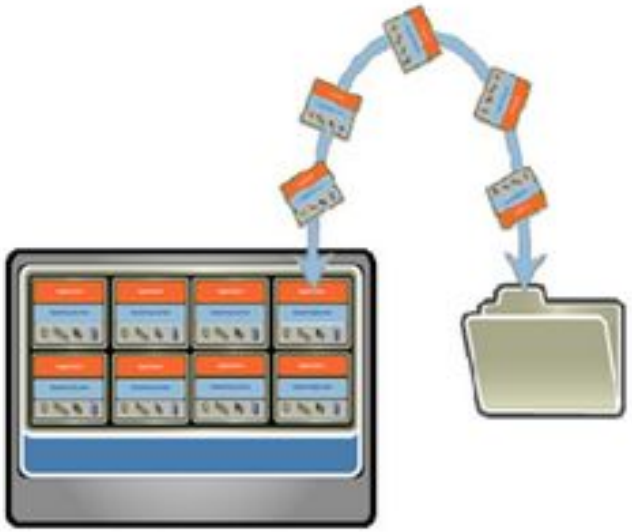
Utilize the physical resources optimally.

Dynamic provisioning/deprovisioning of resources to the VM.

Clustering of VMs to provide High Availability.

Virtualization Key Features - Encapsulation

VM Encapsulation means the entire VM is managed as a file with OS, applications and data within it.



VMs can be captured on the fly as periodic snapshots/clones and restores to point-in-time.

Snapshots help in rapid system provisioning, backup and remote mirroring.

It also offers easy content distribution as pre-configured apps and virtual appliances

Virtualization Key Features - Isolation

Mandate for a secure and reliable environment.

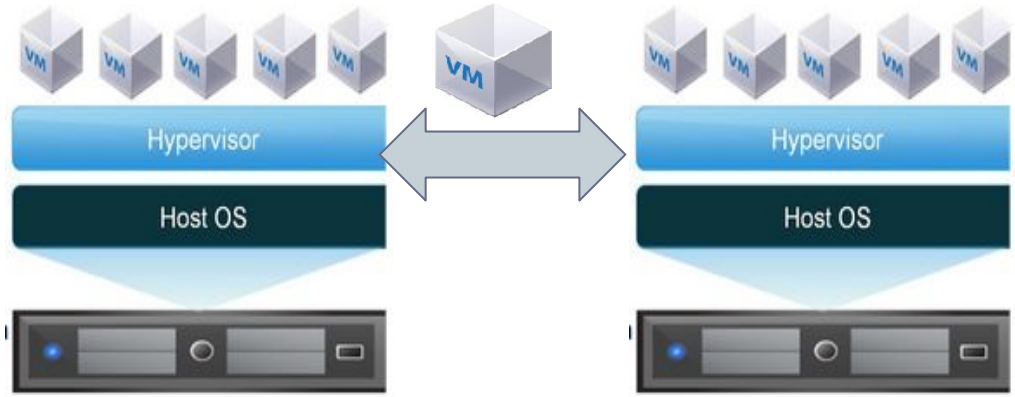
Through hardware abstraction,

- each VM is sufficiently separated
- independent from the operations and activities of other VMs.

Faults in a VM are contained within it which ensures high levels of security and availability of other VMs

Virtualization Key Features - Hardware Independent

The VM once created can run anywhere and can be migrated from one host to the other.



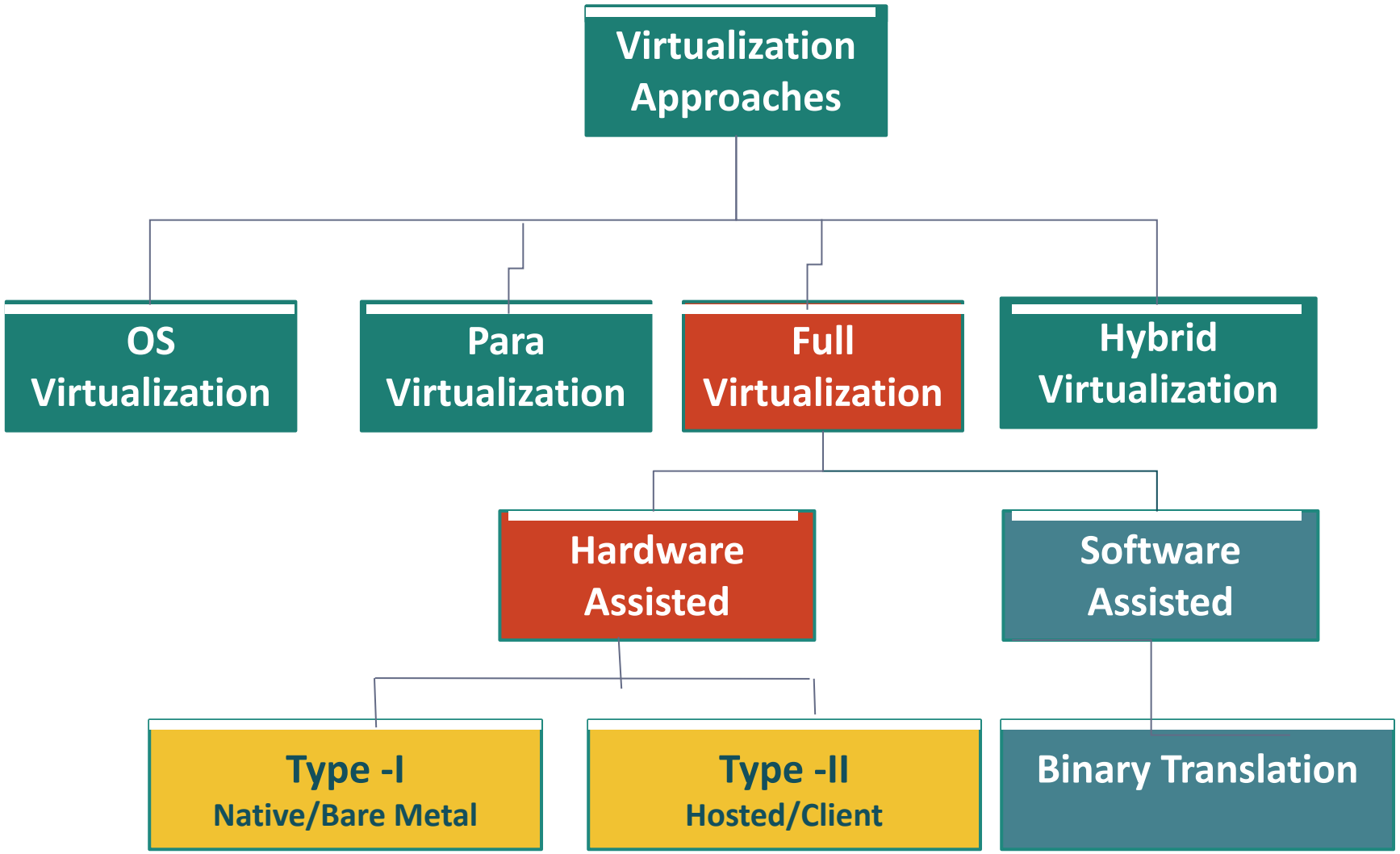
The underlying physical hardware layer is hidden by the virtualization layer and hence it becomes hardware independent.

Run any heterogenous OS on any server without modification.

No Vendor lock-in.

VIRTUALIZATION APPROACHES

Server Virtualization



OS LEVEL VIRTUALIZATION

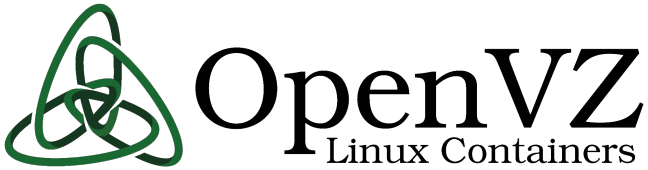
OS Level Virtualization

Kernel creates multiple isolated instances called containers.

A Container is a standard unit of software that packages up code and all its dependencies to run an application reliably in any computing environment

Tools include

- LXC* Containers
- Docker*
- Zones (Solaris)
- Virtual Private Servers (OpenVZ)
- Jails (chroot, FreeBSD)



*Trademarks/Logos are owned by their respective owners

OS Level Virtualization

Container is a lightweight, standalone, executable package of software that comprises of

- Code
- Runtime
- System tools
- System libraries
- Settings

They share the host's OS kernel, supporting programs and libraries but isolated from the host's user mode environment

PARAVIRTUALIZATION

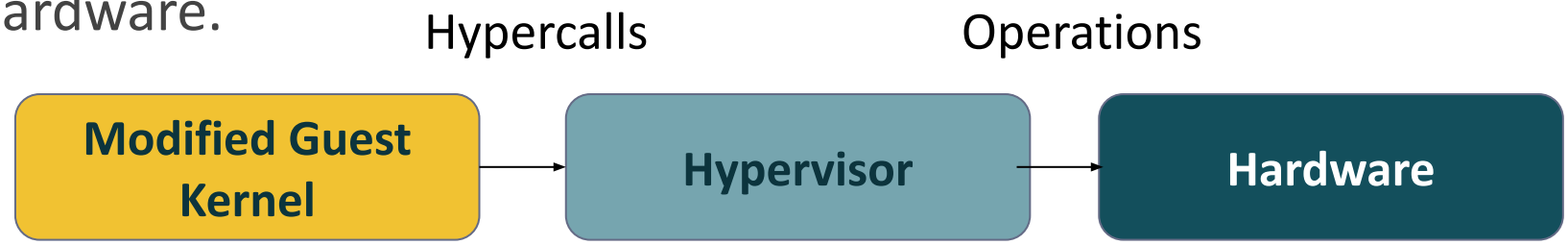
ParaVirtualization

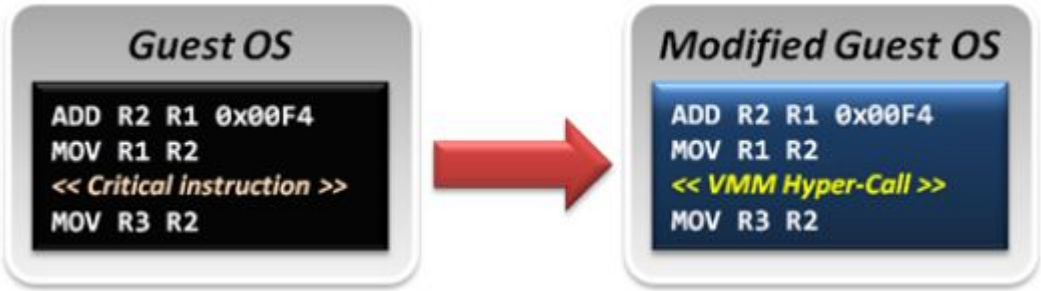
Guest OS source code is modified to use only instructions that can be virtualized.(ie. skip critical instructions)

It replaces non virtualizable instructions with hypercalls that communicate directly with hypervisor

Guest OSs are aware within the system to share resources.

It abstracts the base architecture but does not simulate the hardware.





FULL VIRTUALIZATION

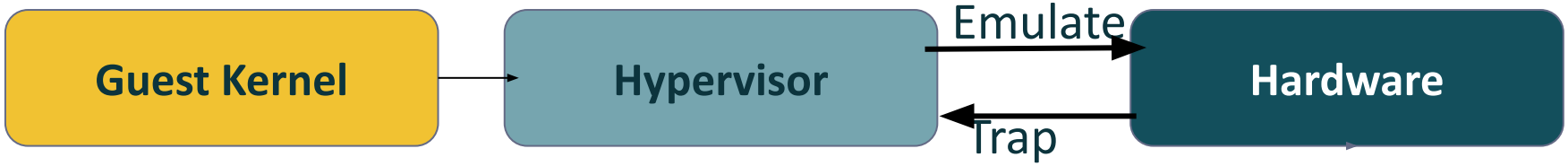
Full Virtualization

The hardware abstraction is an exact replica of the physical hardware.

The VMM simulates hardware to allow an unmodified guest OS[Windows] to run in isolation.

It provides each VM with all the services of the physical system

- Virtual BIOS
- Virtual devices
- Virtualized memory management



Software Assisted Full Virtualization

Software Assisted
Binary Translation
Direct Execution

Binary translation is a technique where the VMM

- emulates an instruction set by translating binary code.
- uses a binary translator
  - replaces the sensitive instructions by equivalent non-sensitive instructions at run-time.
  - leaves non-sensitive instructions unchanged .

eg) VMware Workstation*, VirtualBox*

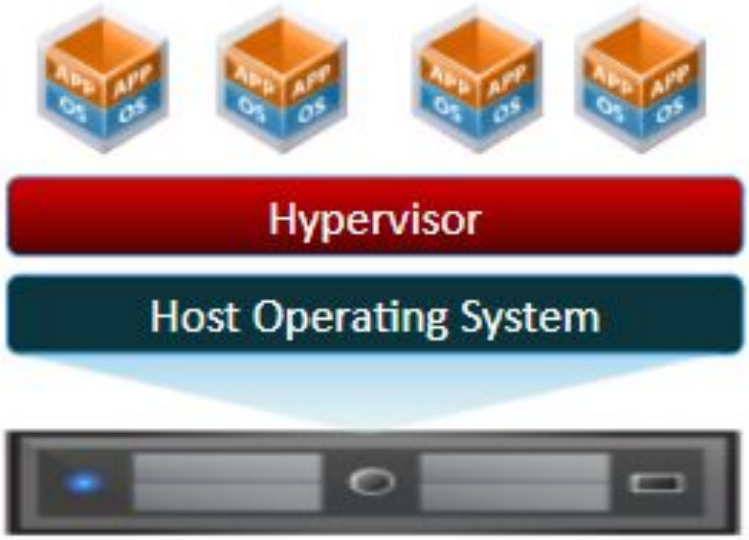
*Trademarks/Logos are owned by their respective owners

Source: https://www.sciencedirect.com/topics/computer-science/binary-translation

Hardware Assisted Full Virtualization Types

Type-I
Bare Metal/Native

Type-II
Hosted/Client



Hardware Assisted Full Virtualization - Hypervisor Type-I

Type-I
Bare Metal/Native

- The hypervisor runs directly on the bare metal server.
- It offers near native performance.

