# Cloud Computing Service Architectures

# Common cloud computing models

- **Service models:**
  - **IaaS**: Infrastructure as a Service
    - e.g., Amazon EC2, Microsoft Azure, Google Compute Engine
  - **PaaS**: Platform as a Service
    - e.g., Google AppEngine, Heroku, Apache Stratos
  - **SaaS**: Software as a Service
    - Microsoft Office365, Amazon DynamoDB, gmail

# Cloud infrastructure management

- **Management models**:
  - **Public clouds**
    - Utility model
    - Shared hardware
    - No control of hardware,
    - Self-managed (e.g., AWS, Azure, GCE)
  - **Private clouds**:
    - More isolated (secure?),
    - Federal compliance friendly
    - Customizable hardware and hardware sharing
    - Example: OpenStack, an open source platform for virtualized clouds

# Virtualization: key to cloud boom

- Physical machine virtualization
  - Full virtualization: masking as physical machine to guest
  - Para-virtualization: Semi-transparent to guest
- Virtualization goes well beyond CPU or physical machines
  - Virtualized I/O (Elastic Block Storage)
  - Virtual private networks
  - Containers (e.g., AWS or Google Container Service)
  - Virtual switches/routers with software-defined networking (SDN)
  - Network function virtualization (NFV) forwarding-plane services

# Cloud pricing models

- Market-driven: variety of pricing models for each of IaaS, PaaS, SaaS service models
- Let's ask some questions:
  - How much does an hour of computing on a commodity core cost today?
  - How much does transferring 1GB of data over a wide-area network cost today?
  - How much does 1GB of storage for a month cost today?

Let's find out what the market looks like today!

# EC2 compute pricing

| | vCPU | ECU | Memory (GiB) | Instance Storage (GB) | Linux/UNIX Usage |
|---|---|---|---|---|---|
| **General Purpose - Current Generation** | | | | | |
| t2.nano | 1 | Variable | 0.5 | EBS Only | $0.0065 per Hour |
| t2.micro | 1 | Variable | 1 | EBS Only | $0.013 per Hour |
| t2.small | 1 | Variable | 2 | EBS Only | $0.026 per Hour |
| t2.medium | 2 | Variable | 4 | EBS Only | $0.052 per Hour |
| t2.large | 2 | Variable | 8 | EBS Only | $0.104 per Hour |
| m4.large | 2 | 6.5 | 8 | EBS Only | $0.12 per Hour |
| m4.xlarge | 4 | 13 | 16 | EBS Only | $0.239 per Hour |
| m4.2xlarge | 8 | 26 | 32 | EBS Only | $0.479 per Hour |
| m4.4xlarge | 16 | 53.5 | 64 | EBS Only | $0.958 per Hour |

Do these numbers match our intuition based on market prices for computers and peripherals?

# Google Cloud compute pricing

| Machine type | Virtual CPUs | Memory | GCEU[1] | Lowest price[2] (USD) per hour with full sustained usage | Typical price[3] (USD) per hour | Full price[4] (USD) per hour without sustained use | Preemptible price[5] (USD) per hour |
|---|---|---|---|---|---|---|---|
| n1-standard-1 | 1 | 3.75GB | 2.75 | $0.035 | $0.038 | $0.050 | $0.015 |
| n1-standard-2 | 2 | 7.5GB | 5.50 | $0.070 | $0.076 | $0.100 | $0.030 |
| n1-standard-4 | 4 | 15GB | 11 | $0.140 | $0.152 | $0.200 | $0.060 |
| n1-standard-8 | 8 | 30GB | 22 | $0.280 | $0.304 | $0.400 | $0.120 |
| n1-standard-16 | 16 | 60GB | 44 | $0.560 | $0.608 | $0.800 | $0.240 |
| n1-standard-32[6] | 32 | 120GB | 88 | $1.120 | $1.216 | $1.600 | $0.480 |

# Spot pricing: market inside a cloud

| | Linux/UNIX Usage | Windows Usage |
|---|---|---|
| **General Purpose - Current Generation** | | |
| m3.medium | $0.0098 per Hour | $0.0591 per Hour |
| m3.large | $0.0196 per Hour | $0.1171 per Hour |
| m3.xlarge | $0.0379 per Hour | $0.1382 per Hour |
| m3.2xlarge | $0.075 per Hour | $0.2752 per Hour |
| m4.large | $0.027 per Hour | $0.1391 per Hour |
| m4.xlarge | $0.0468 per Hour | $0.2785 per Hour |
| m4.2xlarge | $0.0974 per Hour | $0.5557 per Hour |
| m4.4xlarge | $0.201 per Hour | $1.1156 per Hour |
| m4.10xlarge | $0.5948 per Hour | $2.9799 per Hour |

# Cloud Computing

# Infrastructure as a Service

ELECTIVE 3

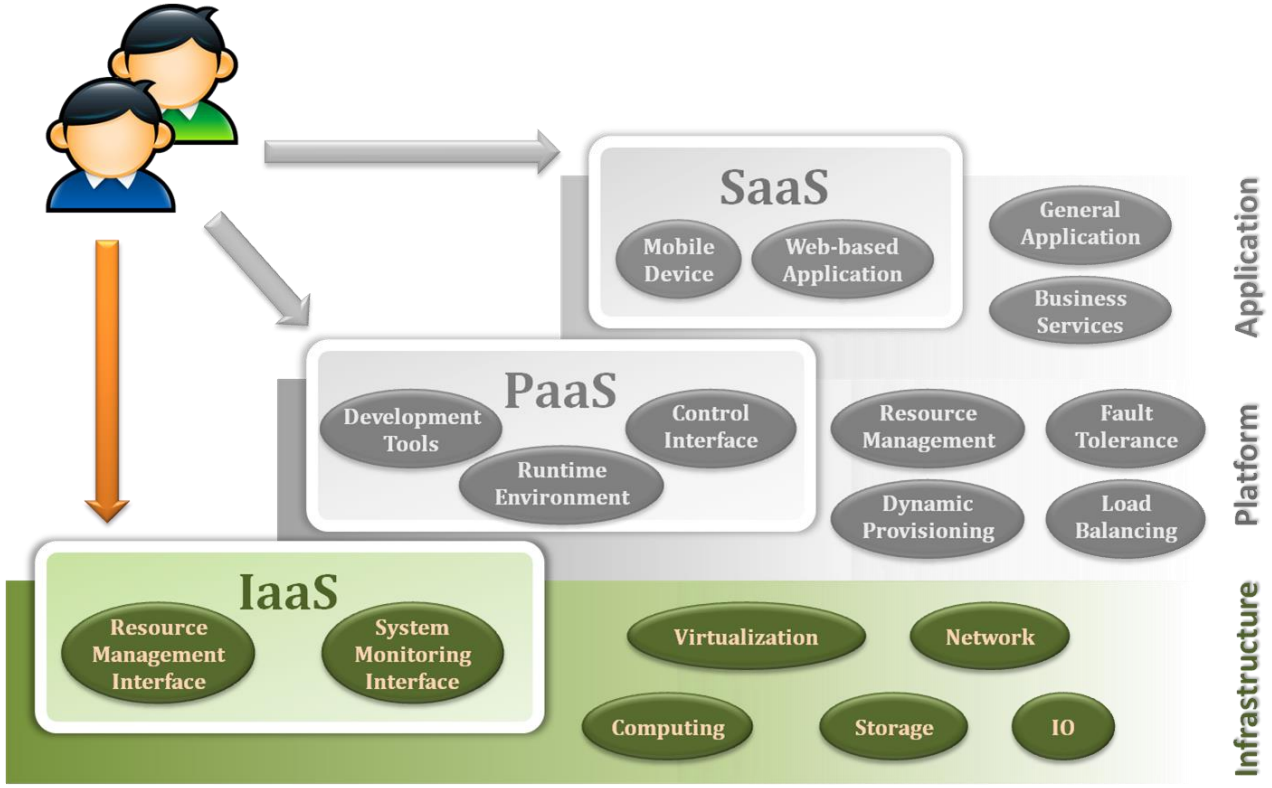
# Agenda

- Overview
  - Why do we need IaaS ?
  - How IaaS meets cloud properties ?

- Enabling Techniques
  - Virtualization Overview
  - Terminology & Taxonomy

# Overview

Why do we need IaaS ?



