SERVE
VIRTUALIZ

-Priyal Lokhandwala

Contents:

1) Virtualization- an overview
2) Server Virtualization
3) VM ware
4) Hyper V
5) Server Deployment

virtualization is a broad term that refers to the abstraction of computer resources

"*Virtualization is the highest impact trend changing infrastructure and operations through 2012. It will change how you manage, how and what you buy, how you deploy, how you plan, and how you charge* ."

- Gartner's Special Report 2008

Types of virtualization:

1) PC virtualization- the decoupling technology that breaks the close ties and dependencies between hardware and software

- Platform/machine/OS virtualization-between hardware and the operating system
-Apllication Virtualization-between the operating system and applications

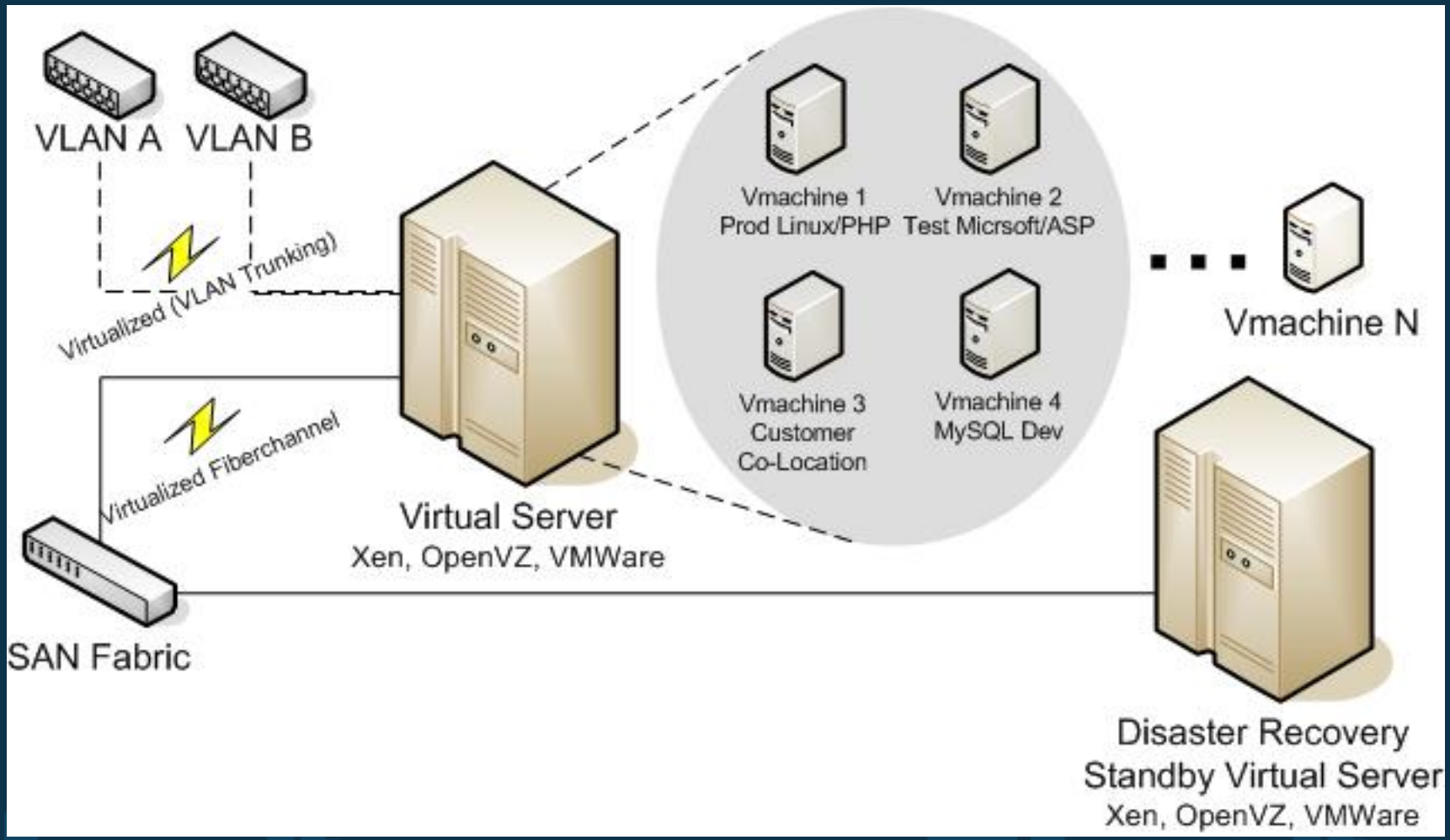
2) Resource virtualization- abstraction of specifc system resources

-Encapsulation - the hiding of resource complexity by the creation of a simplified interface
-Storage Virtualization- abstracting physical from logical storage.