Open source Bare-metal Hypervisors
KVM, Xen, Proxmox

Commercial Hypervisors
RHEV, Citrix XenServer, Hyper-V, VMware ESXi

Hardware Assisted Full Virtualization - Hypervisor Type-I

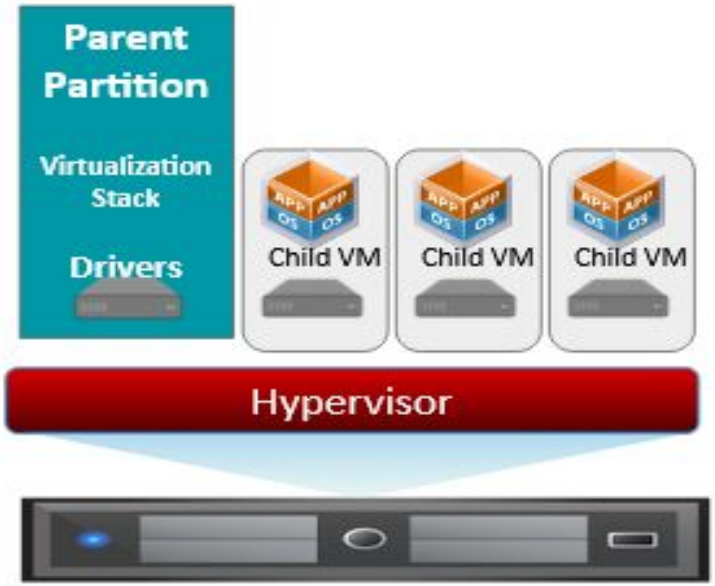
Classification in Type-I Hypervisor Solution

Monolithic

Hypervisor hosted in a single layer that includes the kernel, device drivers and I/O stack

Microkernel

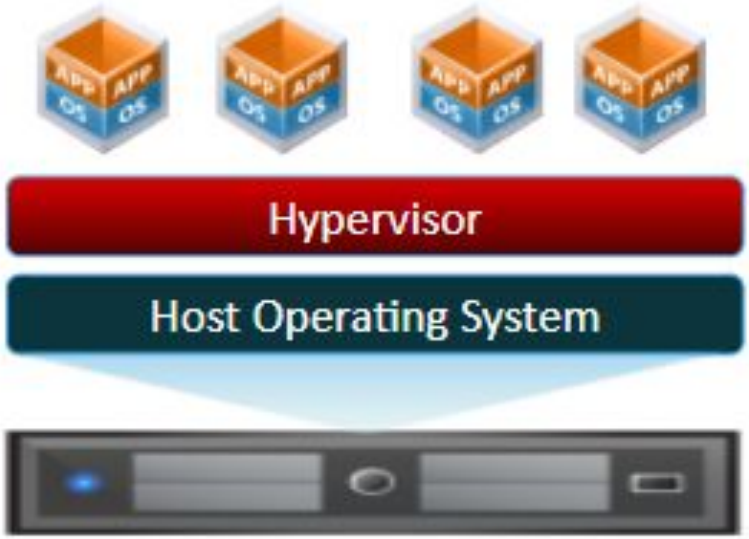
Hypervisor hosted as a very thin,specialized layer that performs only partition isolation and memory management
Does not include any device drivers and I/O stack



Hardware Assisted Full Virtualization - Hypervisor Type-II

- The hypervisor runs on top of the Host operating system.
- They are usually termed as end-user virtualization running VMs with operating systems, application and are completely independent

Type-II
Hosted/Client



Open source hosted hypervisors
QEMU, VirtualBox

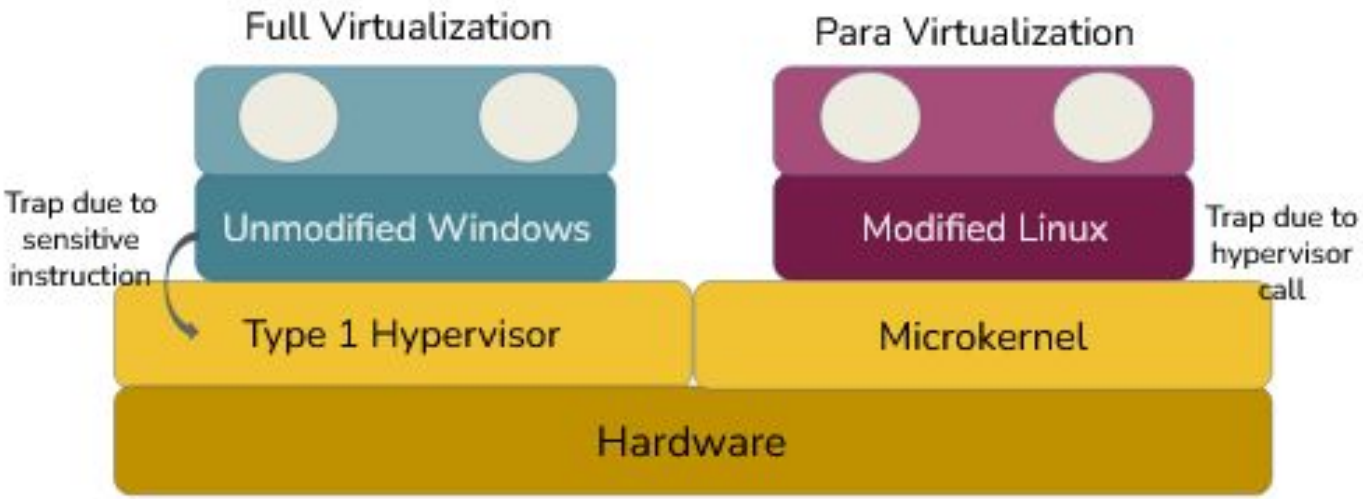
Commercial hosted hypervisors
VMware Workstation
VMware Fusion

HYBRID VIRTUALIZATION

Hybrid Virtualization

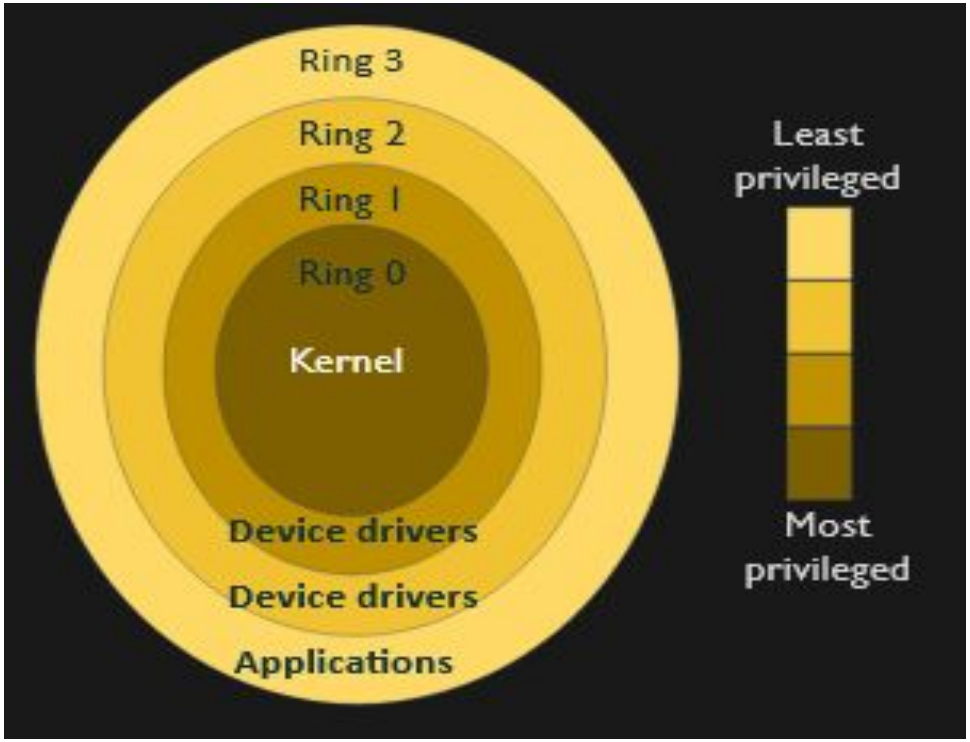
Combination of both

- Para Virtualization for specific hardware drivers, I/O and memory-intense workloads
- Full Virtualization by the the host for other features. .
  eg) Remote Direct Memory Access(RDMA) that exchanges data in main memory without involving CPU, OS uses paravirtual driver to bypass VM kernel



VIRTUALIZATION MODELS

Protection Ring



A protection ring is one of two or more hierarchical *levels* or *layers* of privilege within the architecture of a computer system

Ring 0	Operating system kernel, system drivers
Ring 1	Guest OS, equipment maintenance programs, drivers, programs that work with the ports of the computer I / O
Ring 2	Database management system, the expansion of the OS
Ring 3	**A**pplications, user-run

Source: https://en.bmstu.wiki/Protection_ring_(Operating_Systems)

Virtualization Techniques

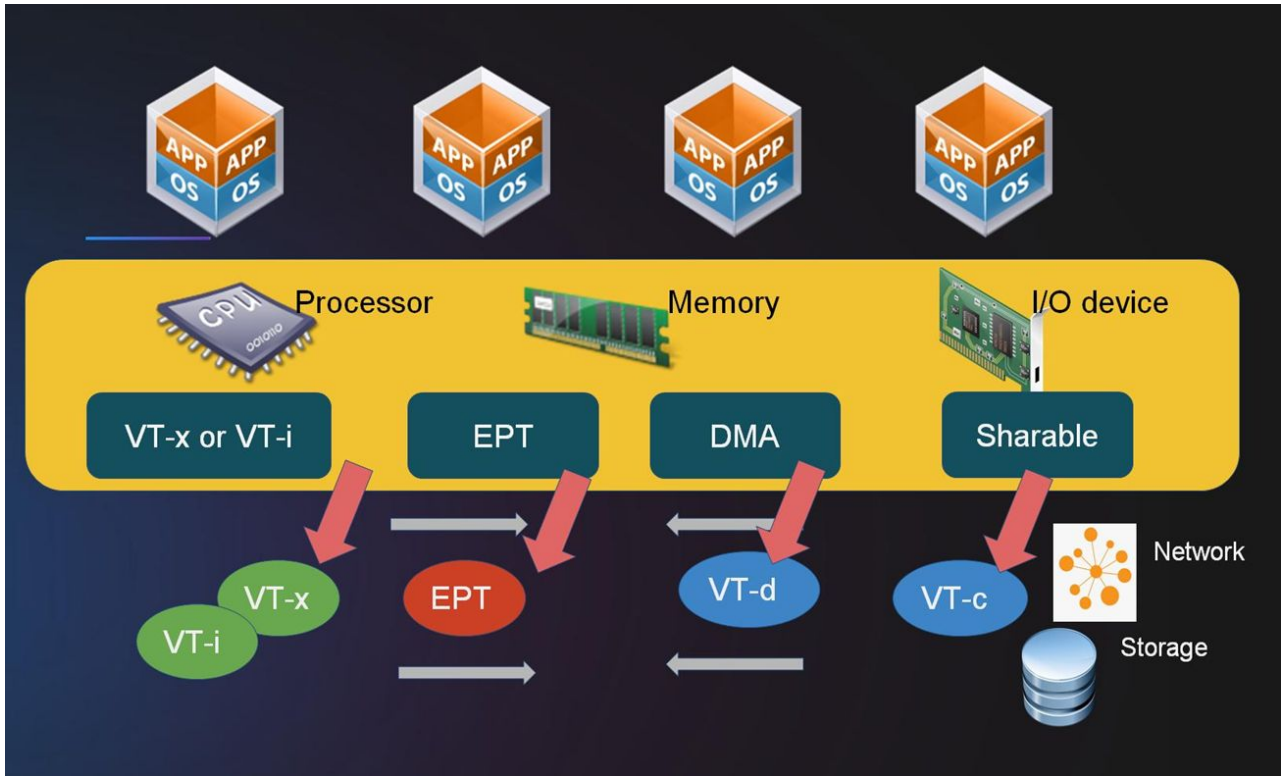
Intel/AMD provides hardware assisted full virtualization that offers

- Processor virtualization
- Memory virtualization
- I/O virtualization

Intel and AMD offer

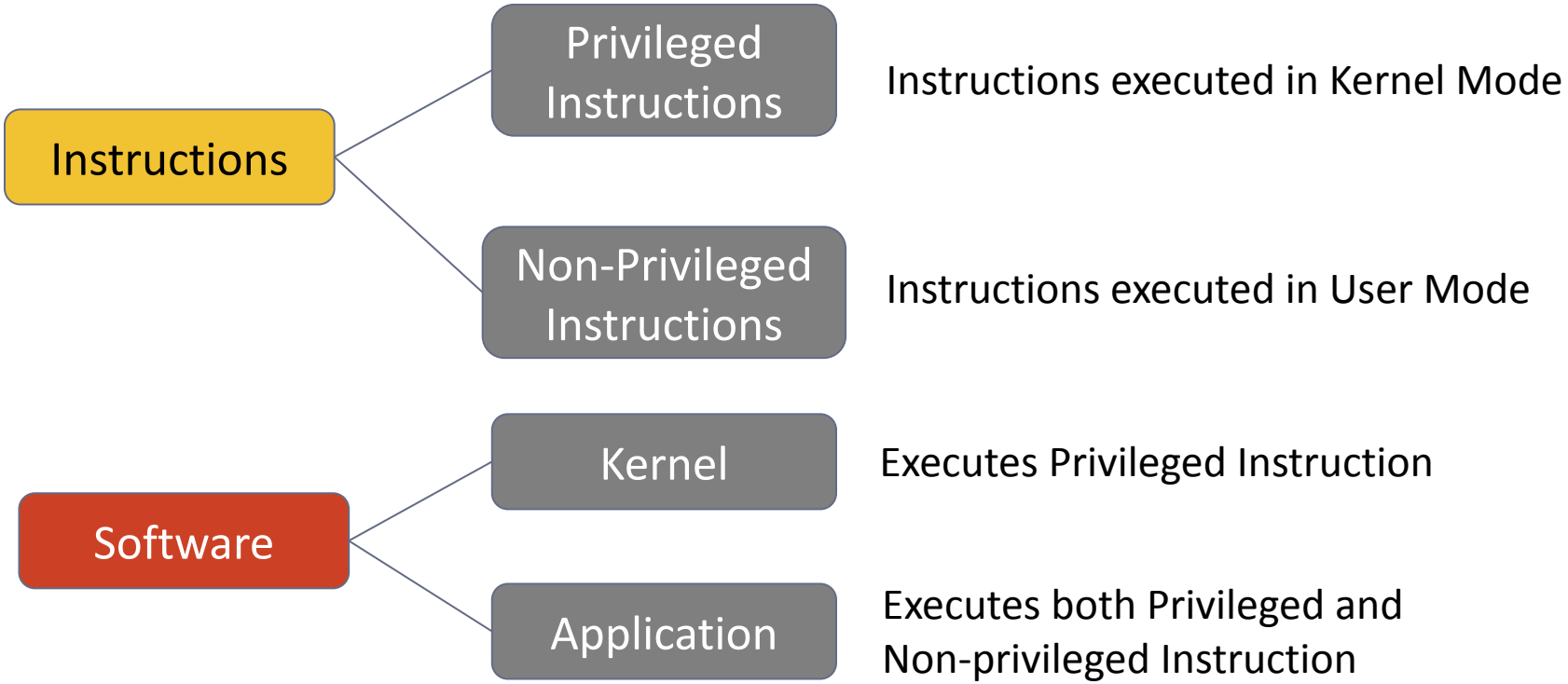
- An additional mode called privilege mode level to x86 processors.
- The VMM runs in a new root level mode below the OS kernel level.

