Advanced Operating Systems

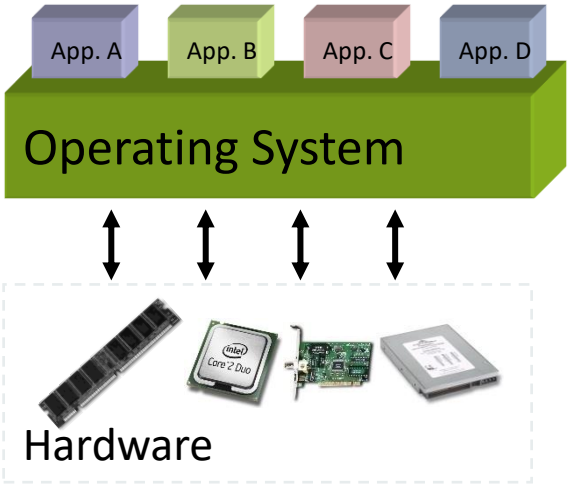
Topic 6. **Virtualization**

Professor:

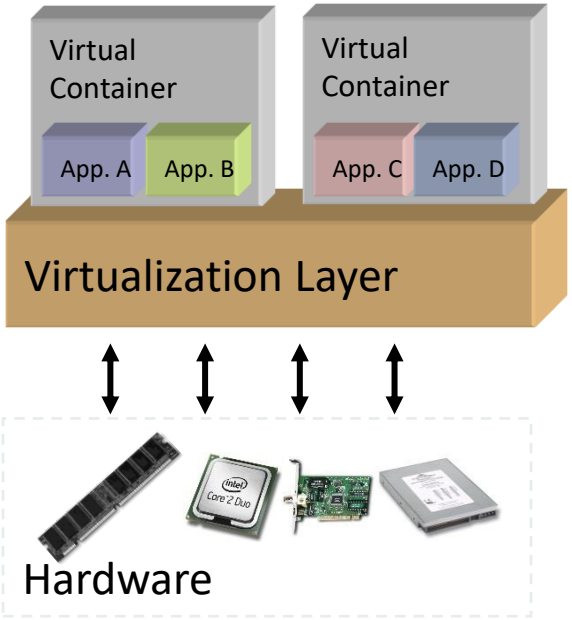
Dr. José Octavio Gutiérrez García

octavio.gutierrez@itam.mx

What is Virtualization?



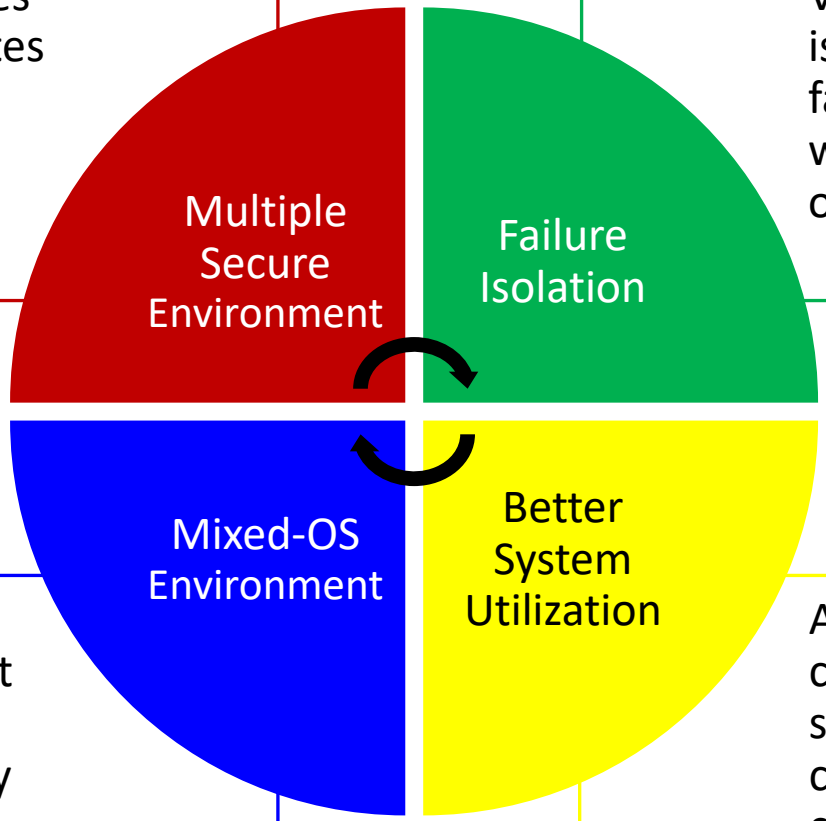
'Nonvirtualized' system
A single OS controls all hardware platform resources



Virtualized system
It makes it possible to run multiple virtual Containers on a single physical platform

Benefits of Virtualization

A system VM provides a sandbox that isolates one system environment from other environments



Virtualization helps isolate the effects of a failure to the VM where the failure occurred

A single hardware platform can support multiple operating systems concurrently

A virtualized system can be (dynamically or statically) re-configured for changing needs

Virtualization Properties

- Fault Isolation
- Software Isolation
- Performance Isolation (accomplished through scheduling and resource allocation)

Isolation

- All VM state can be captured into a file (i.e., you can operate on VM by operating on file– cp, rm)

Encapsulation

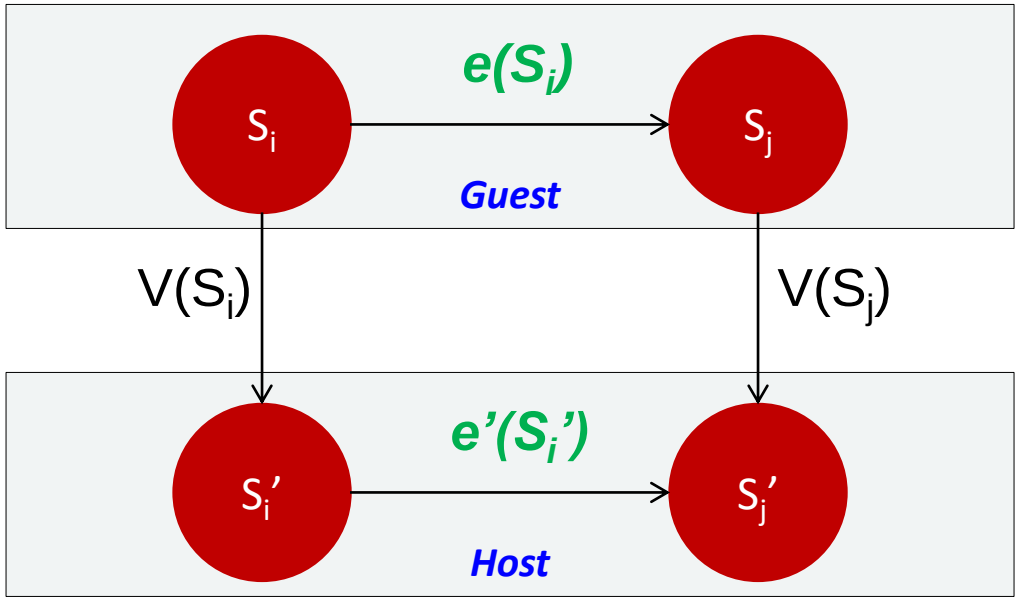
- All guest actions go through the virtualizing software which can inspect, modify, and deny operations

Interposition

What is Virtualization?

- Virtualization basically allows one computer to do the job of multiple computers, by **sharing the resources of a single hardware across multiple environments**

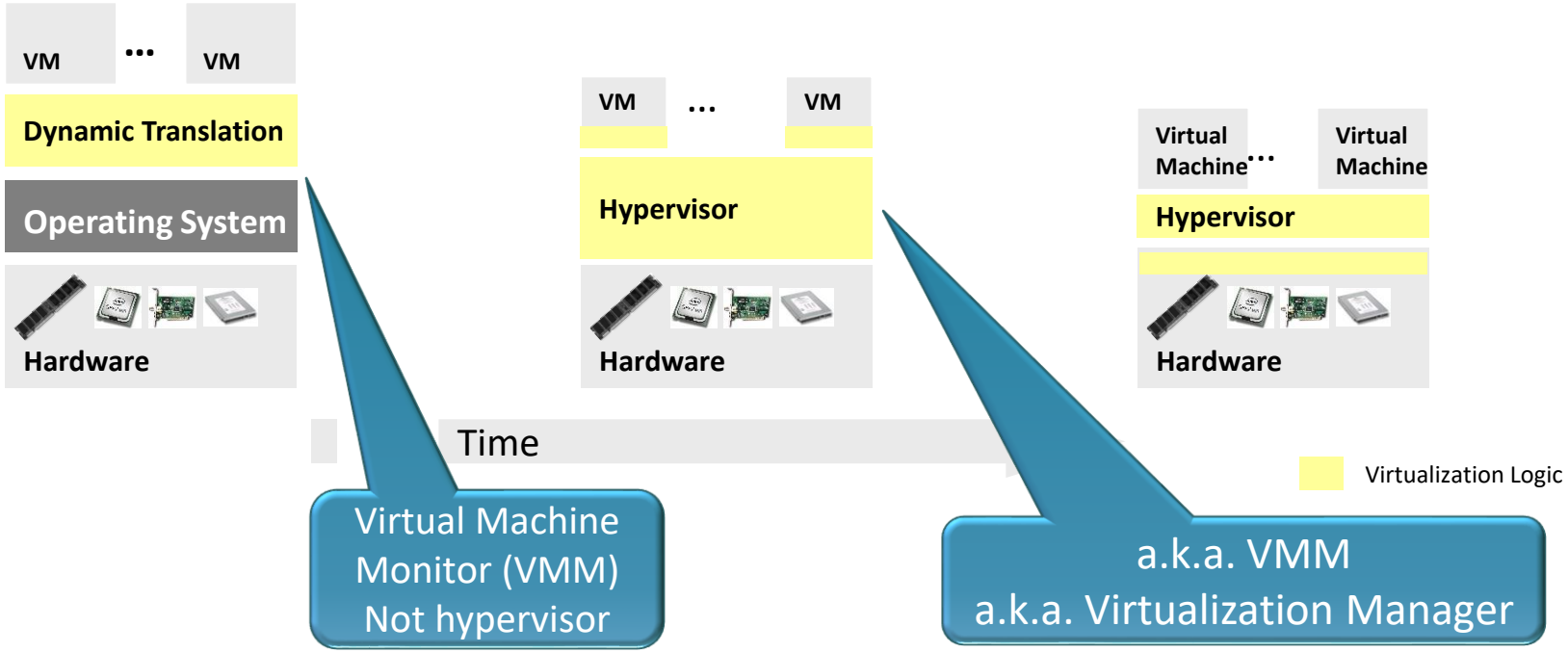
- Formally, virtualization involves the construction of an **isomorphism** that *maps* a virtual guest system to a real host system (Popek and Goldberg 1974)



Introduced by IBM in the 1960s

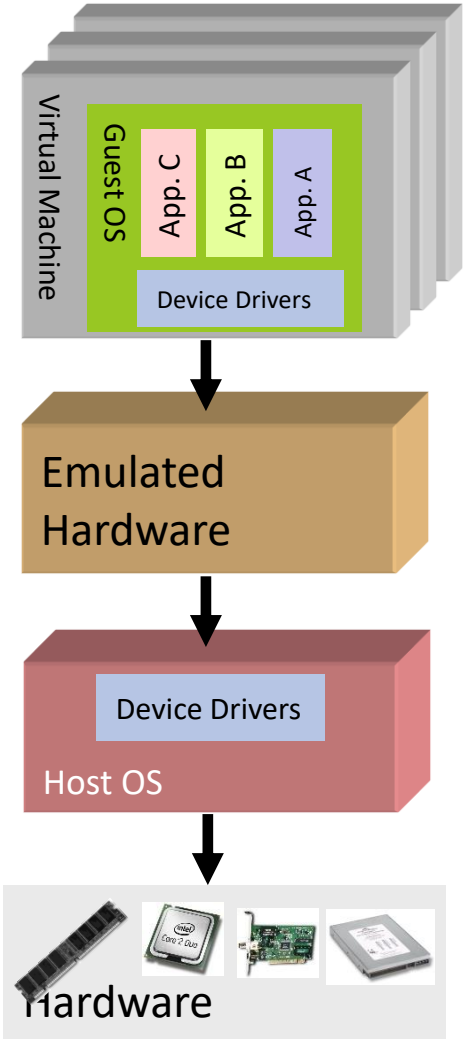
Server virtualization approaches

- 1st Generation: Full virtualization (Binary translation)
  - Software Based
- 2nd Generation: Paravirtualization
  - Cooperative virtualization
  - Modified guest
- 3rd Generation: Silicon-based (Hardware-assisted) virtualization
  - Unmodified guest



