SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641

107 **An Autonomous Institution**

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

COURSE CODE AND NAME:19TS501& CLOUD COMPUTING

UNIT 2: CLOUD ENABLING TECHNOLOGY

TOPIC : SERVICE ORIERNTED ARCHITECTURE & REST

SERVICE ORIENTED ARCHITECTURE

Definition:

- SOA is an architectural approach where software components, called services, are designed to interact and provide functionality as reusable, loosely coupled, and interoperable modules.
- In this architecture, services are provided to form applications, through a network call over the internet.
- These can be created or can use available resources.

Example: Uber Application

It needs 3 services

1.Login page(google,fb)

2.Map(google map)

3.Payment Gateway(gpay,paytm)

Benefit:

1.Reuse

2.Scalable

SOA'S TWO MAIN ROLES:

- Service provider
- Service consumer

SERVICE CONNECTIONS

- Service consumer sends a service request to the service provider, and the service provider sends the service response to the service consumer.
- The service connection is understandable to both the service consumer and service provider.


service request
service response
Message
Consumer
Message
Provider
JavaTpoint

Principles:

- **Loose Coupling:**

  Services are independent and interact through well-defined interfaces, reducing dependencies and allowing for flexibility.

- **Reusability:**

  Services can be reused across different applications, promoting efficient development.

- **Standardized service contract:**

  Specified through one or more service description documents..

Abstraction:

- A service is completely defined by service contracts and description documents.
- They hide their logic, which is encapsulated within their implementation.

Discoverability:

- Services are defined by description documents that constitute supplemental metadata through which they can be effectively discovered.
- Service discovery provides an effective means for utilizing third-party resources

Advantages:

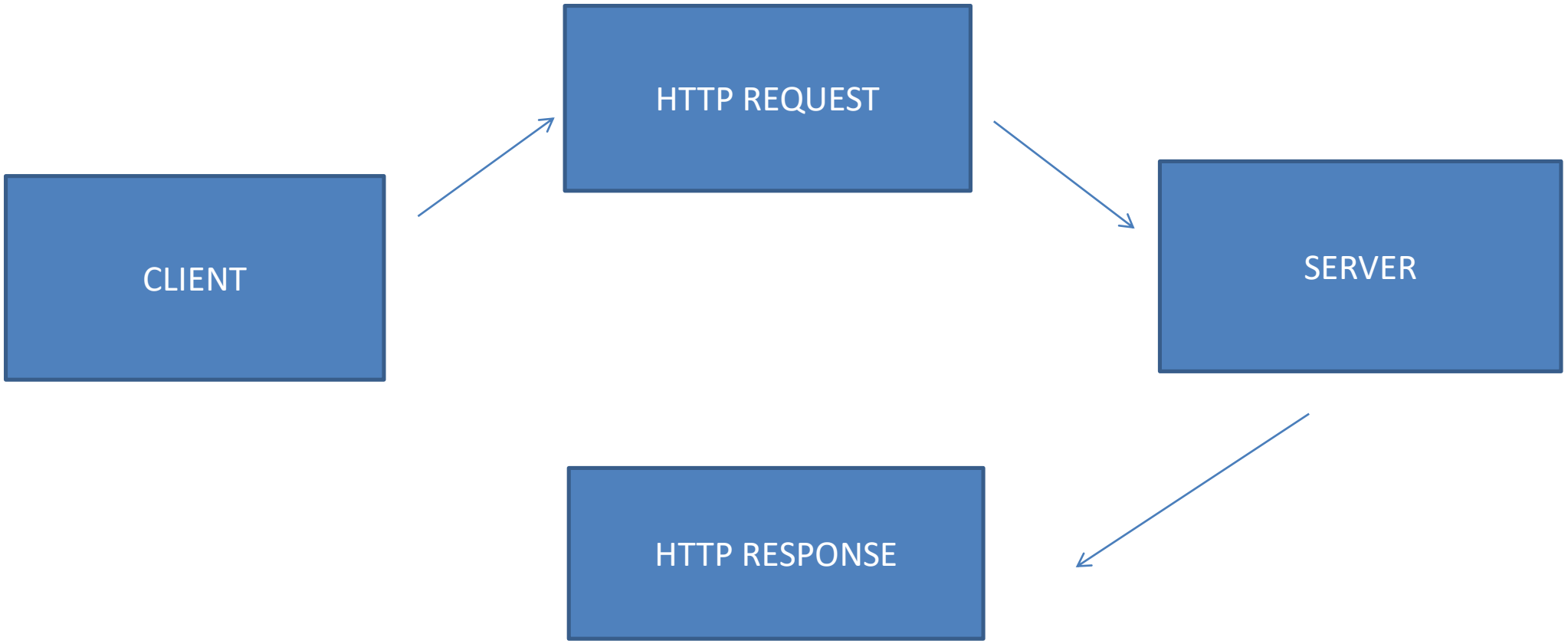
- **Flexibility:** Supports incremental changes and updates.
- **Reusability:** Promotes efficient development through service reuse.
- **Interoperability:** Services from various sources can collaborate seamlessly.
- **Scalability:** Can scale horizontally by adding more instances of services

