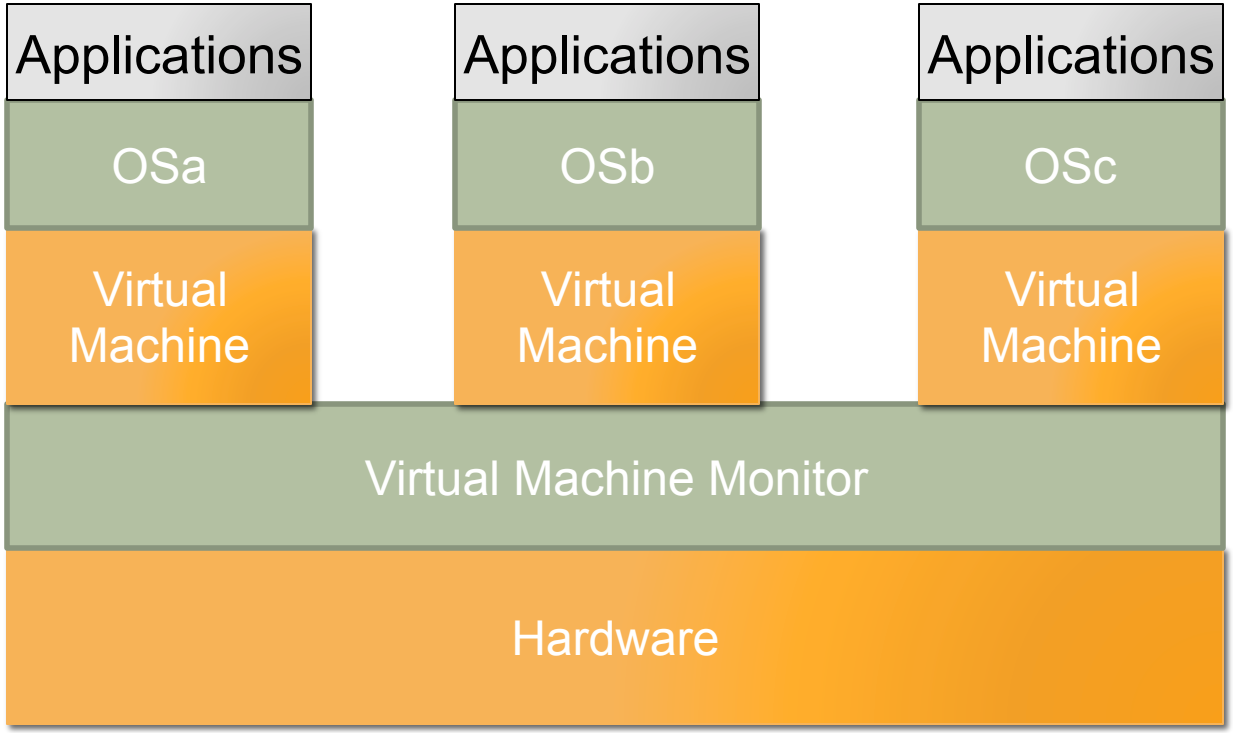
VIRTUAL MACHINES

Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

It's 1964 …

- IBM wants a multiuser time-sharing system
  - CMS
    - single-user time-sharing system for IBM 360
  - CP67
    - virtual machine monitor (VMM)
    - supports multiple virtual IBM 360s
  - Put the two together …
    - a (working) multiuser time-sharing system