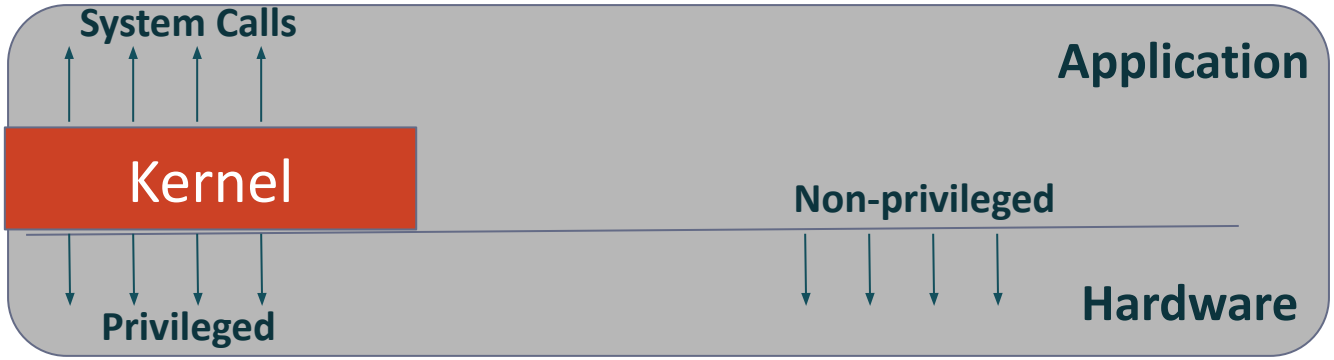
Virtualization Techniques



- For processor, Intel offers the VT-x or VT-i technique.
- For memory virtualization, Intel offers the Extended Page Table (EPT)
- For I/O virtualization, Intel offers
  - VT-d for I/O MMU virtualization
  - VT-c for Network Virtualization

Instructions - Privileged and Non-Privileged





Source: https://slideplayer.com/slide/5103233/

Processor Virtualization

Two modes of Operation

- Supervisor Mode
- Hypervisor Mode

Supervisor Mode

- Execution mode of all instructions including privileged instructions
- System calls - To perform specialized function from user mode to supervisor mode

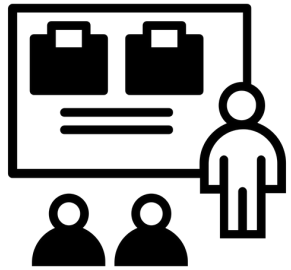
Hypervisor Mode

- x86 virtualization instructions for a hypervisor to control Ring-0
- Intel VT-x and AMD-V creates a new Ring -1 to enable guest OS to run in Ring 0

Processor Virtualization

Critical Instruction Types

- Privileged instructions
- Control-sensitive instructions
- Behavior-sensitive instructions

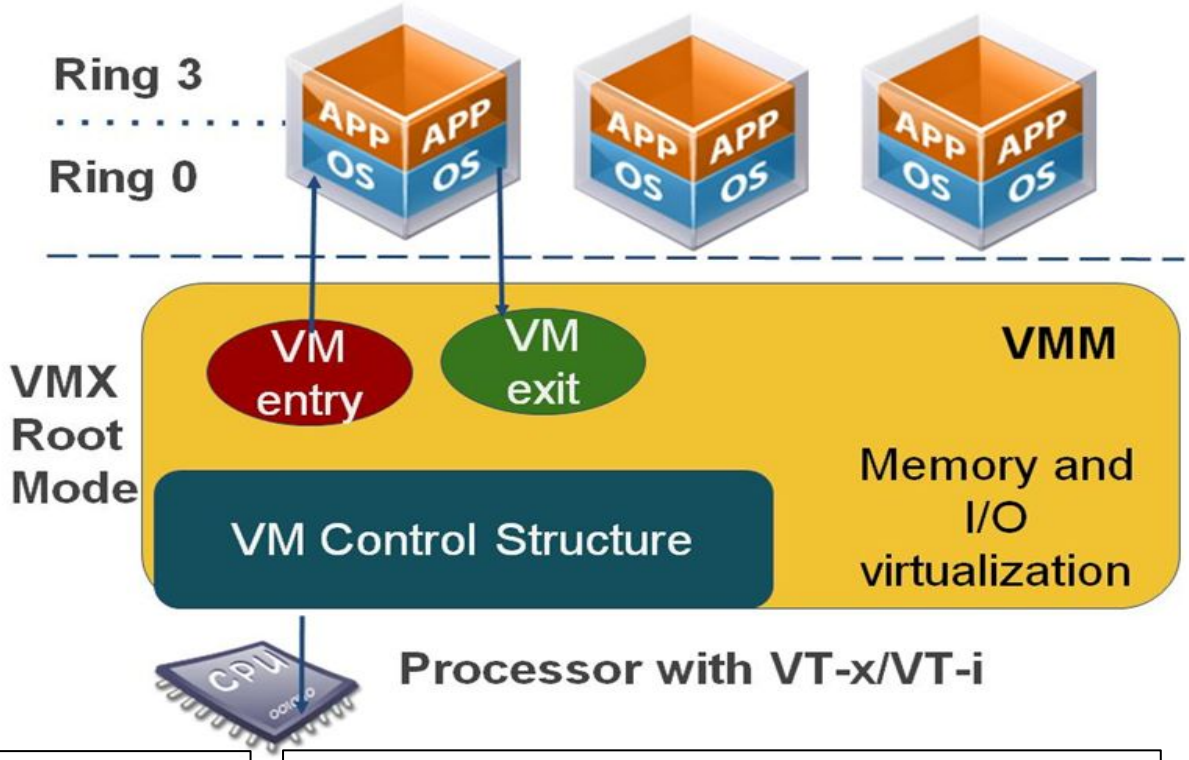


CPU Virtualization has a special instruction set called Virtual Machine Extensions(VMX) with 10 instructions

VMPTRLD Load pointer to VMCS	**VMPTRST** Store Pointer to VMCS
VMCLEAR Clear VMCS	**VMREAD** Read Field from VMCS
VMWRITE Write Field to VMCS	**VMCALL** Call to VM Monitor
VMLAUNCH Launch Virtual Machine	**VMRESUME** Resume Virtual Machine
VMXOFF Leave VMX Operation	**VMXON** Enter VMX Operation

Processor Virtualization

VM Control Structure (VMCS) is a data structure in memory to store and track the VM transitions



Two new modes

VMX root operation - meant for VMM

VMX Non-root operation - meant for Guest OS

Two new transitions

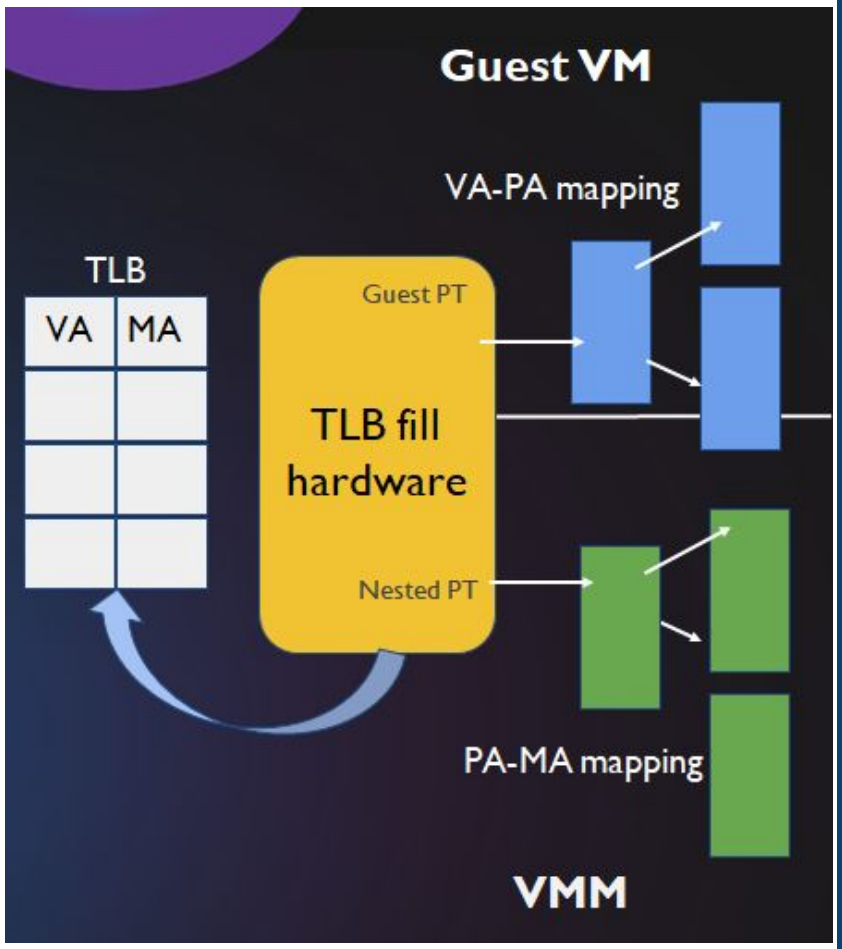
VM Entry - from VMX root operation to non root

VM Exit - from VMX non-root operation to root

Memory Virtualization

To optimize virtual memory performance, all modern x86 CPUs has a

- Memory management unit (MMU)
- Translation lookaside buffer (TLB)

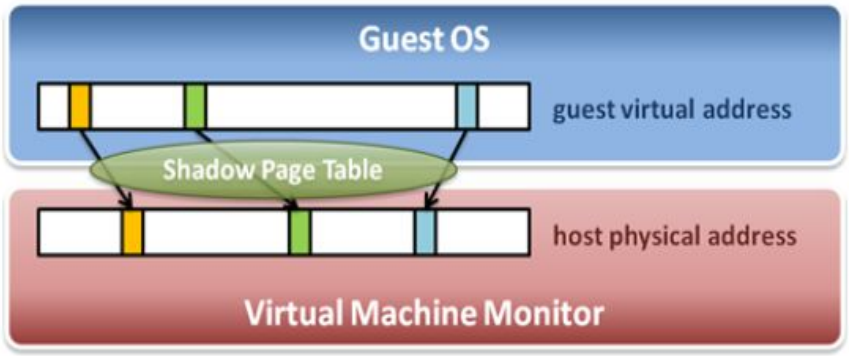


Two stage mapping process between the guest VM and VMM

- Virtual memory to Physical memory
- Physical memory to Machine memory

Memory Virtualization

Shadow Page Table - additional page table of guest OS that resides in VMM



Nested Paging - Technology to provide hardware assistance to the two-stage address translation in a virtual execution environment

Extended Page Table - EPT reduces the need for VMM to keep syncing the shadow pages eliminating the overhead by enabling the CPU to just sync TLB

Source: https://revers.engineering/mmu-ept-technical-details/

I/O Virtualization

Create or Share I/O devices for VMs

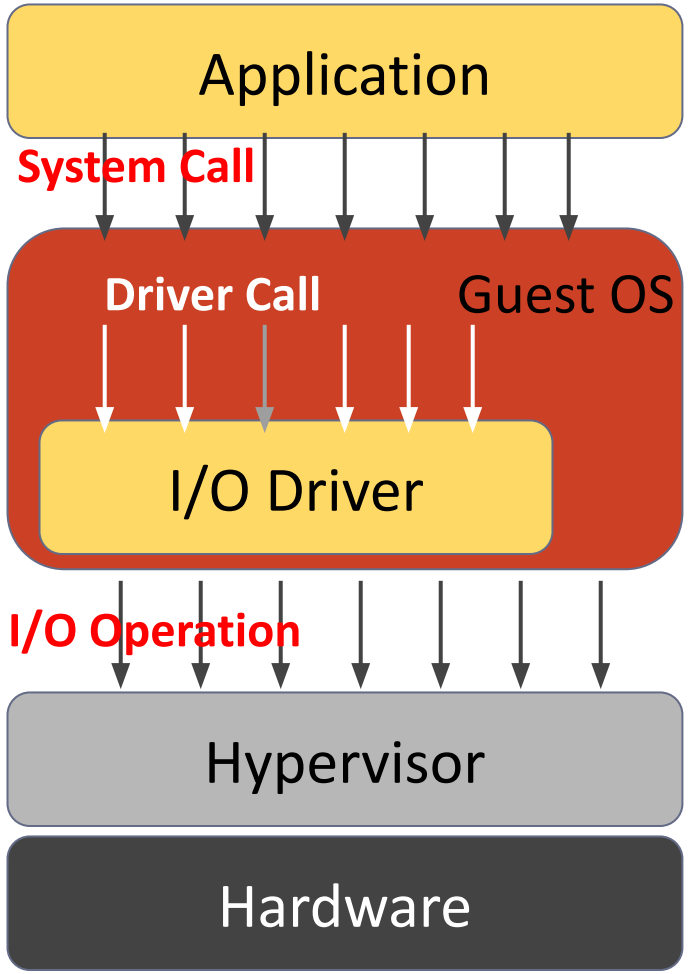
Traditional I/O Techniques

- Direct Memory Access (DMA)
- PCI/PCI Express

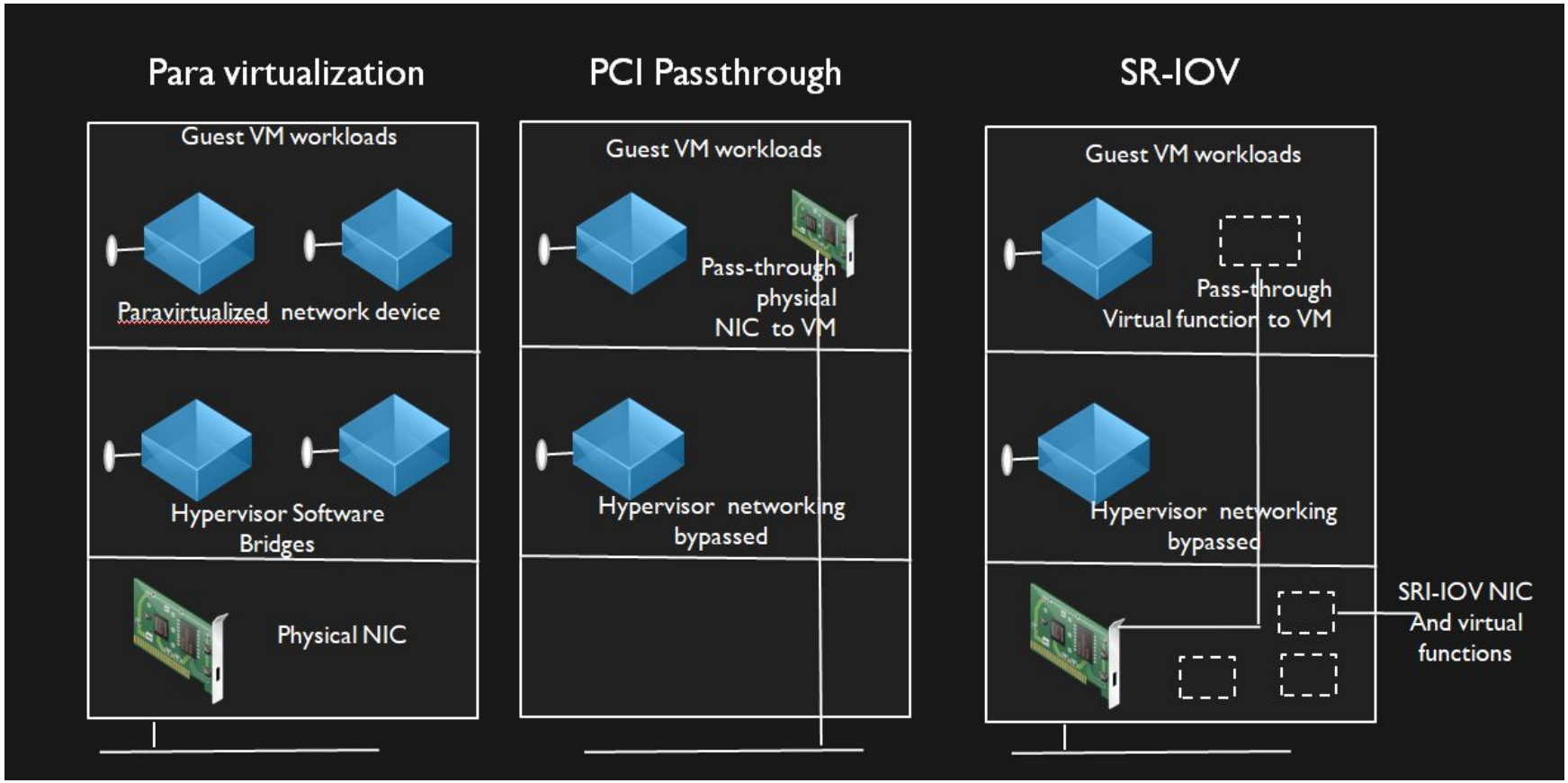
System Call - Interface between Application and Guest OS

Driver Call - Interface between Guest OS and I/O Drivers

I/O Operations - Interface between I/O device drivers and Hypervisor



I/O Virtualization



In I/O virtualization, a virtual device is substituted equivalent for its physical devices,

- Network interface card (NIC)
- Host bus adapter (HBA).

I/O Virtualization

PCI passthrough allows

- Guest to have exclusive access to **PCI** devices for a range of tasks.
- **PCI** devices to appear and behave as if they were physically attached to the guest operating system.
- **PCI** devices are limited by the virtualized system architecture.

