ADC501

Cloud Computing

Module 2

Virtualization

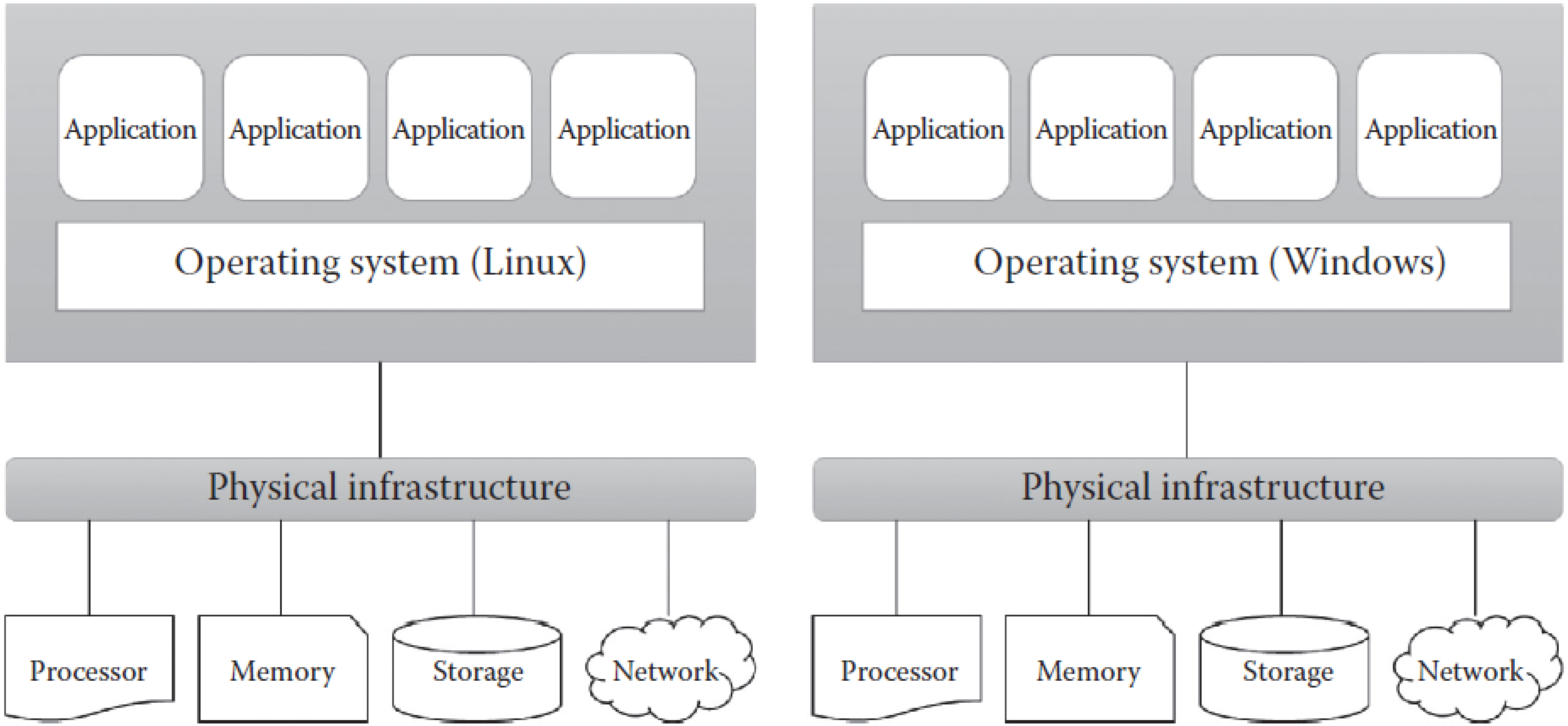


FIGURE 7.1
Before virtualization.

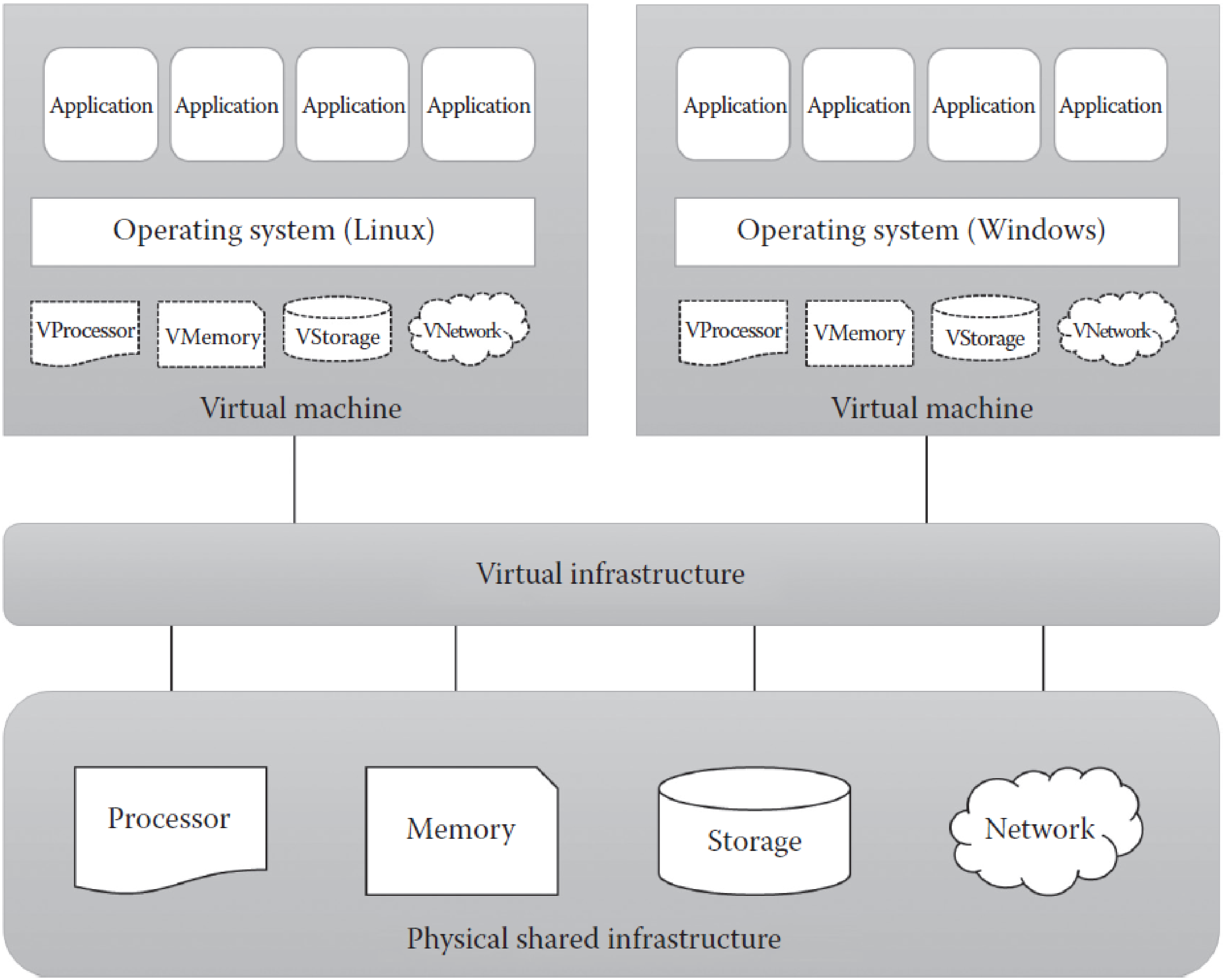


FIGURE 7.2
After virtualization.