Virtual Machines



Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

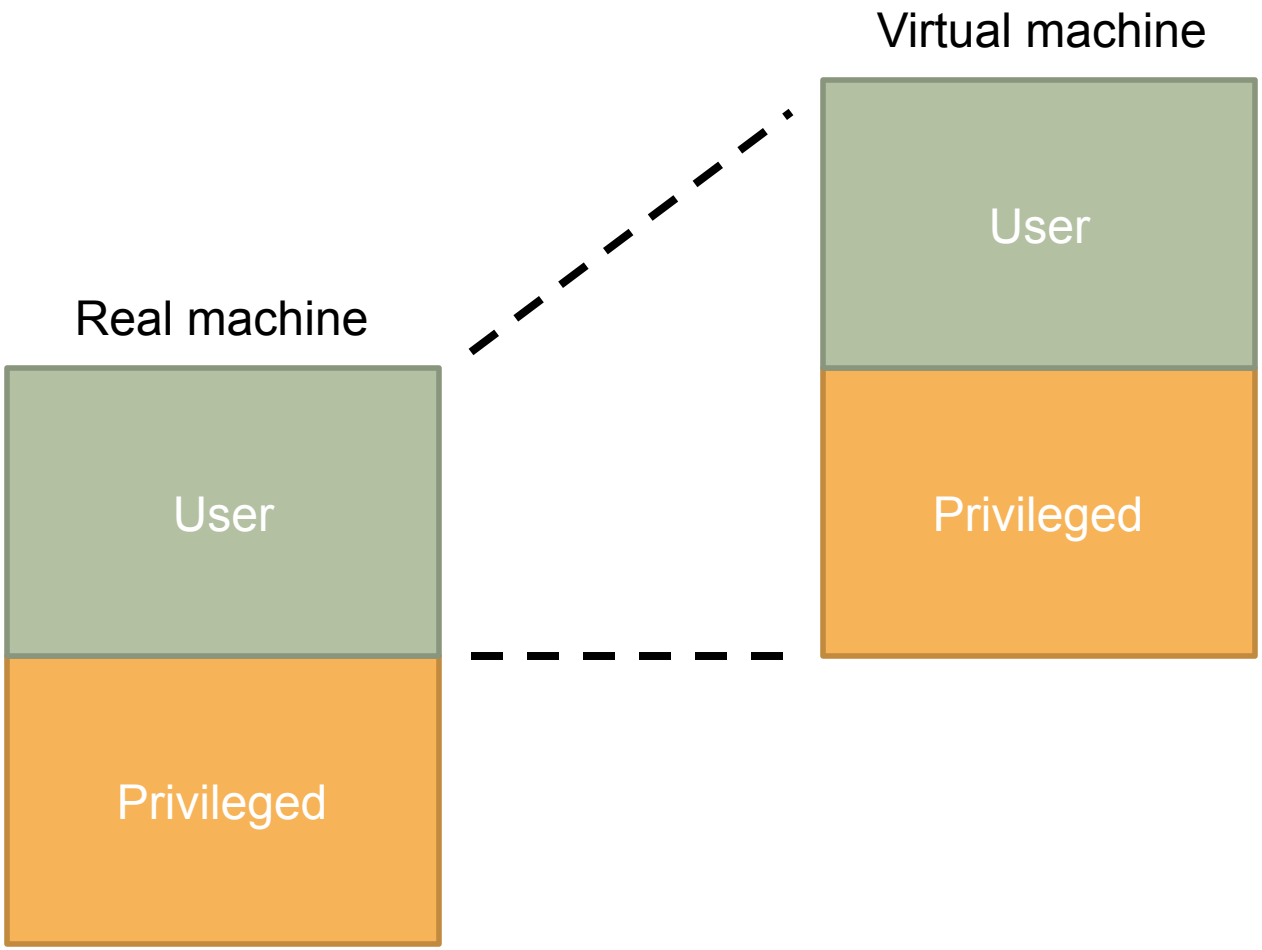
Why?

- Structuring technique for a multi-user system
- OS debugging and testing
- Multiple OSes on one machine
- Adapt to hardware changes in software
- Server consolidation and service isolation

Issues

- Multiplex the processor among virtual machines
  - Instructions are executed directly on the processor
- Make each virtual machine behave just like a “real one”
  - Handle interrupts generated both by real and virtual devices
  - Should the guest OS run in privileged or user mode?

How?



Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

Sensitive and Privileged Instructions

- Popek and Goldberg 1974

- Sensitive instructions
  - Control-sensitive instructions
    - affect the allocation of resources available to the virtual machine
    - change processor mode without causing a trap
  - Behavior-sensitive instructions
    - effect of execution depends upon location in real memory or on processor mode

- Privileged instructions
  - Cause a fault in user mode
  - Work fine in privileged mode

