Virtualization & Clouds

M. EL AOUN

Associate Professor, ESP

November 30, 2024

Outline

Introduction

Virtualization

Clouds

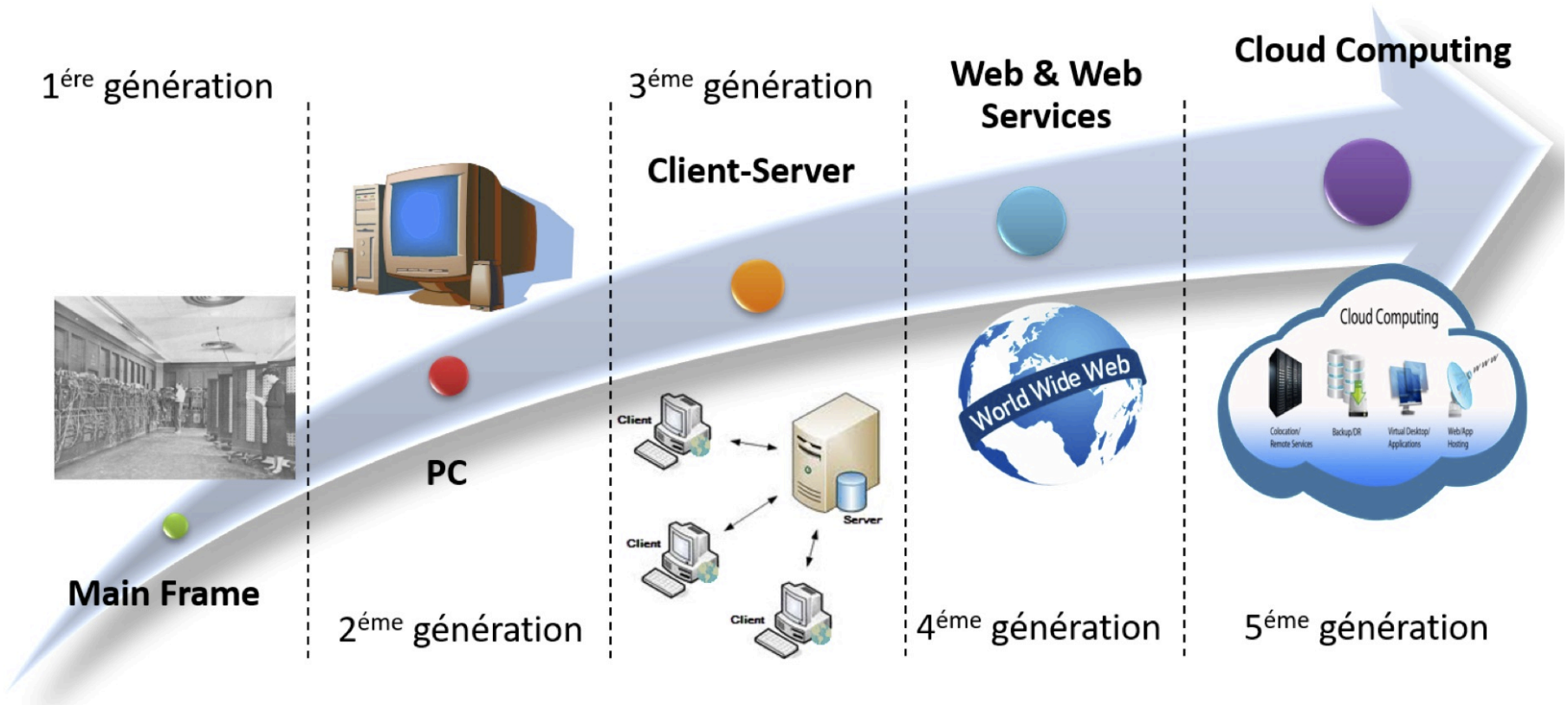
Cloud services

OpenStack

Docker

Conclusion

Introduction

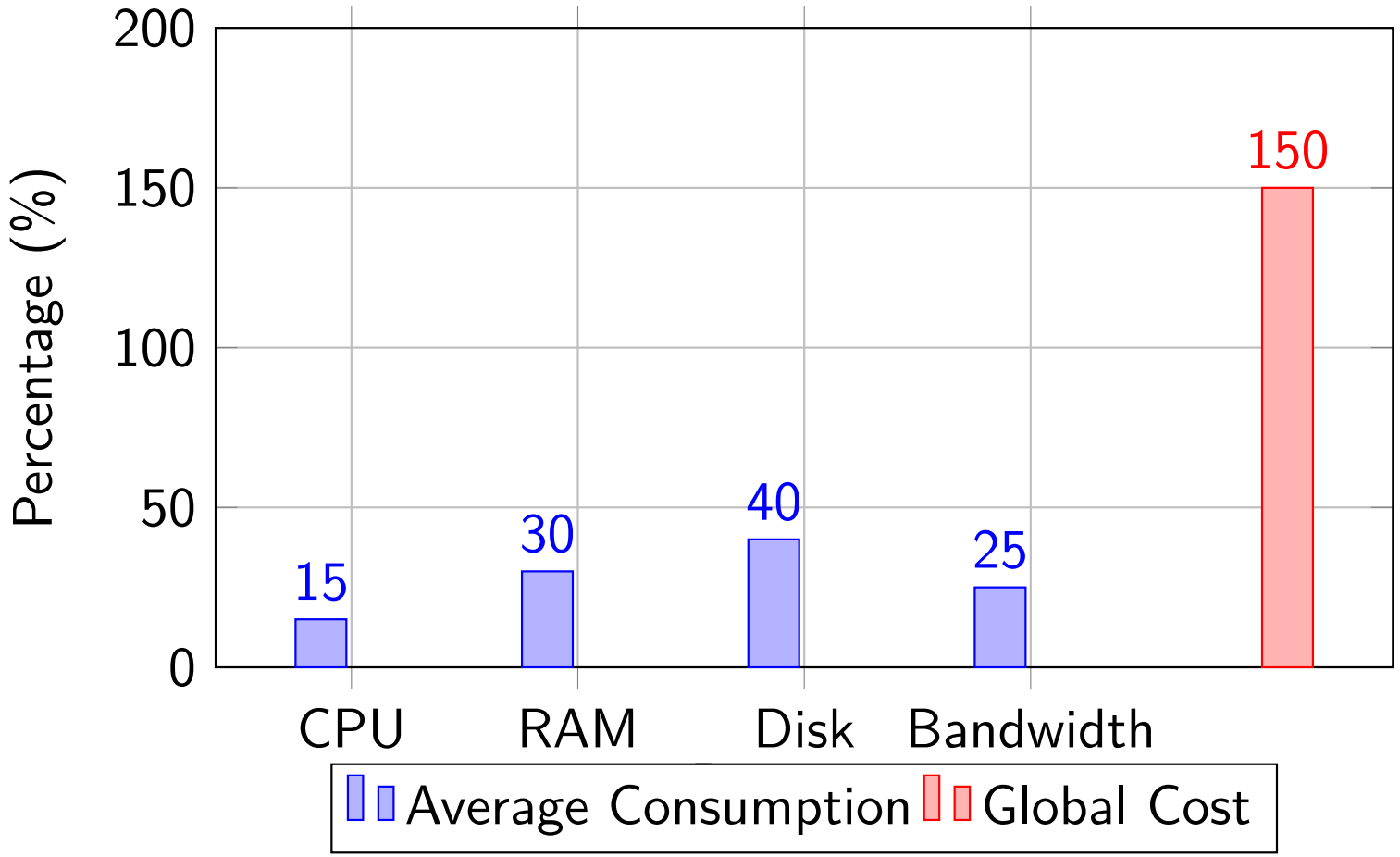


Physical server/Applications

- Physical server
  - Hardware: Processor, memory, motherboard, any other hardware
  - Software: single OS, installed softwares (depending on hardware and OS)
- Applications
  - Executed on top of the OS
  - The OS reserves the necessary resources for running applications
- Resource sharing
  - All server resources are available to a single application.
  - Some resources will be unused

