Virtualization in Cloud Computing

Dr. Amit Praseed

How do Cloud Services Work?

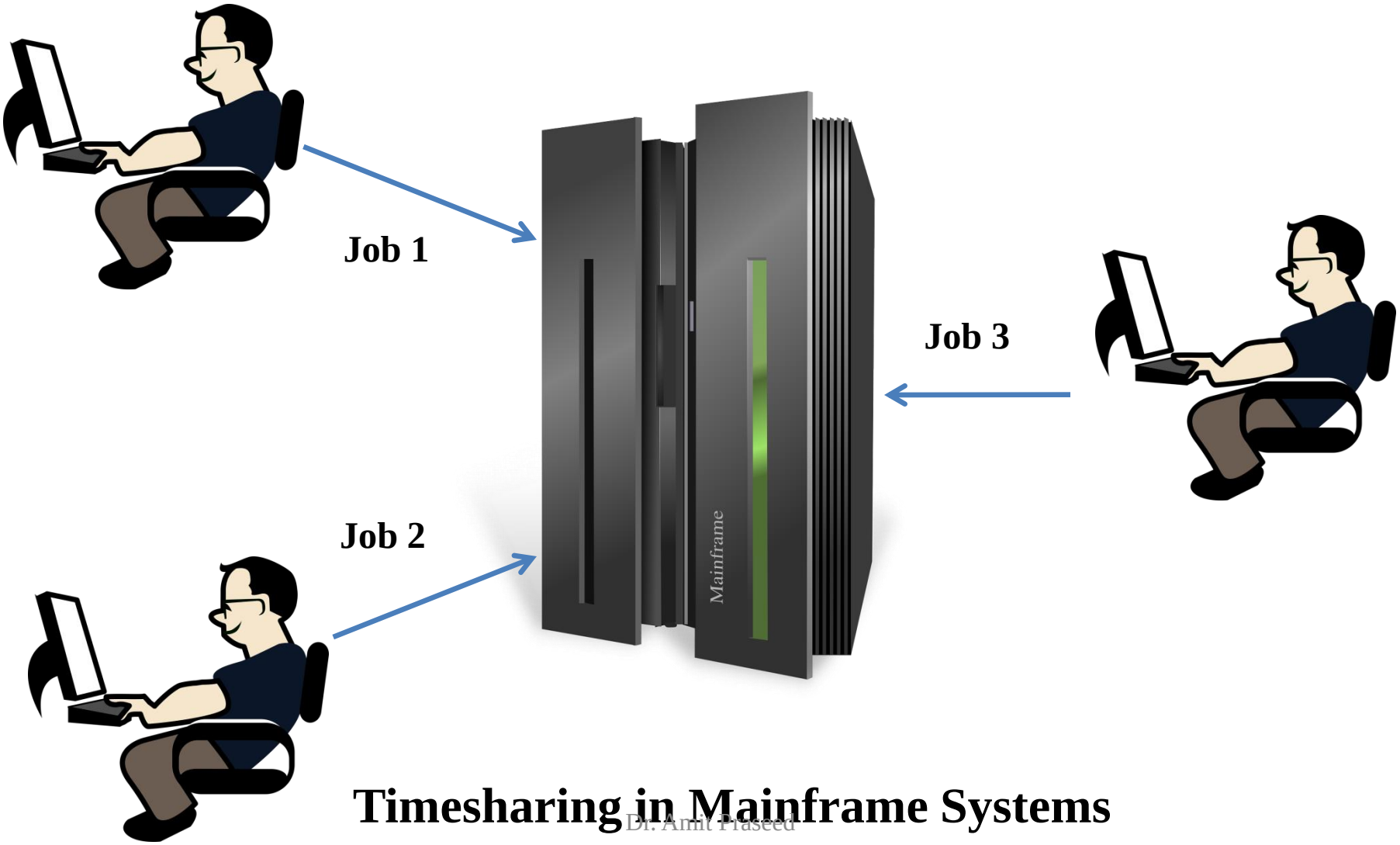
- Assume a cloud service provider has a datacentre with 4 CPUs and 8 GB RAM
  - Alice wants a system with 1 CPU and 2 GB RAM
  - Bob wants a system with 2 CPUs and 2 GB RAM
  - Carol wants a system with 1 CPU and 4 GB RAM
- In a traditional IT setup, this would be impossible!
- Solution: Create **virtual machines** with the required specifications and provide to the customers
- This uses a disruptive technology known as **virtualization**

- Put in simple terms, virtualization means

  creating an illusion of something which is not actually present

- Virtualization is used very commonly nowadays
  - **Virtual memory** gives us the illusion of a significantly larger memory than we physically have
  - **Virtual Reality** games allow users to perceive a world that doesn't physically exist

