# Virtualization

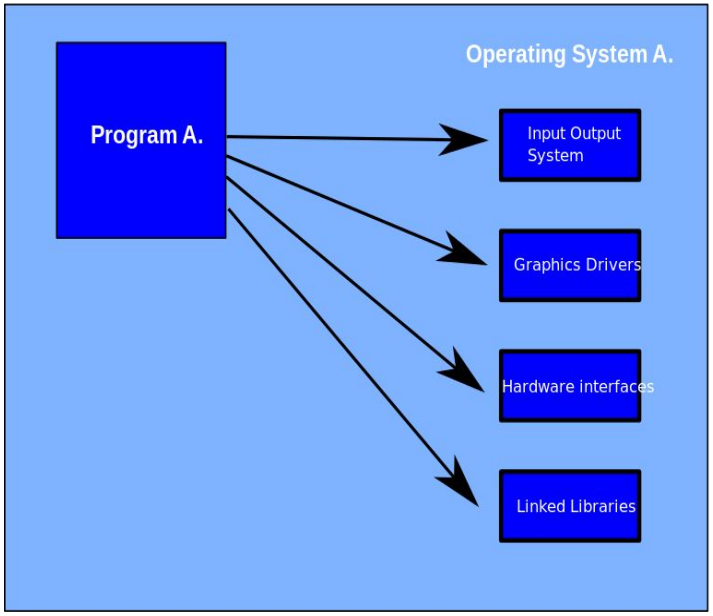
## Module 3 – Part B

# Application Level Virtualization

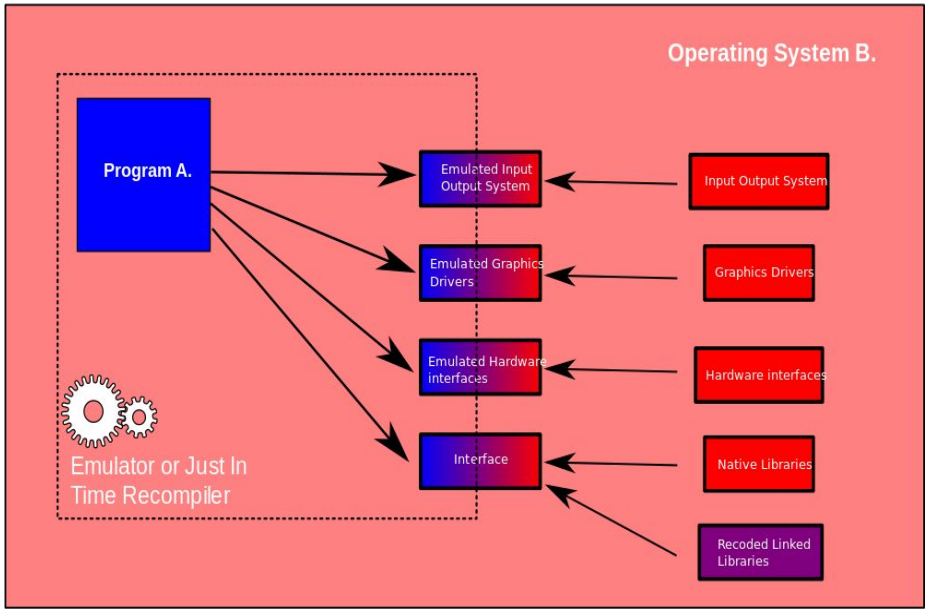
- The application-level virtualization works where there is a desire to virtualize only one application.
- One does not require to virtualize the complete environment of the platform.
- Application virtualization delivers apps to a user's computer independently of the local desktop without having to access the apps on their physical device.
- In this scenario, applications are not installed in the expected runtime environment but are run as though they were.
- Allows applications to be run in runtime environments that do not natively support all the features required by such applications.

***Application virtualization allows*** *applications to run in environments that do not suit the native application. For example, Wine allows some Microsoft Windows applications to run on Linux.*

1. Application in Native Environment




2. Application in Non-Native Environment



# Contd…

- In general, these techniques are mostly concerned with partial file systems, libraries, and operating system component emulation. Such emulation is performed by a thin layer—a program or an operating system component—that is in charge of executing the application.
- Application virtualization is an essential component of a complete desktop virtualization strategy and it applies to:
  - ✔ *Cloud migration services*
  - ✔ *Both software and hardware*
  - ✔ *Servers*
  - ✔ *Storage*

# Contd…

- Emulation can also be used to execute program binaries compiled for different hardware architectures. In this case, one of the following strategies can be implemented:

- **Interpretation.** In this technique, every source instruction is interpreted by an emulator for executing native ISA instructions, leading to poor performance. Interpretation has a minimal startup cost but a huge overhead, since each instruction is emulated.
- **Binary translation.** In this technique, every source instruction is converted to native instructions with equivalent functions. After a block of instructions is translated, it is cached and reused. Binary translation has a large initial overhead cost, but over time it is subject to better performance, since previously translated instruction blocks are directly executed.

# Contd…

- Emulation, as described, is different from hardware-level virtualization. The former simply allows the execution of a program compiled against a different hardware, whereas the latter emulates a complete hardware environment where an entire operating system can be installed.
- Application virtualization is a good solution in the case of missing libraries in the host operating system.
- In this case, a replacement library can be linked with the application, or library calls can be remapped to existing functions available in the host system.
- Another advantage is that in this case the virtual machine manager is much lighter since it provides a partial emulation of the runtime environment compared to hardware virtualization.

# Contd…

- Moreover, this technique allows incompatible applications to run together.
- Application-level virtualization works for a specific environment. It supports all the applications that run on top of a specific environment.
- One of the most popular solutions implementing application virtualization is **Wine**, which is a software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform.

# Storage Virtualization

Problems with Traditional Storage

**Storage is Physical**

- ✔ Connection and Presentation
- ✔ Power and cooling
- ✔ Access and configuration
- ✔ Results in reboots, complexity, downtime, and finally money

**Multiple Management System**

- ✔ Inconsistent, incompatible, and incomplete

**Typical Storage Utilization**

- ✔ Disk utilization is low

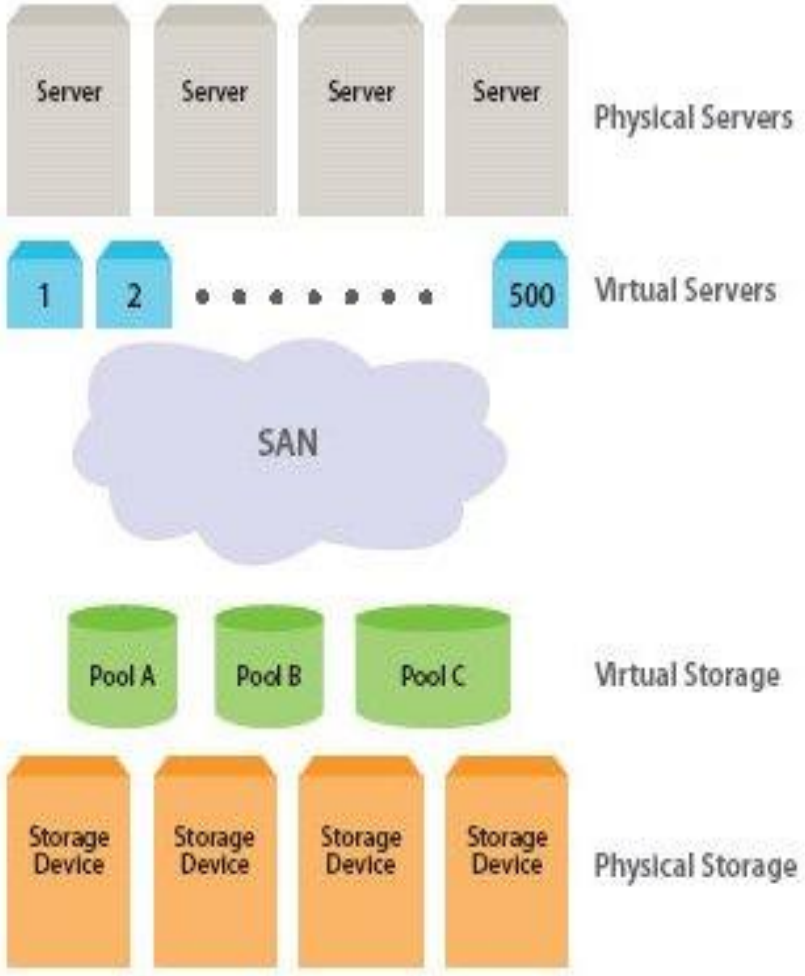
**Availability Requirement**

**High storage management cost**

# Storage Virtualization

- Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation.
- Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.
- Storage virtualization allows us to harness a wide range of storage facilities and represent them under a single logical file system.
- There are different techniques for storage virtualization, one of the most popular being network-based virtualization by means of storage area networks (SANs).
- SANs use a network-accessible device through a large bandwidth connection to provide storage facilities.

