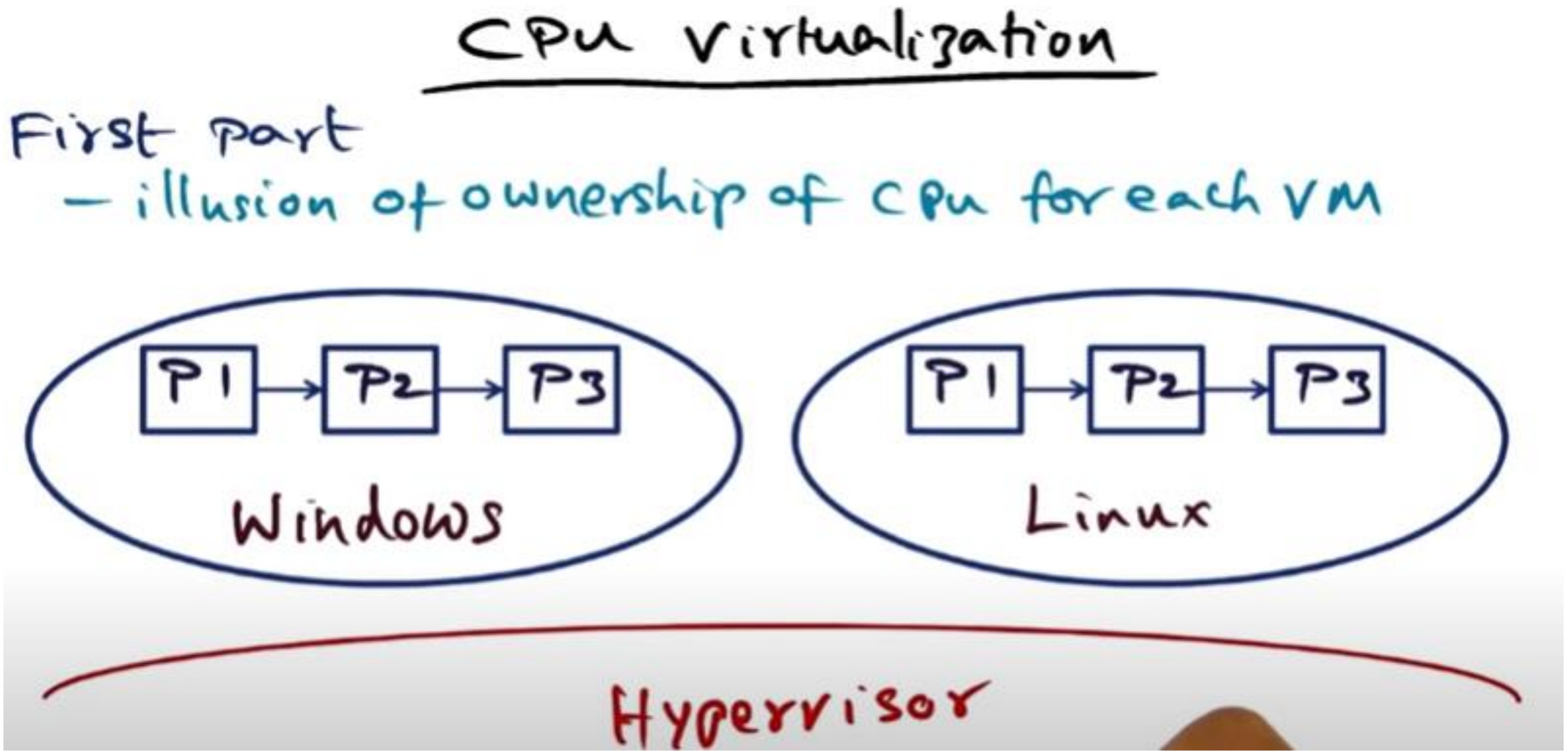
CPU Virtualization

SOMAIYA VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering

Somaiya TRUST

CPU Virtualization



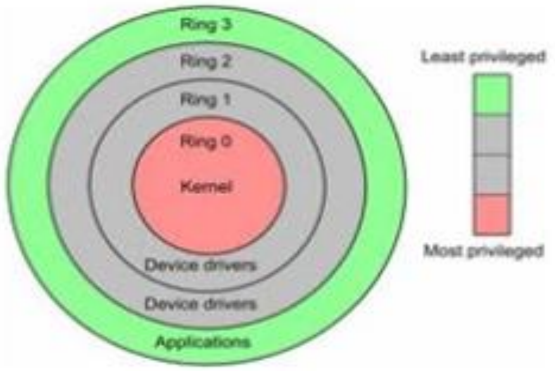
CPU

CPU Virtualization

- With CPU Virtualization, all the virtual machines act as physical machine and distribute their hosting resources just like having various processors.
- The virtual machines get a share of the single CPU allocated to it, being a single-processor acting as dual-processor.

CPU Architecture

- Modern CPU status is usually classified as several modes.