# Overview

- What is the problems in conventional case ?
  - Companies IT investment for peak capacity
  - Lack of agility for IT infrastructure
  - IT maintain cost for every company
  - Usually suffered from hardware failure risk
  - …etc

- These IT complexities force company back !!

# Overview

- How to solve these problem ?
  - Let’s consider some kind of out-sourcing solution
    - Somebody will handle on demand capacity for me
    - Somebody will handle high available resource for me
    - Somebody will handle hardware management for me
    - Somebody will handle system performance for me
    - Somebody will …
  - Frankly, that would be a great solution IF there were “somebody”.
  - But who can be this “somebody”, and provide all these services ?

# Overview

- **Infrastructure as a Service** will be the salvation.
  - IaaS cloud provider takes care of all the IT infrastructure complexities.
  - IaaS cloud provider provides all the infrastructure functionalities.
  - IaaS cloud provider guarantees qualified infrastructure services.
  - IaaS cloud provider charges clients according to their resource usage.

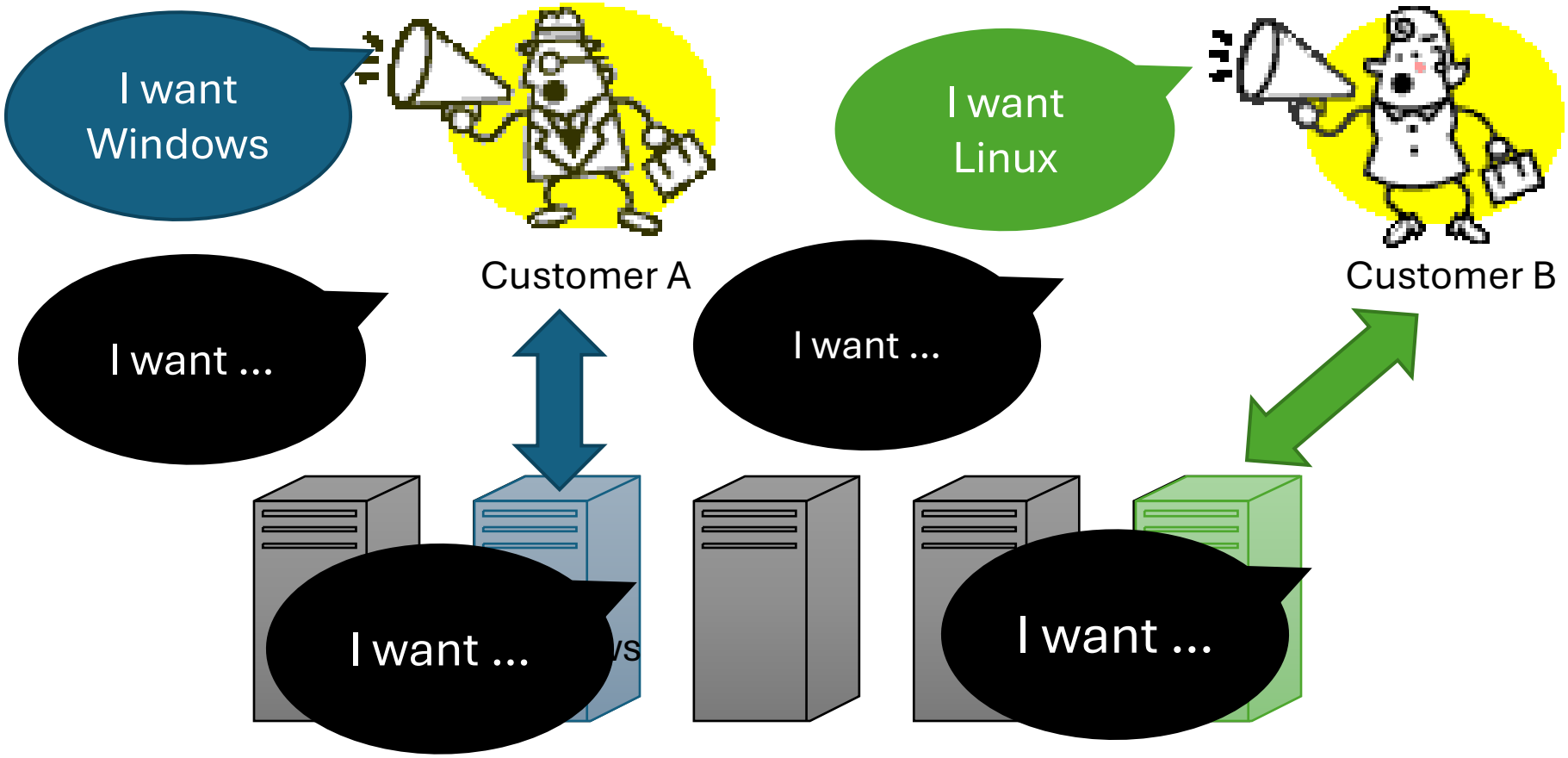
- But, what make all of these happen so magically ?

# Virtualization

- Assume that you are going to be an IaaS cloud provider.
  - Then, what are the problems you are facing ?
    - Clients will request different operating systems.
    - Clients will request different storage sizes.
    - Clients will request different network bandwidths.
    - Clients will change their requests anytime.
    - Clients will …
  - Is there any good strategy ?
    - Allocate a new physical machine for each incomer.
    - Prepare a pool of pre-installed machines for different requests.
    - or …

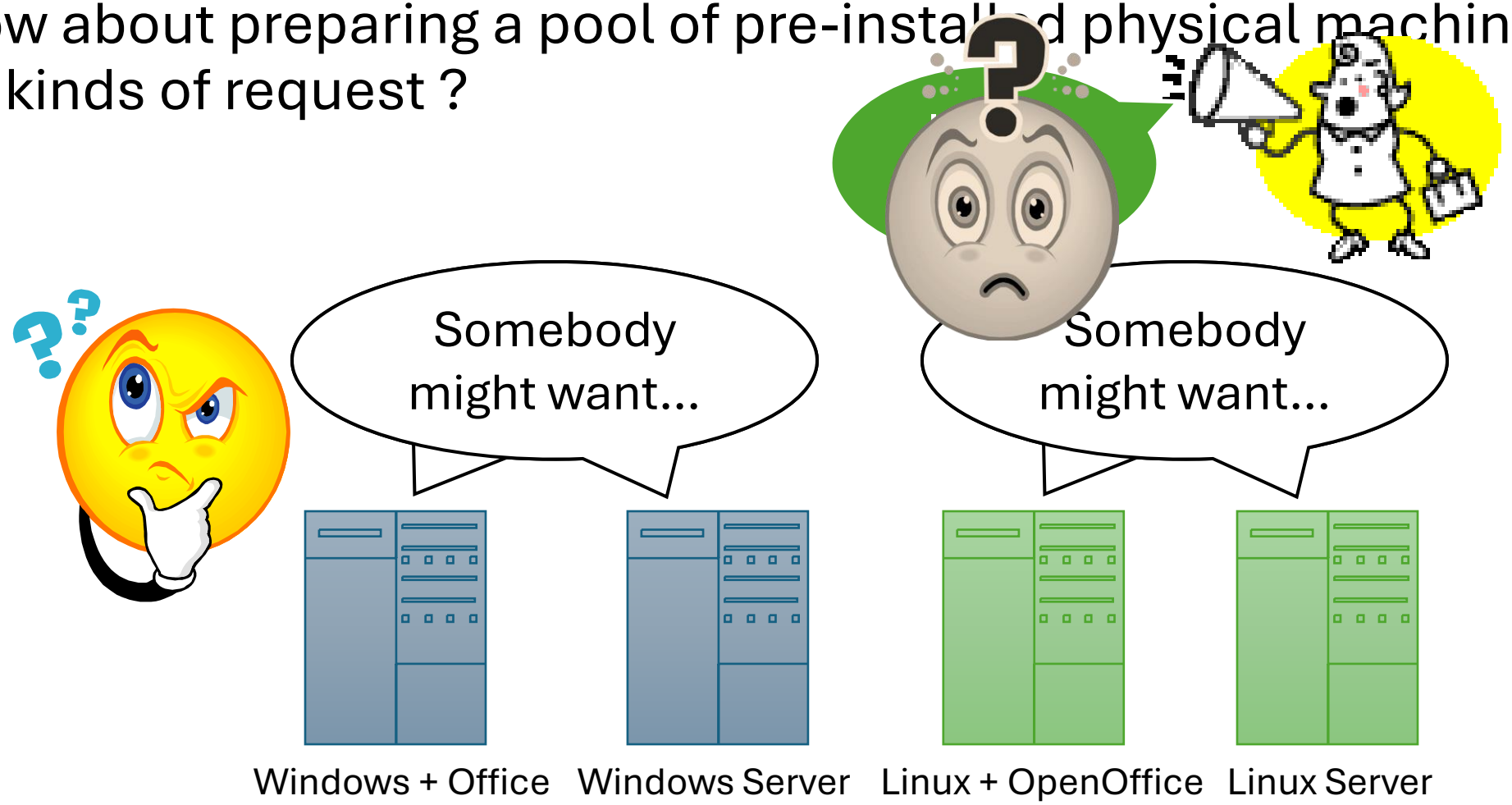
# Virtualization

- What if we allocate a new physical machine for each incomer ?