Main Components of SOA:

1. **Services:** Self-contained, modular units that expose well-defined interfaces and perform specific functions.
2. **Service Consumer:** Applications or components that use and interact with services.
3. **Service Provider:** The entity responsible for creating and maintaining the service.
4. **Service Registry:** A directory where services can be registered and discovered.
5. **Service Contract:** A formal definition of the service's capabilities, inputs, outputs, and behavior.

REST

- **REpresentational State Transfer** (REST) is a software architectural style that defines the constraints to create web services.
- The web services that follows the **REST** architectural style is called **RESTful Web Services**.
- This is architectural style designed for loosely coupled application over **HTTP**
- **Eg: Facebook, Twitter**


HTTP REQUEST
CLIENT
SERVER
HTTP RESPONSE

Rest consist of

1.Style(no specialization)

2.HTTP

3.Status codes

4.Responses

HTTP Methods:

1.GET (get something)

2.POST (create a resource)

3.DELETE(Delete a resource)

Status codes:

- 1** - Informational Response
- 2** -Successful Response
- 3** -Redirects
- 4** -Client Errors
- 5** -Server Errors

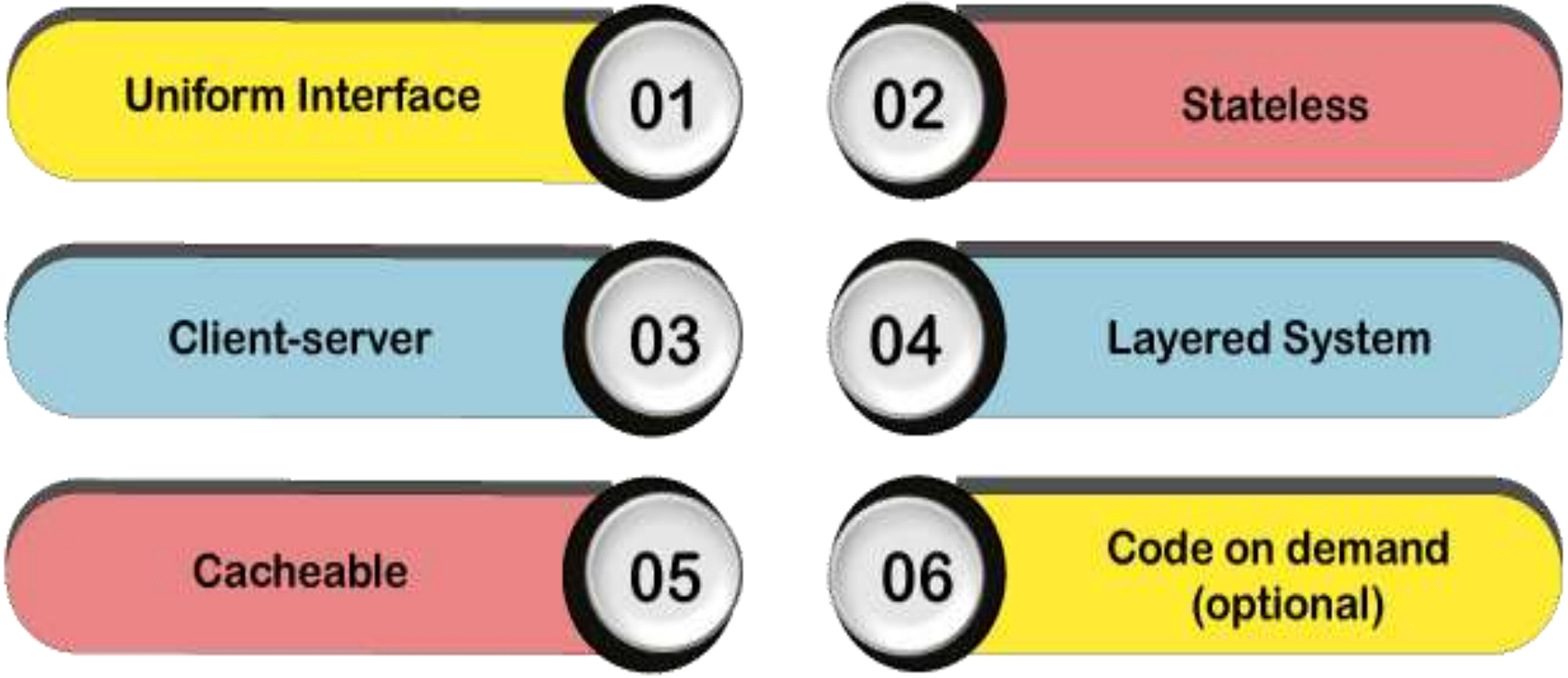
Eg: Servers will host important documents or video.

If client need any resource, it has sent request to servers.

