Virtualization in cloud

Introduction & benefit of Virtualization, Implementation Levels of Virtualization, Types: Full and para virtualization Taxonomy of virtualization techniques - Execution Virtualization, Virtualization and cloud computing, Pros and cons of virtualization

Introduction & benefit of Virtualization

- Virtualization is a technique of how to separate a service from the underlying physical delivery of that service.
- it is a technique that allows multiple users or organizations to make use of a single resource thread or an application among themselves.
- It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource.
- With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

Introduction & benefit of Virtualization

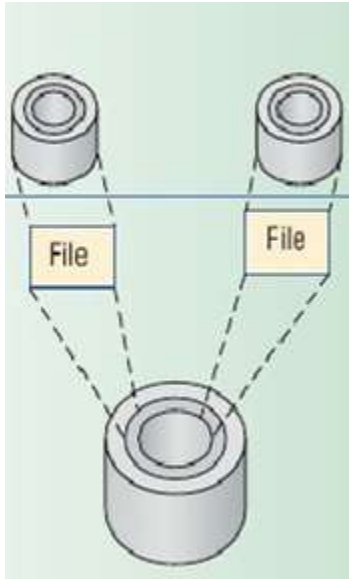
- one of the main cost effective, hardware reducing, and energy saving techniques used by cloud providers is virtualization.
- Virtualization allows to share a single physical instance of a resource or an application among multiple customers and organizations at one time.
- It does this by assigning a logical name to a physical storage and providing a pointer to that physical resource on demand.
- The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing.
- Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

Abstraction and Virtualization

- Computer system is complex, and yet it continue to evolve. Computer requires Processor, RAM, ROM, network
- Computer is designed as hierarchies of well-defined interfaces that separate level of abstraction
- Simplifying abstractions hide lower-level implementation details

Abstraction

- Ex. Disk storage
- Hides hard-disk addressing details (sectors and tracks)
- It appears to application software as a variable sized files.
- User can create, write and read files without knowing the underneath details.

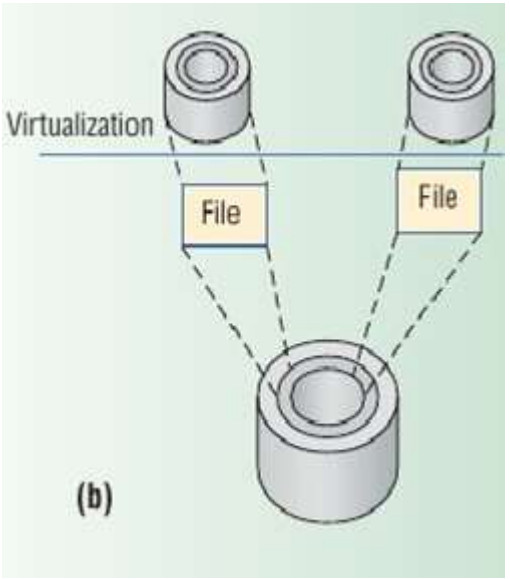


Pros and cons of Abstraction

- Well-defined interfaces permit development of interacting computer subsystems not only in different organization but also at different time.
- Limitation of well-defined interfaces , designed specification to one interface will not work for other.

Virtualization

- Virtualization of system or components like – processor, memory or an I/O device – at a given abstraction level.
- It transforms a entire system or components of the system Ex. disk storage



Abstraction and Virtualization in Cloud Computing

- Cloud computing virtualizes systems by pooling and sharing resources.
- Cloud computing uses abstraction to enable the rapid deployment of data and applications to minimize the complexity and cost of providing the underlying resource, which eventually simplifies operations.
- Definition of virtualization: Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources".
- In other words, Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded.

Virtual Machine

- Virtualization can be applied to entire machine.
- VM can be implemented by adding a software layer to a real machine to support desired architecture.
- VM implementation lie at architected interfaces

Architecture of Virtual Machines

- VM can support individual processes or a complete system
- Virtualization can be from OS to programming languages to processor architecture.
- VMs enhance
  - Software interoperability (to work together)
  - System impregnability (having strength)
  - Platform versatility

Virtualization	Abstraction
It is an umbrella of technologies and concepts that are intended to provide an abstract environment to run applications.	It is the act of representing essential features while hiding the background details from users and developers.
It allows creating a virtual version of something, including computer resources, virtual computer hardware platform, and storage devices.	It allows abstraction of the physical implementation to hide technical details from consumers.
Computer resources can be divided or shared by multiple environments simultaneously, which are known as virtual machines (VMs).	It allows changes to be occurred in the backend without affecting functionalities of the applications in the abstraction layer.

BENEFITS OF VIRTUALIZATION

1. More flexible and efficient allocation of resources.
2. Enhance development productivity.
3. It lowers the cost of IT infrastructure.
4. Remote access and rapid scalability.
5. High availability and disaster recovery.
6. Pay peruse of the IT infrastructure on demand.
7. Enables running multiple operating systems.

Architected Interfaces

- Architecture, as applied to computer systems, refer to a formal specification to an interface in the system, including the logical behavior of the resources managed via the interface.
- Implementation describes the actual embodiment of an architecture.
- Abstraction levels correspond to implementation layers, having its own interface or architecture.

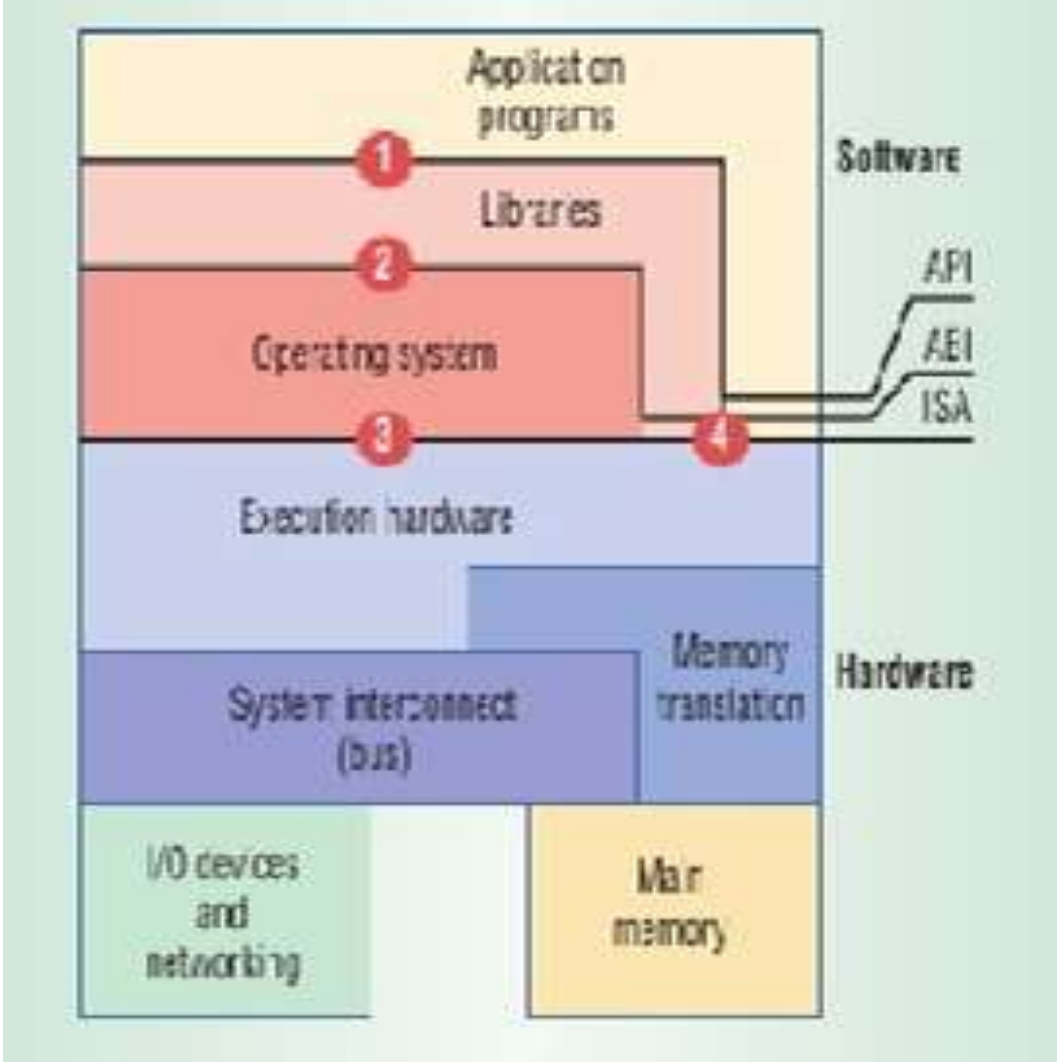
Virtualized Environments

How it works

1. **Hypervisor Layer**: A special software layer called a hypervisor is installed on the physical hardware.
2. **VM Creation**: The hypervisor creates multiple virtual environments (VMs) on top of the physical machine.
3. **Resource Sharing**: Each VM functions as a separate, independent computer, with its own virtual CPU, RAM, and storage, but shares the underlying physical hardware resources with other VMs.
4. **Guest OS**: Each VM can run its own operating system, known as the guest OS, which is distinct from the host OS of the physical machine.

Computer System Architecture

- Interfaces at or near the H/w S/w boundary :
- – ISA – Instruction Set Architecture.
- – API – Application Program Interface
- – ABI – Application Binary Interface



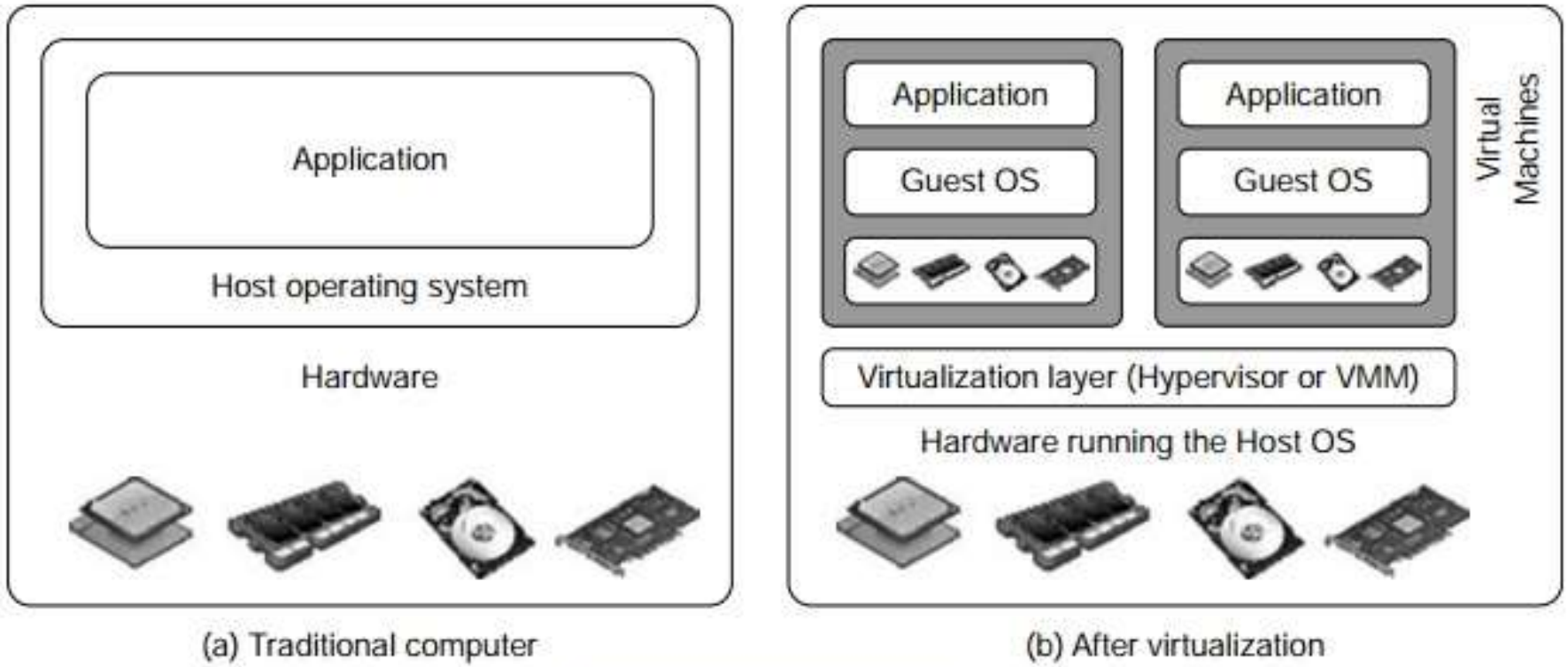
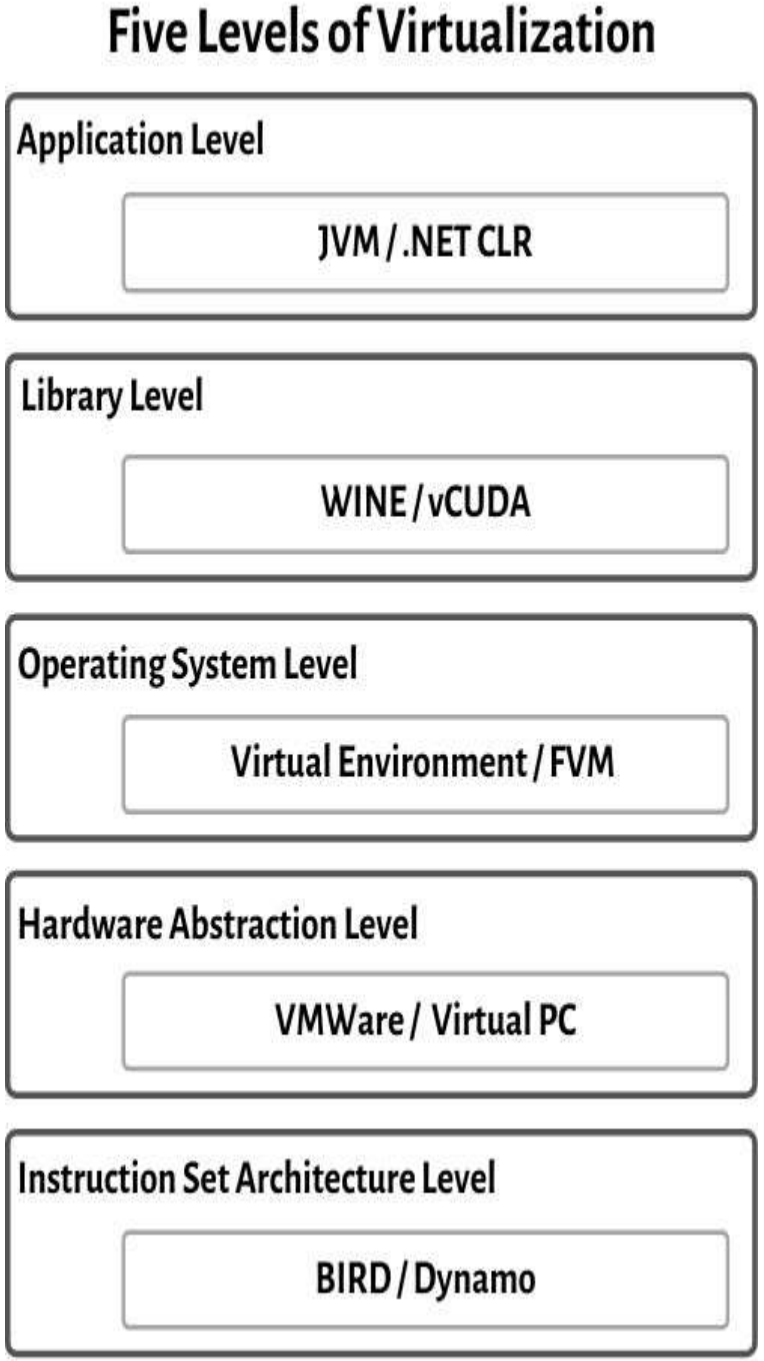


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.

Levels of Virtualization by Layer

1. **Instruction Set Architecture (ISA) Level:** This level involves emulating a CPU's instruction set to run software designed for one architecture on a different one.
2. **Hardware Abstraction Layer (HAL) Level:** Uses a hypervisor to create multiple virtual machines (VMs) from physical hardware, allowing for full virtualization.
3. **Operating System (OS) Level:** This allows multiple isolated user spaces or containers to run on a single OS instance.
4. **Library Level:** Uses APIs to create virtual environments for applications without a full OS emulation, controlling how applications interact with system resources.
5. **Application Level:** This isolates individual applications from the underlying OS, allowing them to run in a virtualized, independent environment, as seen with containerization.