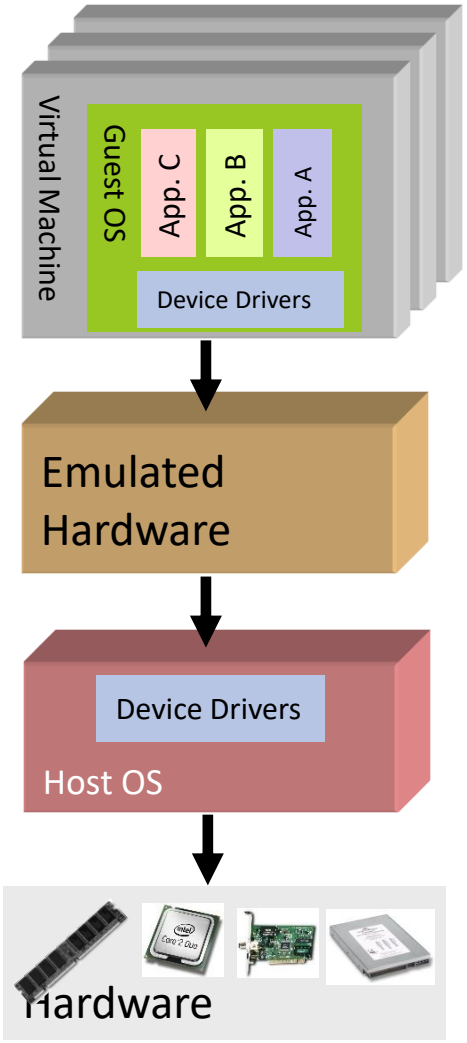
Full virtualization

- Runs **unmodified** guests
- Dynamic binary translation
- **Software emulates** CPU (full instruction set), input/output operations, interrupts, memory access, motherboard, device buses, etc.
  - Guests cannot access hardware
- **Examples:** QEMU, VMWare, Xen HVM, KVM, Microsoft VM, Parallels, virtualbox



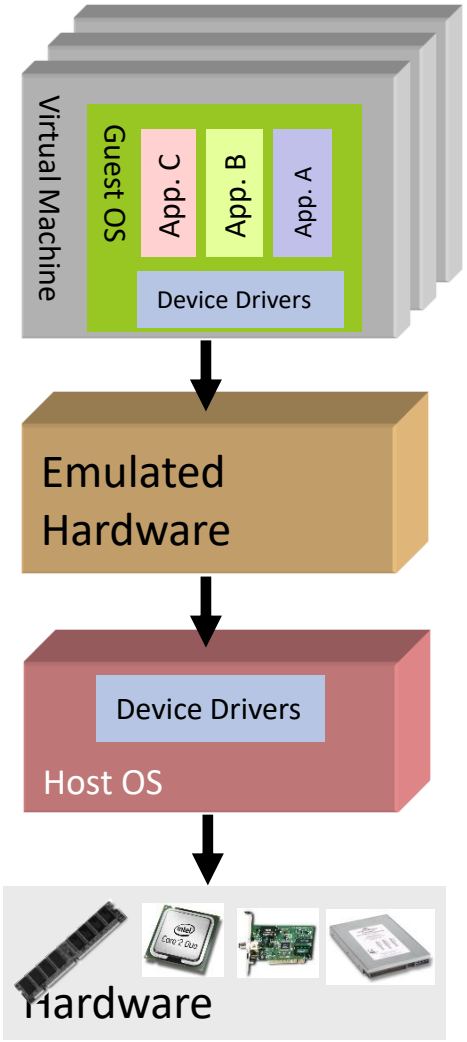
Full virtualization - Advantages

- The emulation layer
  - **Isolates** VMs from the host OS and from each other.
  - **Controls** individual **VM access to** system **resources**, preventing an unstable VM from impacting system performance.
- Total VM portability
  - VMs have the ability to transparently **move between hosts** with **dissimilar hardware**
  - It is possible to **run** an **OS** that was developed for **another architecture** on your own architecture



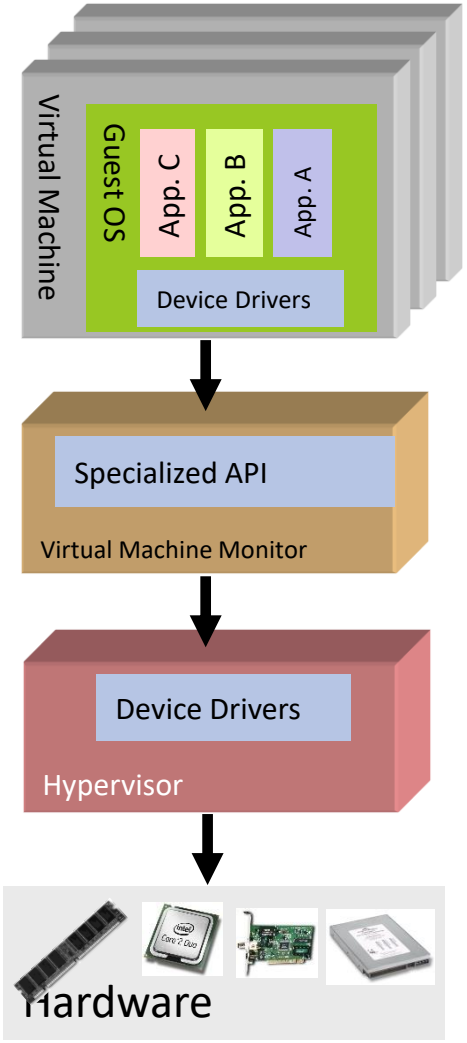
Full virtualization - Disadvantages

- Hardware emulation comes with a performance price. It's the worst performance.
  - ~ usually **less than 2%** for emulated devices such as RAM
  - **from 8% to 20%** for input/output – intensive devices such as network cards and hard disks.



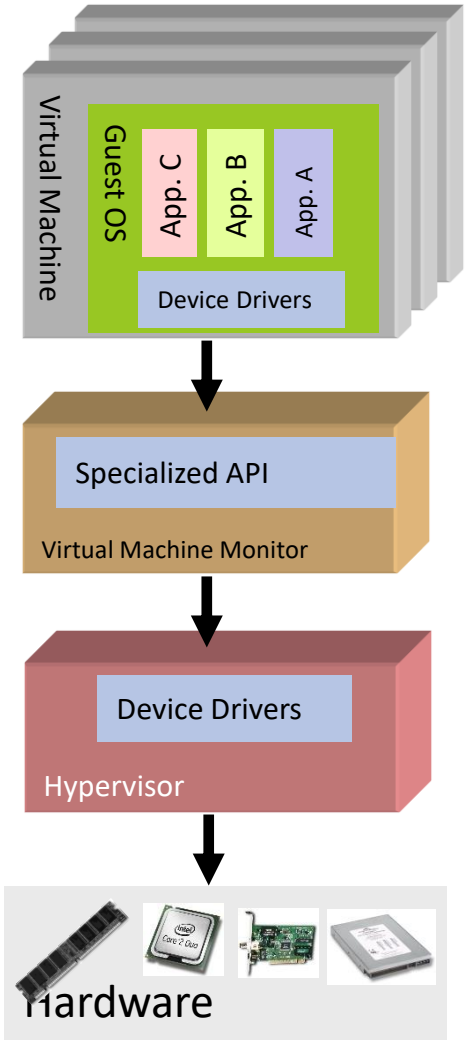
Paravirtualization

- The **Guest OS is modified** and thus
  - The guest is fully aware of how to process **privileged instructions** (access to I/O devices, interrupt handling, etc.)
  - The **guest** operating system **uses** a specialized **API to talk to the VMM** and, in this way, executes the privileged instructions
  - With appropriate device drivers in its kernel, the **guest OS** is now capable of directly **communicating** with the **system hardware**.



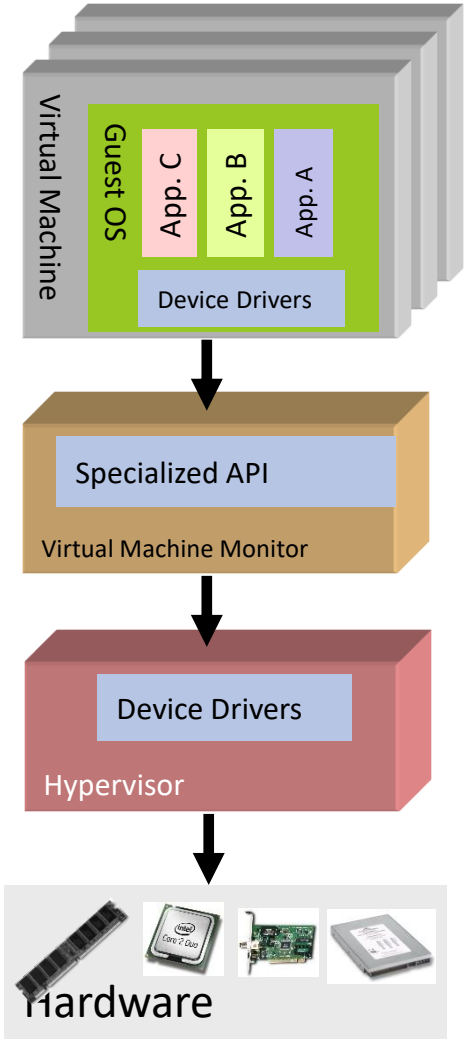
Paravirtualization

- VM ***guest OSs*** are paravirtualized using two different approaches:
  - Recompiling the OS kernel
  - Installing paravirtualized drivers (**partial paravirtualization**)



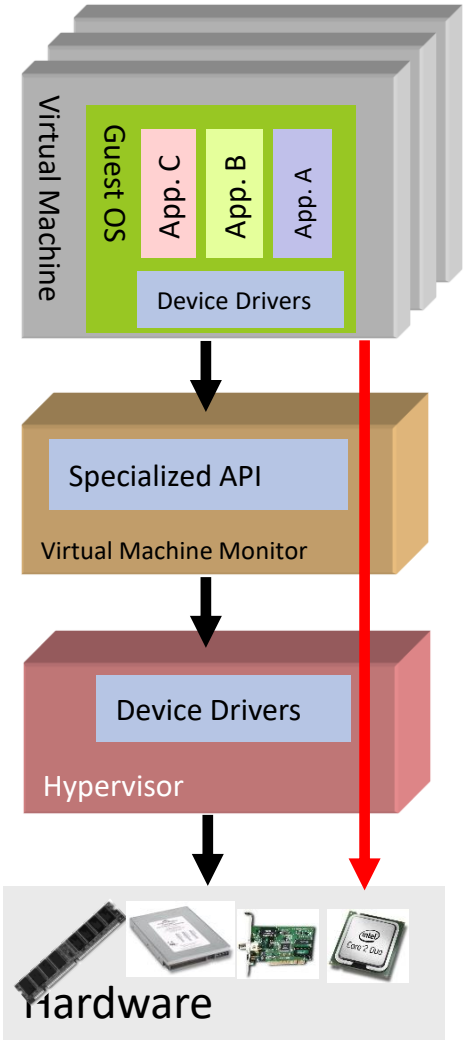
Paravirtualization

- Advantages
  - Better performance than full virtualization.
- Disadvantages
  - Modify the guest OS to cooperate with the VMM
  - In some OSs it is not possible to use complete paravirtualization
  - Guest OS and hypervisor tightly coupled
  - Guest kernel must be recompiled when hypervisor is updated



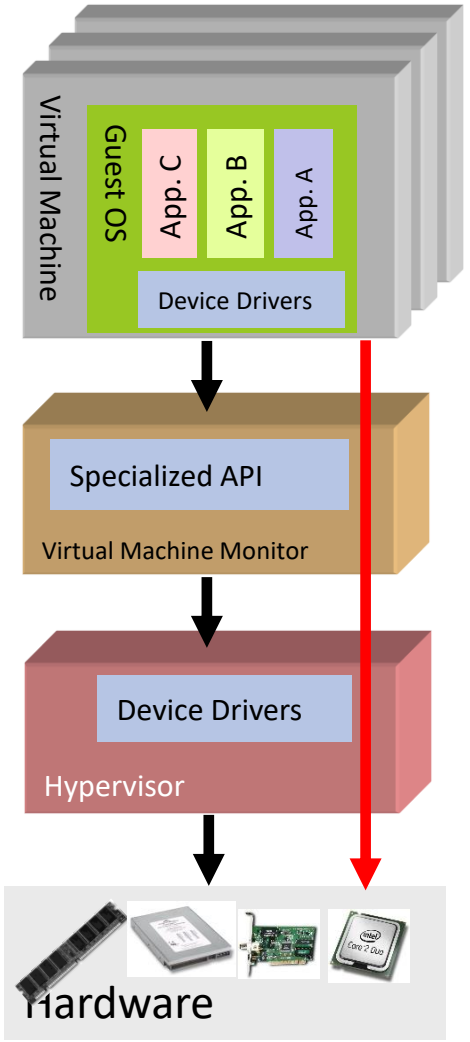
Hardware-assisted virtualization

- **Guest OSs** can process **privileged instructions** without the need for any translation on the part of the VMM
- The VMM uses **processor extensions** (such as Intel®-VT or AMD-V) to intercept and emulate privileged operations in the guest
- Dedicated address space that is assignable to each VM
  - **Chips** containing **Extended Page Tables**