# Virtualization

- How about preparing a pool of pre-installed physical machines for all kinds of request ?



# Virtualization

- Obviously, neither of previous strategies will work.
- We need more powerful techniques to deal with that.

- Virtualization techniques will help.
  - For computation resources
    - Virtual Machine technique
  - For storage resources
    - Virtual Storage technique
  - For communication resources
    - Virtual Network technique

# Overview

Why do we need IaaS ?

How IaaS meets cloud properties ?

# Properties and Characteristics

- As a cloud provider, all of the fundamental properties and characteristics stated in previous lectures should be concerned and implemented.


# Scalability & Elasticity

- What do scalability and elasticity mean in IaaS ?
  - Clients should be able to dynamically increase or decrease the amount of infrastructure resources in need.
  - Large amount of resources provisioning and deployment should be done in a short period of time, such as several hours or days.
  - System behavior should remain identical in small scale or large one.


# Scalability & Elasticity

- How to approach scalability and elasticity in IaaS ?
  - For computation resources :
    - Dynamically create or terminate virtual machines for clients on demand.
    - Integrate hypervisors among all physical machines to collaboratively control and manage all virtual machines.
  - For storage resources :
    - Dynamically allocate or de-allocate virtual storage space for clients.
    - Integrate all physical storage resources in the entire IaaS system
    - Offer initial storage resources by thin provisioning technique.
  - For communication resources :
    - Dynamically connect or disconnect the linking state of virtual networks for clients on demand.
    - Dynamically divide the network request flow to different physical routers to maintain access bandwidth.


# Availability & Reliability

- What do availability and reliability mean in IaaS ?
  - Clients should be able to access computation resources without considering the possibility of hardware failure.
  - Data stored in IaaS cloud should be able to be retrieved when needed without considering any natural disaster damage.
  - Communication capability and capacity should be maintained without considering any physical equipment shortage.


# Availability & Reliability

- How to approach availability and reliability in IaaS ?
  - For computation resources :
    - Monitor each physical and virtual machine for any possible failure.
    - Regularly backup virtual machine system state for disaster recovery.
    - Migrate virtual machine among physical machines for potential failure prevention.
  - For storage resources :
    - Maintain data pieces replication among different physical storage devices.
    - Regularly backup virtual storage data to geographical remote locations for disaster prevention.
  - For communication resources :
    - Built redundant connection system to improve robustness.


# Manageability & Interoperability

- What do manageability and interoperability mean in IaaS ?
  - Clients should be able to fully control the virtualized infrastructure resources which allocated to them.
  - Virtualized resources can be allocated by means of system control automation process with pre-configured policy.
  - States of all virtualized resource should be fully under monitoring.
  - Usage of infrastructure resources will be recorded and then billing system will convert these information to user payment.


# Manageability & Interoperability

- How to approach manageability and interoperability in IaaS ?
  - For computation resources :
    - Provide basic virtual machine operations, such as creation, termination, suspension, resumption and system snapshot.
    - Monitor and record CPU and memory loading for each virtual machine.
  - For storage resources :
    - Monitor and record storage space usage and read/write data access from user for each virtual storage resource.
    - Automatic allocate/de-allocate physical storage according to space utilization.
  - For communication resources :
    - Monitor and record the network bandwidth consumption for each virtual link.
    - Automatically reroute the data path when computation and storage are duplicated.


# Performance & Optimization

- What do performance and optimization mean in IaaS ?
  - Physical resources should be highly utilized among different clients.
  - Physical resources should form a large resource pool which provide high computing power through parallel processing.
  - Virtual infrastructure resources will be dynamically configured to an optimized deployment among physical resources.


# Performance & Optimization

- How to approach performance and optimization in IaaS ?
  - For computation resources :
    - Deploy virtual machine with load balancing consideration.
    - Live migrate virtual machines among physical ones to balance the system loading.
  - For storage resources :
    - Deploy virtual storage with hot spot access consideration.
    - Live migrate virtual storage among physical ones with different performance level.
  - For communication resources :
    - Consider network bandwidth loading when deploying virtual machines and storage.
    - Dynamically migrate virtual machines or storage to balance network flow.


# Accessibility & Portability

- What do accessibility and portability mean in IaaS ?
  - Clients should be able to control, manage and access infrastructure resources in an easy way, such as the web-browser, without additional local software or hardware installation.
  - Provided infrastructure resources should be able to be reallocated or duplicated easily.


# Accessibility & Portability

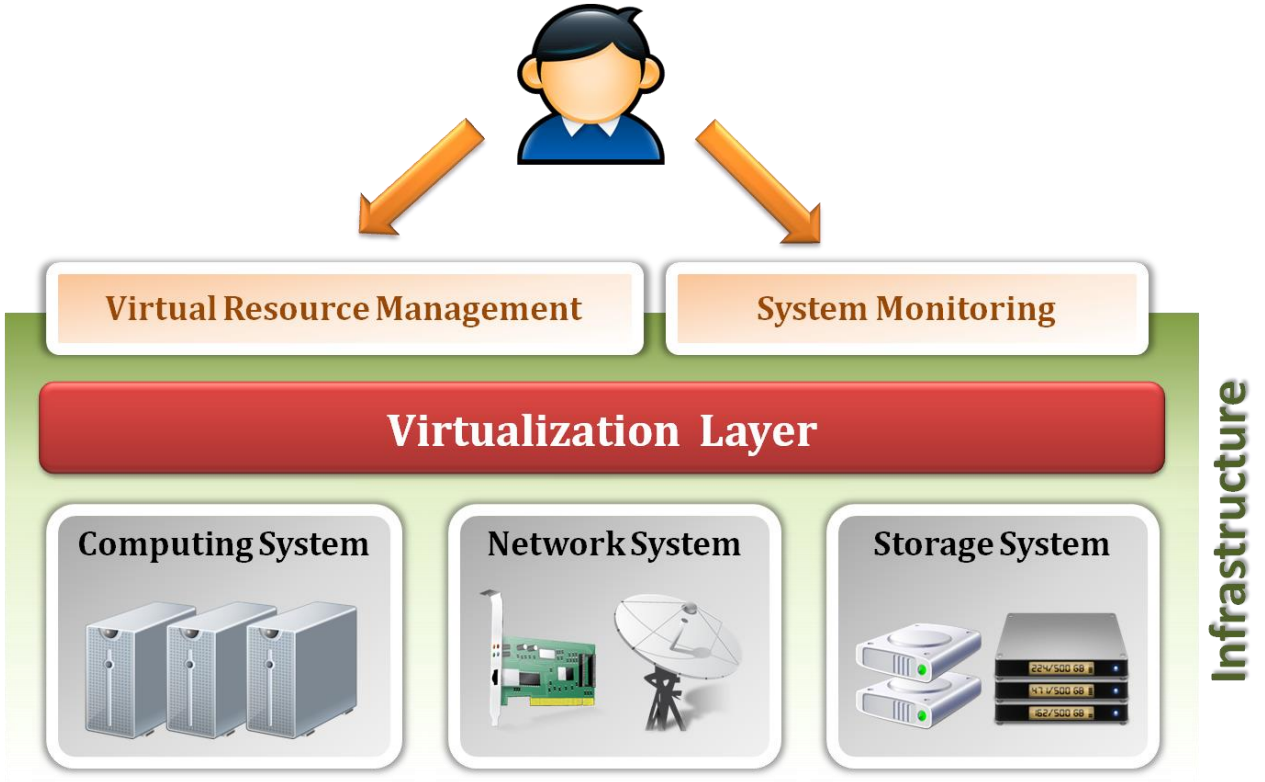
- How to approach accessibility and portability in IaaS ?
  - For computation resources :
    - Cloud provider integrates virtual machine management and access through web-based portal.
    - Comply the virtual machine standard for portability.
  - For storage resources :
    - Cloud provider integrates virtual storage management and access through web-based portal.
  - For communication resources :
    - Cloud provider integrates virtual network management and access through web-based portal.