- In general, we conceptually divide them into two modes :
  - Kernel mode (Ring 0)
    - CPU may perform any operation allowed by its architecture, including any instruction execution, IO operation, area of memory access, and so on.
    - Traditional OS kernel runs in Ring 0 mode.
  - User mode (Ring 1 ~ 3)
    - CPU can typically only execute a subset of those available instructions in kernel mode.
    - Traditional application runs in Ring 3 mode.



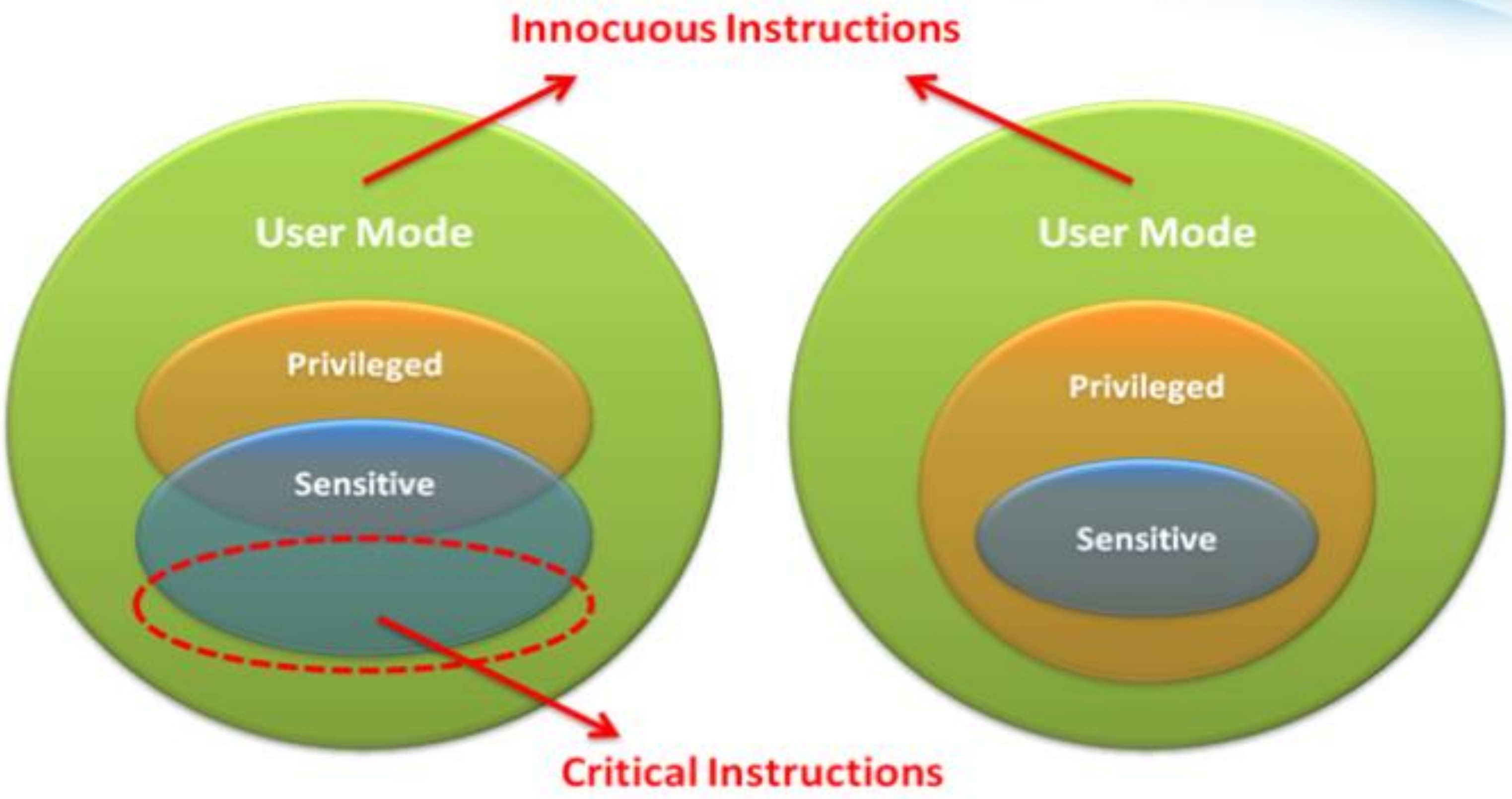
SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Instructions