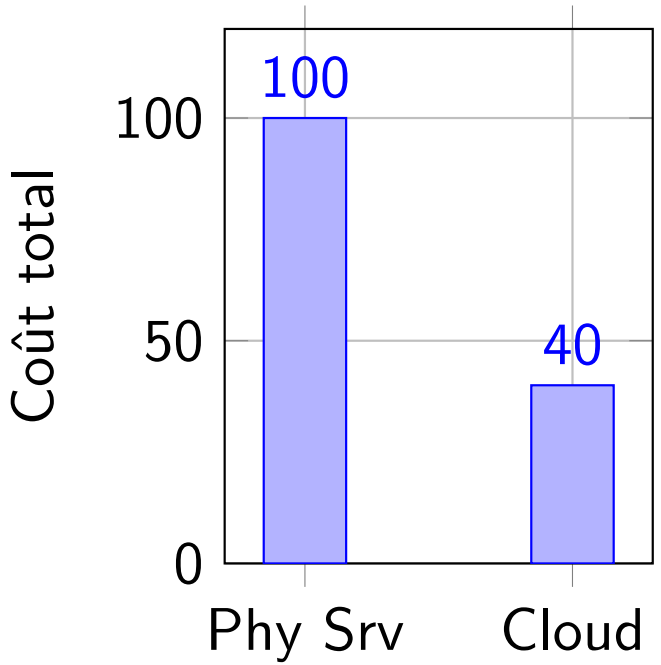
Introduction

- Average daily resource consumption



Introduction

- What if the company only pays for the resources it uses?
  - It would no longer need to invest in purchasing hardware.
  - It would only need to rely on a service provider based on the (*pay-as-you-use*) model $\Rightarrow$ *cloud computing*.



Introduction

- **Benefits of the Pay-as-You-Use Model:**
  - Increased flexibility and adaptability.
  - Reduction in wasted resources.
  - Scalability to meet business demands.
- **Examples:**
  - Major cloud providers: AWS, Azure, Google Cloud.
  - Use cases: Startups and SMEs scaling operations without upfront costs.
- **Challenges:**
  - Dependency on the provider.
  - Potential for unexpected costs if not managed properly.

The Need for Virtualization

- Resource Optimization
  - Traditional physical servers often leave resources underutilized
  - Virtualization maximizes hardware usage by allowing multiple systems to run on a single physical machine
- Cost Efficiency
  - Reduces capital CAPEX on physical infrastructure and operational expenditure
  - Reduces OPEX by lowering power, cooling, and maintenance costs.

Virtualization

- Logical implementation of an element that can have a physical existence
- Examples
  - Networks: VLAN
  - Security: VPN
  - Systems: System virtualization
- System virtualization: simultaneous implementation of multiple operating systems on a single physical machine
- Advantages:
  - Better utilization of hardware resources
  - Dynamic migration of a system and its data
  - Isolation of virtualized environments
  - Segmentation of unsafe environments
  - Sharing of physical infrastructure

Principles/Benefits

- Principle
  - Running multiple OS on the same physical server
  - Sharing physical resources of a server
  - Logical separation of shared resources and service requests
- Benefits
  - Significant cost reduction
  - Easy to manage
  - Easy to Transfer Machines or Data
  - Security (malware attack, backup, restoration)
  - Good way to cloud computing

Host and Guest Operating Systems

- **Host Operating System:**
  - The OS that directly manages physical hardware.
  - Can be specialized as a hypervisor or general to support various programs, including virtualization software.
- **Guest Operating System:**
  - The OS that runs on resources allocated by the host system.
  - It benefits from partitioned physical resources.

Definitions

- **Host Operating System:**
  - The OS that directly manages physical hardware.
  - Can be specialized as a hypervisor or general to support various programs, including virtualization software.
- **Guest Operating System:**
  - The OS that runs on resources allocated by the host system.
  - Benefits from partitioned physical resources.
- **Virtual Machine (VM):**
  - Definition of physical resources forming a functional computer.
  - The guest operating system (all installed programs).
  - Behaves like a physical computer with allocated resources.
- **Container:**
  - A set defined by generic resources allocated by an OS to a program, ensuring exclusive usage (memory, file system, etc.).
  - Includes one or more programs installed on these resources.
  - **Examples:** Docker, Linux runtime environments.

Définitions

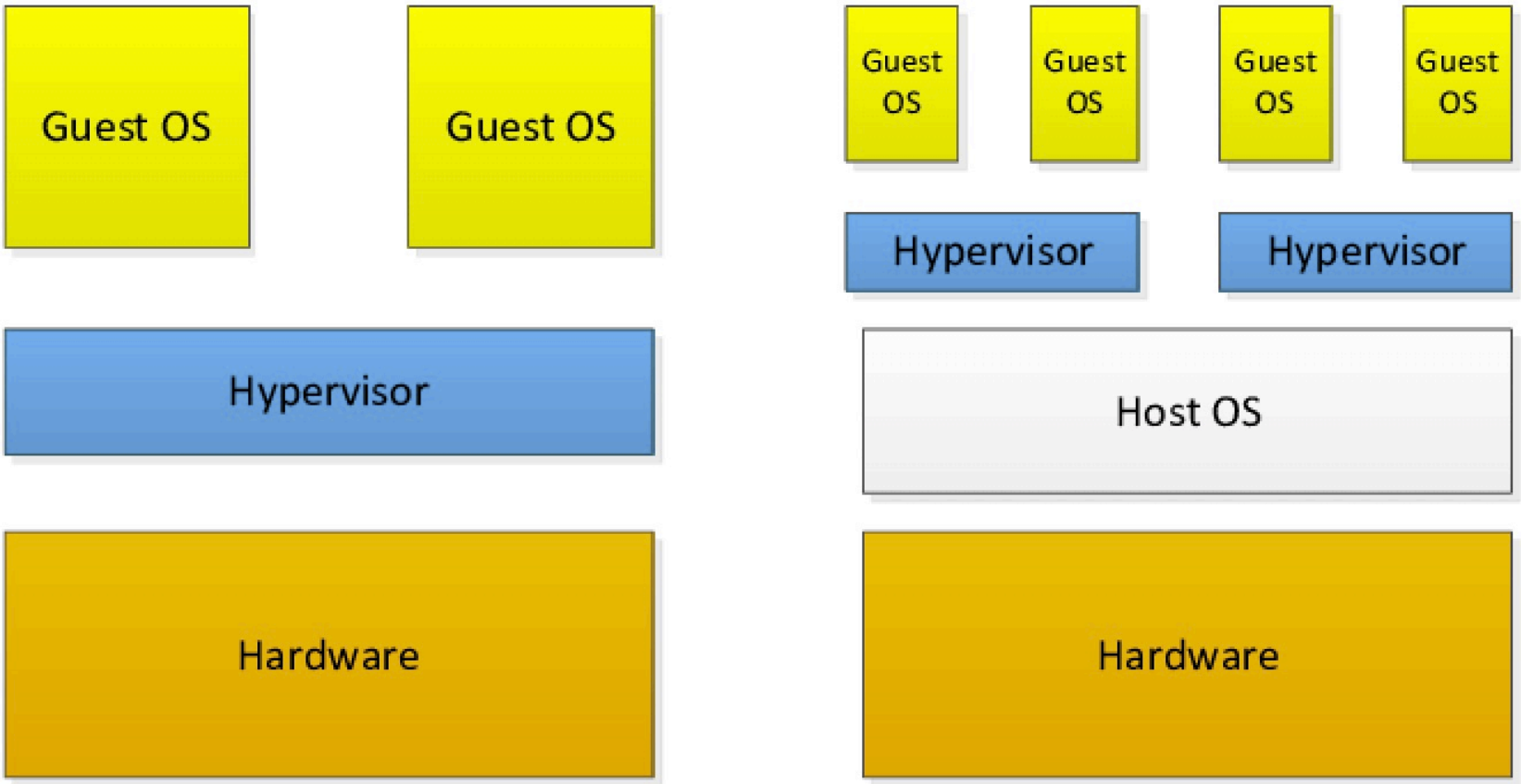
- **Hypervisor:**
  - A specialized host OS designed to host VMs.
  - Installed directly on a physical server.
  - Provides VM lifecycle management: creation, stopping, restarting, etc.
  - Controls resource allocation to VMs.
  - **Examples:** VMware ESXi, Microsoft Hyper-V.
- **Virtual Machine Monitor (VMM):**
  - A program running on a general host OS, enabling VM creation.
  - Unlike a hypervisor, it is treated as an ordinary program by the host.
  - Provides limited resources allocated by the host OS.
  - **Examples:** QEMU, VMware Workstation (Windows), VMware Fusion (macOS), VirtualBox (Windows).