# Enabling Techniques

Virtualization

# IaaS Architecture

- **Infrastructure as a Service (IaaS)** delivers computer infrastructure for cloud user, typically a platform virtualization environment as a service.
- **Virtualization** is an enabling technique to provide an abstraction of logical resources away from underlying physical resources.



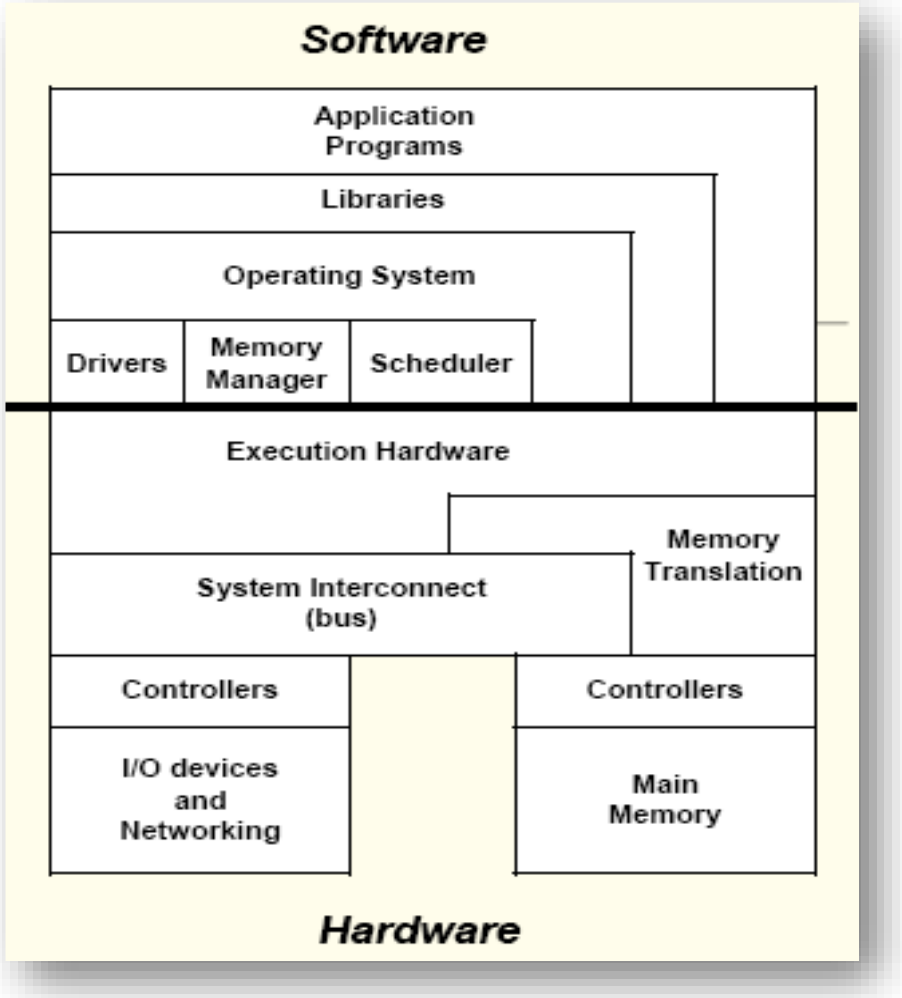
# Virtualization Overview

- What is virtualization ?
  - Virtualization is the creation of a virtual (rather than physical) version of something, such as an operating system, a server, a storage device or network resources.
  - It hides the physical characteristics of a resource from users, instead showing another abstract resource.

- But, where does virtualization come from ?
  - Virtualization is NOT a new idea of computer science.
  - Virtualization concept comes from the component abstraction of system design, and it has been adapted in many system level.
  - Now, let’s take a look of our original system architecture !!

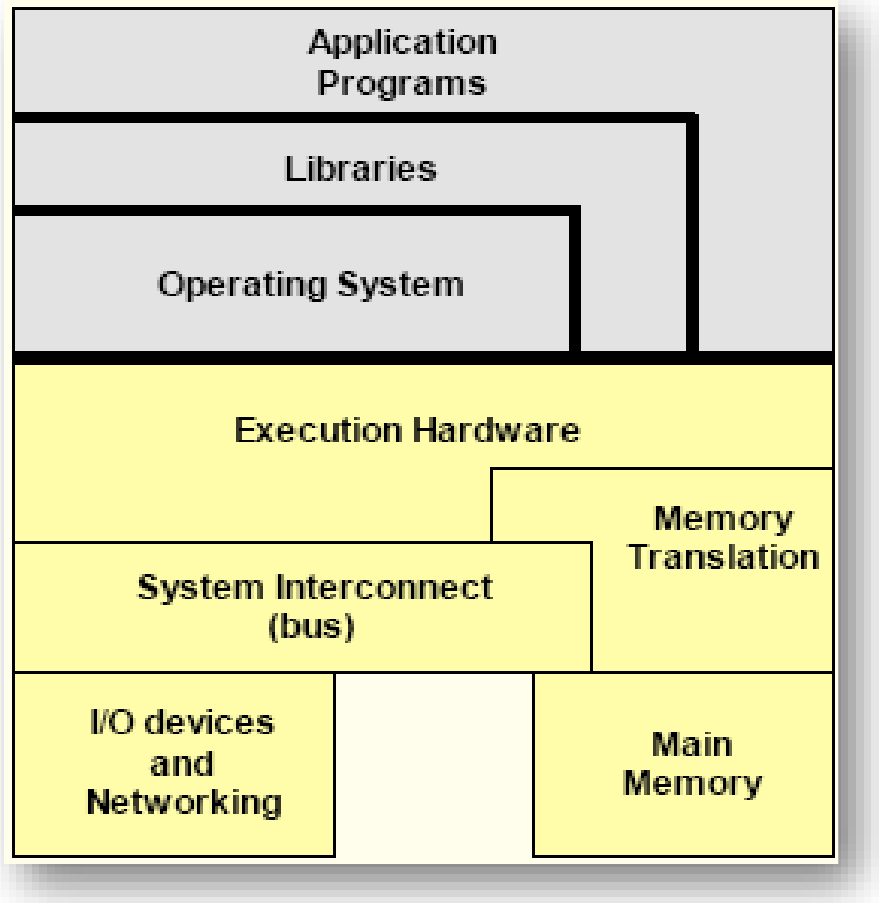
# Virtualization Overview

- System abstraction :
  - Computer systems are built on levels of abstraction.
  - Higher level of abstraction hide details at lower levels.
  - Designer of each abstraction level make use of the functions supported from its lower level, and provide another abstraction to its higher one.
  - Example
    - files are an abstraction of a disk