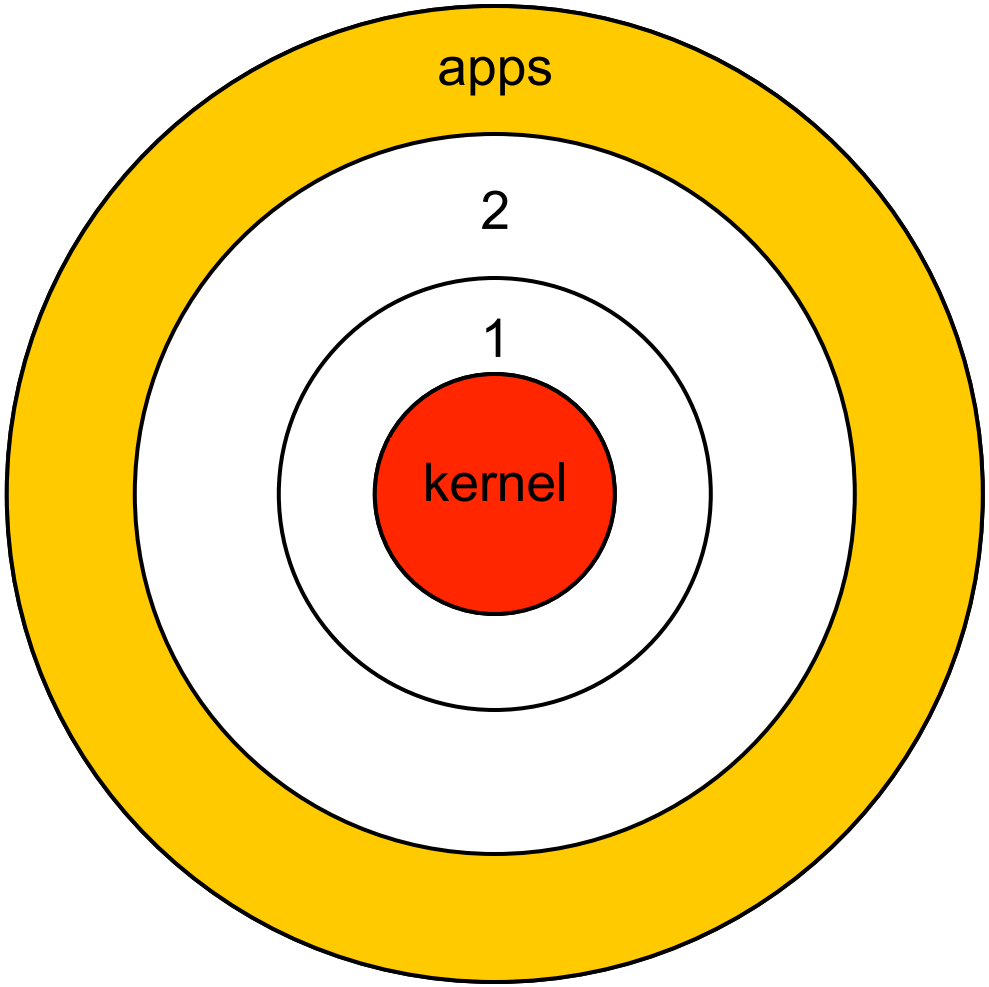
Sensitive and Privileged Instructions

- Popek and Goldberg 1974
- Set of sensitive instructions **is subset of** set of privileged instructions **then** then a virtual machine monitor can be constructed for it.
- If not, it is possible to build a virtualization infrastructure, but it is more complex

Intel x86

- Four execution modes
  - rings 0 through 3
  - not all sensitive instructions are privileged instructions
- Memory is protectable: segment system + virtual memory
- Special register points to interrupt table
- I/O done via memory-mapped registers
- Virtual memory is standard

Rings



A Sensitive x86 Instruction

- popf
  - pops word off stack, setting processor flags according to word's content
    - Ring 0: sets all - including interrupt-disable flag
    - Other rings: just some of them - ignores interrupt-disable flag

Solution 1 – Binary Rewriting

- Rewrite kernel binaries of guest OSes
- Privilege-mode code run via binary translator
  - Replaces sensitive instructions with hypercalls
  - Done dynamically
  - Translated code is cached
    - usually translated just once
- VMWare Workstation (32 bit guests), IBM System/370, VirtualBox, …

Solution 2 – Hardware-assisted Virtualization

- Fix the hardware so it's virtualizable

- Intel Vanderpool technology: VT-x
  - Two modes of operation orthogonal to the four rings:
    - root mode (in which the VMM runs)
    - non-root mode
  - Certain events in non-root mode cause VM-exit to root mode
    - essentially a hypercall
    - code in root mode specifies which events cause VM-exits
  - Non-VMM OSes must not be written to use root mode!

