Cloud Computing
#6 - Virtualization

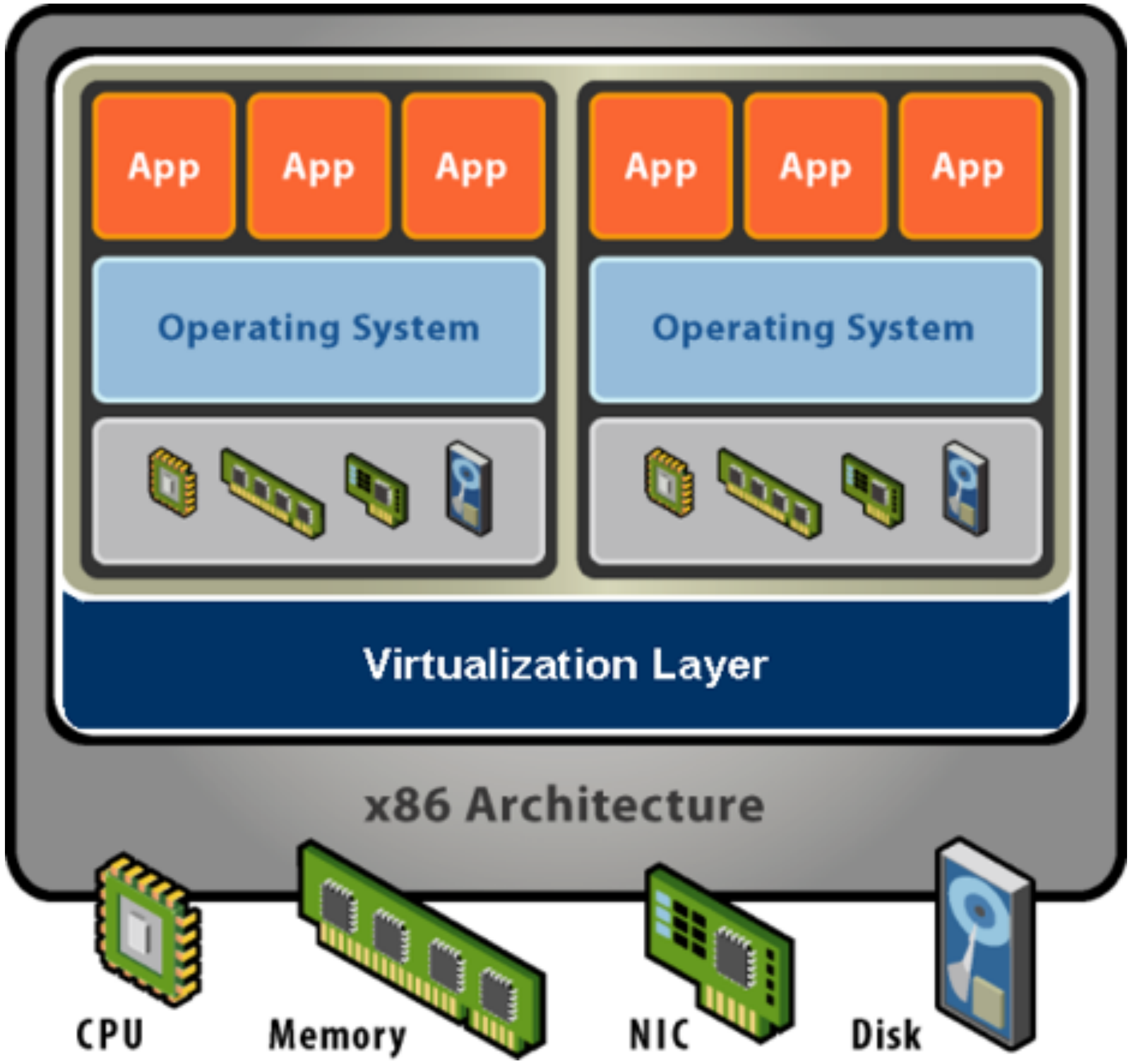
Main source: Smith & Nair, “Virtual Machines”, Morgan Kaufmann, 2005

Today

- What do we mean by virtualization?
- Why is it important to cloud?
- What is the penalty?
- Current trends

Virtualization

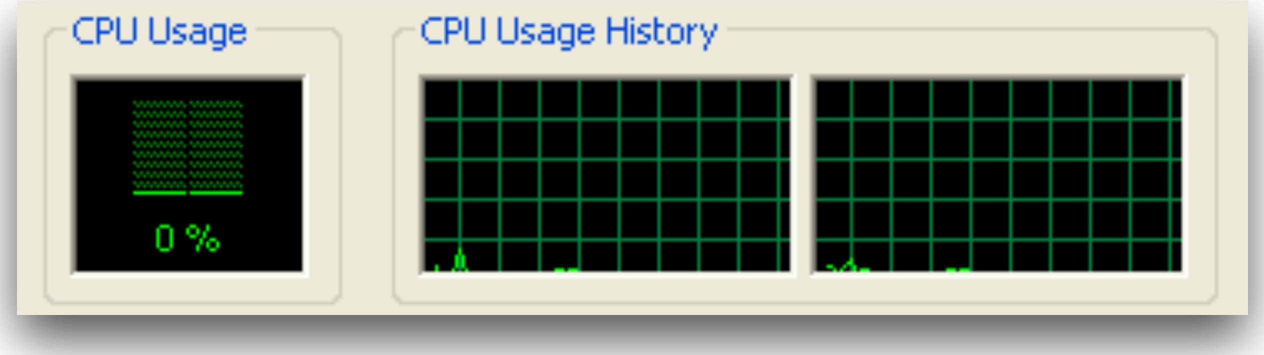
- CPU virtualization
- Memory virtualization
- Storage virtualization
- Device virtualization



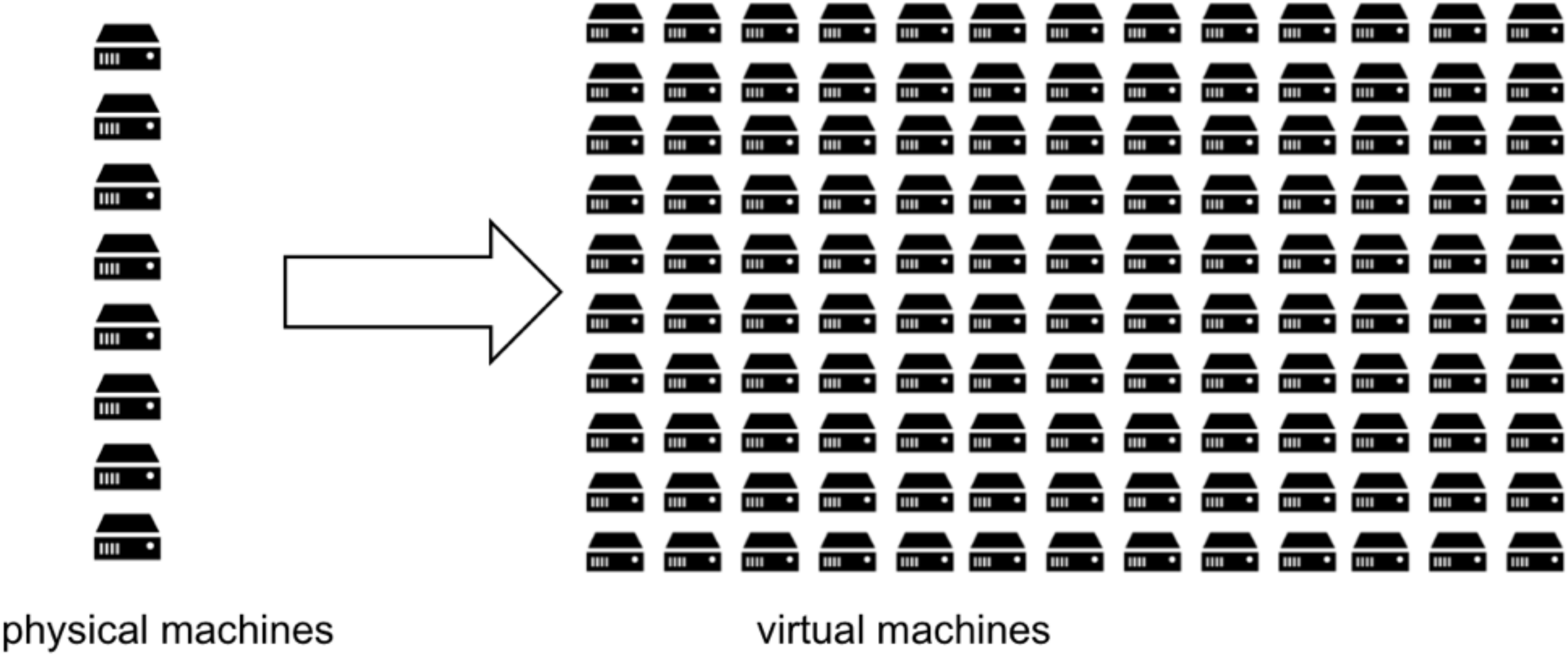
Cloud Motives

- Server Consolidation
  - Improve utilisation (possible to overcommit)
  - Significant cost savings (equipment, space, power)
- Simplified Management
  - Datacenter provisioning and monitoring
  - Dynamic load balancing
  - Migration (dead or alive)