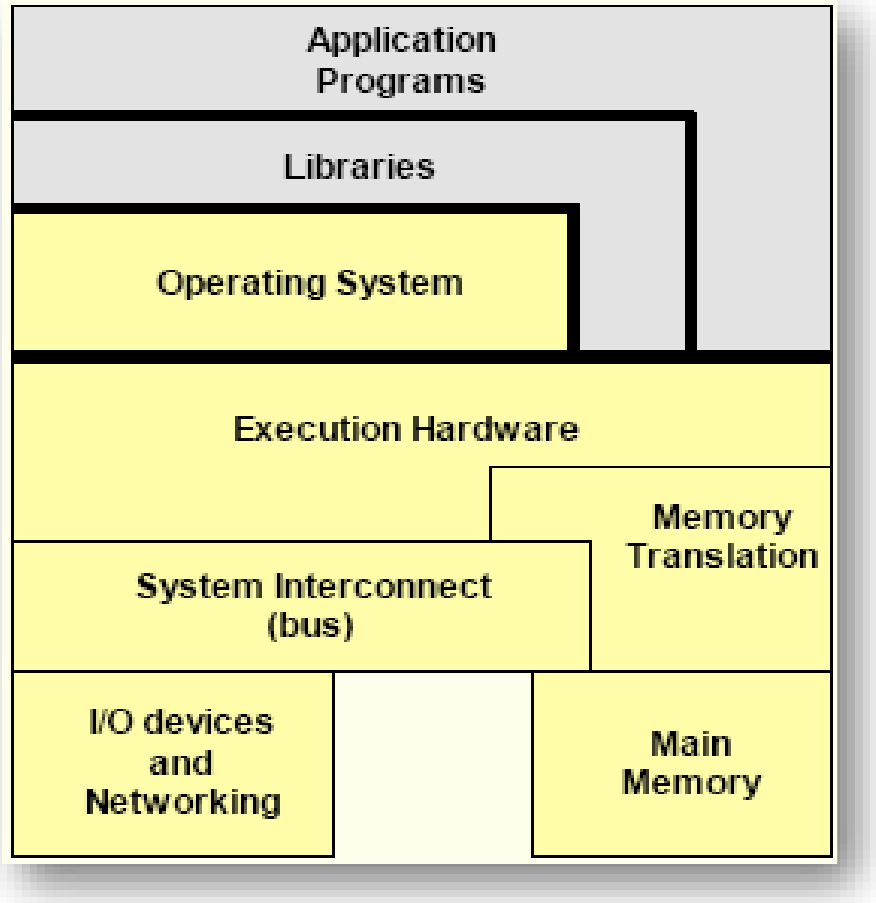
# Virtualization Overview

- Machine level abstraction :
  - For OS developers, a machine is defined by ISA (Instruction Set Architecture).
  - This is the major division between hardware and software.
  - Examples :
    - X86
    - ARM
    - MIPS



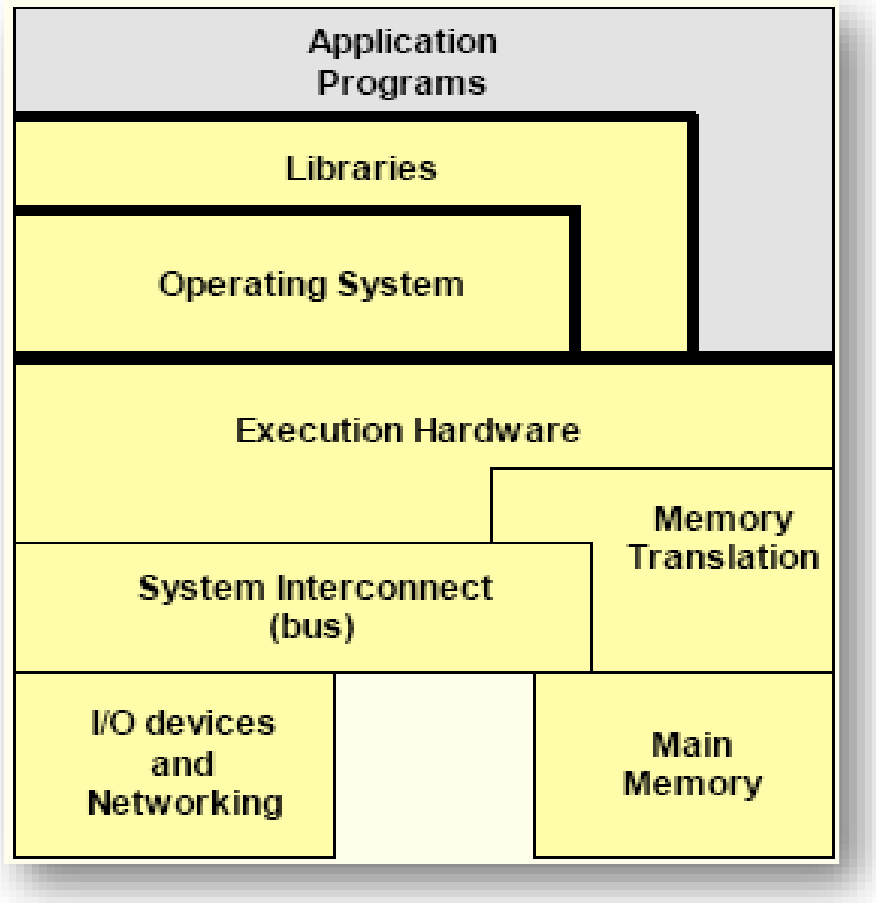
# Virtualization Overview

- OS level abstraction :
  - For compiler or library developers, a machine is defined by ABI (Application Binary Interface).
  - This define the basic OS interface which may be used by libraries or user.
  - Examples :
    - User ISA
    - OS system call



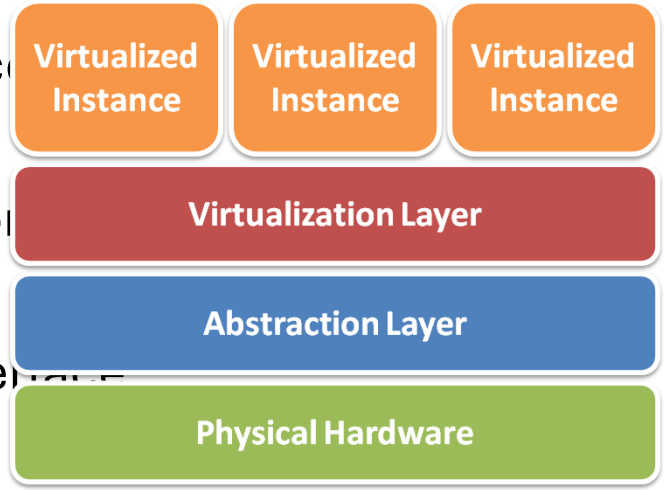
# Virtualization Overview

- Library level abstraction :
  - For application developers, a machine is defined by API (Application Programming Interface).
  - This abstraction provides the well-rounded functionalities.
  - Examples :
    - User ISA
    - Standard C library
    - Graphical library



# Virtualization Overview

- The concept of virtualization is **everywhere** !!
  - In IaaS, we focus the virtualization granularity at each physical hardware device.
- General virtualization implementation level :
  - Virtualized instance
    - Software virtualized hardware instanc
  - Virtualization layer
    - Software virtualization implementatio
  - Abstraction layer
    - Various types of hardware access inte
  - Physical hardware
    - Various types of infrastructure resources
- Different physical resources :
  - Server, Storage and Network



# Virtualization

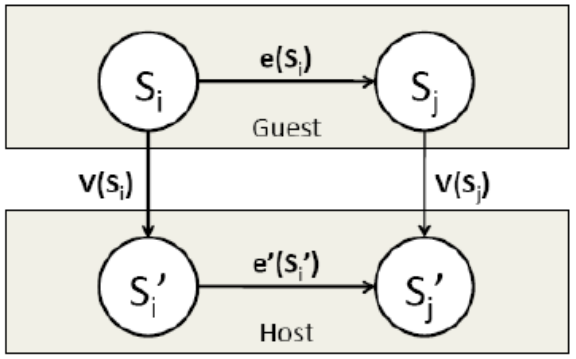
Terminology & Taxonomy

# Virtual Machine

- What is Virtual Machine (VM)?
  - **VM** is a software implementation of a machine (i.e. a computer) that executes programs like a real machine.

- Terminology :
  - Host (Target)
    - The primary environment where will be the target of virtualization.
  - Guest (Source)
    - The virtualized environment where will be the source of virtualization.