Now rest comes,

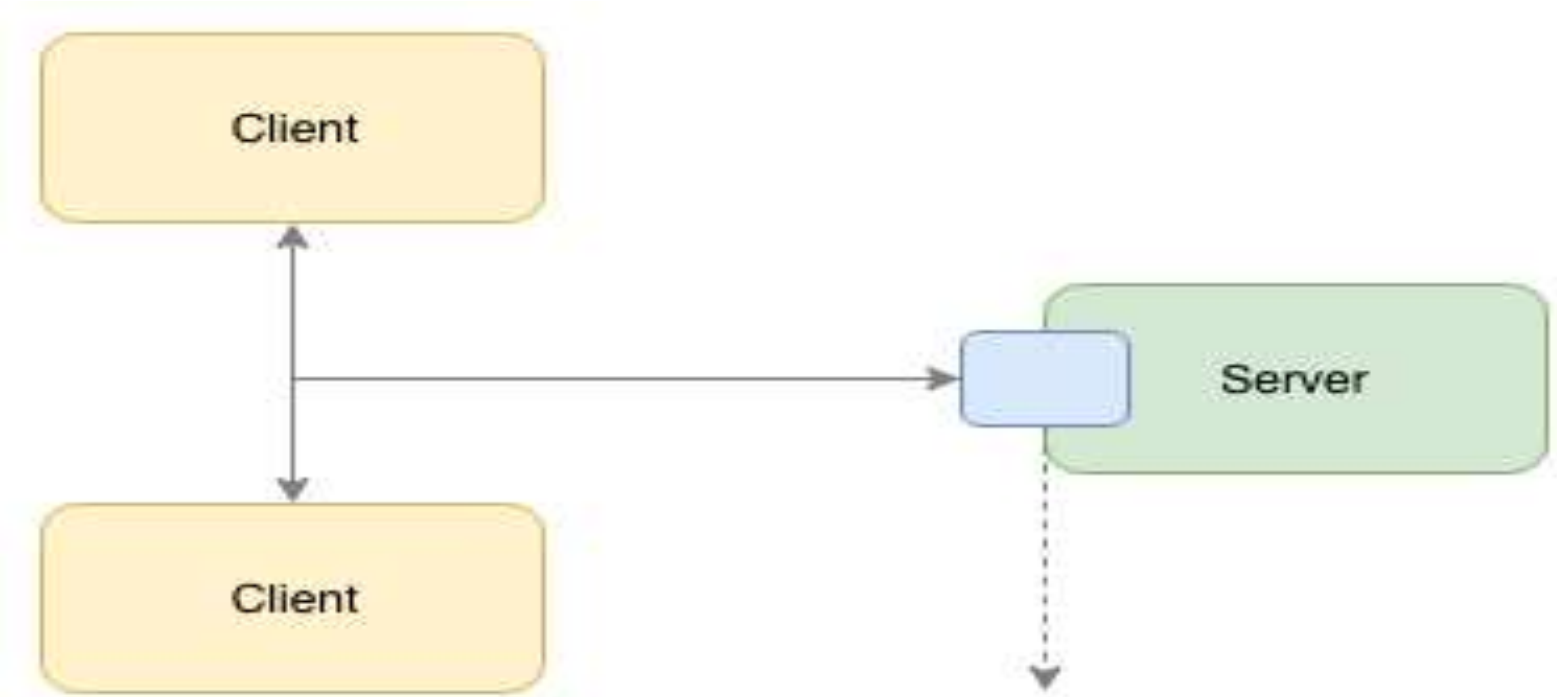
1. **Resources**(ask for resources of employee) http://-----.com)/employee/1
2. **Request verbs** (HTTP Protocols)-GET,PUT
3. **Request headers**(Additional instruction sent with request)
4. **Request body**(To post request in web service)
5. **Response body**(provide detail in xml document)
6. **Response status code**(codes will be generated like code200)

CONSTRAINTS OF REST ARCHITECTURE



Uniform Interface:

- The uniform interface constraint in REST promotes a consistent and standardized way of interacting with resources.
- Through HTTP methods like GET, POST, PUT, and DELETE.
- While this constraint enhances simplicity and visibility, it might not be suitable for all complex cloud computing scenarios
- It require more fine-grained control or custom operations.