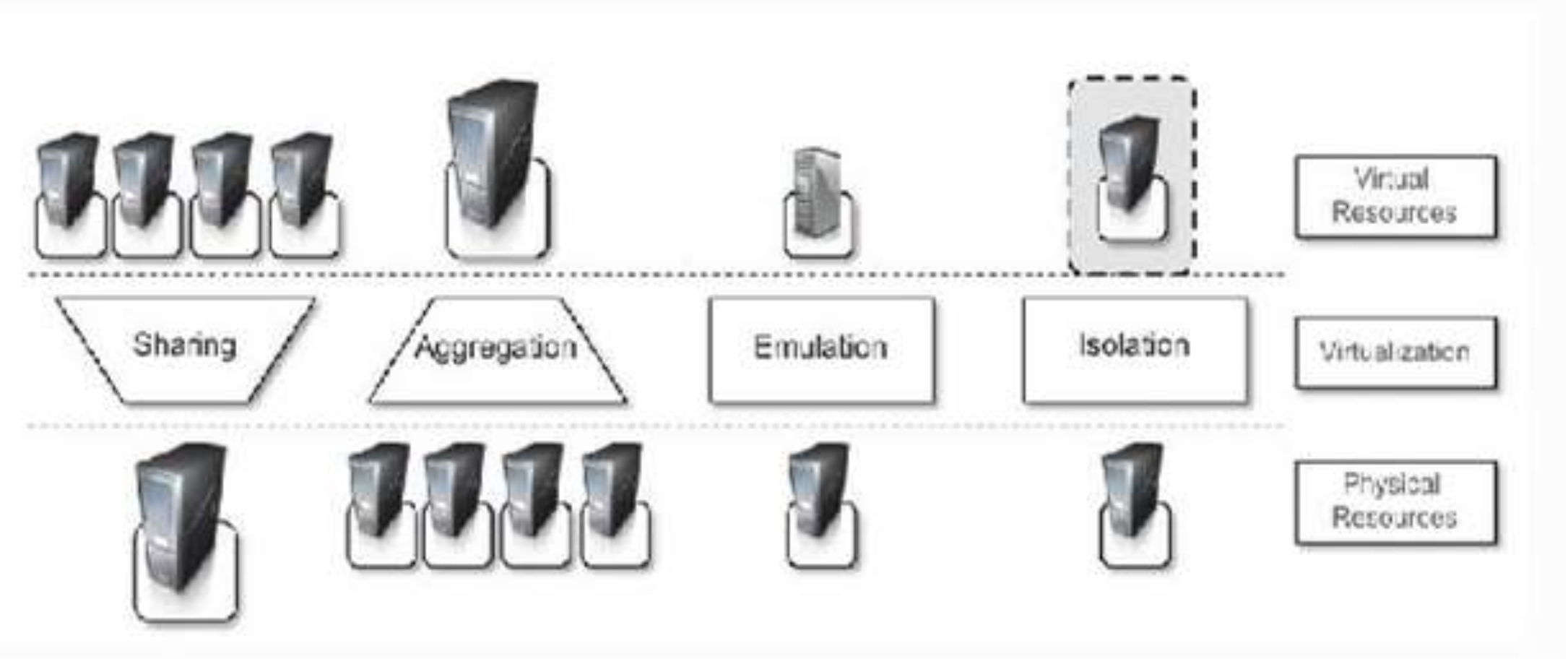
Advantages of Virtualization

- Increased Security
  - Ability to control the execution of a guest
  - Guest is executed in emulated environment.
  - Virtual Machine Manager control and filter the activity of the guest.
  - Hiding of resources.
  - Having no effect on other users/guest environment.

Advantages of Virtualization

- Managed Execution types :
  - Sharing - Creating separate computing environment within the same host. Underline host is fully utilized.
  - Aggregation – A group of separate hosts can be tied together and represented as single virtual host.
  - Emulation –Controlling & Tuning the environment exposed to guest.
  - Isolation Complete separate environment for guests.

Managed Execution



Advantages of Virtualization

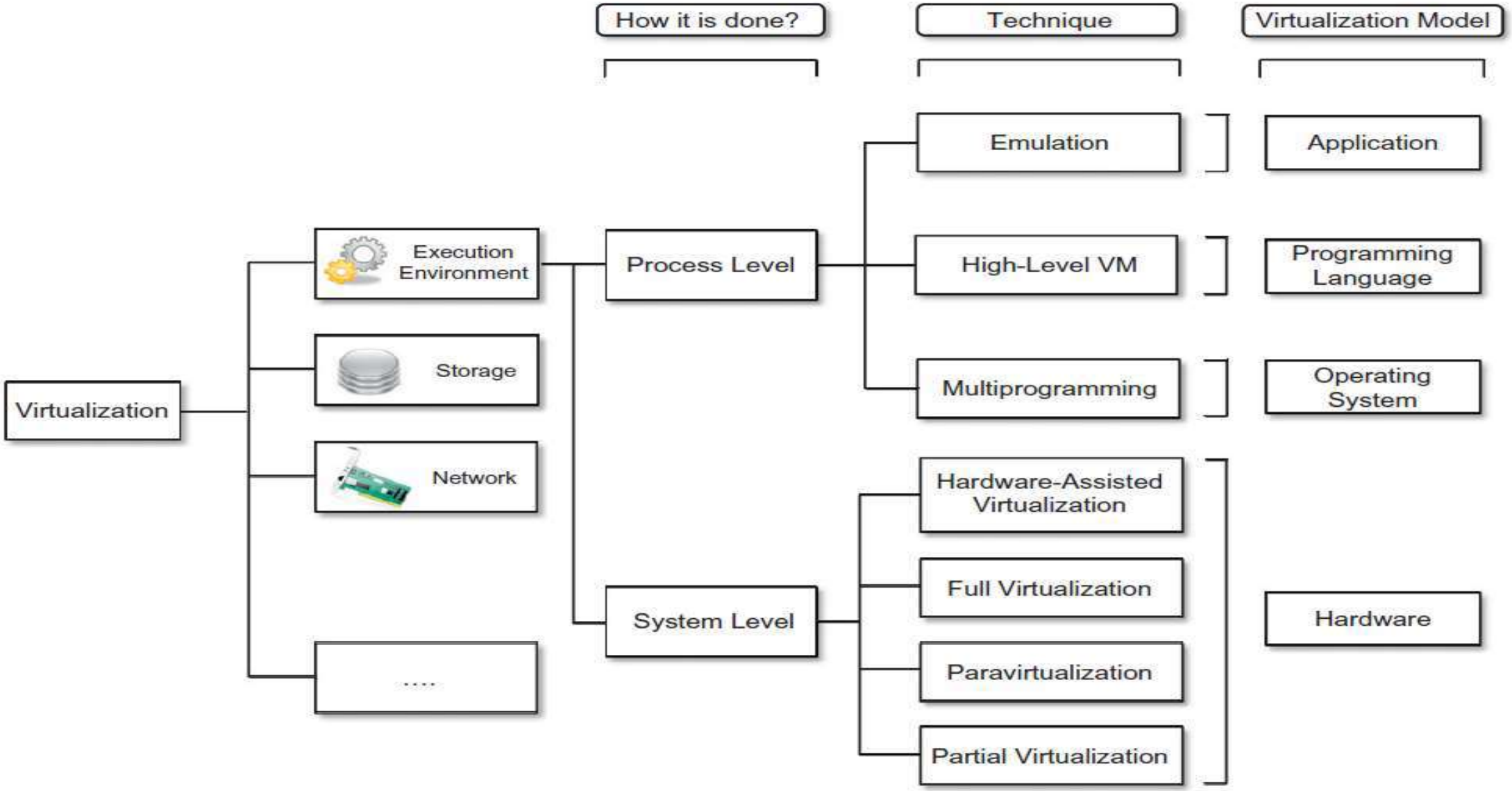
- Performance Tuning – control the performance of guest.
- Virtual Machine Migration – move virtual image into another machine.
- Portability – safely moved and executed on top of different virtual machine. Availability of system is with you.

https://www.youtube.com/watch?v=UBVVq-xz5i0

Taxonomy of Virtualization Techniques

- Virtualization is mainly used to emulate execution environment , storage and networks.
- Execution Environment classified into two :
  - Process-level – implemented on top of an existing operating system.
  - System-level – implemented directly on hardware and do not or minimum requirement of existing operating system

Taxonomy of Virtualization Techniques

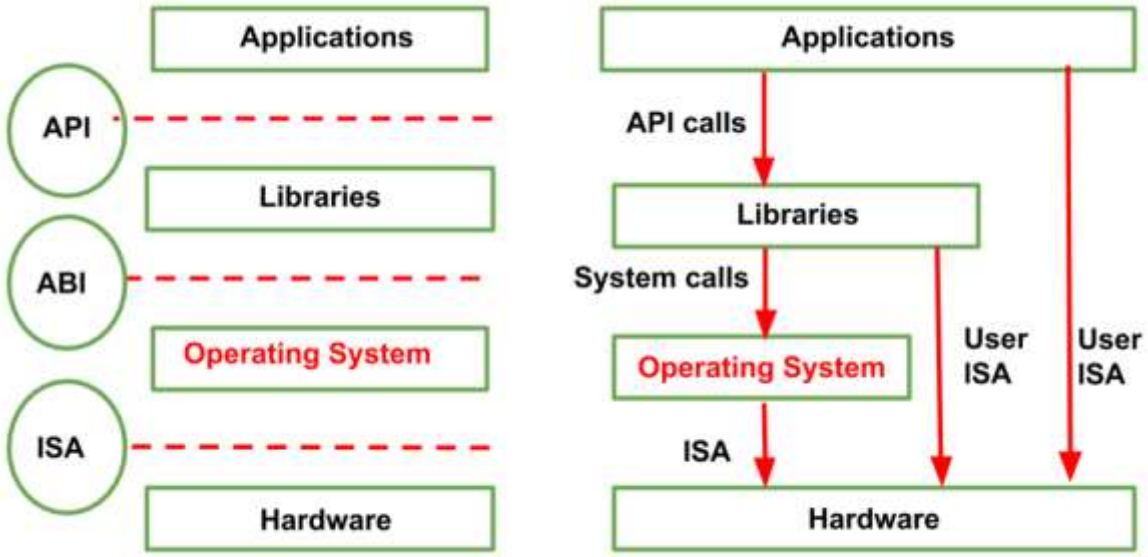


Machine Reference Model

- It defines the interfaces between the levels of abstractions, which hide implementation details.
- Virtualization techniques actually replace one of the layers and intercept the calls that are directed towards it.

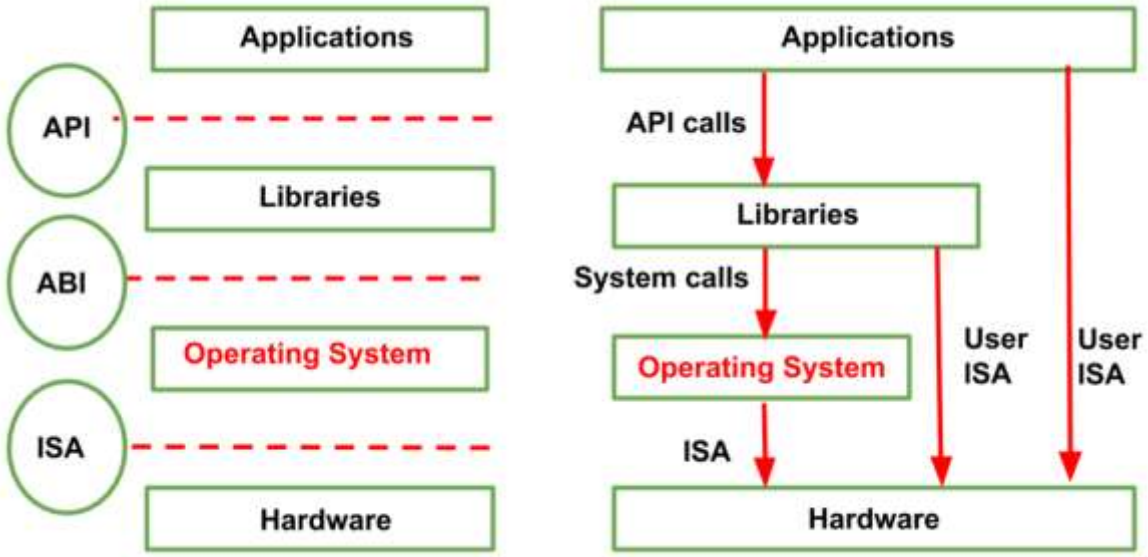
Machine Reference Model

- Hardware is expressed in terms of the Instruction Set Architecture (ISA).
- ISA for processor, registers, memory and the interrupt management.
- Application Binary Interface (ABI) separates the OS layer from the application and libraries which are managed by the OS.
  - System Calls defined
  - Allows portabilities of applications and libraries across OS.



Machine Reference Model

- API – it interfaces applications to libraries and/or the underlying OS.
- Layered approach simplifies the development and implementation of computing system.
- ISA has been divided into two security classes:
  - Privileged Instructions
  - Nonprivileged Instructions



ISA: Security Classes

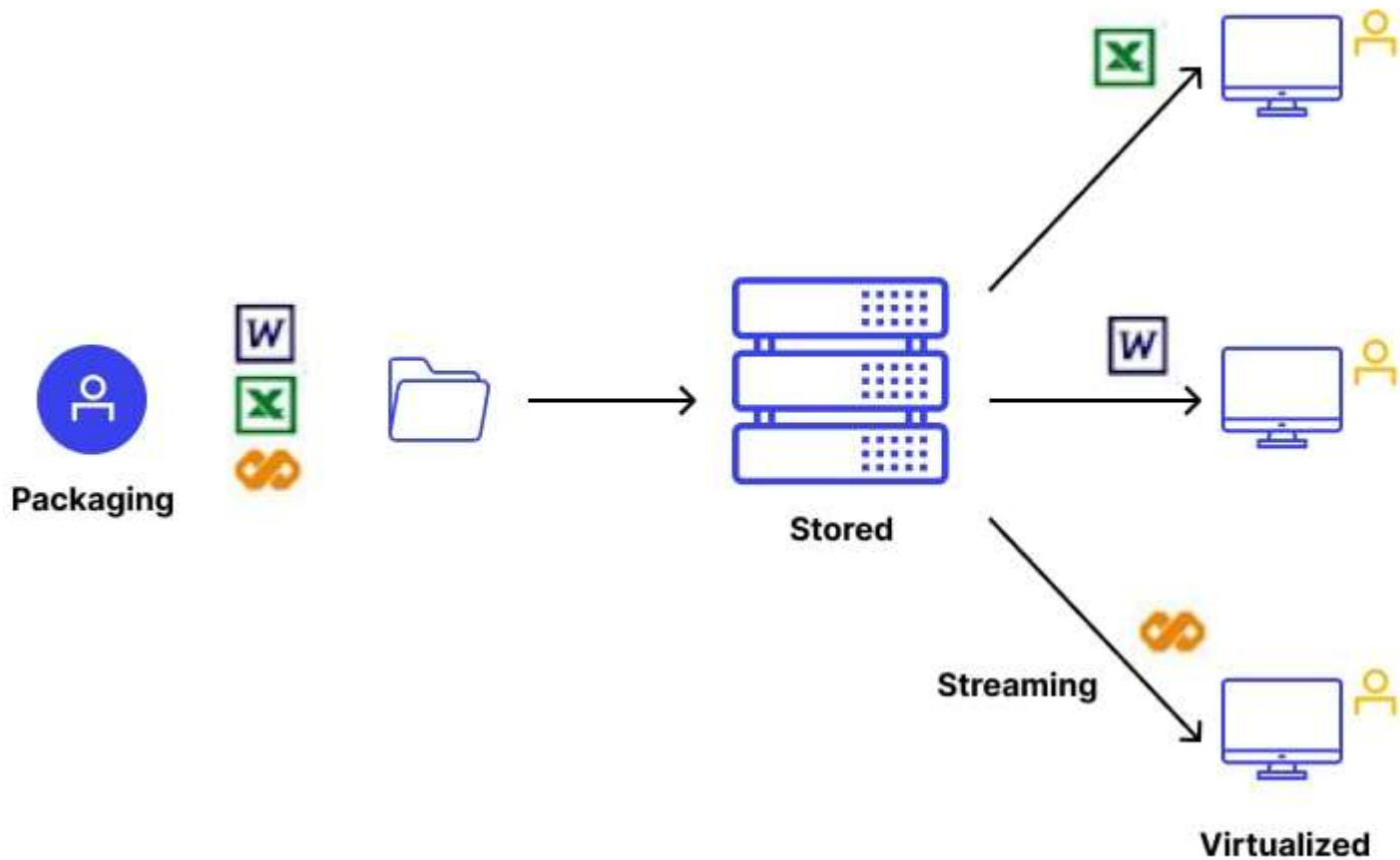
- Nonprivileged instructions – Non-privileged instructions are **those that can be executed by any process, including user-level processes**.
  - That can be used without interfering with other tasks because they do not access shared resources. Ex. Arithmetic , floating & fixed point.
- Privileged instructions - Privileged instructions are **those that can only be executed by the operating system kernel or a privileged process, such as a device driver**.
  - That are executed under specific restrictions and are mostly used for sensitive operations, which expose (behavior-sensitive) or modify (control-sensitive) the privileged state.
    - Behavior-sensitive – operate on the I/O
    - Control-sensitive – alter the state of the CPU register.

Types of Virtualization

1. Application Virtualization.
2. Network Virtualization.
3. Desktop Virtualization.
4. Storage Virtualization.
5. Server Virtualization.
6. Data virtualization.
7. Hardware virtualization.

1. Application Virtualization

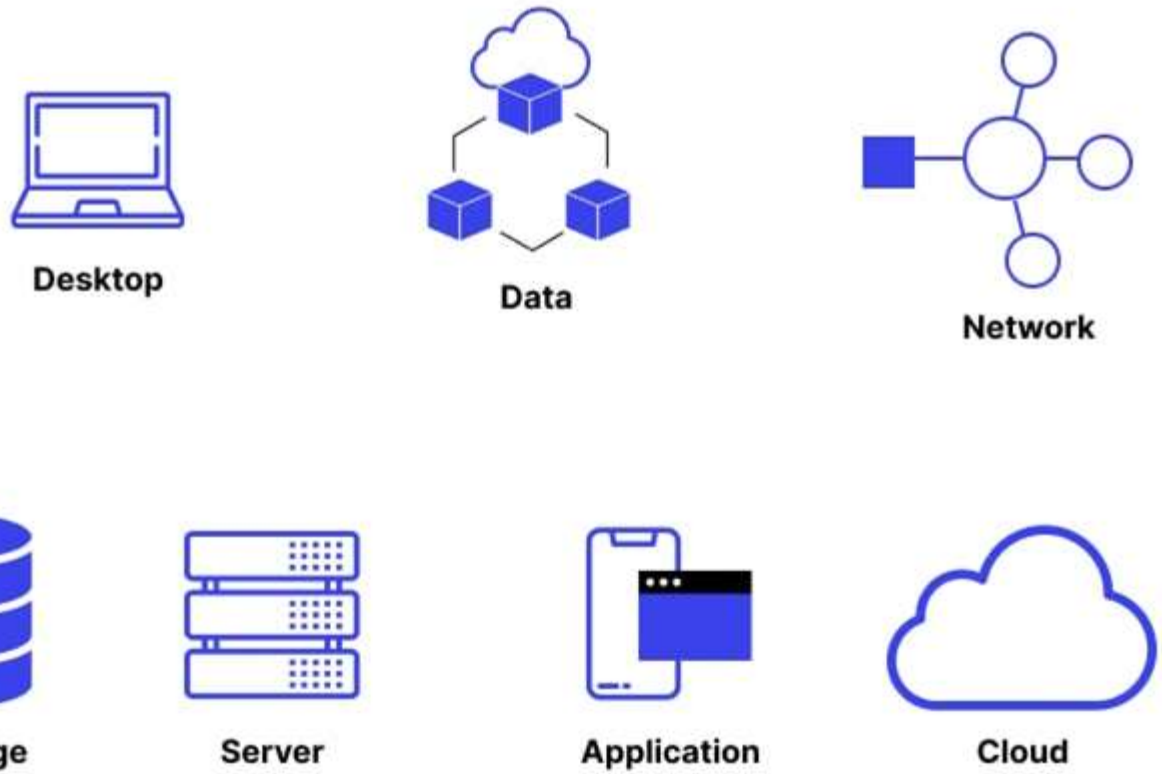
- Application virtualization helps a user to have remote access of an application from a server.
- The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet.
- Example of this would be a user who needs to run two different versions of the same software.
- Technologies that use application virtualization are hosted applications and packaged applications.