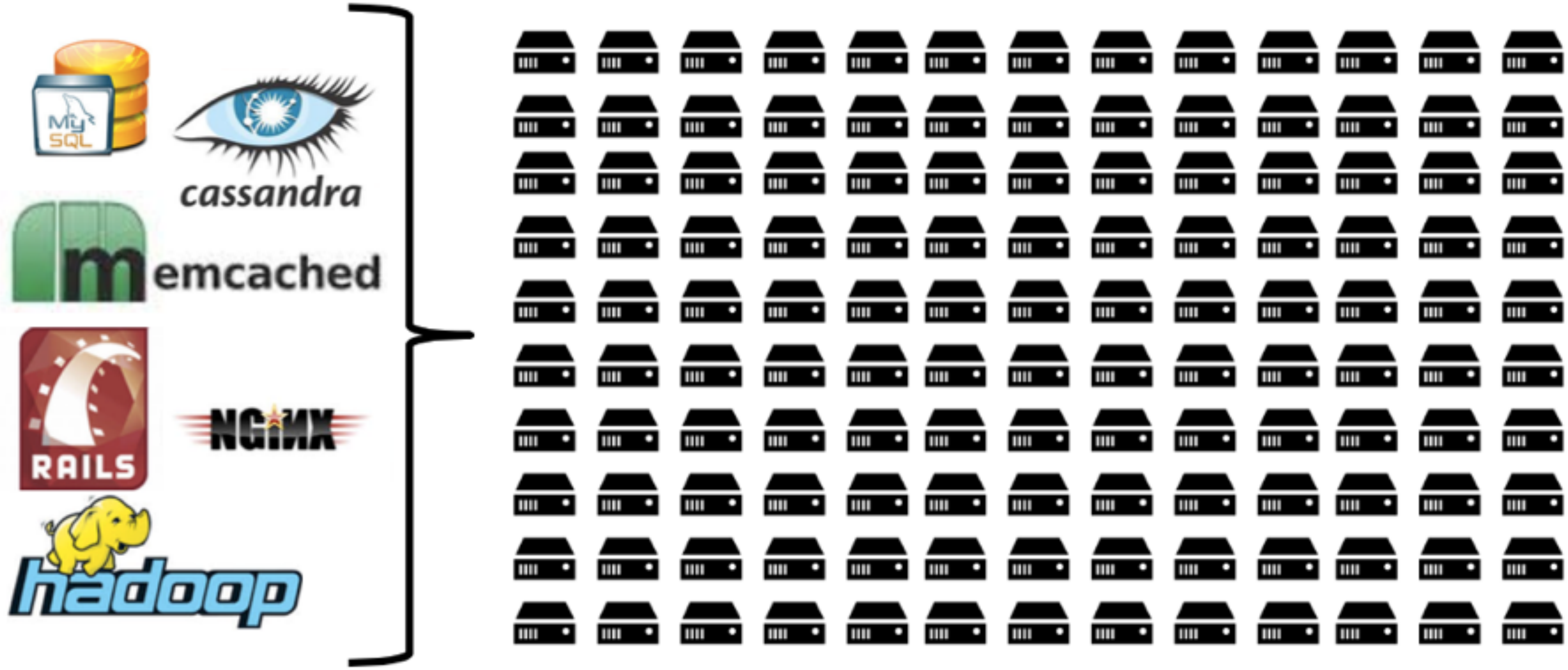
- Improved Availability
  - Checkpointing
  - Fault tolerance
  - Disaster recovery
  - Replication
- Security
- Isolation
- Convenient for users