A Brief History of Virtualization

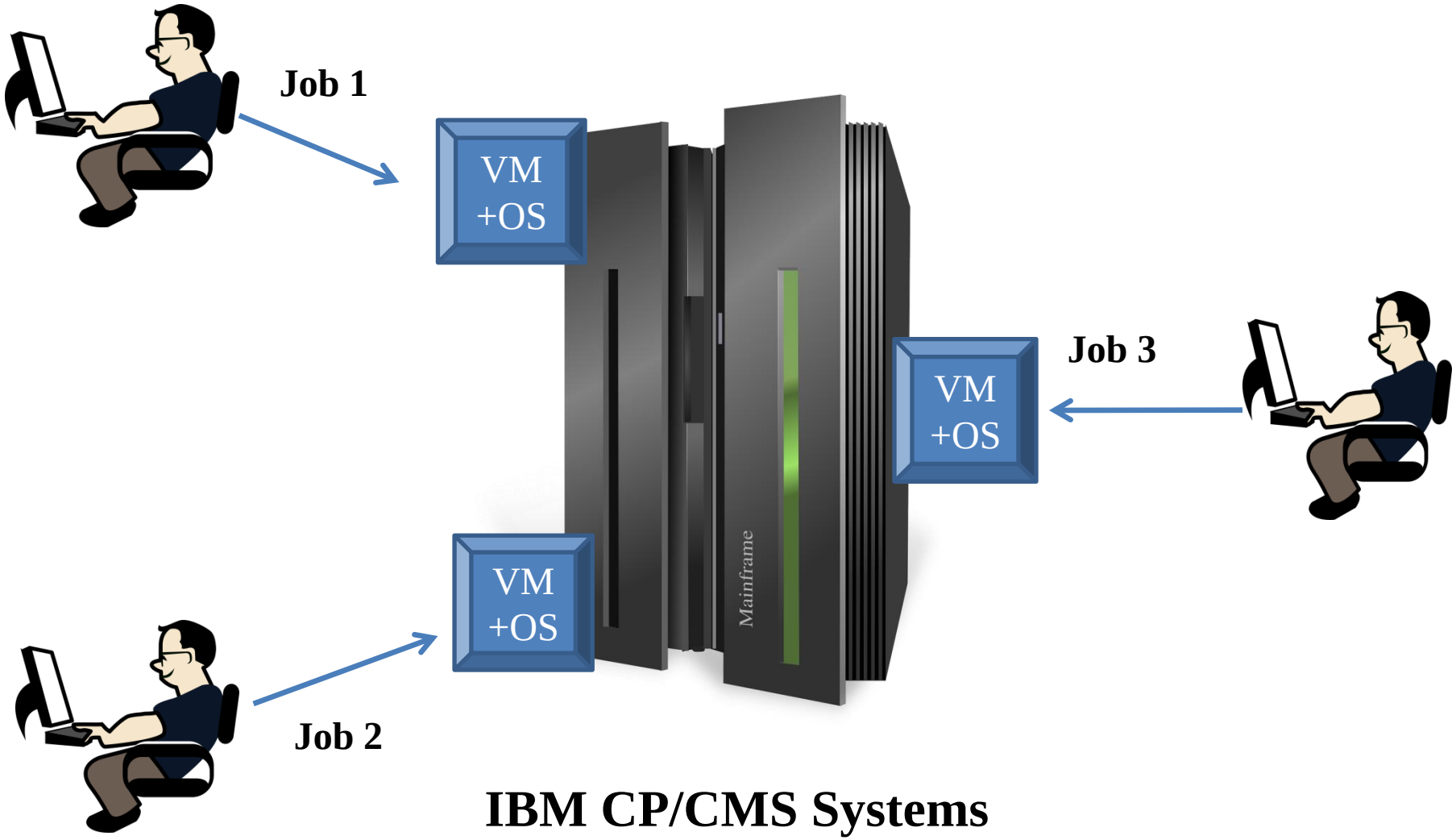


Timesharing in Mainframe Systems

A Brief History of Virtualization

- **Timesharing in Mainframes**
  - Support multiple users through terminals
  - When users block for I/O, system executes jobs from other users
  - System still executes only one job at a time
  - Creates an illusion of multiple jobs being processed at the same time
  - Later, a time quantum was introduced to increase server utilization