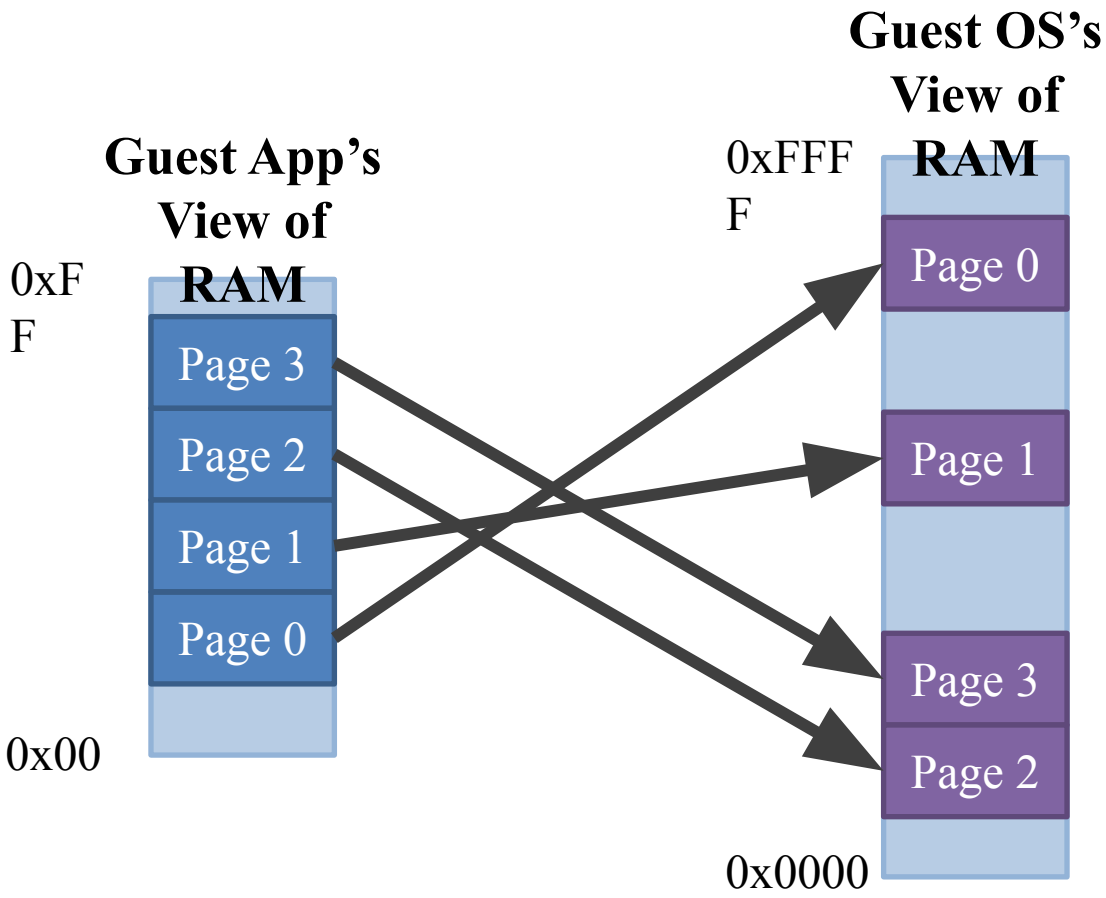
# Contd…


Server
Server
Server
Server
Physical Servers
1
2
500
Virtual Servers
SAN
Pool A
Pool B
Pool C
Virtual Storage
Storage Device
Storage Device
Storage Device
Storage Device
Physical Storage

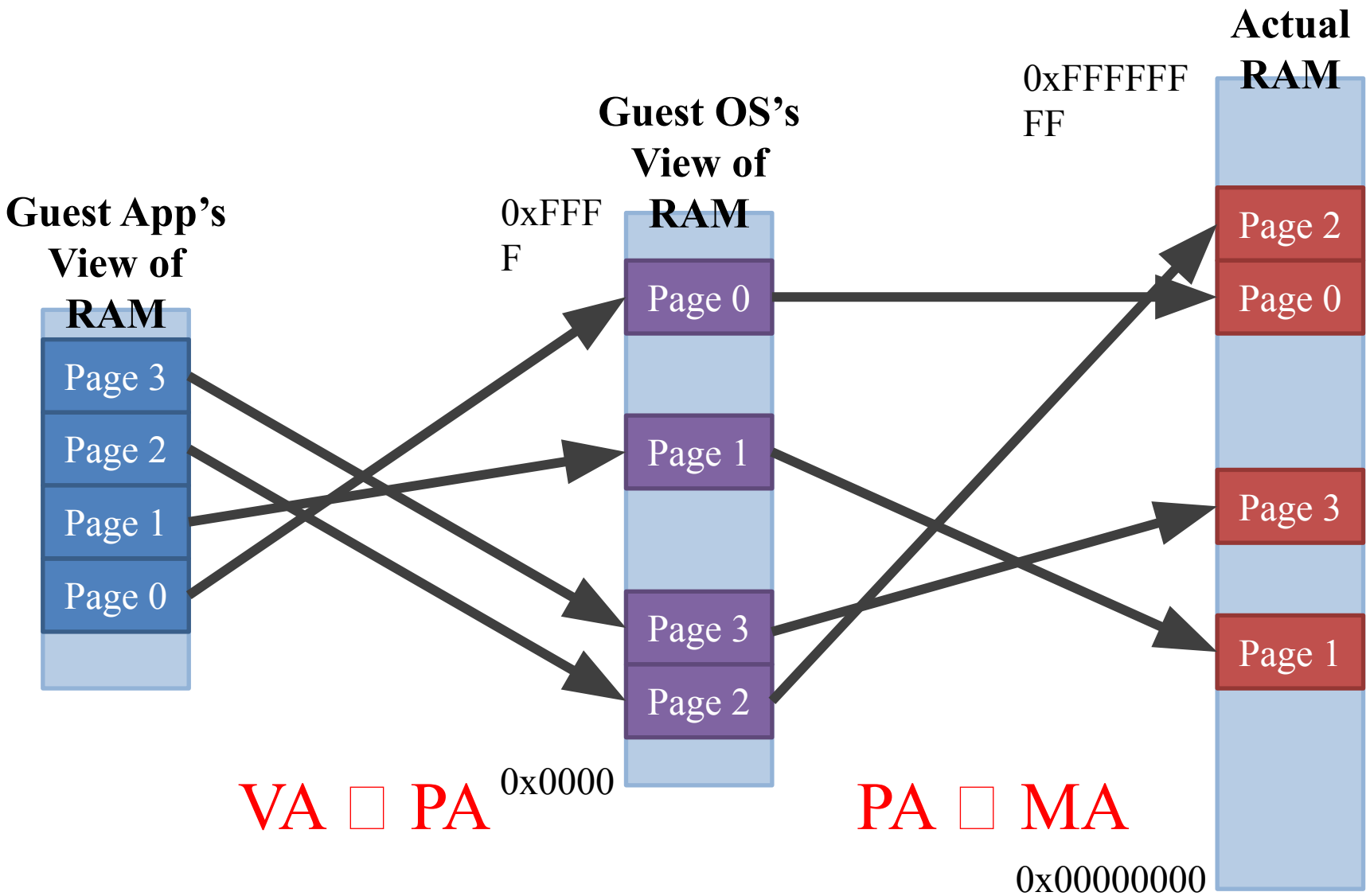
# Memory Virtualization

- Typical x86 architecture has a virtual to physical address mapping (VA □ PA)
- Virtualized x86 architecture requires a two level address translation
  - VA □ PA
  - Physical address (PA) □ Machine Address (MA)
- Guest OS has no idea about this translation
  - Guest continues to maintain page tables containing VA □ PA mappings