Virtualization

- Virtualization - underlying core technology of cloud computing
- provides efficient resource utilization and increased return on investment (ROI)
- low capital expenditures (CapEx) and operational expenditures (OpEx)
- promotes the green IT by reducing energy wastage
- Eases administration
- Improves disaster recovery
- Testing beta software & maintaining legacy applications

Drawbacks

- Single point of failure
- Demands high-end and powerful infrastructure
- May lead to lower performance
- Requires specialized skill set

Virtualization Opportunities

Processor Virtualization

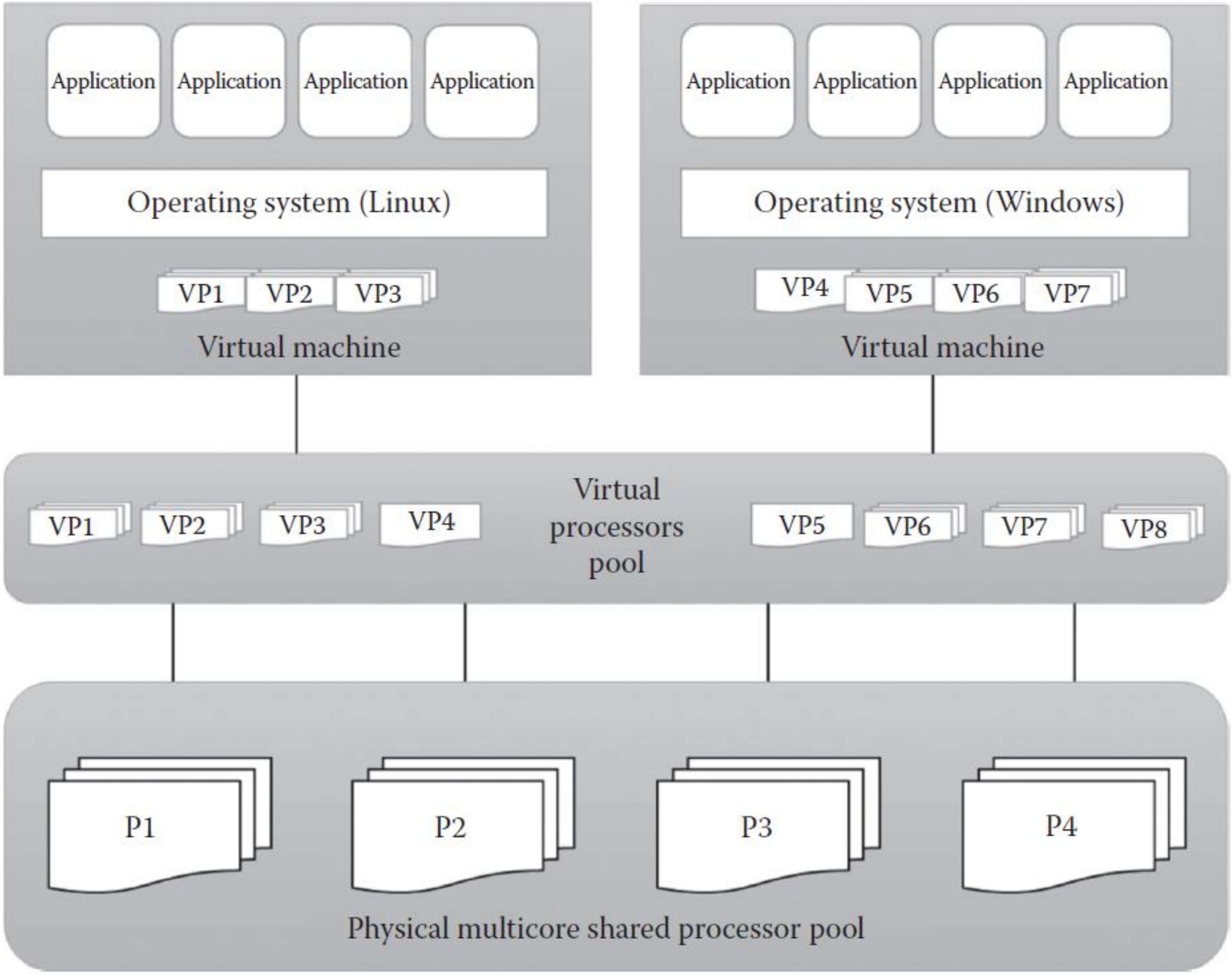


FIGURE 7.3
Processor virtualization.

Memory Virtualization

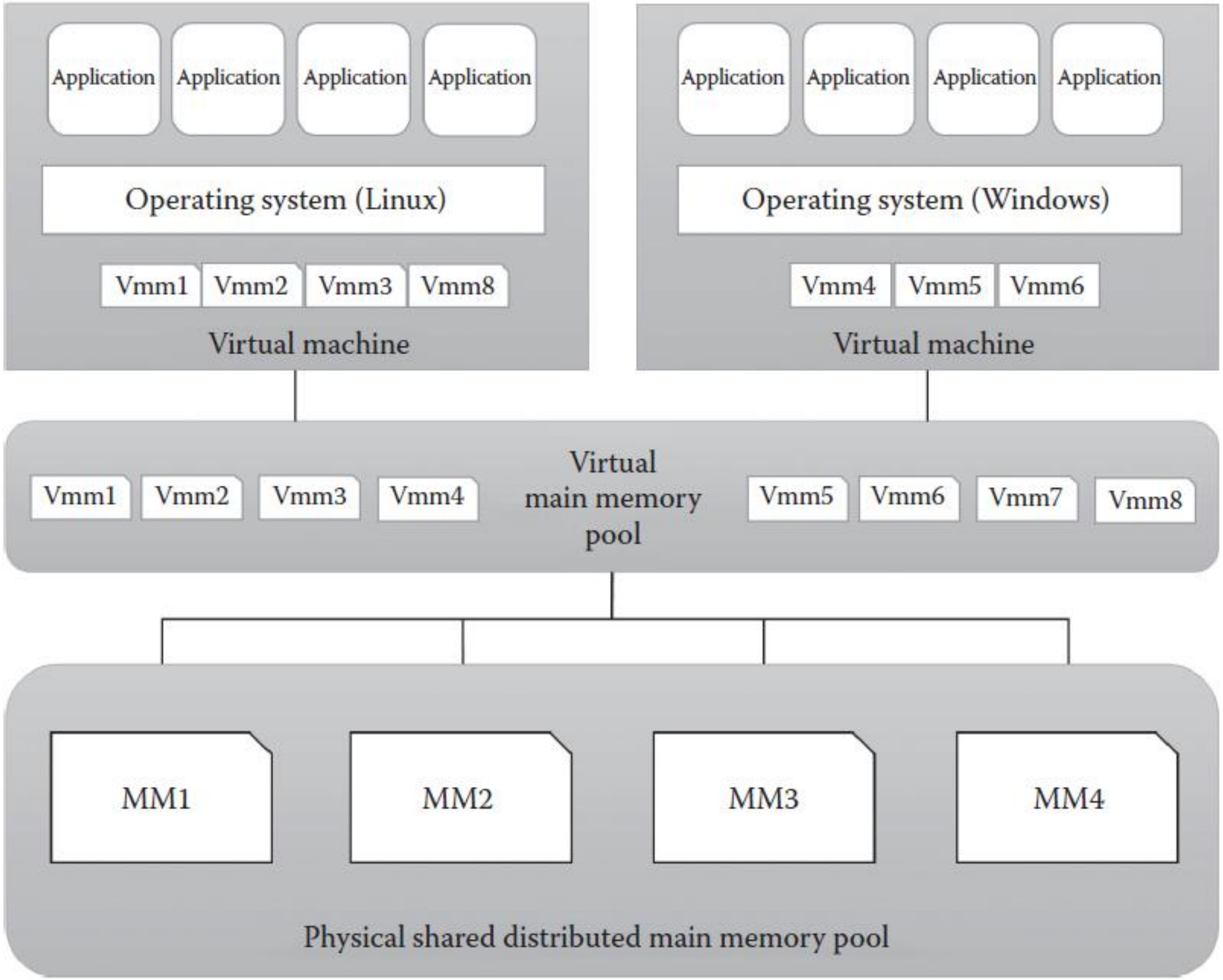


FIGURE 7.4
Main memory virtualization.

