IIT Kharagpur | IIT Madras | IIT Goa | IIT Palakkad

Accelerated AI

Introduction to Operating Systems, Virtualization, Cloud

Dr. Satyajit Das
Assistant Professor
Data Science
Computer Science and Engineering
IIT Palakkad

National Supercomputing Mission

Centre for Development of Advanced Computing

System software

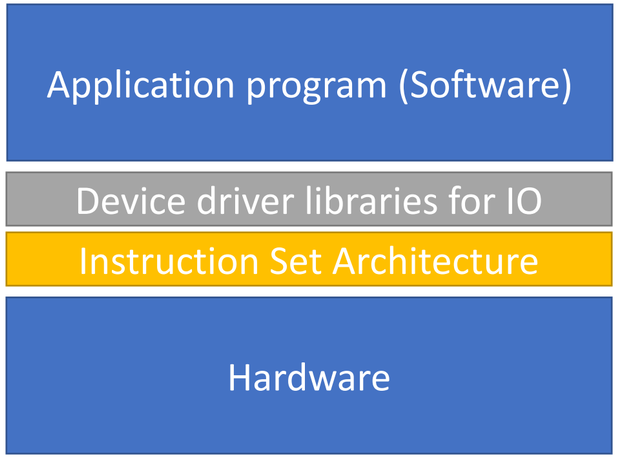
- operating system
- Utilities
- device drivers
- language translators

What is an operating system

- A bunch of software and data residing somewhere in memory.
  - But its not just any software.
- OS is the most privileged software in a computer.
  - Privileged means that OS can do special things, like write to disk, talk over the network, control memory and CPU usage, etc.
- OS manages all system resources
  - CPU, Memory, and I/O devices