A Brief History of Virtualization

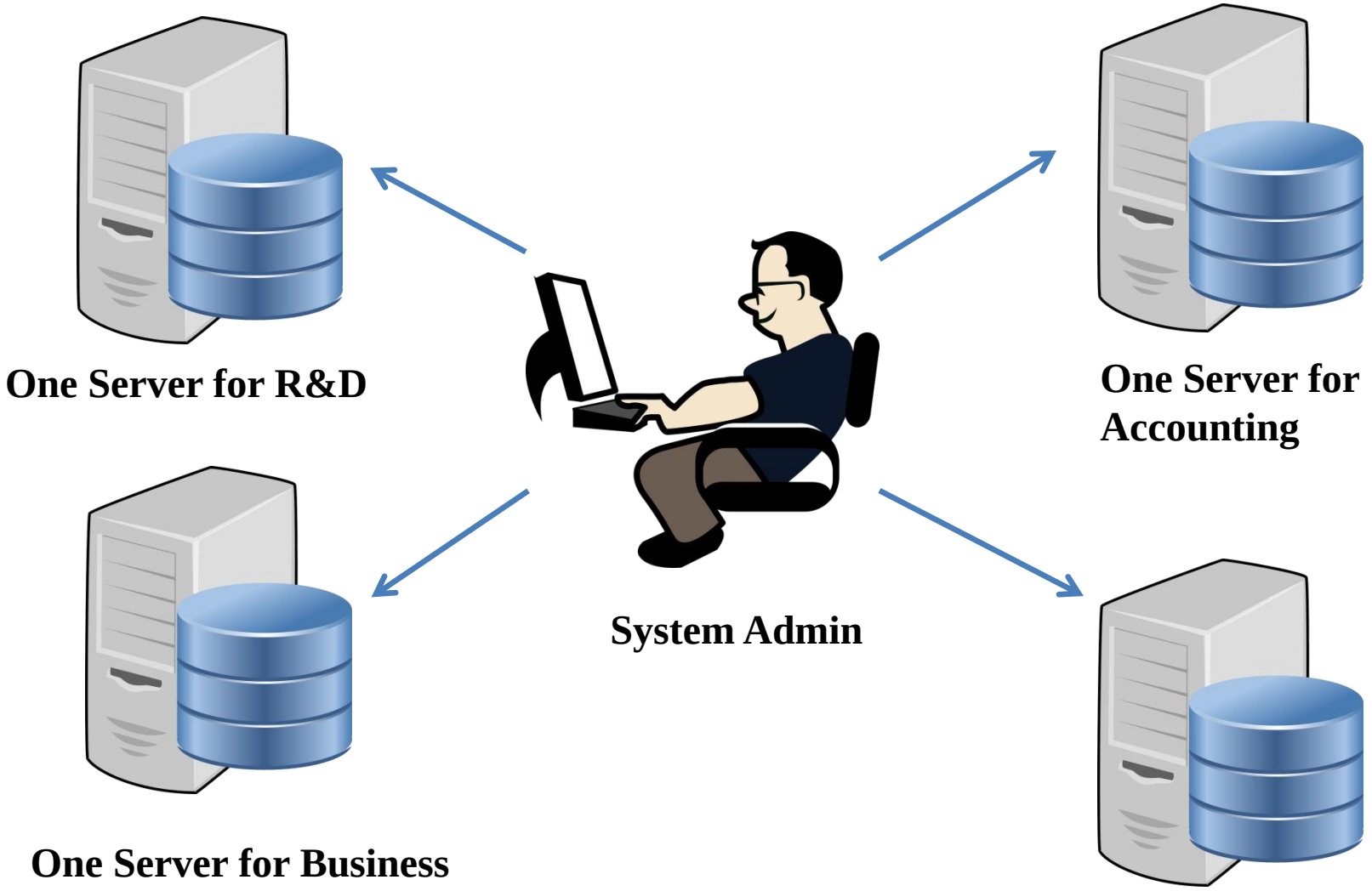


IBM CP/CMS Systems

A Brief History of Virtualization

- **IBM CP/CMS Systems**
  - First virtualized operating system
  - Every user gets a separate “virtual machine” for operating
  - Every user interacts with their own version of OS
  - No concept of time sharing – multiple tasks can be run simultaneously
  - No conflicts between users, so more reliable
  - The rise of personal computers led to a small decline in the importance of virtualization for a period of time

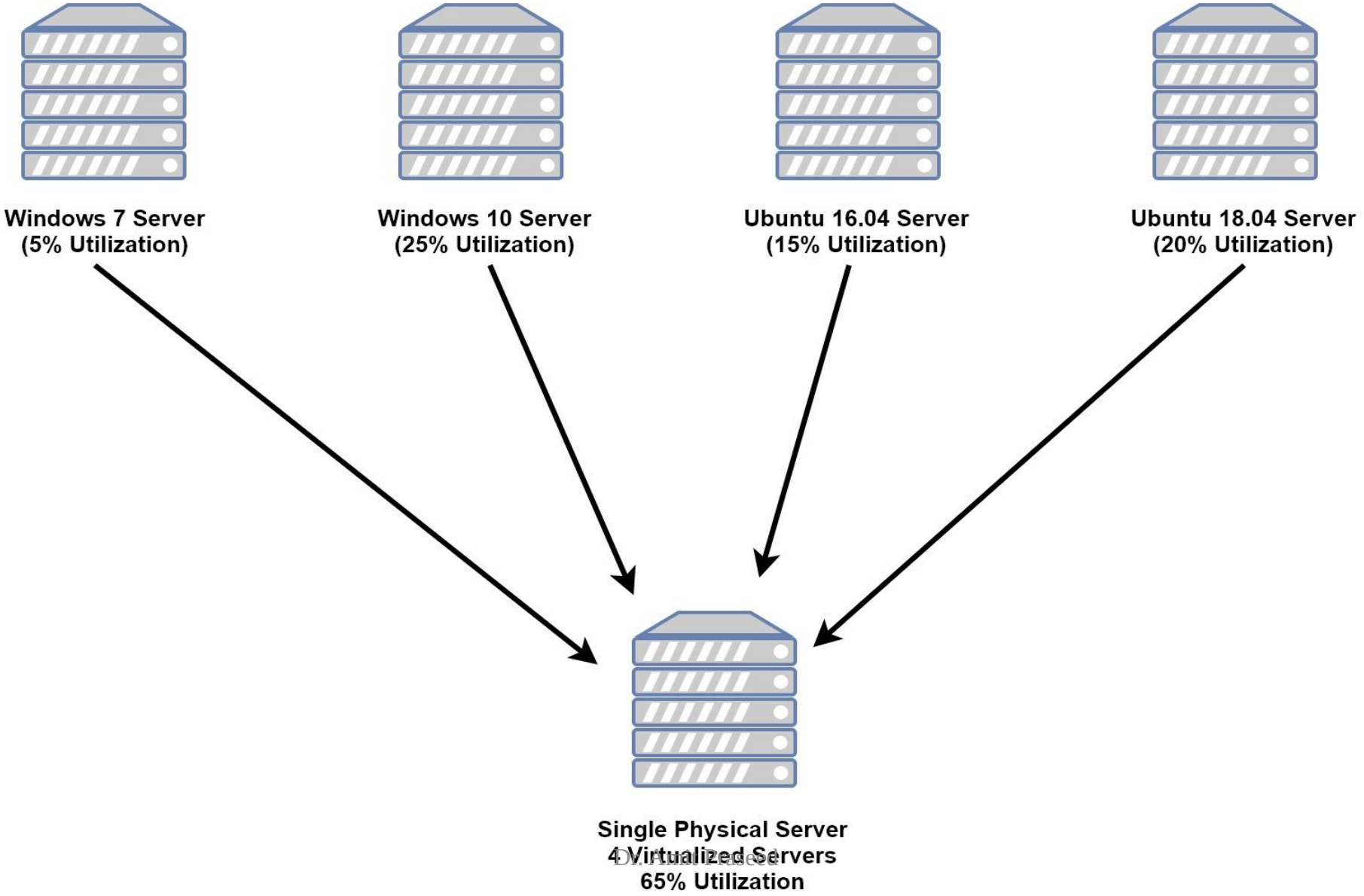
Need for Virtualization in Data Centres



Need for Virtualization in Data Centres

- System administrators allocated one machine per application
  - Increased stability – what if one application interfered with the other?
  - Increased security – hiding “sensitive” data
- Issues
  - Increased capital cost
  - Low server utilization