Définitions

- **Isolation:**
  - A necessary property for resource partitioning.
  - Ensures that a VM's use of its allocated resources has no impact on other VMs.
  - Total isolation requires fine control of hardware by the host OS.
  - Allows VMs to crash, restart, or consume resources without affecting others.
- **Snapshot:**
  - A virtualization feature allowing the capture of a VM's state as a set of files.
  - Enables restoring the VM to a saved state.
  - Provides operational comfort, safety, and security.
- **Migration:**
  - A feature allowing the transfer of a VM's execution from one physical server to another.
  - Should occur without interruption or performance loss.

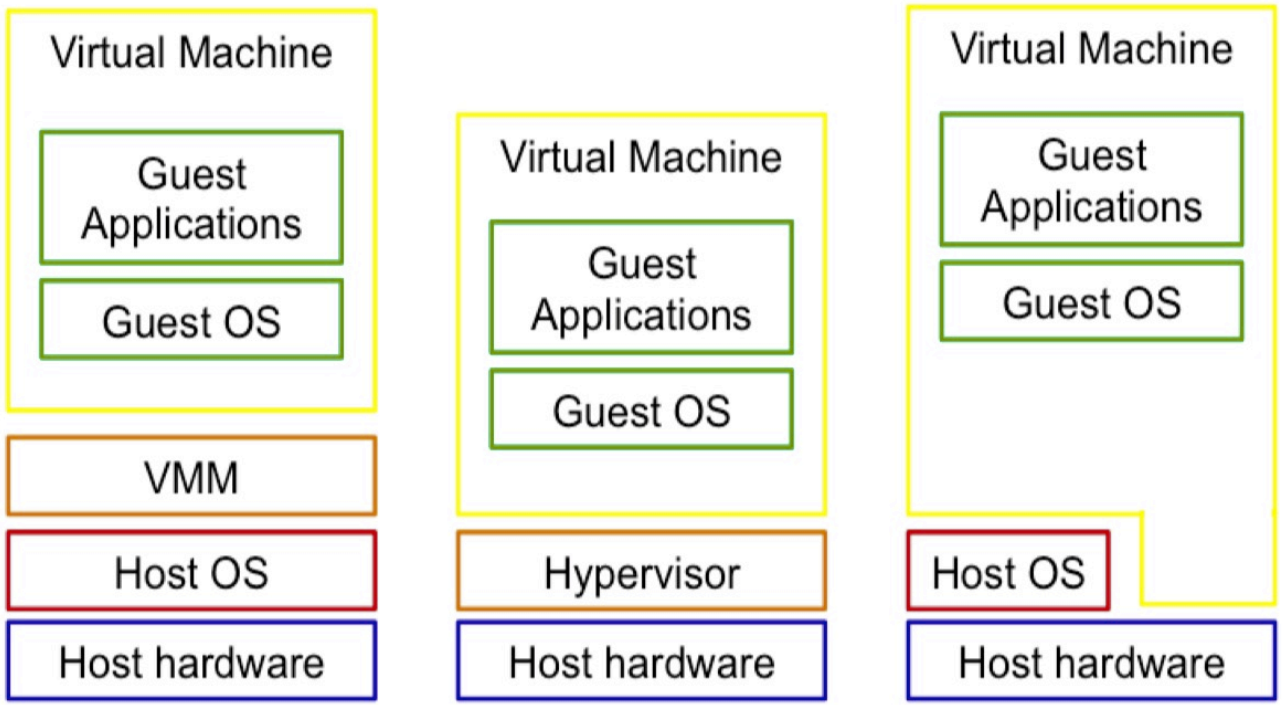
Types of virtualization

- Bare-metal (**native**: Xen, ESX, Hyper-V, KVM)
- Hosted (**using software**: VirtualBox, VMWare)