Storage Virtualization

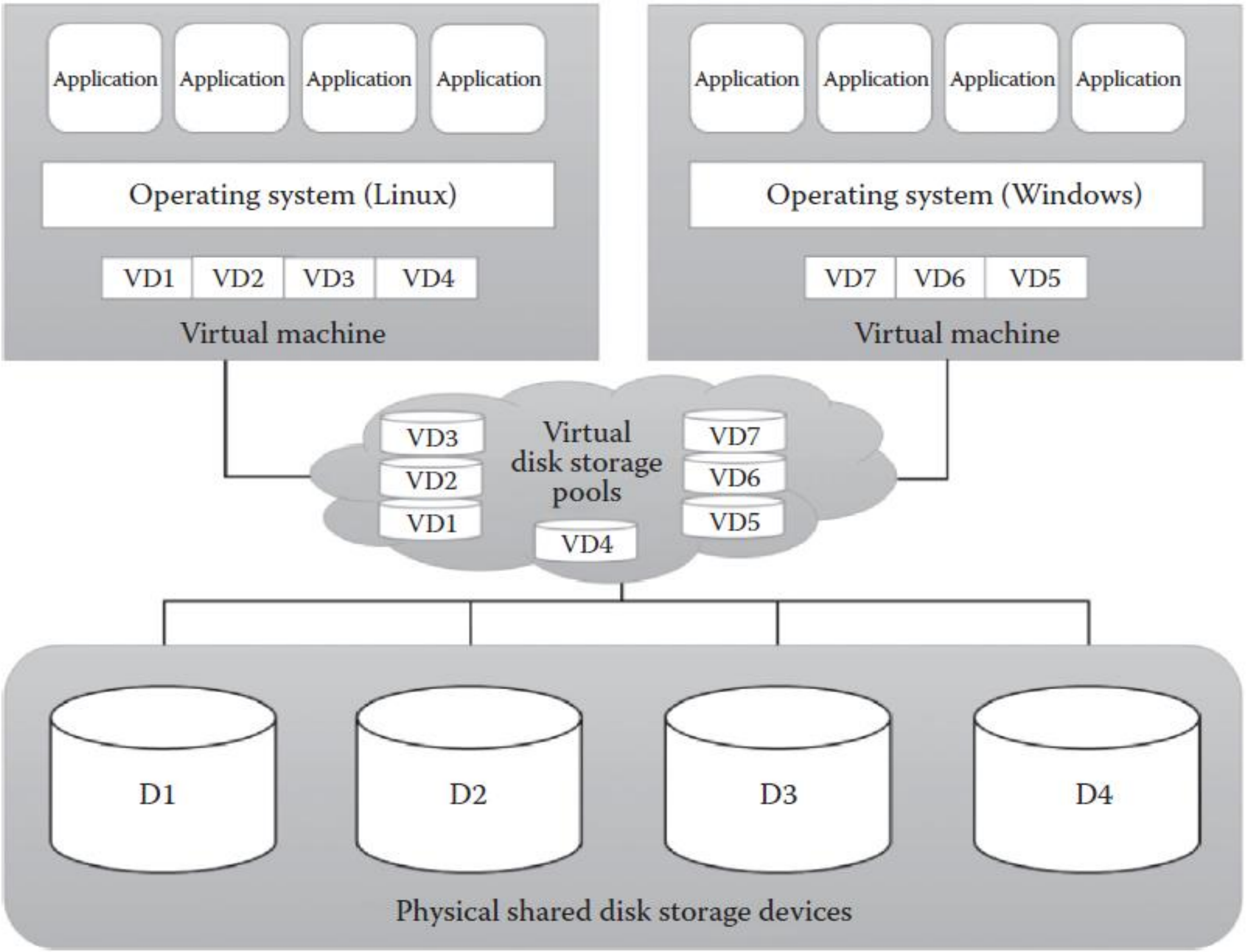


FIGURE 7.5
Storage virtualization.

Network Virtualization

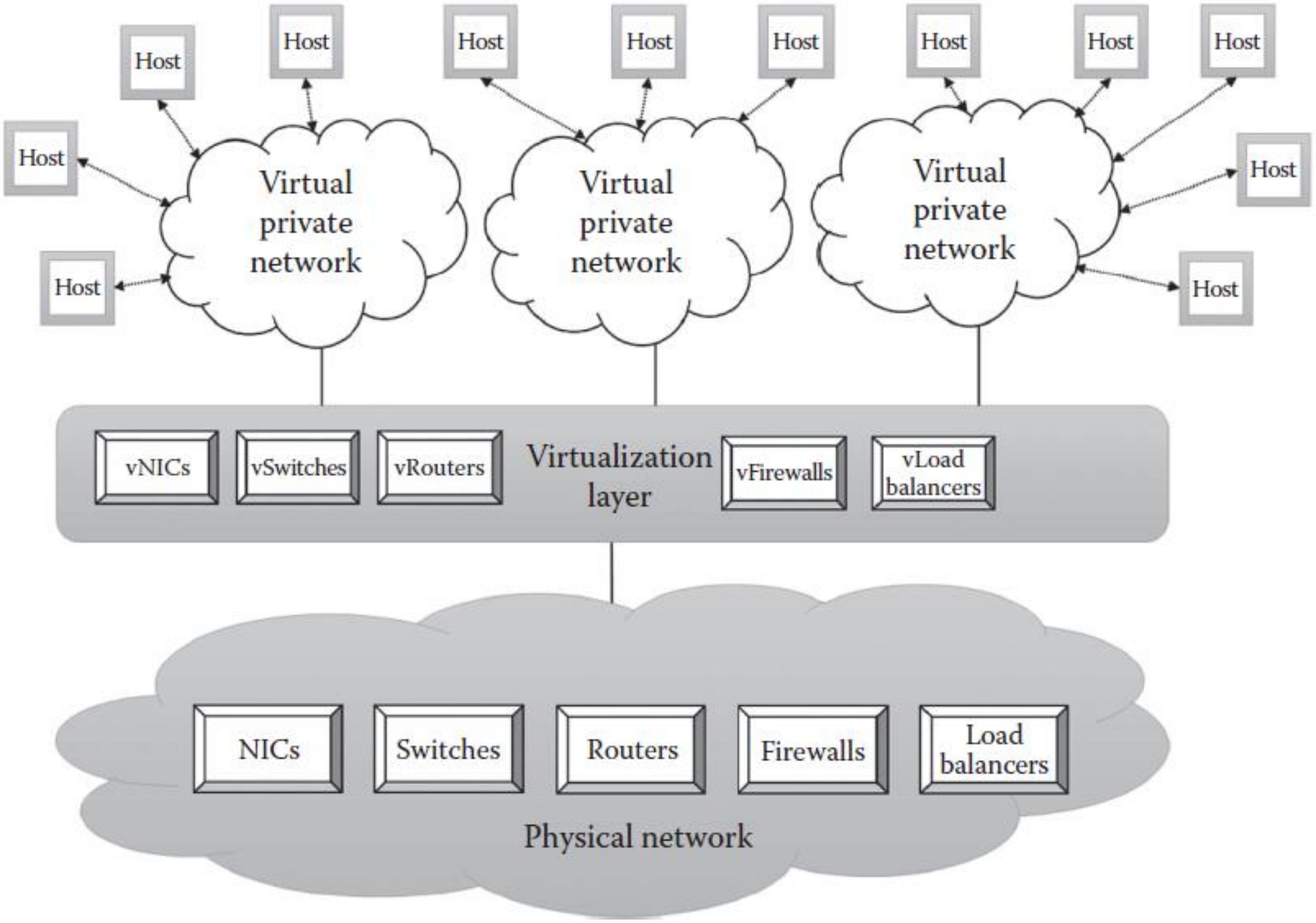


FIGURE 7.6
Network virtualization.