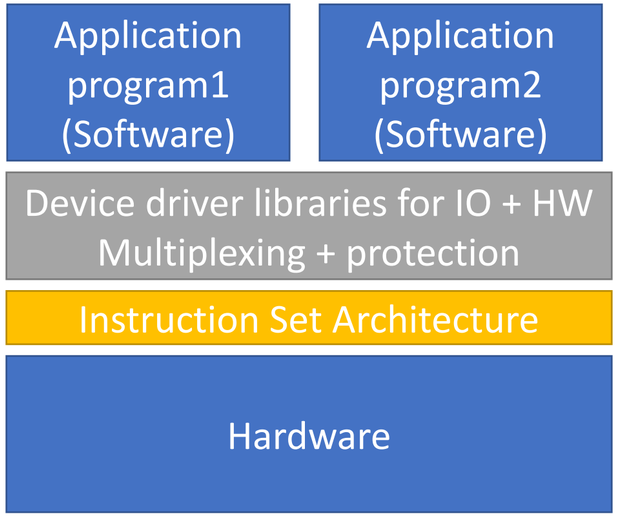
Why there is a need for OS

- Programs do not know how to access system I/O



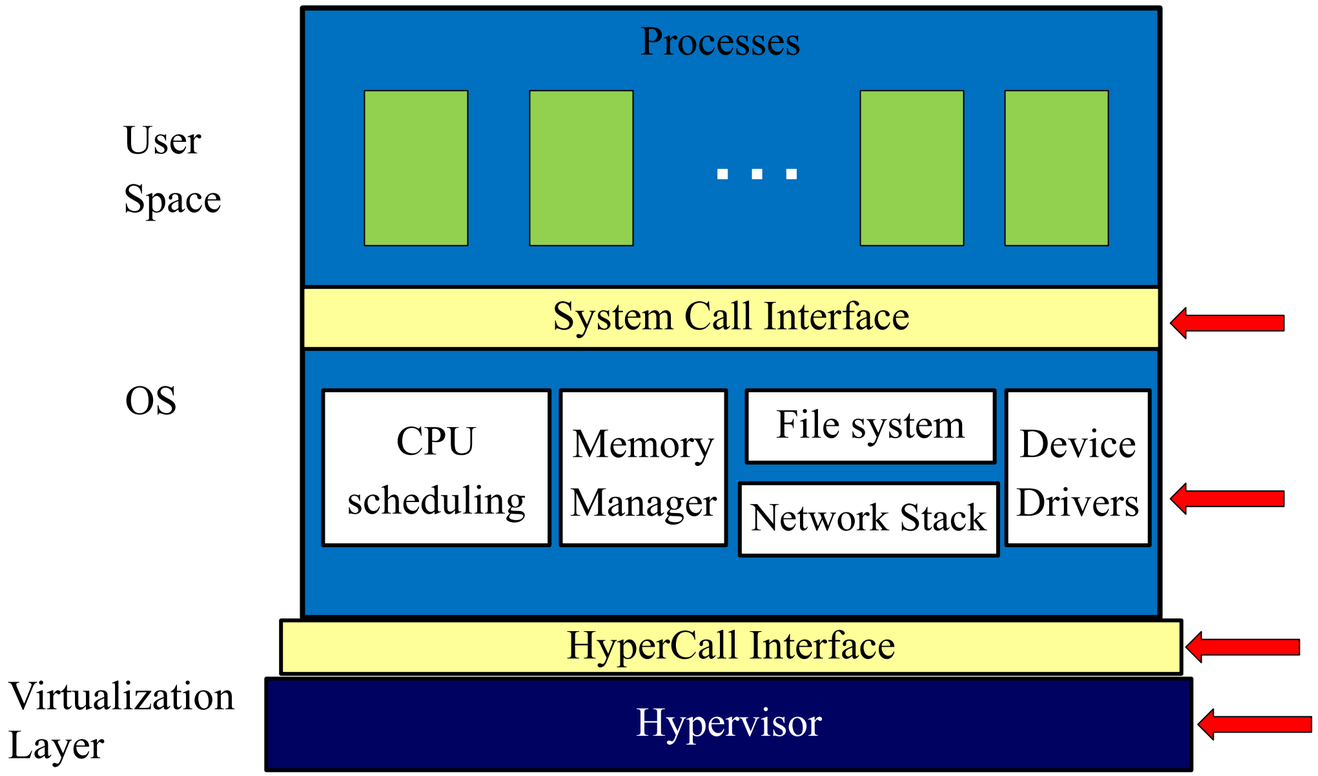
Why there is a need for OS

- What happens when multiple program needs to share the HW



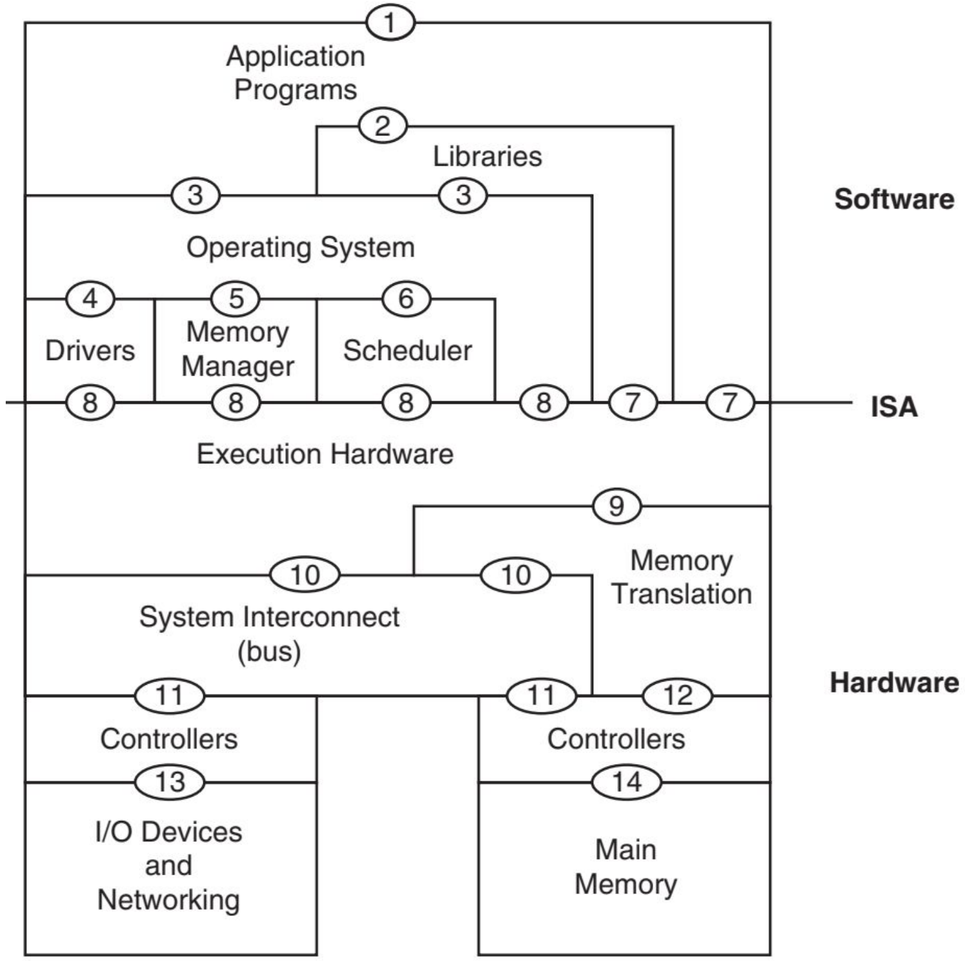
- Two programs does not trust each other
- OS does not trust programs
- HW does not trust programs

Software layers



Credit: Karthik Gopalan

Interfaces available in a computing system



User ISA : 7
System ISA : 8
Syscalls : 3
ABI : 3, 7
API : 2,7

ISA = Instruction Set Architecture
ABI = Application Binary Interface
API = Application Programming Interface

[credit: Smith and Nair]

Virtualization

- Makes a real system appear to be a set of virtual systems

- One-to-many virtualization
  - E.g. one physical machine may appear as multiple virtual machines
  - one physical disk may look like multiple virtual disk
  - one physical network may look like multiple virtual networks

- Many-to-one virtualization
  - Many physical machines/disks/networks may appear to look like one virtual machine/disk/network etc