Isomorphism



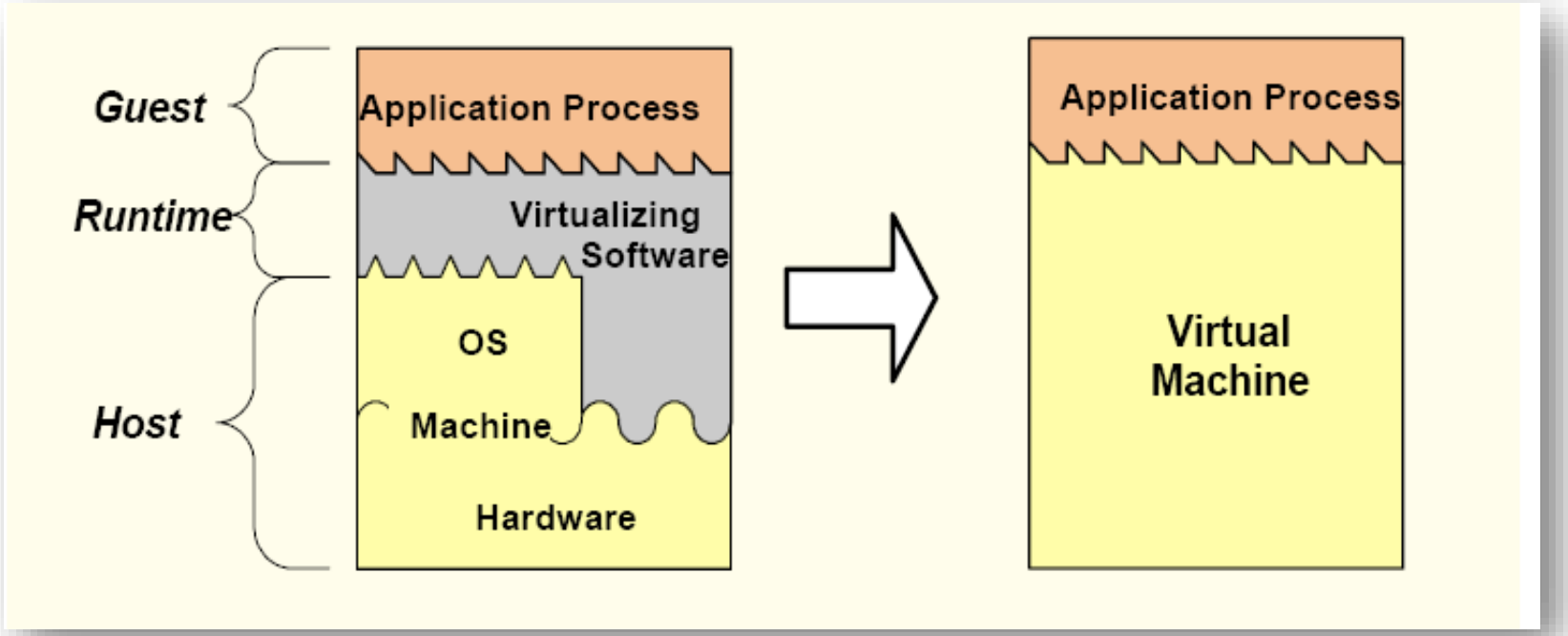

Formally, virtualization involves the construction of an **isomorphism** from **guest** state to **host** state.

# Emulation vs. Virtualization

- Emulation technique
  - Simulate an independent environment where guest ISA and host ISA are different.
  - Example
    - Emulate x86 architecture on ARM platform.
- Virtualization technique
  - Simulate an independent environment where guest ISA and host ISA are the same.
  - Example
    - Virtualize x86 architecture to multiple instances.

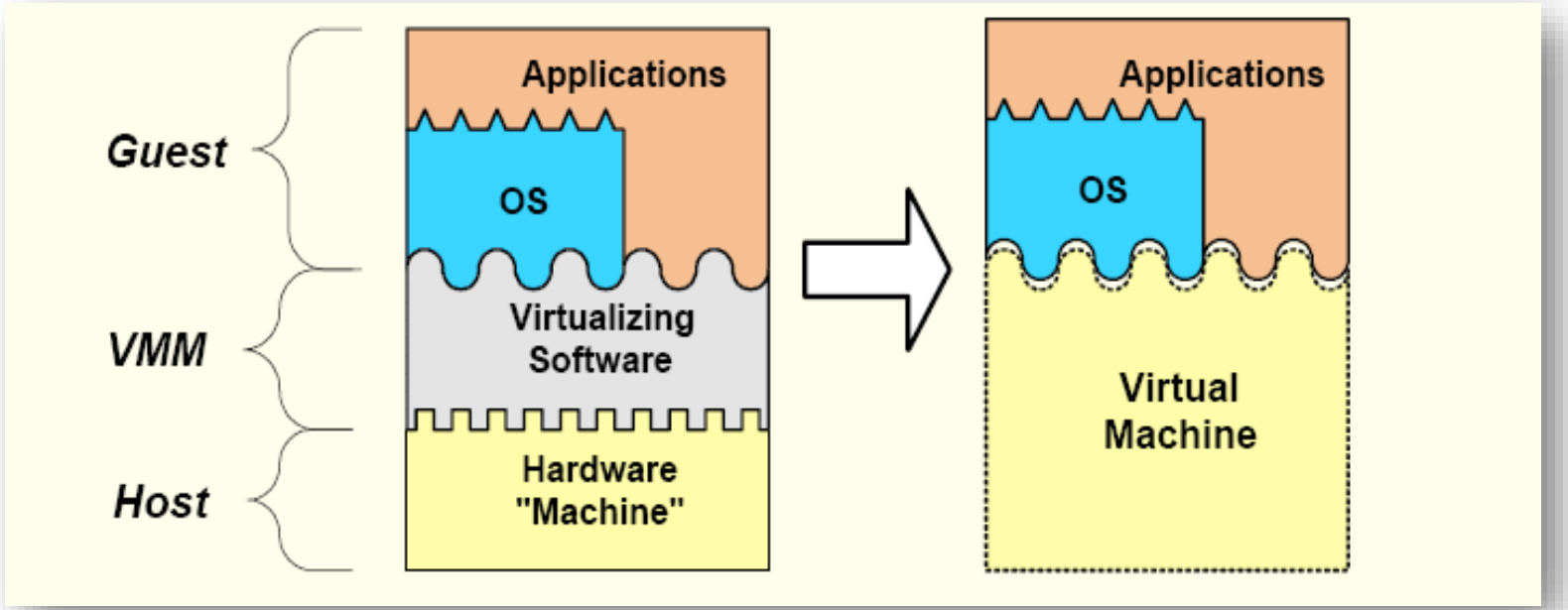
# Process Virtual Machine

- Process virtual machine
  - Usually execute guest applications with an ISA different from host
  - Couple at ABI(Application Binary Interface) level via runtime system
  - Not persistent



# System Virtual Machine

- System virtual machine
  - Provide the entire operating system on same or different host ISA
  - Constructed at ISA level
  - Persistent



# Taxonomy

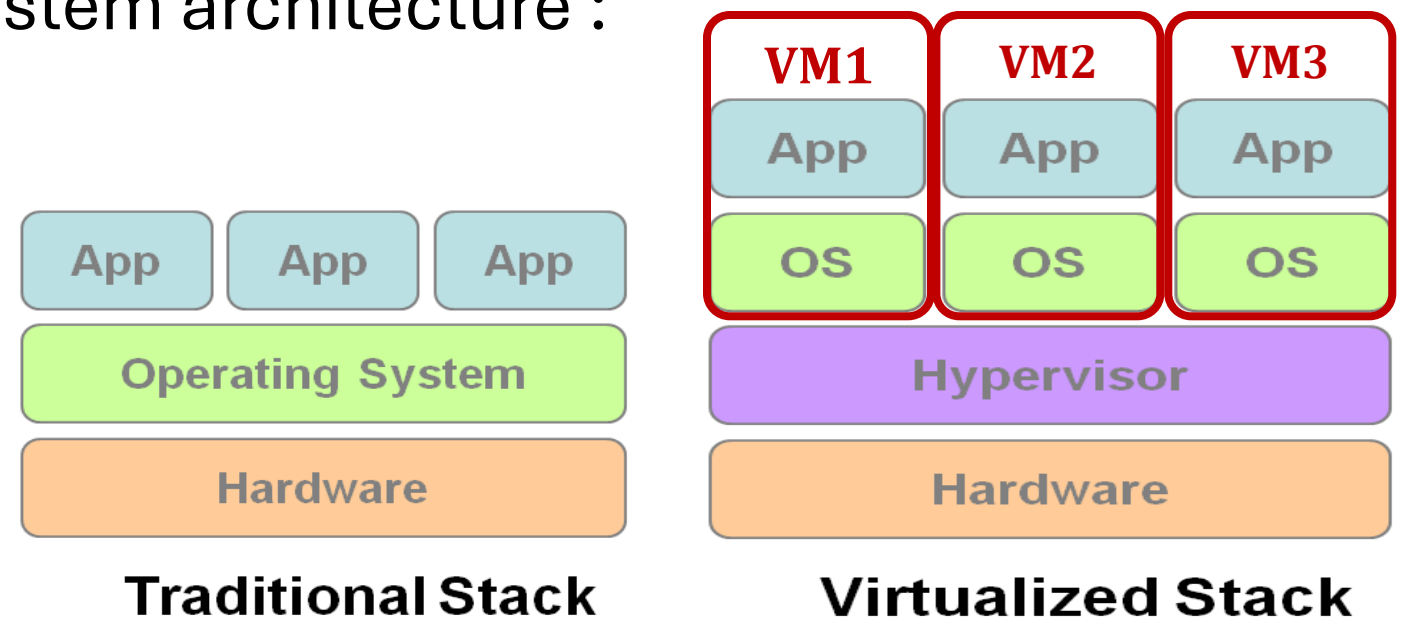
| | System Virtual Machine | Process Virtual Machine |
|---|---|---|
| ***Emulation*** | ***Transmeta Crusoe***<br>*( Emulate x86 on VLIW cpu )* | ***Multi-processing system*** |
| ***Virtualization*** | ***XEN, KVM, VMWare***<br>*( x86 virtualization software )* | ***JVM, Microsoft CLI***<br>*( High level language virtualization )* |