Hardware-assisted virtualization

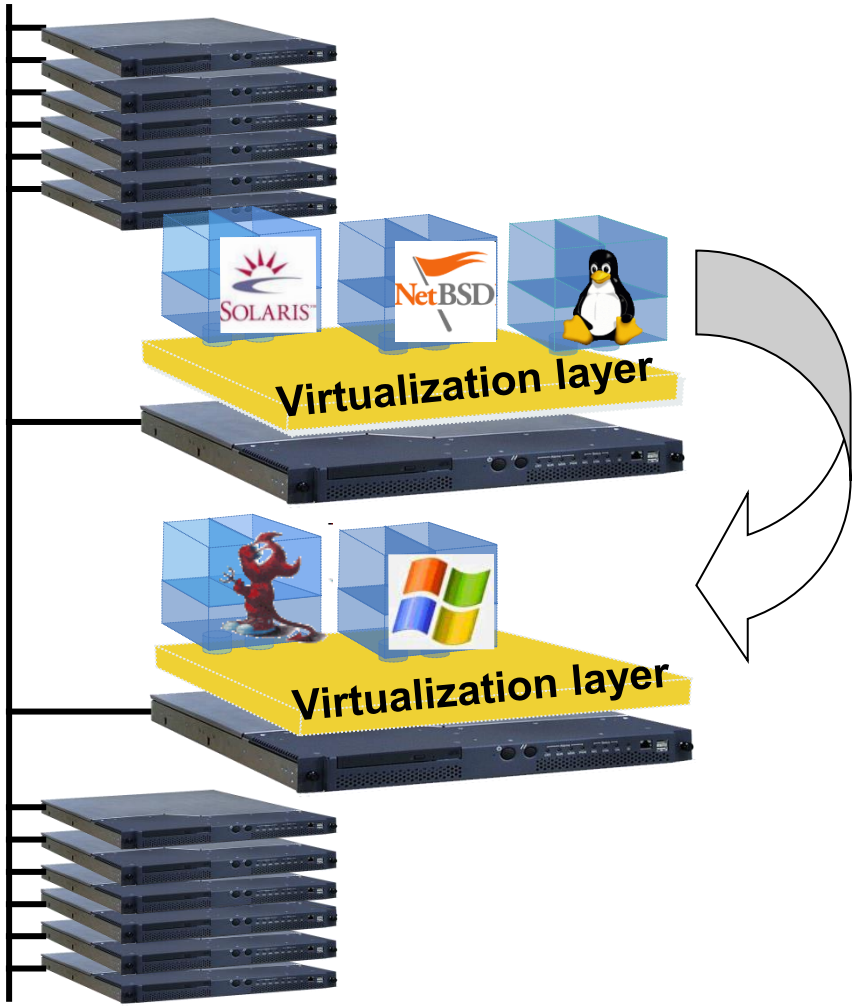
- **Advantages**
  - "Better" CPU performance than full and para-virtualization
  - Improved virtual machine isolation
  - It allows running "unmodified" OSs
- **Disadvantages**
  - Specialized and Expensive Hardware
  - An unmodified OS does not know it is running in a virtualized environment and so, it can't take advantage of any of the virtualization features
    - It can be resolved using partial paravirtualization.



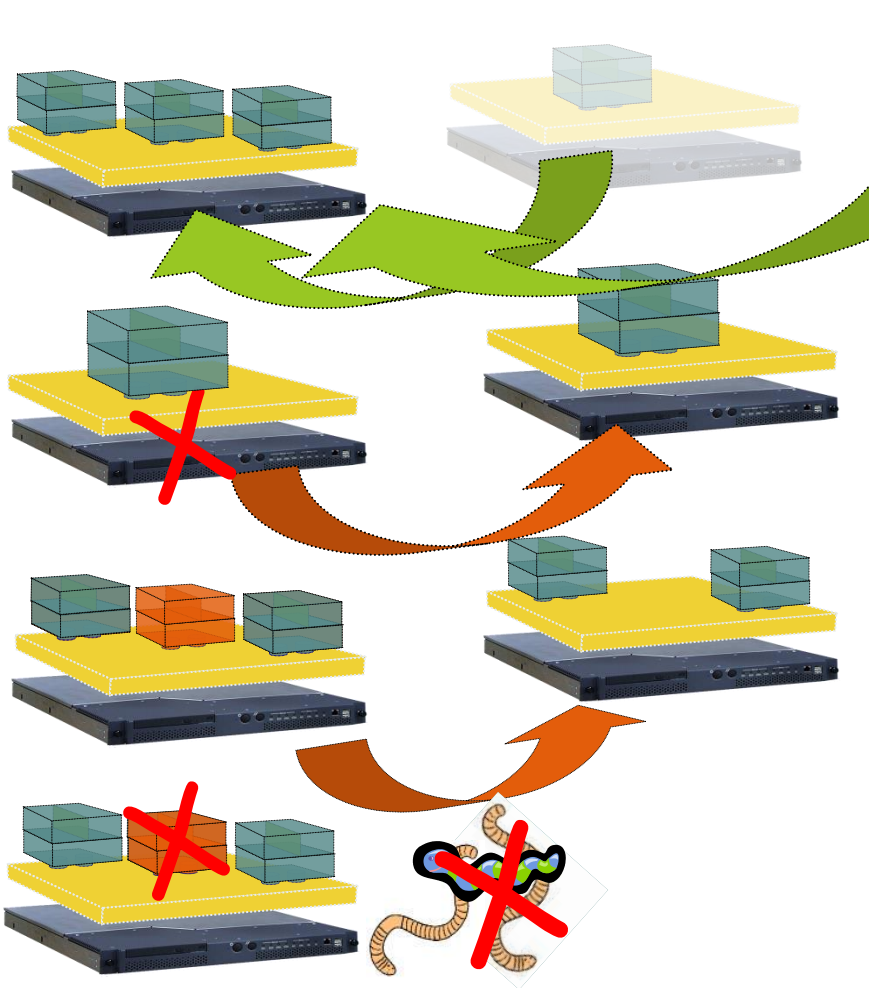

vmware®


all in one

Virtual Machine Migration



Virtual Machine Migration



- Server consolidation
- Reduce downtimes
- Load balancing – SLAs
- Security

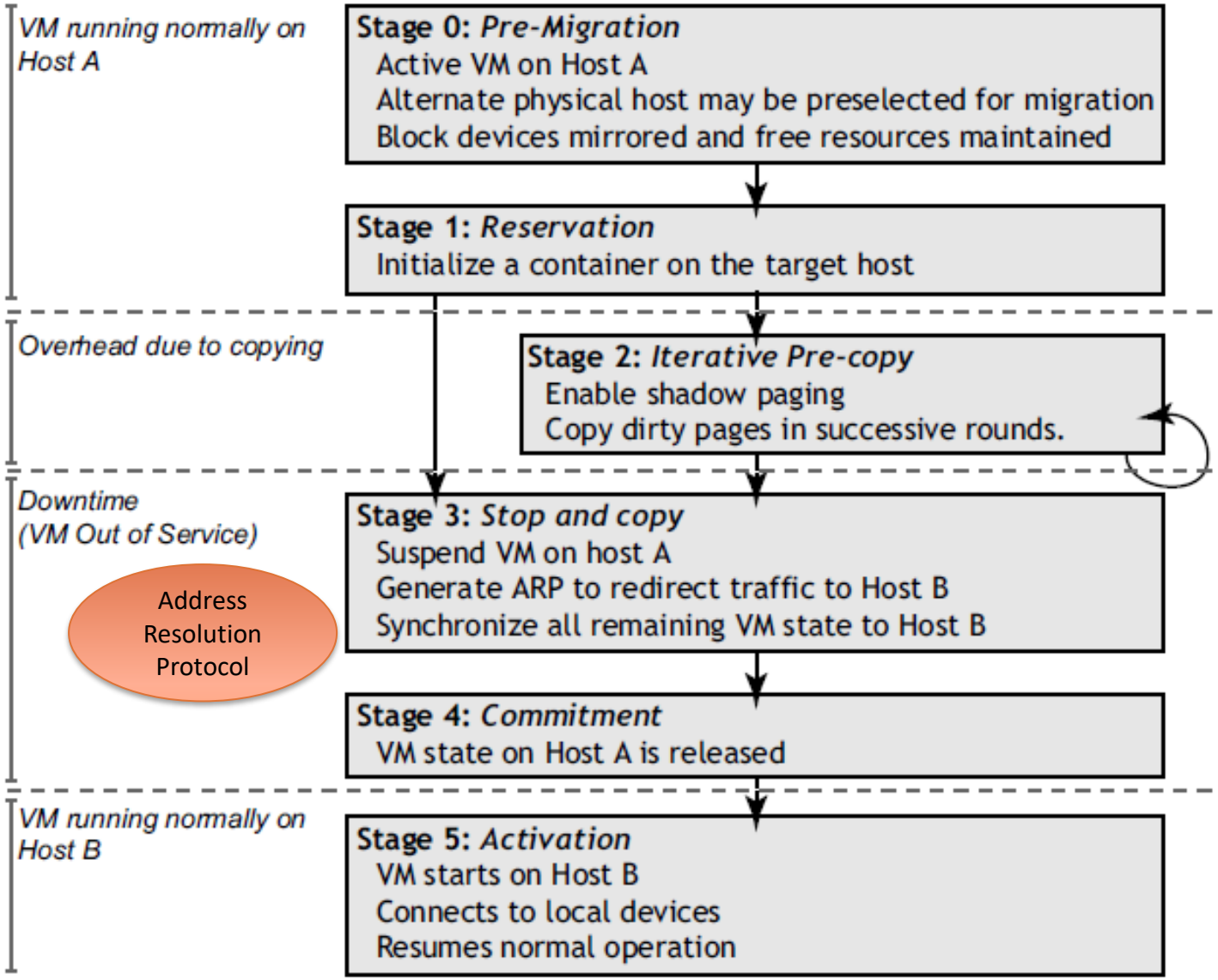
Virtual Machine Migration

- When to migrate?
- Where to move to?
- How much of each resource to allocate?
- How much information needed to make decisions?

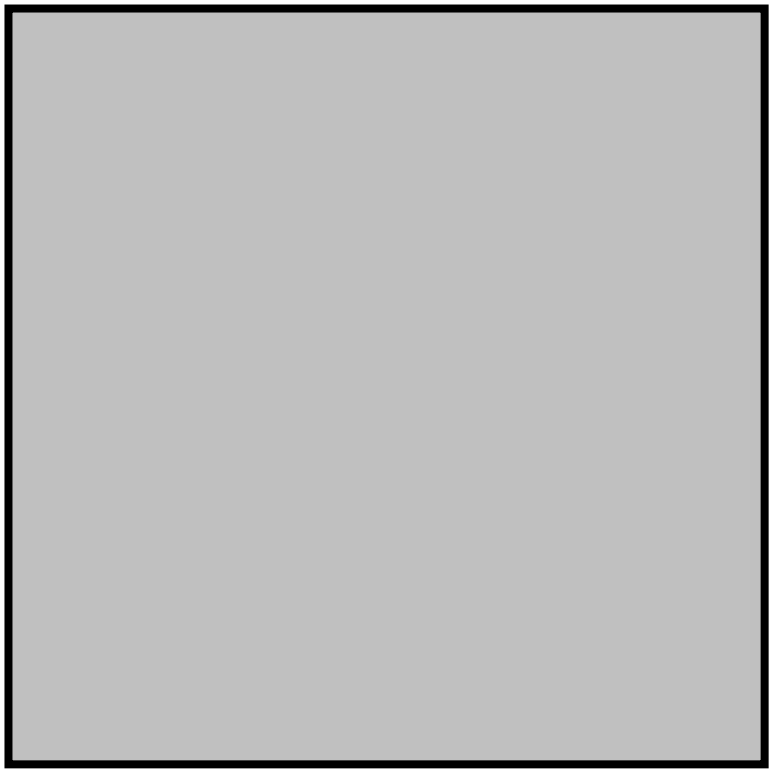
Live Virtual Machine Migration

- Metrics
  - **Downtime**
    - How long the VM is suspended
  - **Total migration time**
    - how long a live migration lasts
  - **Amount of migrated data**
    - How many data is transferred