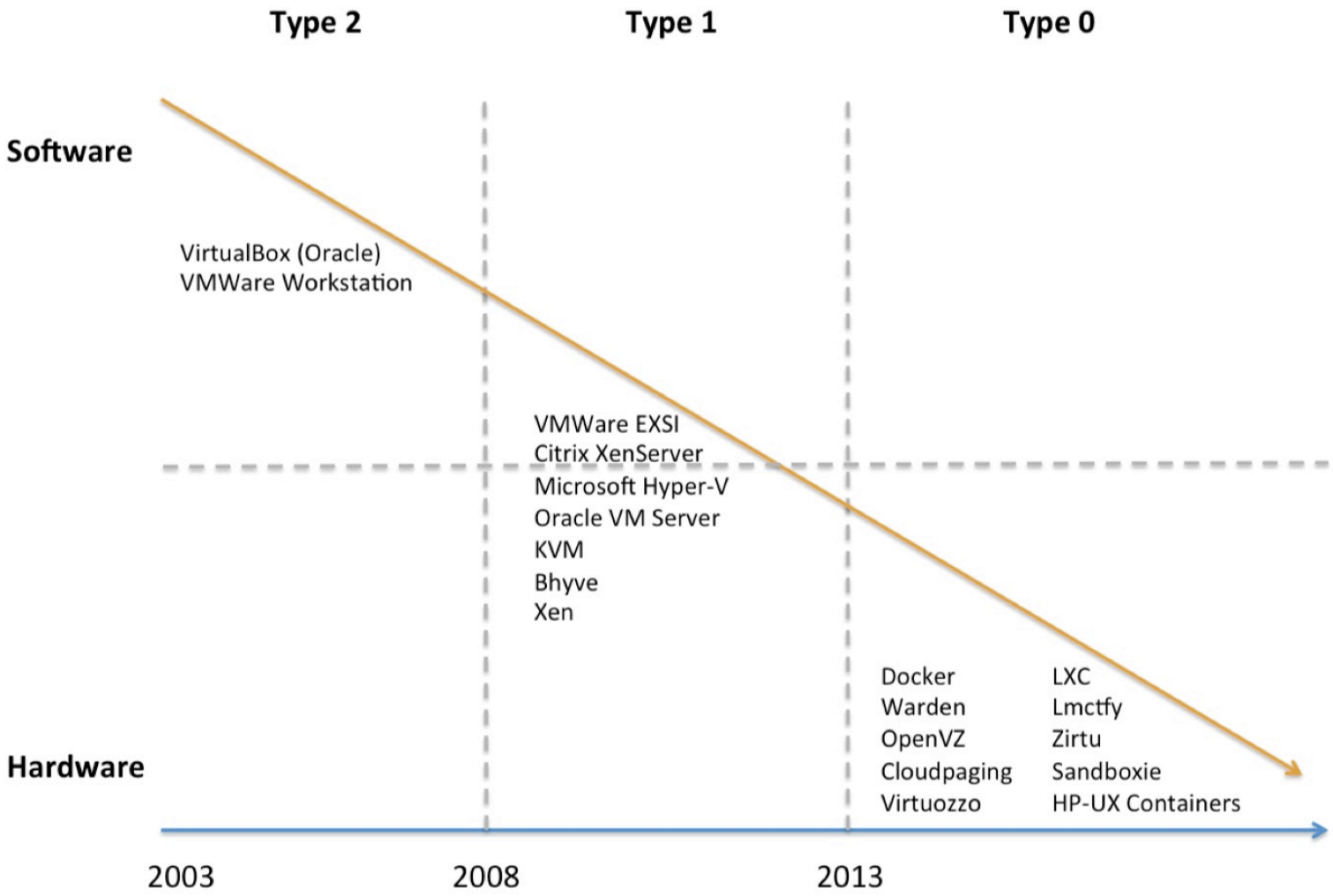


Types of virtualization

- Type 2, Type1, Type 0 :

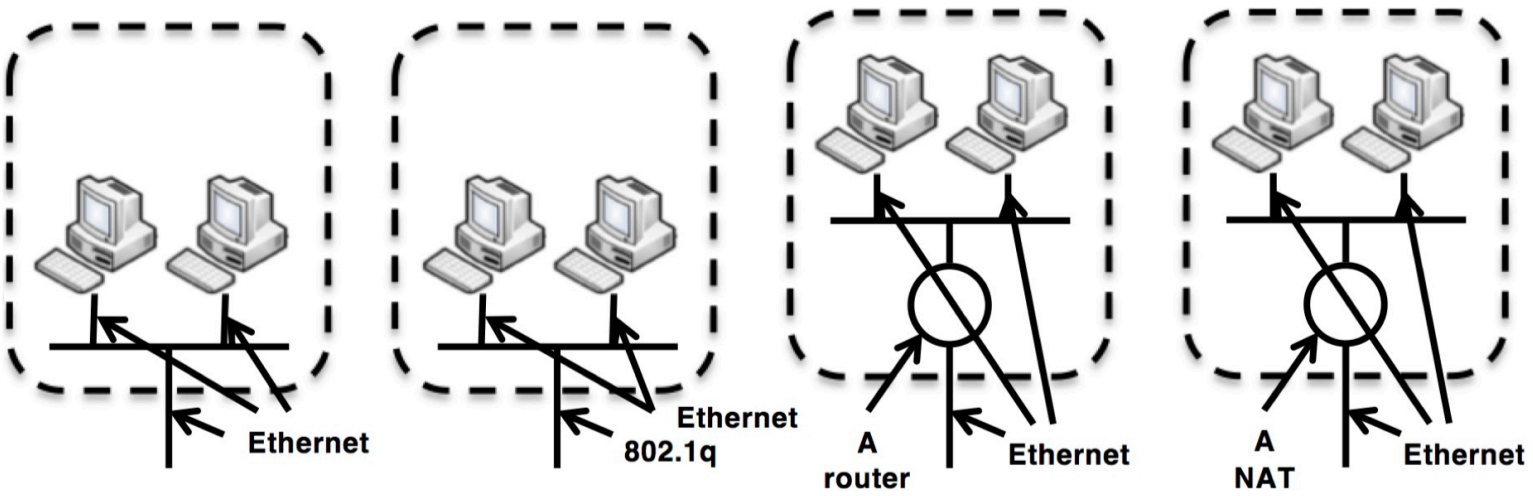


Types of virtualization



Network Virtualization

- **The need for virtualized network function:**
  - Essential for providing network access to each virtualized workstation using the unique host workstation's physical Network Interface Card (physical NIC).
- **Solutions for virtual network access:**
  - VLAN-capable switch.
  - Switch-router.
  - Switch-router performing Network Address Translation (NAT).