Virtualization



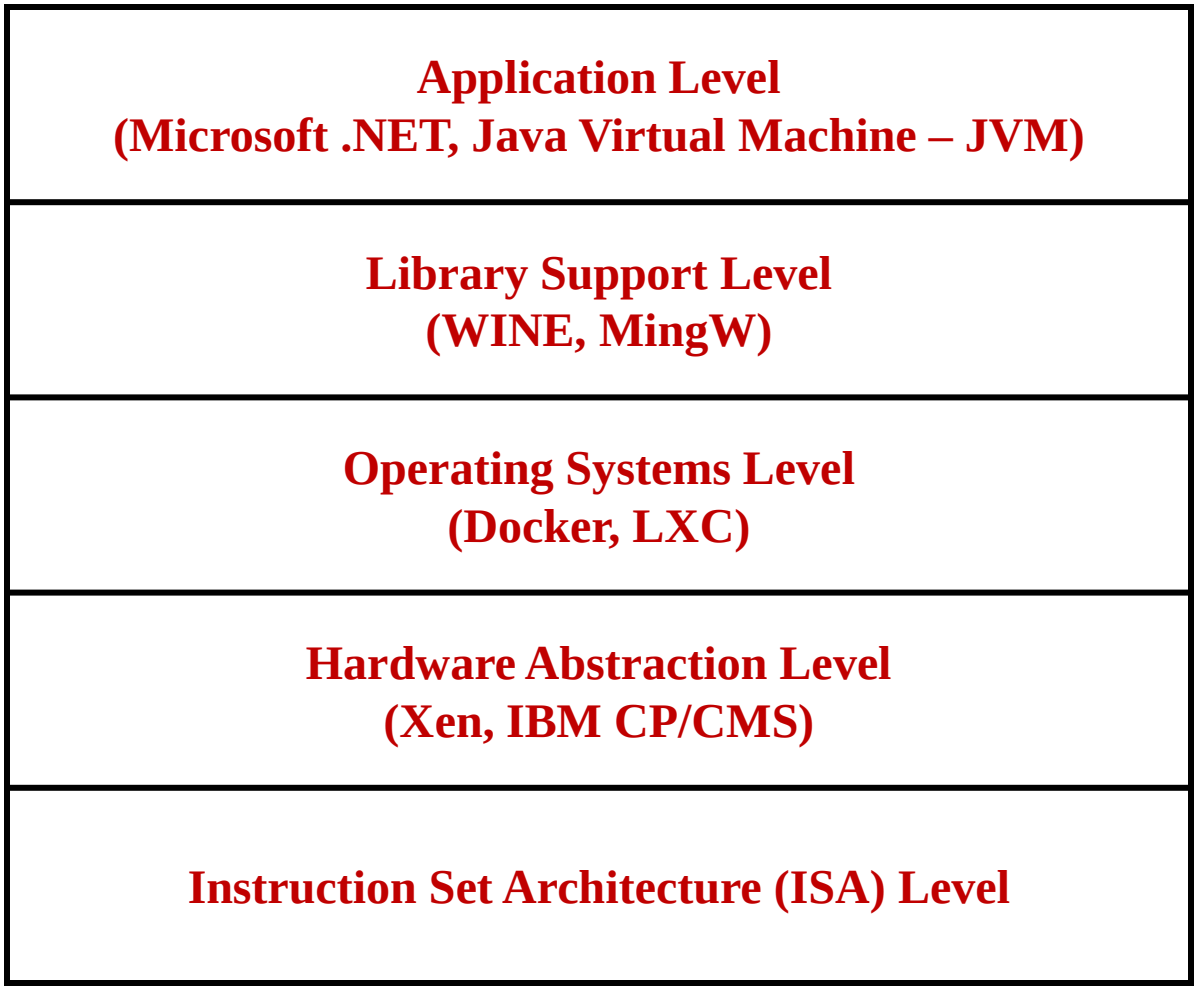
Virtualization

- Process of creating a function of a resource simulated or emulated in software identical to that of the corresponding physical resource
- **Two key points**:
  - It is a software simulation of a physical resource
  - Users must be able to use the virtualized resource exactly as they would use a physical resource

What can be virtualized?

- Desktop
- Application
- Server
- Storage
- Network

Levels of Virtualization



Merits of Different Types of Virtualization

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXX	XXX	XXX
Hardware-level virtualization	XXXX	XXX	XXXX	XXX
OS-level virtualization	XXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXX	XXXX

Conditions for Effective Virtualization

- **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the control program
- **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the control program is to be invoked upon any attempt.
- **Equivalence** : Any program *K* executing with a control program resident performs in a manner indistinguishable from the case when the control program did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended

[Popek and Goldberg 1974: "Formal Requirements for Virtualizable Third Generation Architectures" Communications of the ACM]

A Simple Solution - Emulation

- A machine could simply be emulated in software
  - File to represent disk
  - Case-statement to implement individual opcodes
  - Registers could be implemented as variables
- This violates the efficiency criteria!!!
  - Software does not allow effective implementations of hardware mechanisms such as interrupts
  - Each guest instruction is executed by several host instructions.
- **Instead, can we directly use the host hardware?**
  - VM executes as a process on host
  - Host processor executes its instructions.