**Techniques utilized in IaaS**

# Virtual Machine Monitor

- What's Virtual Machine Monitor (VMM) ?
  - **VMM** or **Hypervisor** is the software layer providing the virtualization.

- System architecture :



# Virtualization Types

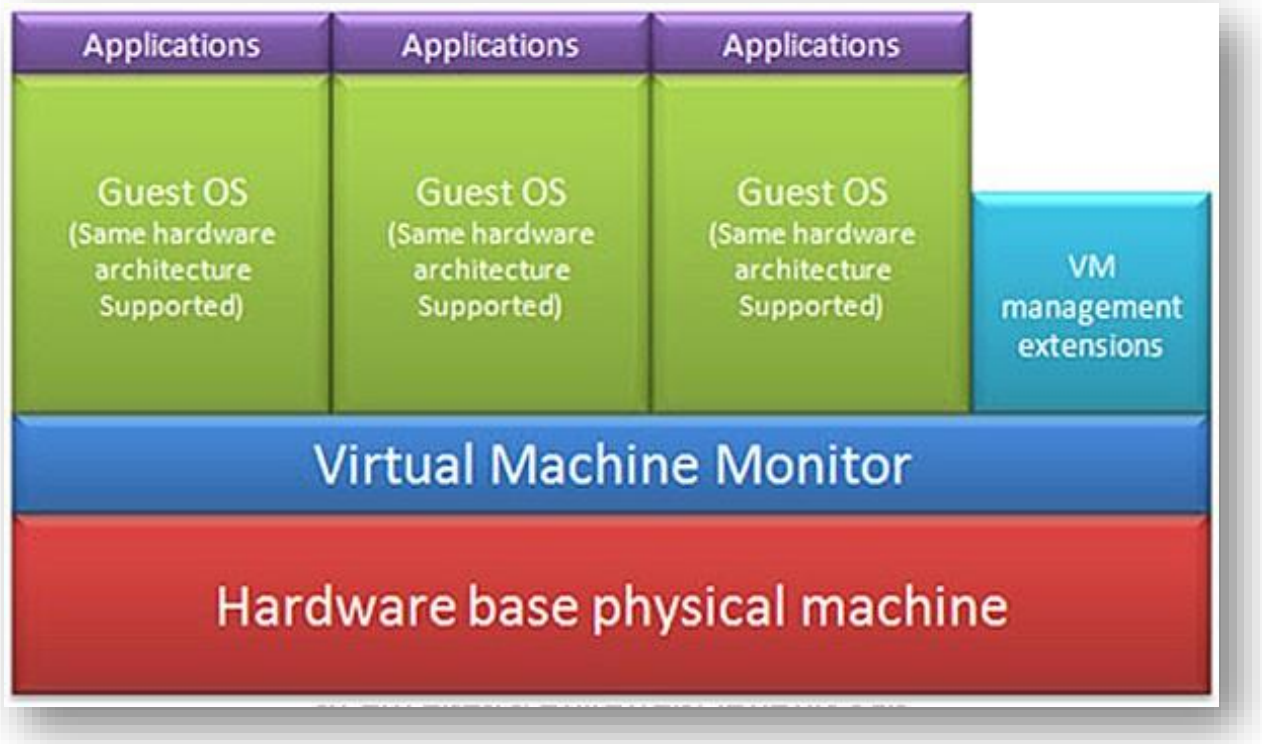
- Virtualization Types :
  - Type 1 – Bare metal
    - VMMs run directly on the host's hardware as a hardware control and guest operating system monitor.
  - Type 2 – Hosted
    - VMMs are software applications running within a conventional operating system.

# Virtualization Approaches

- Virtualization Approaches :
  - Full-Virtualization
    - VMM simulates enough hardware to allow an unmodified guest OS.
  - Para-Virtualization
    - VMM does not necessarily simulate hardware, but instead offers a special API that can only be used by the modified guest OS.

# Virtualization Approaches

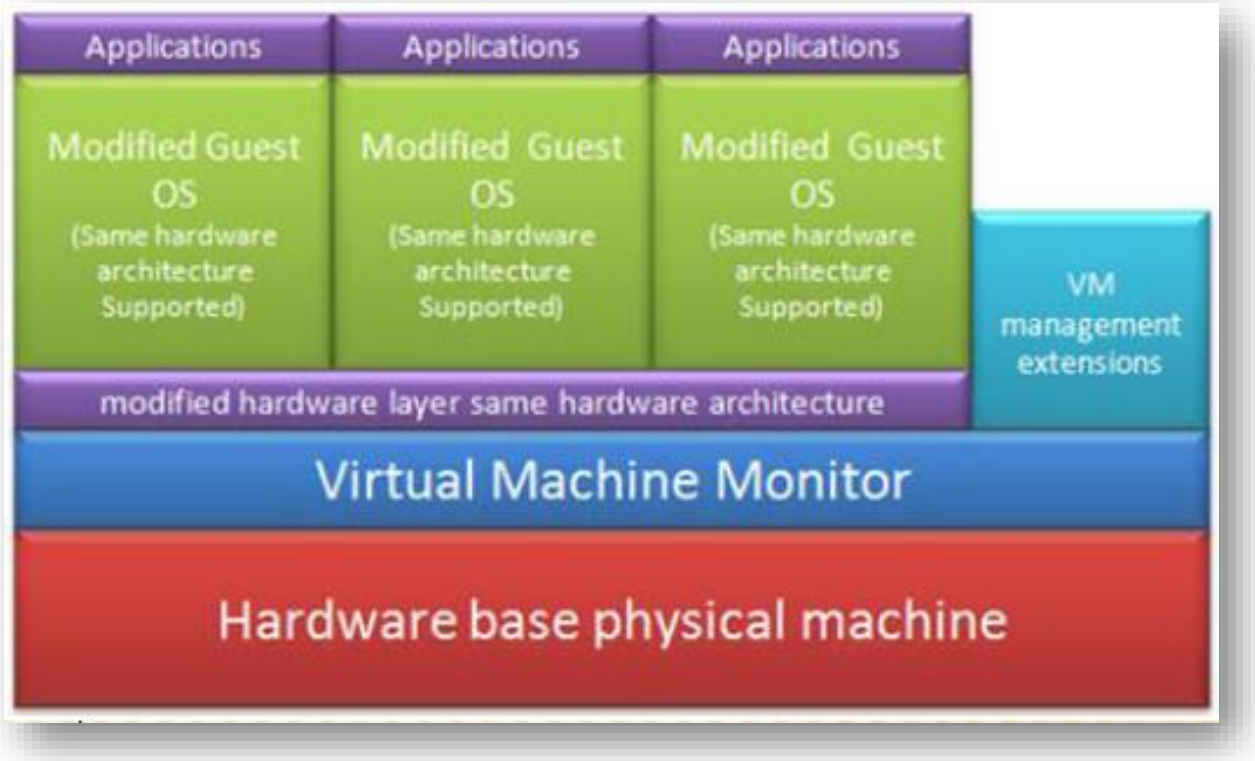
- Full-Virtualization




| | |
|---|---|
| ***Pros*** | **Need not to modify guest OS** |
| ***Cons*** | **Significant performance hit** |