Packaging
Stored
Streaming
Virtualized

benefits of virtualized applications are:

- **Simplified management**
- **Scalability**
- **Security**
- **Simple Installation**
- **Easy deployment**
- **Better Portability**
- **Simple to get rid of applications**
- **Reduced conflicts between applications**
- **Easier Rollback**
- **Improved Security**
- **Easier updates**
- **Simplified Support**
- **Independence from the Operating System**

Types of Application virtualization



Drawbacks of Application Virtualization

- Graphics-intensive applications can **slow down** (lag) during the rendering process.
- A steady and reliable connection to the server is required to provide users with a **solid UX** with the applications.
- Some applications require **device drivers and 16-bit applications running** in the memory.
- Some applications, such as **anti-virus programs, must integrate with the local OS**, as they require continuous access to local data.
- The use of peripheral devices, like printers, can get more complicated with app virtualization.
- System monitoring tools can have trouble with virtualized applications, making it tricky to troubleshoot and isolate performance concerns.

Use Cases for Application Virtualization

- **Cost Control:** If you have a huge number of employees or end-users, then purchasing expensive PCs for everyone can turn out to be drastically expensive. Application virtualization comes to the rescue in such a situation as it allows you to deliver critical applications to any endpoint.
- **Application Mobility:** Enterprise applications should be accessible from any kind of mobile device for ease of use. Application virtualization offers application mobility by allowing applications to be delivered to any endpoint.
- **Secure Remote Access Capabilities:** Application virtualization allows employees to access critical applications from anywhere and that too in a secure manner. Application virtualization is useful for work-from-home scenarios that not only provide ease but also security.
- **Simplified Migrations:** Since application virtualization separates applications from the underlying operating system, there is no need to carry out extensive migrations from one kind of OS to the other.
- **Deploying In-House Applications :** Another important use case of application virtualization is the deployment of in-house applications which are updated frequently by developers. The updates, installation, and delivery of these applications are made remote and quick using application virtualization. Application virtualization is equally important for organizations that deploy in-house applications.

2. Network Virtualization

- The ability to run multiple virtual networks with each has a separate control and data plan.
- It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.
- **Network Virtualization** (NV) refers to abstracting network resources that were traditionally delivered in hardware to software. NV can combine multiple physical networks to one virtual, software-based network, or it can divide one physical network into separate, independent virtual networks.
- Network virtualization software allows network administrators to move virtual machines across different domains without reconfiguring the network. The software creates a network overlay that can run separate virtual network layers on top of the same physical network fabric.

2. Network Virtualization

- **Tools for Network Virtualization :**
- **Physical switch OS** – It is where the OS must have the functionality of network virtualization.
- **Hypervisor –** It is which uses third-party software or built-in networking and the functionalities of network virtualization.
- The basic functionality of the OS is to give the application or the executing process with a simple set of instructions. System calls that are generated by the OS and executed through the libc library are comparable to the service primitives given at the interface between the application and the network through the SAP (Service Access Point).
- The hypervisor is used to create a virtual switch and configuring virtual networks on it. The third-party software is installed onto the hypervisor and it replaces the native networking functionality of the hypervisor. A hypervisor allows us to have various VMs all working optimally on a single piece of computer hardware.

How does network virtualization work?

- Network virtualization **decouples network services from the underlying hardware and allows virtual provisioning of an entire network**. It makes it possible to programmatically **create, provision, and manage** networks all in software, while continuing to leverage the underlying physical network as the **packet-forwarding backplane**.
- Physical network resources, such as switching, routing, firewalling, load balancing, virtual private networks (VPNs), and more, are pooled, delivered in software, and require only Internet Protocol (IP) packet forwarding from the underlying physical network.
- **Network and security services in software are distributed to a virtual layer** (hypervisors, in the data center) and “attached” to individual workloads, such as your virtual machines (VMs) or containers, in accordance with networking and security policies defined for each connected application.
- **When a workload is moved to another host, network services and security policies move with it**. And when new workloads are created to scale an application, necessary policies are dynamically applied to these new workloads, providing greater policy consistency and network agility.

Advantages of Network Virtualization

- Improves manageability
- Reduces CAPEX
- Improves utilization
- Enhances performance
- Enhances security

Disadvantages of Network Virtualization

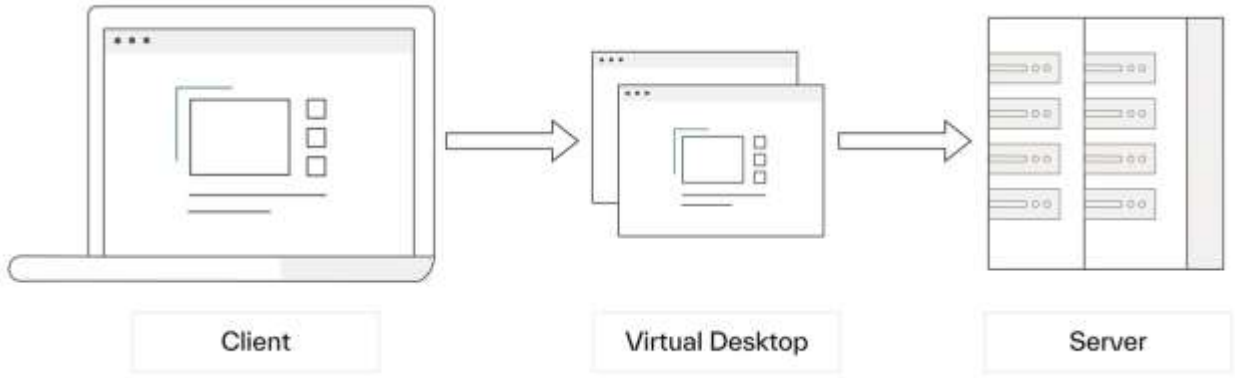
- It needs to manage IT in the abstract.
- It needs to coexist with physical devices in a cloud-integrated hybrid environment.
- Increased complexity.
- Upfront cost.
- Possible learning curve.

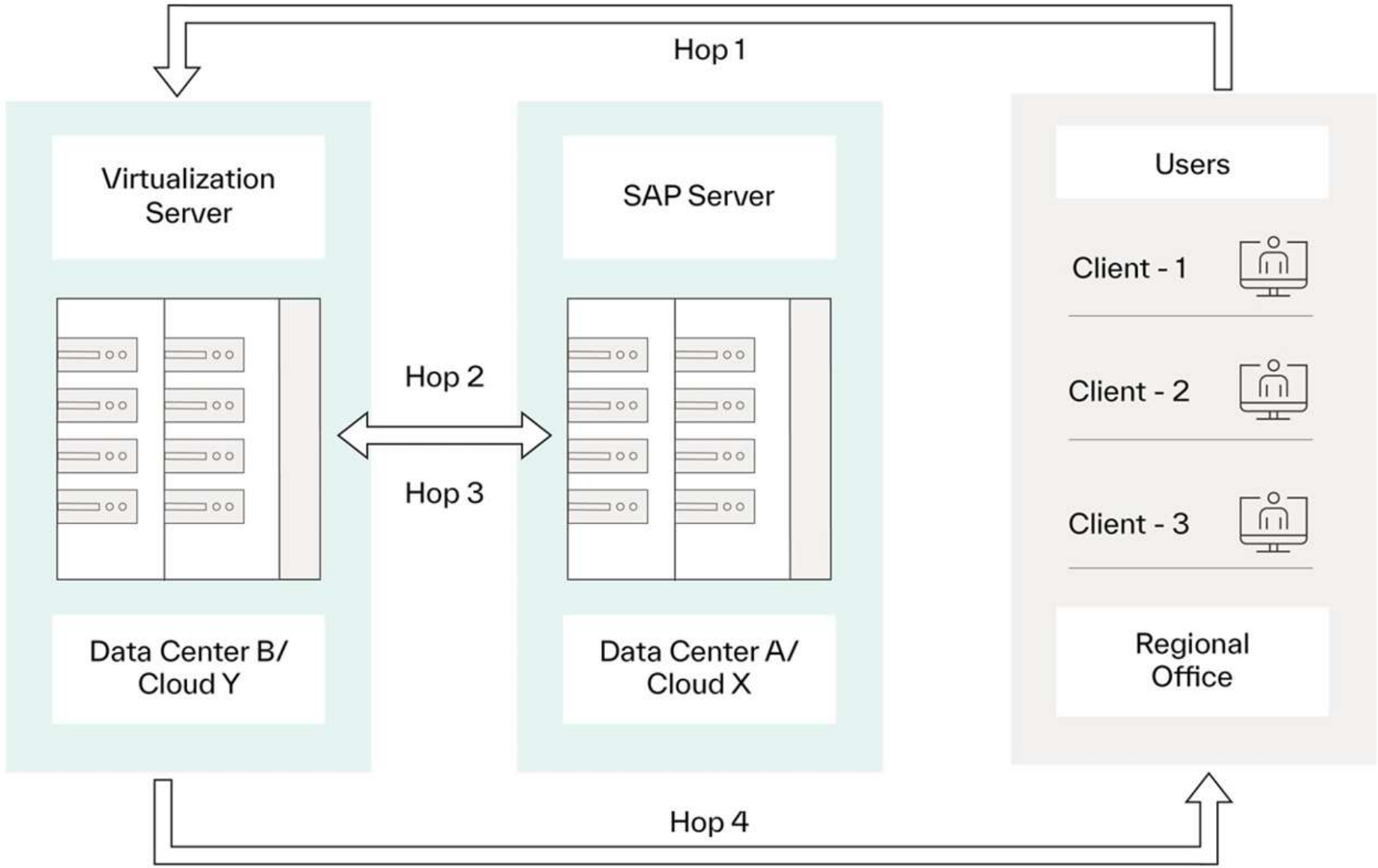
Applications of Network Virtualization

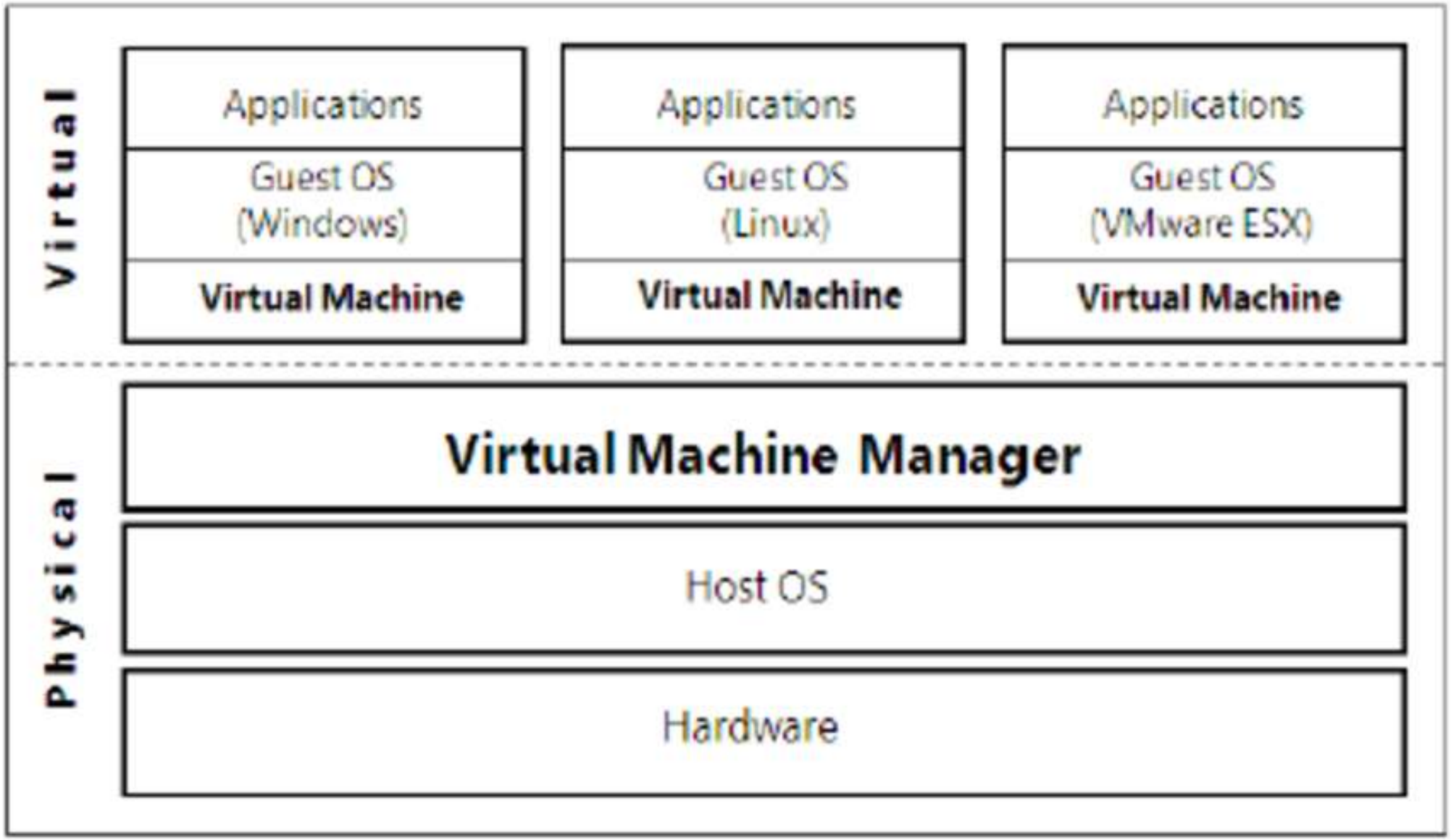
- application testing to mimic real-world hardware and system software.
- It helps us to integrate several physical networks into a single network or separate single physical networks into multiple analytical networks.
- simulation of connections between applications, services, dependencies, and end-users for software testing.
- to deploy applications in a quicker time frame, thereby supporting a faster go-to-market.

3. Desktop Virtualization

- Desktop virtualization allows the users' OS to be remotely stored on a server in the data centre.
- It allows the user to access their desktop virtually, from any location by a different machine.
- Users who want specific operating systems other than Windows Server will need to have a virtual desktop.
- Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates, and patches.




Hop 1
Virtualization Server
SAP Server
Users
Client - 1
Client - 2
Client - 3
Hop 2
Hop 3
Data Center B/ Cloud Y
Data Center A/ Cloud X
Regional Office
Hop 4


Virtual
Applications
Guest OS
(Windows)
Virtual Machine
Applications
Guest OS
(Linux)
Virtual Machine
Applications
Guest OS
(VMware ESX)
Virtual Machine
Physical
Virtual Machine Manager
Host OS
Hardware

Types of Desktop Virtualization

- Hosted Desktop Virtualization: In this desktop virtualization model, a computer server living in a data center **basically hosts the virtual machines, enabling users to connect to the server via standard protocols** like Remote Desktop Protocol (RDP) or connection brokers.
- Hosted desktop virtualization is executed in three main formats:

1. Virtual Desktop Infrastructure (VDI): Here, **the operating system (OS) operates a virtual machine that contains desktop images on a server**. Consequently, a **hypervisor is employed to split the server into disparate desktop images** that users can remotely access through their endpoint devices. (Server on server)

2. Remote Desktop Services (RDS): This variation of hosted desktop virtualization provides **users remote access via shared desktops and applications on Microsoft Windows Server OS (CRM)**.

3. Desktop-as-a-Service (DaaS): This variation operates in a manner similar to that of VDI, as **end-users can access their desktops and computer applications from any endpoint device or platform**. However, the key difference with DaaS is that one has to purchase, deploy, and manage all the hardware components themselves.

Continued..

- Client Virtualization
- **This desktop virtualization model revolves around the installation of a hypervisor on a client device to run multiple operating systems**, thus eliminating the need for users to maintain their own dedicated hardware and software.
- Principally, client virtualization deployment has two key variants:

1. Presentation virtualization
This avails a web-based portal that users can leverage to interact with desktops and apps.

2. Application virtualization
This client virtualization approach enables apps to run on other platforms. For instance, running Windows apps on the Linux OS.