# Paging without Virtualization



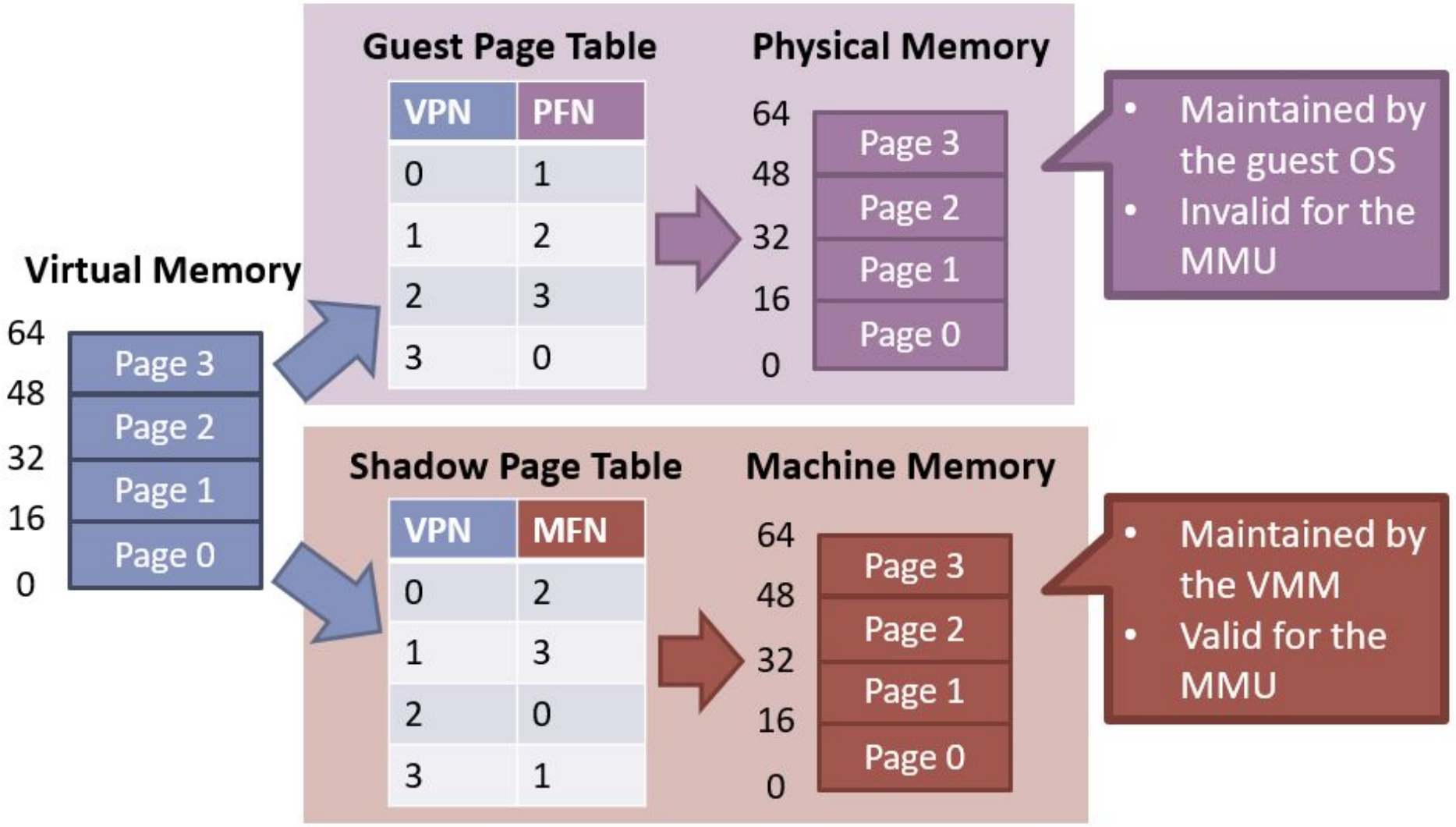
# Paging with Virtualization



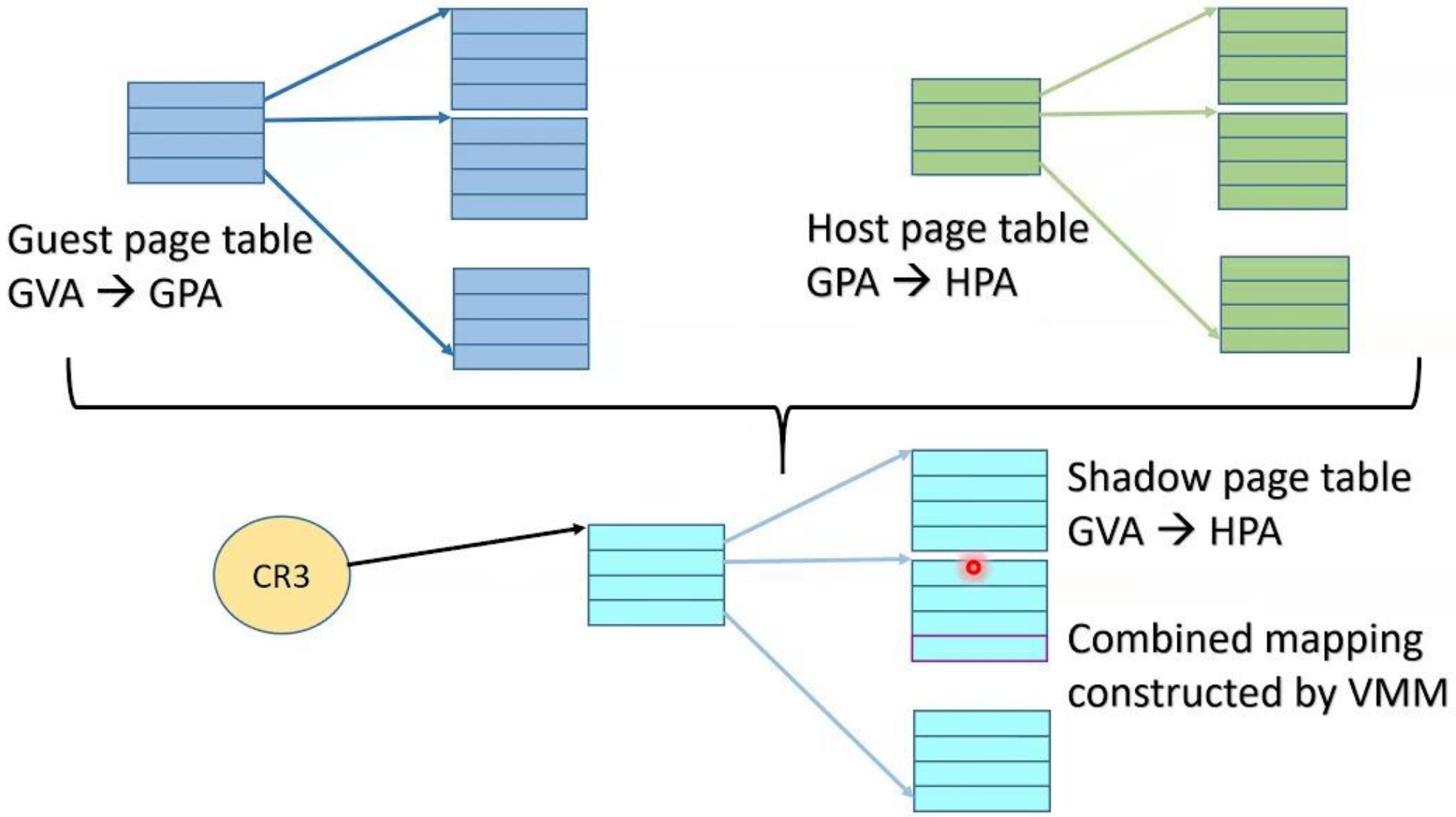
# Does a 2 Level Indirection Work?

- Guest is only aware of VA □ PA mapping
  - Issues only VA to hardware MMU
  - **MMU supports only a single mapping**
- Solution: Shadow Page Table
  - Hypervisor maintains a single shadow page table in the MMU
  - Shadow page table contains direct VA □ MA mapping
  - Trick is to maintain consistency!!!

# Shadow Page Tables



# Shadow page tables



# Building Shadow Page Tables

- The guest can update its page tables at any time
  - Not a privileged instruction – not trapped!
  - Without knowing when the guest OS updates its page table, the hypervisor cannot maintain the correct entry in the shadow page table
- Solution: Mark the guest page tables as read only
  - Writing generates an exception, which can be trapped by hypervisor

# What happens during page faults?

- Two kinds of page faults can occur:
  - **True Miss :** The mapping does not exist in the guest page table
  - **Hidden Miss :** The mapping exists in the guest page table, but is absent in the shadow page table
- The hypervisor should disambiguate between the two
- On every miss, the hypervisor walks the guest page table [Tracing]
  - If a mapping exists, the hypervisor silently updates the shadow page table and retries the instruction
  - Otherwise, the hypervisor forwards the page fault to the guest OS for handling