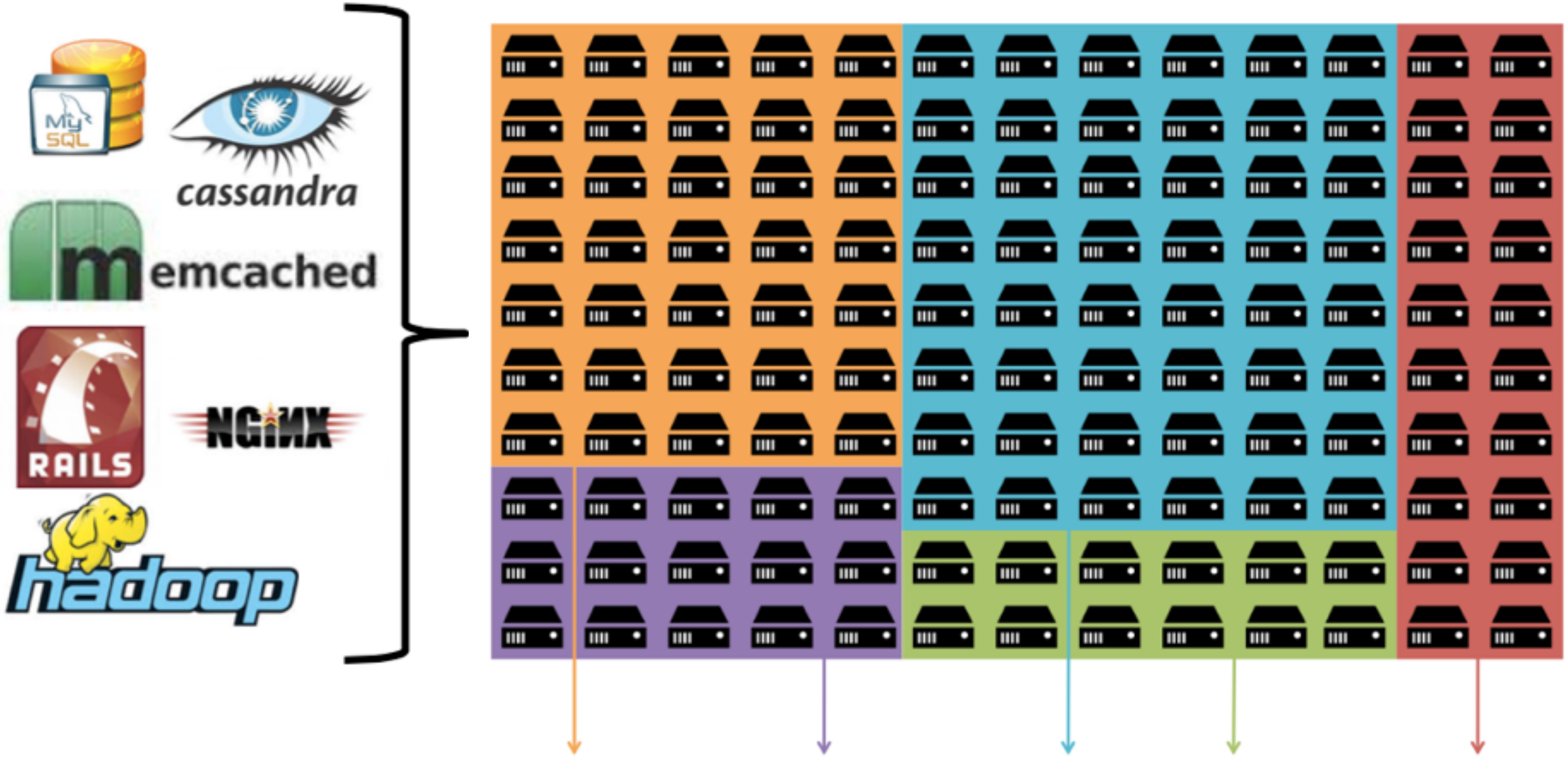
Cloud Resource Management



Cloud Resource Management



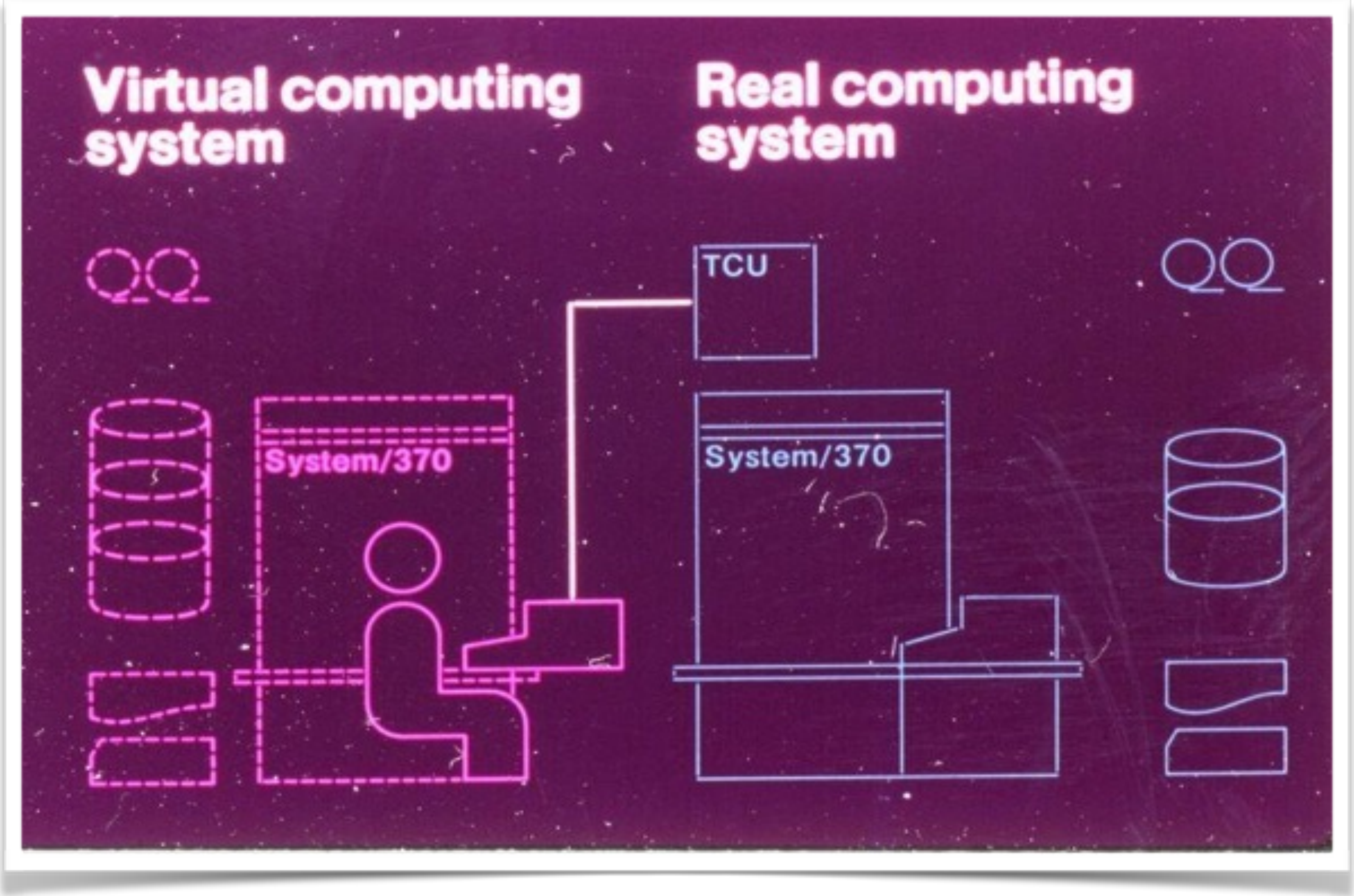
Cloud Resource Management



mysql cassandra Rails/nginx hadoop memcache

Yesterday’s News

- Classical VMM
  - IBM S/360, IBM VM/370
  - Co-designed proprietary hardware, OS, VMM
  - “Trap and emulate” model
- Applications
  - Timeshare several single-user OS instances on expensive hardware
  - Compatibility

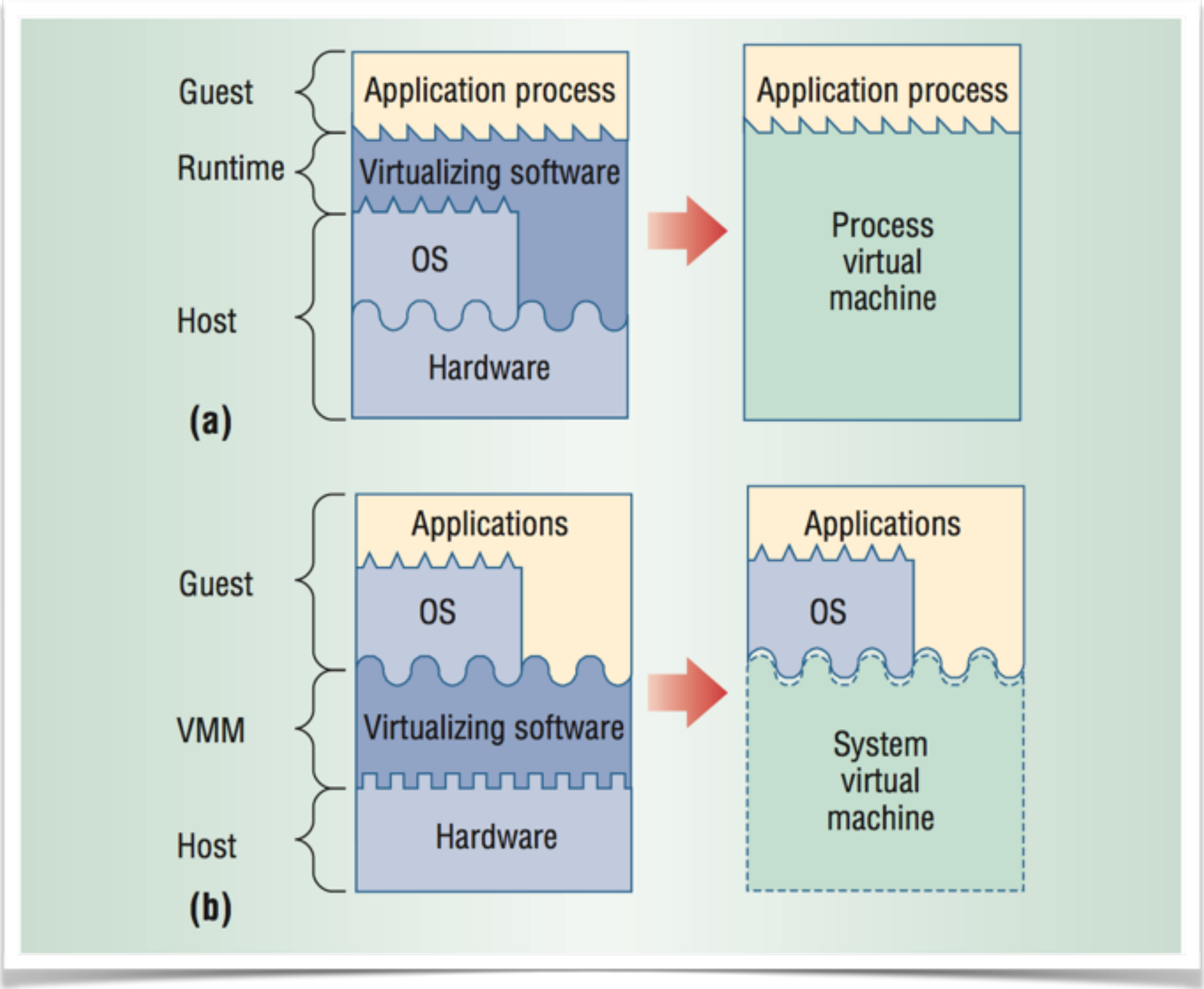


From IBM VM/370 product announcement, *ca*. 1972

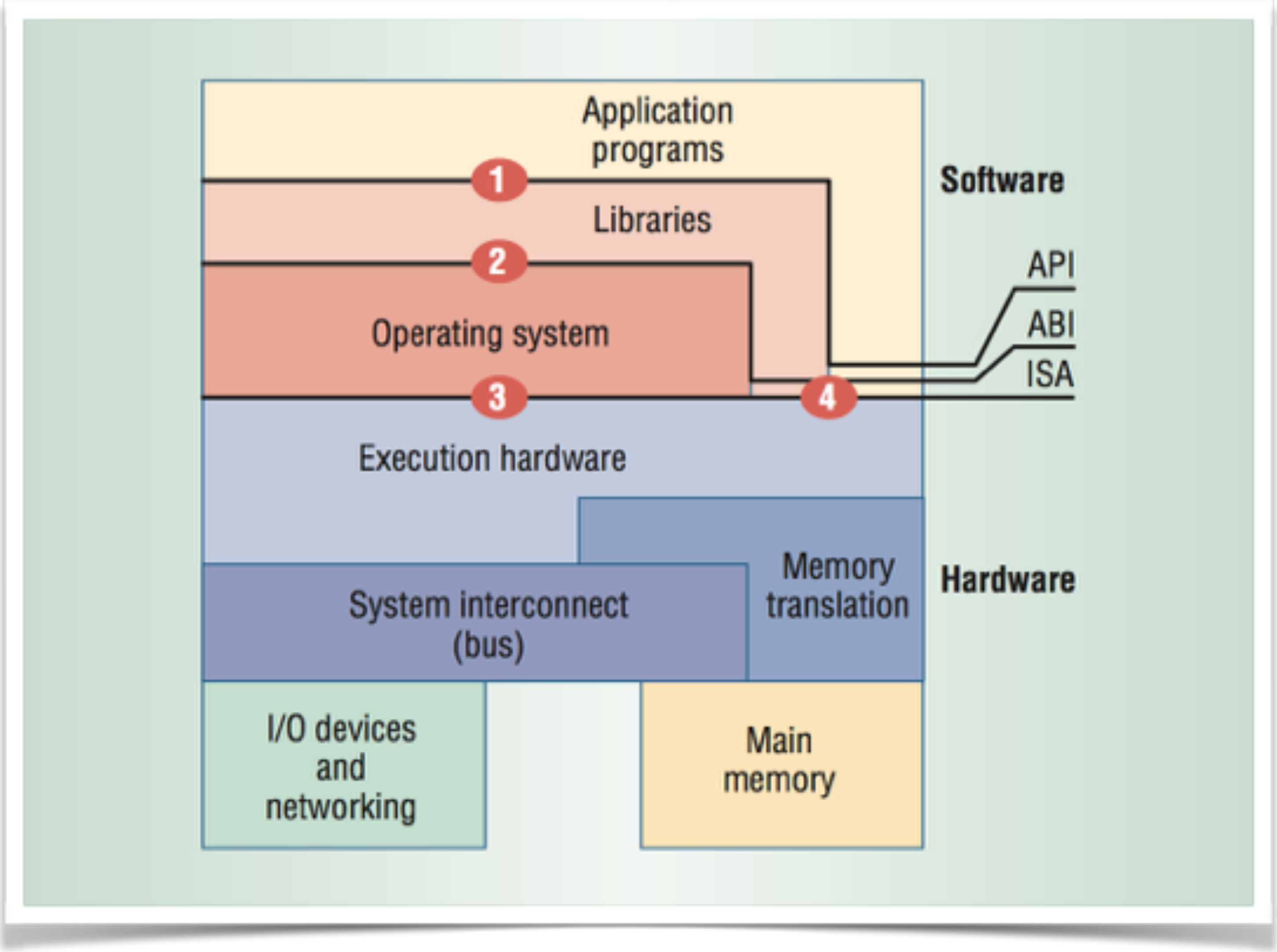
Original Motives ‘65

- Multiprogramming
- Multiple single application VMs
- Multiple secure environments
- Managed application environments
- Mixed OS environments
- Legacy applications
- New systems transitions
- Software development
- OS training
- Help desk support
- Operating system instrumentation
- Event monitoring
- Check pointing