Advantages of Desktop Virtualization

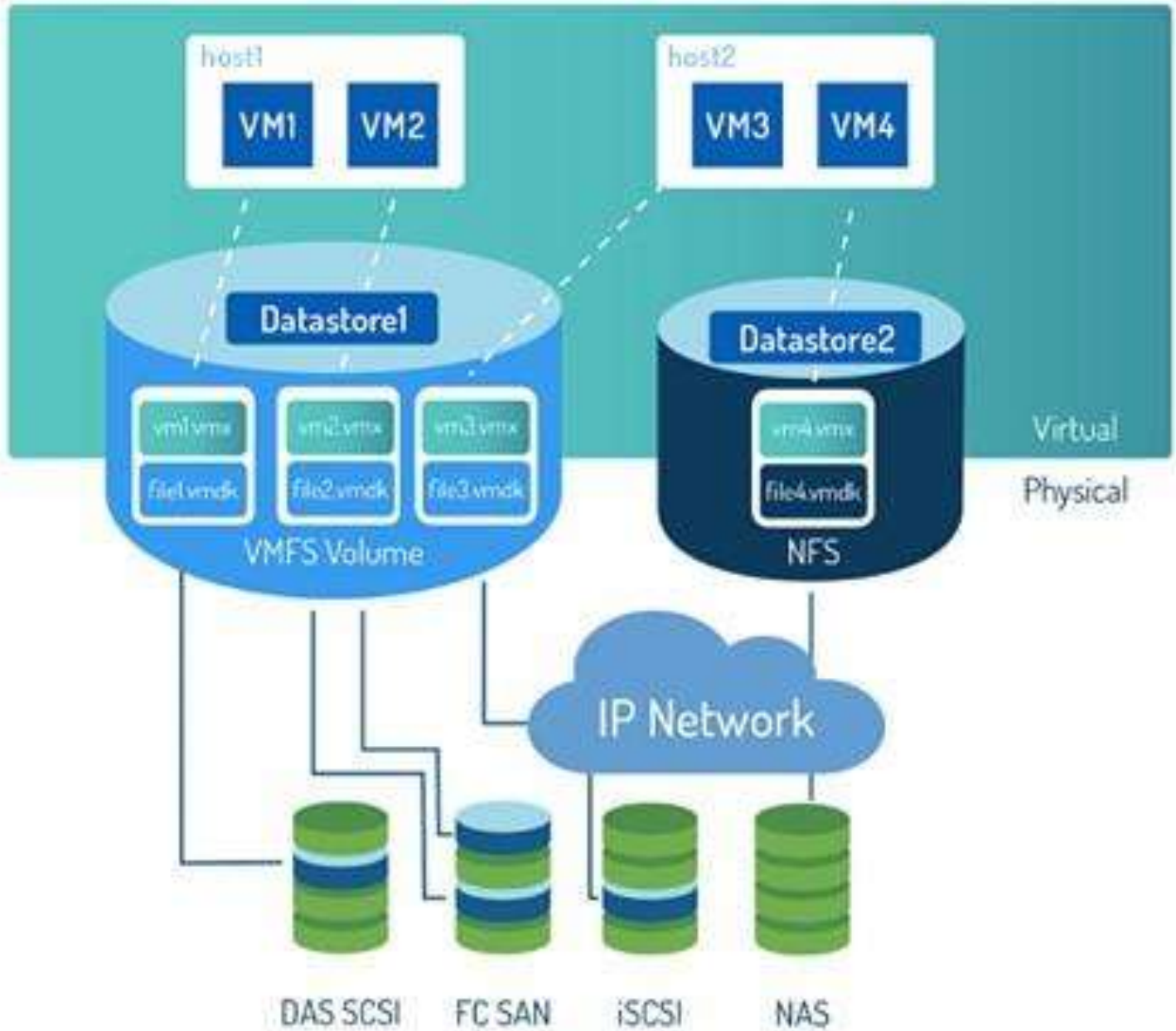
1. **Flexibility**
2. **Cost efficiency**
3. **Enhanced security**
4. **Environmentally Friendly**
5. **Centralized management**
6. **Disaster recovery**
7. **Increase Employee Productivity and Onboarding**

Disadvantages of Desktop Virtualization

- Desktop Virtualization is **Cap-ex intensive**. One needs to buy the Desktop Virtualization Software/Licenses, Servers, Centralized Storage infrastructure, Upgrade Network infrastructure to support more bandwidth, etc in addition to buying computers/ thin-clients for each user.
- There is **no reduction** in the number of end-user client machines (computers) that are needed in the network.
- The **licenses** for Operating Systems, applications etc, still needs to be bought for each user (mostly) and there is no reduction of costs there.
- The **thin-clients** are sometimes as expensive/ more expensive than individual computers as, with huge volumes computer prices plummet drastically as they are manufactured and distributed in bulk quantities.
- The network infrastructure needs to handle all that **extra bandwidth** that Desktop Virtualization is going to introduce. Otherwise, it has to be upgraded. The WAN links need to have sufficient bandwidth to handle all those remote DV users, as well.
- If the bandwidth on the remote end is not sufficient/ if there is congestion in LAN, the **display quality** may not be as good (when images are streamed from server) as processing and viewing applications right from a desktop.
- Its difficult to handle **graphics/ high-definition video** with Desktop Virtualization. But there are some work-around methods that vendors follow to overcome this limitation (Including having local graphic acceleration cards, rendering graphical applications on the desktop, etc).
- Some vendor Desktop Virtualization solutions work only with their **Server Virtualization** counterparts, hence limiting the choices for the customers.
- There is a limit to the **number of Operating Systems** that can be supported by Desktop Virtualization products.

4. Storage Virtualization

- Storage virtualization is an **array of servers that are managed by a virtual storage system**.
- The servers aren’t aware of exactly where their data is stored, and instead function more like worker bees in a hive.
- It makes **managing storage from multiple sources to be managed and utilized** as a single repository.
- Storage virtualization software **maintains smooth operations, consistent performance and a continuous suite of advanced functions** despite changes, break down and differences in the underlying equipment.


host1
VM1
VM2
host2
VM3
VM4
Datastore1
Datastore2
vm1.vmx
file1.vmdk
vm2.vmx
file2.vmdk
vm3.vmx
file3.vmdk
vm4.vmx
file4.vmdk
VMFS Volume
NFS
Virtual
Physical
IP Network
DAS SCSI
FC SAN
iSCSI
NAS

Types of Storage Virtualization

- Object-Level: It's been abstracted into data buckets. API calls from application are used to retrieve this data. This may be a more scalable alternative than block storage for big volumes of data.
- Block-Level: This method operates at a lower level than the file system, presenting virtualized storage in the form of logical block devices (virtual disks) to servers. It aggregates storage from various physical disks into a single, unified pool that servers can access as if they were dedicated volumes, abstracting the underlying physical structure.
- File-Level: This type works above the file system, providing a centralized, unified view of files and directories spread across multiple physical file servers. Users and applications interact with a single, logical file structure without needing to know where the actual files are stored.

Advantages of Storage Virtualization

- It allows for migrations to be performed quickly.
- It creates better workflows.
- It allows more than one type of storage array.
- It is a cheaper option for storage.
- It allows for the costs to become more predictable.
- It provides better access to your data.
- It allows anyone to create business opportunities for themselves.

Disadvantages of Storage Virtualization

- It requires you to deal with multiple vendors.
- It can make upgrades challenging to process.
- It does not always scale to some areas.
- It still has limitations which must be considered.
- It does not eliminate data security risks.
- It may create availability issues.
- It creates more links for data access instead of less

5. Server Virtualization

- This is a kind of virtualization in which **masking of server resources takes place**.
- The **central-server(physical server) is divided into multiple different virtual servers by changing the identity number, processors**.
- So, **each system can operate its own operating systems in isolate manner**. Where each sub-server knows the identity of the central server.
- It causes **an increase in the performance** and **reduces the operating cost** by the deployment of main server resources into a sub-server resource.
- It's **beneficial in virtual migration, reduce energy consumption, reduce infrastructural cost**, etc.

Server virtualization components

Depending on the approach, server virtualization uses a number of different components, including a host machine, VMs, hypervisor, and containers. Here are all the components of a typical virtual server:

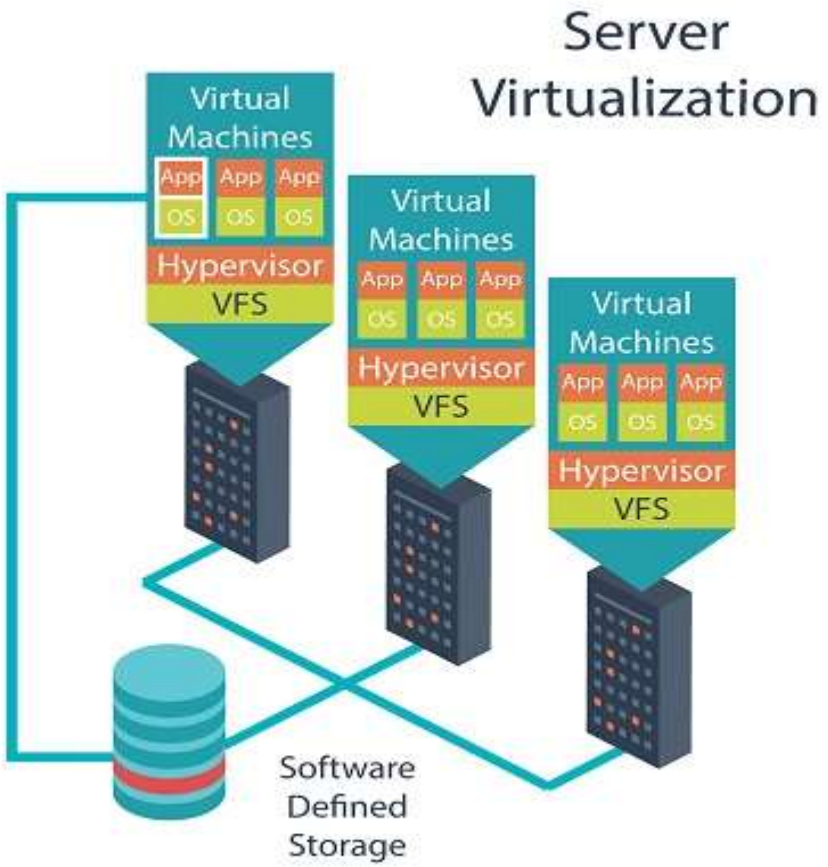
- A **host machine**, which is the physical server hardware where virtualization occurs.
- **Virtual machines (VMs)**, which contain the assets that are abstracted from a traditional server environment.
- A **hypervisor**, which is a specialized software that creates and maintains VMs and can be run natively on bare metal servers or hosted on top of an existing OS.
- **Hypercalls**, which are messages sent between paravirtualized hypervisors and OSs to share resources using an API.
- **Containers**, which are unique user environments created in virtualized OSs. With a container engine, multiple containers can make use of the same interfaces and shared libraries of the underlying host OS. Containers are often deployed inside of hypervisors or VMs to offer an additional layer of isolation from the server’s core host OS.

Benefits of Server Virtualization

- Higher server ability
- Cheaper operating costs
- Eliminate server complexity
- Increased application performance
- Deploy workload quicker

Types of Server Virtualization

- **Full Virtualization:** Full virtualization uses a hypervisor, a type of software that directly communicates with a physical server's disk space and CPU. The hypervisor monitors the physical server's resources and keeps each virtual server independent and unaware of the other virtual servers. It also relays resources from the physical server to the correct virtual server as it runs applications. The biggest limitation of using full virtualization is that a hypervisor has its own processing needs. This can slow down applications and impact server performance. Microsoft Hyper-V, Oracle VM VirtualBox, and VMware vSphere are among the leading full virtualization products.
- **Para-Virtualization:** Unlike full virtualization, para-virtualization involves the entire network working together as a cohesive unit. Paravirtualization uses an application programming interface (API) to send hypercalls between the hypervisor and OSs. This means that each VM is aware of and can communicate with one another to share resources. Paravirtualization, sometimes referred to simply as PV, can run on system architectures that do not have hardware-assisted virtualization support. Most products that enable full virtualization also enable paravirtualization.
- **OS-Level Virtualization:** Unlike full and para-virtualization, OS-level visualization does not use a hypervisor. the physical server's OS contains a virtualization capability that acts like a hypervisor to create multiple user environments called containers. However, all the virtual servers must run that same operating system in this server virtualization method. Oracle Solaris is one of the most prominent OS-level virtualization products.



The server administrator uses a software application to divide one physical server into multiple isolated virtual environments. The virtual environments are sometimes called virtual private servers, but they are also known as guests, instances, containers or emulations.

popular approaches to server virtualization:

1. **Virtual machines are based on the host/guest paradigm:**

- Each guest runs on a virtual imitation of the hardware layer.
- This approach allows the guest operating system to run without modifications.
- It also allows the administrator to create guests that use different operating systems. The guest has no knowledge of the host's operating system because it is not aware that it's not running on real hardware. It does, however, require real computing resources from the host — so it uses a hypervisor to coordinate instructions to the CPU.
- The hypervisor is called a virtual machine monitor (VMM). It validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges.
- VMware and Microsoft Virtual Server both use the virtual machine model.

popular approaches to server virtualization:

2. **The paravirtual machine (PVM) model is also based on the host/guest paradigm**

- It uses a virtual machine monitor too.
- In the paravirtual machine model, however, The VMM actually modifies the guest operating system's code. This modification is called porting.
- Porting supports the VMM so it can utilize privileged systems calls sparingly.
- Like virtual machines, paravirtual machines are capable of running multiple operating systems.
- Xen and UML both use the paravirtual machine model.

popular approaches to server virtualization:

3. **Virtualization at the OS level**

- It isn’t based on the host/guest paradigm.
- In the OS level model, the host runs a single OS kernel as its core and exports operating system functionality to each of the guests.
- Guests must use the same operating system as the host, although different distributions of the same system are allowed.
- This distributed architecture eliminates system calls between layers, which reduces CPU usage overhead.
- It also requires that each partition remain strictly isolated from its neighbors so that a failure or security breach in one partition isn’t able to affect any of the other partitions.
- In this model, common binaries and libraries on the same physical machine can be shared, allowing an OS level virtual server to host thousands of guests at the same time.
- Virtuozzo and Solaris Zones both use OS-level virtualization.

Advantage of Server Virtualization

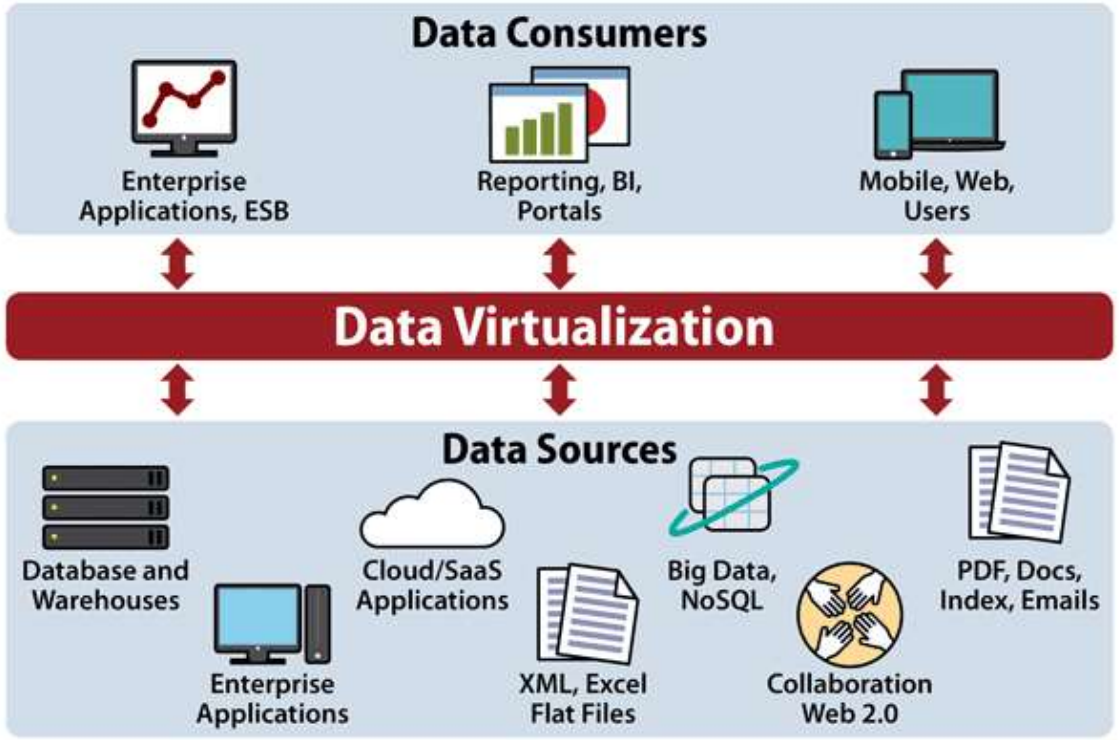
- Cost reduction: This cost reduction can be realized in four distinct areas:
- **1. Hardware savings.** Without the need to purchase or upgrade costly server hardware, companies can reallocate their funds back into growing their business.
- **2. Operational savings.** With a large server array, it is necessary to employ dedicated technicians to maintain it all. Salaries for full-time tech staff are a considerable drain on spending for many companies, but with virtualization, these resources can be reduced to part-time or even channeled into less-costly **managed services**.
- **3. Energy savings.** An extensive collection of physical servers is a massive drain on energy resources. Not only do they need to be kept running, they need to be kept cool—meaning there will be extra energy involved to maintain climate stability in the server room. Virtualization eliminates this need, potentially reducing your energy consumption by a significant degree.
- **4. Real estate savings.**
- Virtualization promotes automation
- Virtualization makes backup and recovery more efficient and reliable
- Cyberattacks, power surges, and **terrestrial disasters**
- Server redundancies reduce downtime to a minimum and prevent outages
- Virtualization supports scale