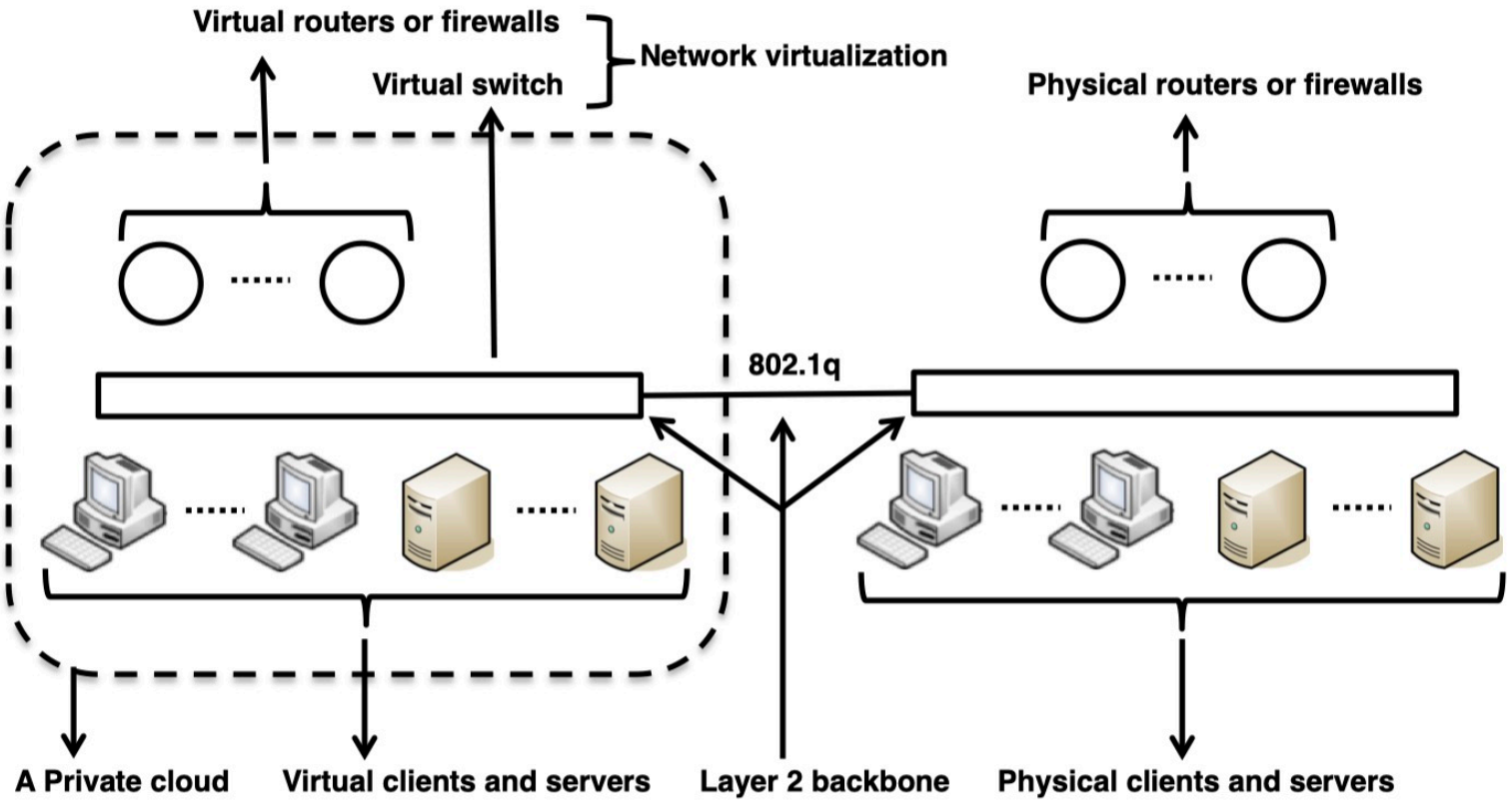


Use Cases of Virtual Network Solutions

- **Using a switch:**
  - Allows adding VMs to the host machine's network.
  - VMs share the IP network of the host machine.
  - Requires each VM and the host to have a unique MAC address.
- **Using a VLAN-capable switch:**
  - Integrates VMs into a layer 2 backbone.
  - VMs can belong to different IP networks in the existing physical network.
  - Utilizes 802.1q protocol on the physical NIC.
  - Host can access one of the networks transported in the backbone.
- **Using a switch-router with NAT:**
  - Ideal for creating VMs without exposing them in the network architecture.
  - VMs can be part of any IP network.
  - Uses Ethernet without needing routing protocols.
  - Common solution as it requires no special NIC and simplifies

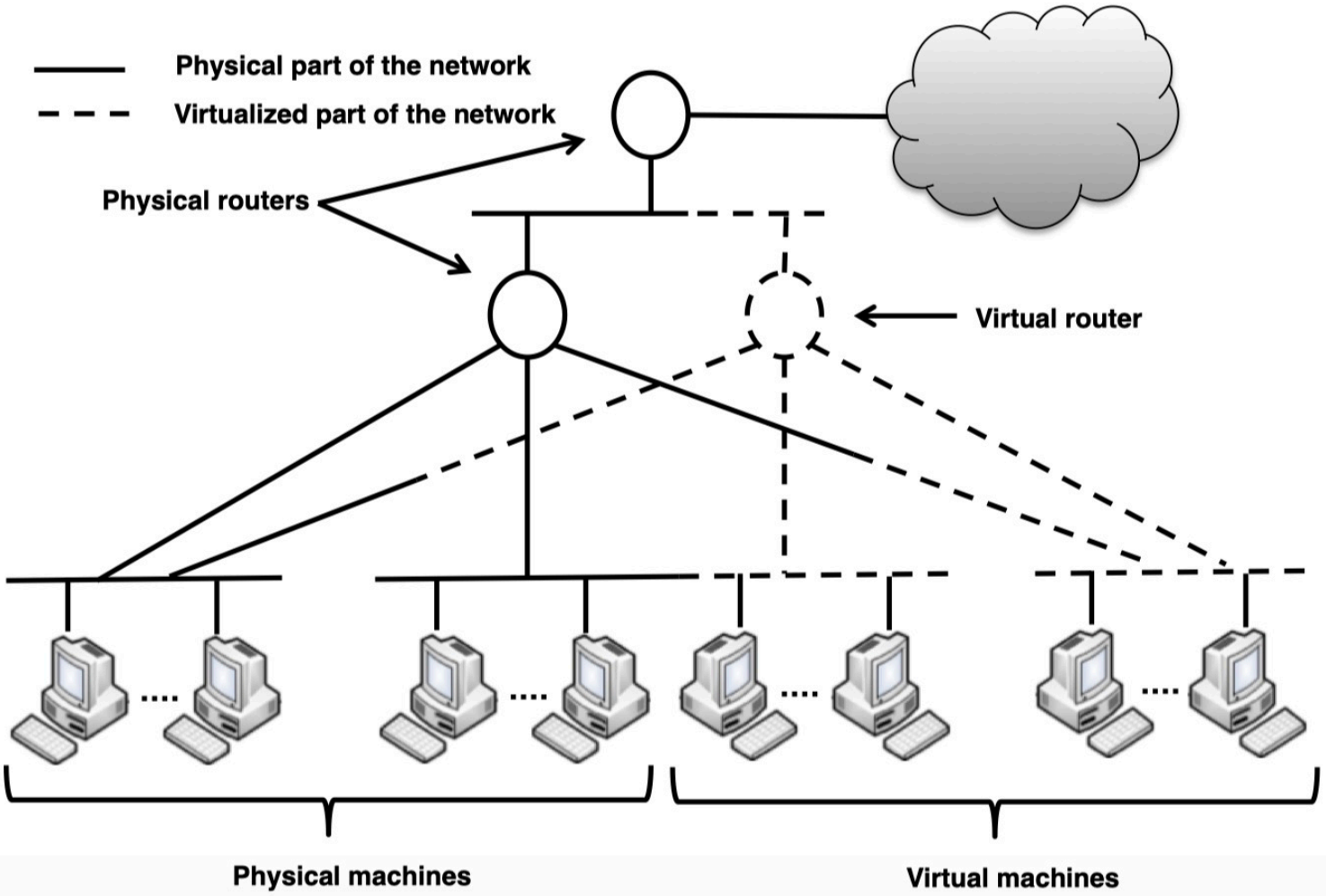
VM production in a datacenter

- Virtual Layer 2



VM production in a datacenter

- Virtual router



Overall concept of cloud

- Principle
  - Virtualization
  - Cloud storage (remote)
  - Remote processing
  - SaaS, IaaS, PaaS, DaaS, Desktop as a service
- Types
  - Public clouds,
  - Private clouds,
  - Hybrid clouds
- Why
  - Reduction of infrastructure costs
  - Better use of resources
  - Abstraction of physical infrastructure layers

Définitions

- Définition 1 : *By using virtualized computing and storage resources and modern Web technologies, cloud computing provides scalable, network-centric, abstracted IT infrastructures, platforms, and applications as on-demand services. These services are billed on a usage basis*
- Définition 2 : *Cloud computing is a model for enabling ubiquitous, 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.*

Sources :
[Cloud Computing: Web-Based Dynamic IT Services]

Overall concept of cloud

- Examples of cloud usage
  - SaaS: Google Docs, Sheets, Slides
  - PaaS: Oracle PaaS
  - IaaS: Amazon, Numergy, why not ESP-DATA-CLUB IaaS?
- Needs:
  - Control and supervision of the entire environment
  - Data protection by duplication
  - Management and reuse of physical resources
  - Adaptation to the specific needs of a client
  - Perfect isolation between clients

How

- Is there any software for creating and managing Clouds?
  - Free?
  - Paid?
- Can we share physical resources
  - Abstraction of physical hardware layers
  - Abstraction of the physical network layers

Open Source Cloud Management Tools

- **OpenStack**
  - Comprehensive platform for building and managing public and private clouds.
- **CloudStack**
  - Open-source software for creating, deploying, and managing cloud services.
- **Kubernetes**
  - Container orchestration tool used for managing cloud-native applications.
- **Proxmox VE**
  - Open-source virtualization environment for managing VMs and containers.
- **Nextcloud**
  - Personal cloud platform for file storage, collaboration, and synchronization.