- Many-to-many virtualization

Credit: Karthik Gopalan

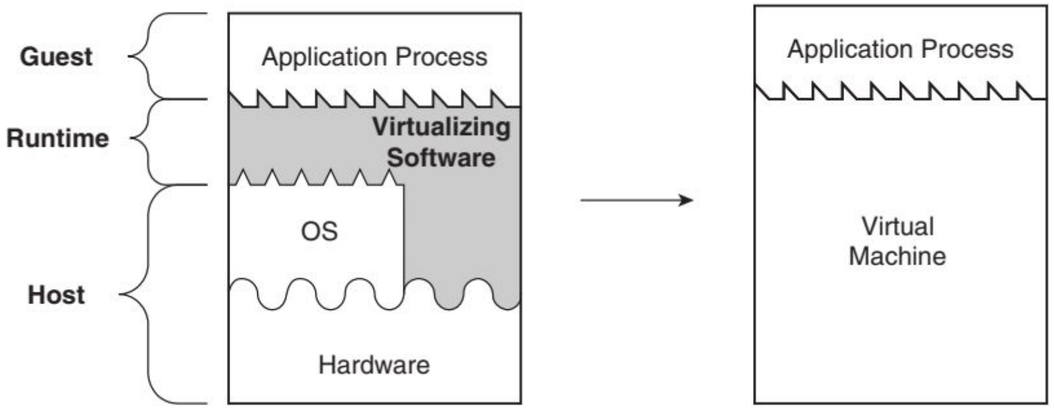
Virtual Machine (VM)

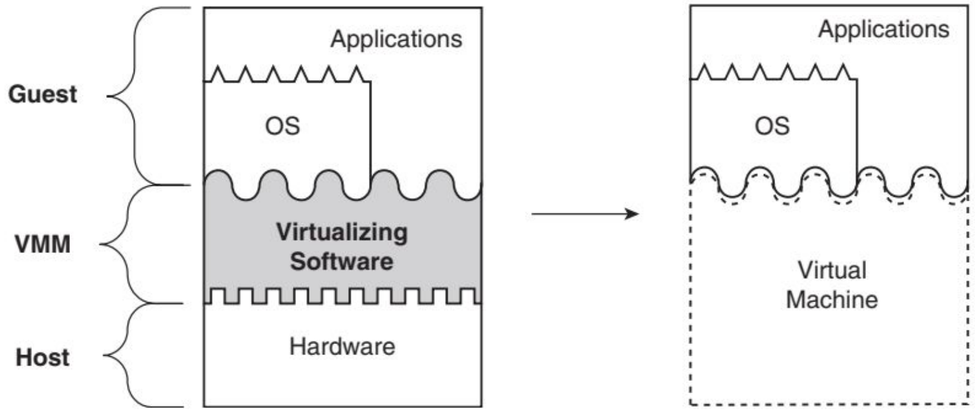
- Logical/Emulated representations of full computing system environment
  - CPU + memory + I/O
  - Implemented by adding layers of software to the real machine to support the desired VM architecture.
- Uses:
  - Multiple OSes on one machine, including legacy OSes
  - Isolation
  - Enhanced security
  - Live migration of servers
  - Virtual environment for testing and development
  - Platform emulation
  - On-the-fly optimization
  - Realizing ISAs not found in physical machines

Credit: Karthik Gopalan

Different types of VMs

- Process VM
  - Virtualizes the ABI
  - Virtualization software => Runtime
  - Runs in non-privileged mode (user space)
  - Performs binary translation.
  - Terminates when guest process terminates.
- System VM
  - Virtualizes the ISA
  - Virtualization software => Hypervisor
  - Runs in privileged mode
  - Traps and emulates privileged instructions



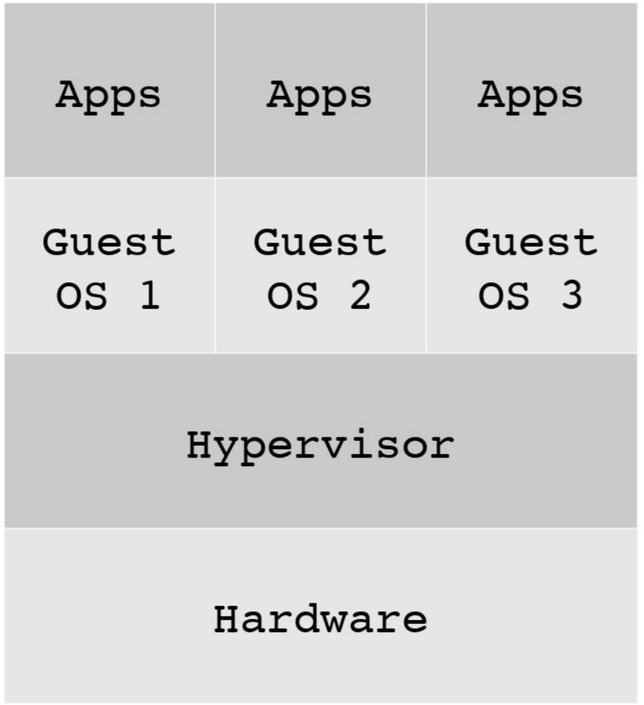


[credit: Smith and Nair]