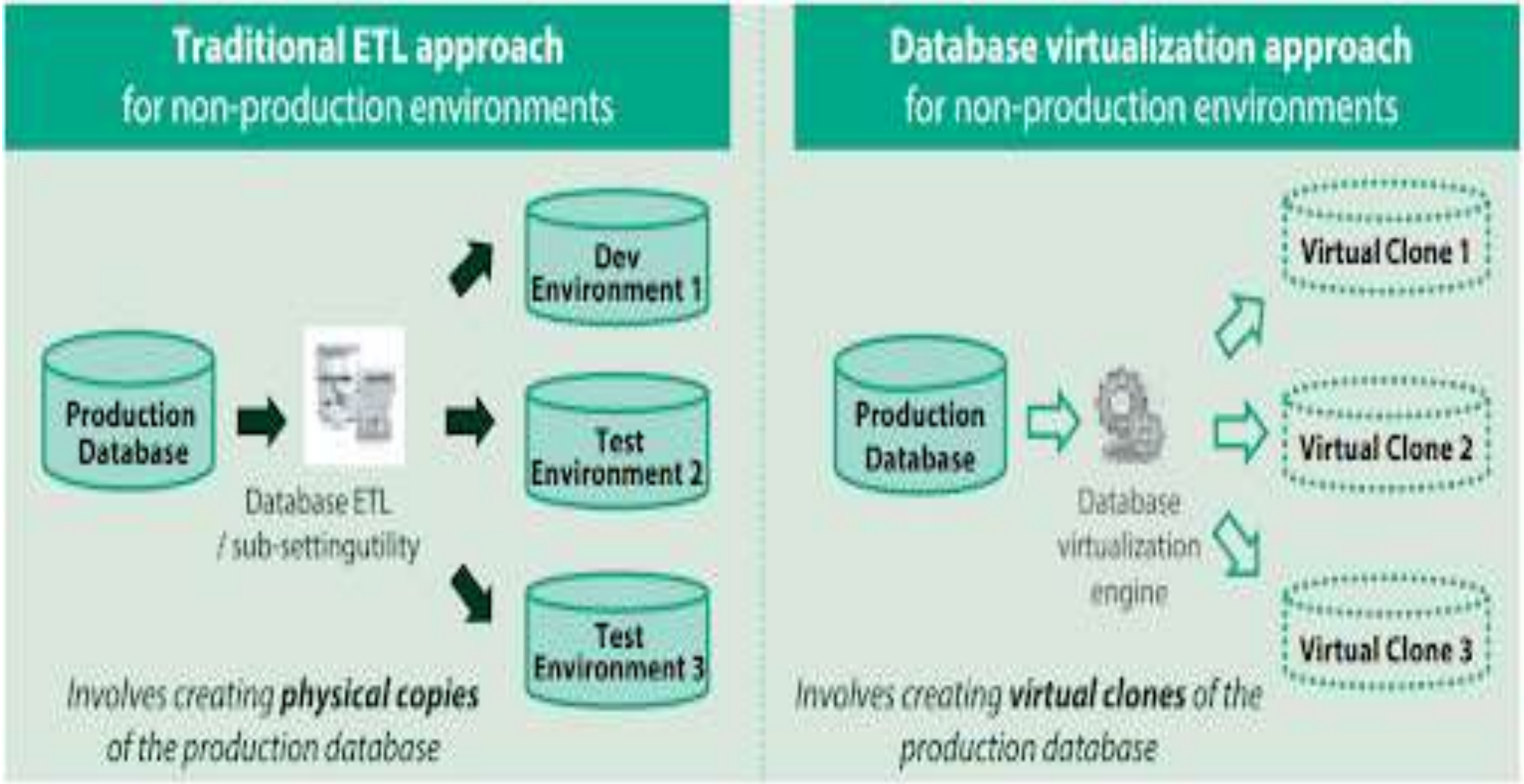
Disadvantages of server virtualization

- **The cost of entry can be prohibitive**
- Virtualization, like any other technological initiative, is **pay-to-play**. For instance, the physical servers that can be virtualized cost more than their traditional counterparts.
- You will also **bear the cost of software licensing.** Fortunately, the cost savings you will see as a result of virtualization should balance that cost out in the end.
- **Not all applications can be virtualized :** You may still need to maintain a hybrid system to ensure all of your applications keep working as they should. Today, **most applications support virtualization**, but if you are running proprietary software, you may want to look at its capabilities before moving forward.
- **Security risks:** Data security is one of the **most significant issues** we face today. Virtualization carries an added security risk, so additional spending will be required to ensure **data safety and integrity**. This can extend to insurance, security hardware, software, and a more robust monitoring policy.

Data virtualization

- This is the kind of virtualization in which the data is collected from various sources and managed that at a single place without knowing more about the technical information like how data is collected, stored & formatted then arranged that data logically so that its virtual view can be accessed by its interested people and stakeholders, and users through the various cloud services remotely.
- Many big giant companies are providing their services like Oracle, IBM, At scale, Cdata, etc.





- **Data virtualization is a virtualized architecture layer that “sits” on top of those data sources and connects them.**

Advantages of data virtualization -

- It allows users to access the data without worrying about where it resides on the memory.
- It offers better customer satisfaction, retention, and revenue growth.
- It provides various security mechanism that allows users to safely store their personal and professional information.
- It reduces costs by removing data replication.
- It provides a user-friendly interface to develop customized views.
- It provides various simple and fast deployment resources.
- It increases business user efficiency by providing data in real-time.
- It is used to perform tasks such as data integration, business integration, Service-Oriented Architecture (SOA) data services, and enterprise search

Disadvantages of Data Virtualization

- It creates availability issues, because availability is maintained by third-party providers.
- It required a high implementation cost.
- It creates the availability and scalability issues.
- Although it saves time during the implementation phase of virtualization but it consumes more time to generate the appropriate result.

Use of data virtualization

- Analyze performance: Data virtualization is used to analyze the performance of the organization compared to previous years.
- Search and discover interrelated data: Data Virtualization (DV) provides a mechanism to easily search the data which is similar and internally related to each other.
- Agile Business Intelligence: It is one of the most common uses of Data Virtualization. It is used in agile reporting, real-time dashboards that require timely aggregation, analyze and present the relevant data from multiple resources. Both individuals and managers use this to monitor performance, which helps to make daily operational decision processes such as sales, support, finance, logistics, legal, and compliance.
- Data Management: Data virtualization provides a secure centralized layer to search, discover, and govern the unified data and its relationships.

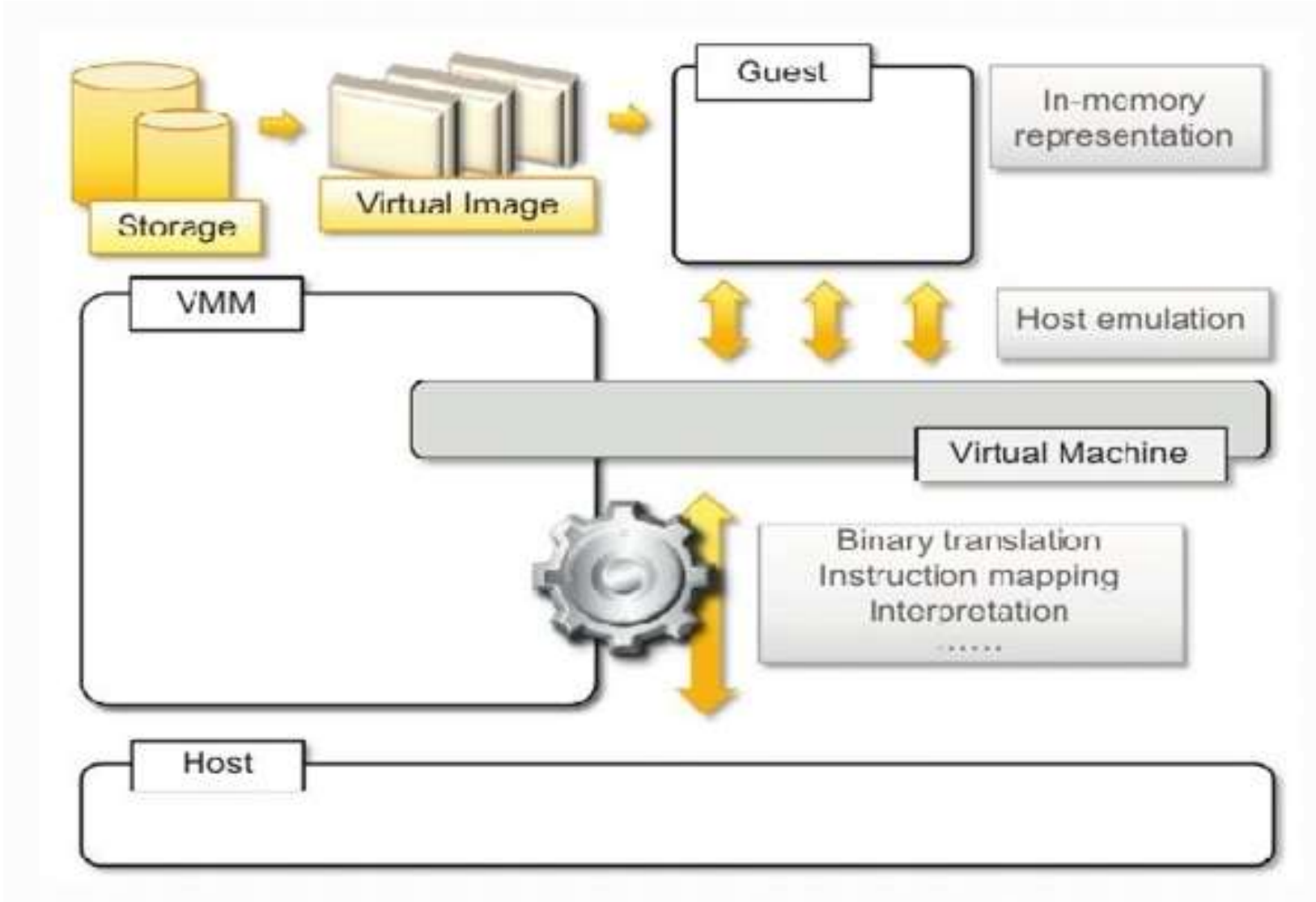
Industries that use Data Virtualization

- **Communication & Technology**
  In Communication & Technology industry, data virtualization is used to increase revenue per customer, create a real-time ODS for marketing, manage customers, improve customer insights, and optimize customer care, etc.
- **Finance**
  In the field of finance, DV is used to improve trade reconciliation, empowering data democracy, addressing data complexity, and managing fixed-risk income.
- **Government**
  In the government sector, DV is used for protecting the environment.
- **Healthcare**
  Data virtualization plays a very important role in the field of healthcare. In healthcare, DV helps to improve patient care, drive new product innovation, accelerating M&A synergies, and provide a more efficient claims analysis.
- **Manufacturing**
  In manufacturing industry, data virtualization is used to optimize a global supply chain, optimize factories, and improve IT assets utilization.

7. **Hardware Virtualization**

- In Cloud Computing, hardware virtualization is used in server platforms since it offers more flexibility as opposed to physical machines.
- When it comes to hardware virtualization, VM software gets installed within the hardware system, known as hardware virtualization.
- It also comprises a hypervisor that controls and monitors the process, hardware resources and memory of the system.
- After the completion of the hardware virtualization process, the concerned user can install a different OS in it and different applications can be used simultaneously.

Hardware Virtualization



Advantages of Hardware Virtualization

- It reduces the maintenance overhead of para-virtualization as it reduces (ideally, eliminates) the modification in the guest operating system.
- It is also significantly convenient to attain enhanced performance.
- A practical benefit of hardware-based virtualization has been mentioned by VMware engineers and Virtual Iron.

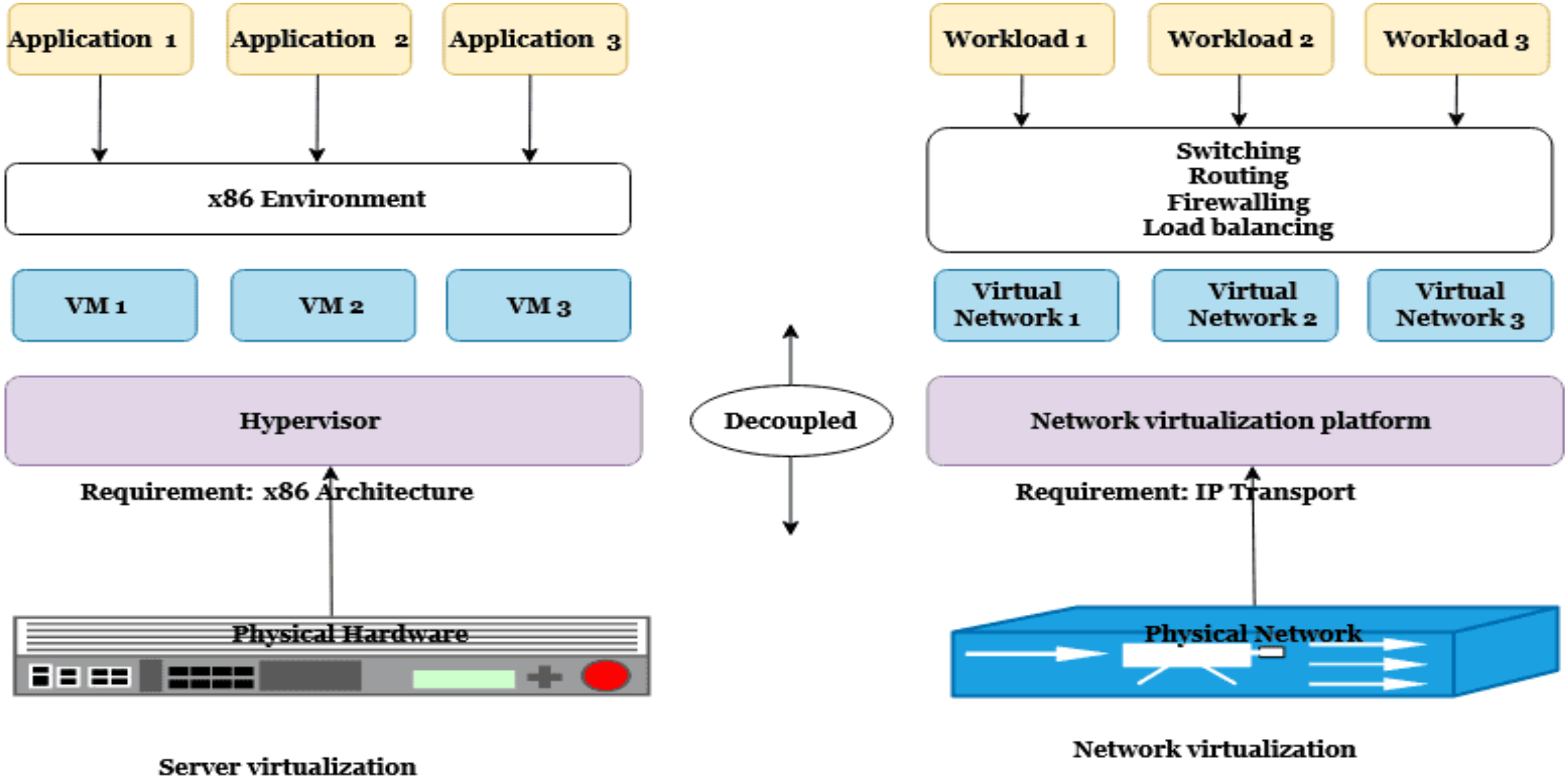
Disadvantages of hardware-based virtualization

Hardware-based virtualization requires explicit support in the host CPU, which may not available on all x86/x86_64 processors.

A “pure” hardware-based virtualization approach, including the entire unmodified guest operating system, involves many VM traps, and thus a rapid increase in CPU overhead occurs which limits the scalability and efficiency of server consolidation.

This performance hit can be mitigated by the use of para-virtualized drivers; the combination has been called “hybrid virtualization”.

Server Vs Network virtualization



App vs. Desktop virtualization

- When it comes to virtualizing for desktops, it is more related to **a remote environment**. **Even if users change tangible OSs, they can continue to utilize similar functionalities by remotely accessing their virtual network through the Web**.
- **No matter what device is used, the screen is reflected to show the identical situation they might have.** This is so that only the user's device may see the **customer's data**, which is stored on a web server.
- On the contrary, **application virtualization solutions are isolated from the OS as well as the hardware it uses to function**.
- Instead of maintaining and executing distinct versions on separate devices, this **isolation enables programs to be operated from network infrastructure hardware**. Users in these situations often connect to the central data database over the Web to view their programs and related data.

App vs. Server virtualization

- As mentioned, virtualization procedures can happen on various levels, including desktop, server, and application levels. Hence, in the case of servers, it is the most frequently used virtualization strategy in the modern technology sector.
- Businesses may benefit immensely from using server virtualization as it isolates the parts of the host operating system to analyze the records more efficiently, which boosts system performance.

Why should virtualization be even considered?

- Machines can run multiple instances of a single application simultaneously.
- Streamlining the IT cost and minimizing the IT administration structure.
- Effective management of large scale installation and server farms.
- Enhanced reliability, security, scalability, device dependence.
- Options to enable multiple OS use on the same hardware.

Terms that are associated with virtualization

- **Hypervisor :** It is the OS that runs on actual hardware and the Virtual counterpart is a part of this OS as a running process. Hypervisors are often seen as Domain 0 or Dom0.
- **Virtual Machine (VM):** It is a virtual computer that runs under a hypervisor.
- **Container :** These are light-weight VMs that are part of the same OS instance as its hypervisor. So, containers are nothing but a group of processes that are running with their respective namespace for process identifiers.
- **Virtualization Software :** It is a software that aids in implementing virtualization on any computer. It can either be a part of a software application package or an OS or a special variant of that OS.
- **Virtual Network :** Virtual Network is a logically separate network within servers that can be extended to other servers or across multiple servers.

Implementation Levels of Virtualization

1) Instruction Set Architecture Level (ISA)

- ISA virtualization can work through ISA emulation. This is used to run many legacy codes that were written for a different configuration of hardware. These codes run on any virtual machine using the ISA.
- With this, a binary code that originally needed some additional layers to run is now capable of running on the x86 machines. It can also be tweaked to run on the x64 machine. With ISA, it is possible to make the virtual machine hardware agnostic.
- For the basic emulation, an interpreter is needed, which interprets the source code and then converts it into a hardware format that can be read. This then allows processing. This is one of the five implementation levels of virtualization in cloud computing.

Implementation Levels of Virtualization

2) Hardware Abstraction Level (HAL)

- True to its name HAL lets the virtualization perform at the level of the hardware. This makes use of a hypervisor which is used for functioning.
- At this level, the virtual machine is formed, and this manages the hardware using the process of virtualization. It allows the virtualization of each of the hardware components, which could be the input-output device, the memory, the processor, etc.
- Multiple users will not be able to use the same hardware and also use multiple virtualization instances at the very same time. This is mostly used in the cloud-based infrastructure.