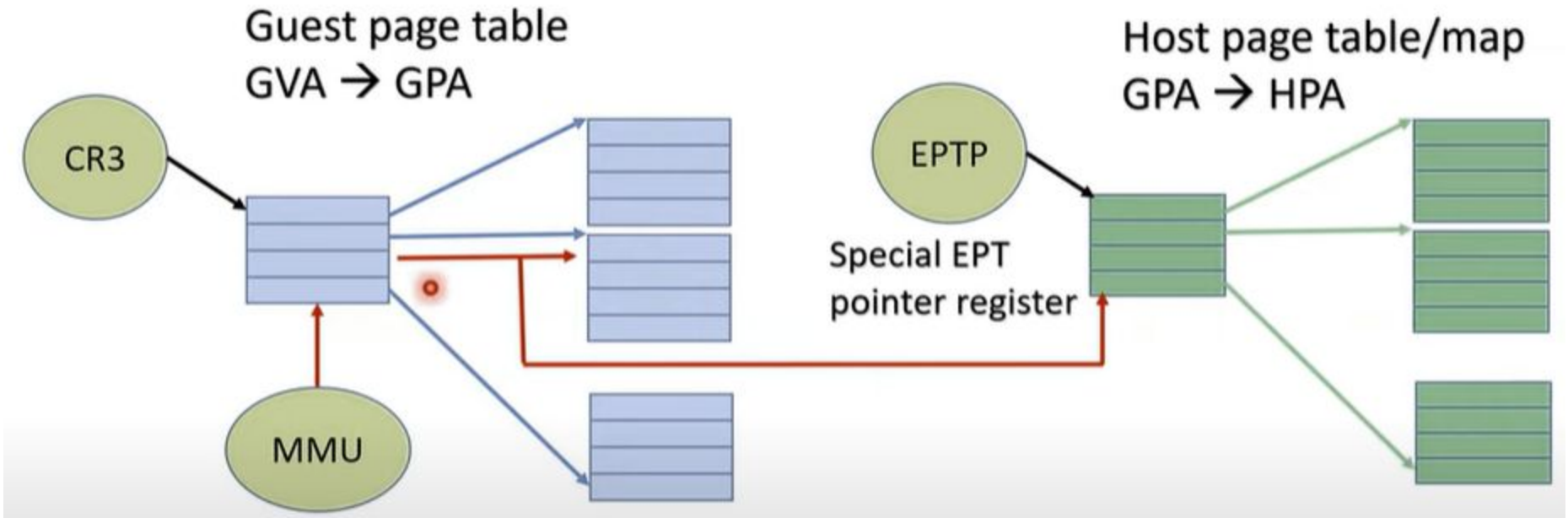
# Pros and Cons

- The good: shadow tables allow the MMU to directly translate guest VPNs to hardware pages
  - Thus, guest OS code and guest apps can execute directly on the CPU
- The bad:
  - Double the amount of memory used for page tables
    - i.e. the guest's tables and the shadow tables
  - Overhead due to VMM traps

# Second Level Address Translation (SLAT)

- Hardware support for memory virtualization
  - Extended Page Tables (EPT) – Intel
  - Nested Page Tables (NPT) - AMD
- Walking the guest and host page tables can be combined into a single multilevel page table
  - Page table depth increases tremendously in some cases!!!
  - Extremely important to use an effective TLB to avoid expensive page table walks
- TLB modified to reduce miss rate
  - Larger TLB
    - More expensive, though!!!
  - Tagged TLB
    - Every entry in TLB now has an address space identifier

# Extended Page Table



# Network Virtualization

- Network virtualization combines hardware appliances and specific software for the creation and management of a virtual network.
- Network virtualization can aggregate different physical networks into a single logical network (external network virtualization) or provide network-like functionality to an operating system partition (internal network virtualization).
- The result of external network virtualization is generally a virtual LAN (VLAN).
- A VLAN is an aggregation of hosts that communicate with each other as though they were located under the same broadcasting domain.

# Contd…

- Internal network virtualization is generally applied together with hardware and operating system-level virtualization, in which the guests obtain a virtual network interface to communicate with.
- There are several options for implementing internal network virtualization:

✔ The guest can share the same network interface of the host and use Network Address Translation (NAT) to access the network;

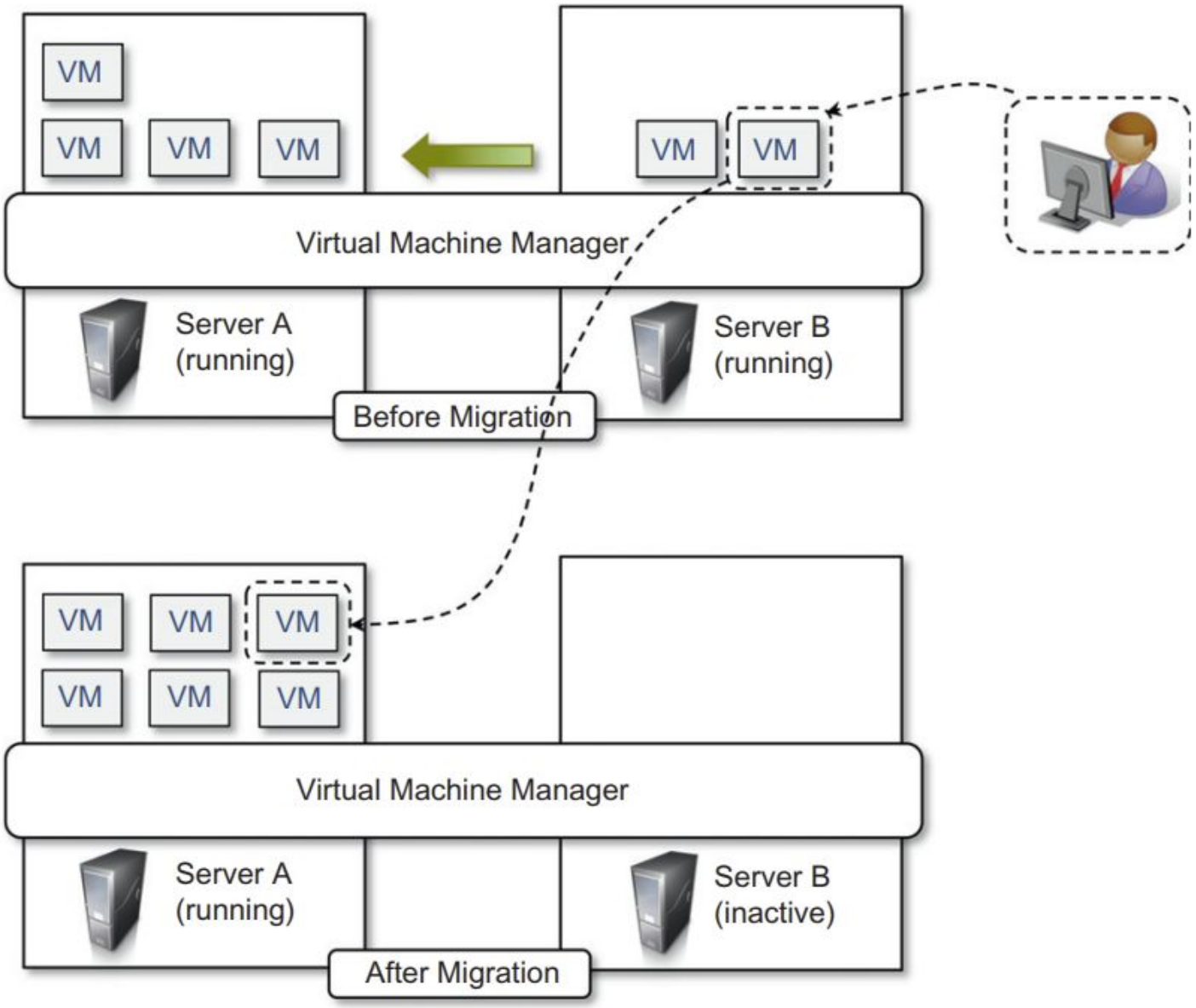
✔ The virtual machine manager can emulate, and install on the host, an additional network device, together with the driver;

✔ The guest can have a private network only with the guest.

# Desktop Virtualization

- It abstracts the desktop environment available on a personal computer in order to provide access to it using a client/server approach.
- It makes the same desktop environment accessible from everywhere.
- Strictly refers to the ability to remotely access a desktop environment.
- The desktop environment is stored in a remote server or a data center that provides a high-availability infrastructure and ensures the accessibility and persistence of the data.
- Virtual desktop infrastructure (VDI), Remote desktop services (RDS), and Desktop-as-a-Service (DaaS)