Free Cloud Management Tools

- **OwnCloud**
  - Similar to Nextcloud, offering file sharing and personal cloud services.
- **Terraform (Open Source Version)**
  - Infrastructure as code tool for building, changing, and versioning infrastructure safely and efficiently.
- **Apache CloudStack**
  - A scalable, open-source platform to deploy and manage large networks of virtual machines.

Paid Cloud Management Tools

- **VMware vCloud Suite**
  - Integrated solution for creating and managing private and hybrid clouds.
- **Microsoft Azure**
  - Full-service cloud platform offering IaaS, PaaS, and SaaS for enterprises.
- **Amazon Web Services (AWS)**
  - Industry leader providing a wide range of tools and cloud services for different use cases.
- **Google Cloud Platform (GCP)**
  - Offers a suite of cloud computing services that runs on the same infrastructure as Google's end-user products.

Software as a service

- Définition (source NIST) :
  The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings.

source : `https://csrc.nist.gov/glossary/term/software_as_a_service`

Platform as a service

- Définition (source NIST)

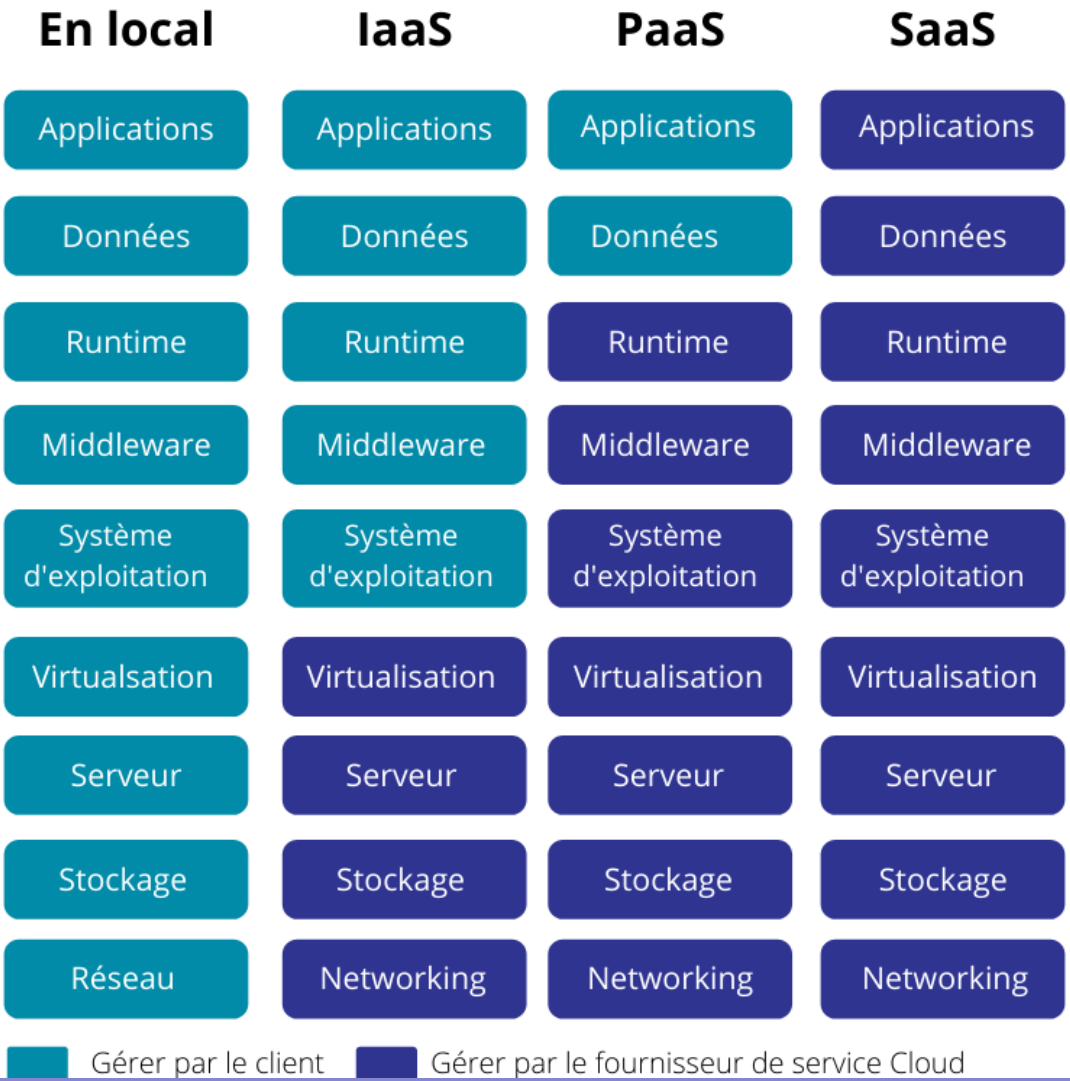
The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a service

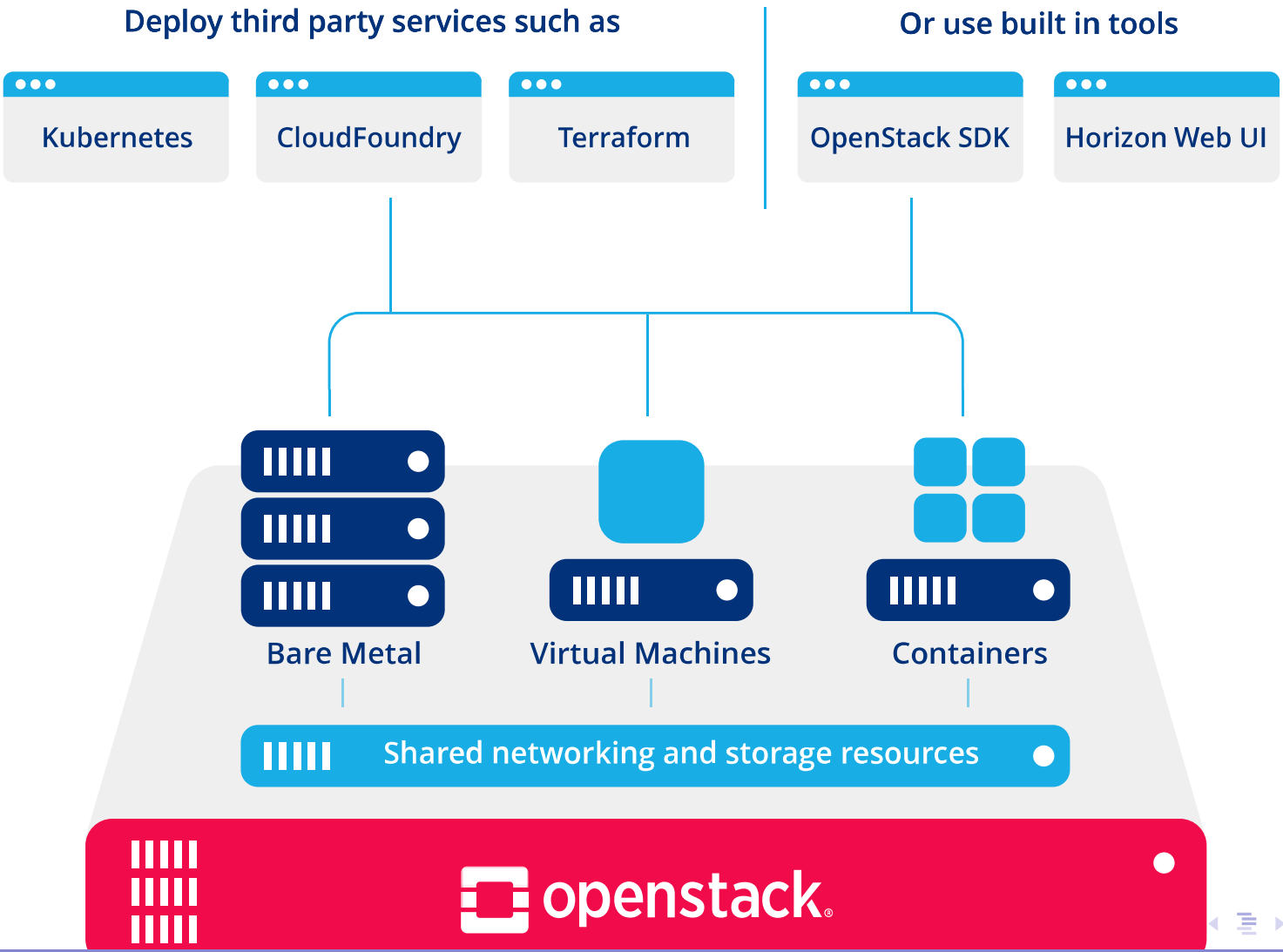
- Définition (source NIST)