Live Virtual Machine Migration

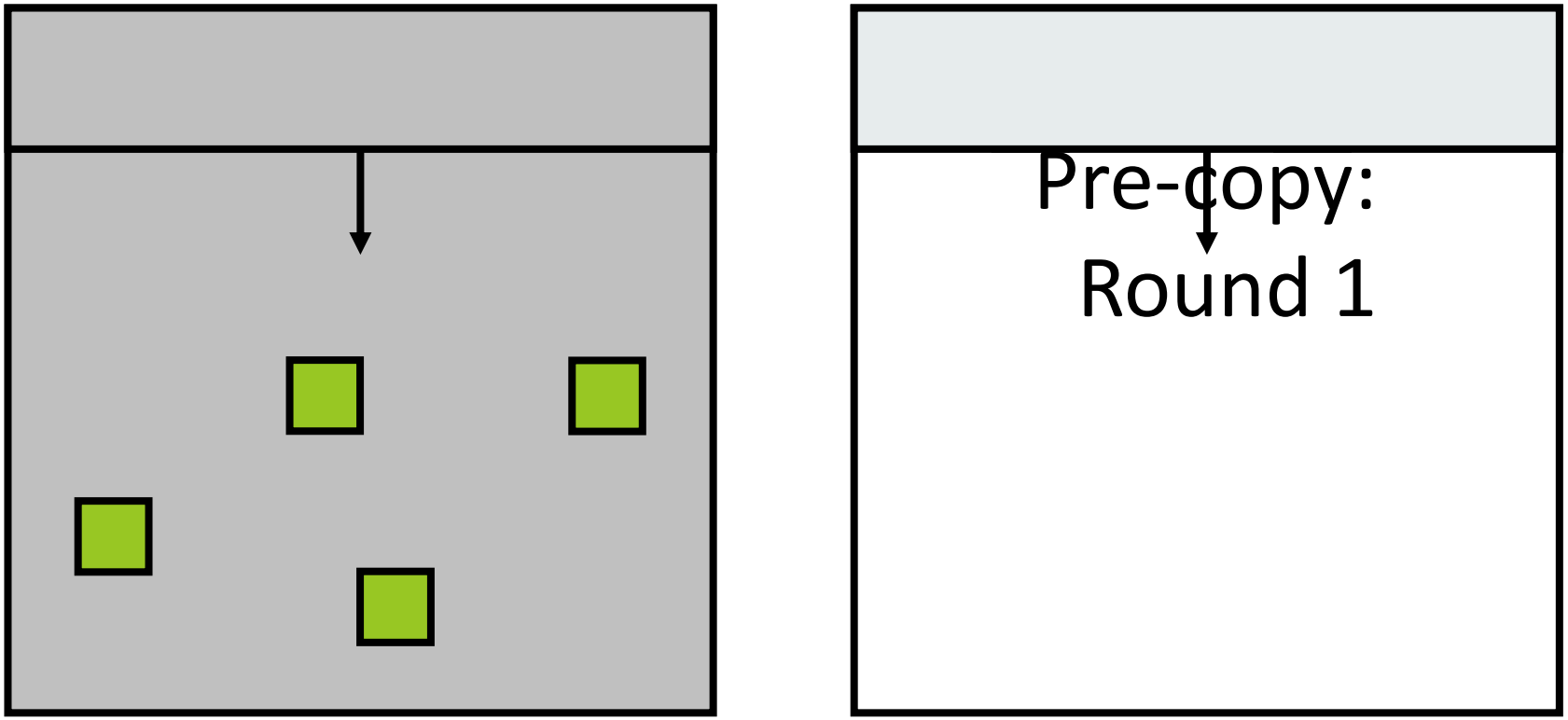


Live Virtual Machine Migration

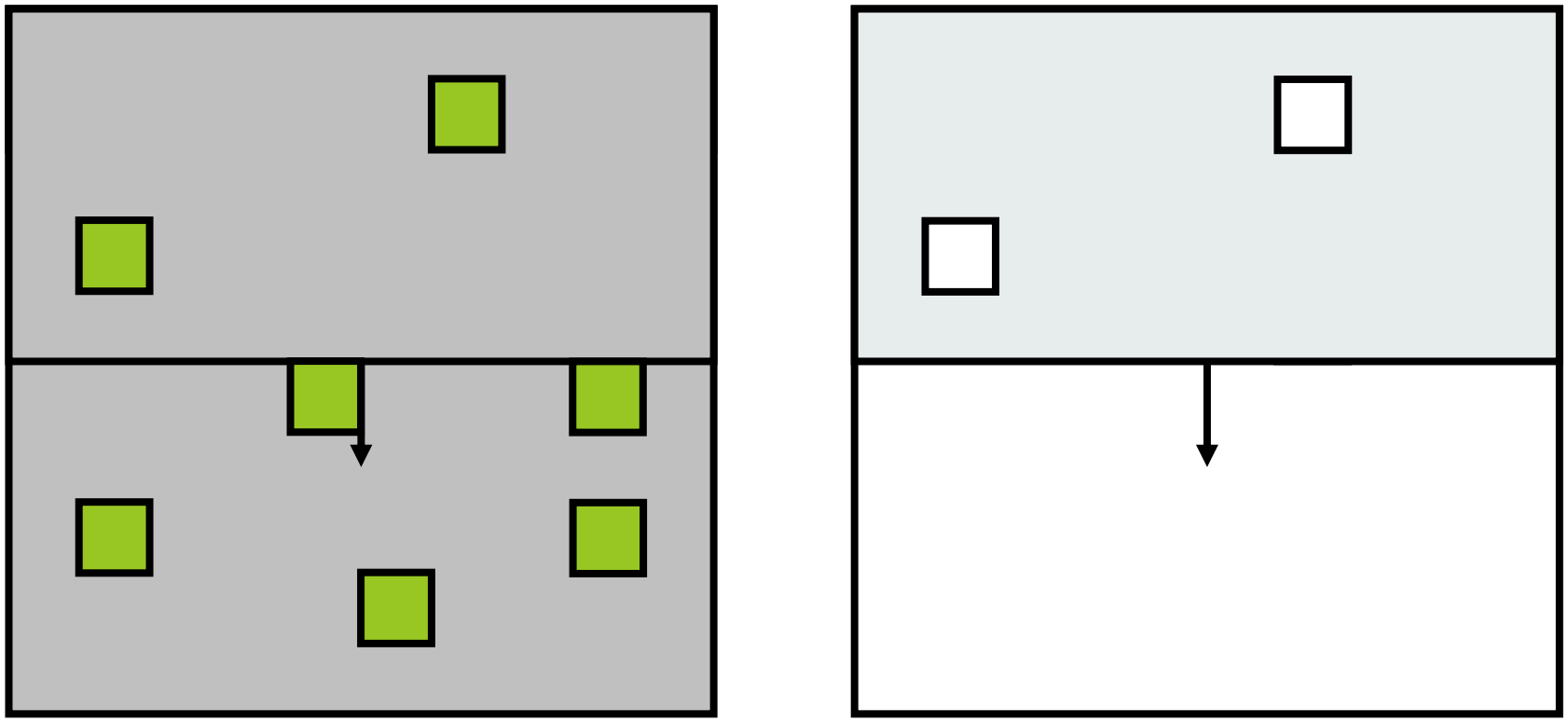


Pre-copy:
Round 1

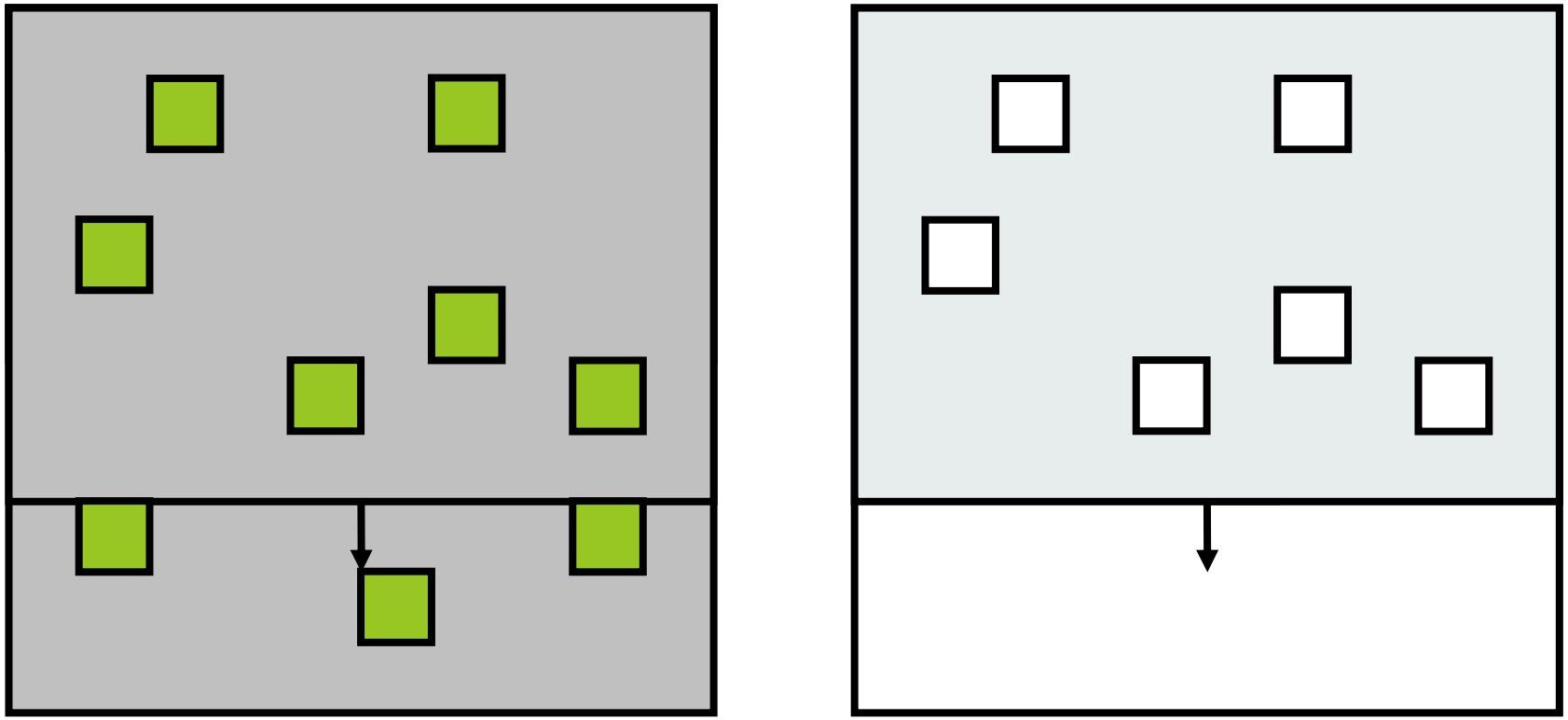
Live Virtual Machine Migration



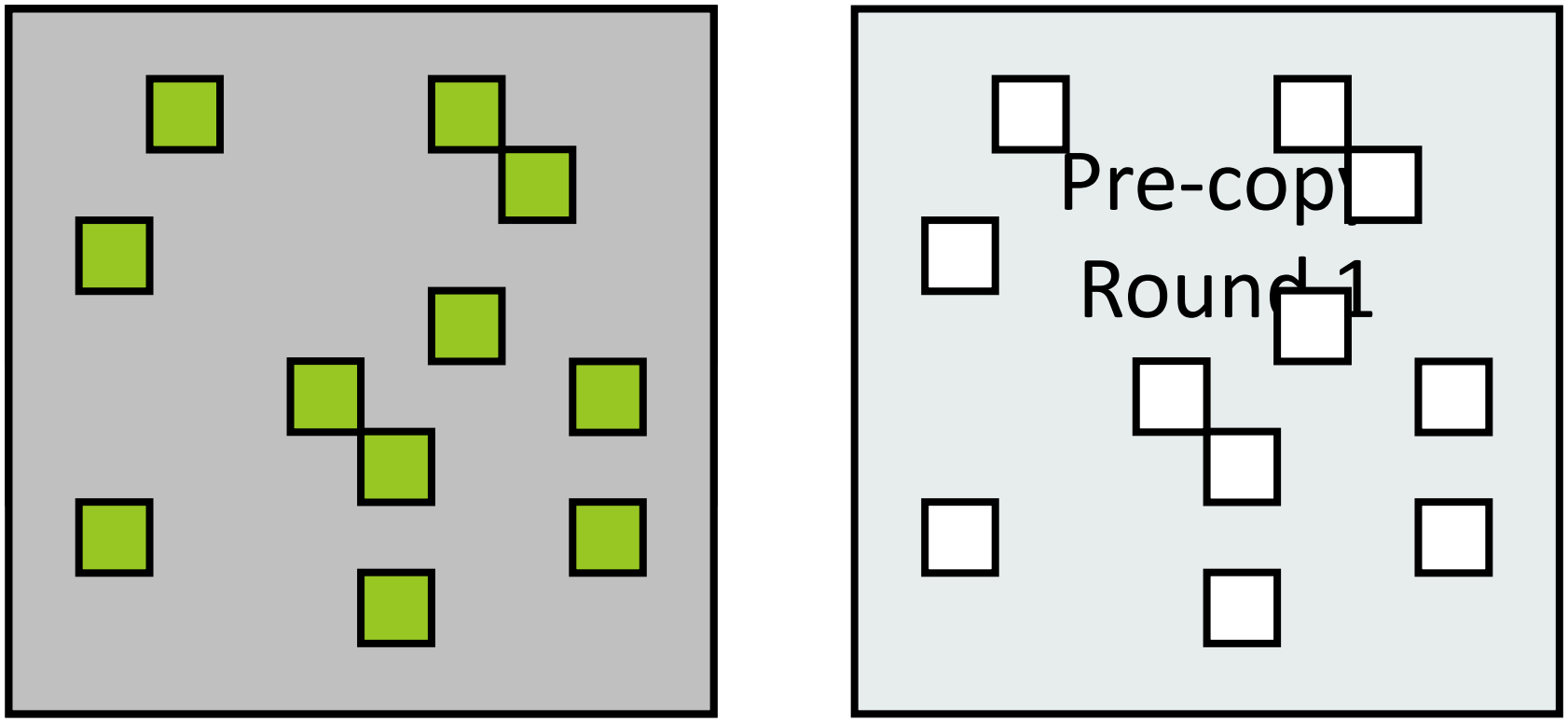
Live Virtual Machine Migration



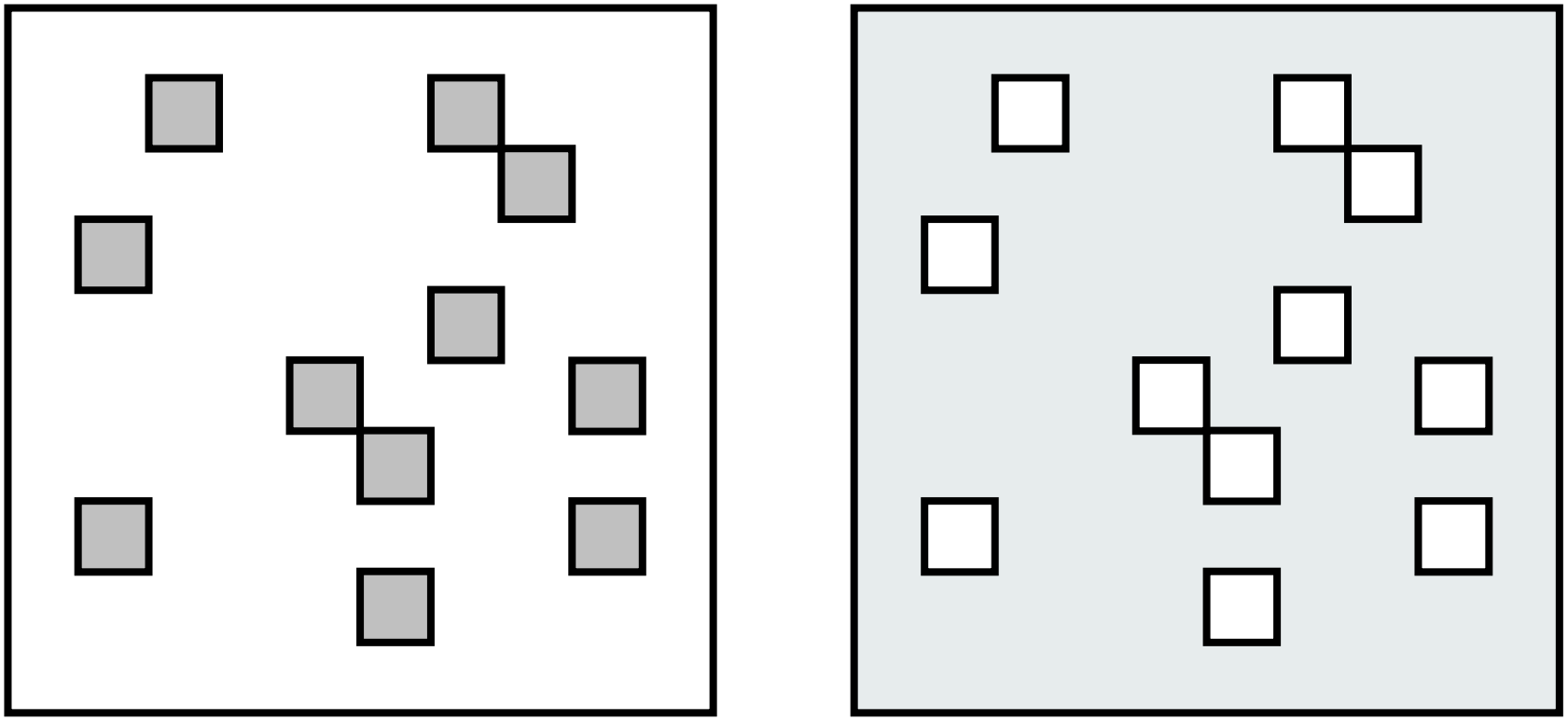
Live Virtual Machine Migration