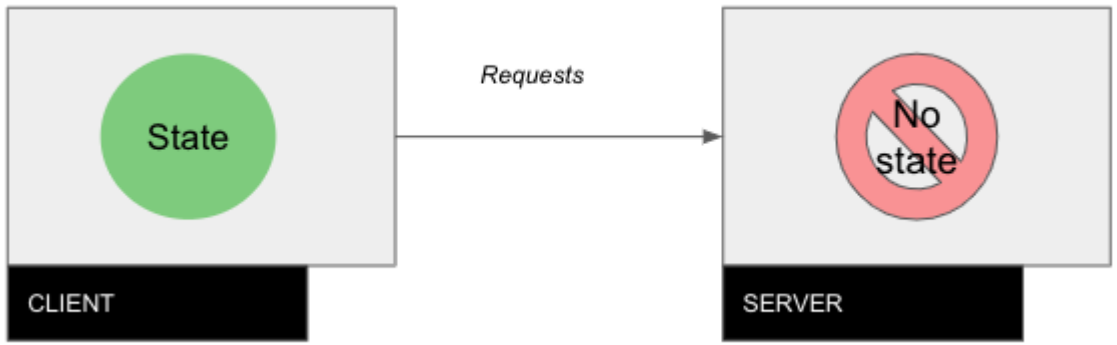


Interface

It uses the same interface for all requestors

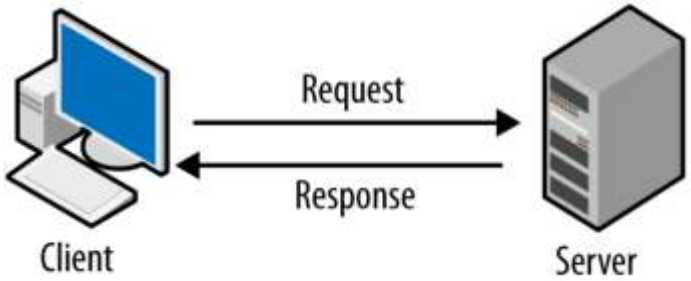
Statelessness:

- REST is designed to be stateless, meaning that each request from a client to the server must contain all the information necessary to understand and process the request.
- While statelessness simplifies server design and improves scalability, it might lead to challenges in cloud computing scenarios where maintaining session state across multiple requests or services is required.



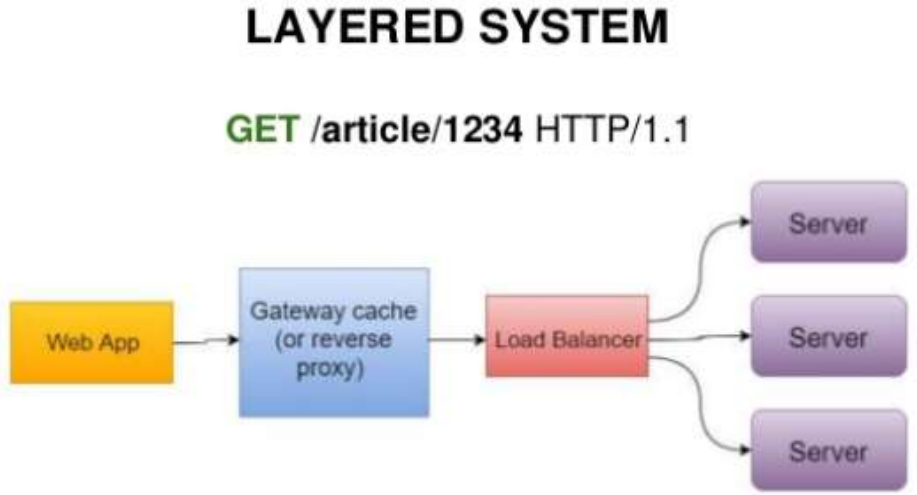
Client-Server Separation:

- REST emphasizes the separation of client and server concerns, which promotes scalability and independent evolution of components.
- However, in certain cloud computing scenarios, this separation might hinder performance optimization or efficient resource utilization, as the server may lack the knowledge of specific client requirements.



Layered System:

- REST allows for the use of intermediary components (e.g., load balancers, caching servers) between clients and servers, enhancing scalability and encapsulation.
- However, the introduction of additional layers can introduce latency and complexity, potentially affecting the responsiveness of cloud applications.



Code on Demand (Optional):

- This constraint allows the server to send executable code (e.g., JavaScript) to clients, enabling dynamic behavior.
- In cloud computing, security concerns might arise from executing code from external sources, potentially exposing systems to vulnerabilities or malicious attacks.

Cacheable:

- On the World Wide Web, customers can cache responses.
- Therefore, responses clearly define themselves as unacceptable or prevent customers from **reusing** stale or **inappropriate data** to further requests.
- **Well-managed** caching eliminates some client-server interactions to improving scalability and performance.

SYSTEMS AND SYSTEMS

- A System of Systems (SoS) in the context of cloud computing refers to a complex, interconnected collection of individual software systems, services, and components that collaborate and interact to achieve a higher-level goal or provide enhanced functionality.
- In other words, a System of Systems is a composition of independent systems working together as a cohesive whole, often leveraging cloud computing resources and principles to achieve scalability, flexibility, and efficiency.

CHARACTERISTICS

Interconnectedness:

- A System of Systems involves multiple independent software systems or services that communicate and interact with each other to provide more extensive capabilities or address larger and more complex problems.