System VM

Hypervisors

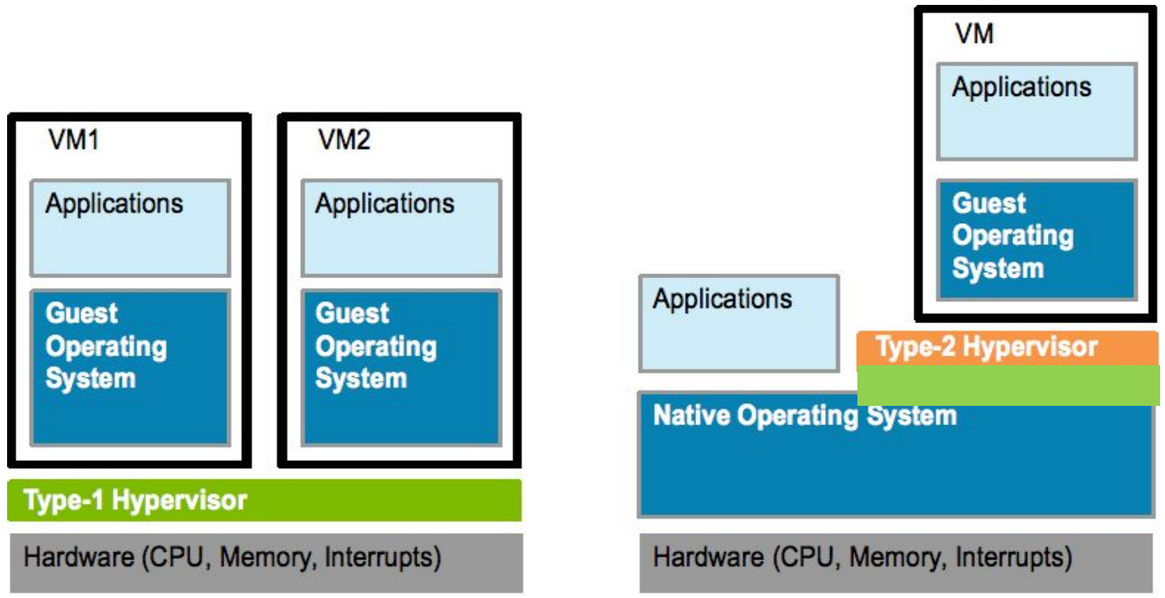
- Also called Virtual Machine Monitor (VMM)
- A hypervisor is an operating system for operating systems
  - Provides a virtual execution environment for an entire OS and its applications
  - Controls access to hardware resources
  - When guest OS executes a privileged instruction, Hypervisor intercepts the instruction, checks for correctness and emulates the instruction.



Credit: Karthik Gopalan

Type of Hypervisors

- Type 1 Hypervisor (bare metal, native): supports multiple virtual machines and runs directly on the hardware (e.g., VMware ESX , Xen, Denali)
- Type 2 Hypervisor (hosted) VM - runs under a host operating system (e.g., user-mode Linux)



https://microkerneldude.files.wordpress.com/2012/01/type1-vs-2.png

Para-virtualized VMs

- Modify guest OS for better performance
- Traditional Hypervisors provide full-virtualization
  - They expose to VMs virtual hardware that is functionally identical to the underlying physical hardware.
  - Advantage : allows unmodified guest OS to execute
  - Disadvantage: Sensitive instructions must be trapped and emulated by Hypervisor.
  - E.g. KVM and VMWare ESX provide full virtualization
- Para-virtualized VM
  - Sees a virtual hardware abstraction that is similar, but not identical to the real hardware.
  - Guest OS is modified to replace sensitive instructions with “hypercalls” to the Hypervisor.
  - Advantage: Results in lower performance overhead
  - Disadvantage: Needs modification to the guest OS.
  - Xen provides both para-virtual as well as full-virtualization
- Often traditional Hypervisors are partially para-virtualizated
  - Device drivers in guest OS may be para-virtualized whereas CPU and Memory may be fully virtualized.