- Unprivileged instructions:
  - without interfering other task, no shared resources,
  - arithmetic instructions
- Privileged instructions
  - Execute under specific restrictions in a privileged mode and will be trapped if executed outside this mode.
    - Control-sensitive (modify) instructions
      - Attempt to change the configuration of shared resources used. I/O instructions
    - Behavior-sensitive(expose) instructions.
      - Alter the state of CPU registers.

CPU Architecture

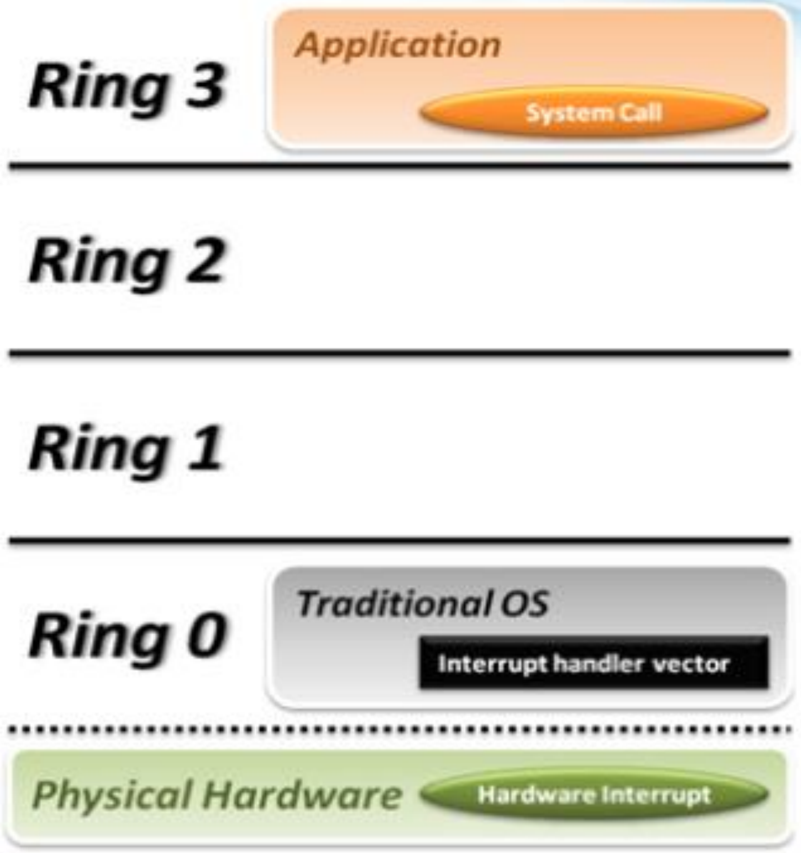


Software based CPU Virtualization

- Application code gets executed then privileged code gets translated first and that translated code gets executed directly on the processor.
- This translation is purely known as Binary Translation (BT).
- The guest programs that are based on unprivileged coding runs very smooth and fast.
- The code programs or the applications that are based on privileged code components that are significant such as system calls, run at a slower rate in the virtual environment.