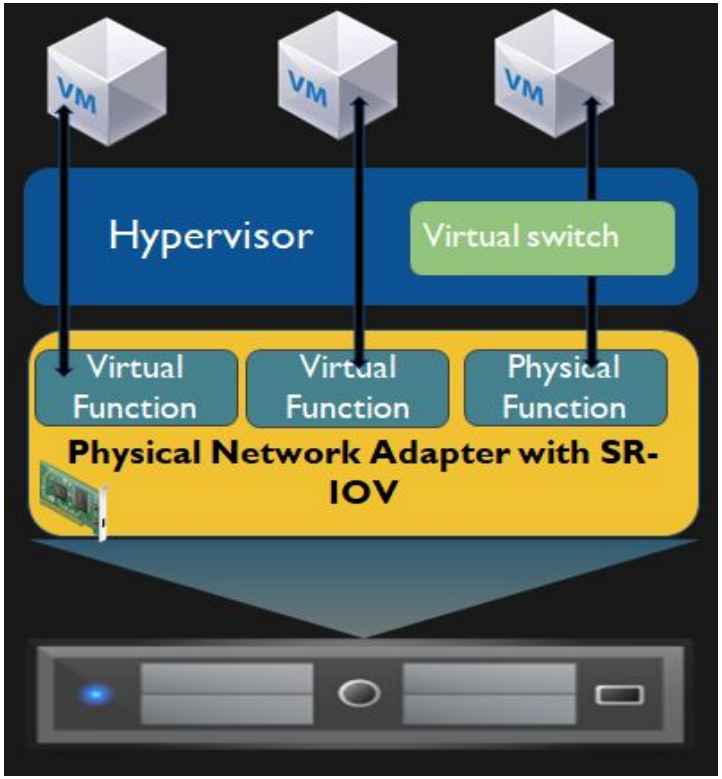
Single root I/O virtualization (SR-IOV) interface

- an extension to the PCI Express (PCIe) specification.
- allows a device, such as a network adapter, to separate access to its resources among various PCIe hardware functions.

I/O Virtualization

Isolation of PCI Express (PCIe) to one or more physical/virtual PCIe devices called Virtual Function Devices

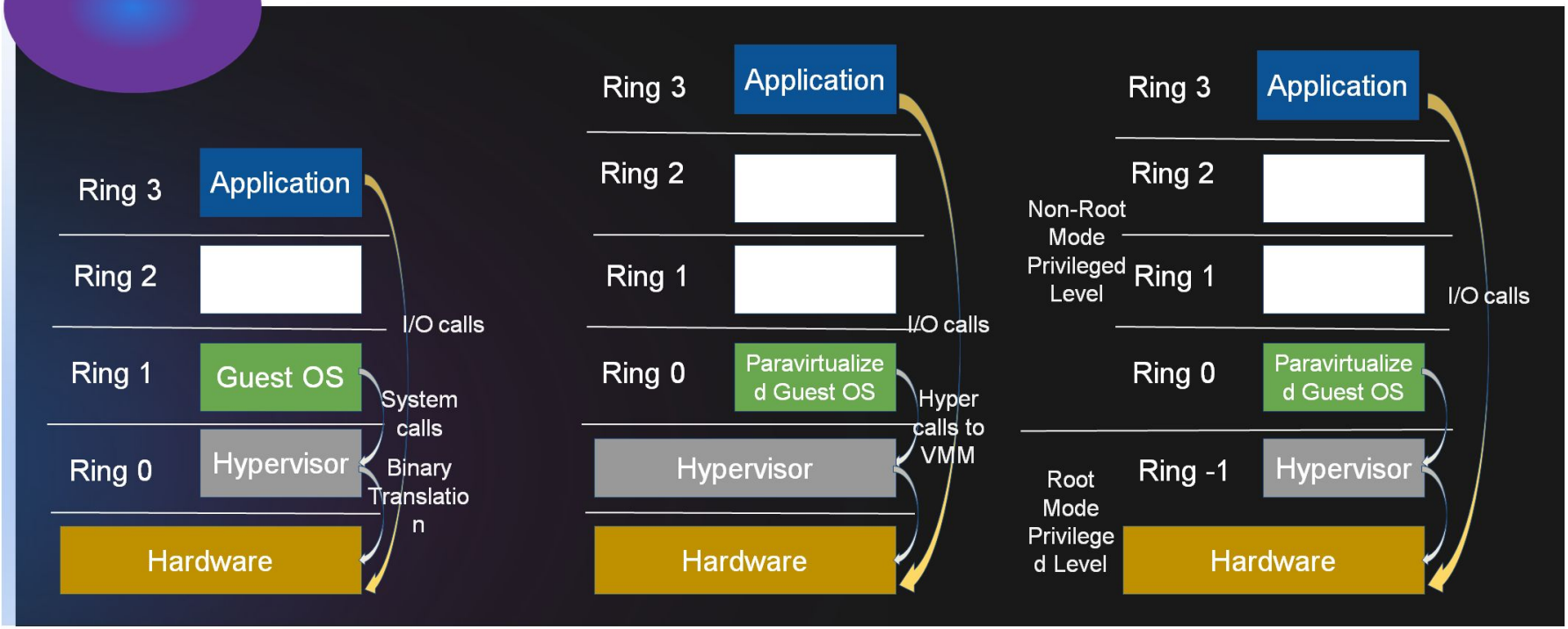
PCI pass-through is the predecessor technology to SR-IOV but the drawback is the physical network adapter is completely dedicated to a single VM guest



Benefits

1. Better Virtualization Network Manageability
2. Increase Virtual Machine Performance

Full vs Para vs Hardware Assisted Virtualization



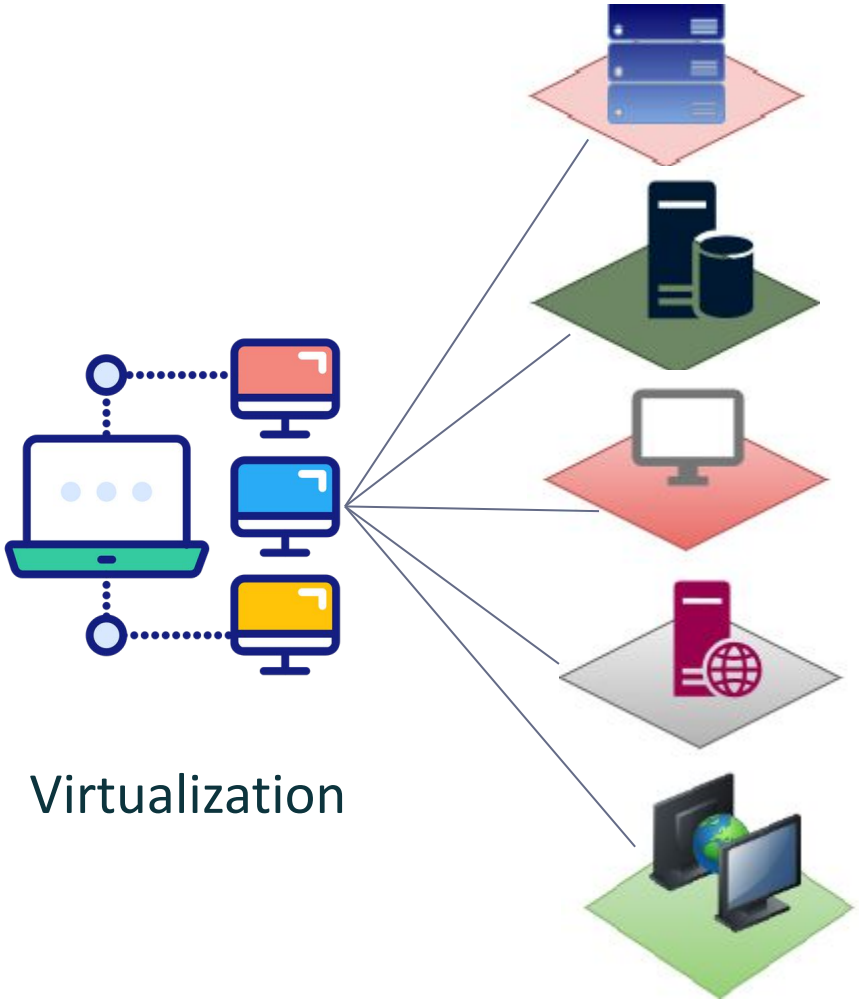
Full Virtualization	Para Virtualization	Hardware Assisted Full Virtualization
• Guest OS in Ring 1 • Guest OS system calls are binary translated • System call triggers interrupt and pass control to kernel	• Kernel of Guest OS is modified and it is in Ring 0 • Replace any privileged operations to hypercalls. • Hypervisor perform the task of guest kernel	• Guest OS in Ring 0 and Hypervisor in Ring -1 • All the privileged and sensitive instructions are trapped in the hypervisor automatically

Full vs Para vs Hardware Assisted Virtualization

	Full Virtualization	Hardware-Assisted	Paravirtualization
Technique	Binary Translation Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification	Unmodified Guest OS	Unmodified Guest OS	Modified Guest OS
Supported OS	Linux,Windows	Linux,Windows	Linux
Portability and Compatibility	More portable and excellent compatibility	More portable and excellent compatibility	Less portable and poor compatibility
Hypervisor Independent	Yes	Yes	No
Products	VMware, KVM, Microsoft Hyper-V	VMware, KVM, Microsoft Hyper-V	Xen, VMware, KVM

VIRTUALIZATION TYPES

Virtualization Types



Server Virtualization

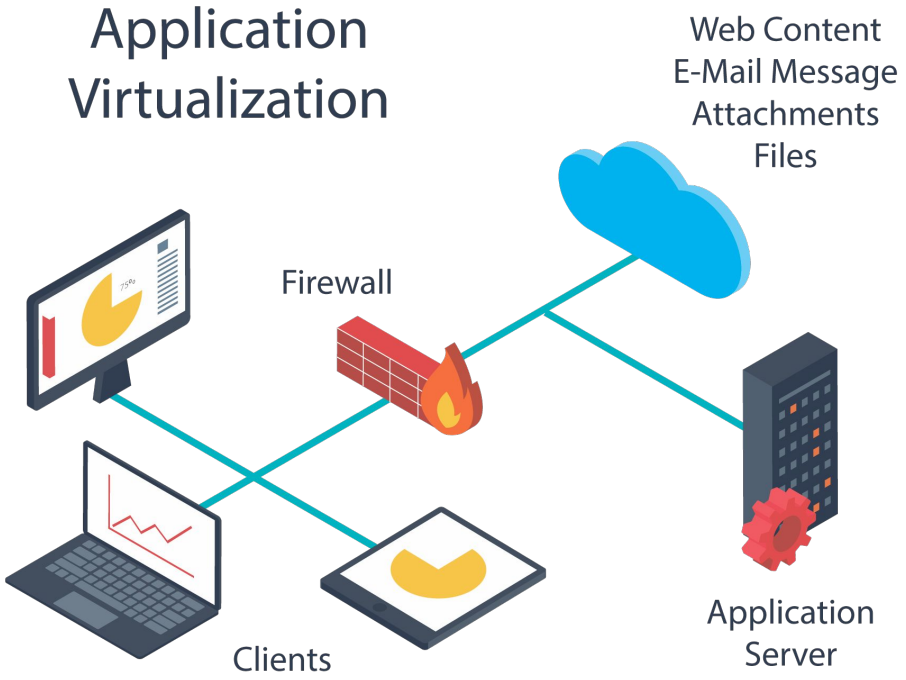
Storage Virtualization

Desktop Virtualization

Application Virtualization

Network Virtualization

Application Virtualization



Technology that allows users to access applications residing on a remote server with few resident programs

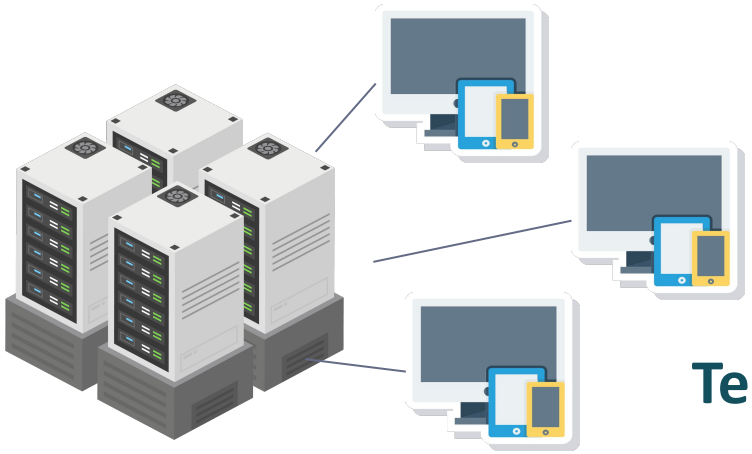
- Thin client
- a terminal
- a network workstation

- IT admins can set up remote applications on a server and deliver the apps to an end user's computer.
- The user interface is transparent, the virtual application runs locally, and the application runs remotely.

Application Virtualization - Use case

Digital Workspaces

Desktop Virtualization



Terminal Services

Application Virtualization

Benefits

Simplified Application installation

Simplified Application retirement

No more Application conflicts

Multiple run-time environments

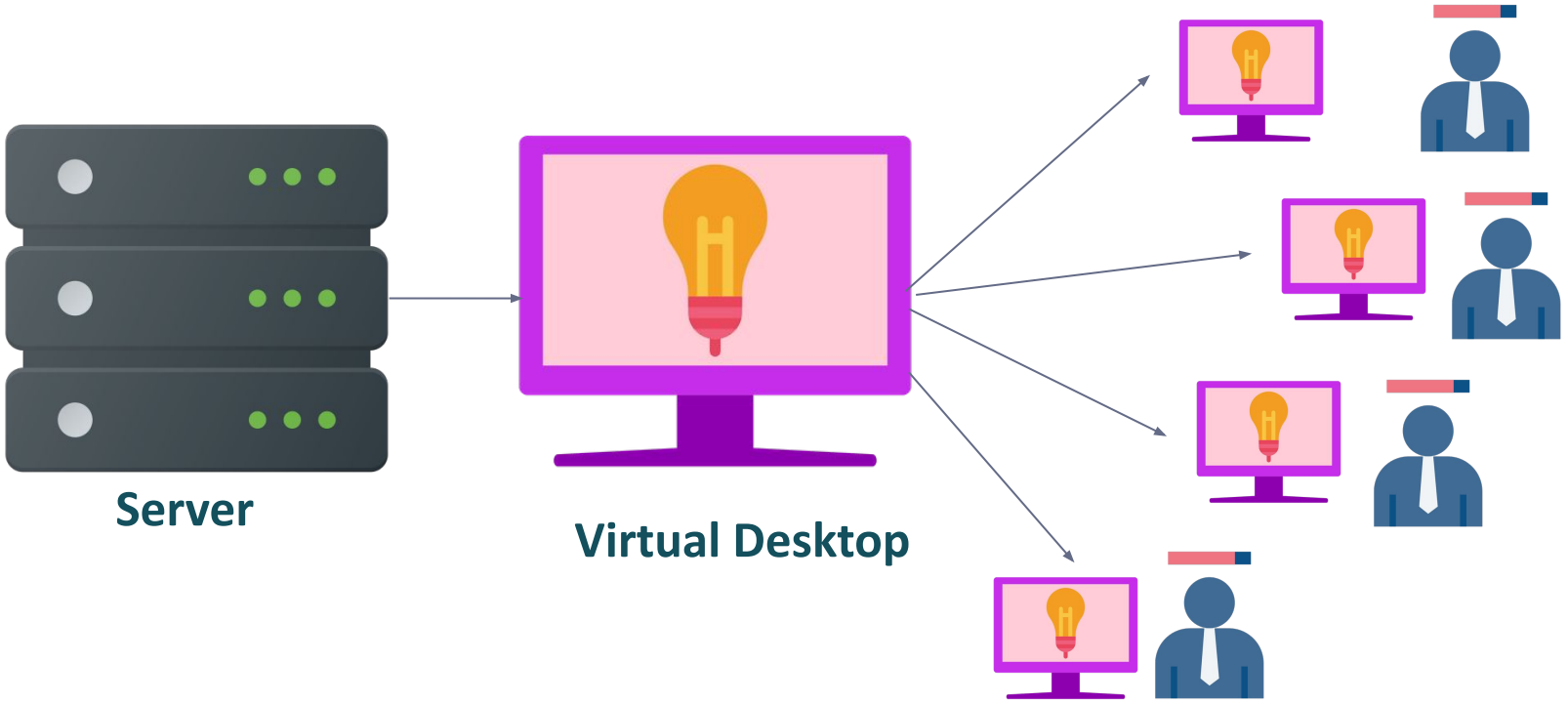
Multiple versions of the same application

Application updates

Rollback mechanism

Desktop Virtualization

- It allows a central administrator (or automated administration tool) to deploy simulated desktop environments to hundreds of physical machines at once.
- It allows admins to perform mass configurations, updates, and security checks on all virtual desktops.