Implementation Levels of Virtualization

3) Operating System Level

- At the level of the operating system, the virtualization model is capable of creating a layer that is abstract between the operating system and the application.
- This is an isolated container that is on the operating system and the physical server, which makes use of the software and hardware. Each of these then functions in the form of a server.
- When there are several users, and no one wants to share the hardware, then this is where the virtualization level is used. Every user will get his virtual environment using a virtual hardware resource that is dedicated. In this way, there is no question of any conflict.

Implementation Levels of Virtualization

4) Library Level

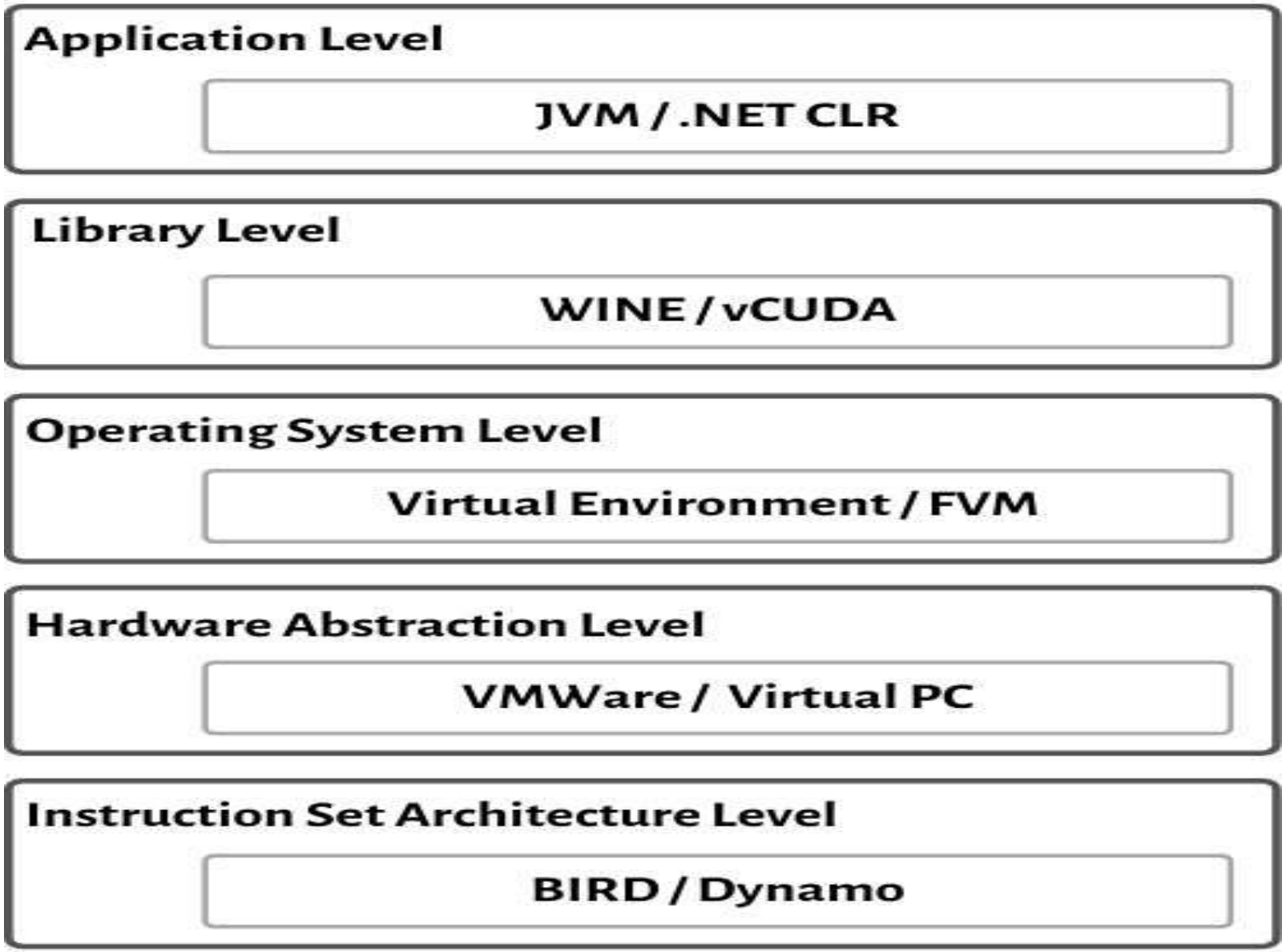
- The operating system is cumbersome, and this is when the applications make use of the API that is from the libraries at a user level.
- These APIs are documented well, and this is why the library virtualization level is preferred in these scenarios.
- API hooks make it possible as it controls the link of communication from the application to the system.

Implementation Levels of Virtualization

5) Application Level

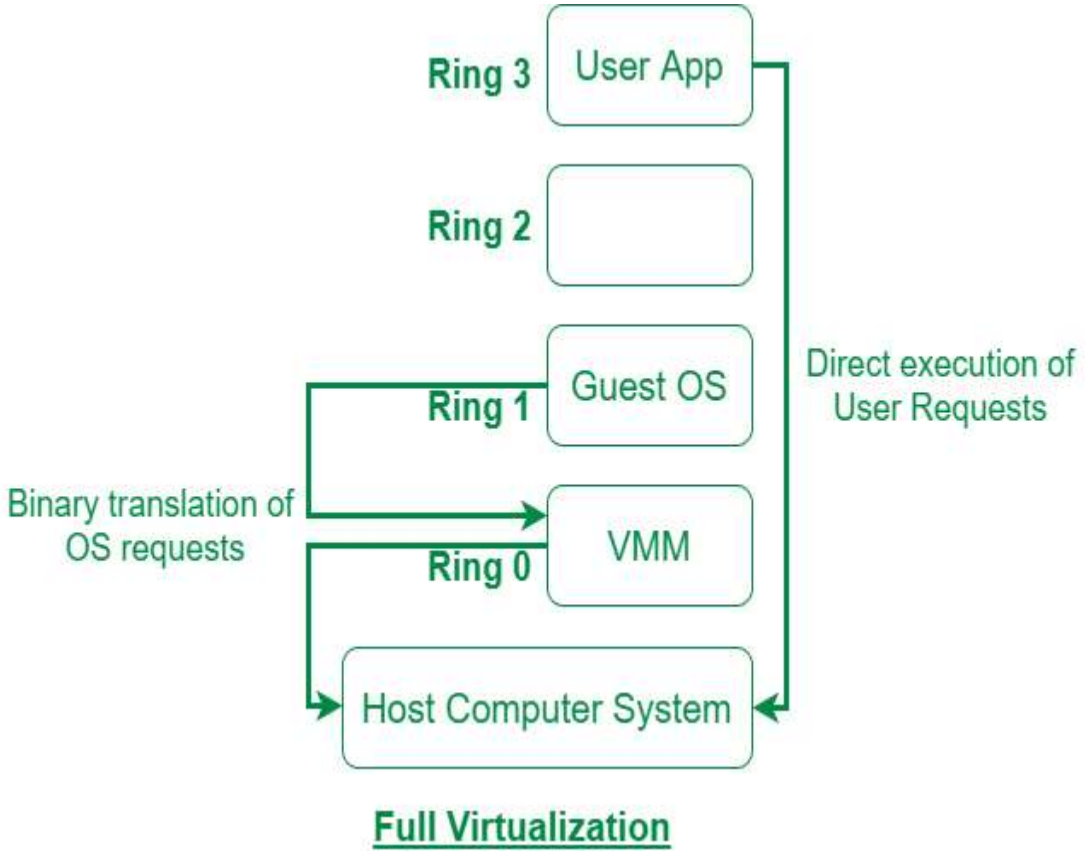
- The application-level virtualization is used when there is a desire to virtualize only one application and is the last of the implementation levels of virtualization in cloud computing. One does not need to virtualize the entire environment of the platform.
- This is generally used when you run virtual machines that use high-level languages. The application will sit above the virtualization layer, which in turn sits on the application program.
- It lets the high-level language programs compiled to be used in the application level of the virtual machine run seamlessly.

Five Levels of Virtualization



Type of Virtualization: Full Virtualization

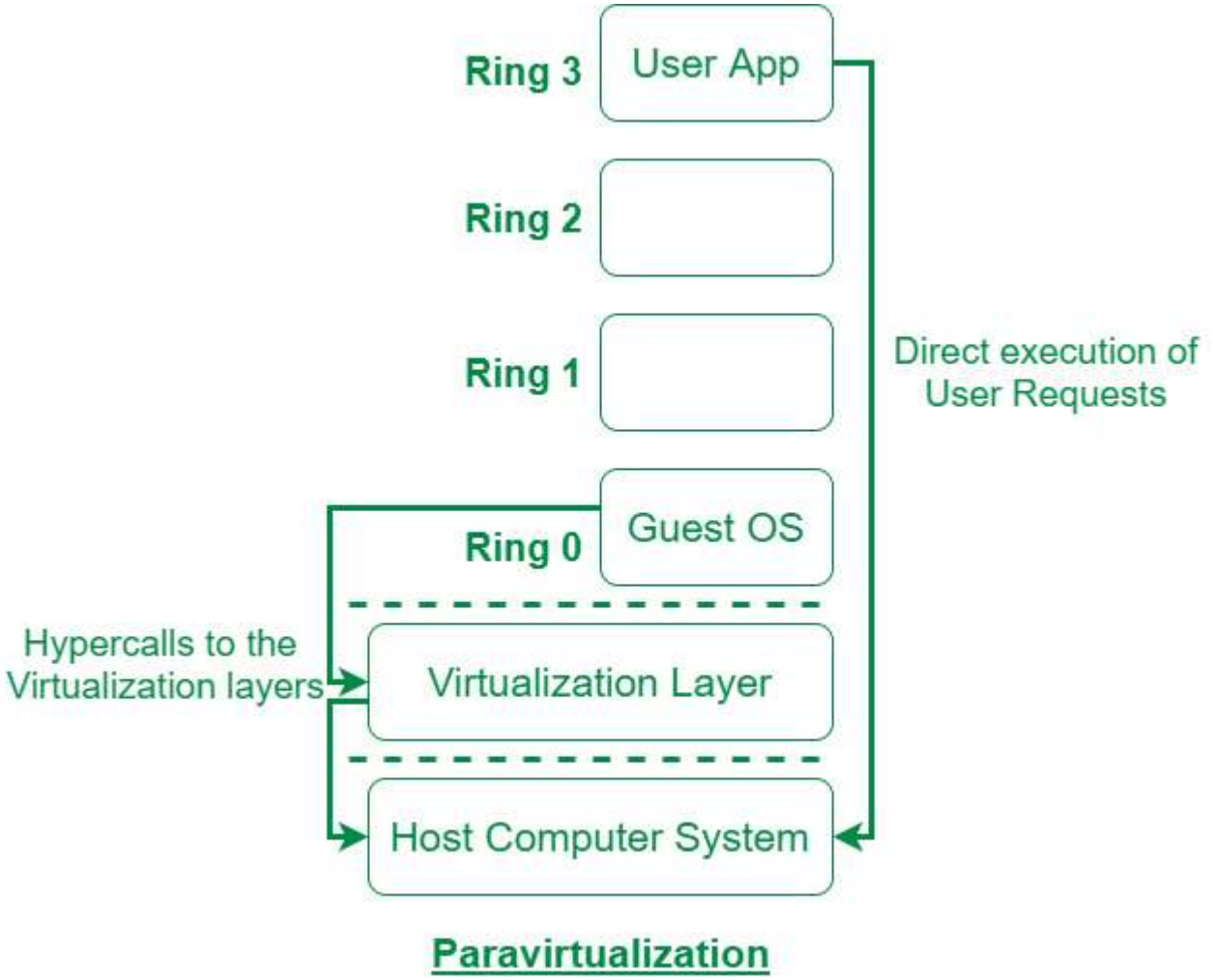
In full virtualization, guest OS is completely isolated by the virtual machine from the virtualization layer and hardware. Microsoft and Parallels systems are examples of full virtualization.



Full Virtualization

Type of Virtualization: Para Virtualization

Paravirtualization is the category of CPU virtualization which uses hypercalls for operations to handle instructions at compile time. In paravirtualization, guest OS is not completely isolated but it is partially isolated by the virtual machine from the virtualization layer and hardware. VMware and Xen are some examples of paravirtualization.



Difference between Full and Para Virtualization

S.No.	Full Virtualization	Paravirtualization
1.	In Full virtualization, virtual machines permit the execution of the instructions with the running of unmodified OS in an entirely isolated way.	In paravirtualization, a virtual machine does not implement full isolation of OS but rather provides a different API which is utilized when OS is subjected to alteration.
2.	Full Virtualization is less secure.	While the Paravirtualization is more secure than the Full Virtualization.
3.	Full Virtualization uses binary translation and a direct approach as a technique for operations.	While Paravirtualization uses hypercalls at compile time for operations.
4.	Full Virtualization is slow than paravirtualization in operation.	Paravirtualization is faster in operation as compared to full virtualization.
5.	Full Virtualization is more portable and compatible.	Paravirtualization is less portable and compatible.

Difference between Full and Para Virtualization

S.No.	Full Virtualization	Paravirtualization
6.	Examples of full virtualization are Microsoft and Parallels systems.	Examples of paravirtualization are Microsoft Hyper-V, Citrix Xen, etc.
7.	It supports all guest operating systems without modification.	The guest operating system has to be modified and only a few operating systems support it.
8.	The guest operating system will issue hardware calls.	Using the drivers, the guest operating system will directly communicate with the hypervisor.
9.	It is less streamlined compared to para-virtualization.	It is more streamlined.
10.	It provides the best isolation.	It provides less isolation compared to full virtualization.

Pros of Virtualization in Cloud Computing

- **Utilization of Hardware Efficiently –**
  With the help of Virtualization Hardware is Efficiently used by user as well as Cloud Service Provider. In this the need of Physical Hardware System for the User is decreases and this results in less costly.In Service Provider point of View, they will vitalize the Hardware using Hardware Virtualization which decrease the Hardware requirement from Vendor side which are provided to User is decreased.
- **Availability increases with Virtualization –**
  It allow virtual instances to be available all the times. It also has capability to move virtual instance from one virtual Server another Server which is very tedious and risky task in Server Based System. During migration of Data from one server to another it ensures its safety. Also, we can access information from any location and any time from any device.
- **Disaster Recovery is efficient and easy –**
  With the help of virtualization Data Recovery, Backup, Duplication becomes very easy. In traditional method , if somehow due to some disaster if Server system Damaged then the surety of Data Recovery is very less. But with the tools of Virtualization real time data backup recovery and mirroring become easy task and provide surety of zero percent data loss.

Pros of Virtualization in Cloud Computing

- **Virtualization saves Energy –**
  Virtualization will help to save Energy because while moving from physical Servers to Virtual Server's, the number of Server's decreases due to this monthly power and cooling cost decreases which will Save Money as well.
- **Quick and Easy Set up –**
  In traditional methods Setting up physical system and servers are very time-consuming. with the help of virtualization the entire process is done in very less time which results in productive setup.
- **Cloud Migration becomes easy –**
  Most of the companies those who already have spent a lot in the server have a doubt of Shifting to Cloud. But it is more cost-effective to shift to cloud services because all the data that is present in their server's can be easily migrated into the cloud server and save something from maintenance charge, power consumption, cooling cost, cost to Server Maintenance Engineer etc.

Cons of Virtualization

- **Data can be at Risk –**
  Working on virtual instances on shared resources means that our data is hosted on third party resource which put's our data in vulnerable condition. Any hacker can attack on our data or try to perform unauthorized access. Without Security solution our data is in threaten situation**.**
- **Learning New Infrastructure –**
  As Organization shifted from Servers to Cloud. They required skilled staff who can work with cloud easily. Either they hire new IT staff with relevant skill or provide training on that skill which increase the cost of company.
- **High Initial Investment –**
  It is true that Virtualization will reduce the cost of companies but also it is truth that Cloud have high initial investment. It provides numerous services which are not required and when unskilled organization will try to set up in cloud they purchase unnecessary services which are not even required to them.

Hypervisor

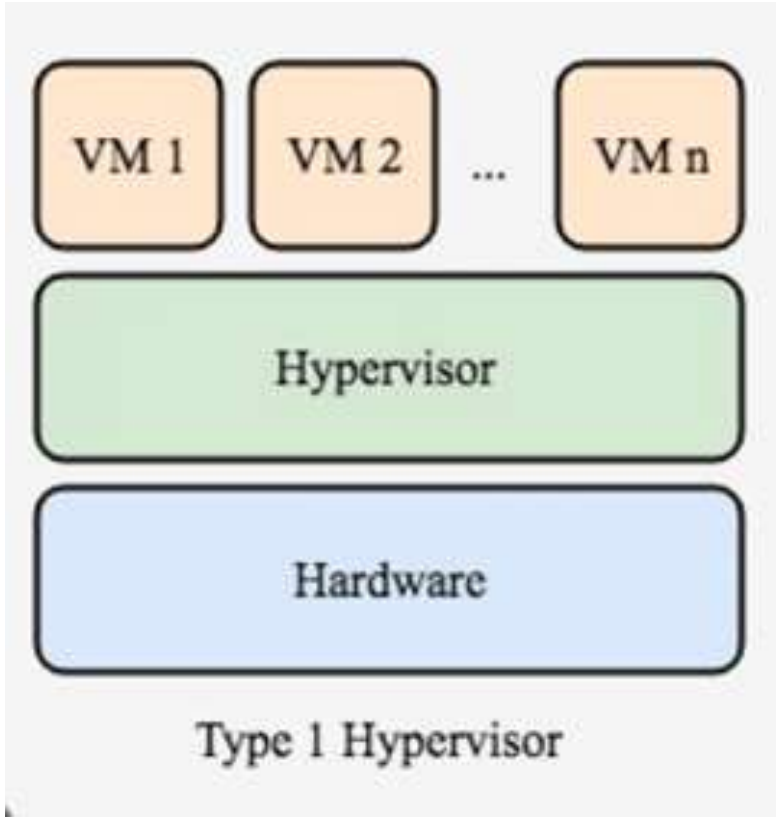
- Understanding the importance of Hypervisors, Type I & Type II Hypervisors.