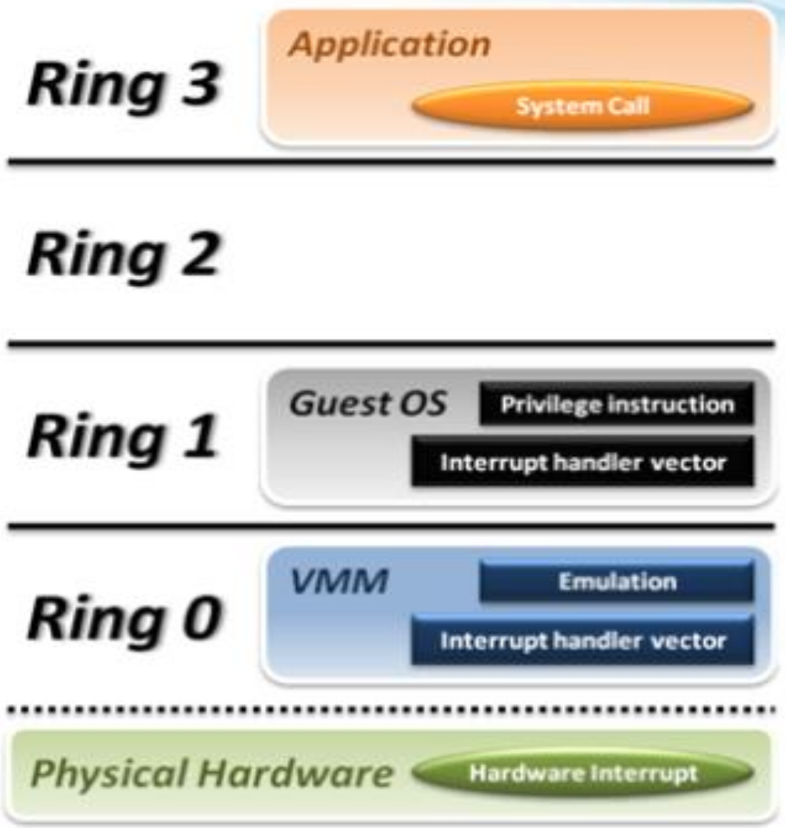
Trap and Emulate Model

- Traditional OS :



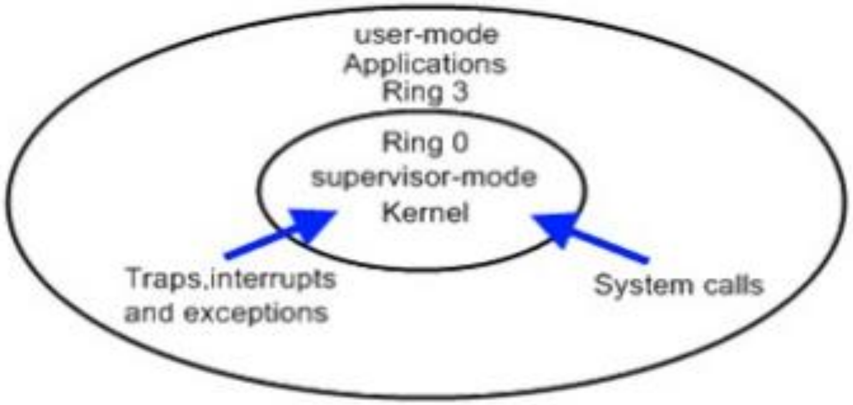
Trap and Emulate Model

- VMM and Guest OS :



Trap and Emulate Model

- VMM virtualization paradigm *(trap and emulate)*:
    1. Let normal instructions of guest OS run directly on processor in user mode.
    2. When executing privileged instructions, hardware will make processor trap into the VMM.
    3. The VMM emulates the effect of the privileged instructions for the guest OS and return to guest.



Hardware-Assisted Virtualization

- There is hardware that gets assistance to support CPU Virtualization from certain processors.
- The guest user uses a different version of code and mode of execution known as a guest mode.
- The guest code mainly runs on guest mode.
- System calls runs faster than expected.
- Workloads that require updation of page tables get a chance of exiting from guest mode to root mode that eventually slows down the performance and efficiency of the program.