Data Virtualization

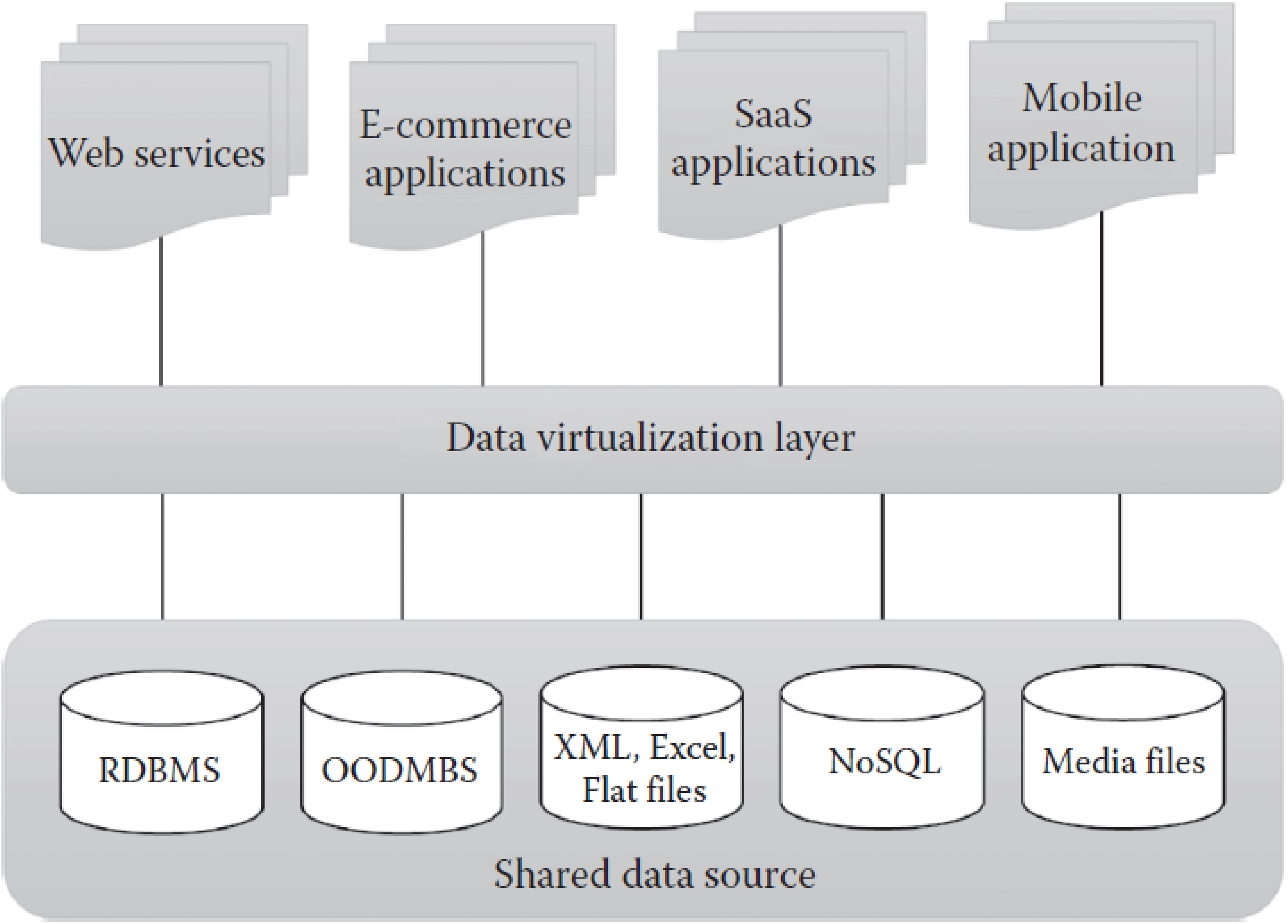


FIGURE 7.7
Data virtualization.

Application Virtualization

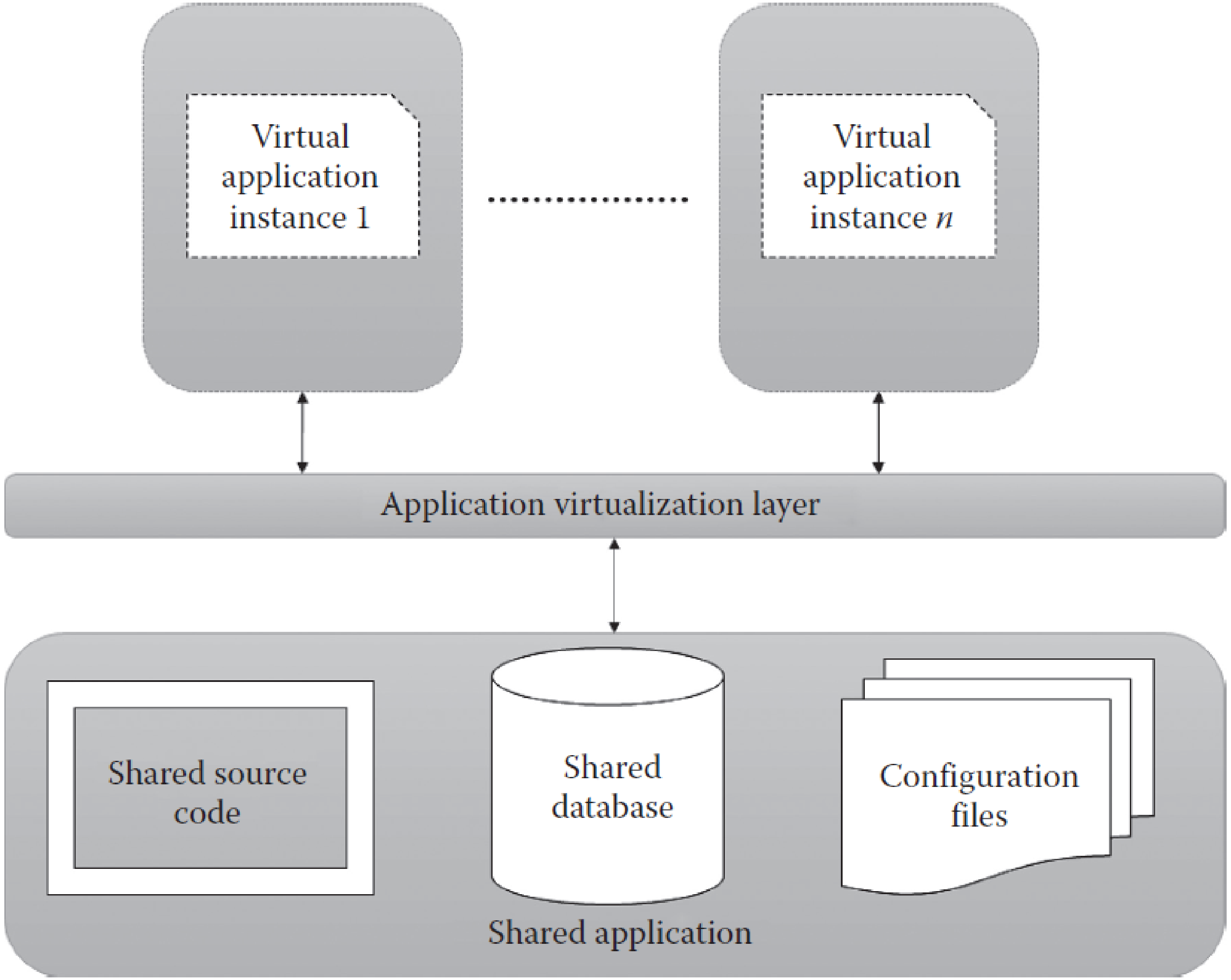


FIGURE 7.8
Application virtualization.

Approaches to Virtualization

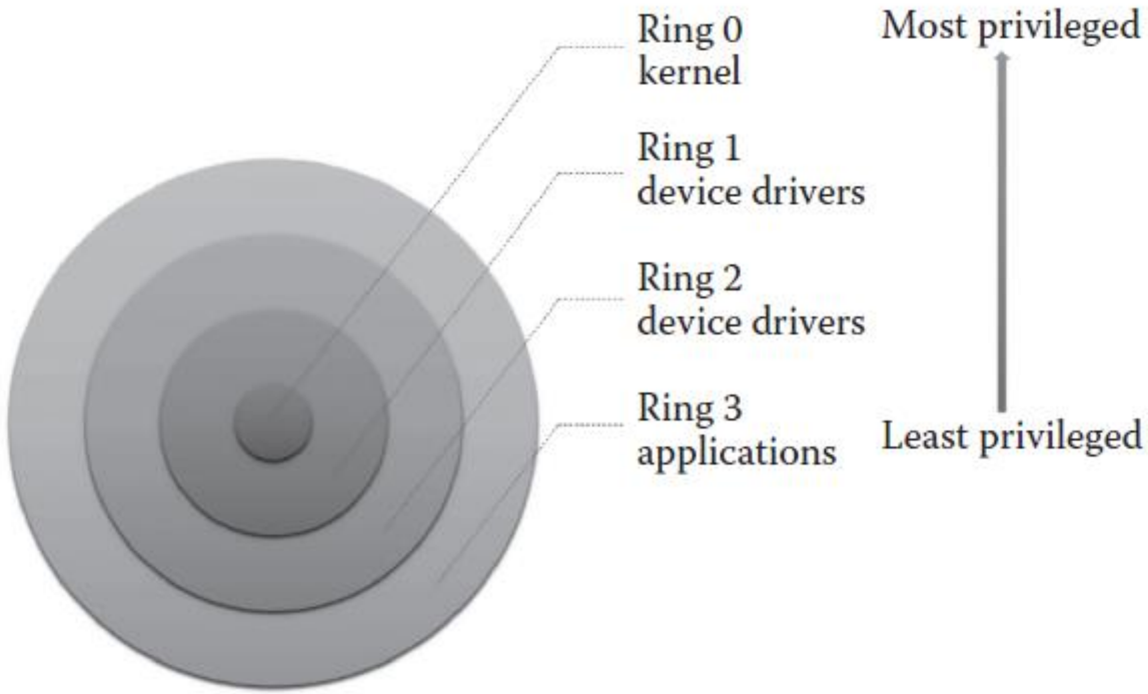
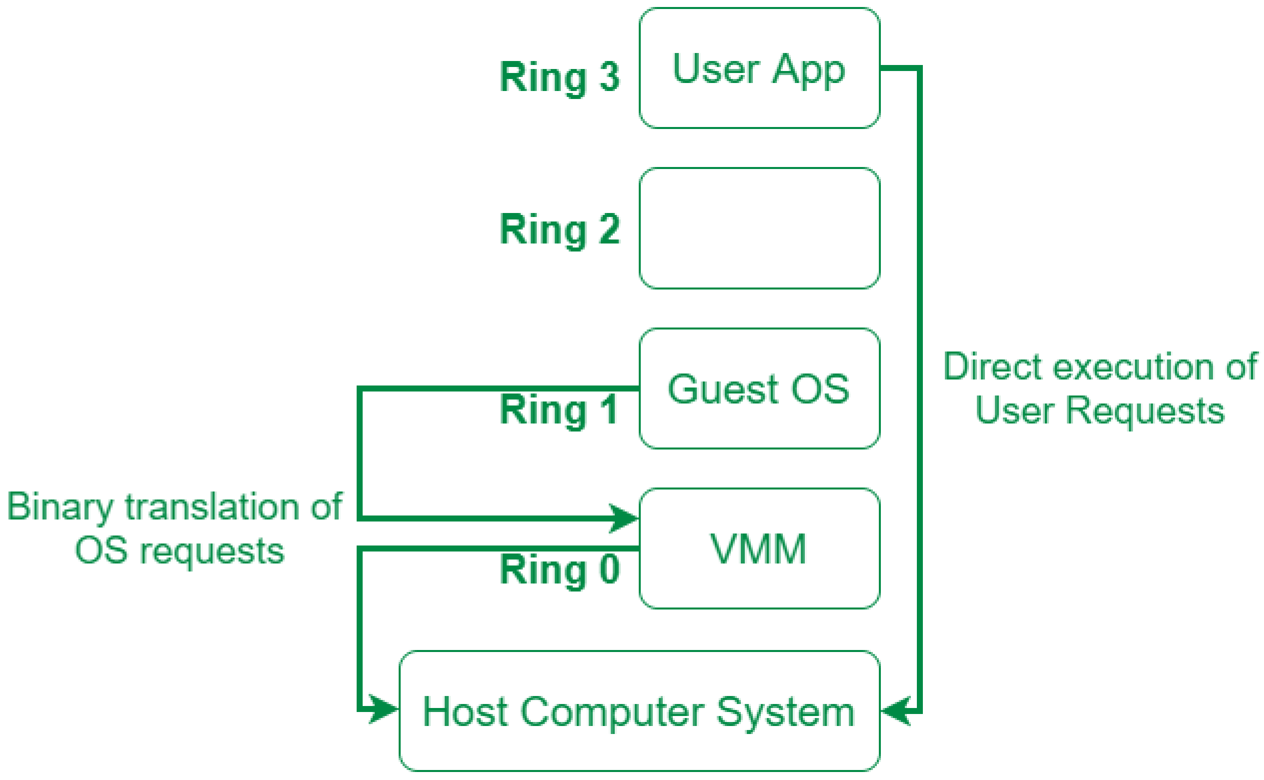