- VMWare workstation (64-bit guests) , Xen 3.x, KVM, …

CPU Virtualization

- Scheduling problem
- Issues:
  - Detect when the VMs processor is idle
    - Some OSs execute idle processes (to check for work)

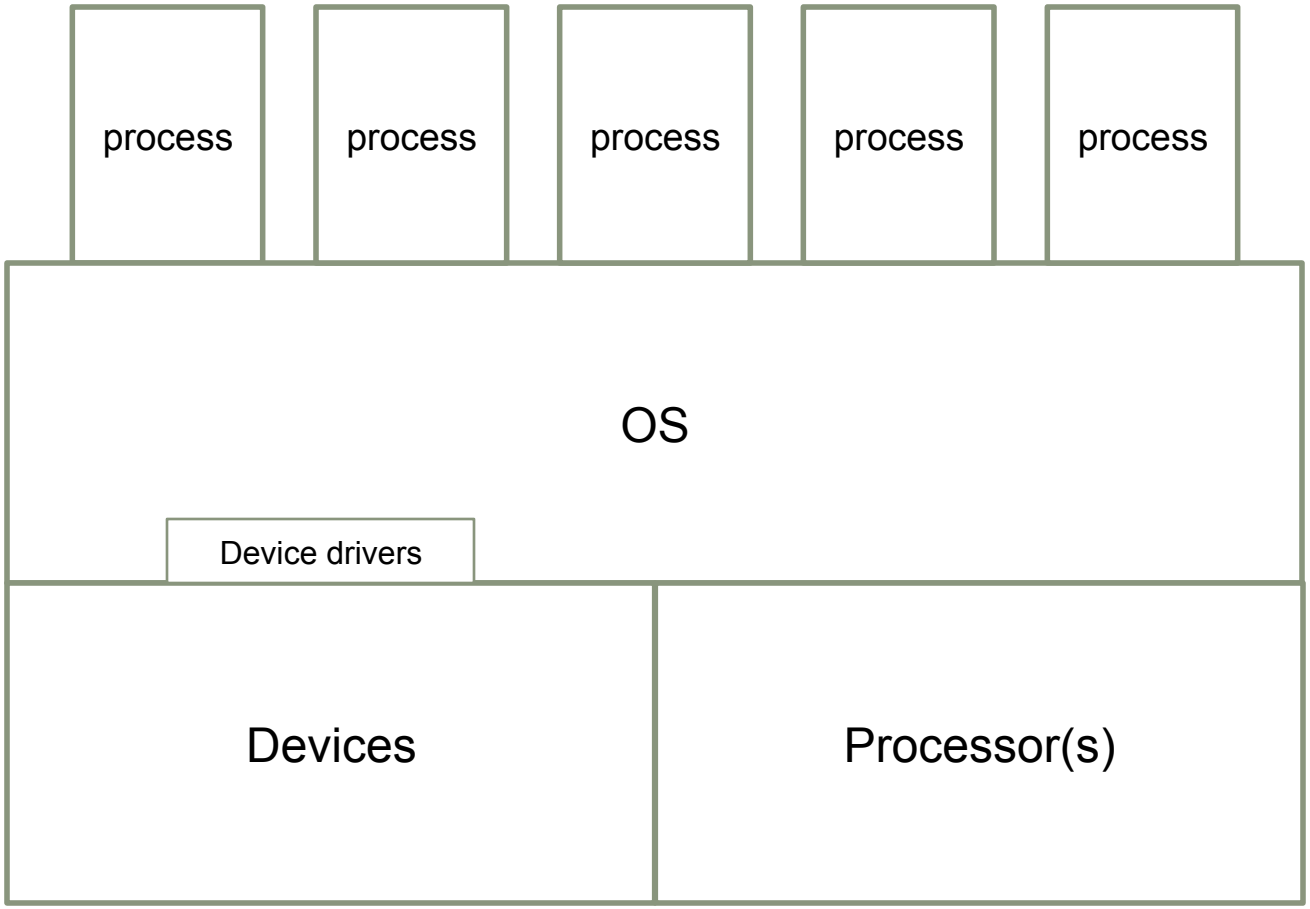
    →Time slicing

  - Double multiplexing
    - Virtualize timer → virtual time
    - What about time-outs?
    - Cannot provide both virtual and real time transparently