Hardware Solution

- Let's go back to trap model :
  - Some trap types do not need the VMM involvement.
    - For example, all system calls invoked by application in guest OS should be caught by gust OS only. There is no need to trap to VMM and then forward it back to guest OS, which will introduce context switch overhead.
  - Some critical instructions should not be executed by guest OS.
    - Although we make those critical instructions trap to VMM, VMM cannot identify whether this trapping action is caused by the emulation purpose or the real OS execution exception.

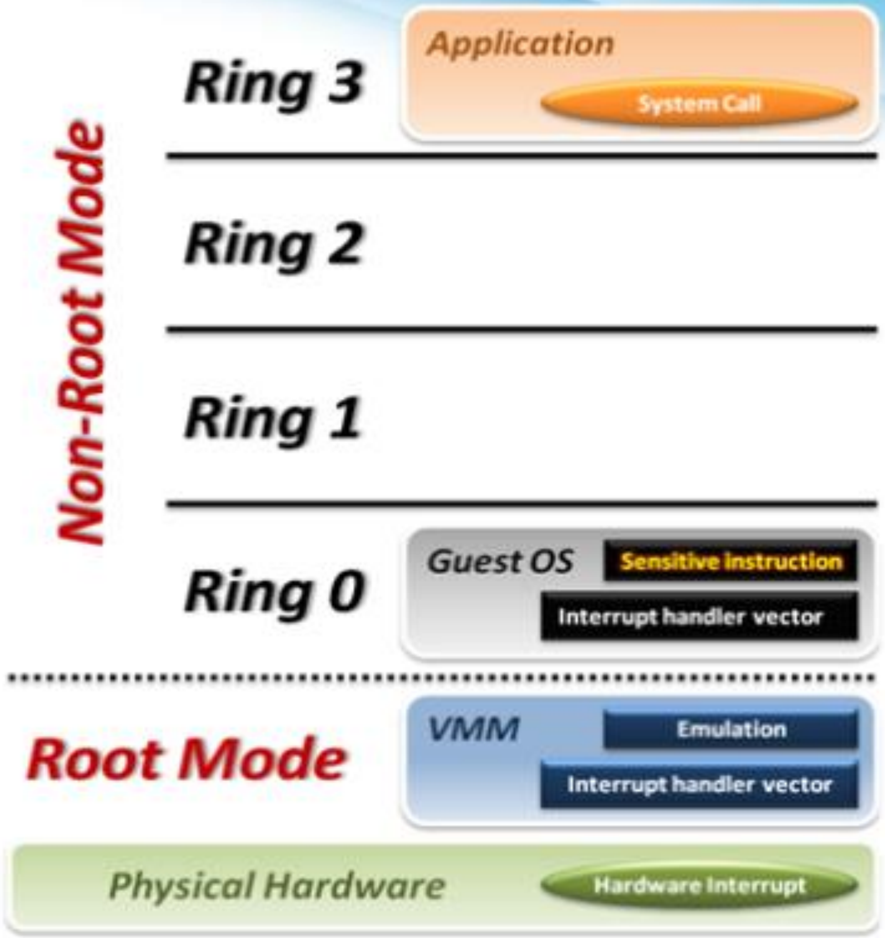
- Solution :
  - We need to redefine the semantic of some instructions.
  - We need to introduce new CPU control paradigm.