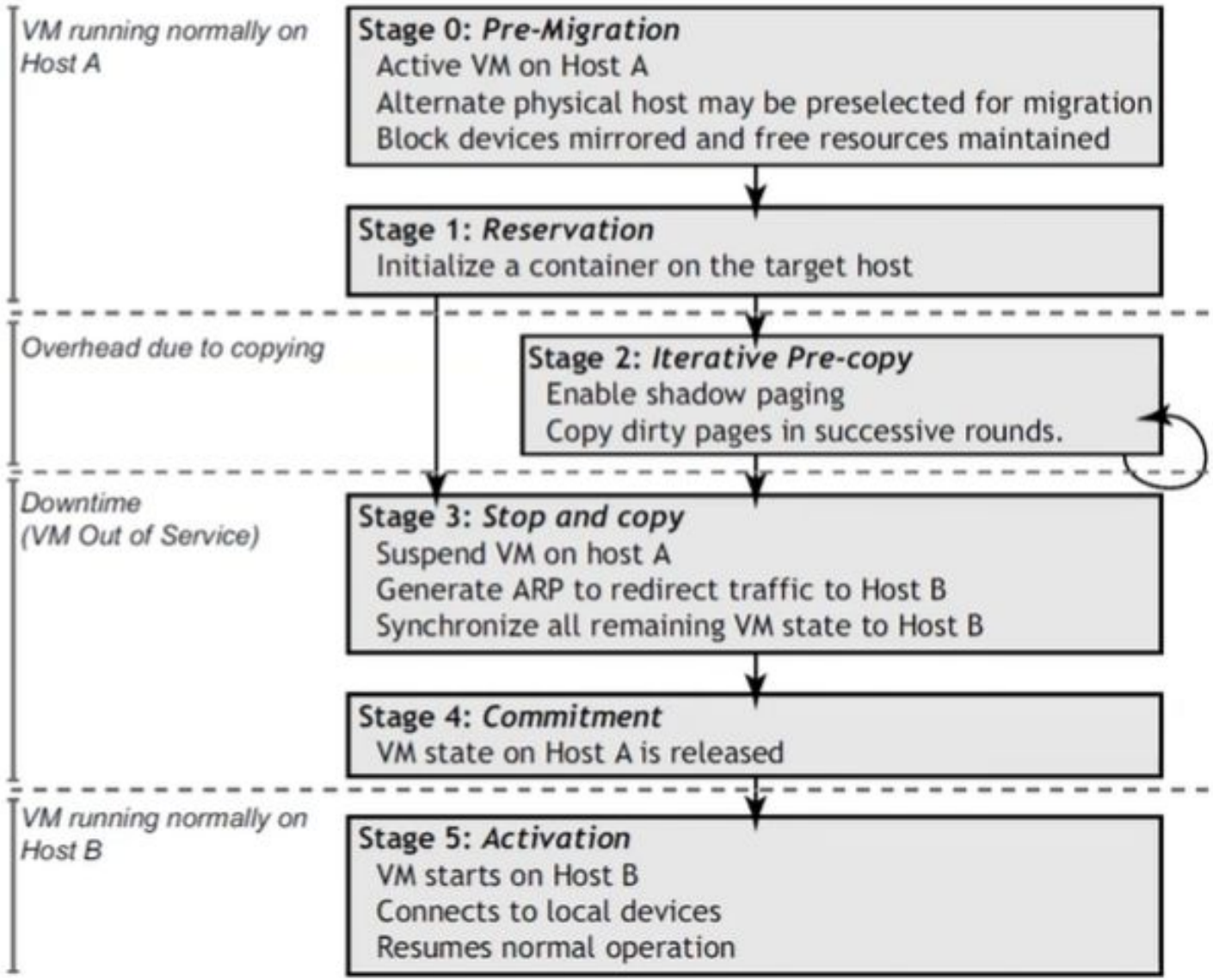
# Live Migration and Server Consolidation



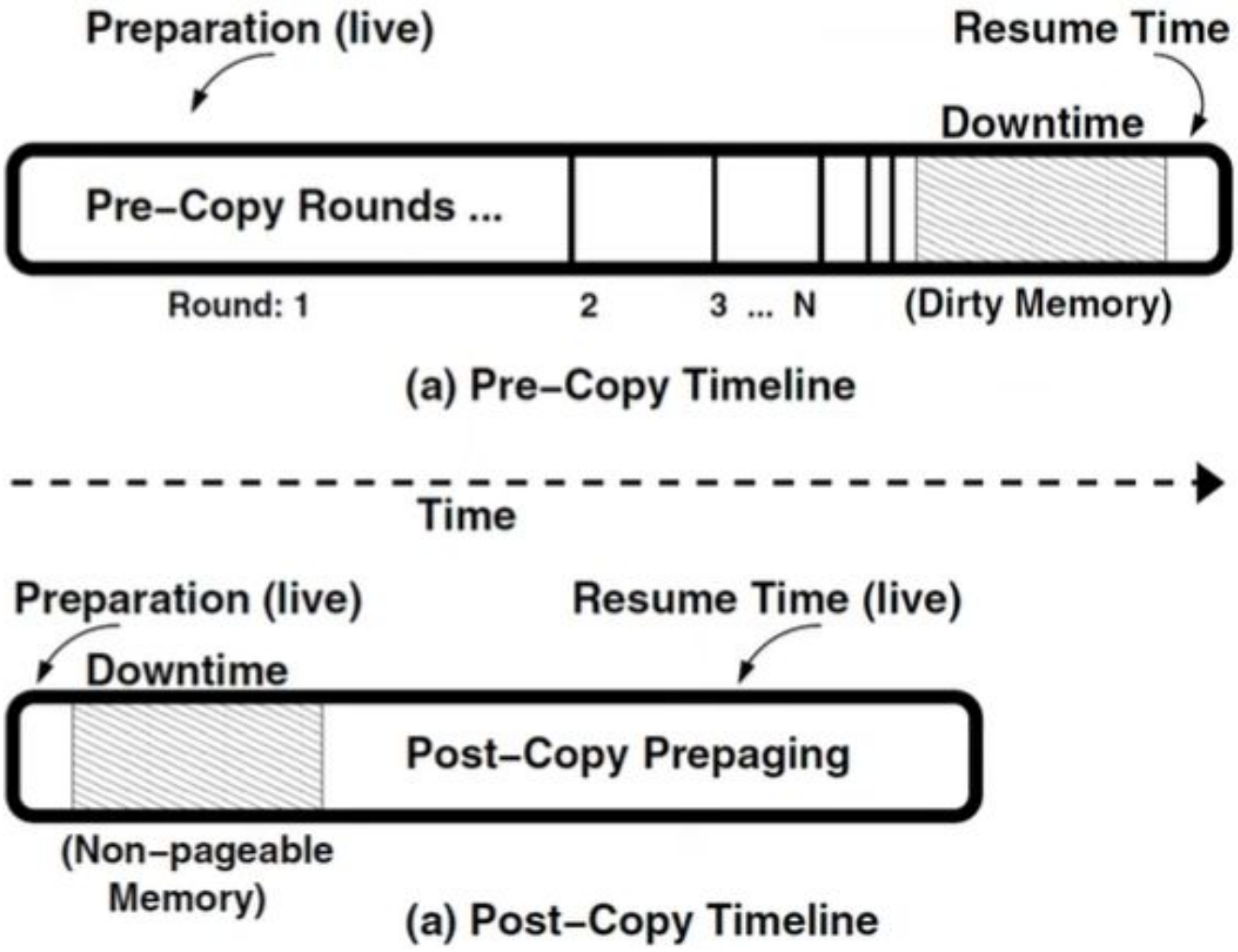
# Live Migration

- Broad steps in any migration technique: Suppose we are migrating a VM from host A to host B
  1. Setup target host B, reserve resources for the VM
  2. Push phase: push some memory of VM from A to B
  3. Stop-and-copy: stop the VM at A, copy CPU context, and some memory
  4. Pull phase: Start VM at host B, pull any further memory required from A
  5. Clean up state from host A, migration complete
- Total migration time: time for steps 2,3,4
- Service downtime: time for step 3
- Other metrics: impact on application performance, network bandwidth consumed, total pages transferred

# Pre-copy


VM running normally on Host A
Stage 0: Pre-Migration
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained
Stage 1: Reservation
Initialize a container on the target host
Overhead due to copying
Stage 2: Iterative Pre-copy
Enable shadow paging
Copy dirty pages in successive rounds.
Downtime (VM Out of Service)
Stage 3: Stop and copy
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B
Stage 4: Commitment
VM state on Host A is released
VM running normally on Host B
Stage 5: Activation
VM starts on Host B
Connects to local devices
Resumes normal operation