I/O Virtualization

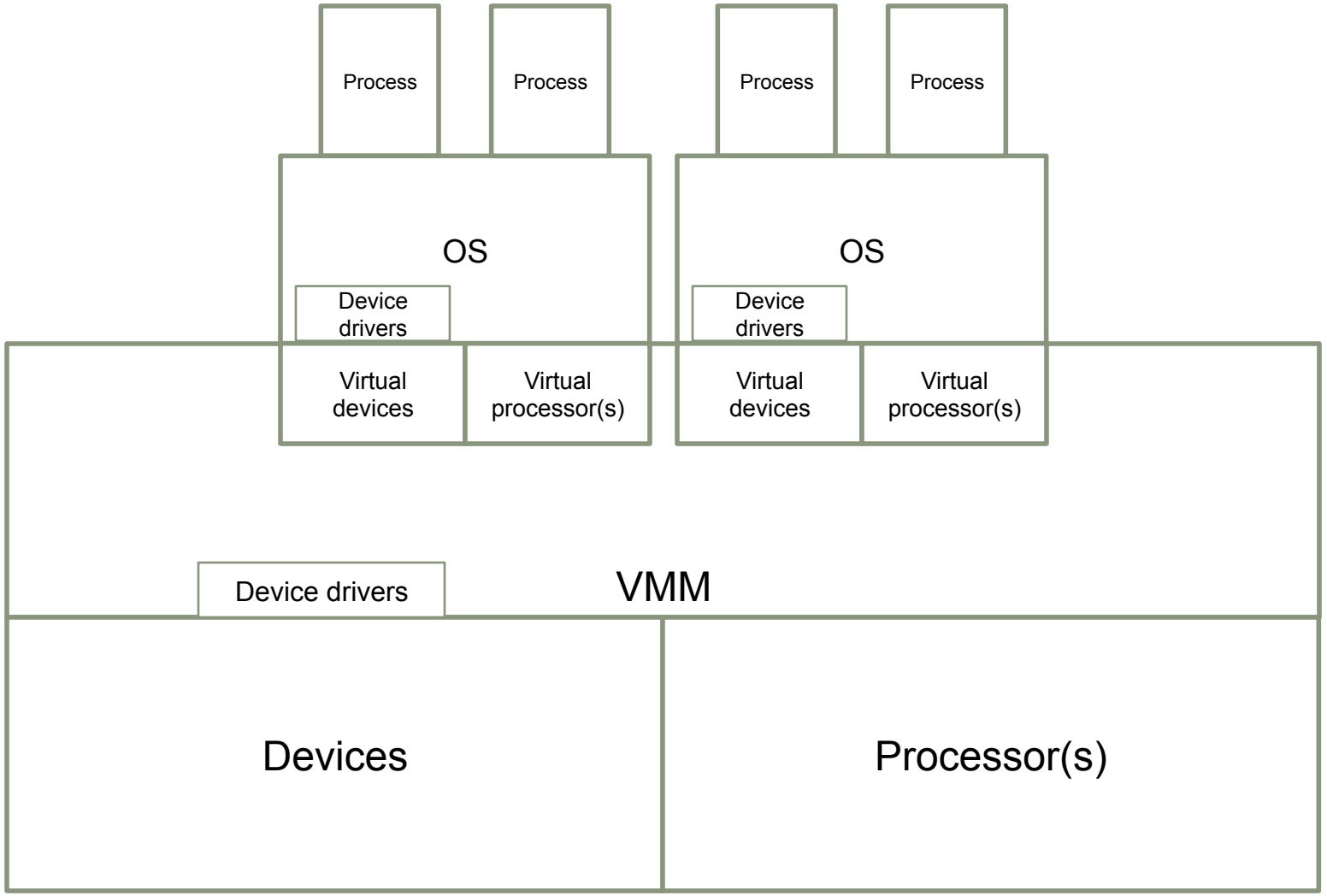
- Lots and lots and lots of device drivers
- Must VMM handle all of them?