4) Server Virtualization

Server virtualization is the masking of server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. The server administrator uses a software application to divide one physical server into multiple isolated virtual environments. The virtual environments are sometimes called virtual private servers, but they are also known as partitions, guests, instances, containers or emulations.

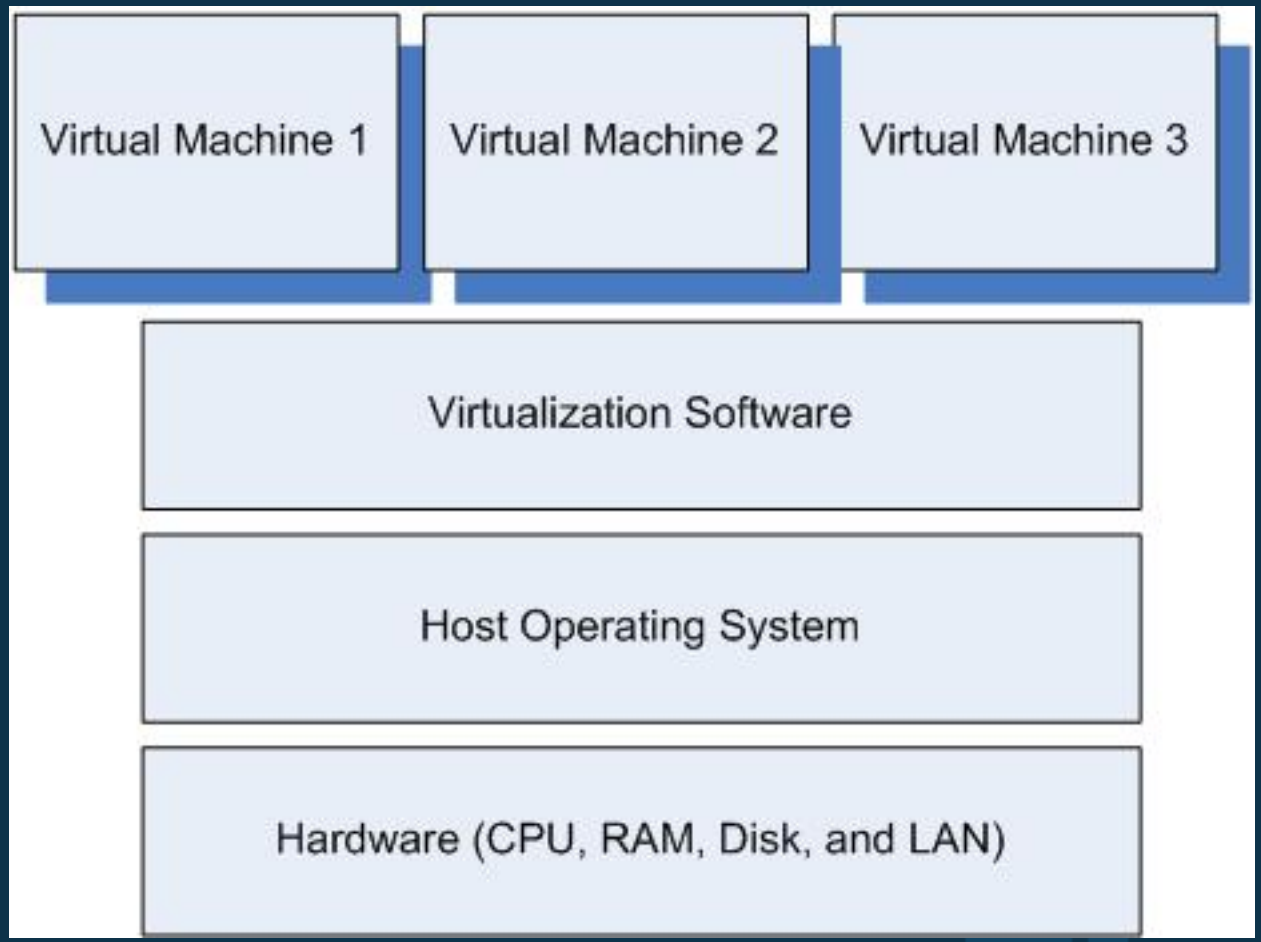
Virtualization Architecture:



Approaches to Server Virtualization:

1) Virtual Machine Model Approach
2) ParaVIrtual Machine Model Approach
3) Virtualization at OS layer

Server Virtualization Architecture:



VIRTUAL MACHINE MODEL -VMM

Virtual machines are based on the host/guest paradigm. Each guest runs on a virtual imitation of the hardware layer. This approach allows the guest operating system to run without modifications. It also allows the administrator to create guests that use different operating systems. The guest has no knowledge of the host's operating system because it is not aware that it's not running on real hardware. However, it does require real computing resources from the host, for which it uses a "hypervisor" also called a virtual machine monitor to coordinate instructions to the CPU. It validates all the guest-issued CPU instructions and manages any executed code that requires addition privileges.

By encapsulating an entire machine, including CPU, memory, operating system, and network devices, and a virtual machine is completely compatible with all standards x86 operating systems, applications, and device drivers. We can safely run several operating systems and applications at the same time on a single computer, with each having access to the resources it needs when it needs them.

Example: VMware and Microsoft's Virtual Server, Hyper-V

PARA VIRTUAL MACHINE MODEL -PVM

-The paravirtual machine (PVM) model is also based on the host/guest paradigm
-it uses a virtual machine monitor too.
-In this model however, The VMM actually modifies the guest operating system's code.
-This modification is called PORTING.
-Porting supports the VMM so it can utilize privileged systems calls sparingly.
-Like virtual machines, PVMs are also capable of running multiple operating systems.

Examples: Xen and UML.

SERVER VIRTUALIZATION AT OS LEVEL

-not based on host guest paradigmn
-In the OS level model, the host runs a single OS kernel as its core and exports operating system functionality to each of the guests.
-Guests must use the same operating system as the host, although different distributions(containers) of the same system are allowed.
-On Unix systems, this technology can be thought of as an advanced implementation of the standard chroot mechanism.
-In addition to isolation mechanisms, the kernel often provides resource management features to limit the impact of one container's activities on the other containers.

Example: FreeVPS and Solaris Zones.

Hypervisor based Architectures:

Difference between microft's Virtual Server(MVS), Hyper-V and VMware's ESX

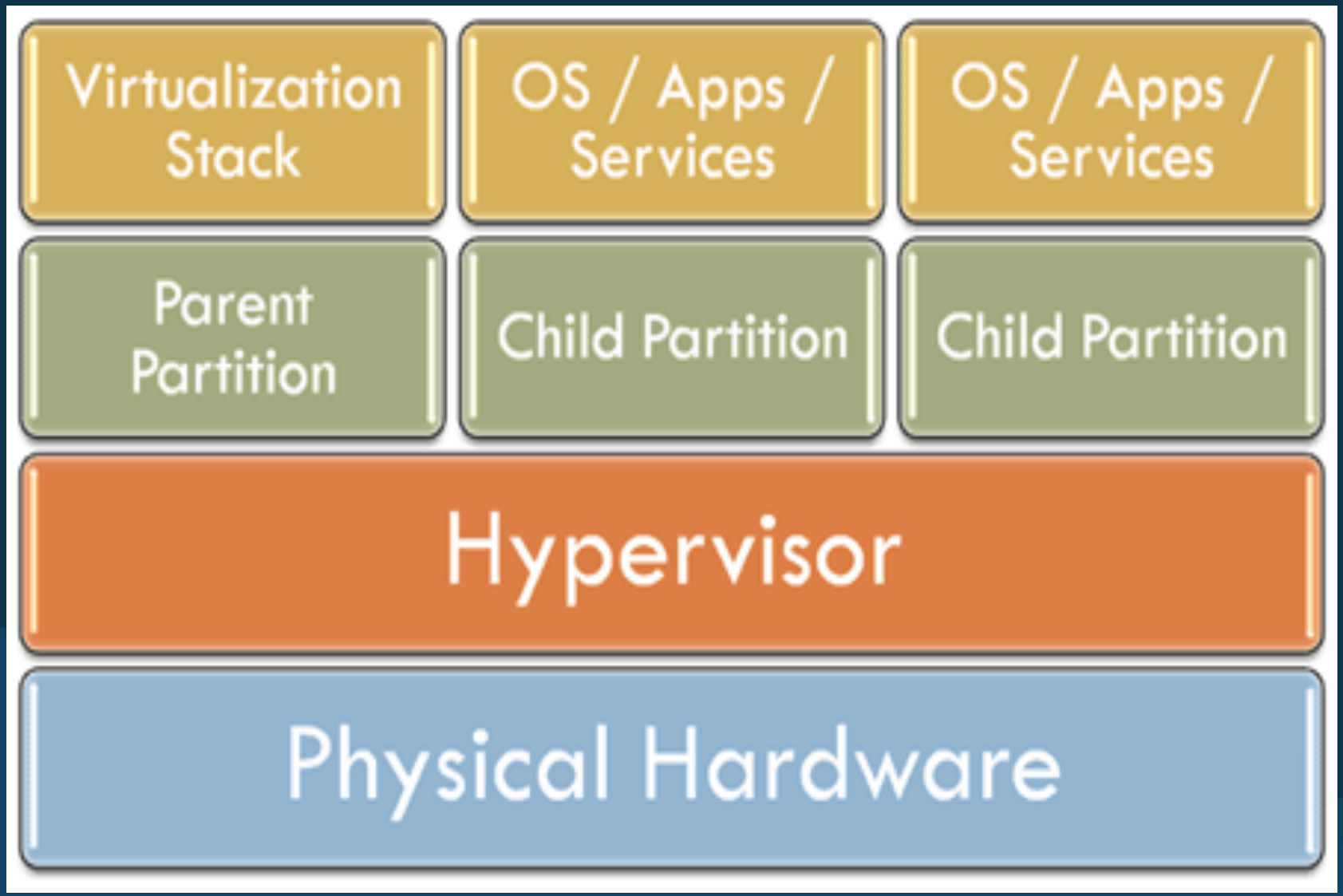
MVS is known as a "Type-2" virtualization architecture. This means that MVS runs as an application that is installed on top of an existing operating system (OS).

Hyper-V, which is called a Microkernelized Type-1 architecture.

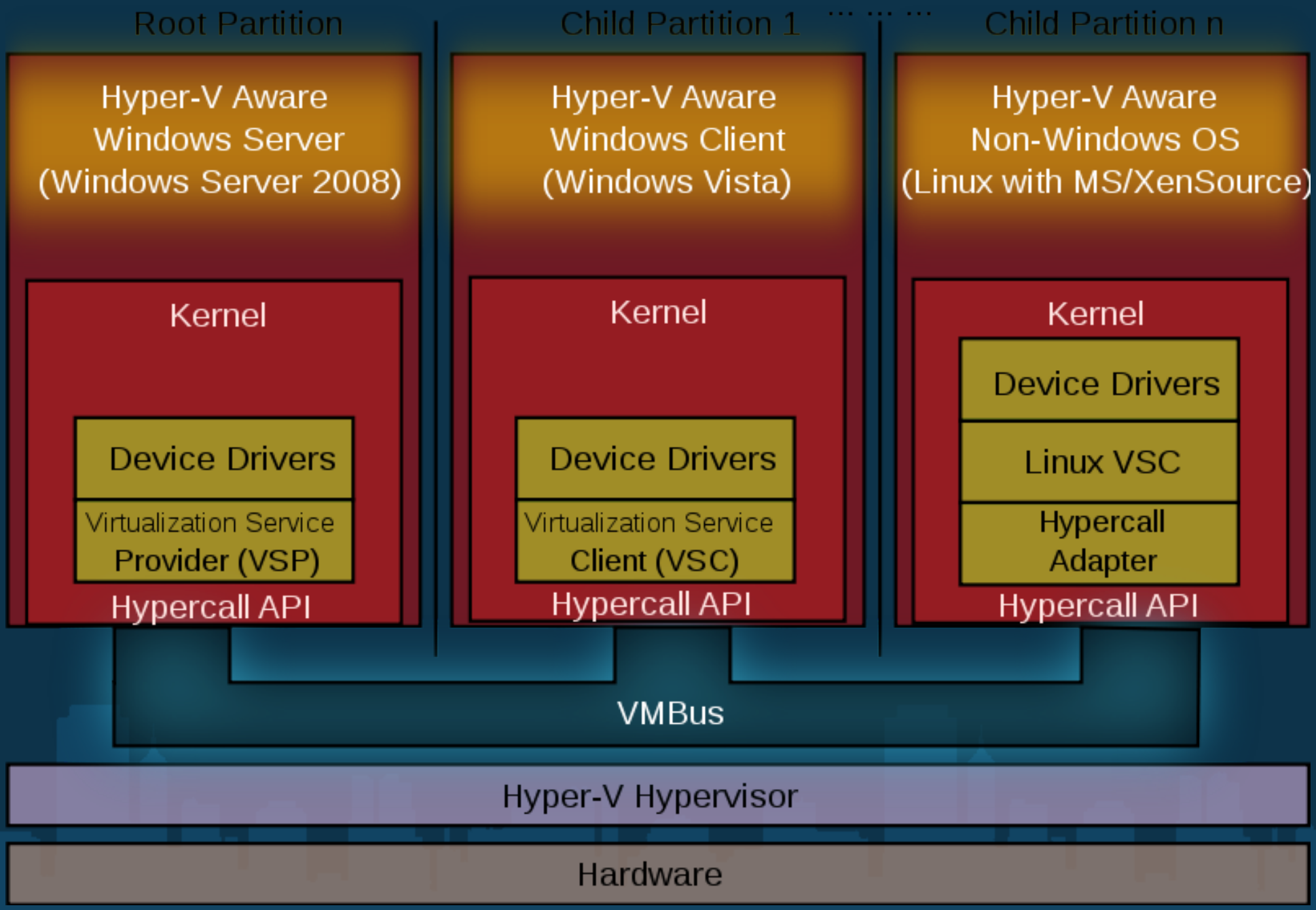
-"Type-1" in this case means that the virtualization layer actually sits directly on top of the physical hardware instead of above a regular OS.