System VM vs Process VM

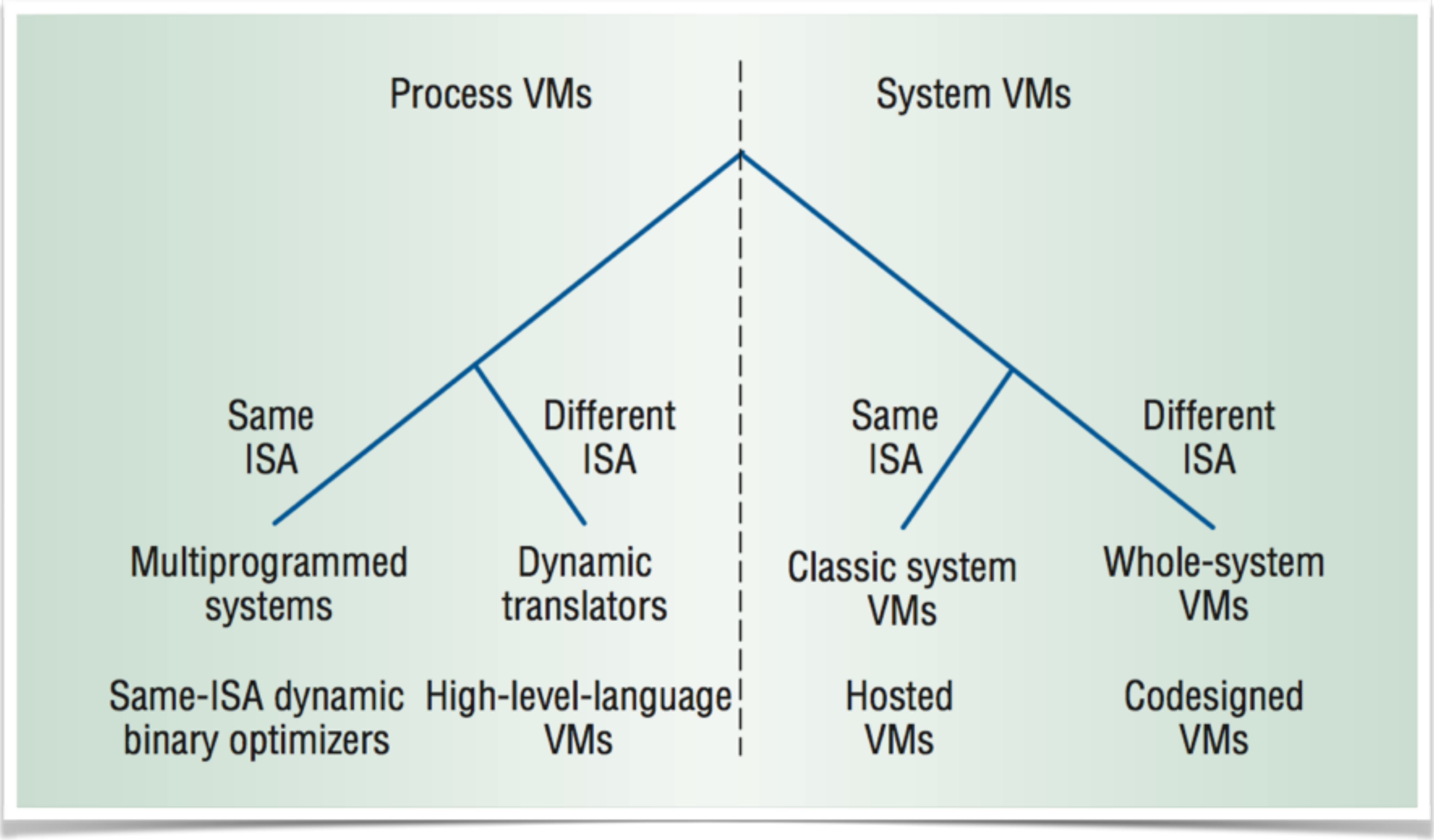


Virtualization Interfaces

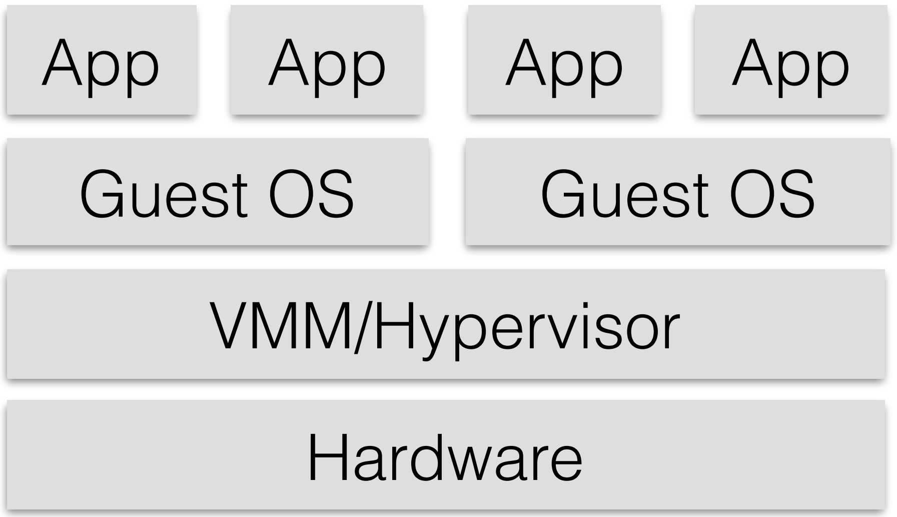


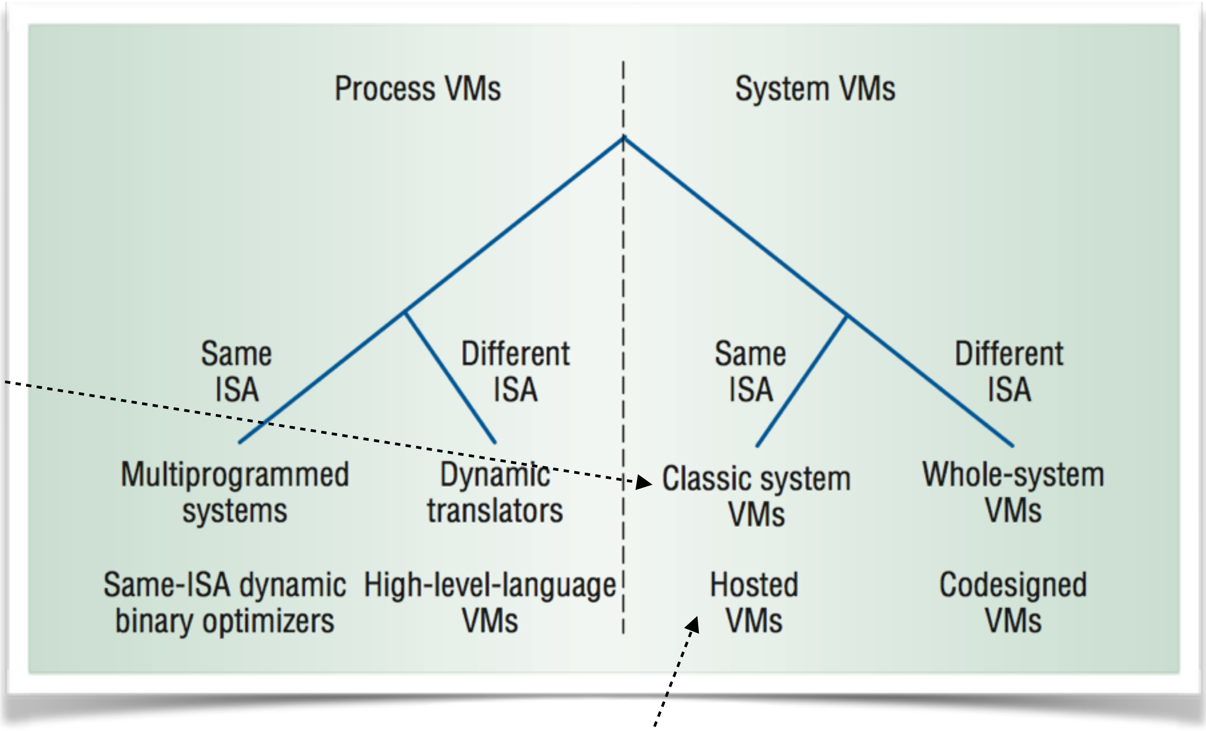
- OS → ISA
  - Instruction Set Architecture
- Compiler → ABI
  - Application Binary Interface
  - User ISA + ABI
- Application → API
  - Application Programming Interface
  - User ISA + API