Desktop Virtualization

It creates a software-based (or virtual) version of an end user's desktop environment and operating system (OS) that is decoupled from the end user's computing device or client.

Deployment Models

1. Virtual Desktop Infrastructure (VDI)
2. Remote Desktop Services (RDS)
3. Desktop-as-a-Service(DaaS)

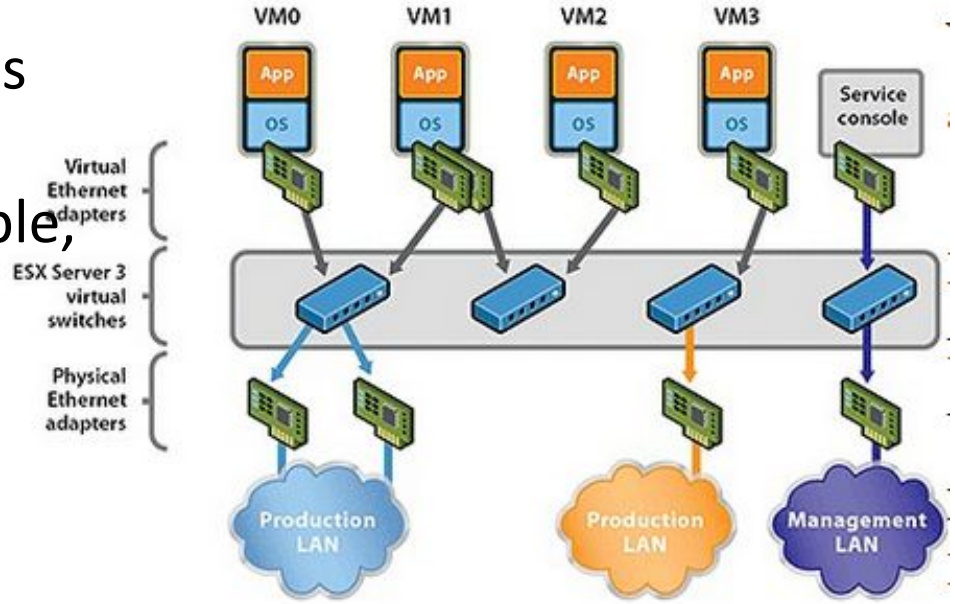
Network Functions Virtualization

Network functions virtualization (NFV) separates a network's key functions to distribute among environments

- Directory services
- File sharing
- IP configuration

Virtualizing networks reduces the number of physical components to create multiple, independent networks

- Switches
- Routers
- Servers
- Cables
- Hubs



Network Functions Virtualization

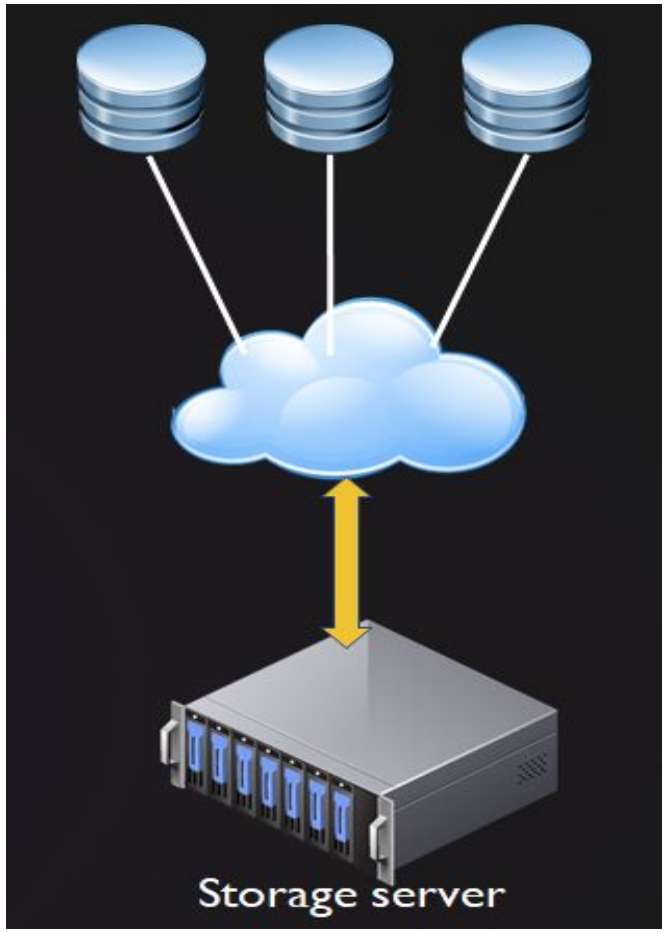
- Reduction in the number of physical network devices
- Segmentation of Networks made easier
- Failover mode – automated switching of the defective disk to a backup, and the failed component can be repaired, while the system continues to run
- Rapid Change/Scalability and Agile deployment

Storage Virtualization

Pooling of physical storage from multiple network storage devices into a single storage device that is managed from a central console

Storage virtualization is commonly used in SANs made up of several components

- disk arrays
- switches
- all the separate storage disks on the network are grouped and then merged in an array.
- Servers can then access the array as though it were a local storage device.



Storage Virtualization

Benefits

Enables dynamic storage utilization and virtual scalability of attached storage resources, both block and file.

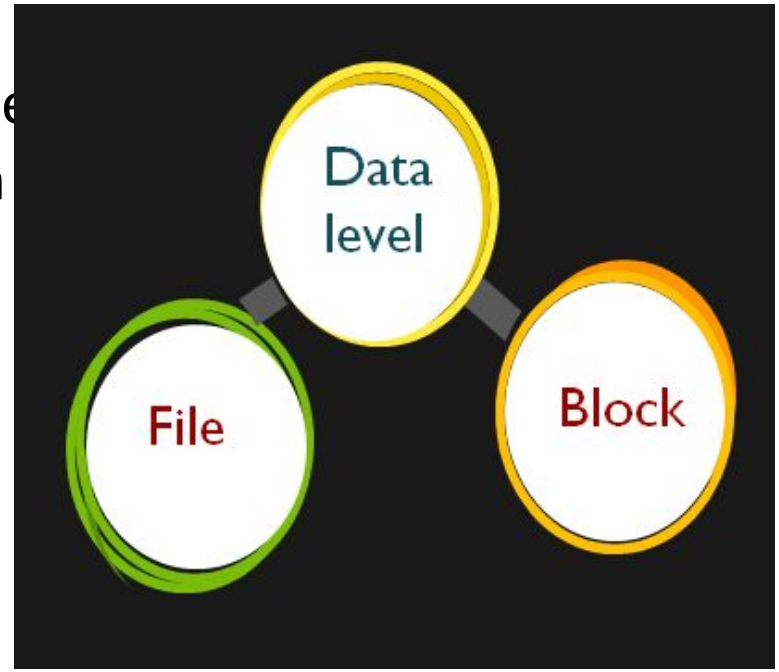
Avoids downtime during data migration. Virtualization operates in the background to maintain data's logical address to preserve access.

Centralizes a single dashboard to manage multi-vendor storage devices, which saves management overhead and money.

Can add storage intelligence like tiering, caching, replication and a centralized management interface in a multi-vendor environment.

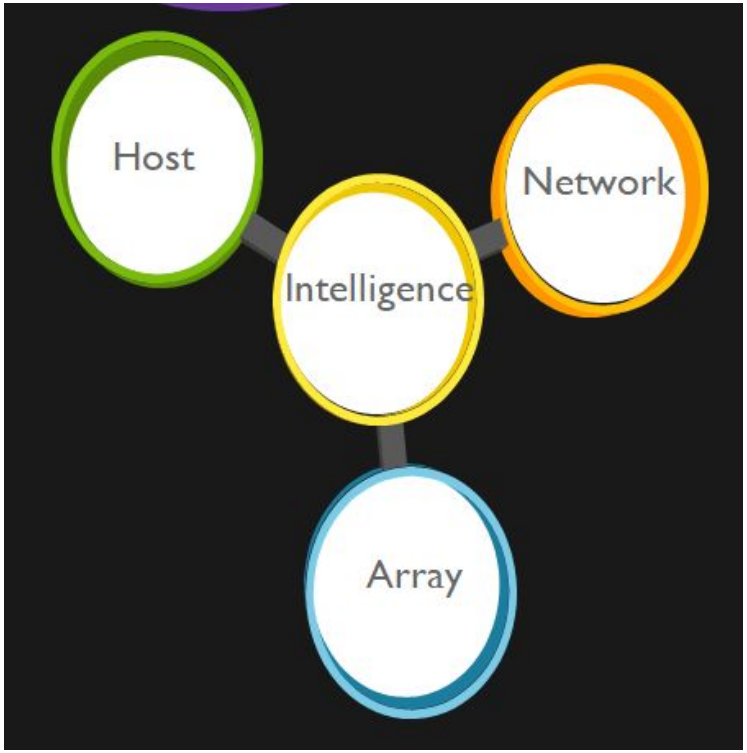
Storage Virtualization based on Data level

- Block-level storage
  - The block-based virtual storage storage such as drive partition in a storage device.
  - Physical components
    - memory blocks
    - Storage media
  - Logical components
    - drive partitions



- File-level storage
  - File-based storage virtualization is a specific use case, applied to network-attached storage (NAS) systems.
  - File migrations are easier to handle when we pool the NAS resources which will help improve performance

Storage Virtualization based on Intelligence



Based on intelligence the storage is classified as

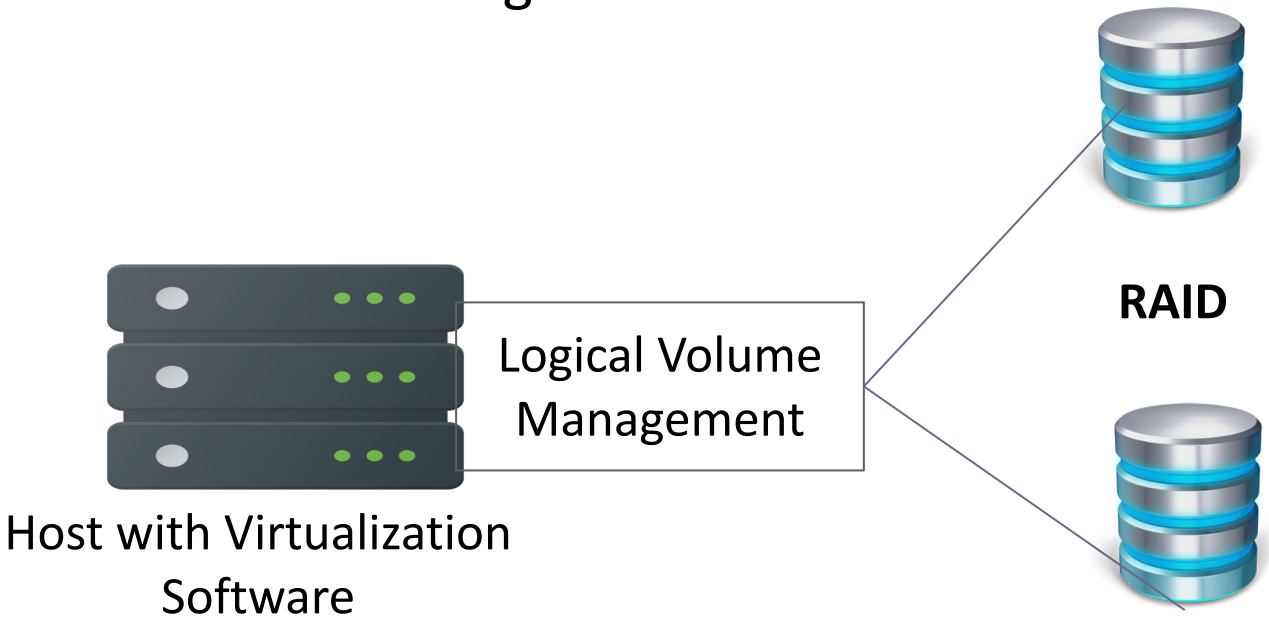
- Host-based
- Array-based
- Network-based

Storage Virtualization based on Intelligence

Host based

A hyper-converged system made up of multiple hosts, presents virtual drives of varying capacity to the guest machines in a data centre environment or cloud storage.

OS virtualizes available storage to optimize capacity and automate tiered storage schedules.



Storage Virtualization based on Intelligence

Array based storage

- acts as the primary storage controller and runs virtualization software, enabling it to pool the storage resources of other arrays for use as storage tiers.

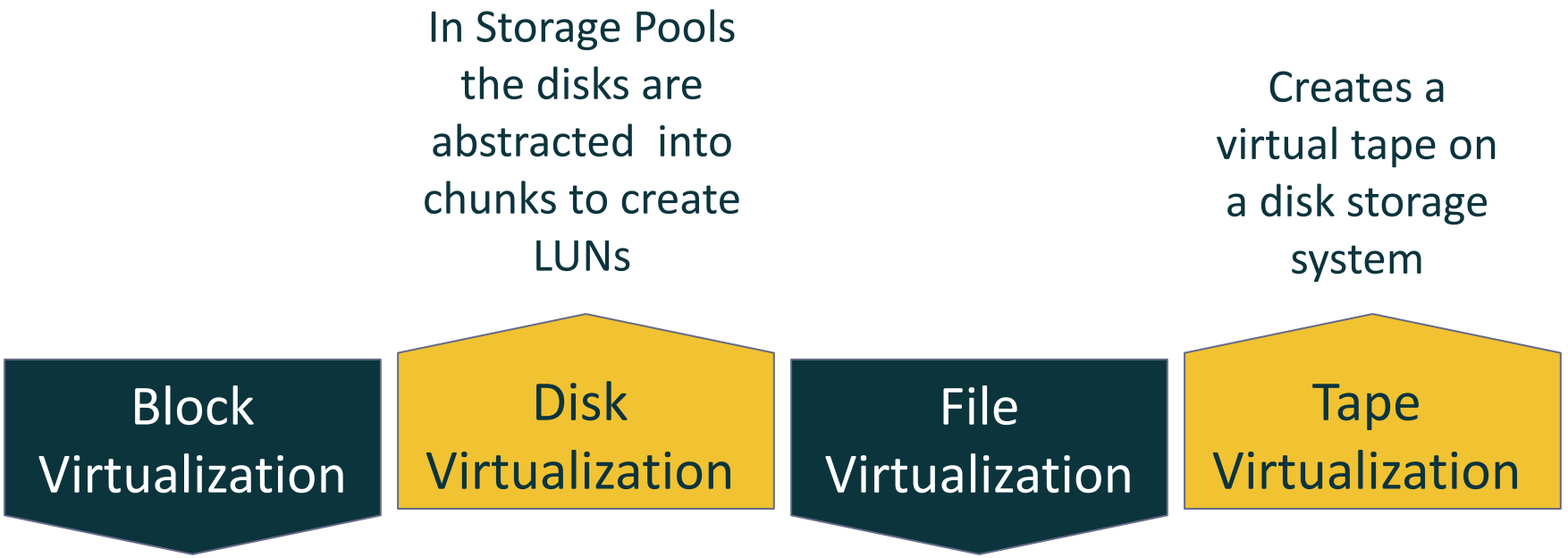
Storage tiering is a specialized storage controller that intercepts I/O requests from secondary storage controllers and automatically tiers data within connected storage systems