FIGURE 7.9
Protection rings in OSs.

Approaches Virtualization

Full virtualization

- uses a special kind of software called a hypervisor
- isolation among the various VMs
- concurrent execution of multiple OSs
- no change required in the guest OS
- binary translation
- The virtualization technique in which the machine language code of an application for the guest operating system is translated to the machine language code of the host operating system



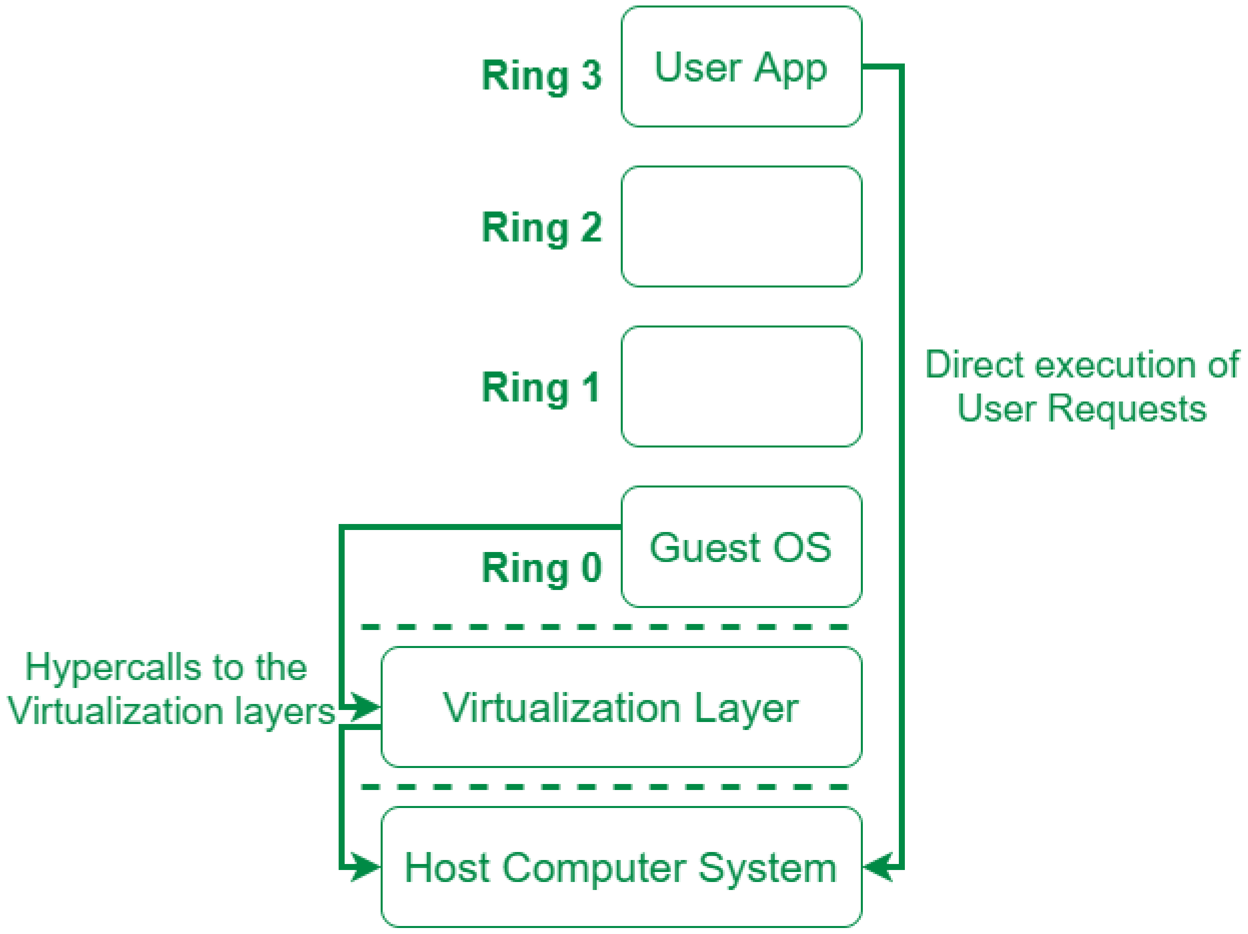
Full Virtualization

Paravirtualization

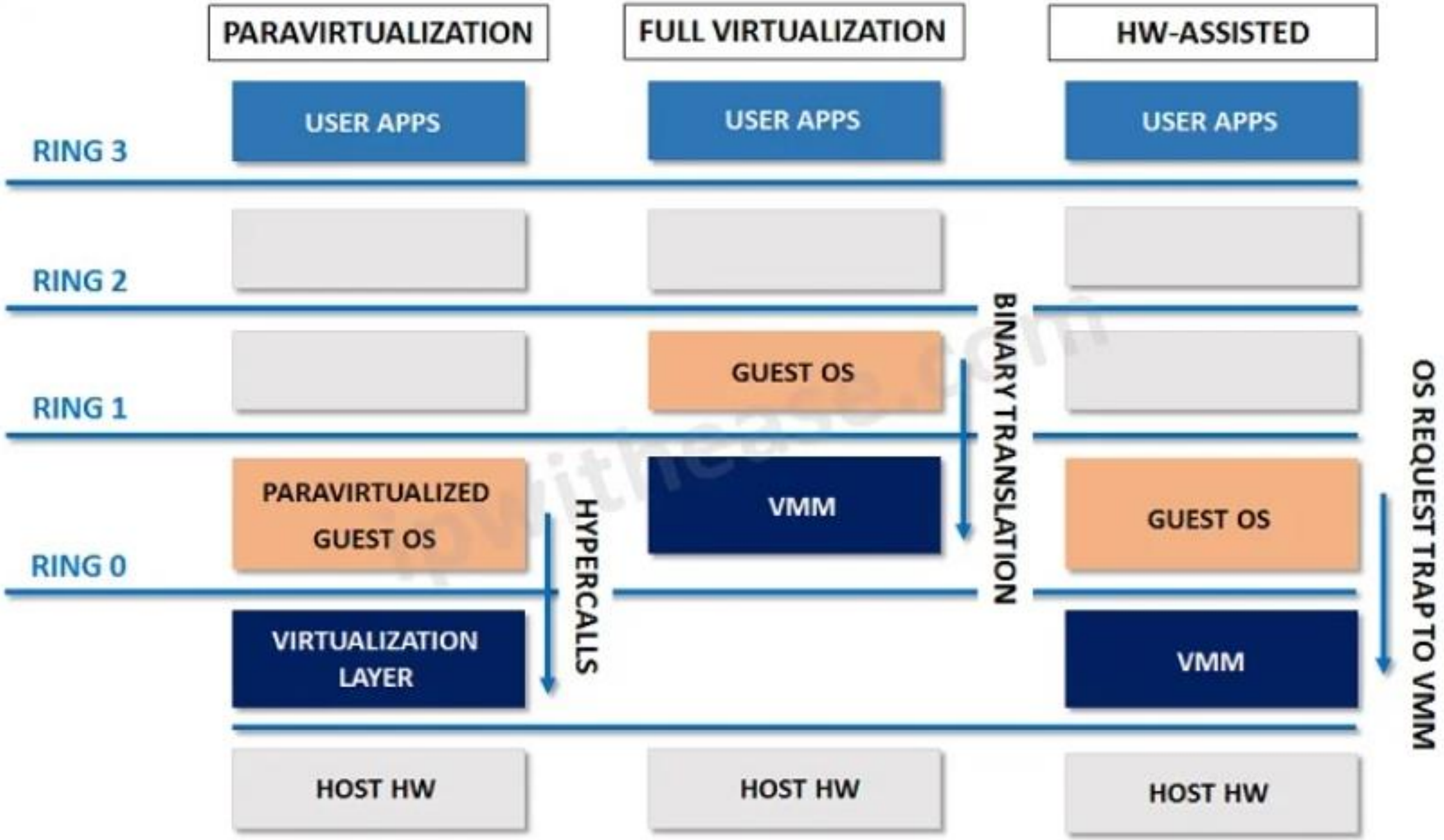
- partial virtualization or OS-assisted virtualization
- modification of the guest OS is required
- improves the overall system performance by eliminating the overhead of binary translation