Hypervisor

- A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware.
- The program which provides partitioning, isolation, or abstraction is called a virtualization hypervisor.
- The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time.
- A hypervisor is sometimes also called a virtual machine manager(VMM).

Types of Hypervisor – TYPE-1 Hypervisor

- The hypervisor runs directly on the underlying host system. It is also known as a “Native Hypervisor” or “Bare metal hypervisor”. It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor.

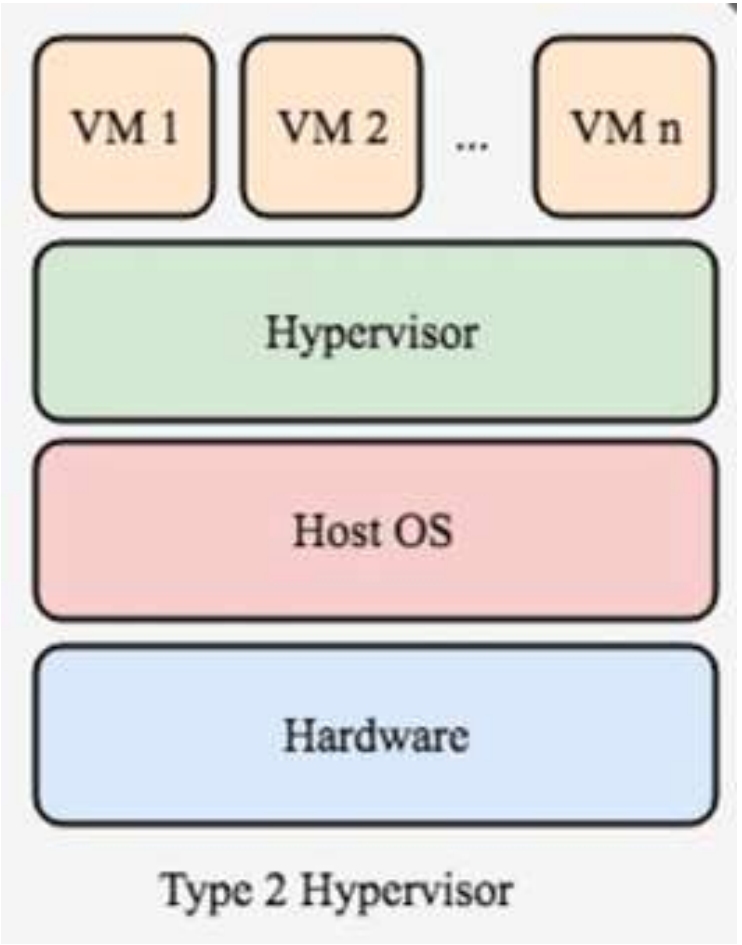


Types of Hypervisor – TYPE-1 Hypervisor

- **Pros & Cons of Type-1 Hypervisor:**
- **Pros:** Such kinds of hypervisors are very efficient because they have direct access to the physical hardware resources(like Cpu, Memory, Network, and Physical storage). This causes the empowerment of the security because there is nothing any kind of the third party resource so that attacker couldn’t compromise with anything.
- **Cons:** One problem with Type-1 hypervisors is that they usually need a dedicated separate machine to perform their operation and to instruct different VMs and control the host hardware resources.

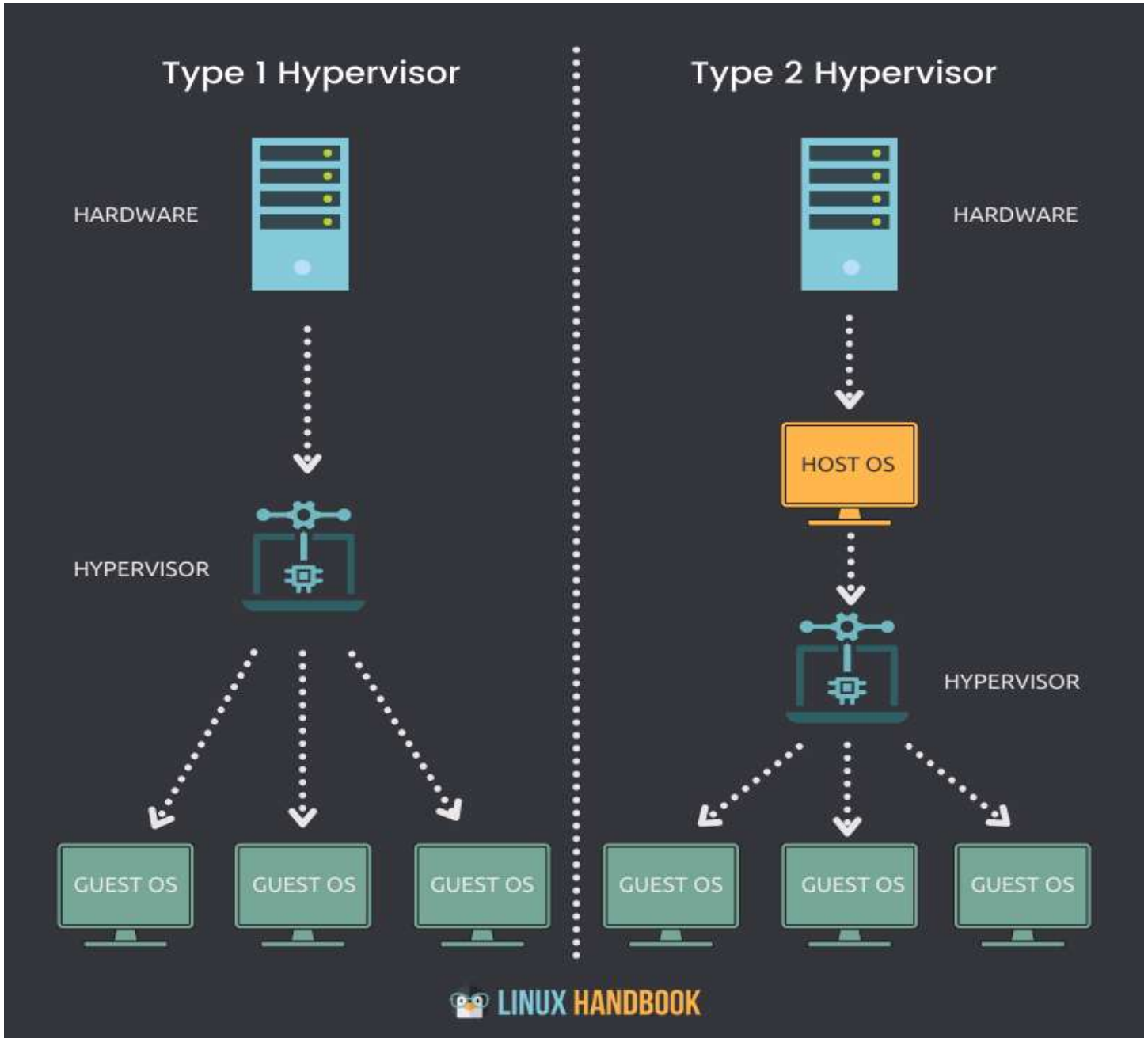
Types of Hypervisor – TYPE-2 Hypervisor

- A Host operating system runs on the underlying host system. It is also known as ‘Hosted Hypervisor”. Such kind of hypervisors doesn’t run directly over the underlying hardware rather they run as an application in a Host system(physical machine). Basically, the software is installed on an operating system. Hypervisor asks the operating system to make hardware calls. Example VMware Player or Parallels Desktop.
- Hosted hypervisors are often found on endpoints like PCs. The type-2 hypervisor is very useful for engineers, and security analysts (for checking malware, or malicious source code and newly developed applications).



Types of Hypervisor – TYPE-2 Hypervisor

- **Pros & Cons of Type-2 Hypervisor:**
- **Pros:** Such kind of hypervisors allows quick and easy access to a guest Operating System alongside the host machine running. These hypervisors usually come with additional useful features for guest machines. Such tools enhance the coordination between the host machine and the guest machine.
- **Cons:** Here there is no direct access to the physical hardware resources so the efficiency of these hypervisors lags in performance as compared to the type-1 hypervisors, and potential security risks are also there an attacker can compromise the security weakness if there is access to the host operating system so he can also access the guest operating system.


Type 1 Hypervisor
HARDWARE
HYPERVISOR
GUEST OS
GUEST OS
GUEST OS
Type 2 Hypervisor
HARDWARE
HOST OS
HYPERVISOR
GUEST OS
GUEST OS
GUEST OS
LINUX HANDBOOK

Choosing the right hypervisor

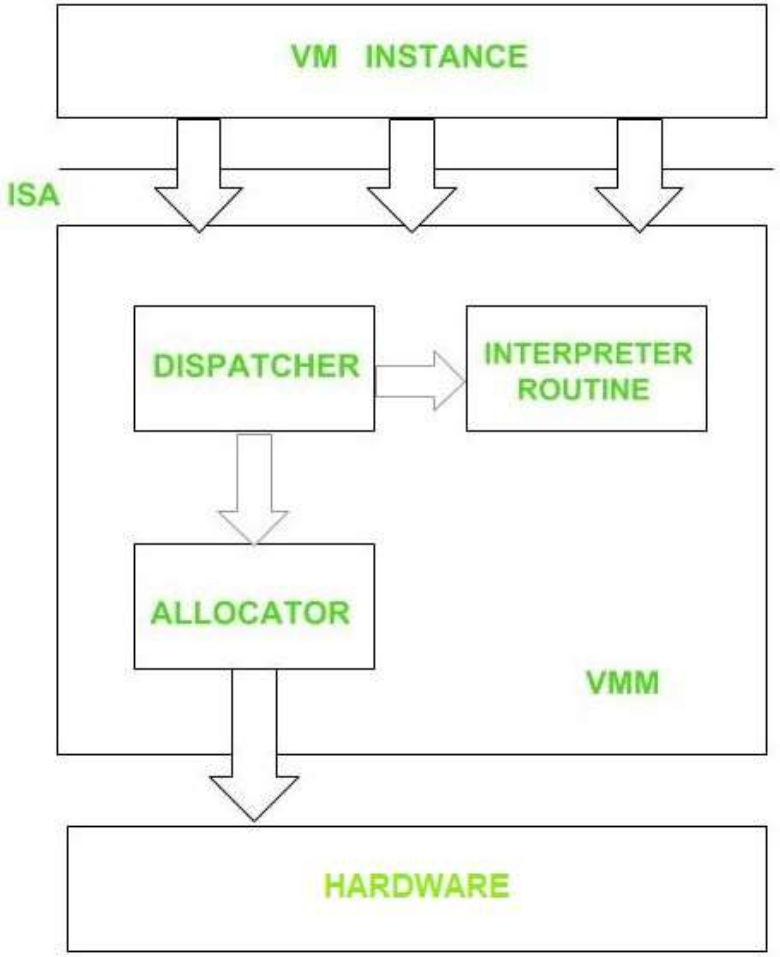
•**Type 1 hypervisors offer much better performance than Type 2** ones because there's no middle layer, making them the logical choice for mission-critical applications and workloads. But that's not to say that hosted hypervisors don't have their place – they're much simpler to set up, so they're a good bet if, say, you need to deploy a test environment quickly. One of the best ways to determine which hypervisor meets your needs is to compare their performance metrics. These include CPU overhead, the amount of maximum host and guest memory, and support for virtual processors.

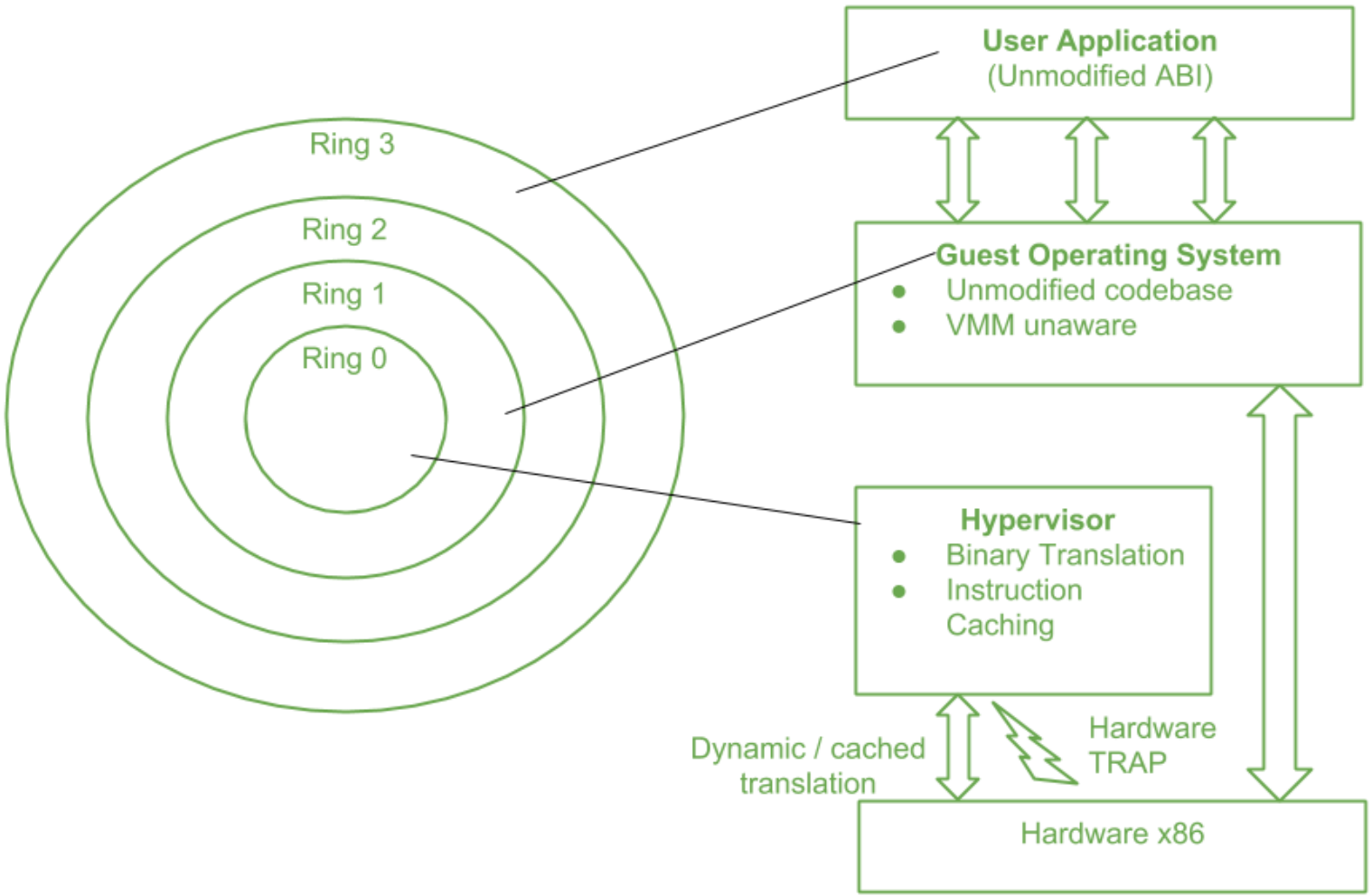
Choosing the right hypervisor

- **Understand your needs**
- **The cost of a hypervisor:** the right balance between cost and functionality. While a number of entry-level solutions are free, or practically free, the prices at the opposite end of the market can be staggering.
- **Virtual machine performance:** Virtual systems should meet or exceed the performance of their physical counterparts, at least in relation to the applications within each server.
- **Ecosystem:** It’s tempting to overlook the role of a hypervisor’s ecosystem – that is, the availability of documentation, support, training, third-party developers and consultancies, and so on – in determining whether or not a solution is cost-effective in the long term.
- **Test for yourself:** gain basic experience from existing desktop or laptop. Run both VMware vSphere and Microsoft Hyper-V in either VMware Workstation or VMware Fusion to create a nice virtual learning and testing environment.

HYPERVISOR REFERENCE MODEL

- **DISPATCHER:**
  The dispatcher behaves like the entry point of the monitor and reroutes the instructions of the virtual machine instance to one of the other two modules.
- **ALLOCATOR:**
  The allocator is responsible for deciding the system resources to be provided to the virtual machine instance. It means whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with the virtual machine, the allocator is invoked by the dispatcher.
- **INTERPRETER:**
  The interpreter module consists of interpreter routines. These are executed, whenever a virtual machine executes a privileged instruction.