Independence:

- Each individual system within the SoS retains its autonomy and is designed to function independently.
- This independence allows for flexibility and the ability to replace or update individual components without affecting the entire system.

Cloud Computing Integration:

- Cloud computing technology plays a crucial role in building and operating Systems of Systems.
- Cloud resources such as computing power, storage, and networking can be utilized to support the individual systems within the SoS, enabling efficient resource allocation and dynamic scalability.

Resource Sharing:

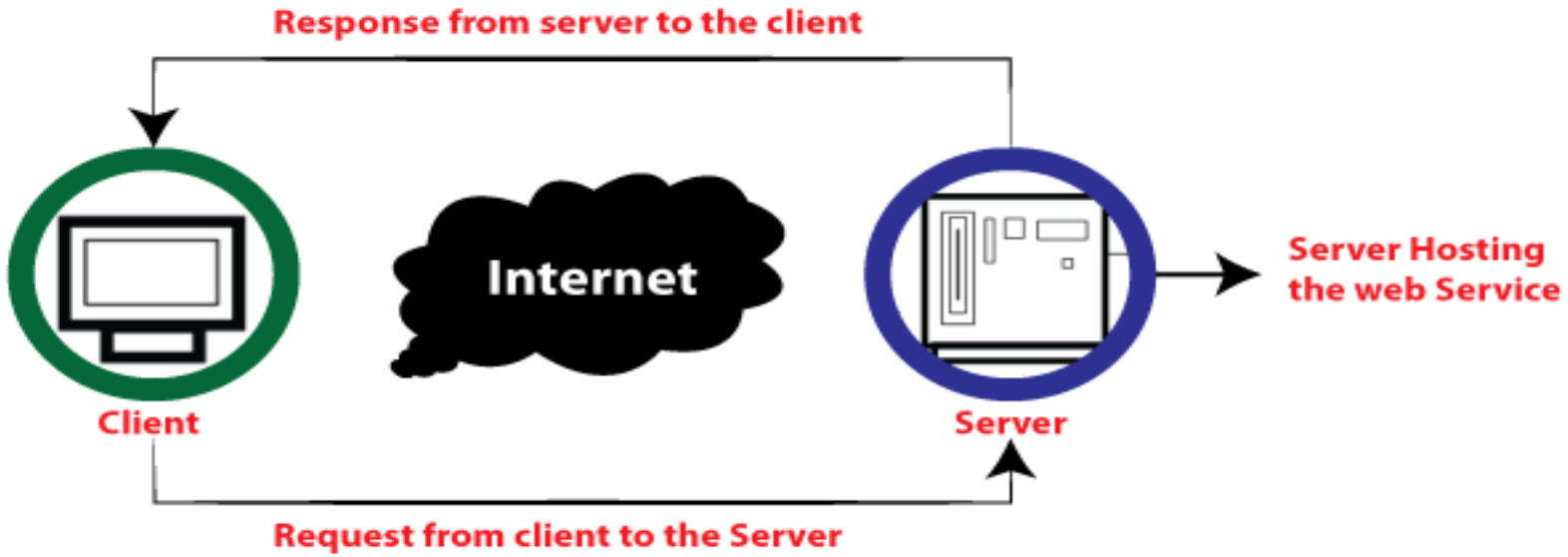
Cloud-based Systems of Systems can take advantage of resource sharing and efficient utilization of infrastructure, reducing costs and optimizing performance.

WEB SERVICES

- A web service is a standardized method for propagating messages between client and server applications on the World Wide Web.
- A web service is a software module that aims to accomplish a specific set of tasks.
- Web services can be found and implemented over a network in cloud computing.

How does web service work?

- The client will use requests to send a sequence of web service calls to the server hosting the actual web service.