Hardware-Assisted Virtualization

- Intel (VT-x) and AMD (AMD-v) have developed processors supporting the virtualization through the hardware extension
- eliminates the overhead of binary translation and paravirtualization
- lack of support from all vendors



Paravirtualization


PARAVIRTUALIZATION
FULL VIRTUALIZATION
HW-ASSISTED
USER APPS
USER APPS
USER APPS
RING 3
RING 2
RING 1
RING 0
GUEST OS
PARAVIRTUALIZED GUEST OS
VMM
GUEST OS
VIRTUALIZATION LAYER
VMM
HYPERCALLS
BINARY TRANSLATION
OS REQUEST TRAP TO VMM
HOST HW
HOST HW
HOST HW

Full Virtualization	*Para-Virtualization or OS-Assisted Virtualization*	*Hardware-Assisted Virtualization*
Guest OS has no role in virtualization.	Guest OS plays role in virtualization.	Guest OS has no role in virtualization.
Guest OS remains unaware about the virtualization.	Guest OS has to be aware about the virtualization.	Guest OS remains unaware about the virtualization.
Normal version of available OS can be used as guest OS.	Modified version of available OS is required.	Normal version of available OS can be used as guest OS.
It provides good options for guest OS.	It provides lesser options for guest OS.	It provides good options for guest OS.
Guest OS is not hypervisor-specific.	Guest OS is tailored to be hypervisor-specific.	Guest OS is not hypervisor-specific.
Here it requires no special feature in the host CPU.	Here it requires no special feature in the host CPU.	Here it requires explicit features in the host CPU.
Hardware does not play role in virtualization.	Hardware does not play role in virtualization.	Hardware plays role in virtualization.
Hypervisor takes care of all of the virtualization tasks.	Guest OS along with hypervisor take care of the virtualization tasks.	Specialized hardware device along with hypervisor take care of virtualization tasks.
Virtualization overhead of hypervisor is more.	Virtualization overhead of hypervisor is less.	Virtualization overhead of hypervisor is less.
Virtualization performance is little slow.	Virtualization performance is better.	Virtualization performance is better.
It provides high level of security as all of the virtualization controls remain with the hypervisor.	Here the security is compromised as guest OS has some control in virtualization.	Here the security is compromised as calls from guest OS can directly access the hardware.

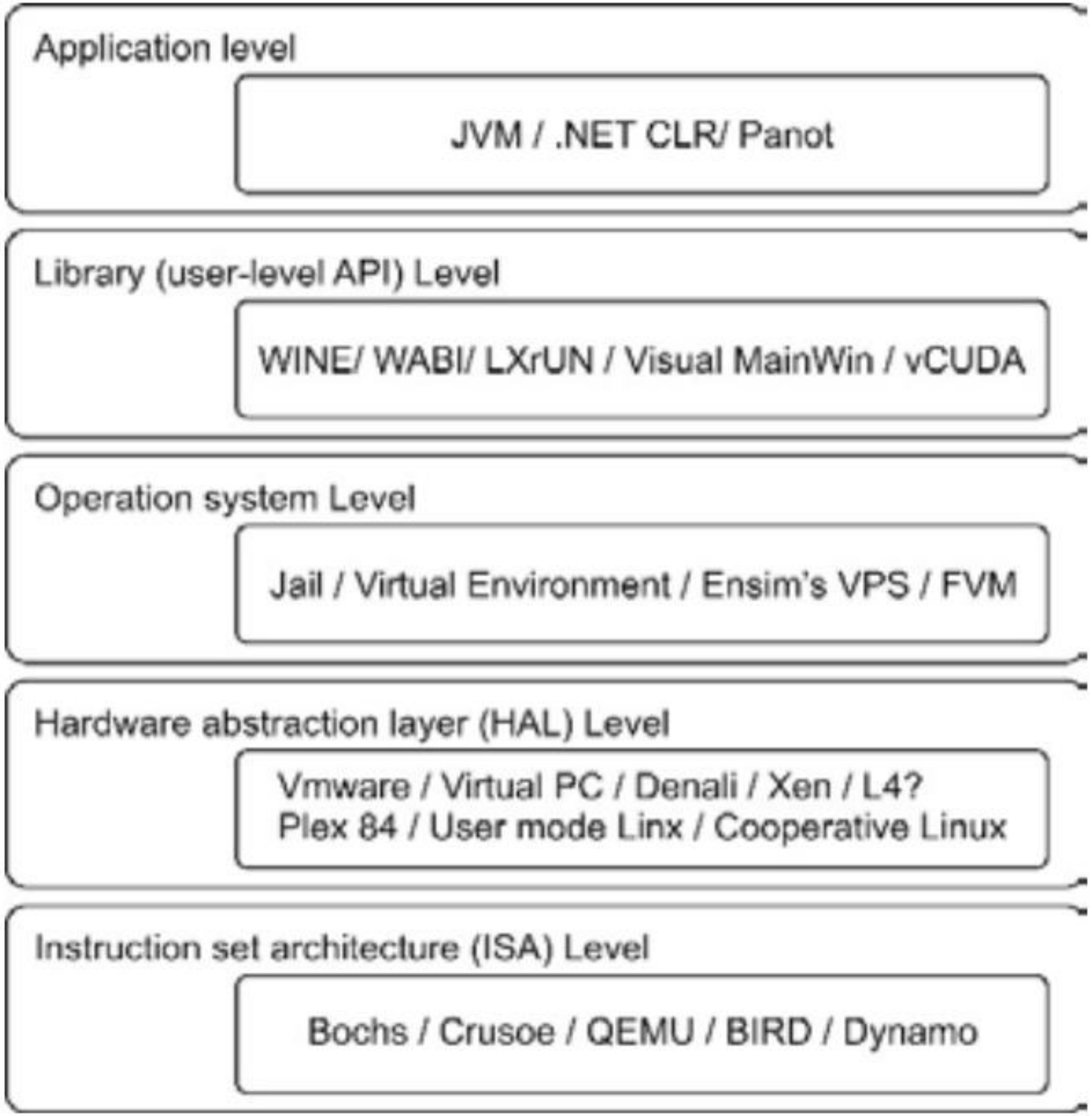


Figure 2: Implementation Levels of Virtualization

Virtualization at the Instruction Set Architecture (ISA) Level

- The host machine is a physical platform comprising various components, including memory, process, Input/Output (I/O) devices, buses
- VMM installs guest systems
- These guest systems issue instructions for the emulator to process and execute. The instructions are received by the emulator
- Emulator transforms them into a native instruction set
- native instructions are run on the host machine’s hardware.
- **Positives**
  makes it easy to implement multiple systems on a single physical structure.
  host system to adjust to a change in the architecture of the guest system,
- **Negatives**
  instructions need to be interpreted before being executed
  poor performance

Virtualization at the Hardware Abstraction Layer

- exploits the similarity in architectures of the guest and host platforms to cut down the interpretation latency

- most common technique used in computers on x86 platform,

- cannot fully virtualize all the platforms through this technique.