Ring 3
Ring 2
Ring 1
Ring 0
User Application
(Unmodified ABI)
Guest Operating System
Unmodified codebase
VMM unaware
Hypervisor
Binary Translation
Instruction Caching
Dynamic / cached translation
Hardware TRAP
Hardware x86

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

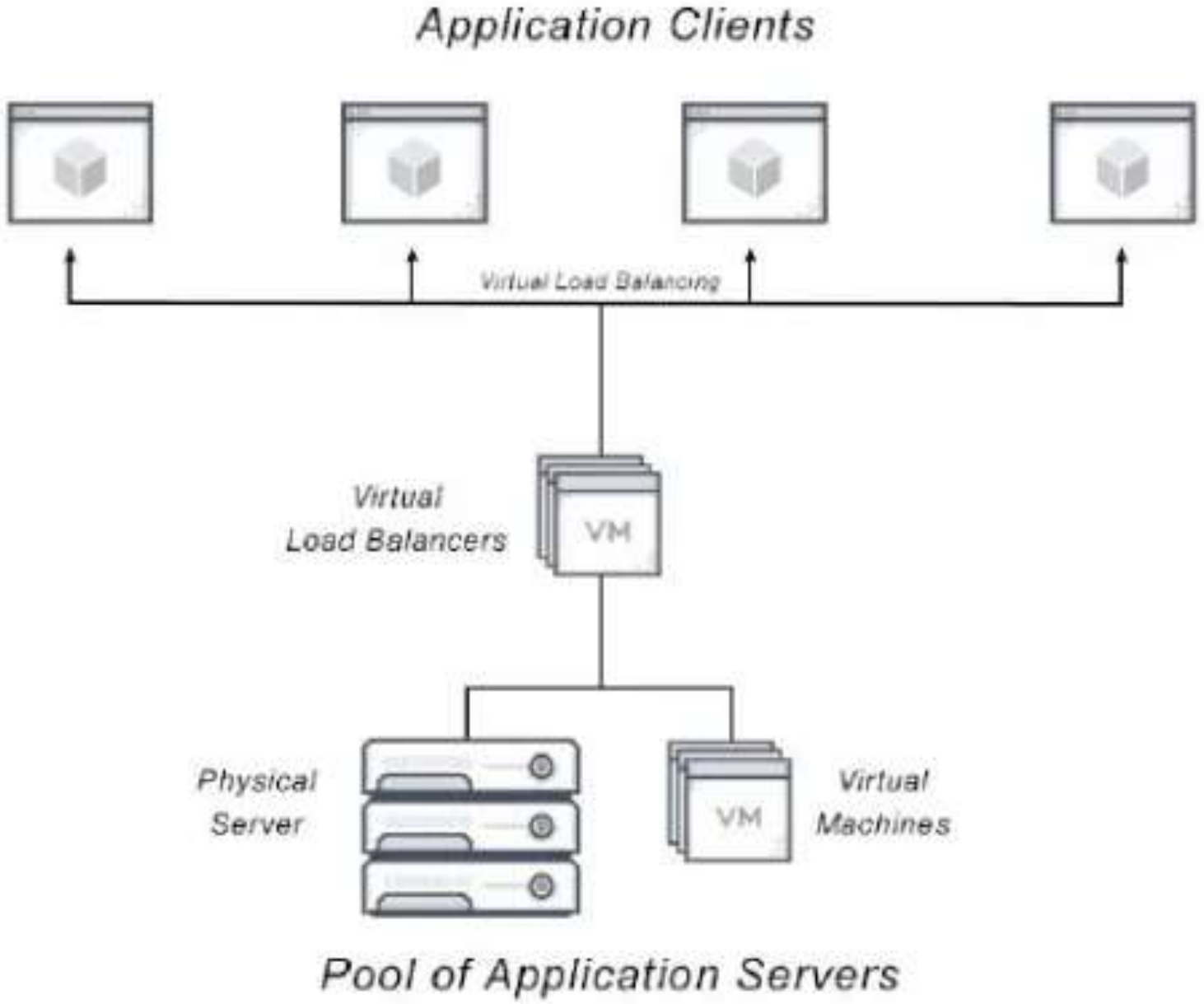
Load Balancing and Virtualization

What is load balancing and virtualization in cloud computing?

A Virtual Load Balancer provides more flexibility to balance the workload of a server by distributing traffic across multiple network servers. Virtual load balancing aims to mimic software-driven infrastructure through virtualization. It runs the software of a physical load balancing appliance on a virtual machine. A virtual network load balancer promises to deliver software load balancing by taking the software of a physical appliance and running it on a virtual machine load balancer.

Virtual load balancers, however, are a short-term solution. The architectural challenges of traditional hardware appliances remain, such as limited scalability and automation, and lack of central management (including the separation of control plane and data plane) in data centers.

Load Balancing and Virtualization



How virtual load balancer work?

The traditional application delivery controller companies build virtual load balancers that utilize code from legacy hardware load balancers. The code simply runs on a virtual machine. But these virtual load balancers are still monolithic load balancers with static capacity.

Virtual vs. Hardware load balancers

- The complexity and limitations of a virtual load balancer is similar to that of a hardware load balancer.
- A hardware load balancer uses rack-mounted, on-premises physical hardware. Hardware load balancers are proven to handle high traffic volume well. But the hardware can be expensive and limit flexibility.
- A virtual load balancer uses the same code from a physical appliance. It also tightly couples the data and control plane in the same virtual machine. This leads to the same inflexibility as the hardware load balancer.
- For example, while an F5 virtual load balancer lowers the CapEx compared to hardware load balancers, virtual appliances are in reality hardware-defined software.

Load Balancing and Virtualization

Types:

o Static

o Dynamic

Need of load balancing:

- Improving the performance Maintaining the system stability
- Quality of service (QoS)
- Building fault tolerance

Load balancing algorithms

Scheduling algorithms

- FCFS
- Round Robin

Soft computing based algorithms

- Stochastic algorithm
- Genetic algorithm
- Ant colony optimization

Static load balancing algorithms

Static load balancing algorithms in distributed systems minimize specific performance functions by associating a known set of tasks with available processors. These types of load balancing strategies typically center around a router that optimizes the performance function and distributes loads.

Static vs. dynamic load balancing

- A load balancer is a device that acts as a reverse proxy and distributes network or application traffic across a number of servers.
- Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name Service server process.
- Load Balancing can be classified into two types based on the behavior of the algorithm:
  - Static Load Balancing: Static load balancing is the method of dividing the incoming load on a server using algorithms that have prior information about the existing servers in the distributed network. These load balancing schemes have a pre-defined load schedule that determines a fixed amount of load that can be shed on other systems.
  - Dynamic Load Balancing: It is a more versatile scheme of load balancing which can dynamically identify the amount of load that needs to be shared during runtime and which system should bear the load.

Sr. No.	Static Load Balancing	Dynamic Load Balancing
1.	Designed for the system with low fluctuation in incoming load.	Designed for the system with high fluctuation in incoming load.
2.	Traffic is equally divided among the servers.	Traffic is dynamically divided among the servers.
3.	It requires deeper information about available system resources.	It does not necessarily need deeper information about system resources beforehand.
4.	It does not require real-time communication with the servers.	It requires real-time communication actively with the servers.
5.	The allocated load cannot be retransferred to other servers during runtime.	The allocated load can be retransferred among servers to reduce the under utilization of resources.
6.	Example: Round Robin algorithm for load	Example: Least Connection algorithm for load

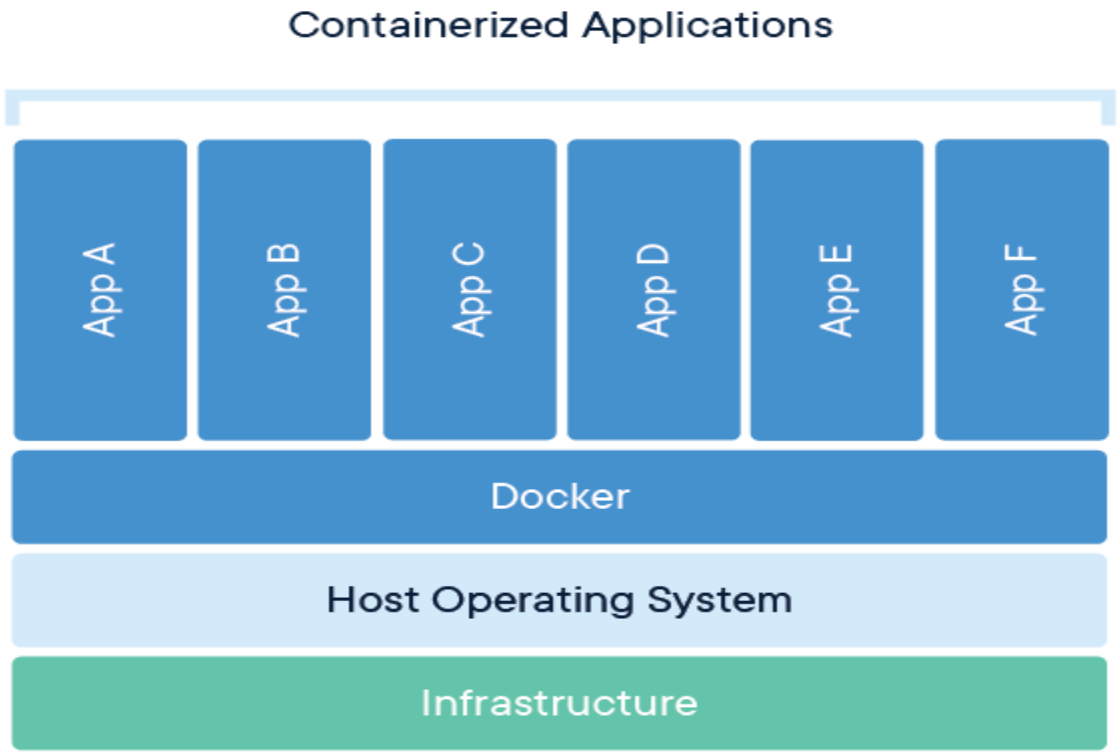
Docker Containerization

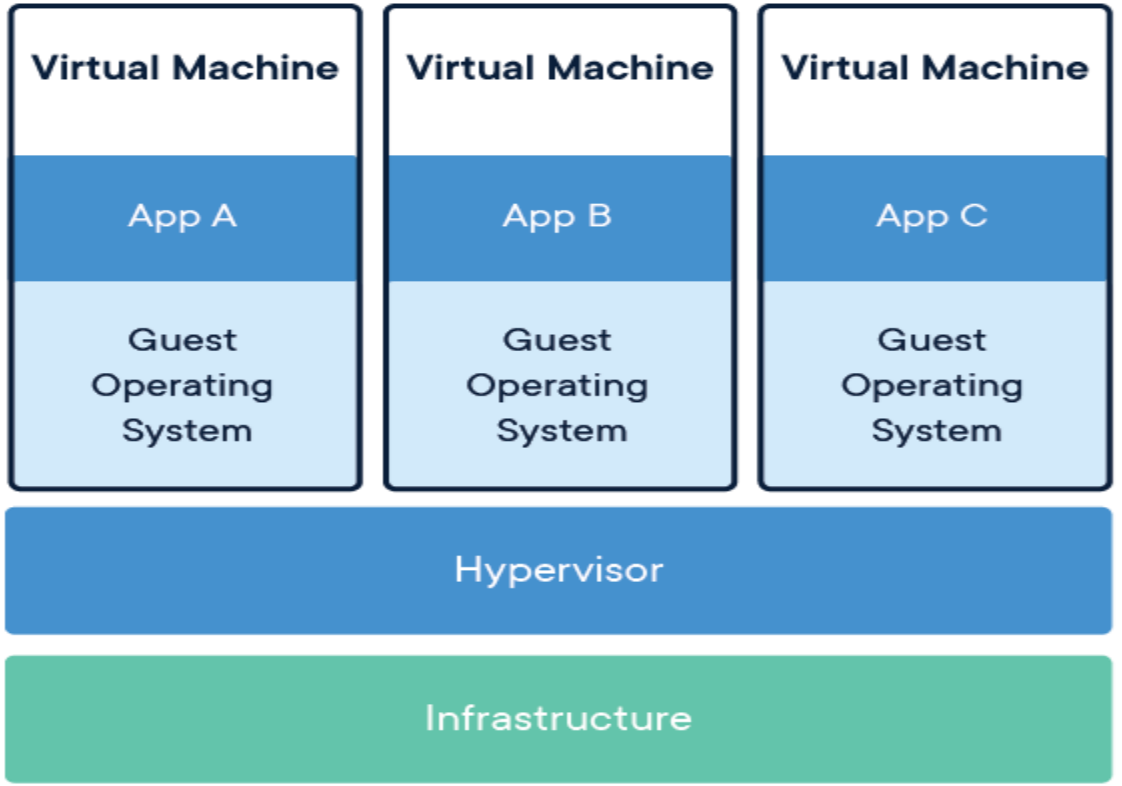
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers – images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker functions based on a client-server architecture. The core components and their interactions are as follows:

- Docker Daemon (dockerd): This is the server component that runs on the host machine. It performs the heavy lifting, including building, running, and distributing Docker containers. It listens for API requests from the Docker client.
- Docker Client (docker): This is the command-line interface (CLI) or a graphical user interface (GUI) that users interact with. It sends commands to the Docker Daemon via a REST API, which can communicate over UNIX sockets or a network interface.
- Docker Images: These are read-only templates containing the application, its dependencies, and configuration. Images are built from a Dockerfile, which specifies the instructions for creating the image.
- Docker Containers: These are runnable instances of Docker images. When a container is launched, it provides an isolated environment with its own filesystem, network interfaces, and process space, all sharing the host machine's operating system kernel.





CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

VIRTUAL MACHINES

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries – taking up tens of GBs. VMs can also be slow to boot.

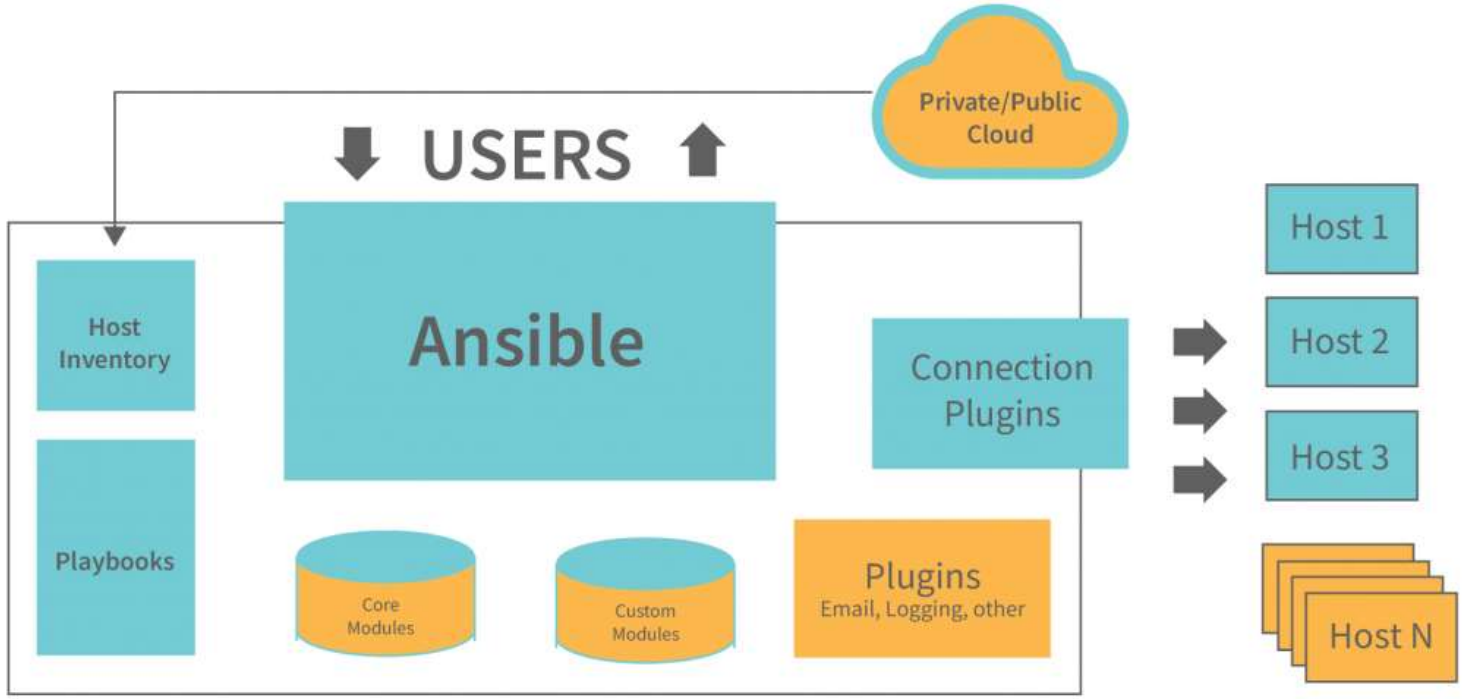
How docker works:

- Dockerfile: You define your application's environment and dependencies in a Dockerfile.
- Docker Build: The Docker client sends the Dockerfile and context to the Docker Daemon, which then builds a Docker image based on the instructions.
- Docker Run: The Docker client instructs the Docker Daemon to run a container from a specific image.
- Container Execution: The Docker Daemon creates an isolated container instance, mounts the necessary resources, and starts the application within it.

This process ensures that applications run consistently across different environments, from development to production, as the container encapsulates all required components.

Feature	Virtual Machines (VMs)	Docker Containers
Virtualization	Full OS Virtualization	OS-Level Virtualization
Isolation	Strong (OS-level)	Process-level
Resources	Resource-intensive	Lightweight, efficient
Startup Time	Slower (minutes)	Faster (seconds/ms)
OS Kernel	Each VM has its own	Share host OS kernel
Portability	Good	Excellent

Ansible Architecture



Ansible is an open-source configuration management tool owned by RedHat. Ansible can configure any resource on a server through its idempotent playbooks and even run ad-hoc scripts. Ansible takes complex or cumbersome manual tasks and orchestrates them by automating the process. Tasks done wrong are typically repetitive in nature.

The Ansible framework operates on a control-node-to-managed-node architecture, designed for agentless IT automation.

1. Control Node: This is the machine where Ansible is installed and from which automation tasks are initiated. Users define desired states and automation workflows in YAML-based Playbooks on the control node.

2. Managed Nodes (Inventory): These are the remote systems (servers, network devices, cloud instances) that Ansible manages. Managed nodes are listed and organized within an "Inventory" file on the control node, which serves as a list of targets for automation.

3. Connection and Module Execution: Ansible connects to managed nodes, typically using SSH for Linux/Unix systems and WinRM for Windows. It then pushes small, temporary programs called "modules" to the managed nodes. These modules execute specific tasks (e.g., installing software, configuring services, managing users). After execution, the modules are removed from the managed node.

4. Playbooks: Playbooks are the core of Ansible automation, written in human-readable YAML syntax.

They define a series of "plays," where each play targets a specific group of hosts from the inventory and outlines a sequence of "tasks" to be performed using various modules.

5. Idempotency: Many Ansible modules are designed to be idempotent, meaning they will only make changes if the desired state has not already been achieved. This ensures that repeating a

Ansible works by:

Defining the target systems in an inventory.

Describing the desired configuration or actions in a playbook.

Connecting to the managed nodes and executing modules to achieve the specified tasks, without requiring agents on the managed nodes.

Feature	Docker	Ansible
Primary Goal	Application containerization and portability	Infrastructure automation and configuration management
Unit of Work	Containers (isolated application environments)	Playbooks (tasks to configure infrastructure)
Approach	Packages applications with dependencies	Configures and manages existing infrastructure
Agent Requirement	No external agent (relies on Docker daemon)	Agentless (uses SSH for remote execution)
Output	Portable container images	Configured infrastructure state