Storage Virtualization based on Intelligence

Network based

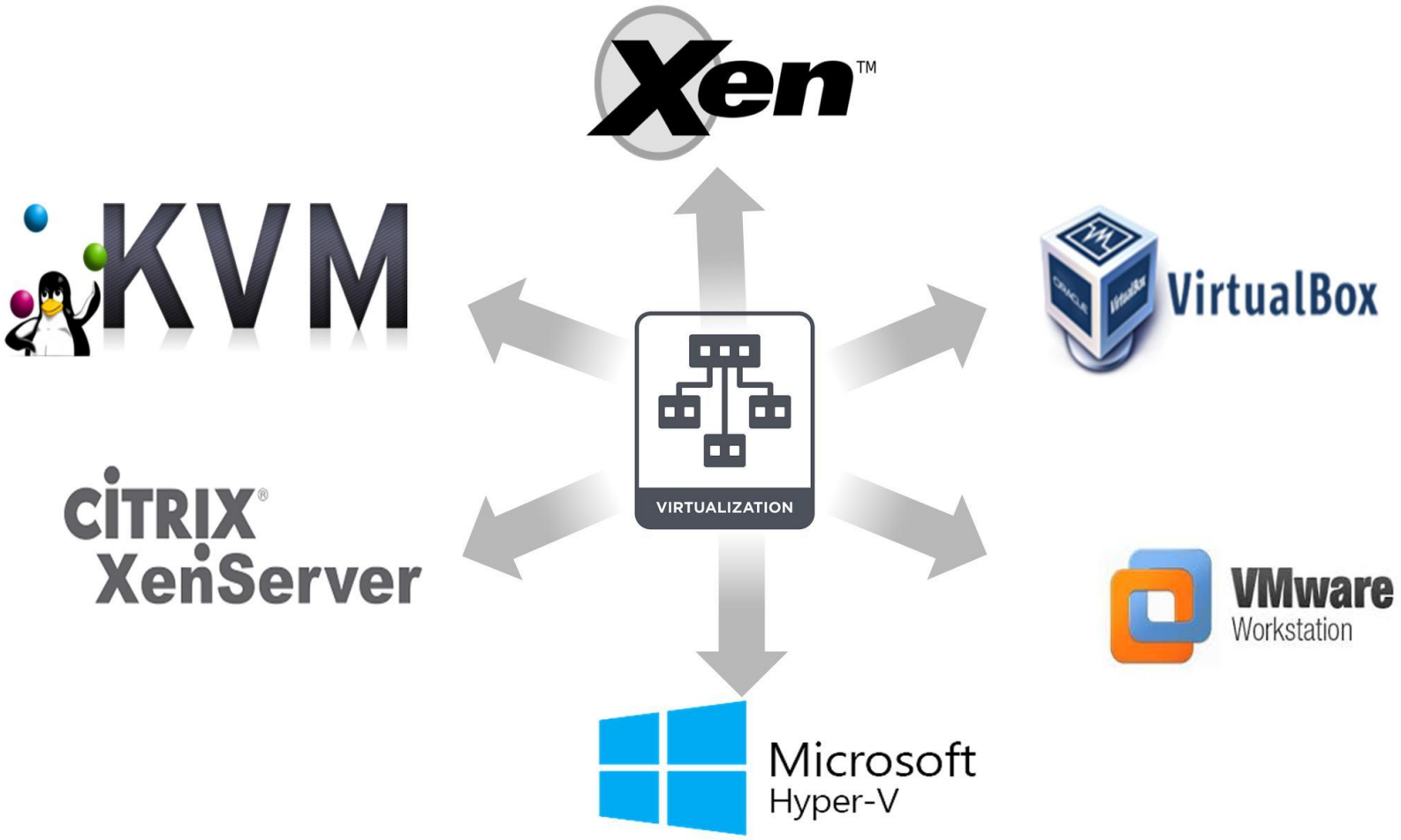
- The intelligence runs from a server or switch, across Fibre Channel or iSCSI networks.
- abstracts storage I/O running across the storage network, and can replicate data across all connected storage devices

Storage Virtualization



VIRTUALIZATION TOOLS

Virtualization Tools



KVM

KVM (Kernel-based Virtual Machine) is an open source hypervisor technology for virtualizing compute infrastructure running on multiple architectures (x86, PowerPC,ia64 (Itanium) compatible hardware.

Linux Kernel Module	Full Virtualization	Intel VT-x/AMD-v Extension
VMM Desktop Interface	Virtio ParaVirtualization	QEMU Libvirt API

KVM

Red Hat Virtualization

RED HAT® OPENSHIFT

openstack™

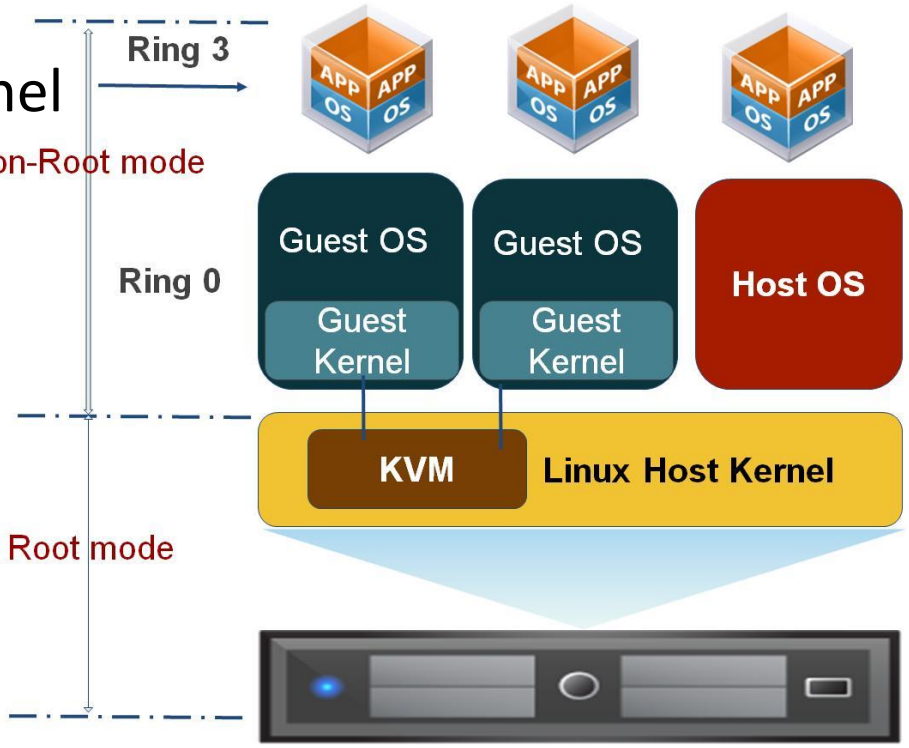
KVM

KVM tap into QEMU's emulation powers to compliment its own hardware acceleration features, presenting its guests with an emulated chipset and PCI bus

KVM uses direct access to a kernel with CPU-specific module (kym-intel or kvm-amd).

Two KVM kernel modules

- kvm.ko module
- kvm-intel.ko or kvm-amd.ko



QEMU + KVM + Libvirt

QEMU and libvirt is a powerful combination that interacts with KVM to provide a virtualization stack that is secure, effective and fully functional

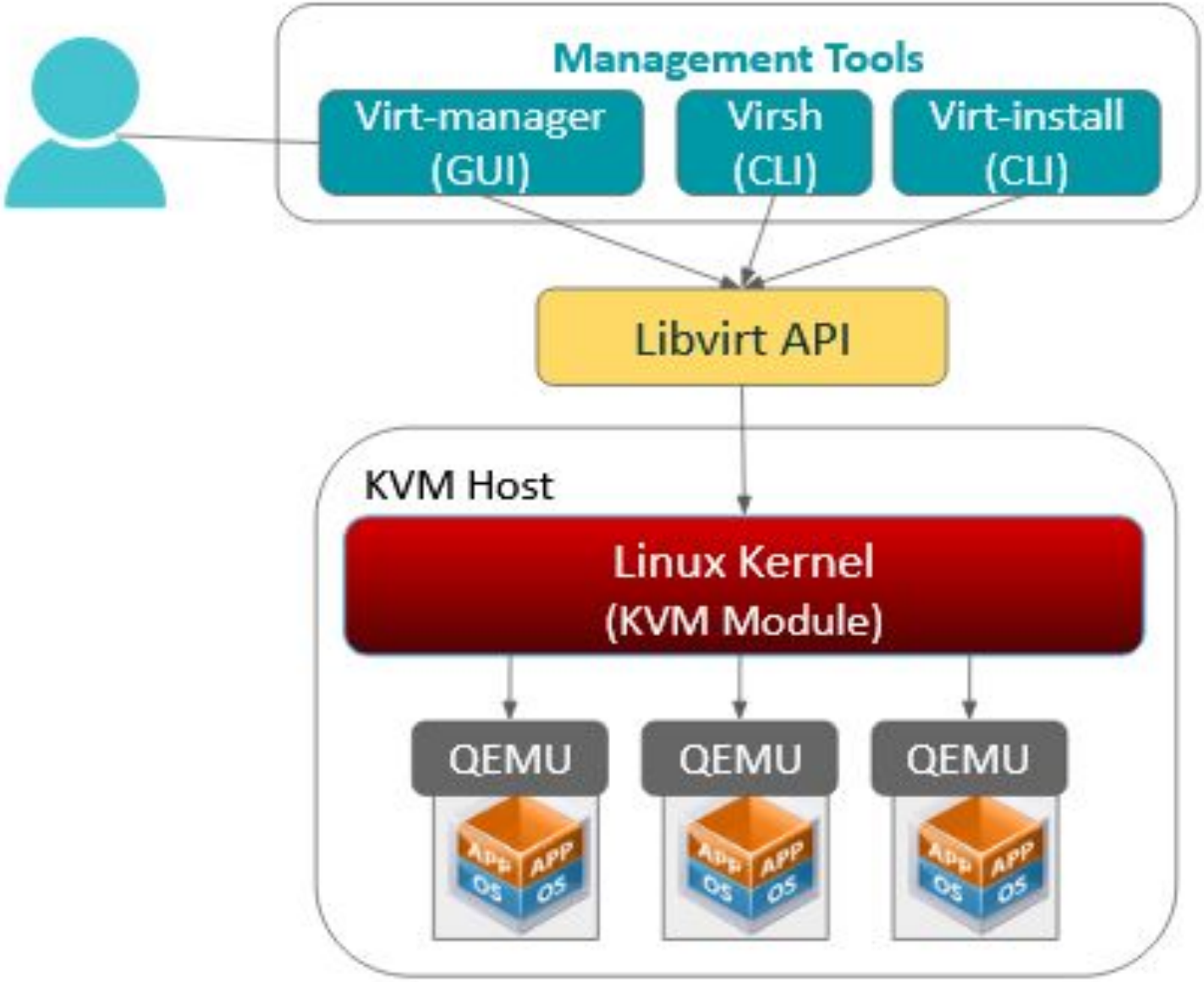
QEMU

- A generic and open source machine emulator and virtualizer.
- Provides hardware emulation and a low-level interface to the VM.
- Each VM is a QEMU process

- Toolkit to manage Virtualization platform
- Exposes a consistent API atop man virtualization technologies.
- APIs are consumed by client tools for provisioning and managing VMs.

libvirt VIRTUALIZATION API

QEMU + KVM + Libvirt



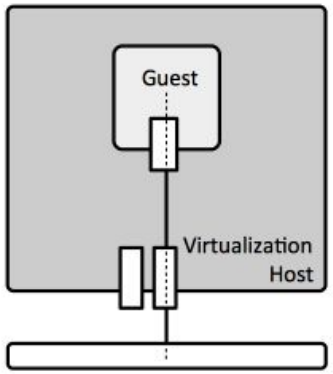
KVM Key Features

- EPT support (server boost)
- KSM (Kernel Same Page Merging) - share memory with COW
- Disk image cloning, sharing, snapshot
- Ballooning
- Live migration with shared storage
- Nested full virtualization
- Virtio paravirtualization
- PCI-passthrough VT-D/IOMMU support

KVM Networking Modes

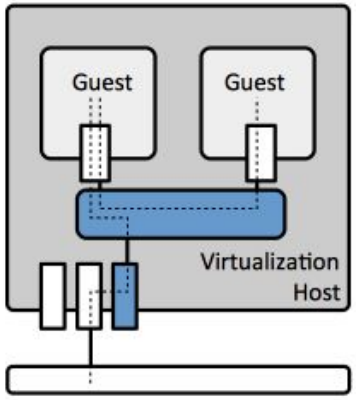
Isolated

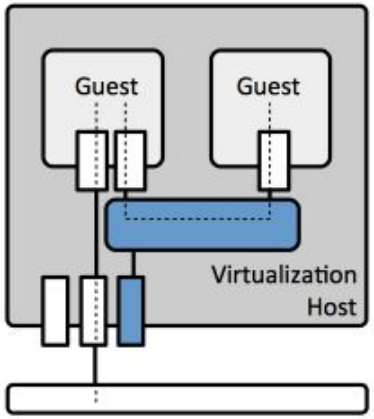
The guests are connected to a network that does not allow any traffic beyond the virtualization host



Routing

The guests are connected to a network that routes traffic between the guest and external hosts without performing any NAT



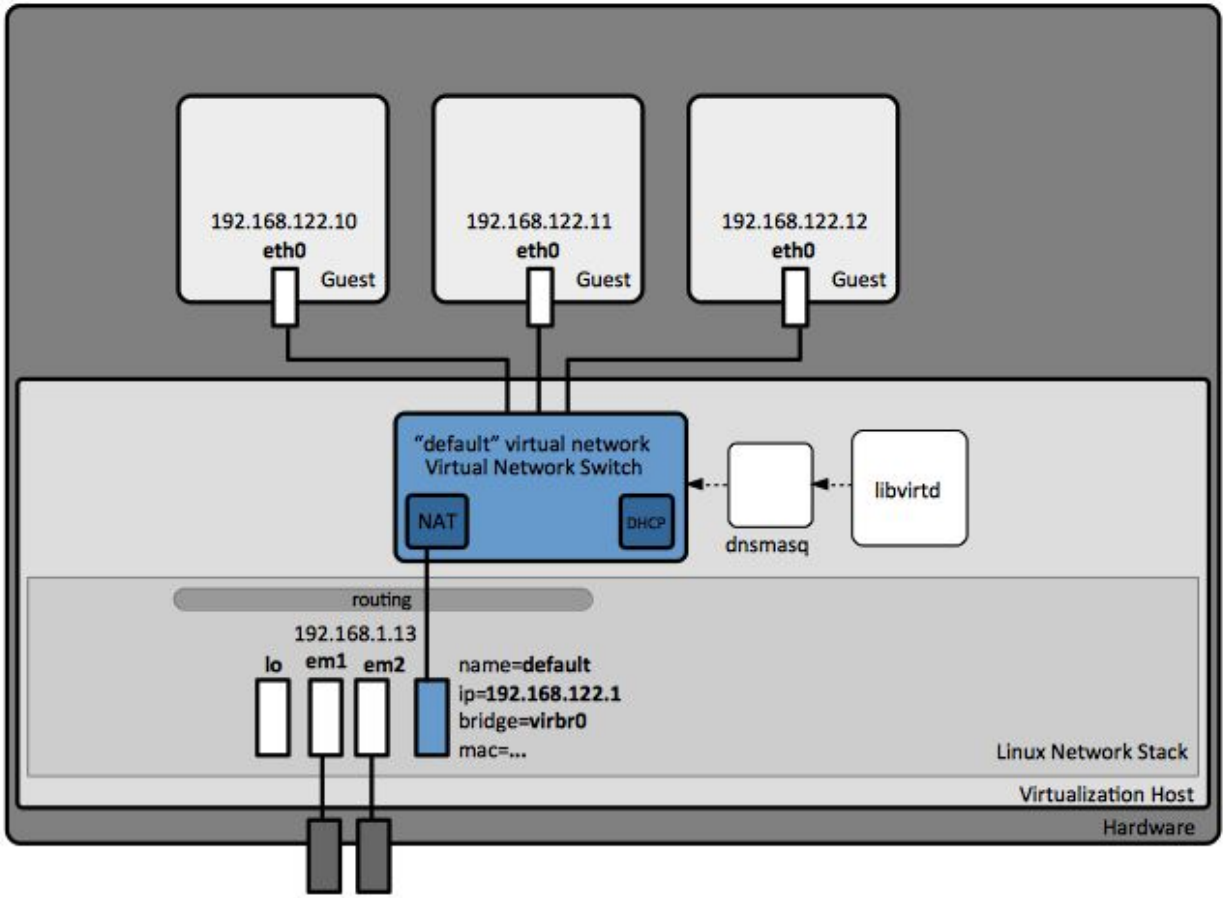


Bridged

The guests are connected to a bridge device that is also connected directly to a physical ethernet device connected to the local ethernet