- requires a lot of time to be spent in the installation and administration of the virtual system before you can think of testing or running applications.

- physical and virtual OSs are the same, this kind of virtualization results in duplication of your efforts

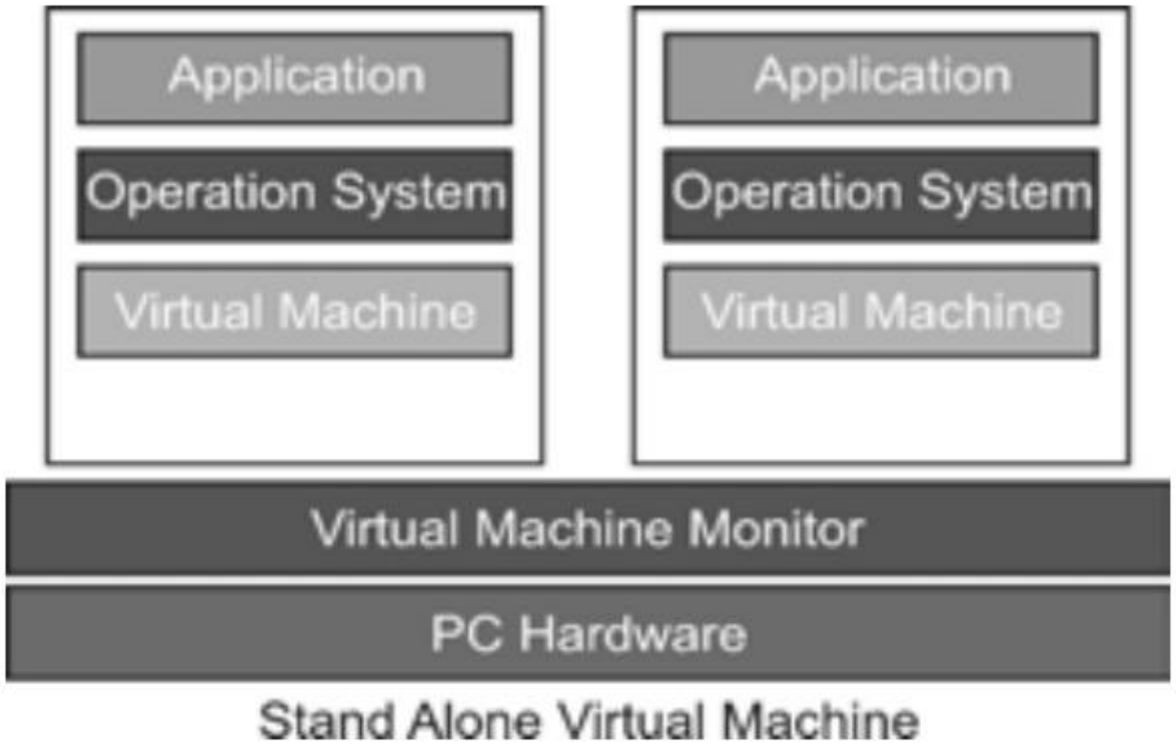


Figure 3: Virtualization at HAL

Virtualization at the OS Level

- This technique includes sharing of both the hardware and the OS.

Virtualization at the Application Level

- The arrival of the Java Virtual Machine (JVM) brought a new dimension to virtualization,

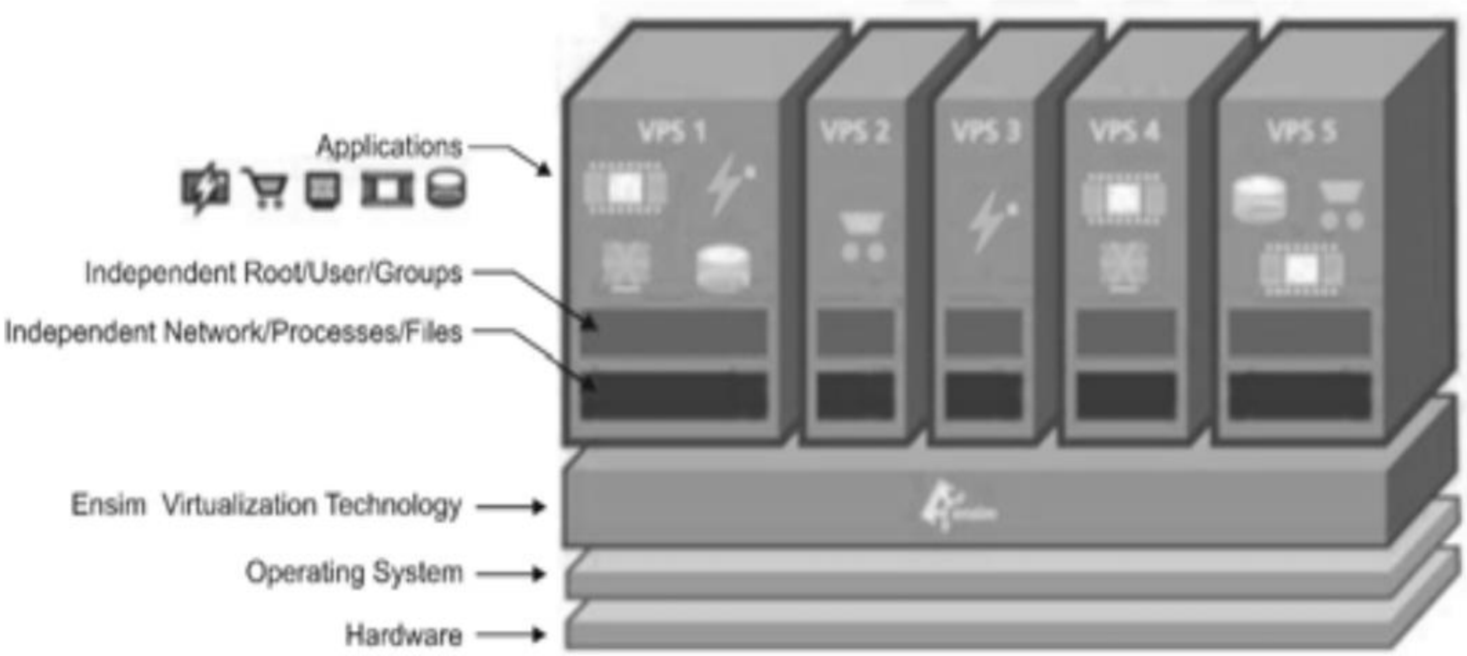


Figure 4: Application-level Virtualization

Virtualization at the Programming Language Level or Library Level

- An application programming interface (API) is **a mediator that allows applications to communicate with each other**.
- to be exported by implementing various libraries at the user-level.
- and enable programmers to write programs easily.

in Table 1.

Table 1: Relative Merits and Demerits of Different Levels of Virtualization Implementation

Implementation Level	Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	Very Poor Performance	Excellent	Medium	Medium
HAL	Excellent Performance	Medium	High	Very Good

Virtualization Design Requirements

- **Equivalence Requirement**
- **Efficiency Requirement**
- **Resource Control Requirement**

Virtualization Providers

- **Microsoft**

  Virtual PC - guest systems may have Linux or any OS other than Windows, but the host must be a Windows computer., does not provide integration facilities for the 64-bit guest systems,

  Virtual Server 2005 – enterprises, available for free download on the Microsoft website, can run only on a Web-based console,

  Hyper-V

- **VMWare**

  VMware Workstation – released in 1999, robustness, timeliness, all-platform support, and support for guest machines of any kind, make it the most utilized virtualization platform across the world. **Not free**