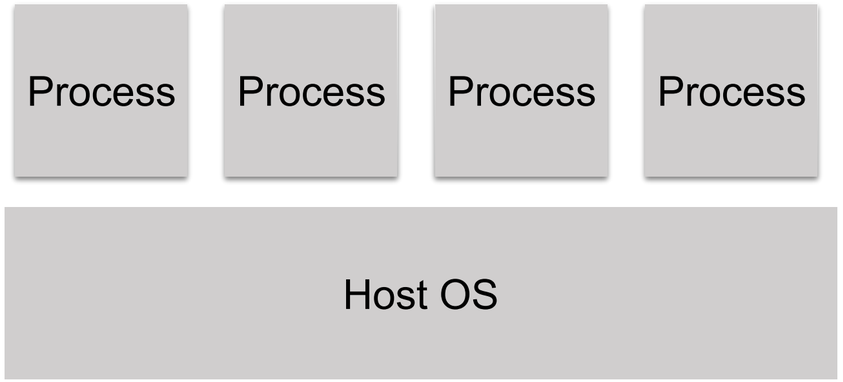
Credit: Karthik Gopalan

Examples of hypervisors

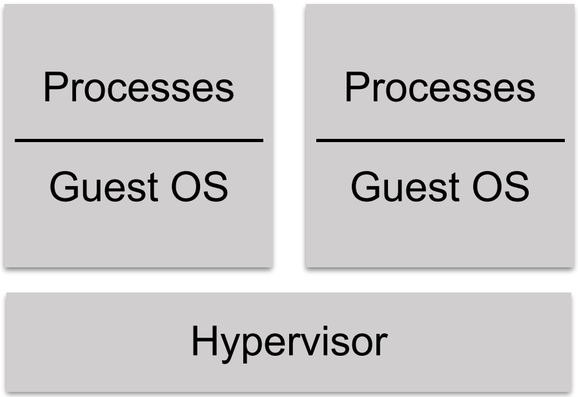
Name	Host ISA	Guest ISA	Host OS	guest OS	Company
Integrity VM	*x86*-64	*x86*-64	HP-Unix	Linux,Windows HP Unix	HP
Power VM	Power	Power	No host OS	Linux, AIX	IBM
z/VM	z-ISA	z-ISA	No host OS	Linux on z-ISA	IBM
Lynx Secure	*x86*	*x86*	No host OS	Linux, Windows	LinuxWorks
Hyper-V Server	*x86*-64	*x86*-64	Windows	Windows	Microsoft
Oracle VM	*x86*, *x86*-64	*x86*, *x86*-64	No host OS	Linux, Windows	Oracle
RTS Hypervisor	*x86*	*x86*	No host OS	Linux, Windows	Real Time Systems
SUN xVM	*x86*, SPARC	same as host	No host OS	Linux, Windows	SUN
VMware EX Server	*x86*, *x86*-64	*x86*, *x86*-64	No host OS	Linux, Windows Solaris, FreeBSD	VMware
VMware Fusion	*x86*, *x86*-64	*x86*, *x86*-64	MAC OS *x86*	Linux, Windows Solaris, FreeBSD	VMware
VMware Server	*x86*, *x86*-64	*x86*, *x86*-64	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Workstation	*x86*, *x86*-64	*x86*, *x86*-64	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Player	*x86*, *x86*-64	*x86*, *x86*-64	Linux Windows	Linux, Windows Solaris, FreeBSD	VMware
Denali	*x86*	*x86*	Denali	ILVACO, NetBSD	University of Washington
Xen	*x86*, *x86*-64	*x86*, *x86*-64	Linux Solaris	Linux, Solaris NetBSD	University of Cambridge

Credit: Anil Madhavapeddy

Traditional Process vs traditional VMs



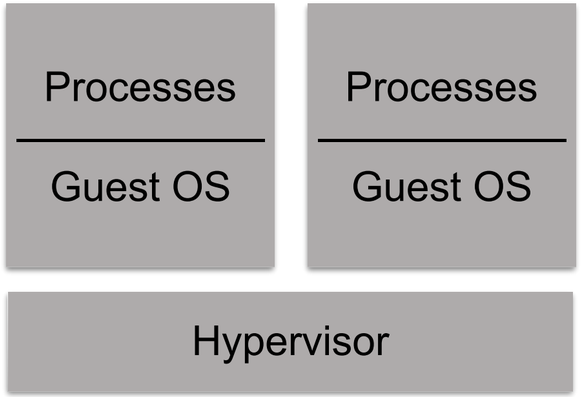
Traditional Processes



Traditional VMs

System VMs

- Each VM has its own
  - Guest OS
  - Guest physical memory (“virtualized” view of memory seen by guest OS)
  - One or more virtual CPUs
  - Virtual I/O devices: virtual disk, virtual network
- Ideally: Co-located VMs don’t see/share ANYTHING