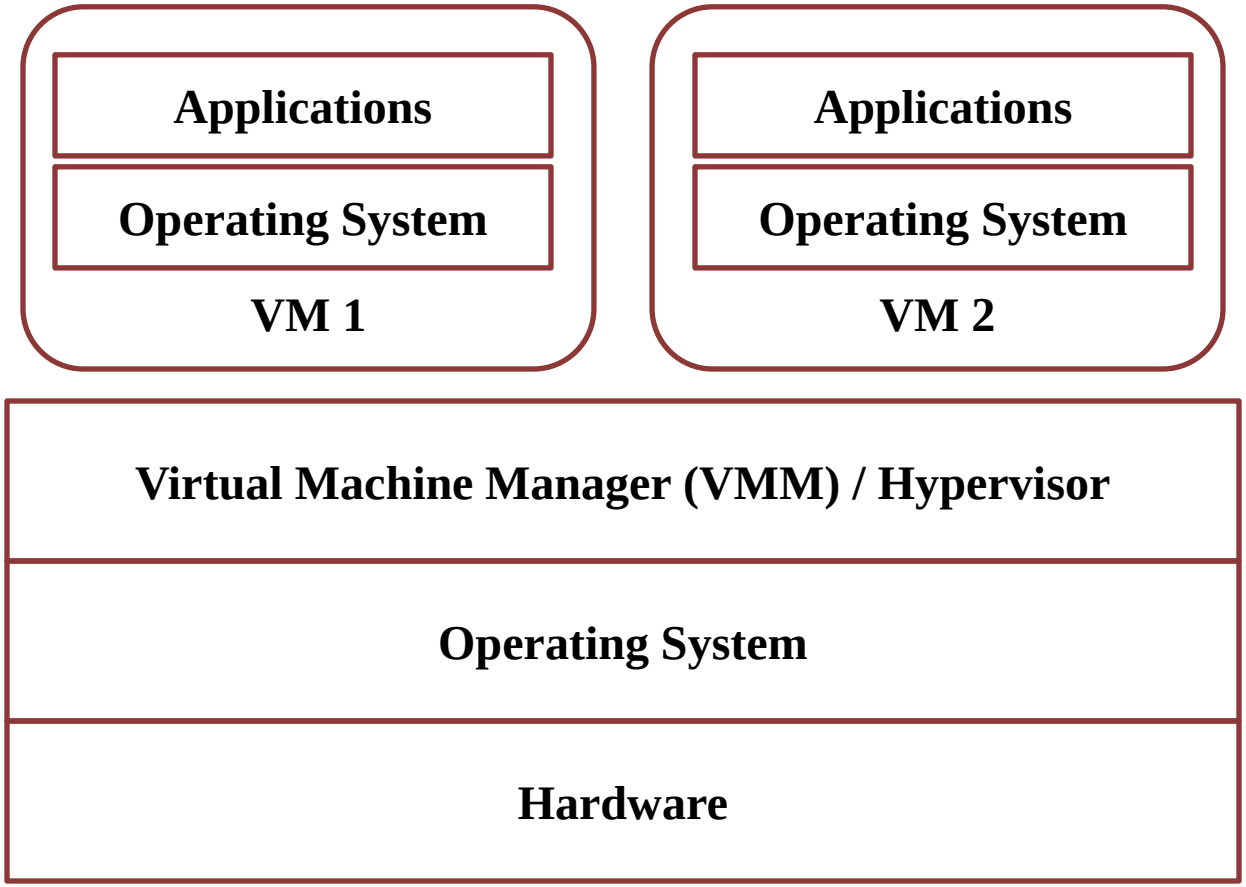
Revisiting the Rules

- **Rules for Efficient Virtualization**
  - **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the **control program**
  - **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the **control program** is to be invoked upon any attempt.
  - **Equivalence** : Any program *K* executing with a **control program** resident performs in a manner indistinguishable from the case when the **control program** did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended
- The "CONTROL PROGRAM" is usually a layer of software that mediates between the VMs and the underlying hardware
- Called a Virtual Machine Manager (VMM) or hypervisor