Real-Machine OS Structure



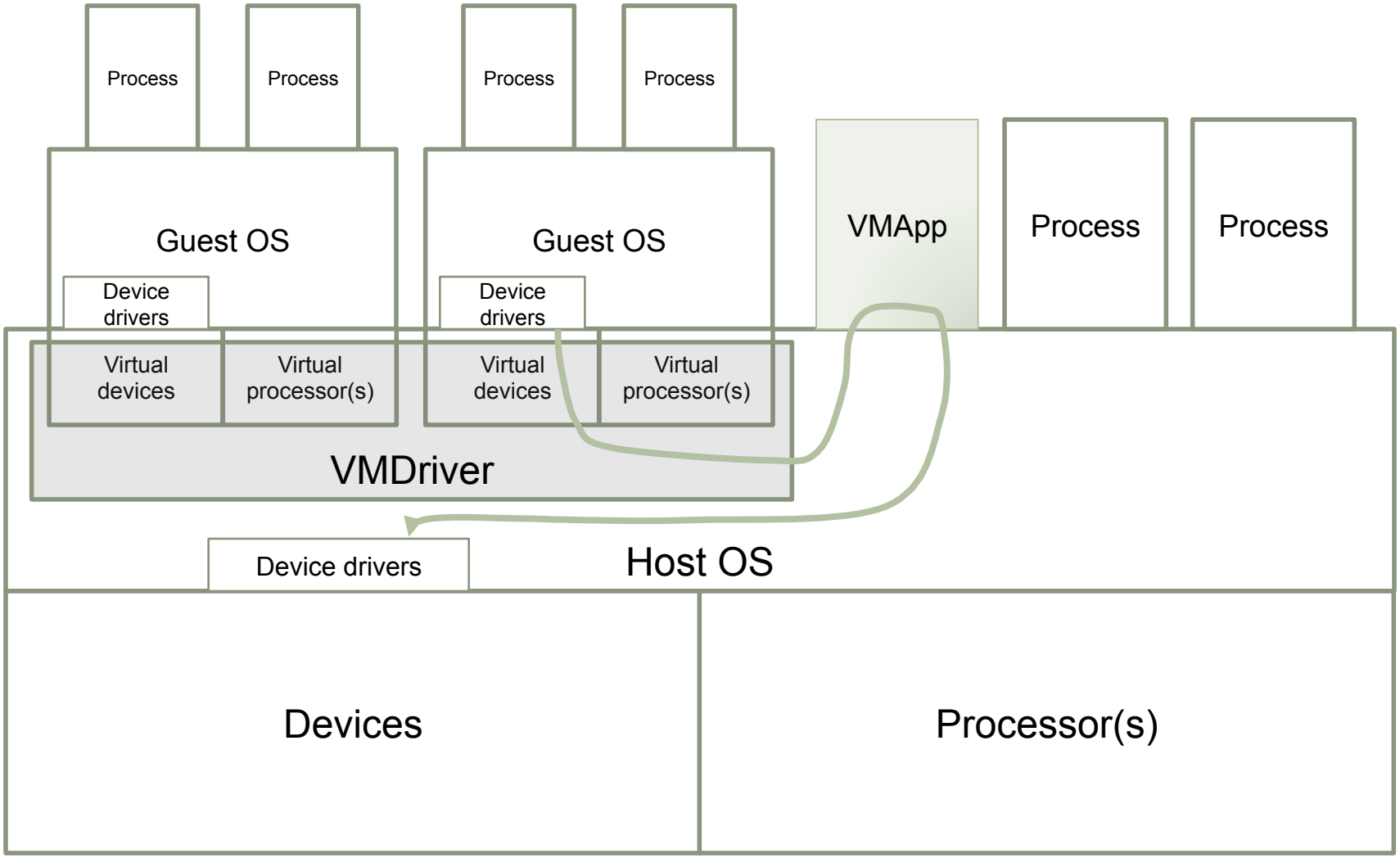
Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