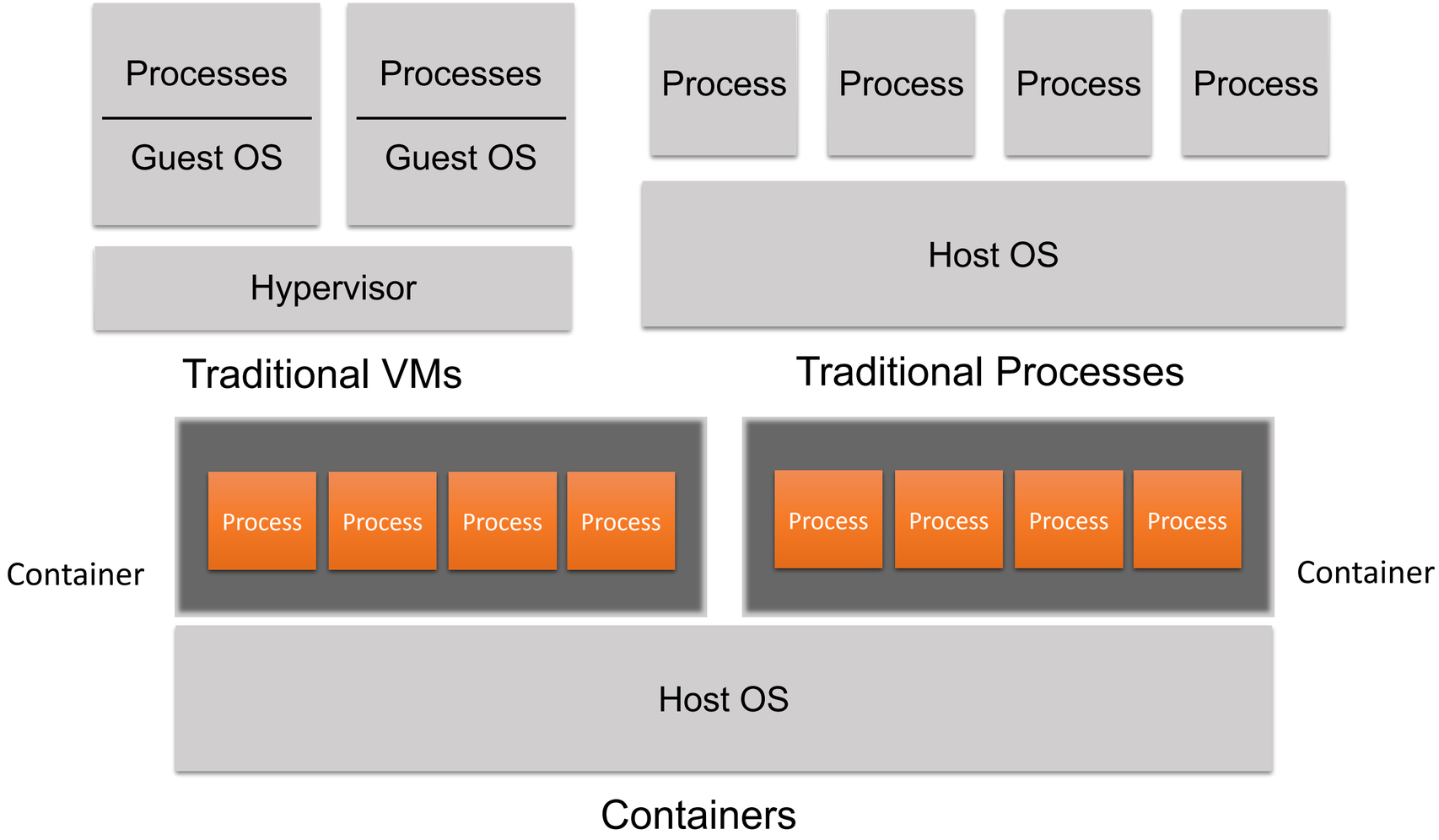


Isolation is important

- Limiting what/who a process/application can see.
- Limiting who can see a process/application

- Processes share too much
  - Great performance but not isolated enough
  - System VMs are too heavy
- Great Isolation but too heavy due to separate guest OS per VM
  Operating-system-level virtualization
  - Multiple isolated user-spaces
  - Share one kernel
  - Native performance

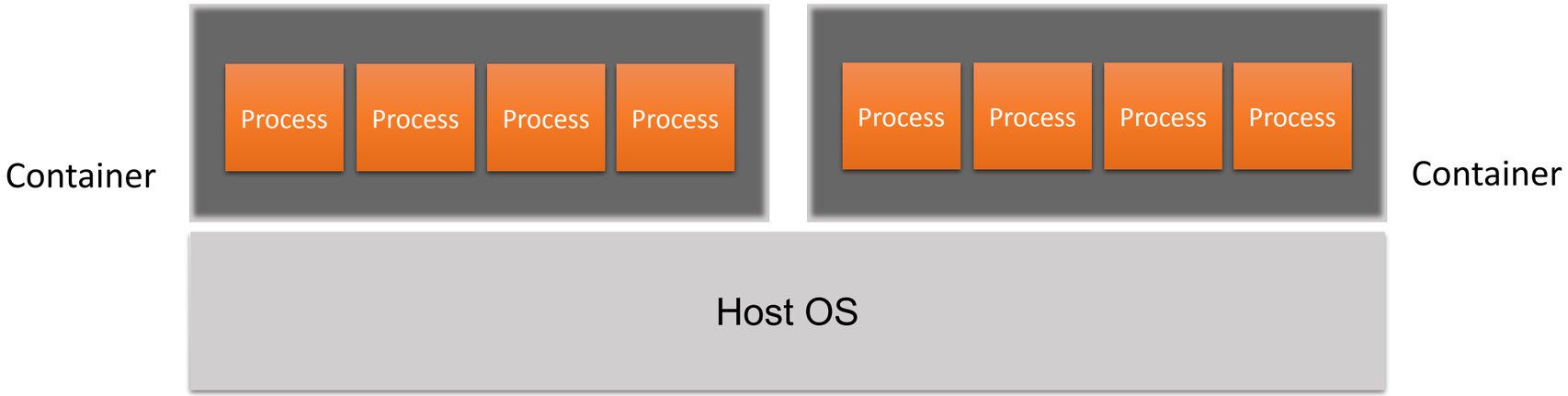
Process VMs and Containers



Credit: Karthik Gopalan

Containers

- Containers
  - group traditional processes together and
  - restrict what resources they can see/access.
- In Linux, containers consist of
  - Namespaces
  - Control Groups (cgroups)



Containers

Cloud Computing

- Virtualized distributed processing, storage, and software resources and a service.

- Delivering computing as a on-demand, pay-as-you-go service.