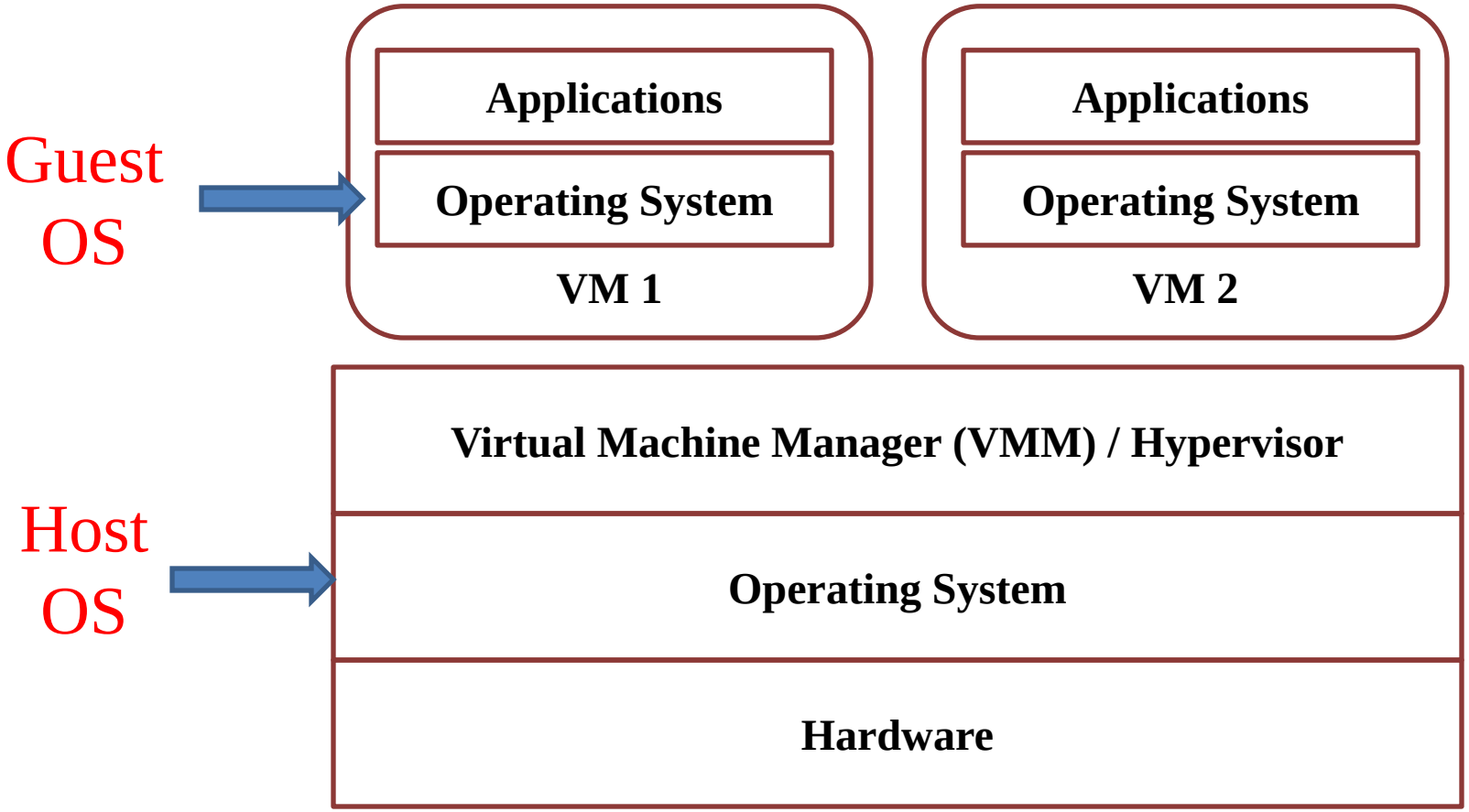
Role of Hypervisor

- Provide an environment for programs which is essentially identical to the original machine
- Ensure that programs run in this environment should show, at worst, only minor decreases in speed
- Ensure complete control of the system resources.
  - Allocation
  - Separation
  - Preemption