On a Virtual Machine …



Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

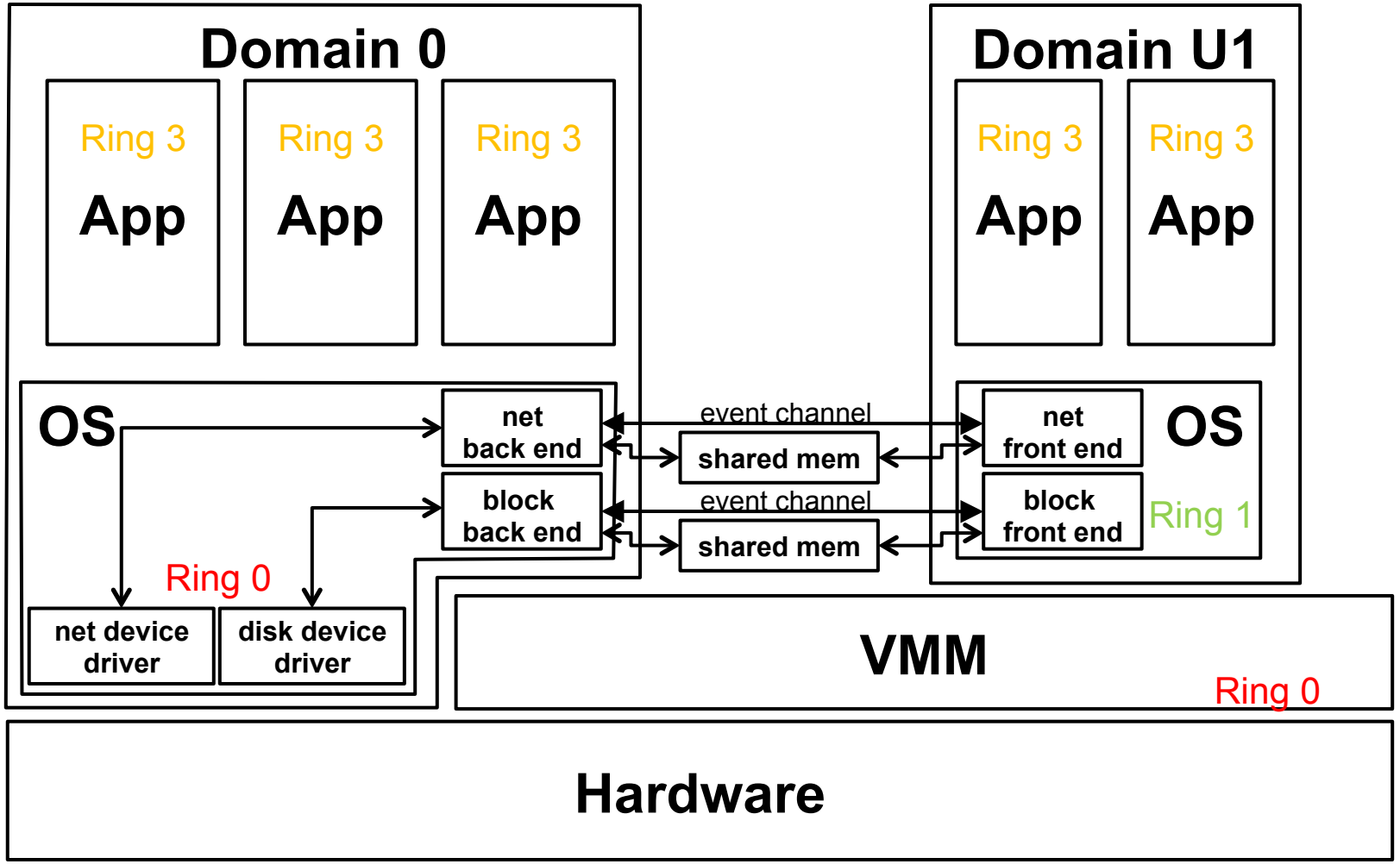
VMware Workstation



Solution 3 - Paravirtualization

- Virtual machine differs from real machine
  - Provides more convenient interfaces for virtualization
  - *Hypervisor* interface between virtual and real machines
  - Guest OS source code is modified
- Sensitive instructions replaced with hypervisor calls
  - traps to VMM
- Virtual machine provides higher-level device interface
  - guest machine has no device drivers