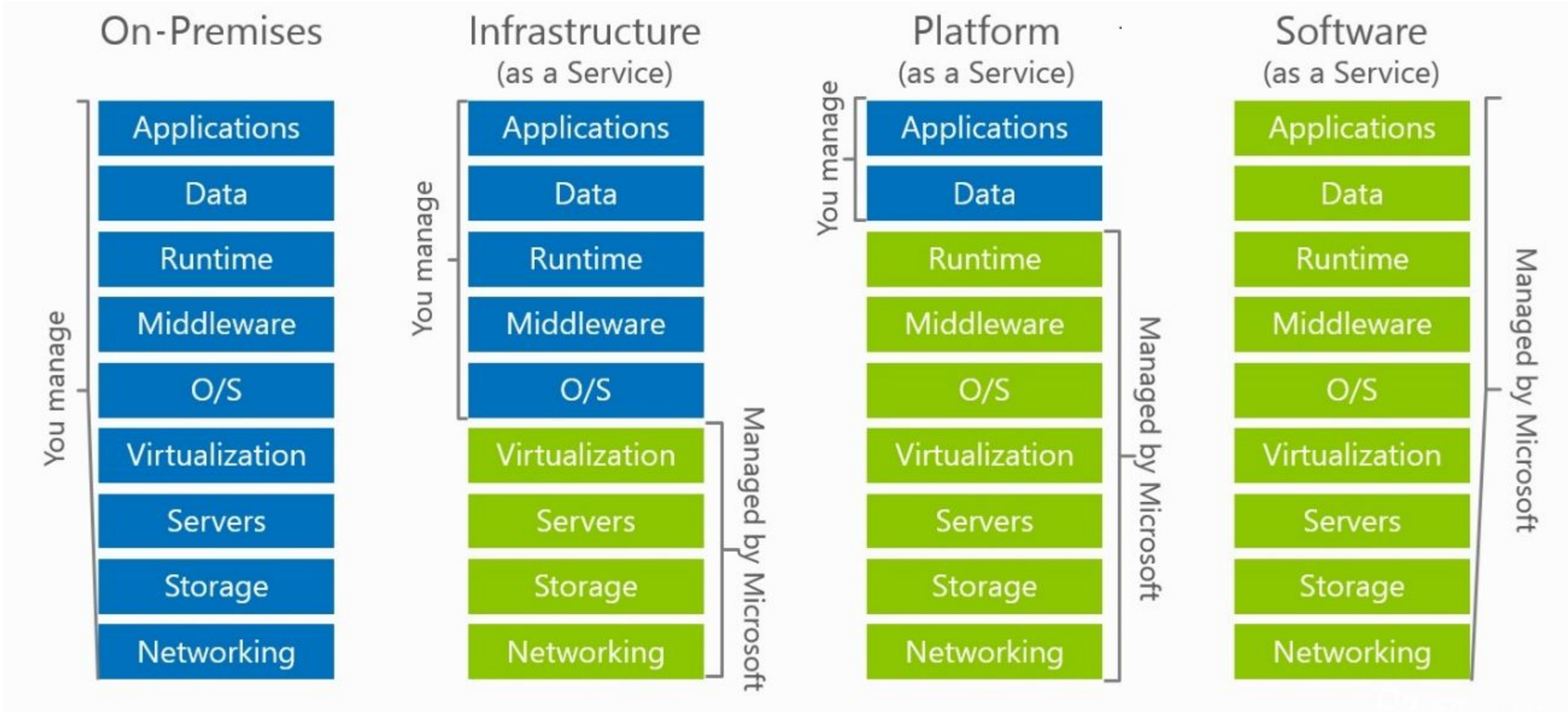
- “A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” – NIST definition

Service models

- IaaS: Infrastructure as a Service
  - Consumer can provision computing resources within provider's infrastructure upon which they can deploy and run arbitrary software, including OS and applications
- PaaS: Platform as a Service
  - Consumer can create custom applications using programming tools supported by the provider and deploy them onto the provider's cloud infrastructure
- SaaS: Software as a Service
  - Consumer uses provider’s applications running on provider's cloud infrastructure

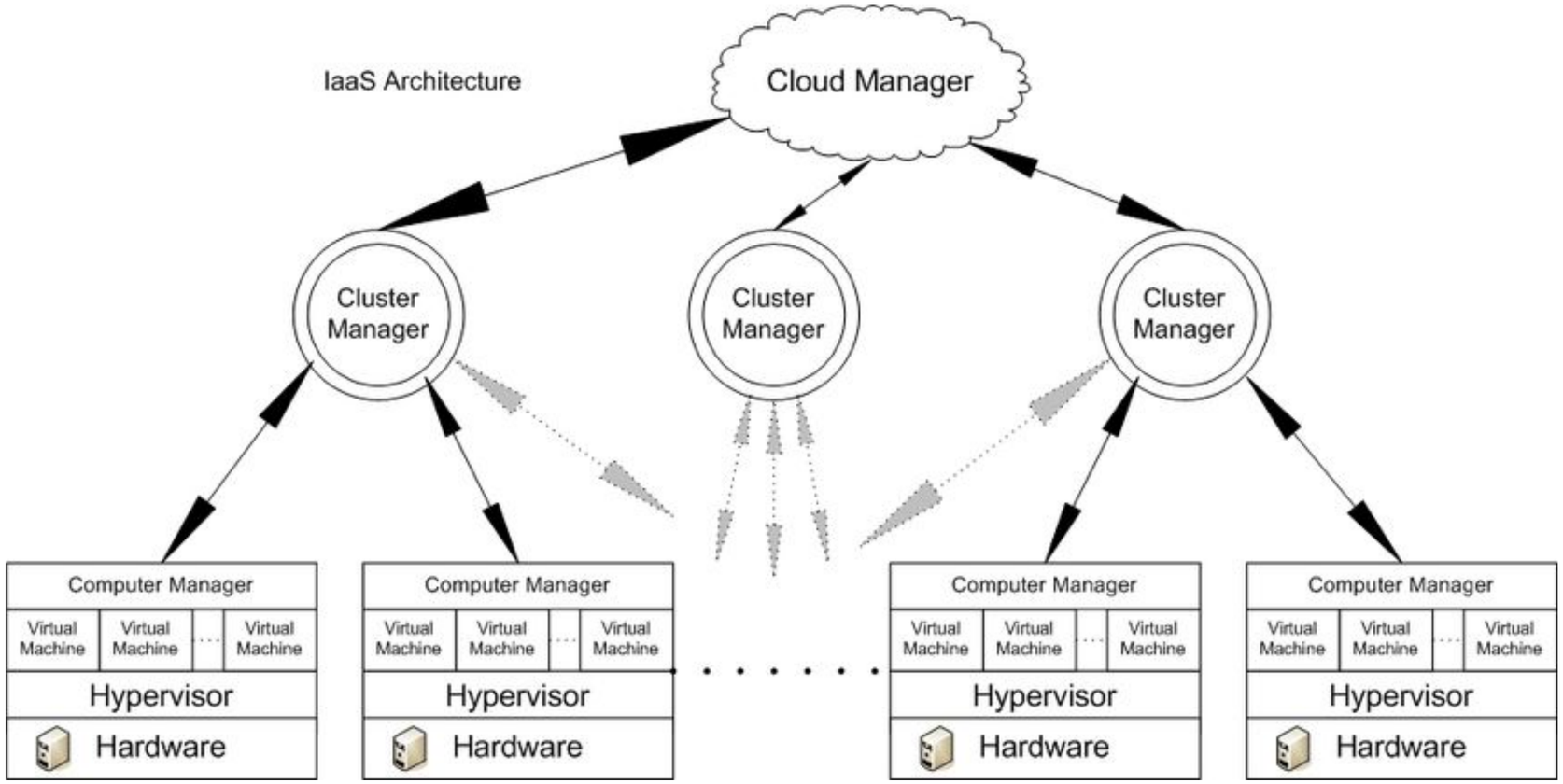
IaaS	PaaS	SaaS
• Virtual Machines • Virtual Networks	• Auto Elastic • Continuous Integration	• Built for Cloud • Uses PaaS