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Les services du cloud



Introduction to OpenStack



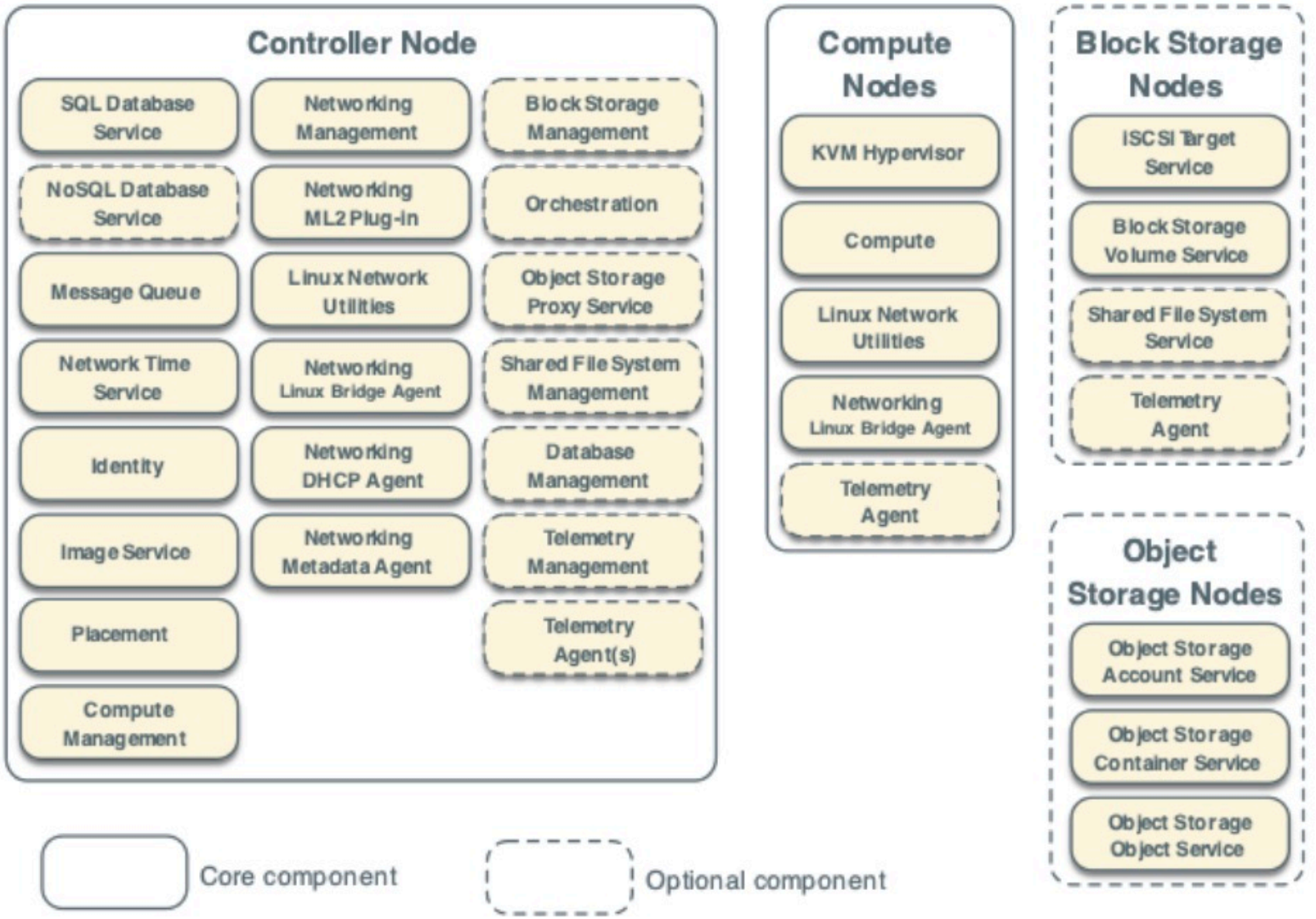
Introduction to OpenStack

- **Definition:**
  - OpenStack is an open-source cloud computing platform for public and private clouds.
  - Designed to be simple to implement, massively scalable, and feature-rich.
- **Core Services:**
  - **Compute (Nova):** Manages the lifecycle of compute instances.
  - **Networking (Neutron):** Manages and connects networks for VMs.
  - **Storage:**
    - **Block Storage (Cinder):** Provides persistent block storage.
    - **Object Storage (Swift):** Stores and retrieves unstructured data.
- **Main Use:** Enables Infrastructure-as-a-Service (IaaS) solutions.