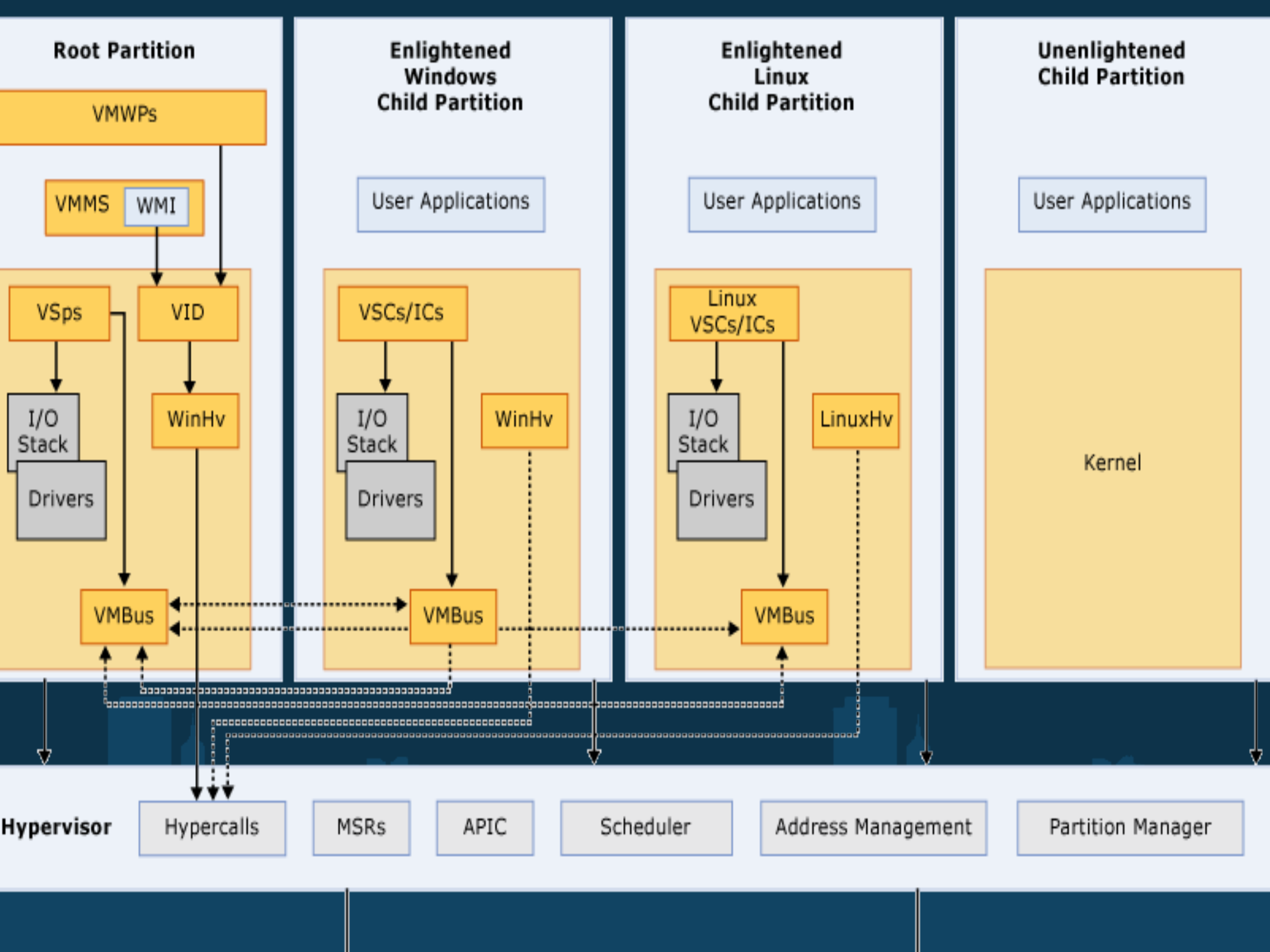
-"Microkernelized" means that it's an extremely thin hypervisor that is made even thinner (thinner than what you see with products like VMware's ESX) because it contains no drivers. In fact, Hyper-V's hypervisor is around 600kb in size