Service models

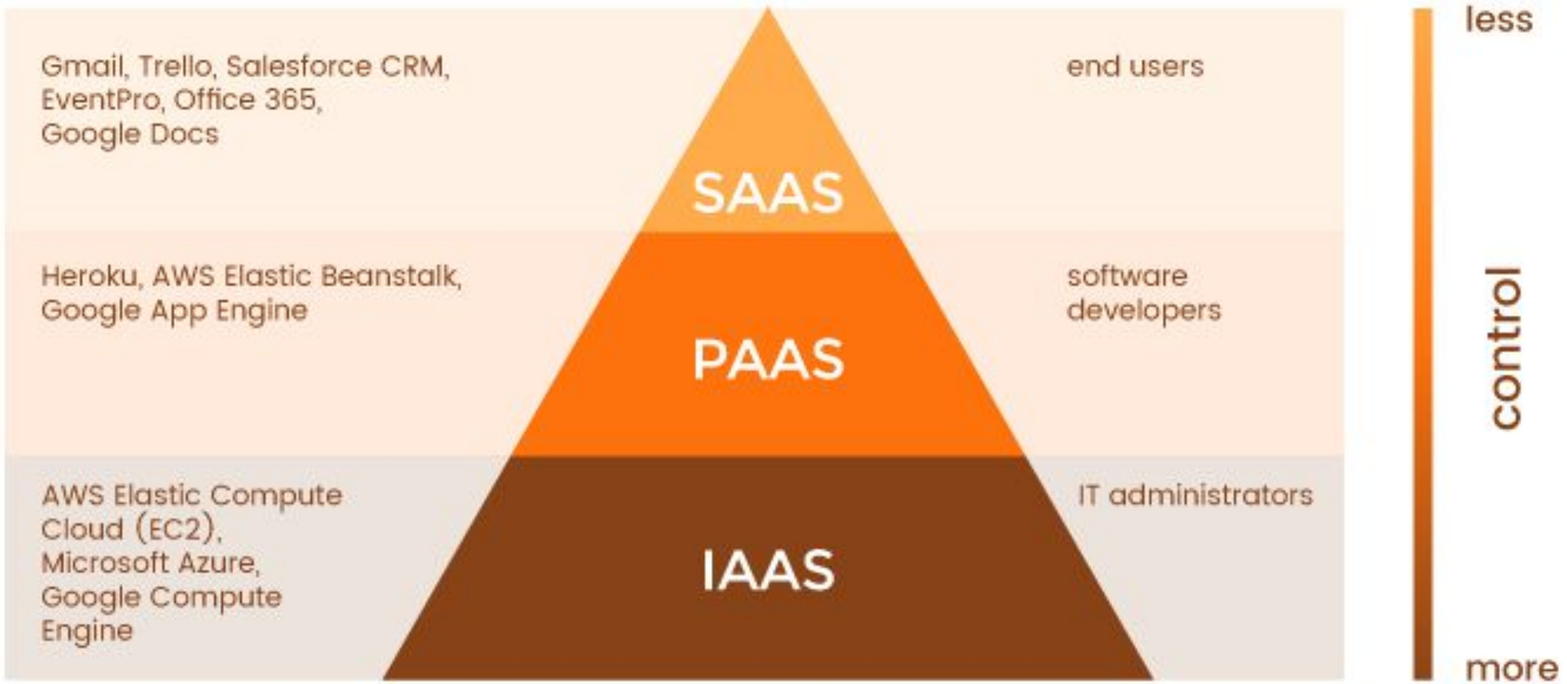


Credit: Microsoft Azure

IAAS cloud acrhitecture



Cloud providers



Credit: rubygarage

Further reading

- The Architecture of Virtual Machines

Thank You