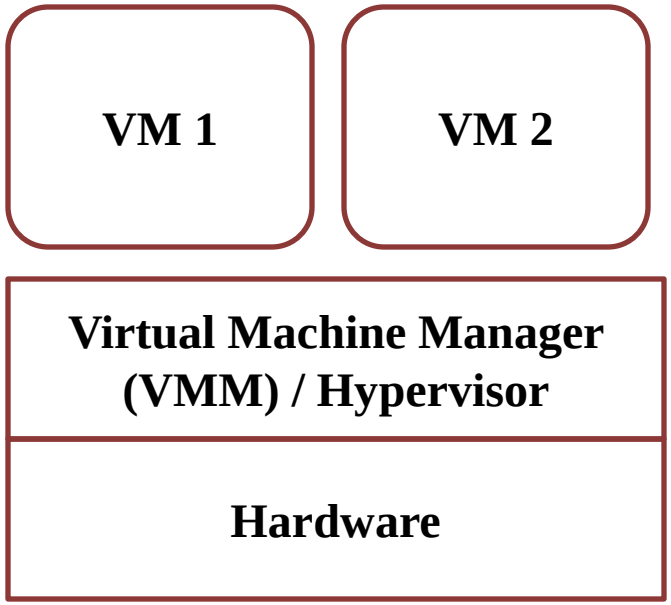
A General Architecture



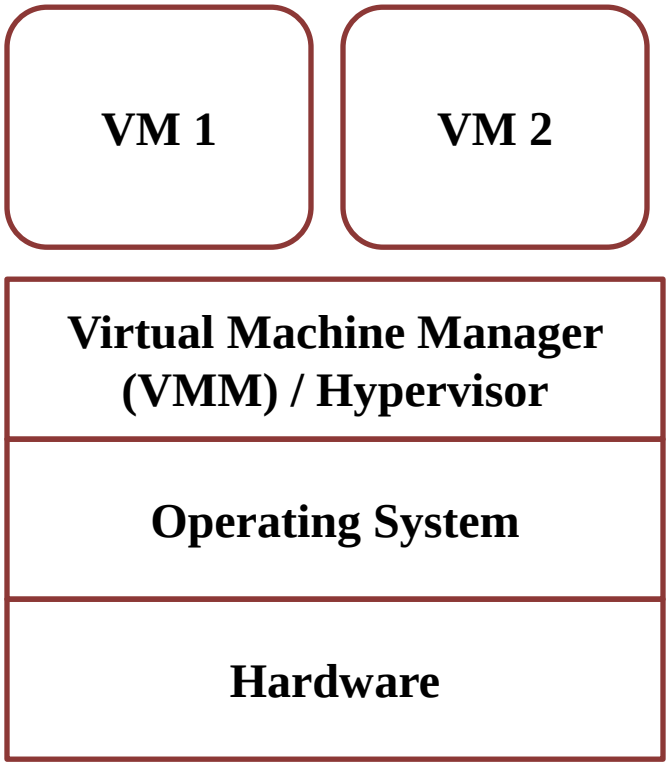
A General Architecture



Types of Hypervisors



Type 1 (Bare-Metal) Hypervisor



Type 2 (Hosted) Hypervisor

Comparison of Hypervisors

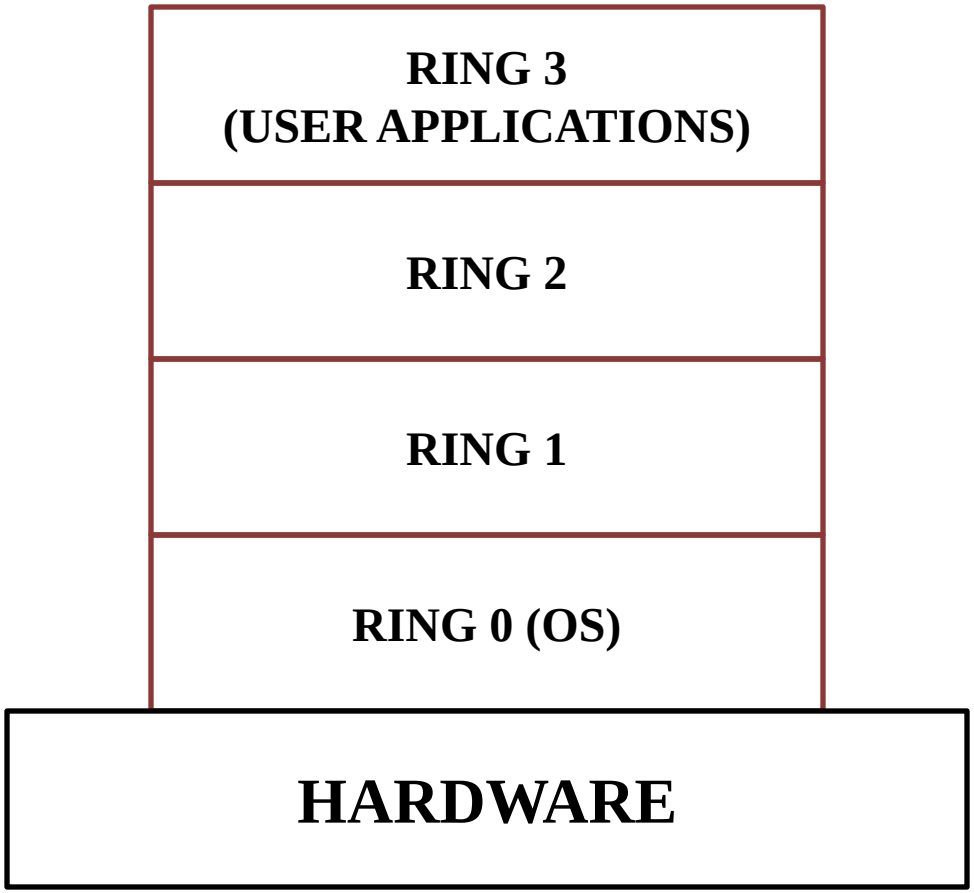
Type - 1 Hypervisor

- Resides directly on the hardware (“bare metal”)
- Communicates directly with the hardware resources
- More efficient
- More secure
- Eg: Citrix/Xen Server, VMware ESXi and Microsoft Hyper-V

Type – 2 Hypervisor

- Resides on top of the operating system (“hosted”)
- Communicates with hardware through the OS
- Less efficient
- Less secure
- Eg: Oracle Virtual Box, VMware Workstation etc.