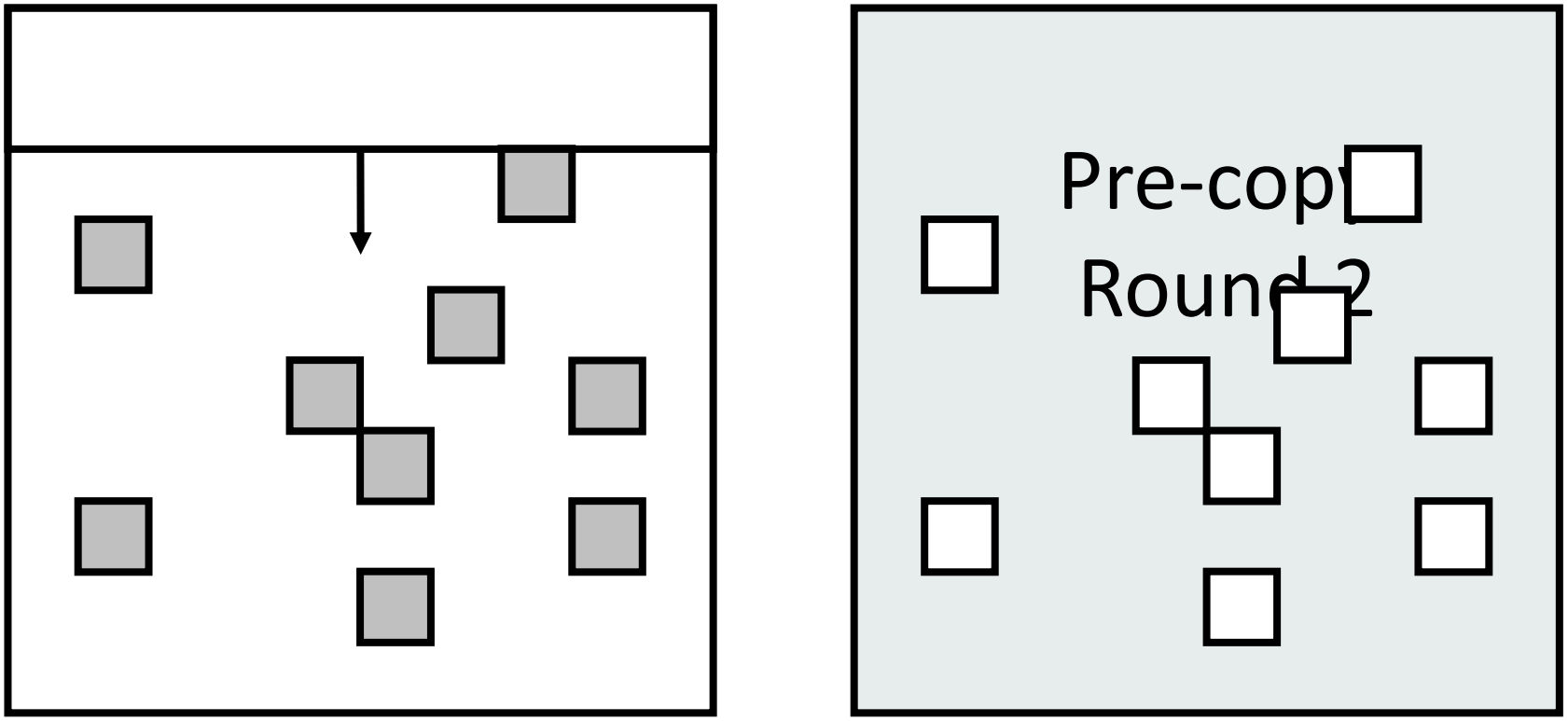
Live Virtual Machine Migration



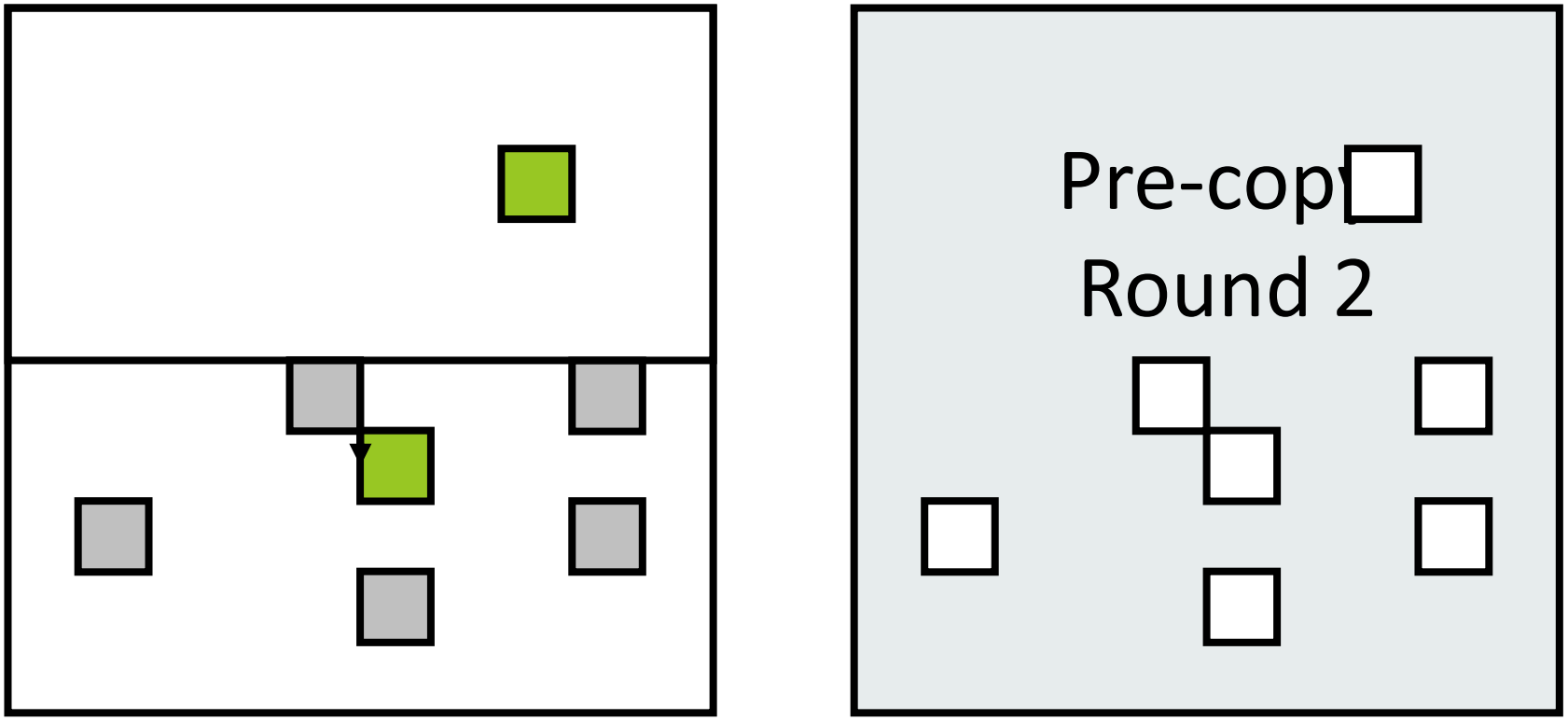
Live Virtual Machine Migration



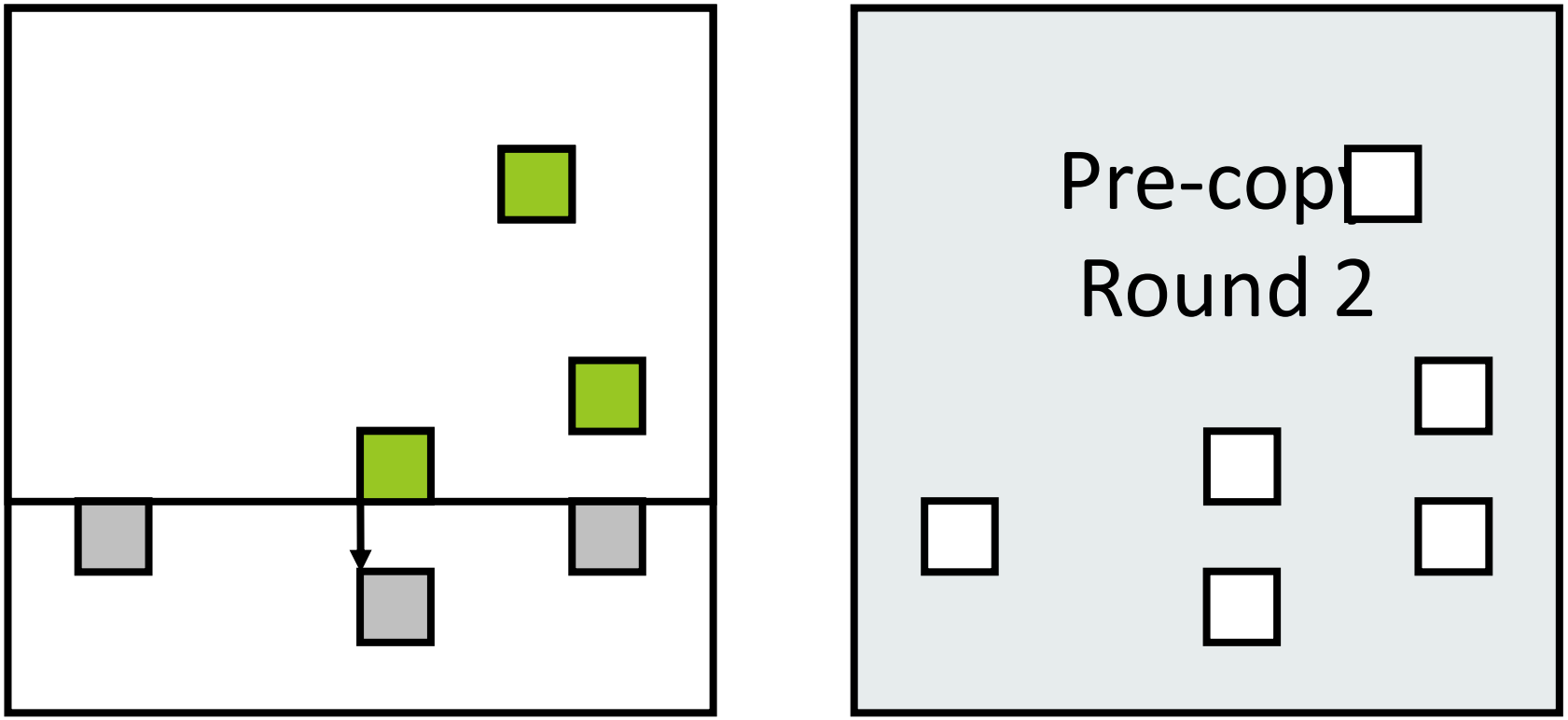
Live Virtual Machine Migration



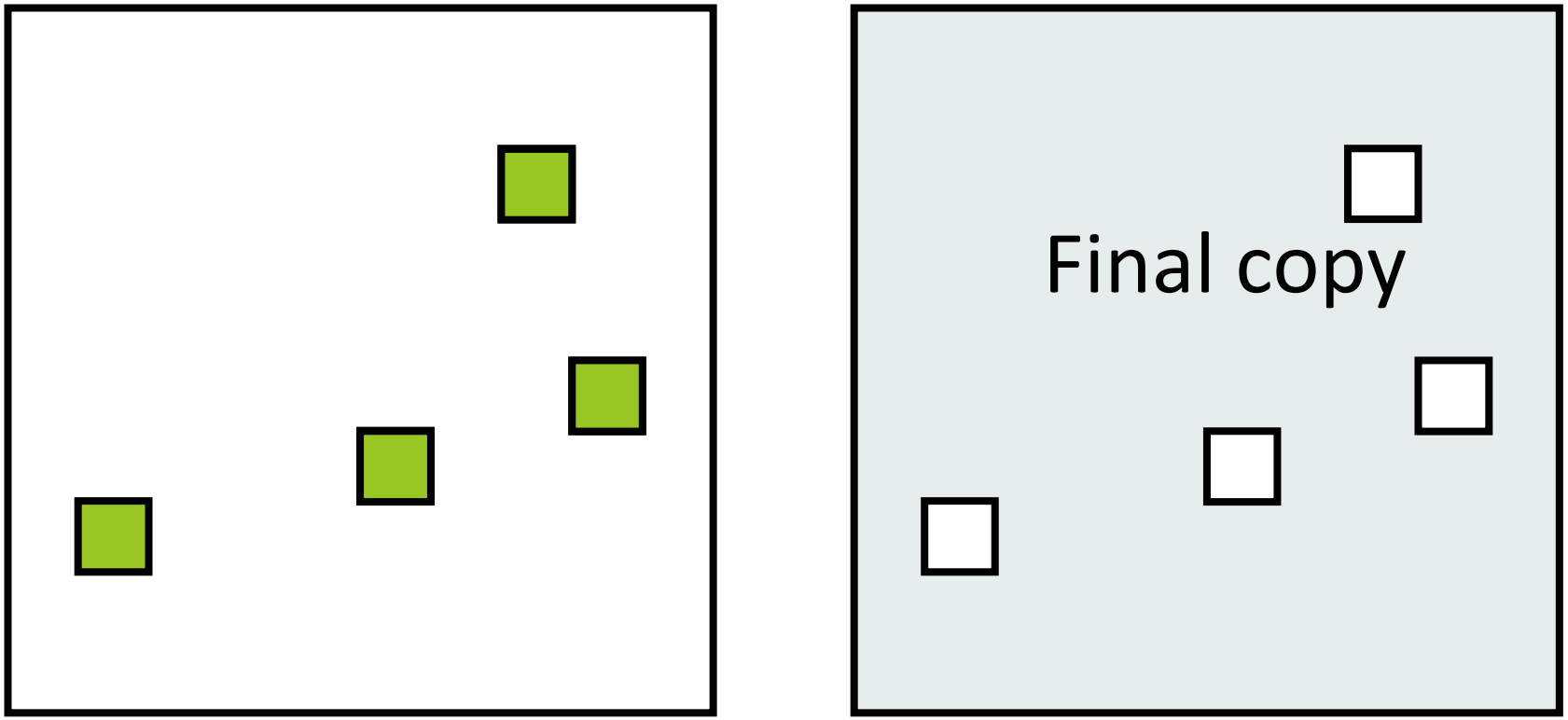
Live Virtual Machine Migration



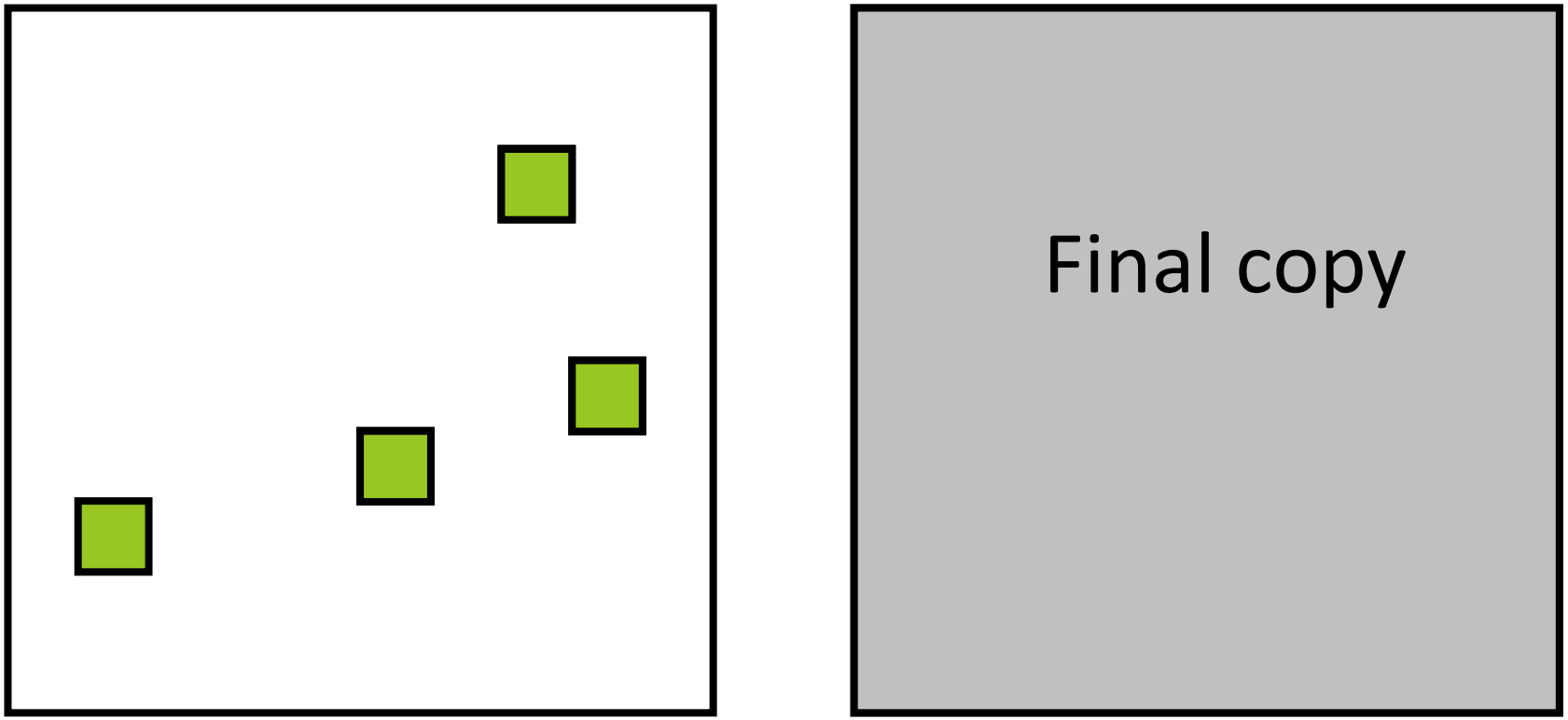
Live Virtual Machine Migration