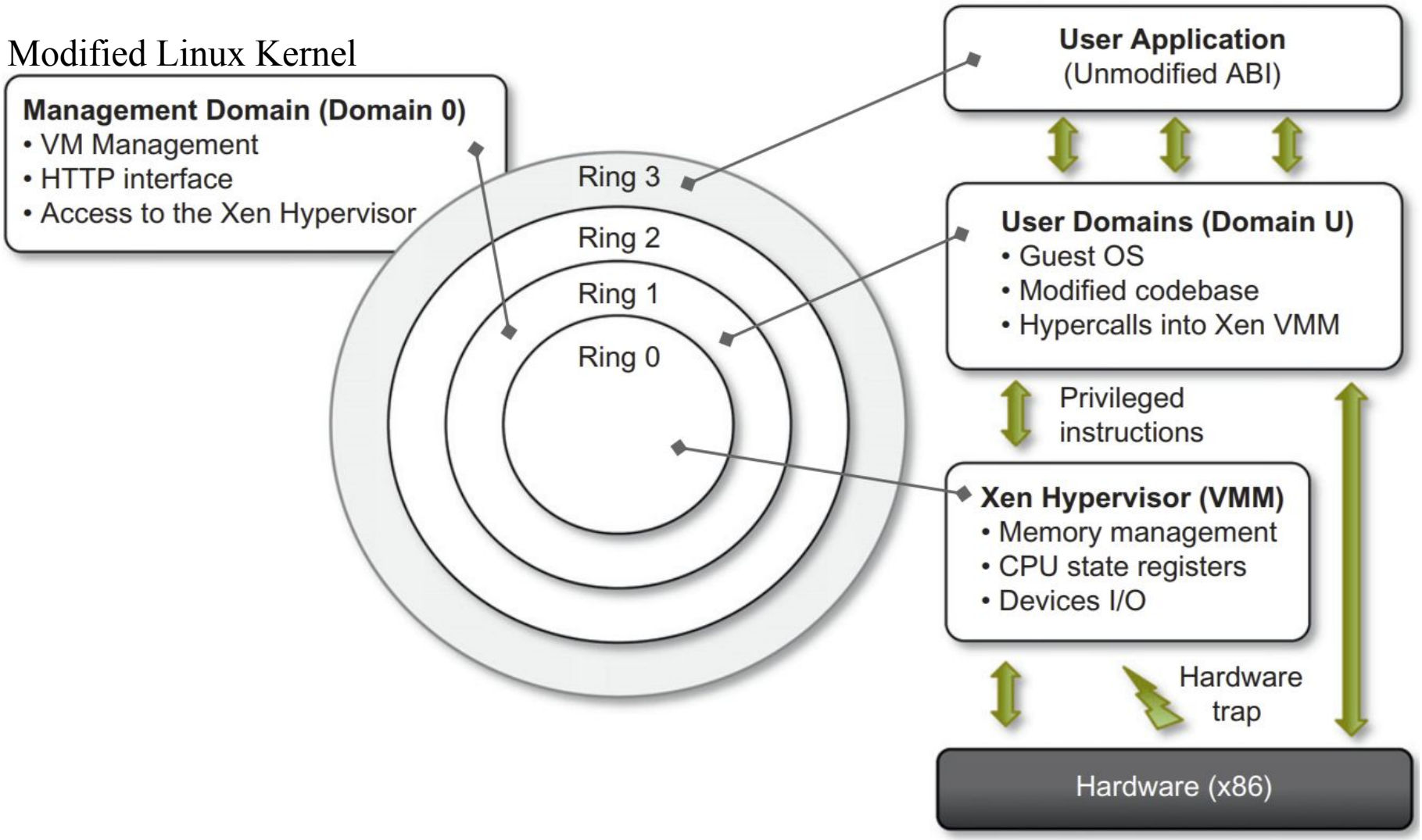
# Post-copy


Preparation (live)
Resume Time
Downtime
Pre–Copy Rounds ...
Round: 1
2
3 ... N
(Dirty Memory)
(a) Pre–Copy Timeline
Time
Preparation (live)
Resume Time (live)
Downtime
Post–Copy Prepaging
(Non–pageable Memory)
(a) Post–Copy Timeline

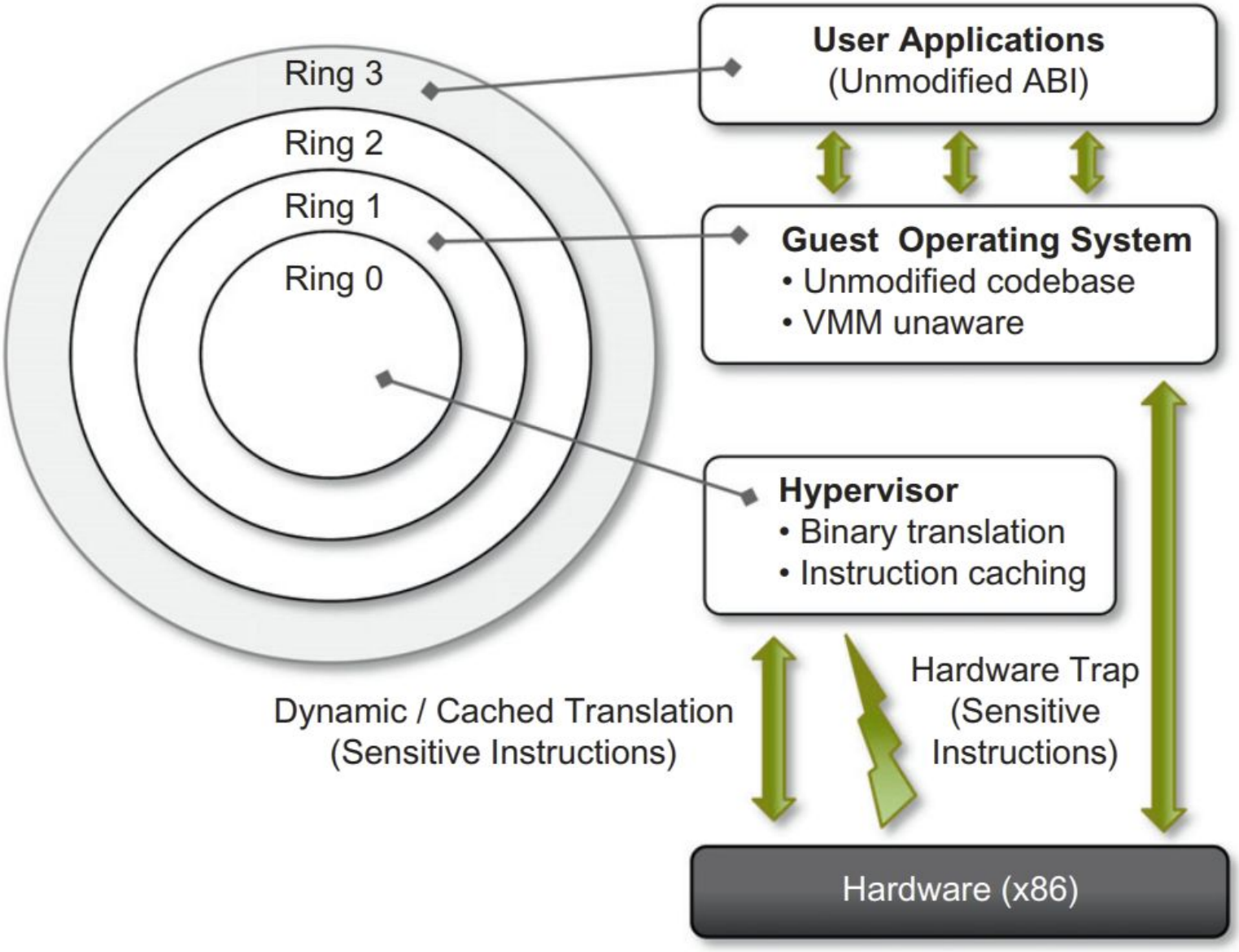
# Xen Hypervisor

- Xen is an open-source virtualization platform based on **paravirtualization.**
- Initially developed by a group of researchers at the University of Cambridge in the UK.
- Xen-based technology is used for either desktop virtualization or server virtualization.
- It has also been used to provide cloud computing solutions by means of Xen Cloud Platform (XCP).