KVM Networking

The network selection option defaults to Virtual network 'default': NAT. If the default network is not active, virt-manager will prompt to start it. $ virsh net-start default



Each VM (in default network) will be a member of 192.168.122.0/24, with an IP address in the range of 192.168.122.2 to 192.168.122.254

KVM Libvirt Client Tools

- **virsh**
  - Command-line tools for communicating with libvirt
- **virt-manager**
  - GUI to manage KVM, qemu/kvm, xen, and lxc.
  - Contains a VNC and SPICE client for direct graphical access to VMs.
  - GUI alternative to virsh, albeit less capable.
- **virt-install**
  - Helper tools for creating new VM guests.
  - Part of the virt-manager project.
- **virt-viewer**
  - UI for interacting with VMs via VNC/SPICE.
  - Part of the virt-manager project.

KVM Installation

- Hardware virtualization extensions from BIOS
  # egrep 'vmx|svm' /proc/cpuinfo
- Install packages and add username (example: cloud) to the kvm and libvirt groups

```
$ sudo apt install qemu-kvm libvirt-clients libvirt-daemon  virtinst
libvirt-daemon-system virt-manager bridge-utils
$ sudo usermod -aG kvm cloud
$ sudo usermod -aG libvirt cloud
```

- Default directory to hold VM images is **/var/lib/libvirt/images**

To ensure all client utilities default to qemu:///system, add the following configuration to .config directory.

```
$ sudo cp -rv /etc/libvirt/libvirt.conf /home/cloud/.config/libvirt/ &&\
$ sudo chown cloud:cloud  /home/cloud/.config/libvirt/libvirt.conf
```

KVM Installation

Edit the file /home/cloud/.config/libvirtd.conf

```
uri_default = "qemu:///system"
```

Edit the file /etc/libvirt/qemu.conf

```
user = "root"
group = "kvm"
```

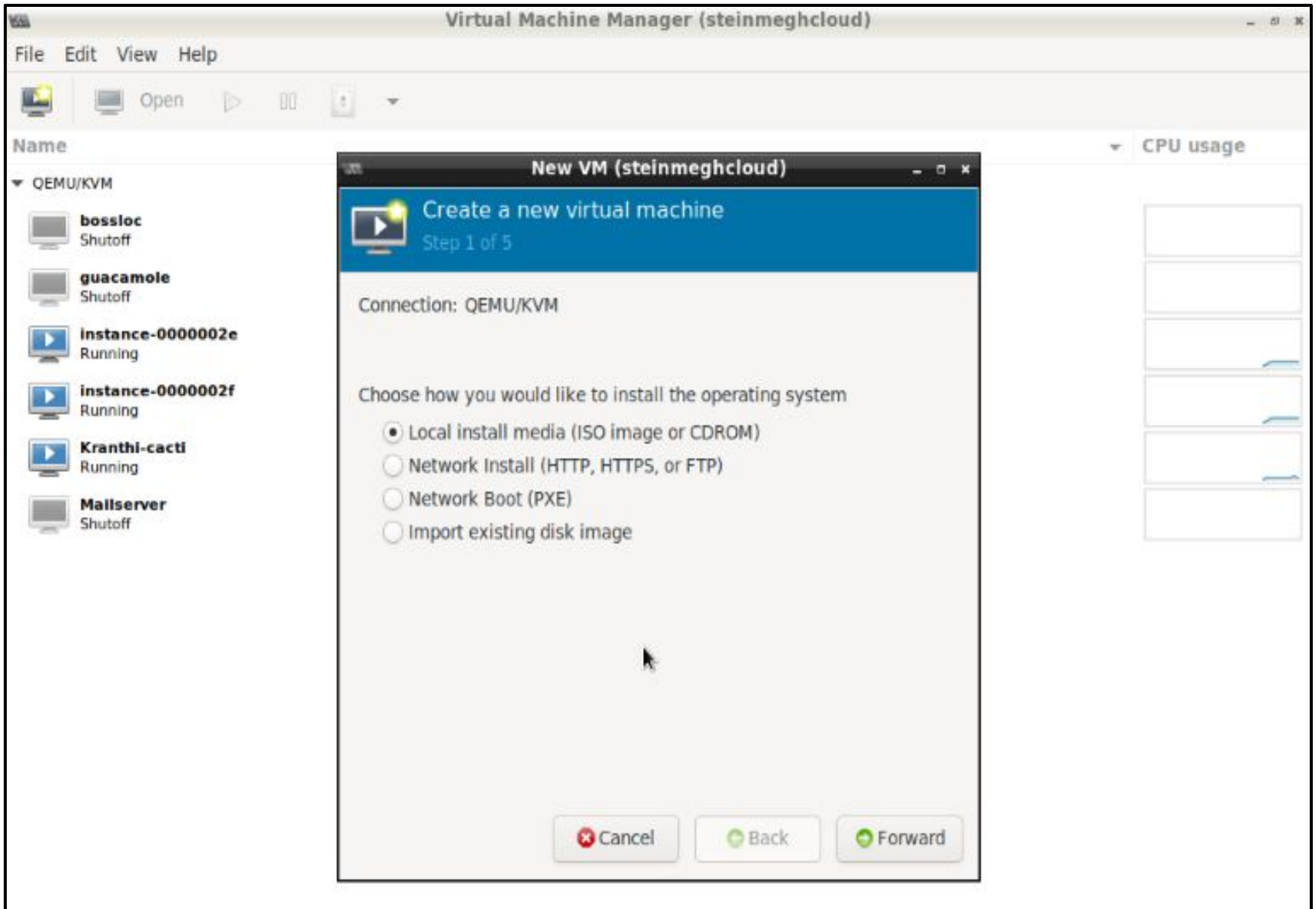
Start the Libvirtd process

```
$ sudo systemctl start libvirtd
$ sudo systemctl enable libvirtd
$ sudo systemctl status libvirtd
```

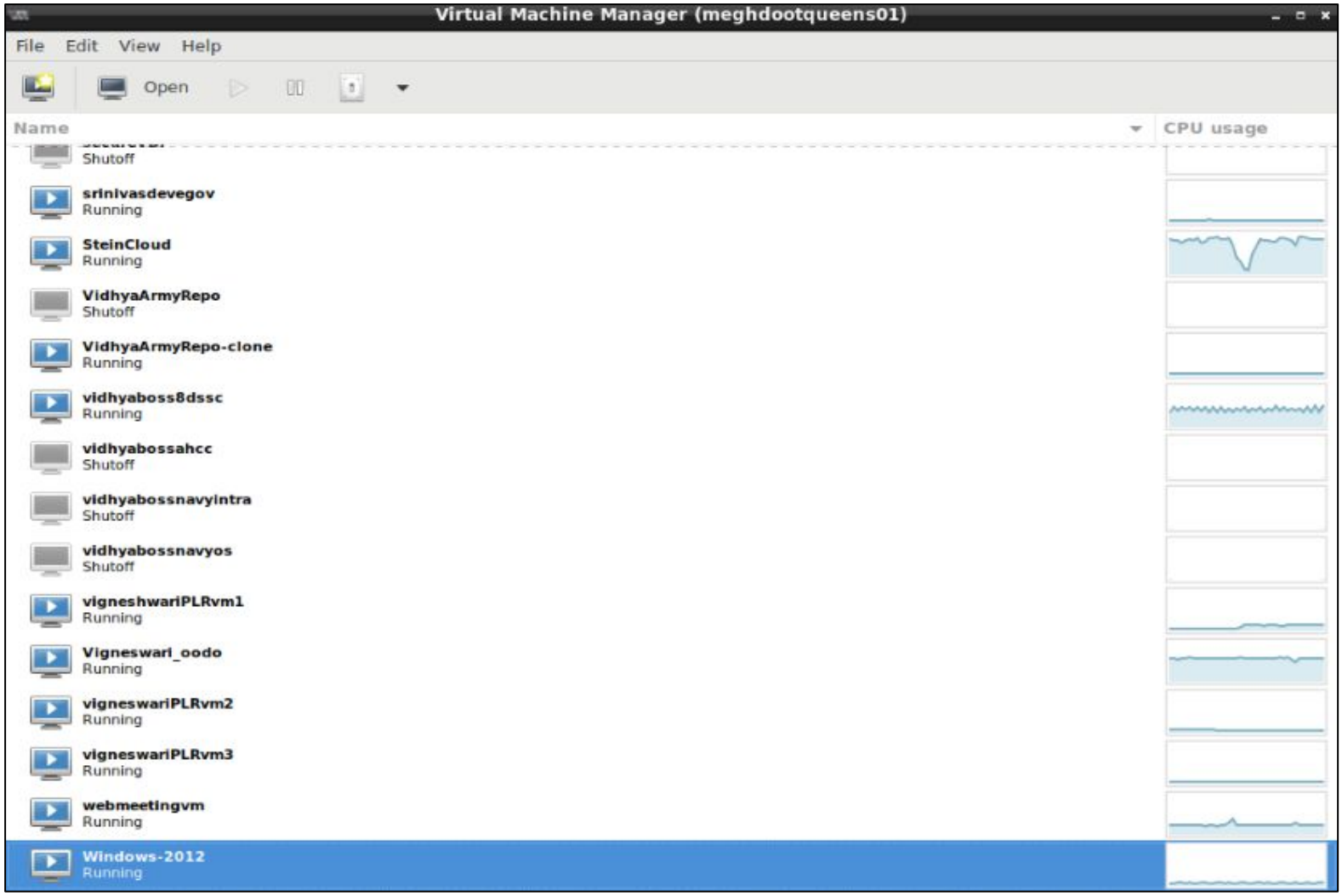
Create a VM using virt-manager

```
$ sudo virt-manager
```

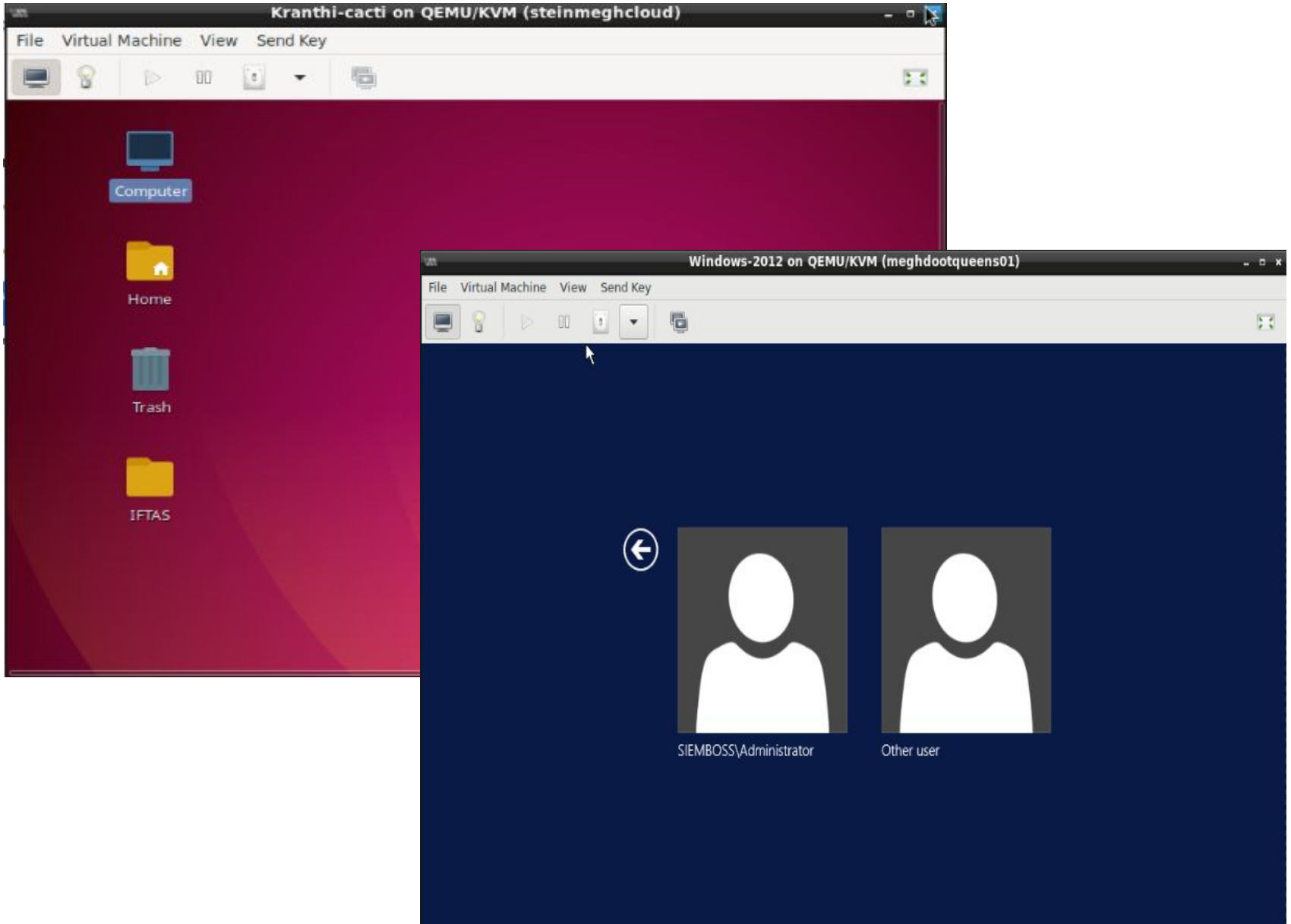
KVM Installation - Virt-manager



KVM Installation - Virt-manager



KVM Installation - Virt-manager



KVM VM creation using virt-install

Create a VM using virsh CLI

```
$ virt-install --name ubuntu --ram 2048 --disk \
path=/var/lib/libvirt/images/u19.qcow2, size=8 --vcpus 2 \
--os-type linux --os-variant generic --console pty, \
target_type=serial --cdrom \
/var/lib/libvirt/isos/ubuntu-18.04.4-live-server-amd64.iso
```

Clone the VM

```
$ virt-clone --original ubuntu --name cloned-ubuntu \
 --file /var/lib/libvirt/images/cu.qcow2
```

KVM Installation using QEMU

Create an image using Qemu

```
$ qemu-img create -f qcow2 ubuntu.img 10G
```

Create a VM using Qemu

```
$ qemu-system-x86_64 -enable-kvm -name ubuntu -m 2048 -hda
ubuntu.img -cdrom ubuntu-18.04.4-live-server-amd64.iso -boot d
-vnc:19
```