VM Taxonomy

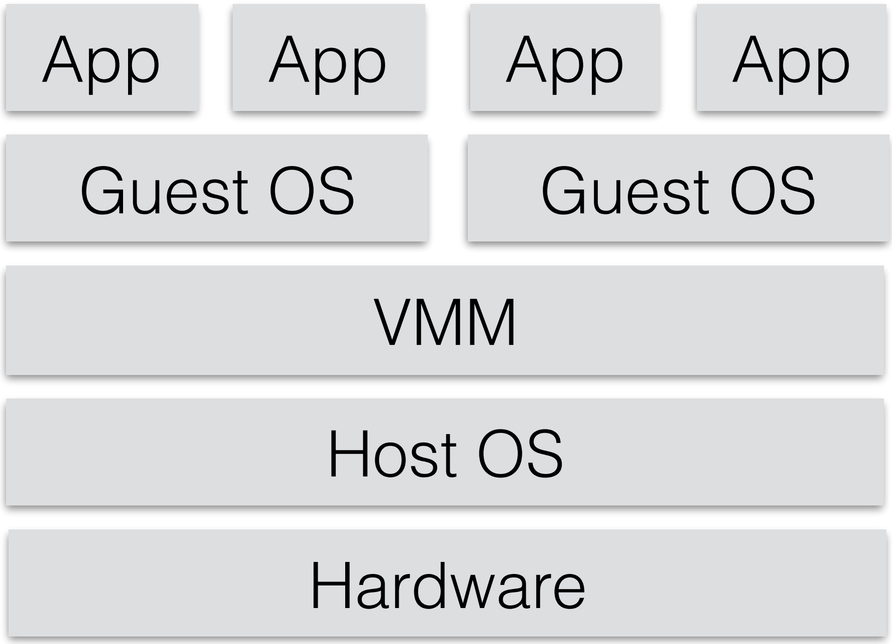

Process VMs
System VMs
Same ISA
Different ISA
Same ISA
Different ISA
Multiprogrammed systems
Dynamic translators
Classic system VMs
Whole-system VMs
Same-ISA dynamic binary optimizers
High-level-language VMs
Hosted VMs
Codesigned VMs

VM Taxonomy

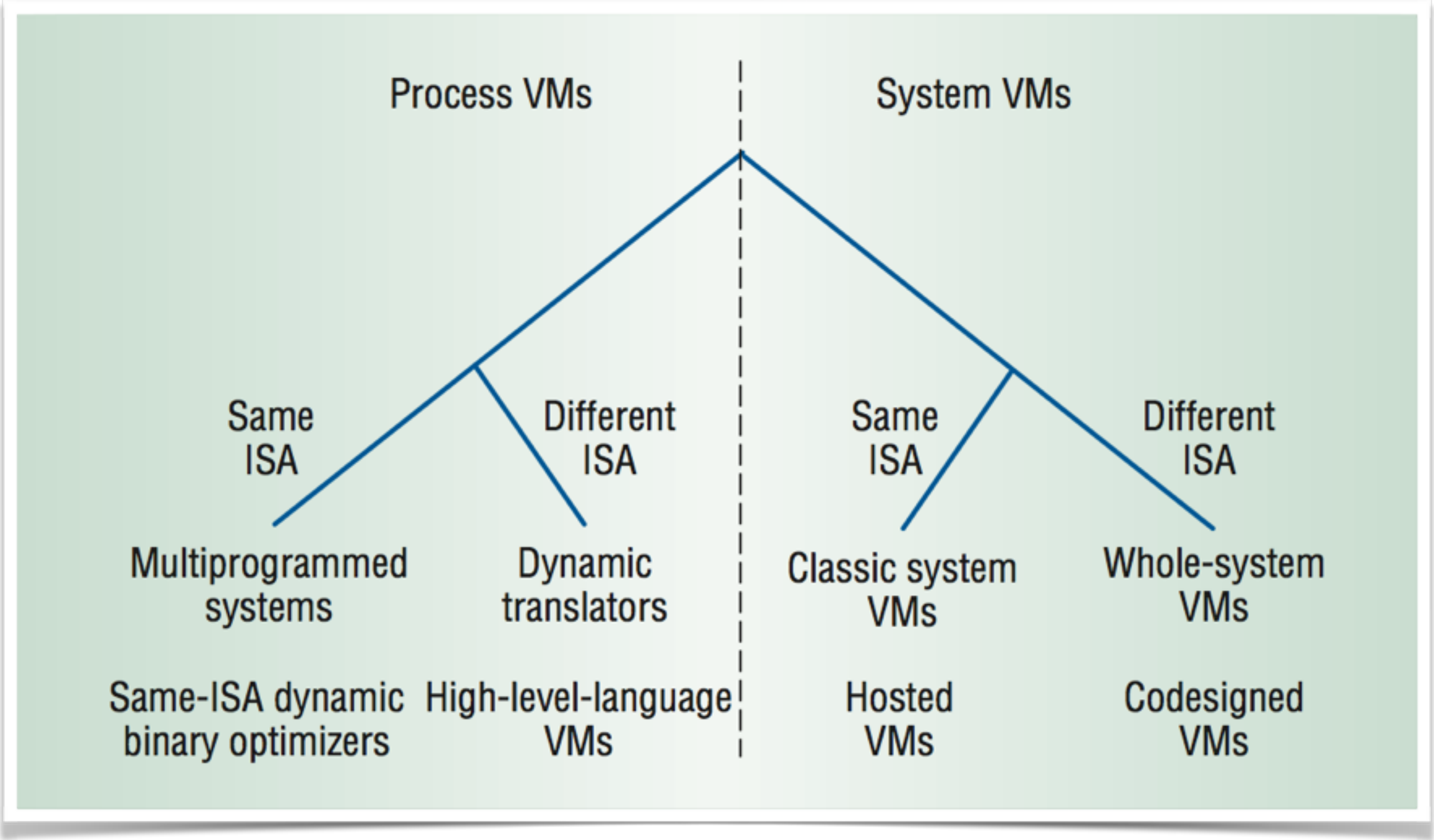




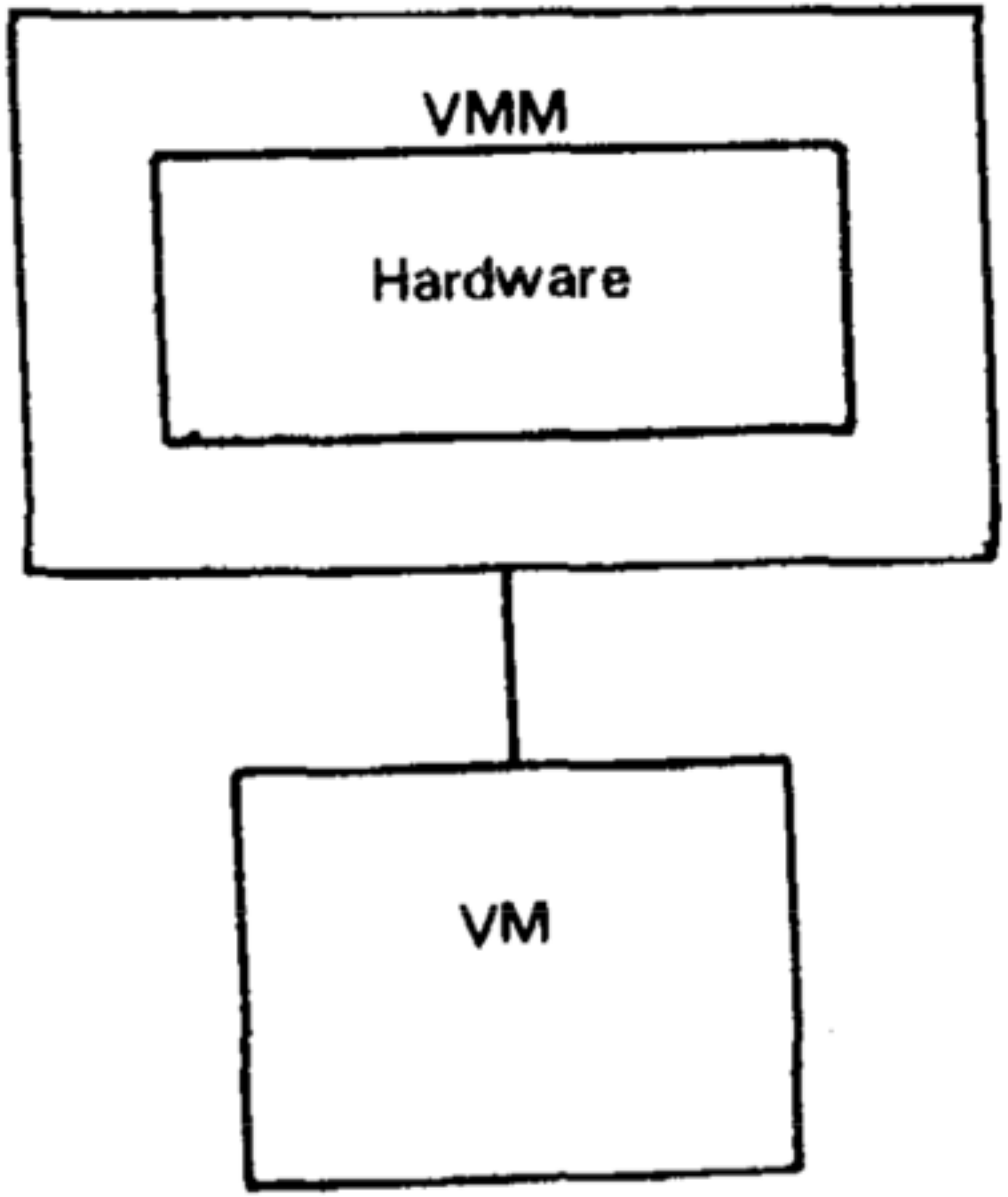
- Virtual Machine Monitor
- Type-1: “bare metal”
  - VMM = Hypervisor
  - OS/370 (CP), VMWare ESXi
- Type-2: “hosted”
  - KVM, VirtualBox