# Xen Architecture and Guest OS Management

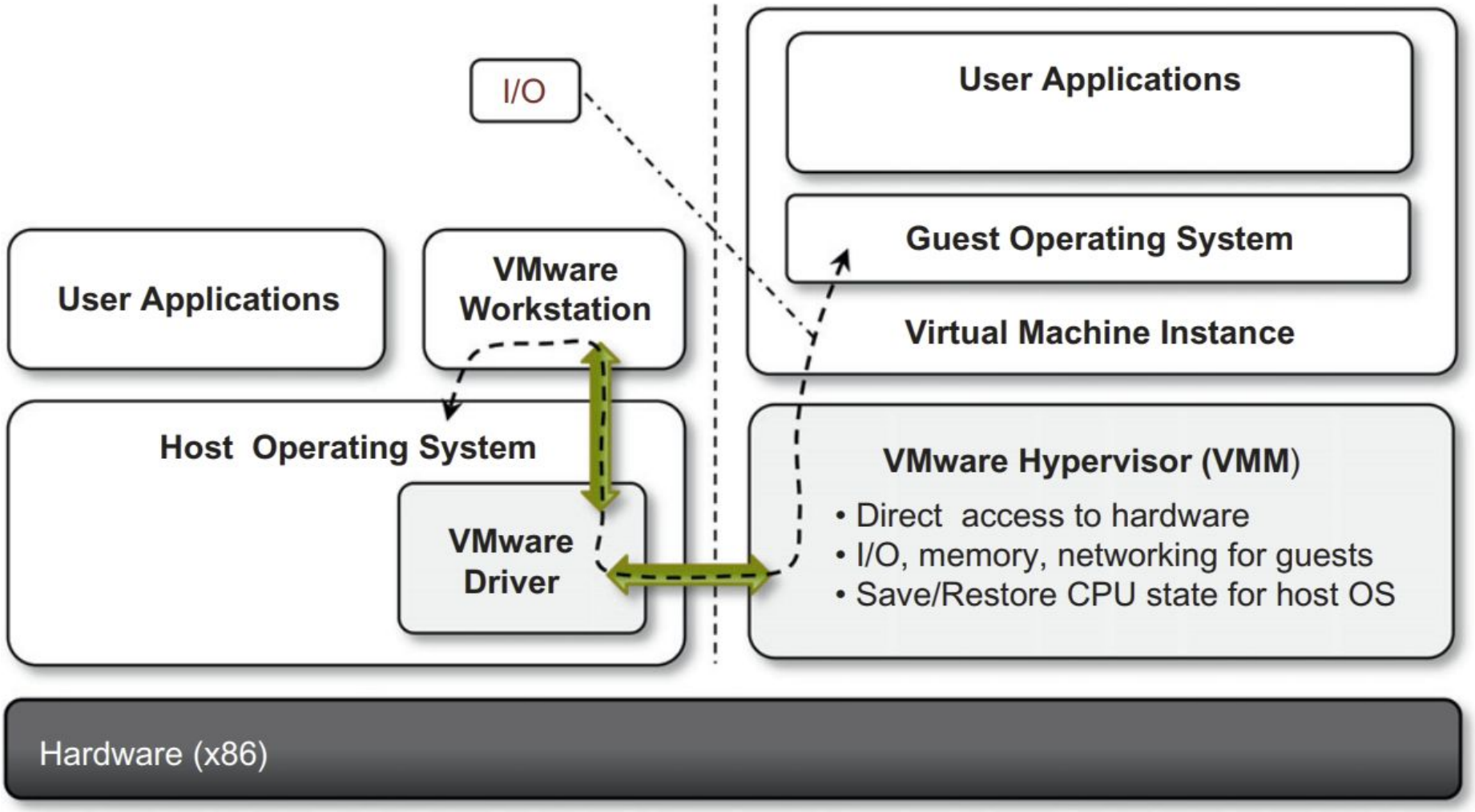


# VMWare – Full Virtualization



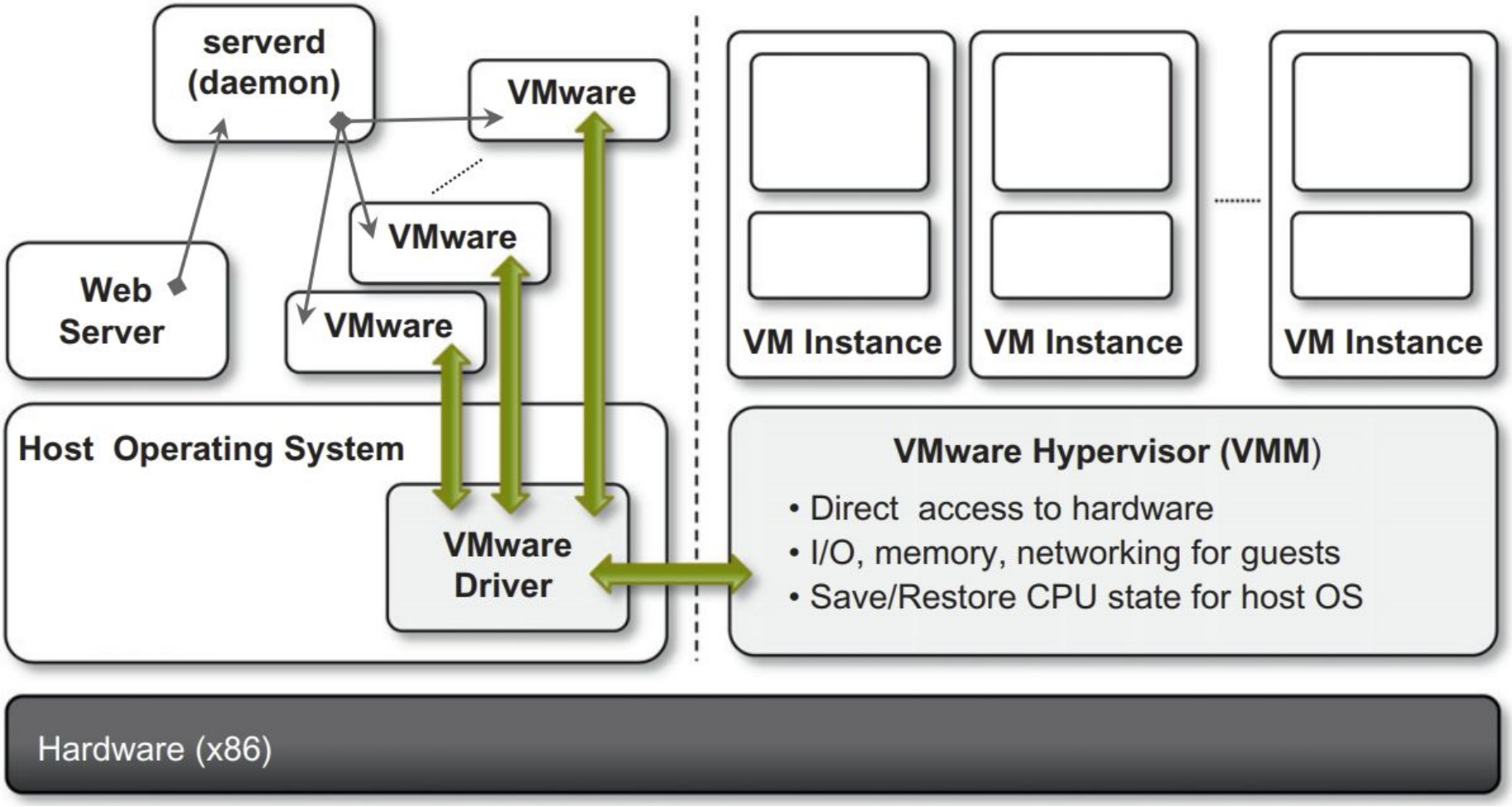
# VMware workstation architecture

VMware supports virtualization of operating system environments and single applications on end user computers.



# VMware GSX server architecture

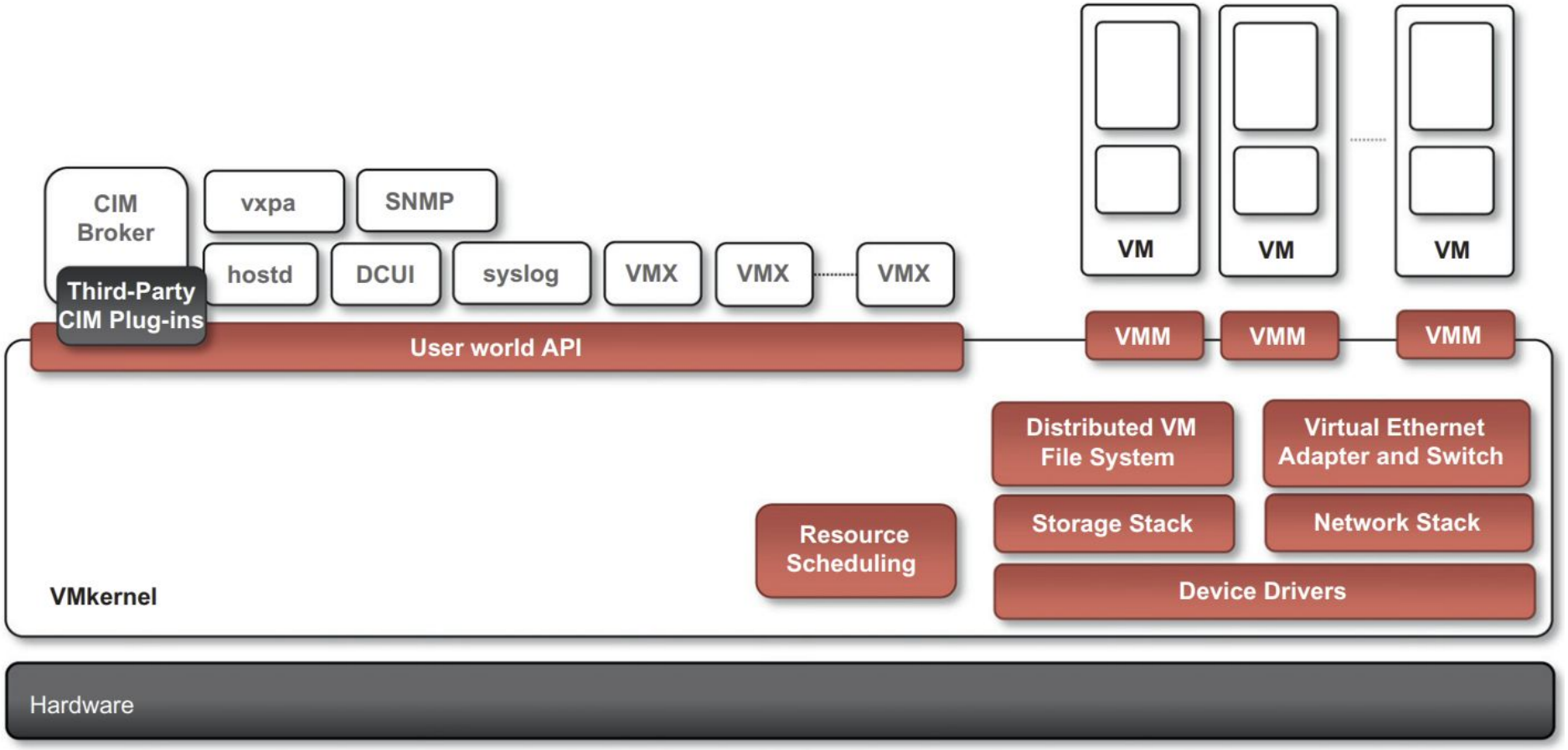
Provides Server Virtualization



# Contd…

- A daemon process, called serverd, controls and manages VMware application processes.
- These applications are then connected to the virtual machine instances by means of the VMware driver installed on the host operating system.
- Virtual machine instances are managed by the VMM.
- User requests for virtual machine management and provisioning are routed from the Web server through the VMM by means of serverd.