Live Virtual Machine Migration



Live Virtual Machine Migration



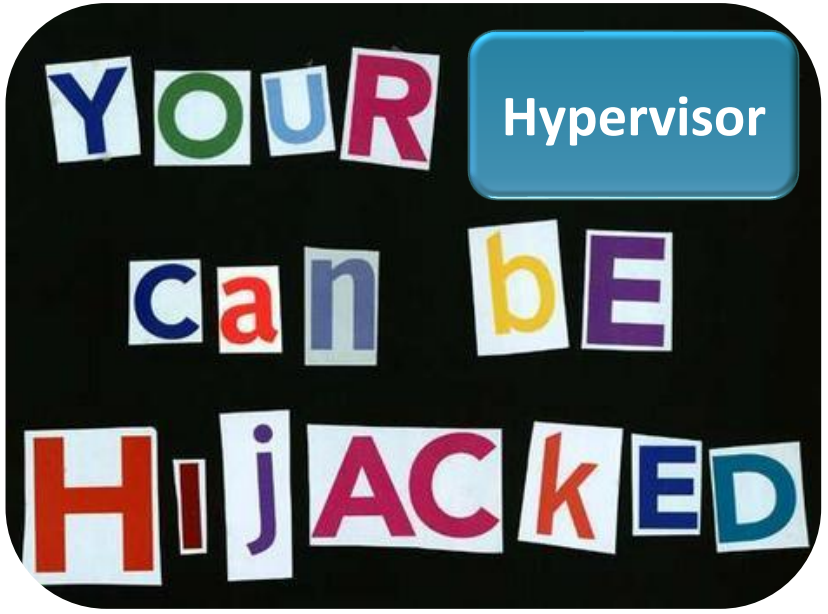
VMware: (Anti-)Affinity rules

- VM-to-VM
- VM-to-Physical server

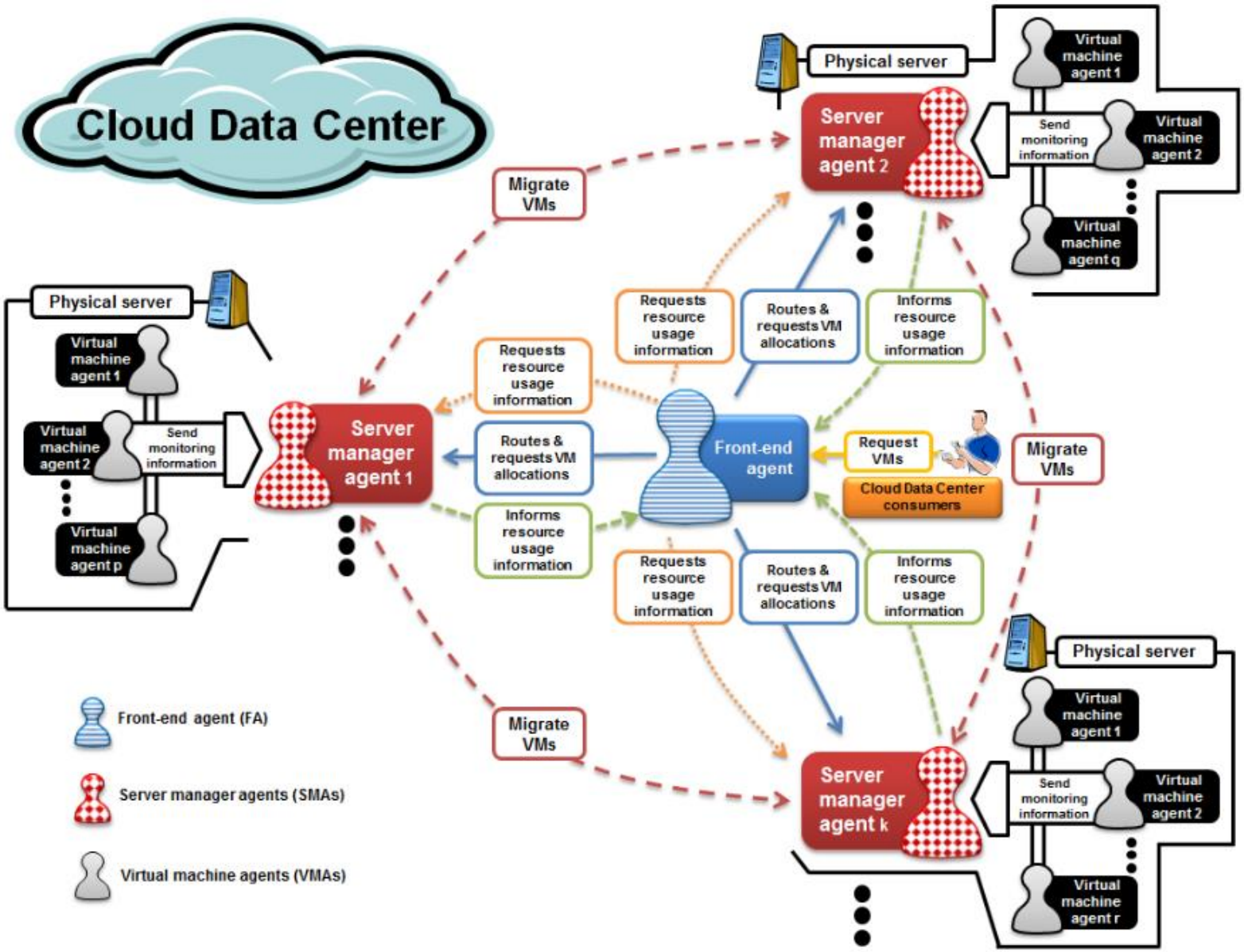
Affinity Solution

Some security issues

Multitenancy

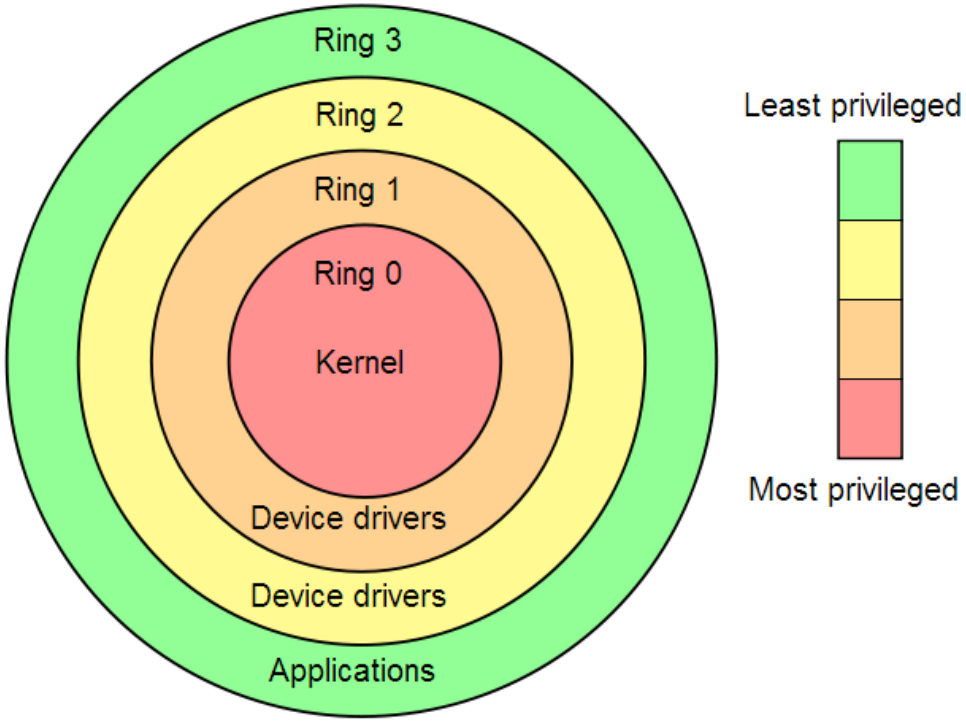


Agent-based load balancing using live migration of VMs