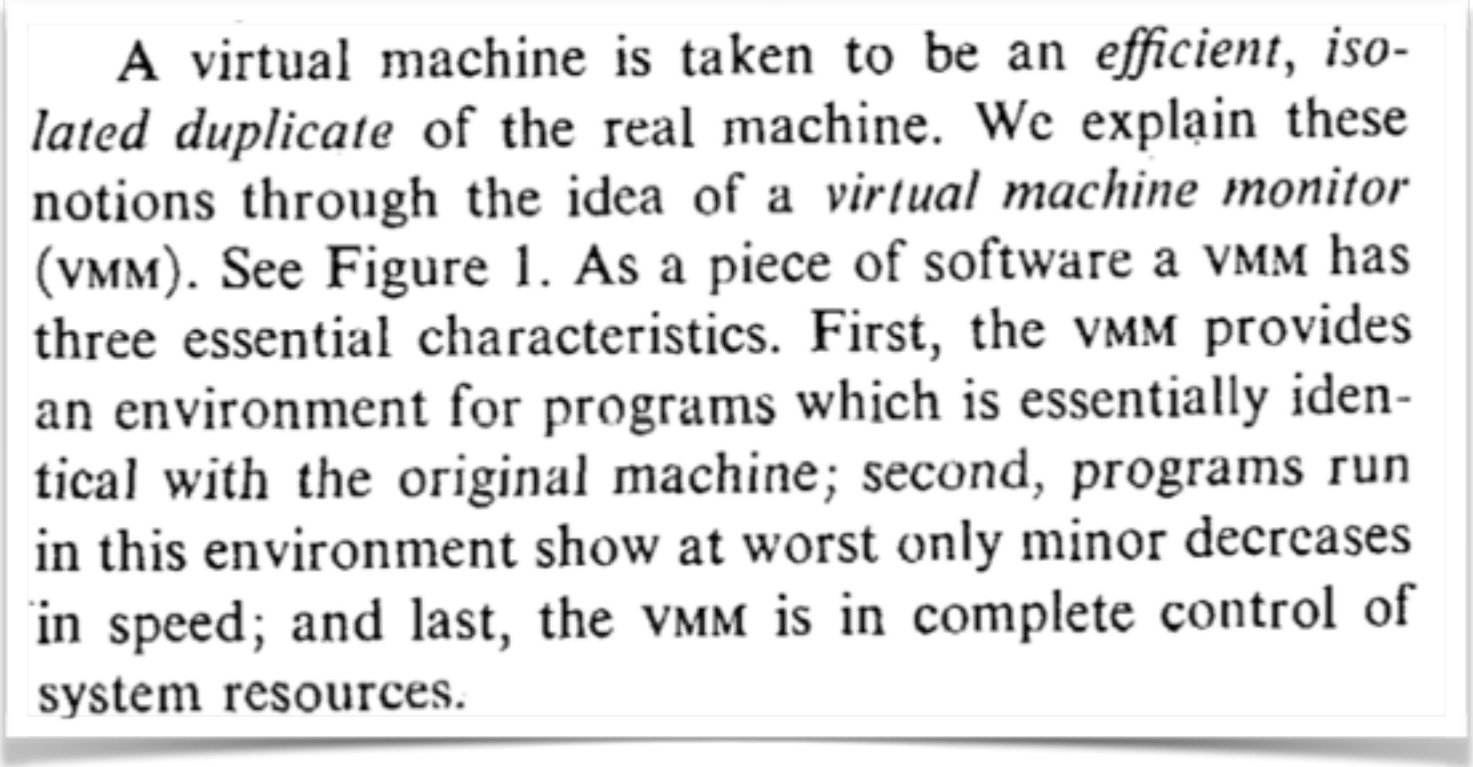


Where is the PaaS & IaaS?



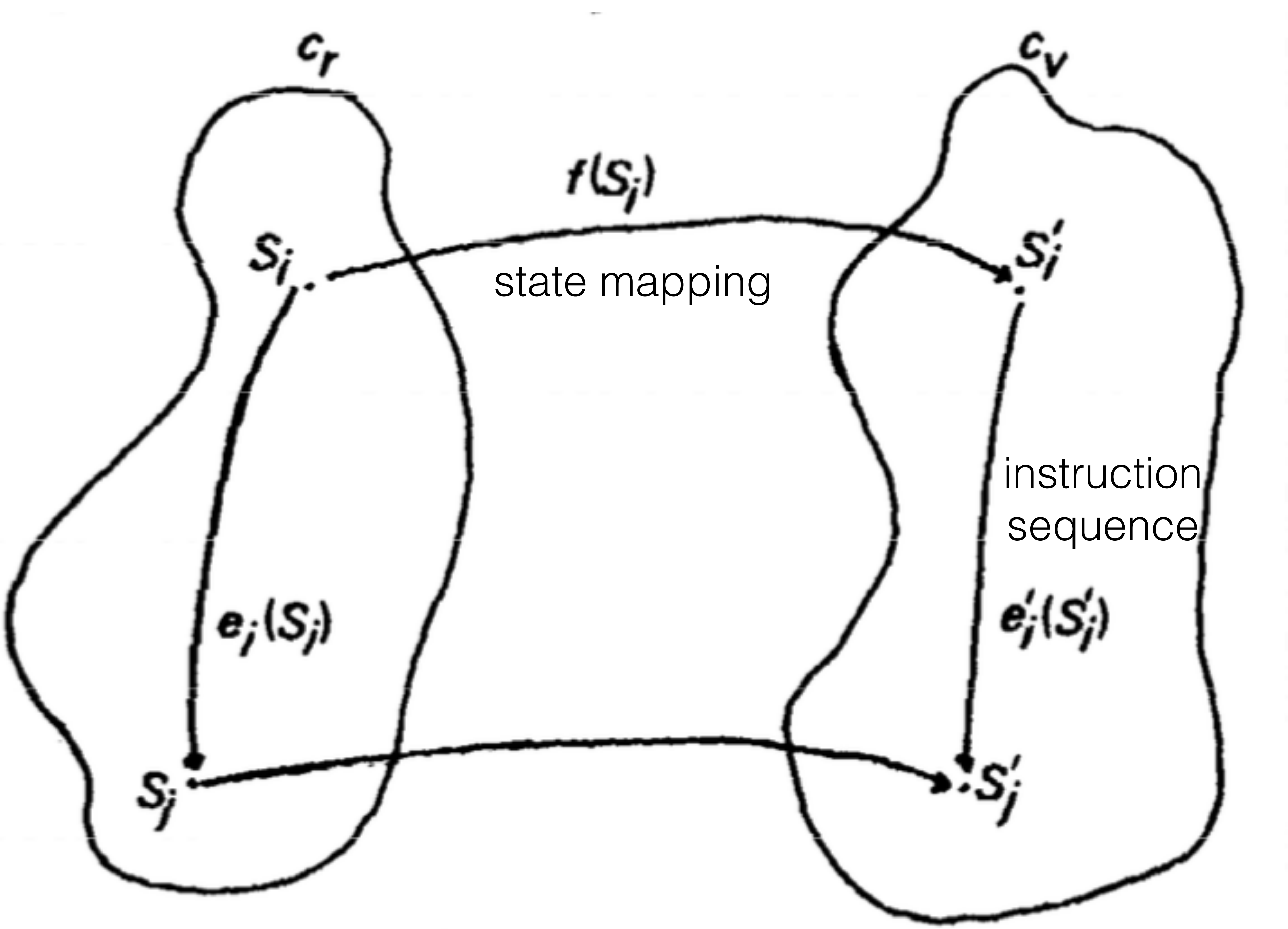
Popek & Goldberg ‘74




A virtual machine is taken to be an *efficient*, *isolated duplicate* of the real machine. We explain these notions through the idea of a *virtual machine monitor* (VMM). See Figure 1. As a piece of software a VMM has three essential characteristics. First, the VMM provides an environment for programs which is essentially identical with the original machine; second, programs run in this environment show at worst only minor decreases in speed; and last, the VMM is in complete control of system resources.