OpenStack Architecture

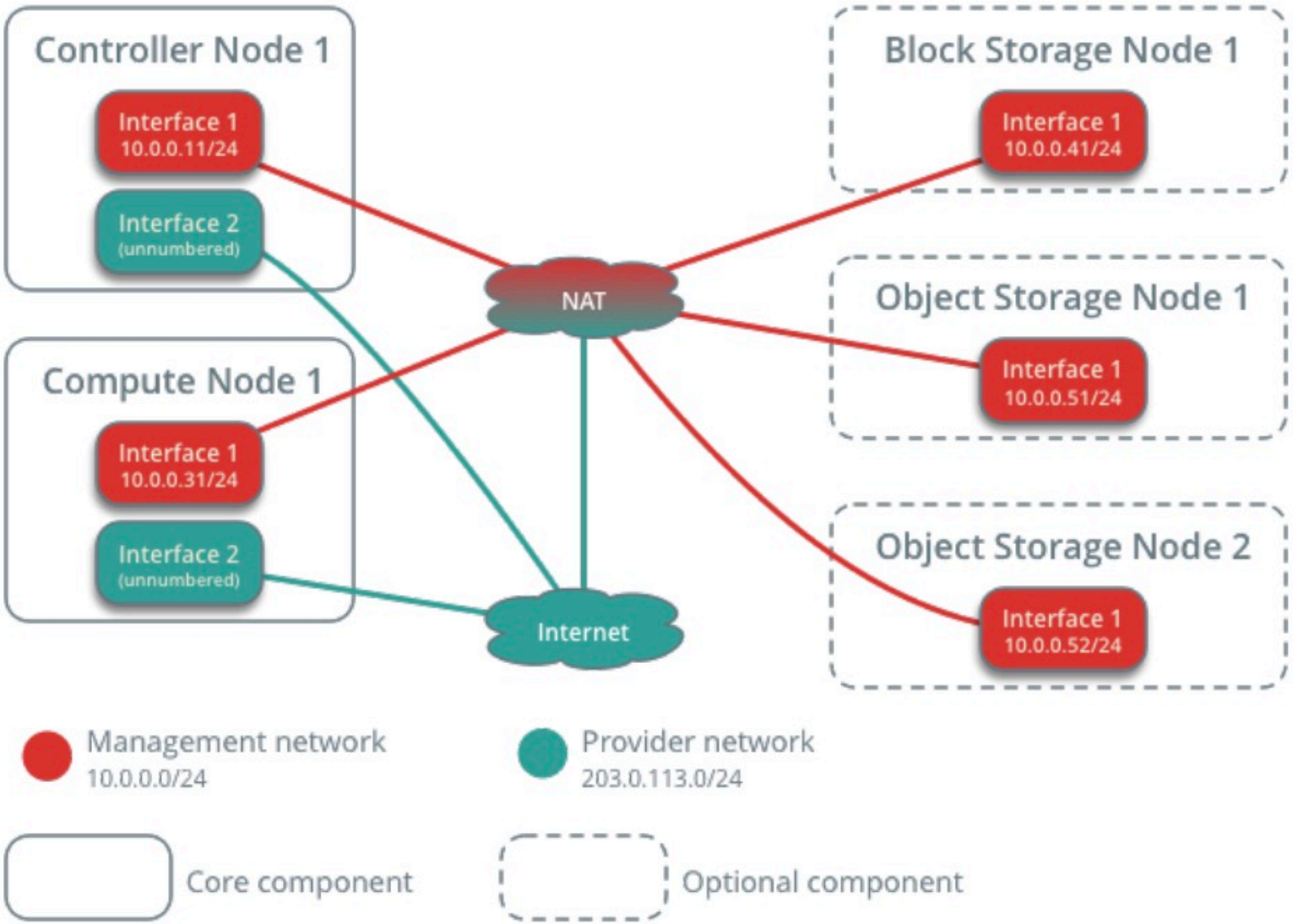
- **Modular Components:**
  - **Controller Node:** Manages identity, dashboard, and service management.
  - **Compute Node:** Runs the hypervisor for virtual instances.
  - **Storage Nodes:** Dedicated for block and object storage.
- **Communication:**
  - Uses **APIs** for inter-service communication.
  - Supports **AMQP** (e.g., RabbitMQ) for messaging between processes.
- **Interfaces:**
  - **Horizon Dashboard:** Web-based user interface.
  - Command-Line Clients.
  - RESTful APIs for integration.

Introduction to OpenStack



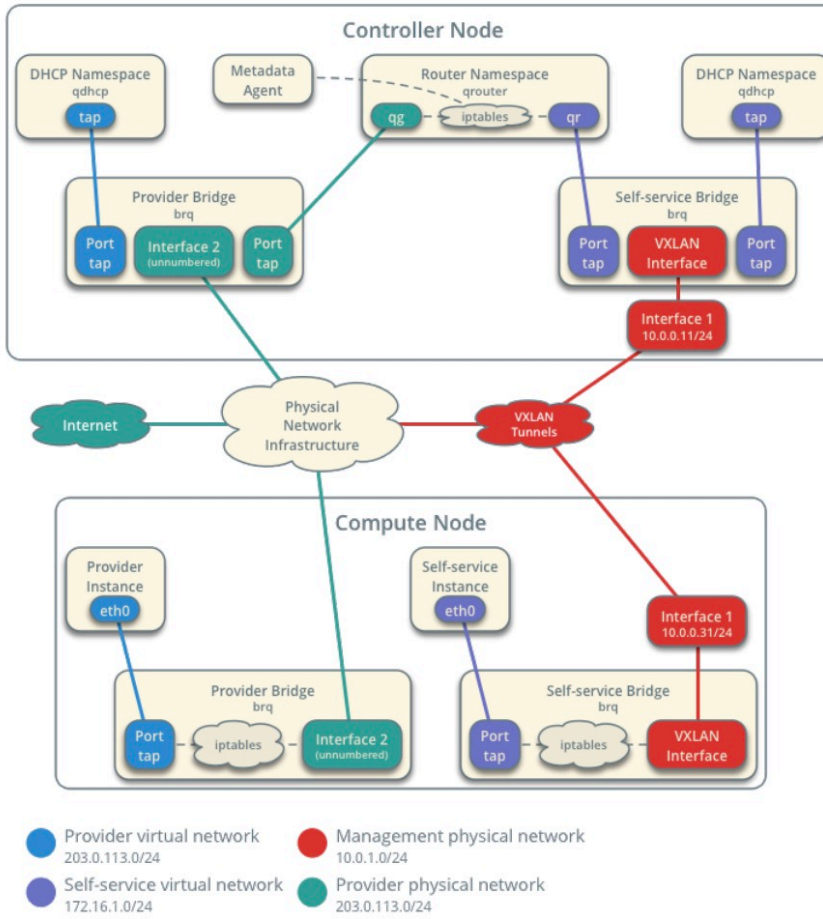
Networking in OpenStack

- **Neutron:** The networking component in OpenStack.
- **Functions:**
  - Provides network connectivity-as-a-service.
  - Manages networks, subnets, and routers.
  - Supports VLANs, VXLANs, and flat networks for layer 2 connectivity.
- **Types of Networks:**
  - **Provider Networks:** Directly mapped to physical networks.
  - **Tenant Networks:** Isolated virtual networks created per project.
- **Security Groups:**
  - Define rules for inbound and outbound traffic for VMs.


Controller Node 1
Interface 1
10.0.0.11/24
Interface 2
(unnumbered)
Compute Node 1
Interface 1
10.0.0.31/24
Interface 2
(unnumbered)
NAT
Internet
Block Storage Node 1
Interface 1
10.0.0.41/24
Object Storage Node 1
Interface 1
10.0.0.51/24
Object Storage Node 2
Interface 1
10.0.0.52/24
Management network
10.0.0.0/24
Provider network
203.0.113.0/24
Core component
Optional component

VM Connectivity and Internet Access in OpenStack

- **Connecting VMs**
  - VMs can communicate within the same tenant network.
- **Internet Access for VMs:**
  - Achieved through routers with external gateways.
  - Floating IPs are used to give VMs public access.
  - NAT allows private IPs of VMs to be translated to public IPs for external communication.
  - A VM sends traffic to the router, which forwards it to the external network via NAT.



Advantages of Using OpenStack

- **Flexibility:** Customizable for cloud deployment needs.
- **Open-Source:** Free and supported by a large, active community.
- **Scalability:** Suitable for small setups to large data centers.
- **Ecosystem Support:** Integrates with tools like Ansible and Puppet for automation.
- **Multi-Platform Compatibility:** Available on Linux distributions (e.g., Ubuntu, CentOS).

Docker

Conclusion

- Virtualization and cloud technologies provide enhanced resource utilization, scalability, and flexibility.
- The transition from physical servers to virtualized environments offers significant cost savings and operational efficiency.
- OpenStack, as an open-source cloud platform, enables robust IaaS solutions with comprehensive modular services.
- Networking in OpenStack supports complex architectures, ensuring secure and isolated communication for VMs.
- The use of tools like Docker facilitates containerized applications, making deployment and management simpler.
- Automation and orchestration tools will be key for future cloud management.

Final Thoughts

- The choice of virtualization and cloud strategies should align with specific business needs for optimal benefits.
- Open-source solutions like OpenStack and Docker empower organizations to create scalable, cost-effective cloud environments.
- Staying up-to-date with industry trends and best practices ensures continued success in leveraging these technologies.