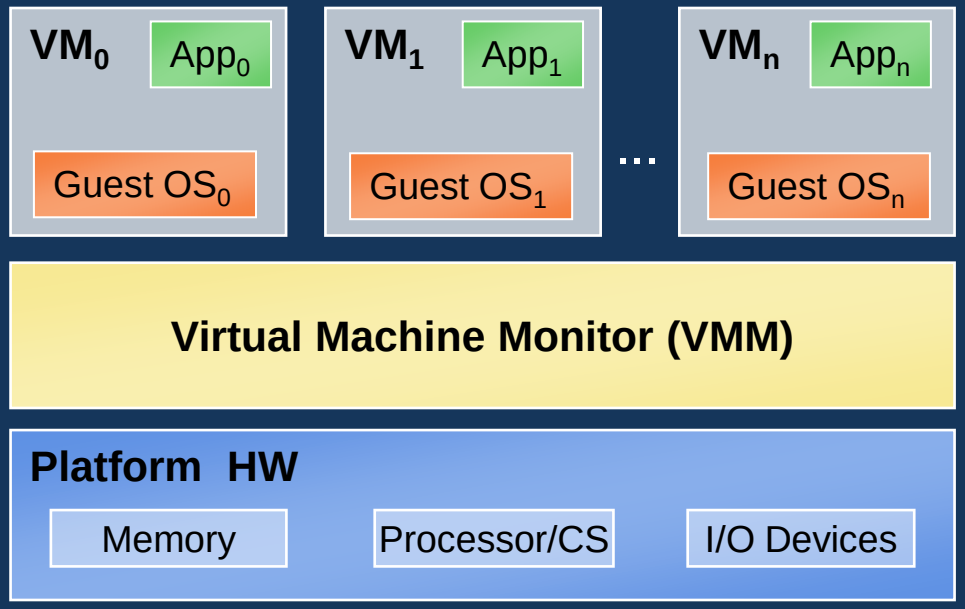
Hardware-assisted virtualization

- The hypervisor/VMM runs at Ring -1
  - super-privileged mode



Understanding Intel® Virtualization Technology (VT)

Virtual Machine Monitors (VMMs)

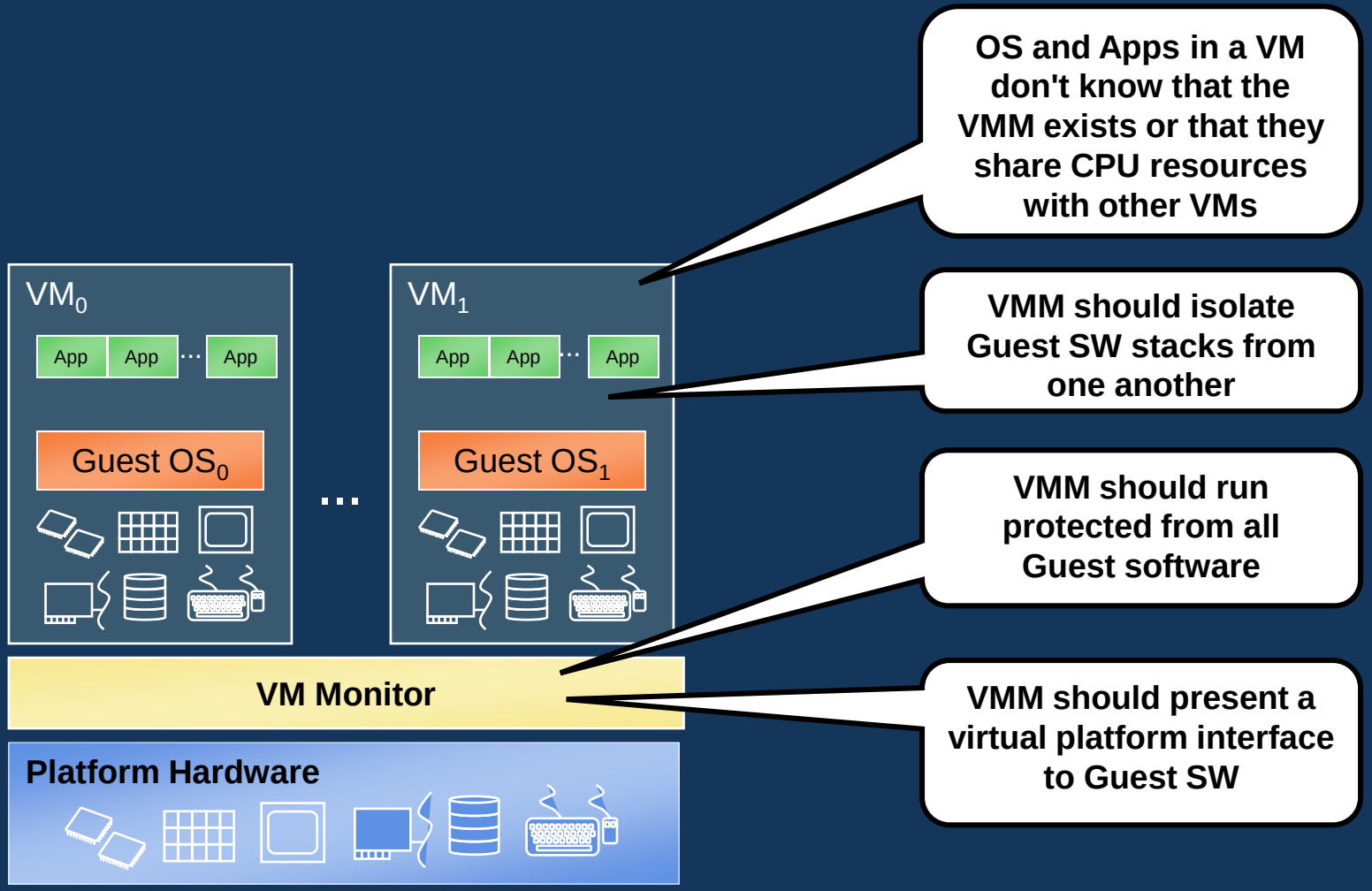


- VMM is a layer of system software
  - Allows Apps to run without modifications

What is Intel® Virtualization Technology ?