Popek & Goldberg ‘74

guest host



state mapping

instruction sequence

existence of map & instruction sequences such that:

$$f(e_i(S_i) = e_i'(f(S_i))$$

Popek & Goldberg ‘74

Essential Properties of a VMM

- Equivalence:
  - Running on VMM = Running directly on HW
- Performance:
  - Performance on VMM ≈ Performance on HW
- Resource control:
  - The VMM must have complete control of the virtualized resources

Popek & Goldberg ‘74

When is it possible to fulfil these requirements?

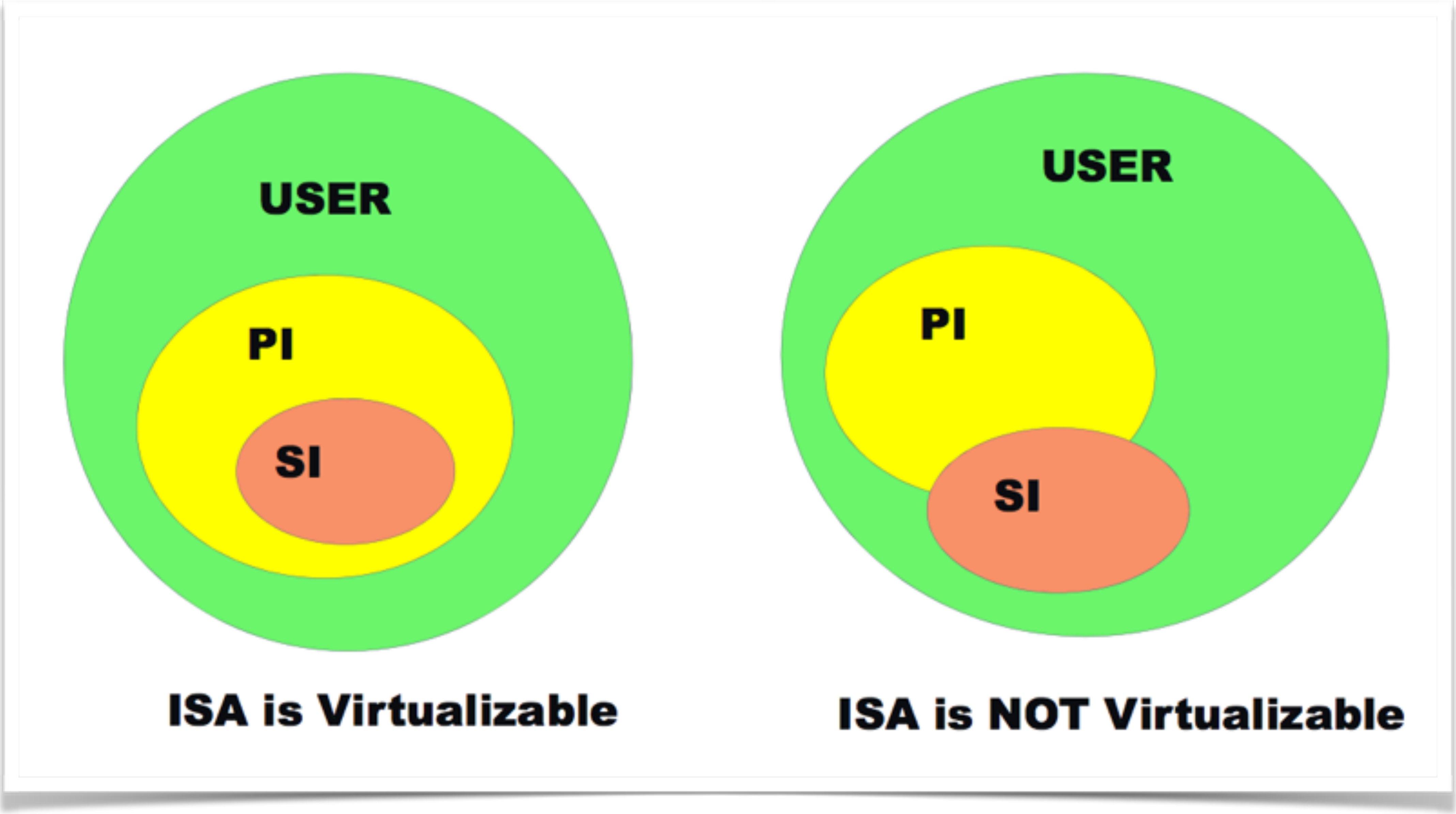
Popek & Goldberg ‘74

A Few Definitions

- Privileged instructions (PI): *must* generate trap when executed in any but the most privileged level
    - Execute in privileged mode, trap in user mode
- Privileged state: determines resource allocation
    - Privilege mode, addressing context, exception vectors, ...
- Sensitive instructions (SI): instructions whose behavior depends on the current privilege level
    - Control sensitive: change privileged state
    - Behavior sensitive: exposes privileged state

Popek & Goldberg '74

Virtualizable ISA If:



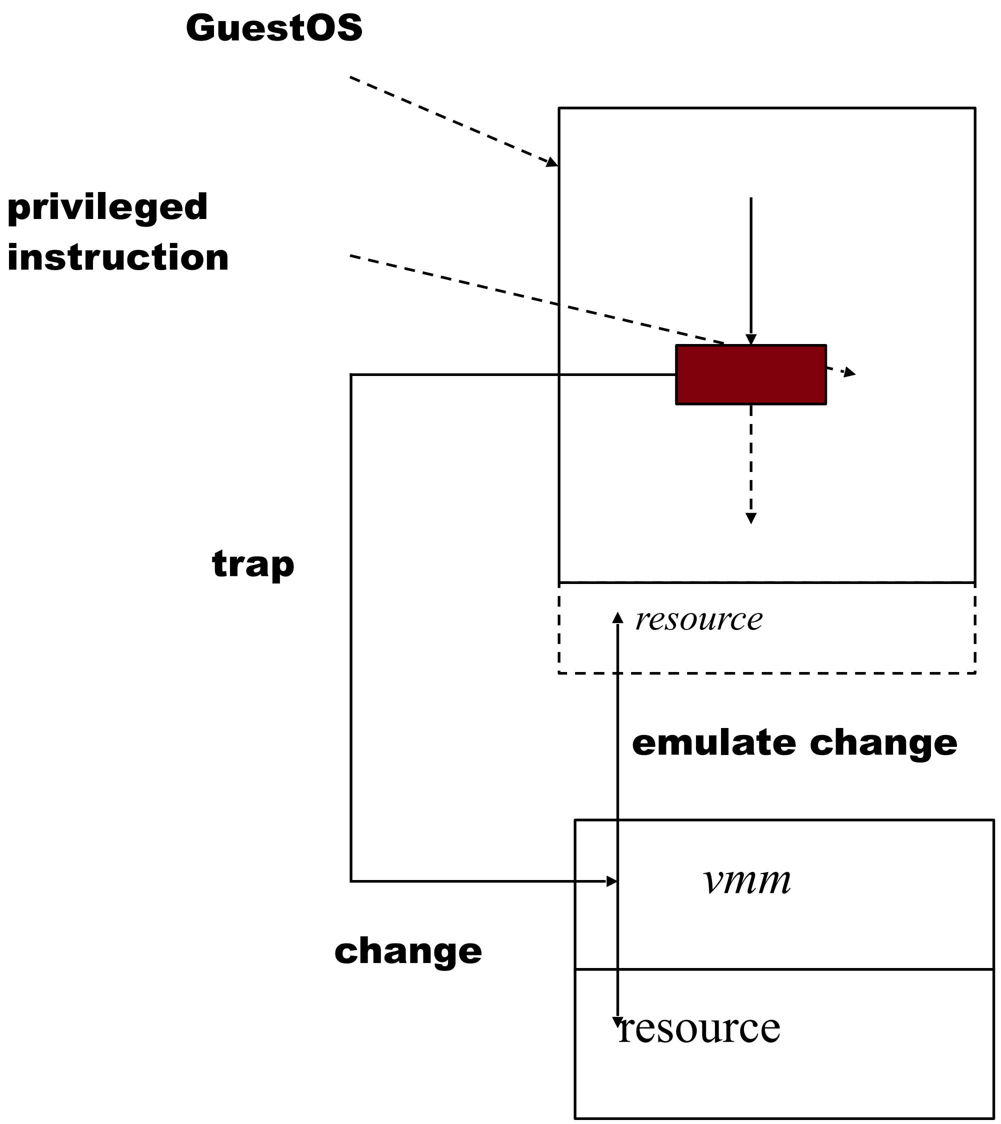
Theorem 1: A VMM may be constructed if the set of SI’s is a subset of the set of PI’s

Popek & Goldberg ‘74

Virtualization Approaches

- Trap-and-emulate
- Binary translation
- Paravirtualization
- Hardware-assisted Virtualization