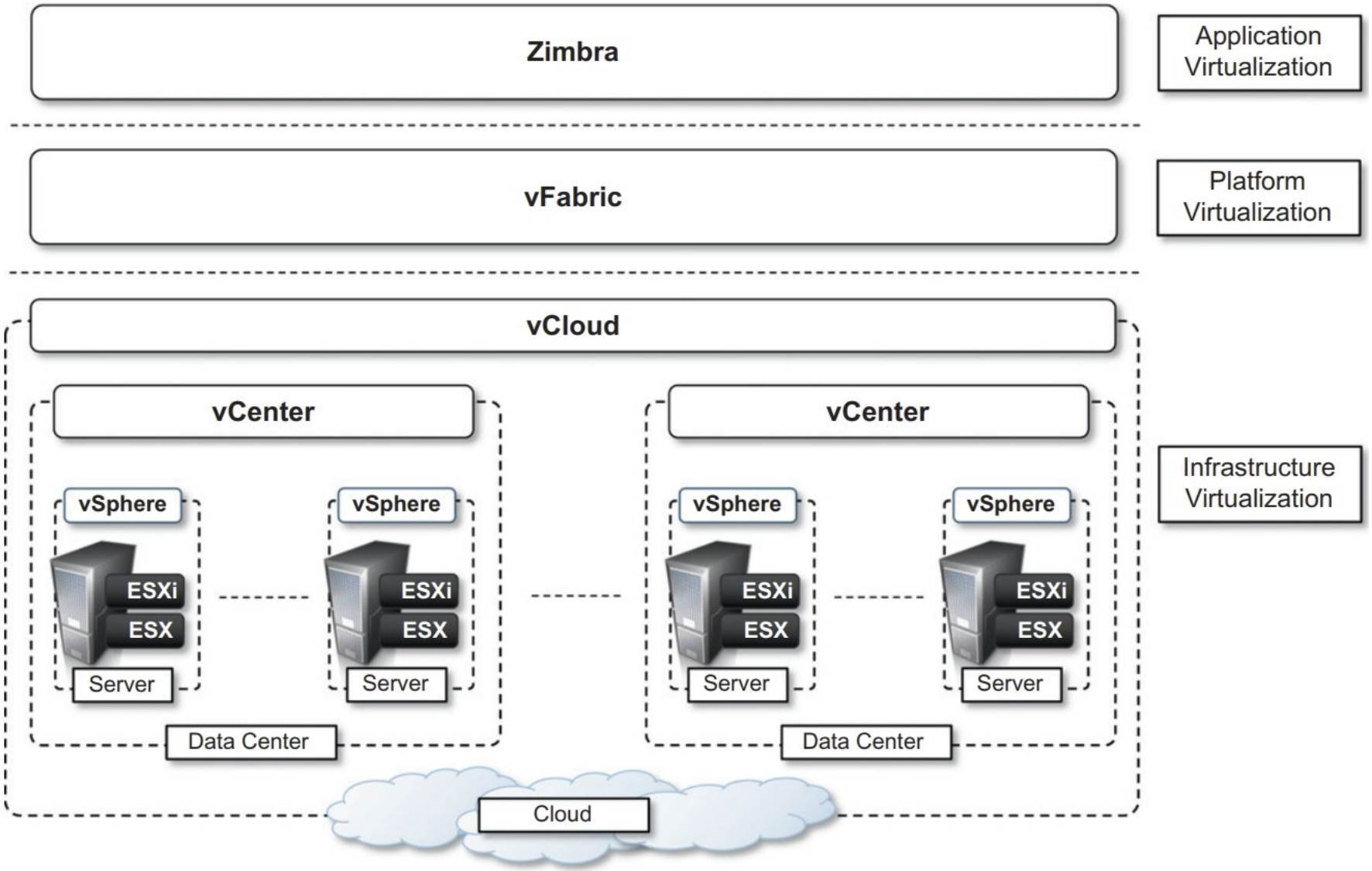
# VMware ESXi server architecture



# Contd…

- The base of the infrastructure is the VMkernel, which is a thin Portable Operating System Interface (POSIX) compliant operating system that provides the minimal functionality for processes and thread management, file system, I/O stacks, and resource scheduling.
- The kernel is accessible through specific APIs called User world API.
- These APIs are utilized by all the agents that provide supporting activities for the management of virtual machines.
- Remote management of an ESXi server is provided by the CIM Broker, a system agent that acts as a gateway to the VMkernel for clients by using the Common Information Model (CIM) protocol.
- The ESXi installation can also be managed locally by a Direct Client User Interface (DCUI), which provides a BIOS-like interface for the management of local users.

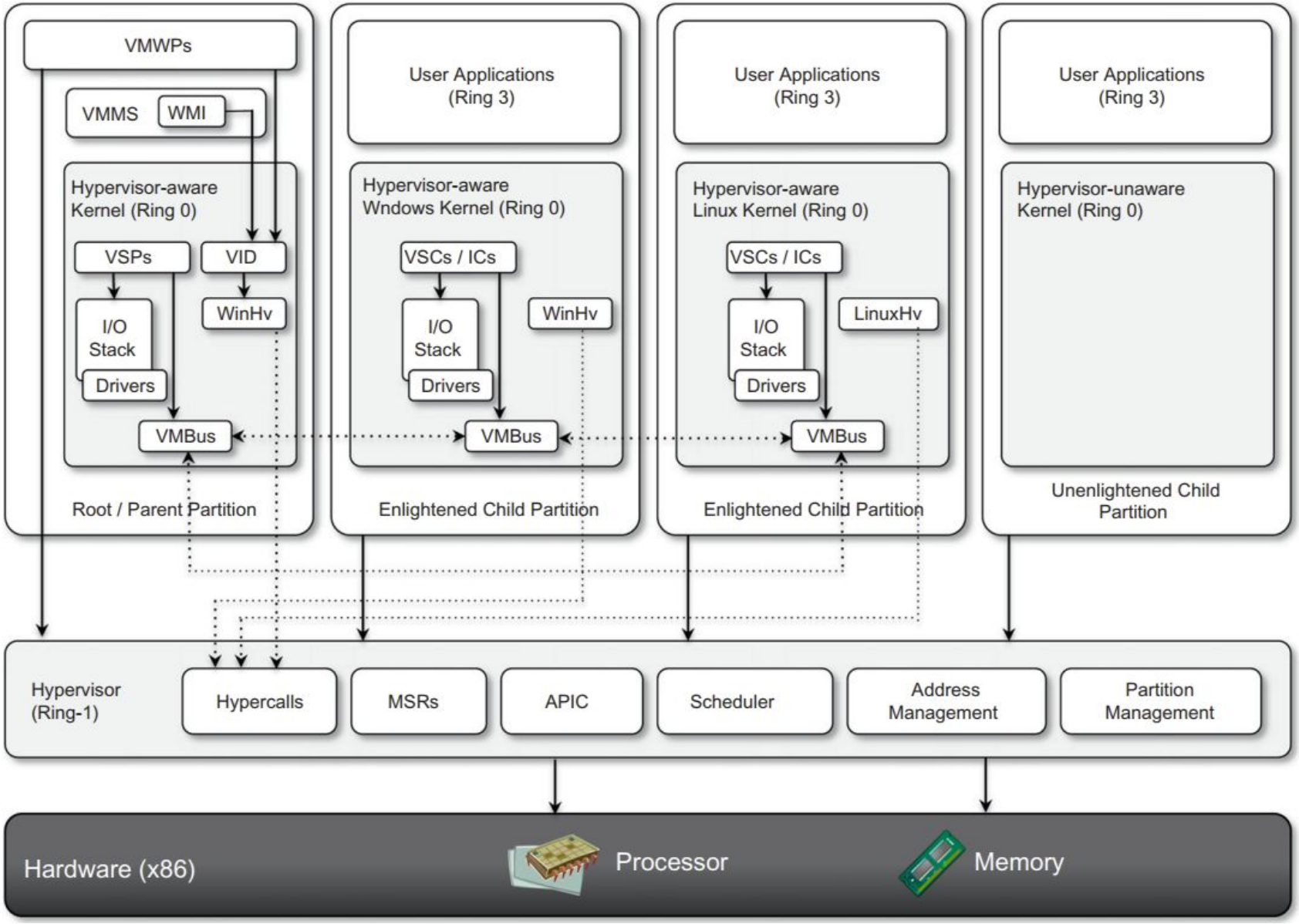
# VMware Cloud Solution stack



# Contd…

- vSphere provided a set of basic services.
- The management of the infrastructure is operated by VMware vCenter, which provides centralized administration and management of vSphere installations in a data center environment.
- A collection of virtualized data centers are turned into a IaaS cloud by VMware vCloud.
- VMware also provides a solution for application development in the cloud with VMware vFabric.
- VMware provides Zimbra, a solution for office automation, messaging, and collaboration that is completely hosted in the cloud and accessible from anywhere.

# Microsoft Hyper-V architecture



Thank You !!!