# Virtualization Approaches

- Para-Virtualization




| | |
|---|---|
| ***Pros*** | **Light weight and high performance** |
| ***Cons*** | **Require modification of guest OS** |

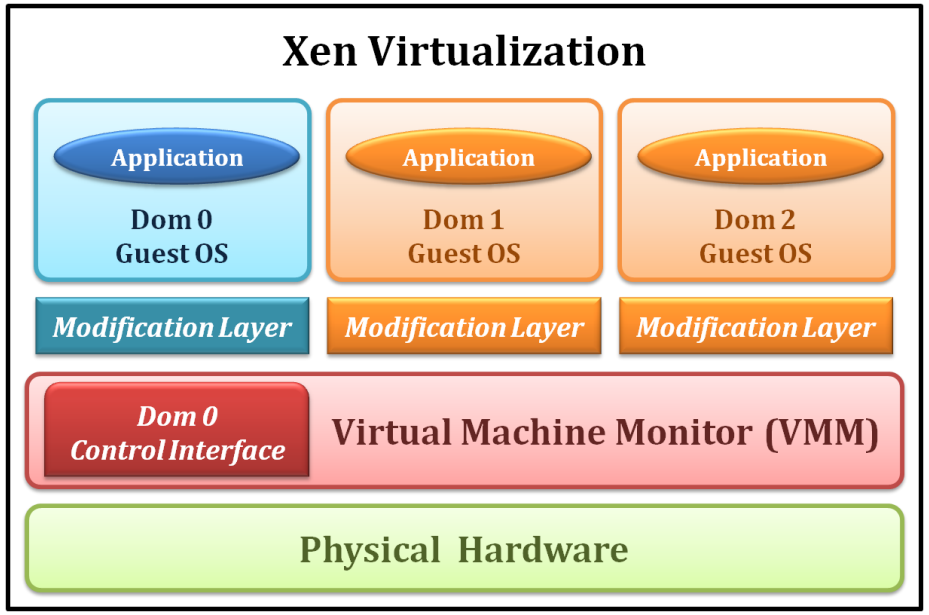
# Examples

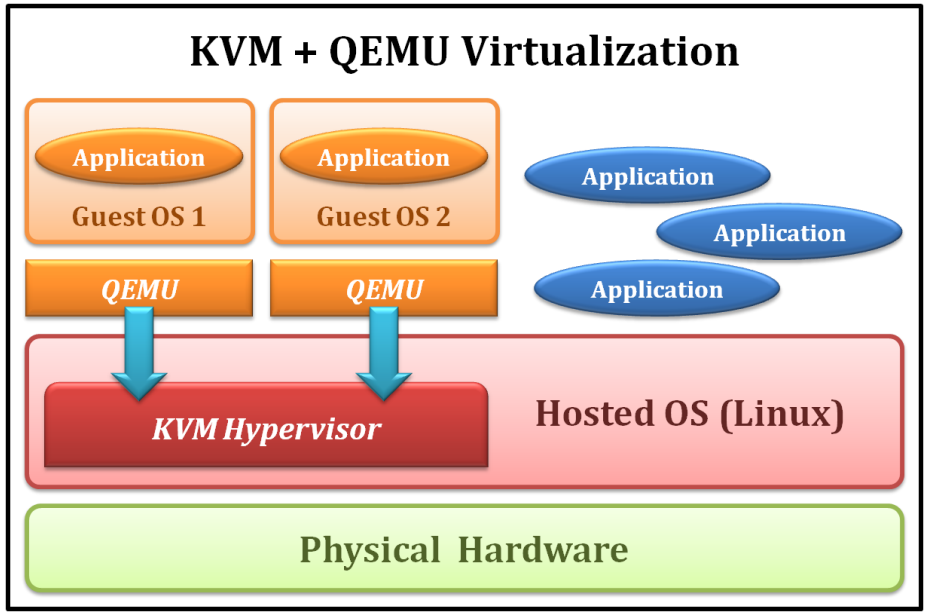
**Xen**

- Type 1 Virtualization
- Para-Virtualization

**KVM**

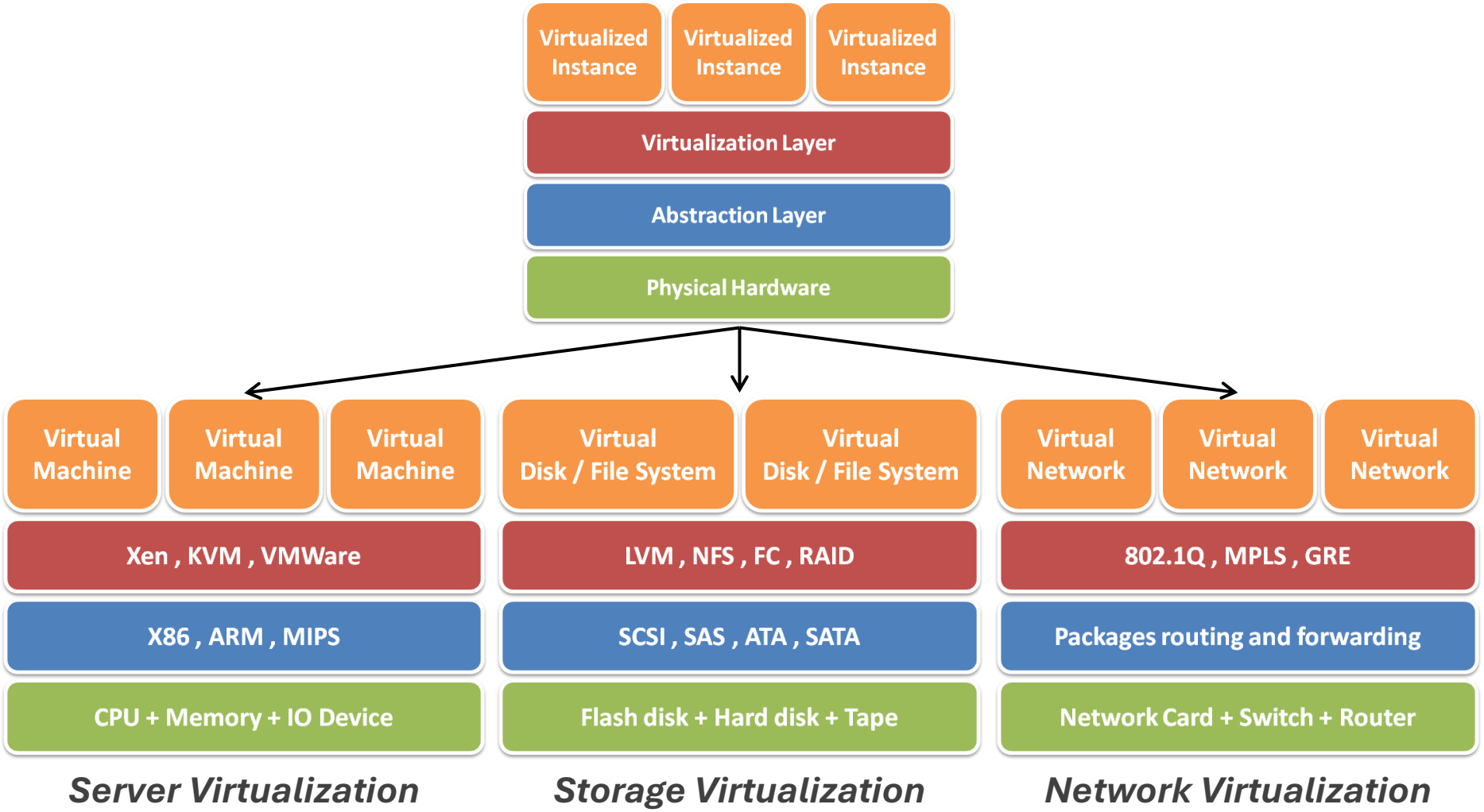
- Type 2 Virtualization
- Full-Virtualization



# To be continued...

- Virtualization techniques



# References


- James Smith and Ravi Nair, “Virtual Machines: Versatile Platforms for Systems and Processors”.
- Xen. http://www.xen.org/
- Kernel-based Virtual Machine (KVM). http://www.linux-kvm.org/page/Main_Page
- From Wikipedia, the free encyclopedia
- All resources of the materials and pictures were partially retrieved from the Internet.