  VMware Server – free, requires a host OS, which is either Windows or Linux,

Hypervisor

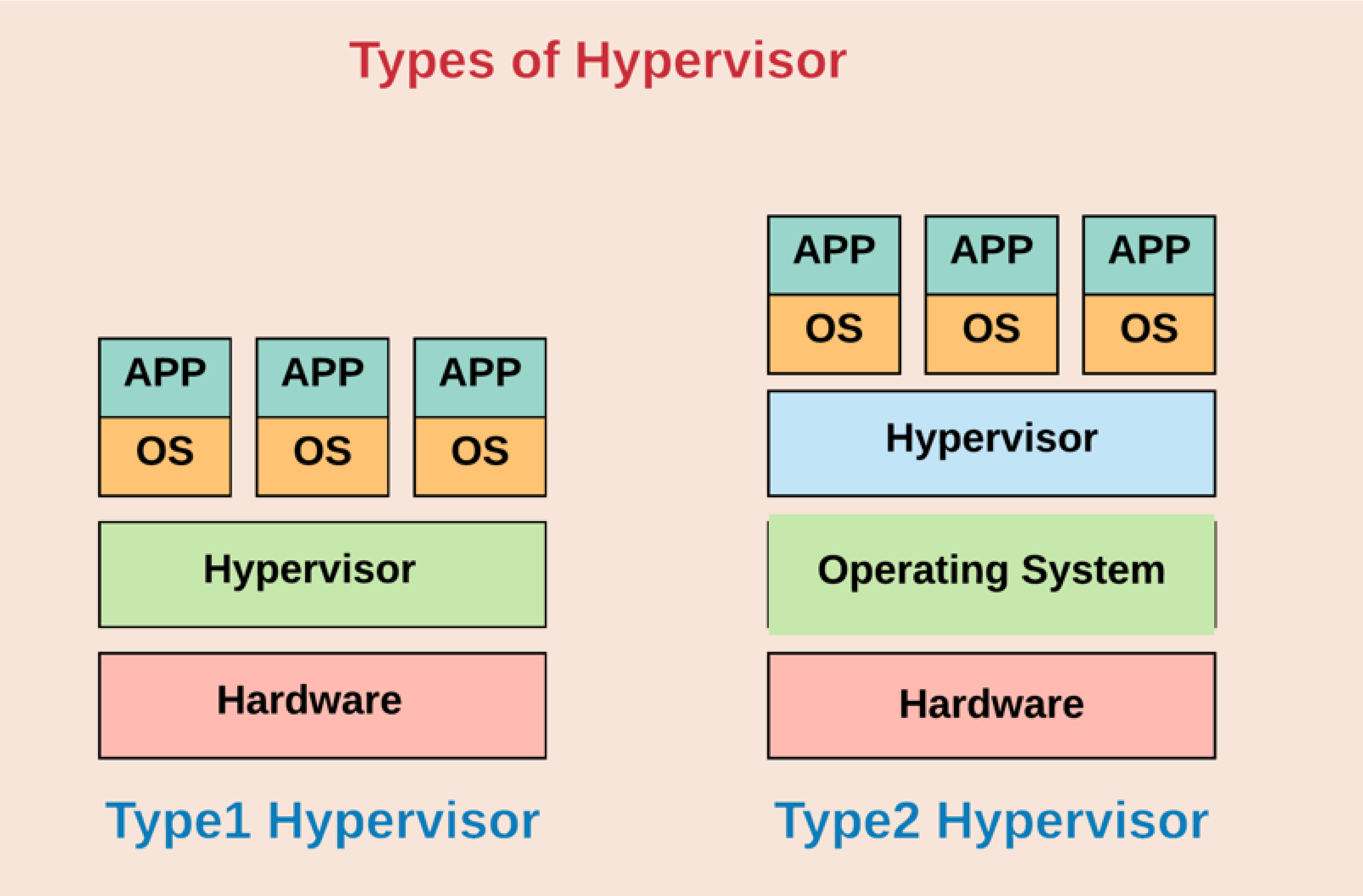
- VMware, Xen, Hyper-V, KVM
- help to run multiple OSs concurrently on a physical system sharing its hardware
- hypervisor is actually controlling the host processor and resources and in turn allocates what is needed to each OS
- hypervisor also makes sure that the guest OSs (called VMs) do not interrupt each other
- mainly two types

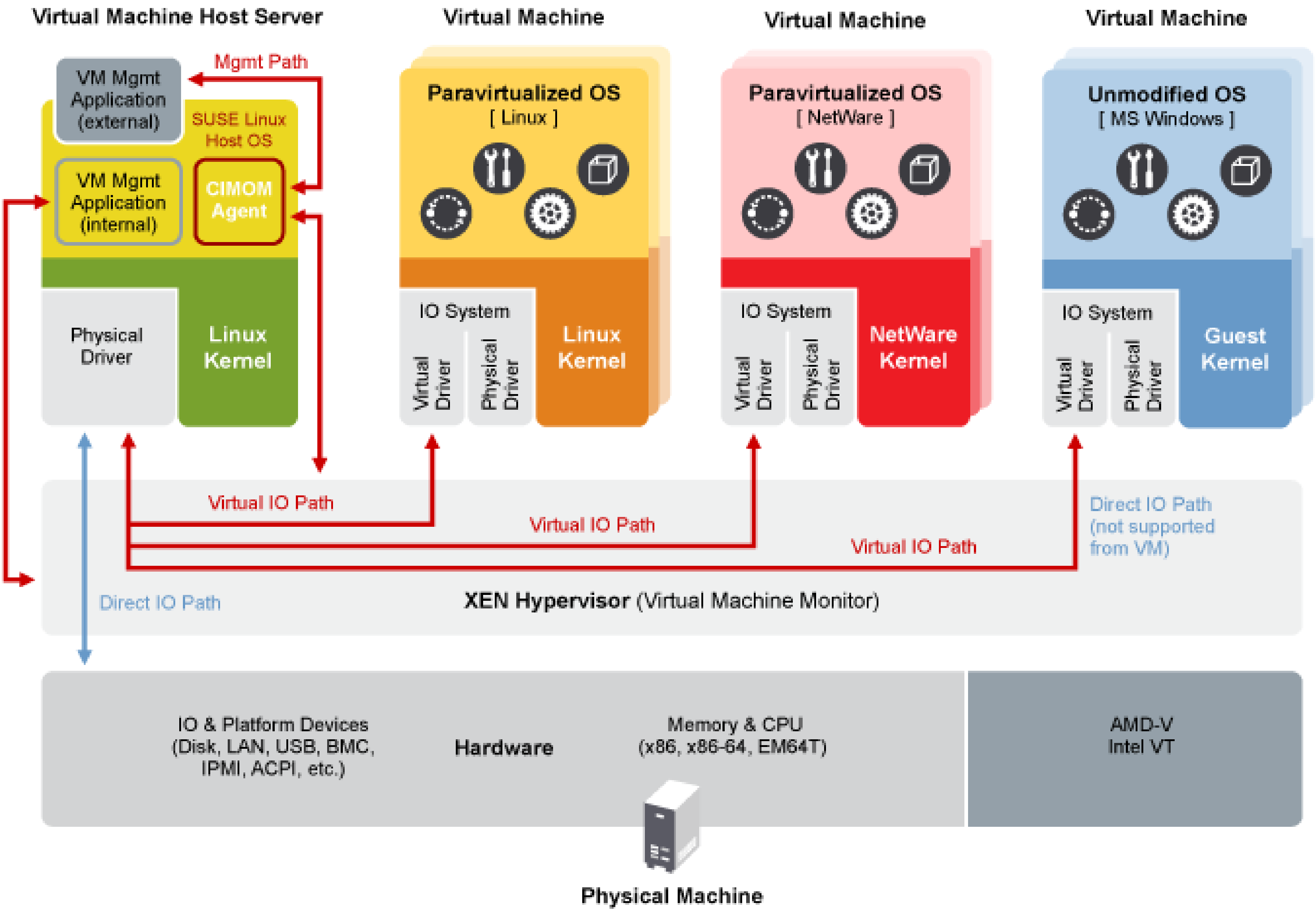
  Type 1 hypervisor

  runs directly on the host computer's hardware
  native or bare-metal hypervisors.
  VMware ESXi, Citrix Xen Server, and Microsoft Hyper-V hypervisor

  Type 2 hypervisor

  runs within a formal OS environment
  runs as a distinct second layer while the guest OS runs as a third layer above the hardware
  hosted hypervisors
  VMware Workstation and VirtualBox, KVM


Types of Hypervisor
APP
OS
APP
OS
APP
OS
Hypervisor
Hardware
Type1 Hypervisor
APP
OS
APP
OS
APP
OS
Hypervisor
Operating System
Hardware
Type2 Hypervisor



Virtual Machine Host Server
Virtual Machine
Virtual Machine
Virtual Machine
VM Mgmt Application (external)
Mgmt Path
SUSE Linux Host OS
VM Mgmt Application (internal)
CIMOM Agent
Physical Driver
Linux Kernel
Paravirtualized OS [Linux]
IO System
Virtual Driver
Physical Driver
Linux Kernel
Paravirtualized OS [NetWare]
IO System
Virtual Driver
Physical Driver
NetWare Kernel
Unmodified OS [MS Windows]
IO System
Virtual Driver
Physical Driver
Guest Kernel
Virtual IO Path
Virtual IO Path
Virtual IO Path
Direct IO Path (not supported from VM)
Direct IO Path
XEN Hypervisor (Virtual Machine Monitor)
IO & Platform Devices (Disk, LAN, USB, BMC, IPMI, ACPI, etc.)
Hardware
Memory & CPU (x86, x86-64, EM64T)
AMD-V Intel VT
Physical Machine