Xen

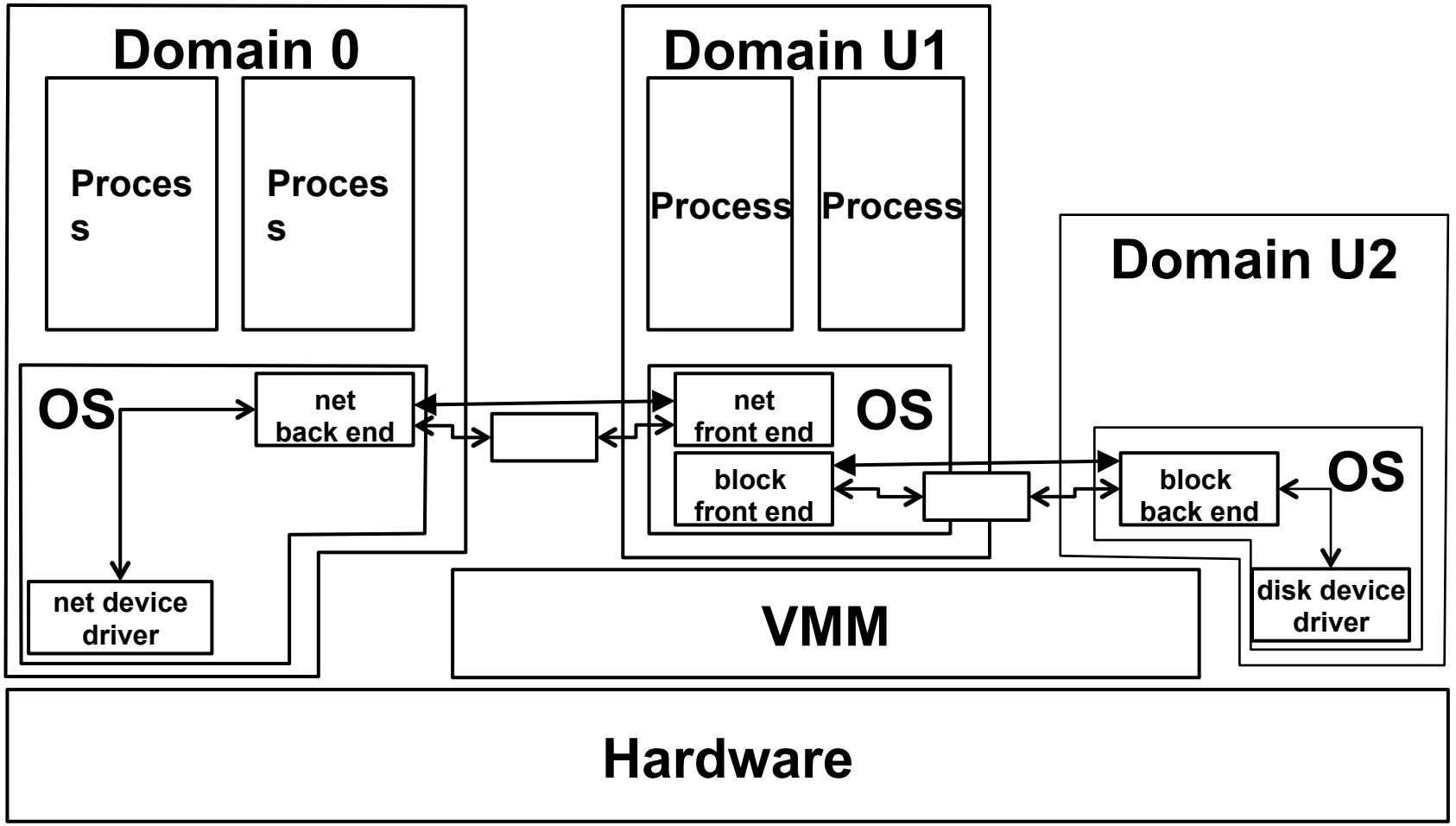


Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”

Additional Applications

- Sandboxing
  - Isolate web servers
  - Isolate device drivers
- Migration
  - VM not tied to particular hardware
  - Easy to move from one (real) platform to another

Xen with Isolated Driver



Slides adapted from Thomas Doeppner “Operating Systems In Depth: Design and Programming”