Image format Conversion

```
$ qemu-img convert  -O qcow2 ubuntu.img ubuntu.qcow2
```

KVM Installation using CLI

- Start network

  $ virsh net-start <network_name>

- List networks

  $ virsh net-list

- List VMs

  $ virsh list --all

- Start, reboot, and shutdown VM

  $ virsh start <VM>

  $ virsh reboot <VM>, and

  $ virsh shutdown <VM>

KVM - Other Alternatives

- KVMtool - Lightweight tool for hosting Guest VMs
- VMM rewritten in Rust programming Language listed below

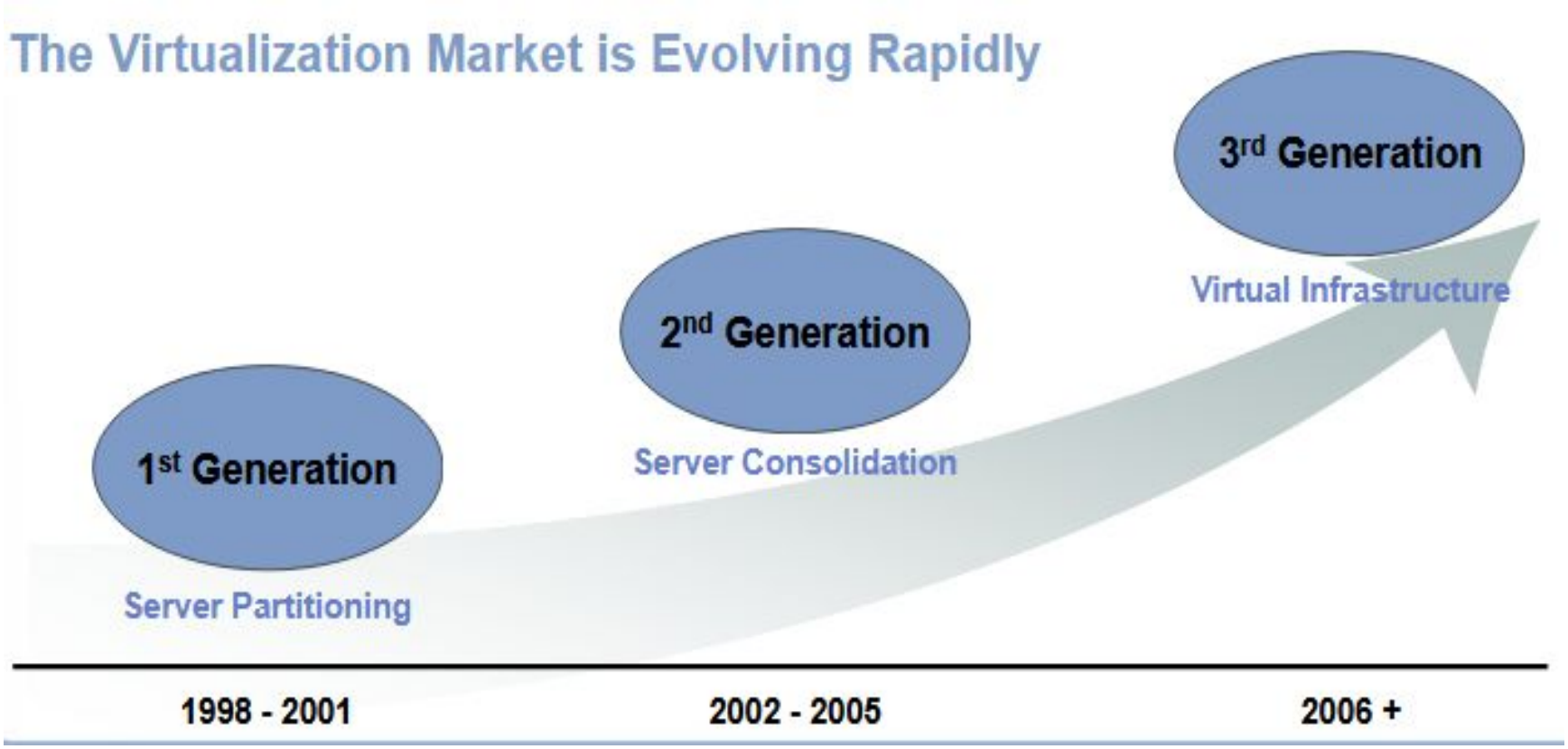
CrosVM	KVM based VMM developed by Google to run Linux applications inside ChromeOS.
Firecracker	VMM developed by Amazon for running Lambda functions
Rust-VMM	a collaboration by Amazon to develop common libraries for virtualization projects written in Rust
cloud-hypervisor	The rust-vmm reference implementation by Intel
Enarx	A platform abstraction for trusted execution environments (TEEs) using KVM developed by Redhat

VMWARE

VMware

- VMware, Inc. is an American cloud computing and virtualization technology company with headquarters in California.
- VMware was the first commercially successful company to virtualize the x86 architecture.
- In the late 1960s and early 1970s, VMware revisited the virtual machines that IBM pioneered for mainframe systems
- VMware changed the model of using proprietary OS for VMs in mainframe systems by enabling virtualization without requiring changes to industry-standard processors or operating systems.

VMware Generations



VMware Products

Desktop Virtualization software

- VMware Workstation
- VMware Fusion
- VMware Workstation Player

Server Virtualization Software

- VMware ESXi - bare metal virtualization
- VMware vCenter

Storage Virtualization Software

- VMware vSAN - is software-defined storage that in VMware's ESXi hypervisor
- VMware Site Recovery Manager(SRM) for DR

VMware Products

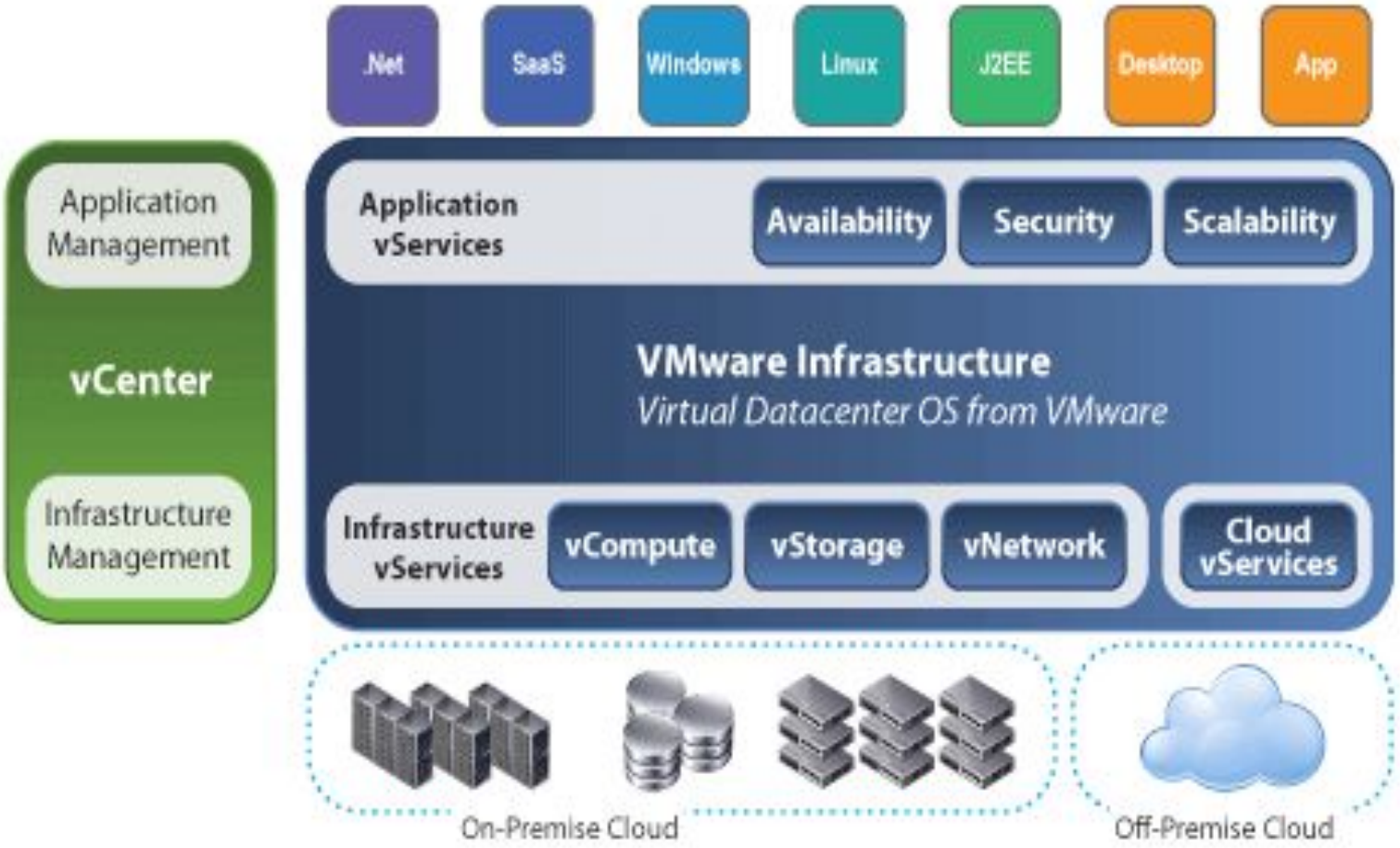
Cloud Management software

- VMware vRealize Suite - Hybrid Cloud Management Platform
- VMware GO - web based deployment service
- VMware Cloud Foundation - To deploy and operate a private cloud on an integrated SDDC system.
- VMware Horizon - Virtual Desktop Infrastructure (VDI)

vRealize Suite

VMware Cloud Foundation™

VMware Infrastructure



VMware vSphere Components

VMware vSphere is a term that encompasses the core virtualization solutions

Core Components of vSphere

- ESXi hypervisor
- vCenter Server
- vSphere Client

VMware vSphere Components

ESXi hypervisor

Type 1 bare metal hypervisor to manage host servers and run multiple guest VMs

vCenter Server

Management platform that enables the datacenter features, including ESXi clustering, vMotion, etc

vSphere Client

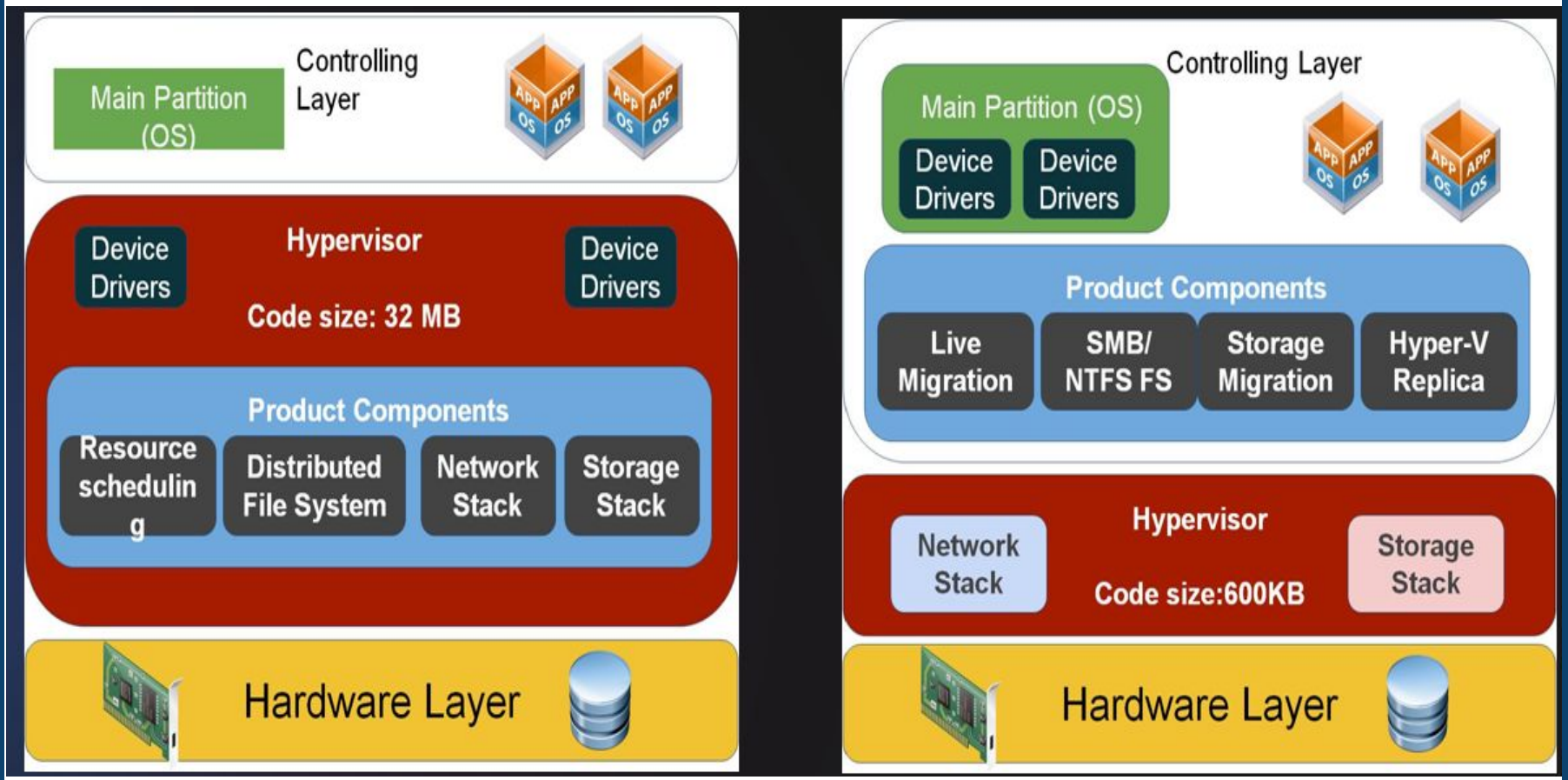
HTML5 driven management interface for managing, monitoring, and configuring vSphere and its associated plugins


प्रगत संगणन विकास केंद्र
सी डैक
CDAC
॥ ज्ञानादेव तु कैवल्यम्॥
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

THANKS

VMWARE vs HYPER-V

VMware[Monolithic] VS Hyper-V[Microkernel]



Monolithic design

Microkernel design

Hyper-V vs VMware

Hyper-V

Type I native hypervisor of Microsoft

Microkernel hypervisor

Partitions are divided into the parent and the child partitions.Two types of OS in child partition

- Enlightened -direct communication
- Unenlightened - Emulation

VMware

VMware ESX server is Type I native hypervisor of VMware

Monolithic hypervisor

High performance of ESXi is ensured by VMkernel. VMkernel runs directly on hosts and provides connection between VMs and the physical hardware

Hyper-V vs VMware

Management Tool
System Centre Virtual Machine Manager
Storage Deployment
Microsoft ReFS

Virtual Disk Image Format
VHD/VHDX
Clustered File system
Resilient File system(ReFS)

Guest OS Services
Microsoft Integration Services
Snapshot Technology
Checkpoints

Management Tool
VMware vCenter Server

Storage Deployment
vSphere VMFS
Virtual Disk Image Format
VMDK
Clustered File system
Virtual Machine File System (VMFS)

Guest OS Services
VMware Tools
Snapshot Technology-
Snapshots

Hyper-V vs VMware

Change Tracking
Resilient Change Tracking(RCT)

Memory Management
Dynamic Memory

Workload Migration
LiveMigration

Security
Guarded Fabric,Host Guardian Service and Shielded VMs

Change Tracking
Changed Block Tracking(CBT)

Memory Management
Over subscription

Workload Migration
vMotion

Security
Encrypted vMotion