HYPER-V aka VEREDIAN ARCHITECTURE:



A MORE DETAILED VIEW:




Root Partition
Enlightened Windows Child Partition
Enlightened Linux Child Partition
Unenlightened Child Partition
VMWPs
VMMS
WMI
User Applications
User Applications
User Applications
VSps
VID
VSCs/ICs
Linux VSCs/ICs
I/O Stack
Drivers
WinHv
I/O Stack
Drivers
WinHv
I/O Stack
Drivers
LinuxHv
Kernel
VMBus
VMBus
VMBus
Hypervisor
Hypercalls
MSRs
APIC
Scheduler
Address Management
Partition Manager

REQUIREMENTS: (WINDOWS SERVER 2008 WITH HYPER V)

1) 64 BIT PLATFORM
2) HADWARE ASSISTED VIRTUALIZATION
- virtualization technology inserted in the chips.
3) NO EXECUTE BIT ENABLED IN BIOS and COLD REBOOT
4) MINIMUM TWO NETWORK ADAPTERS
-one for management network (parent partition)
-one(or more) for VM networking.

CAPABILITIES:

1) SUPPORT FOR BOTH 32 BIT AND 64 BIT VMs
2) support for large memory- 64GB per VM
3) SMP -Some Multiple processors-4 cores per VM
4) Live backups- Volume shadow copy services
5) Synthetic Device Model- enlightened client
6) Live Snapshots of VM
7) Robust Support for VLAN
8) DTMF standard for WMI Managment Interface.
9) Built on Windows Server 2008-Enables Hyper-V to support a multitude of devices and

Security

Isolation

- No sharing of virtualized devices
- Separate VMBus instance per vm to the parent
- No sharing of memory
  - Each has its own address space
- VMs cannot communicate with each other, except through traditional networking
- Guests can't perform DMA attacks because they're never mapped to physical devices
- Guests cannot write to the hypervisor
- Parent partition cannot write to the hypervisor

Metric	VMware	Microsoft	Citrix
• Hypervisor Operation Overhead	Negligible	1 CPU Core	1 CPU Core
• Maximum Memory (Host)	256 GB	32 GB to 2 TB*	128 GB
• RAM for Hypervisor	32 MB+**	512 MB +***	256 to 512 MB +
• Maximum CPU Sockets (Host)	32 cores	24 cores	Unlimited****
• Maximum CPU Sockets (Guest)	4	4*****	8
• Required Management NIC(s)	1	1	1
• Maximum number of servers			

Why use Server Virtualization

1) Saves money
2) Saves resources
3) Makes migration, backup and recovery easy and manegable.

The End