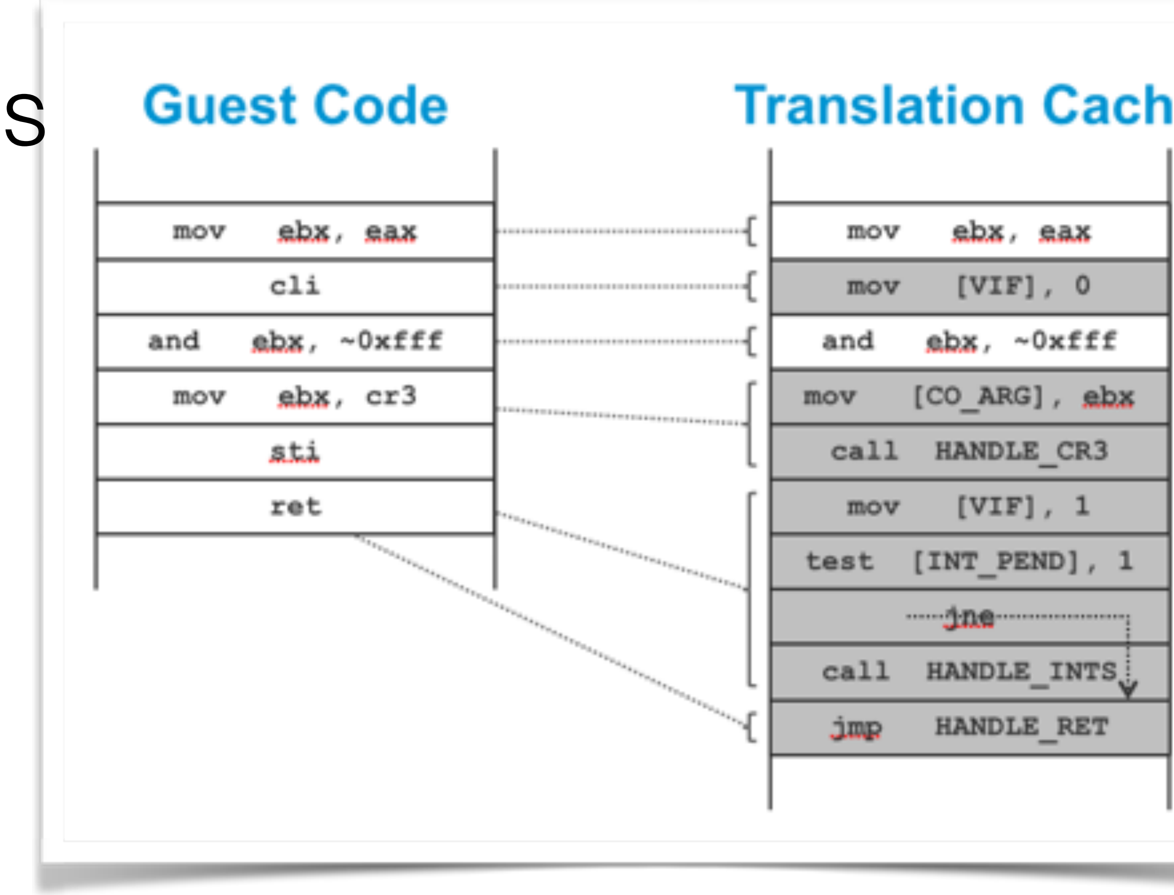
Trap & Emulate



- De-privileging
  - Run guest OS in unprivileged mode
- Privileged instructions trap, and VMM emulates
- Execute guest instructions on real CPU when possible

Binary Translation

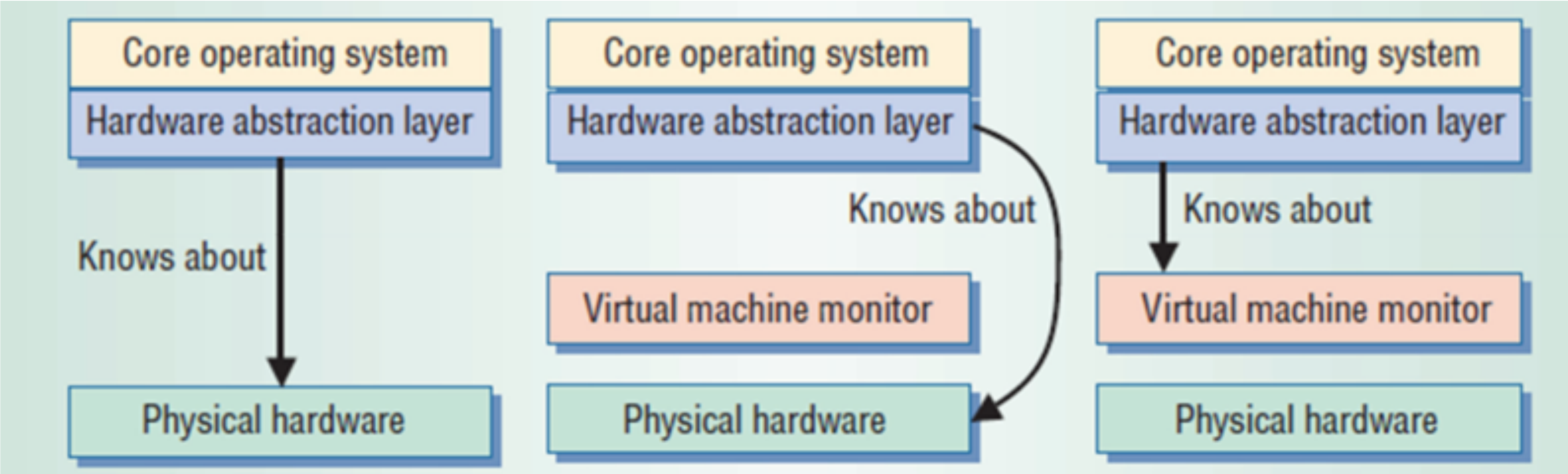
- Interpret the binary code
  - Replace privileged instructions
- Dynamic or static
- Use cache to speed up
- Hosted VM
- Popularised by VMWare on x86



```
mov    ebx, eax
cli
and    ebx, ~0xfff
mov    ebx, cr3
sti
ret
```

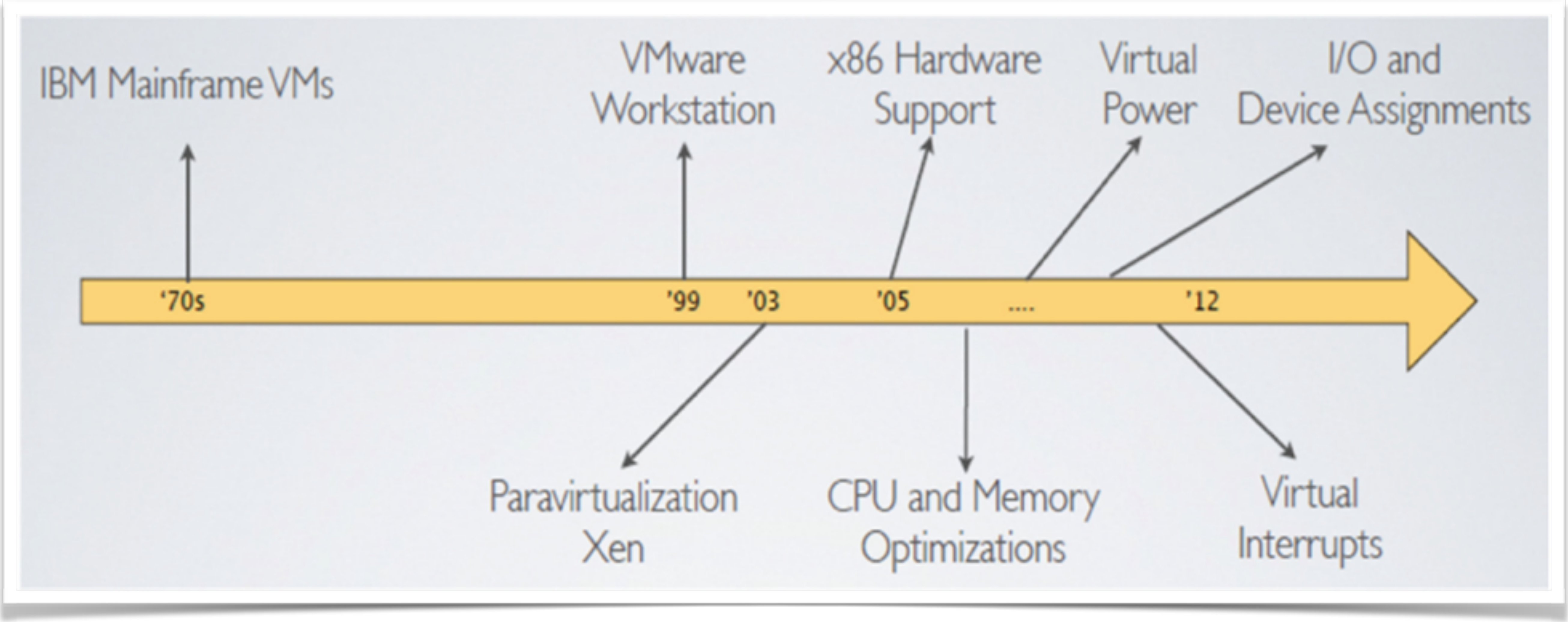
```
mov    ebx, eax
mov    [VIF], 0
and    ebx, ~0xfff
mov    [CO_ARG], ebx
call   HANDLE_CR3
mov    [VIF], 1
test   [INT_PEND], 1
jne
call   HANDLE_INTS
jmp    HANDLE_RET
```

Paravirtualization



- Less of a duplicate for better performance
- OS or system devices are virtualization aware
  - Recompile the OS
  - Guest applications unaffected
- Popularised by XEN for x86

Virtualization Timeline



Next

- Hardware virtualization (Robert Marklund)
- Containers & Docker (Linus)