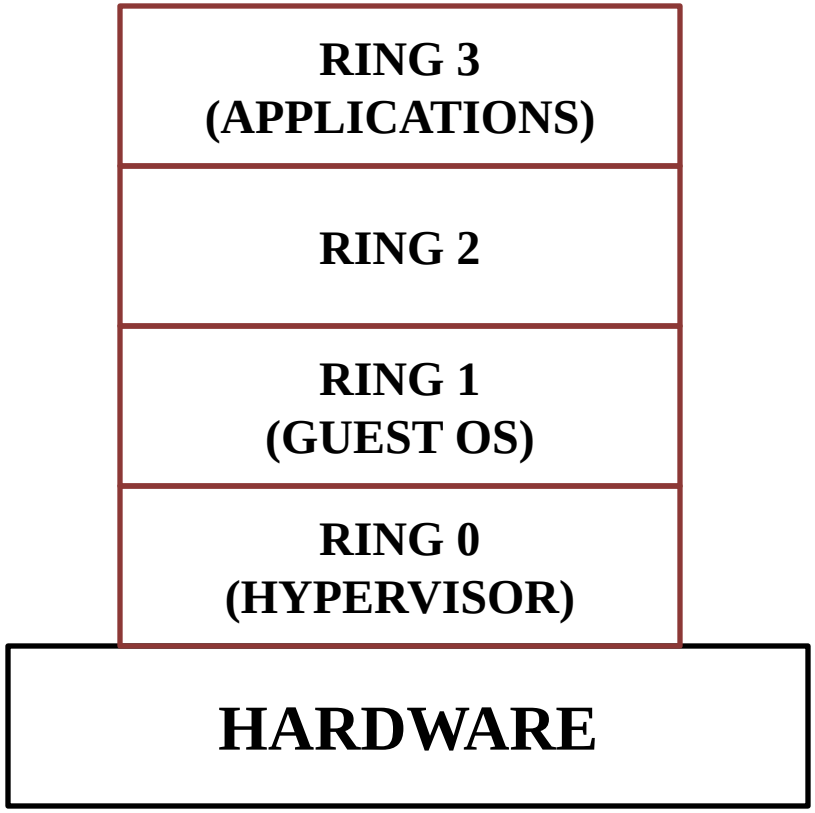
“Protection Ring” Concept



The Difficulty with Virtualization

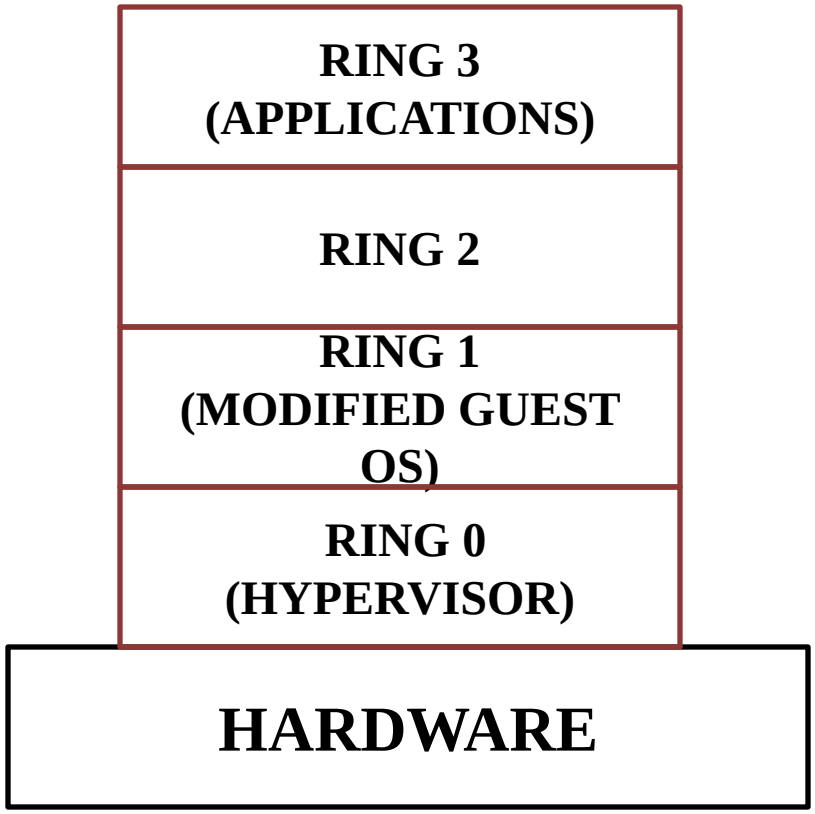
- All modern operating systems are built to run at Ring 0
  - They are designed to issue privileged instructions designed to modify memory and hardware directly
- For virtualization, guest OS resides on top of a hypervisor
  - Guest OS can only operate at a Ring > 0
  - This causes problems when the guest OS issues privileged instructions
  - The hypervisor must intercept and translate privileged instructions before passing it over to the hardware

Full Virtualization



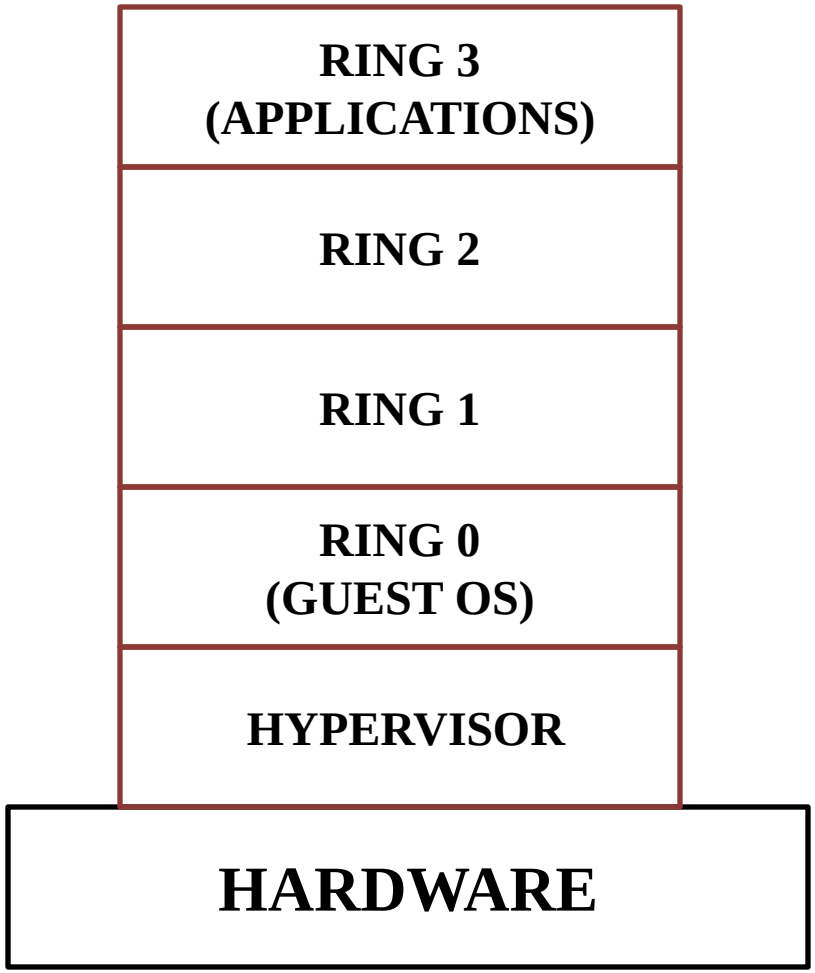
- Hypervisor operates at Ring 0
- Hypervisor scans the request stream
  - Captures and translates privileged instructions
  - Guest OS thinks it is directly working with hardware
- Performance is impacted due to binary translation

Para Virtualization



- Hypervisor resides in a privileged layer beneath the guest OS
- Guest OS is modified, so it doesn't execute privileged instructions
  - It executes hypercalls to the hypervisor
- Better performance
  - Limited use due to the need to modify OS

Hardware Assisted Virtualization



- Hardware allows hypervisor to reside in a privileged ring
- Privileged and sensitive calls are set to automatically trap to the hypervisor
- Can use unmodified OS + better performance
- Requires hardware support

Types of Virtualization

Type of Virtualization	Requires Hardware Support?	Requires Guest OS Modification?
Full Virtualization	No	No
Para virtualization	No	Yes
Hardware Assisted Virtualization	Yes	No