Intel VT-x

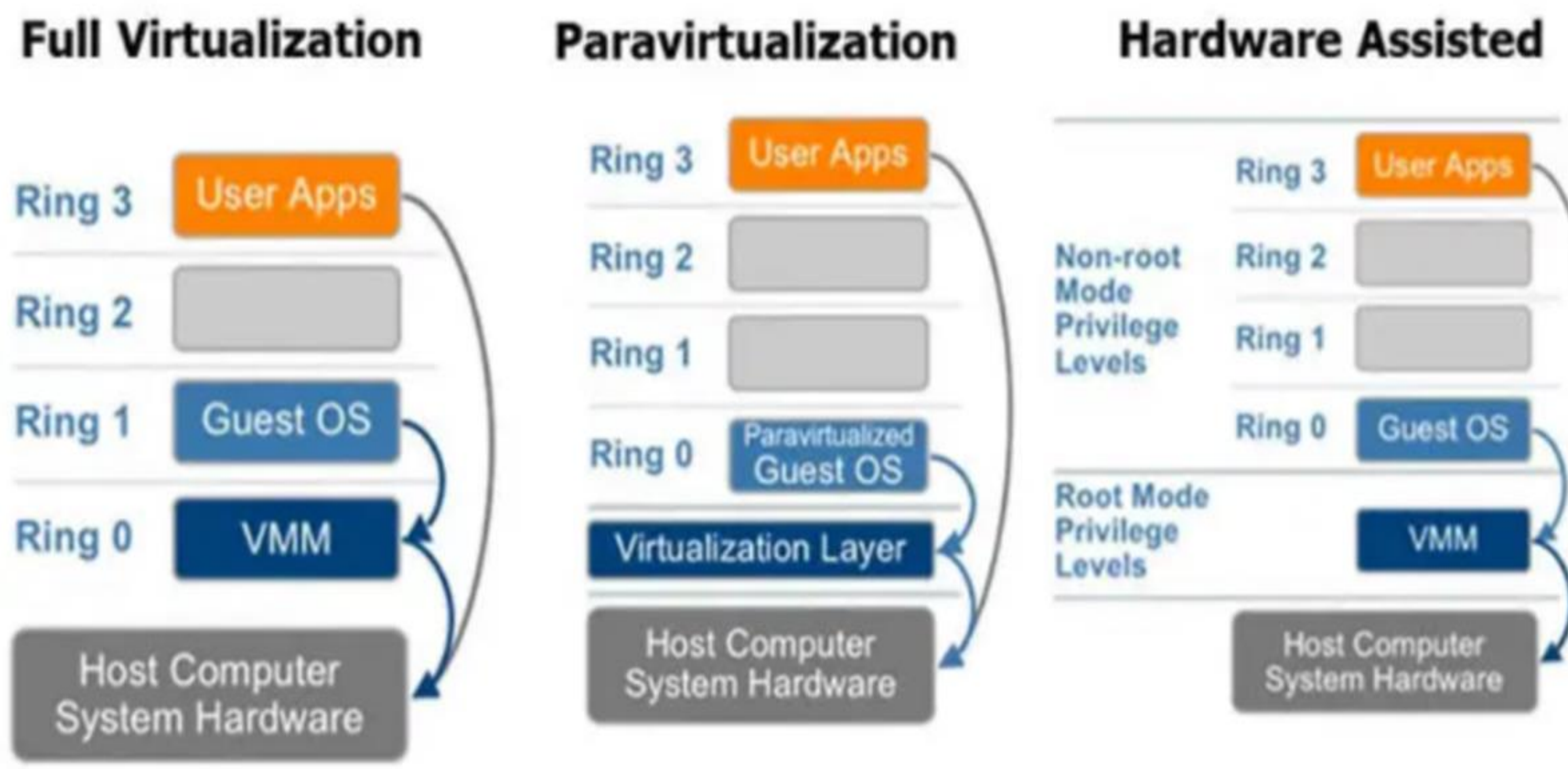
- In order to straighten those problems out, Intel introduces one more operation mode of x86 architecture.
  - VMX Root Operation (Root Mode)
    - All instruction behaviors in this mode are no different to traditional ones.
    - All legacy software can run in this mode correctly.
    - VMM should run in this mode and control all system resources.
  - VMX Non-Root Operation (Non-Root Mode)
    - All sensitive instruction behaviors in this mode are redefined.
    - The sensitive instructions will trap to Root Mode.
    - Guest OS should run in this mode and be fully virtualized through typical "*trap and emulation model*".

Intel VT-x

- VMM with VT-x :



Architectural Comparison



Benefits of CPU Virtualization

- Overall performance and efficiency are improved
- Security: The VM machines are also kept separate from each other and because of that any cyber-attack or software glitch unable to create damage to the system, as a single machine cannot affect another machine.
- Cost is very less
- It provides the best backup of computing resources since the data is stored and shared from a single system.
- It also offers great and fast deployment procedure options

Virtualization

- Server Virtualization:

  https://slideplayer.com/slide/5103233/
- https://www.educba.com/cpu-virtualization/
- http://www.brainkart.com/article/Virtualization-of-CPU,-Memory,-and-I-O-Devices_11337/
- https://www.slideshare.net/drgst/cs6703-grid-and-cloud-computing-unit-3