- VT is a set of hardware enhancements to Intel server and client platforms
- VT is designed to simplify virtualization software
- VT-x and VT-i are the first in the VT series of Intel processor and chipset innovations

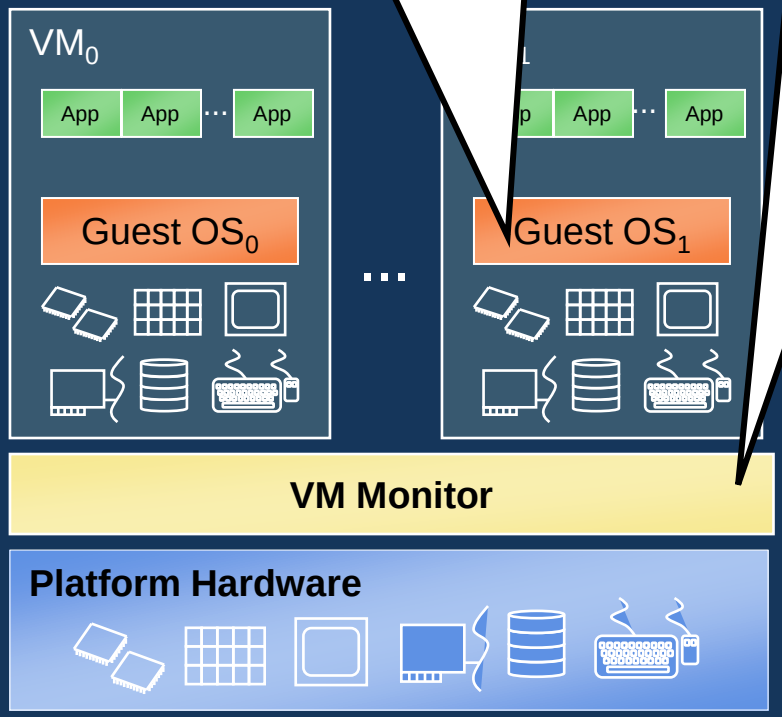
Challenges of Running a VMM



SW Solution: Guest Ring Deprivileging

Run Guest OS above Ring-0 and have privileged instructions generate faults...

Run VMM in Ring-0 as a collection of fault handlers



Top IA (Intel Architecture) Virtualization Holes :

- Ring Aliasing
- Excessive Faulting
- CPU state context switching
- Addr Space Compression

…

Virtualization of current IA CPUs requires complex software workarounds

Challenges

Ring Aliasing

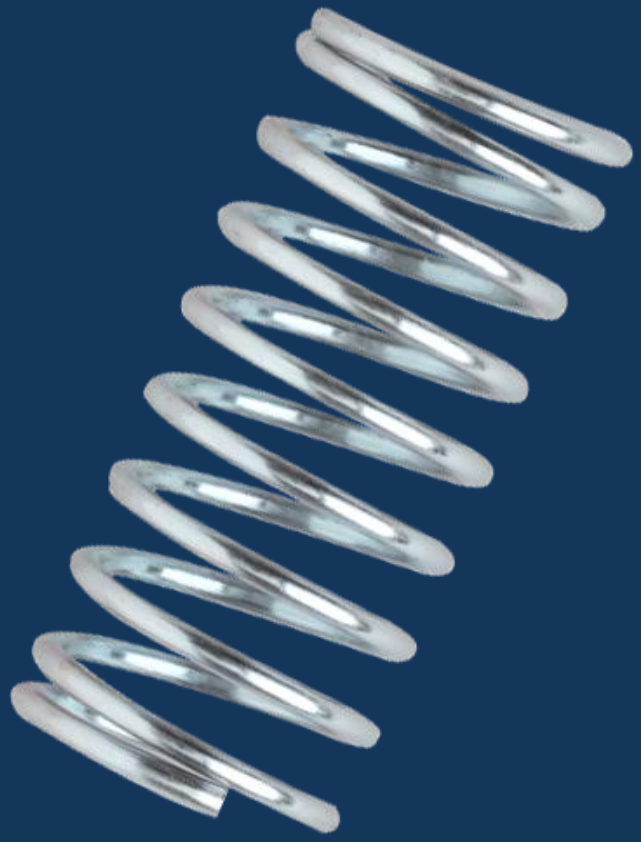
- The problem that arise when software is run at a privilege level other than the privilege level for which it was written
- An existing OS may be written to run with ring 0
- VMM must run with ring 0

Challenges

Address-Space Compression

- VMM must use some of the guest's virtual-address space to manage transition between guest OS and VMM

- VMM's address spaces must be protected
  - Guest could detect that it is running in a VM



Challenges

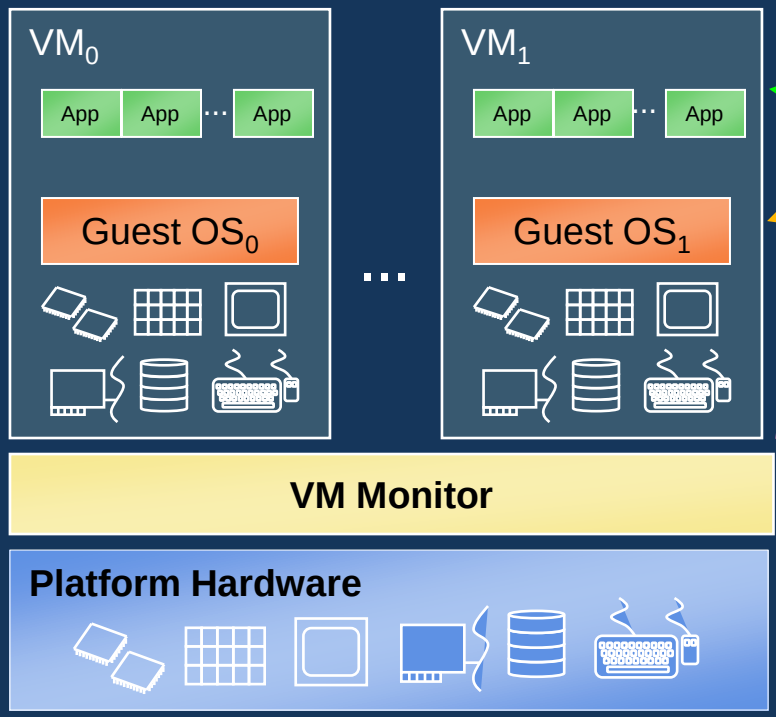
Faulting Access to Privileged State

- In most cases, accessing privilege states result in faults
- Performance is compromised by excessive faults

Ring Compression

- Guest OS must run at ring 3
- Guest OS runs at the same privilege level as applications

Intel® Virtualization Technology



Guest SW runs deprivileged in a new operating mode:

- Apps run deprivileged in ring 3
- OS runs deprivileged in ring 0
- VMM runs in new mode with full privilege

VMM preempts execution of Guest SW via new HW-based transition mechanism

By design, VT eliminates virtualization holes and the need for complex software workarounds

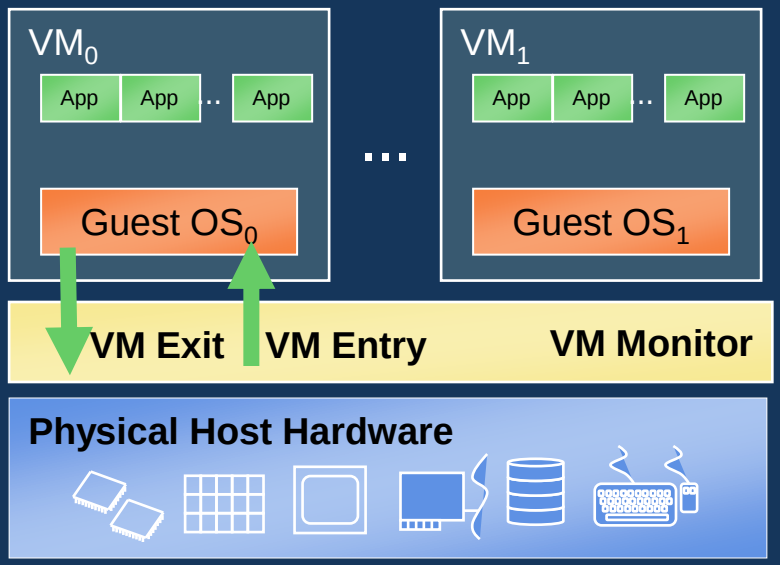
Operating Modes

- VMX root operation:
  - Fully privileged, intended for VM monitor
- VMX non-root operation:
  - Not fully privileged, intended for guest software
  - Reduces Guest SW privilege w/o relying on rings
  - Solution to Ring Aliasing and Ring Compression

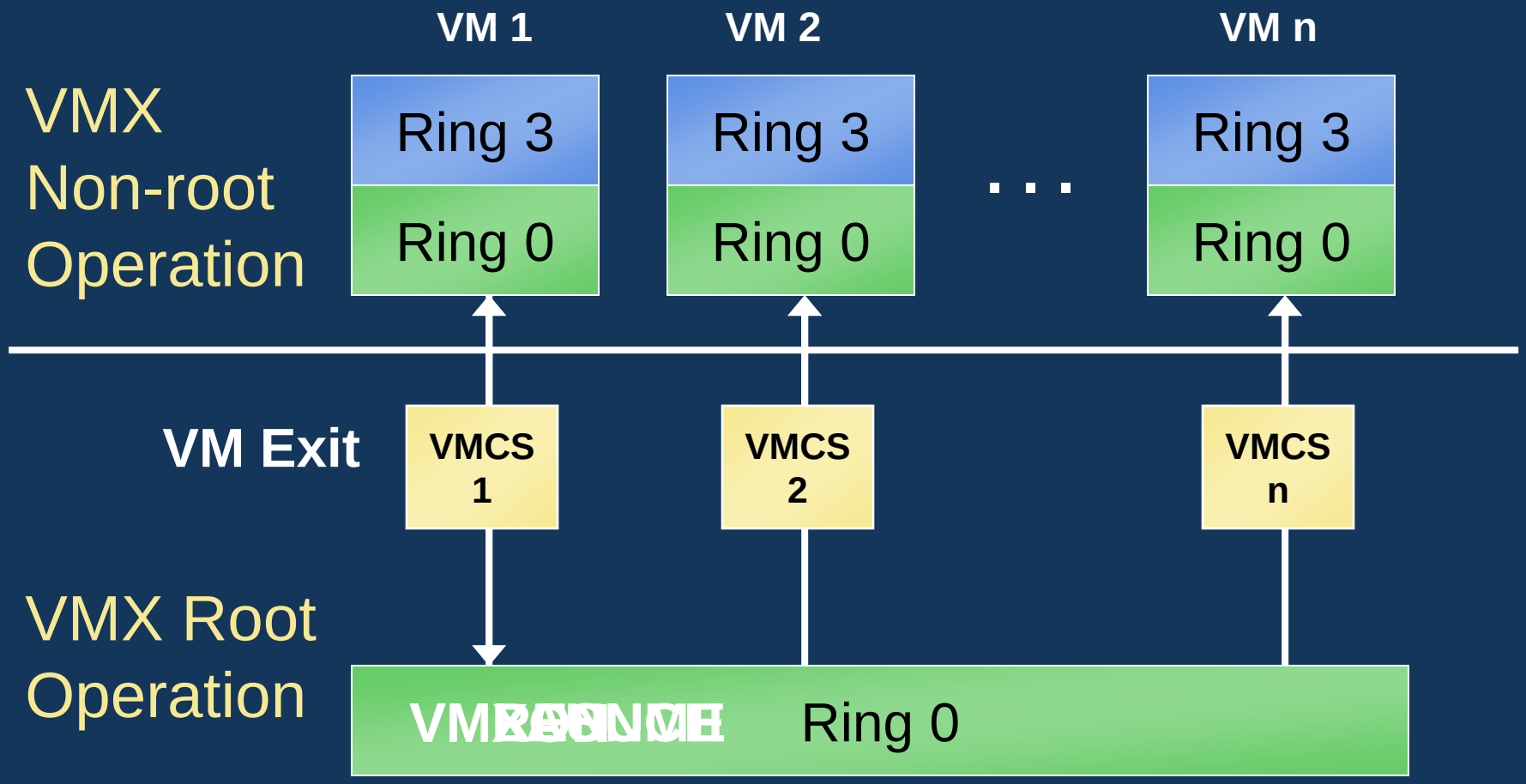
VM Entry and VM Exit

Virtual Machine Control Structure

- VM Entry
  - Transition from VMM to Guest
  - Enters VMX non-root operation
  - Loads Guest state and Exit criteria from VMCS
  - VMLAUNCH instruction used on initial entry
    VMRESUME instruction used on subsequent entries
- VM Exit
  - VMEXIT instruction used on transition from Guest to VMM
  - Enters VMX root operation
  - Saves Guest state in VMCS
  - Loads VMM state from VMCS



VT-x Operations



Virtual Machine Control Structure (VMCS)

- VMCSs are Control Structures in Memory
  - Only one VMCS active per virtual processor at any given time
- VMCS Payload (body data):
  - VM execution, VM exit, and VM entry controls
  - Guest and host state
  - VM-exit information fields
- VMCS Format not defined and may vary

Some Causes of VMEXIT

- Paging state exits allow page-table control
  - Selectively exit on page faults
- Selective exception and I/O exiting reduce unnecessary exits
- Controls provided for asynchronous events
- Detection of guest inactivity to support VM scheduling
  - HLT, MWAIT, PAUSE