Components of Web Services

The following are the critical components of web services:

1. SOAP

- The acronym SOAP stands for Simple Object Access Protocol.
- A convention based on XML for accessing web services.
- W3C recommendation for application-to-application communication.
- Independent of both platforms and languages

2. UDDI

- Universal Description, Discovery, and Integration is referred to as UDDI.
- System based on XML.
- Information on web administrations

3. WSDL

- The acronym for WSDL is Web Services Description Language.
- It is an XML file.
- It contains information about web administrations, such as the name of the technology, the strategy parameter, and directions to the site.
- It serves as a connection point for web administration software.

Features of Web Services

XML-Based

- Records are transported and represented using XML in web services.
- Using XML eliminates the requirement for bindings on platforms, operating systems, or networks.
- Web-based applications with a middle level of interactivity are widespread.

Loosely Coupled

- An Internet service provider and the subscriber may only sometimes be close to one another.
- A web service provider's user interface might evolve without impairing the user's communication ability.
- Because the server's and mentor's decisions are closely related to one another.

Ability to be Synchronous or Asynchronous

- Synchronization is the relationship between the client and the function's execution.
- When the service is finished, synchronous clients receive their results instantly, and asynchronous clients receive them later.

Coarse Grain

- Object-oriented systems like Java have several ways of making their services available, and corporate operations are too large for character techniques to be effective.
- A Java application must be built from the ground up using various granular strategies integrated to create a coarse grain provider used by the customer or service.

PUBLISH SUBSCRIBE MODEL

- Publishers produce information in form of events, which are then consumed by subscribers.
- Subscribers can declare their interest on a subset of the whole information issuing subscriptions.

There are two major roles:

- Publisher
- Subscriber

There are two major strategies for dispatching the event to the subscribers.

Push strategy:

It is the responsibility of the publisher to notify all the subscribers. Eg: Method invocation.

Pull strategy :

- The publisher simply makes available the message for a specific event.
- It is the responsibility of the subscribers to check whether there are messages on the events that are registered.
- Subscriptions are used to filter out part of the events produced by publishers.

There are three main components to the Publish Subscribe Model:

- Publishers
- Eventbus/broker
- Subscribers

Publishers:

Broadcast messages, with no knowledge of the subscribers.

Subscribers:

They 'listen' out for messages regarding topic/categories that they are interested in without any knowledge of who the publishers are.

Event Bus:

Transfers the messages from the publishers to the subscribers.

Quality of Service in Publish/Subscribe Systems

- ❑Reliable delivery
- ❑Timeliness
- ❑Security and trust

Subscription Models

- Topic based Model
- Type based Model
- Concept based Model
- Content based Model

Topic based Model

- ❑ Events are grouped in topics.
- ❑ A subscriber declares its interest for a particular topic to receive all events pertaining to that topic.

Type based Model :

- Pub/sub variant events are actually objects belonging to a specific type, which can thus encapsulate attributes as well as methods.
- Types represent a more robust data model for application developer.

Concept based Model :

- Allows to describe event schema at a higher level of abstraction by using ontologies.

Content based Model

- System allows subscribers to receive messages based on the content of the messages.
- Subscribers themselves must sort out junk messages from the ones they want

Applications:

Used in a wide range of group communication applications including

- Software Distribution
- Internet TV
- Audio or Video-conferencing
- Virtual Classroom
- Multi-party Network Games
- Distributed Cache Update

Benefits:

- Loose coupling
- Improved security
- Improved testability

Drawbacks :

- Increased complexity.
- Increased maintenance effort
- Decreased performance

BASICS OF VIRTUALIZATION

- Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines.
- Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.
- 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.

- Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization.
- A Virtual machine provides an environment that is logically separated from the underlying hardware.
- The machine on which the virtual machine is going to create is known as **Host Machine** and that virtual machine is referred as a **Guest Machine**

Hardware virtualization

- Hardware virtualization is accomplished by abstracting the physical hardware layer by use of a hypervisor or VMM (Virtual Machine Monitor).
- When the virtual machine software or virtual machine manager (VMM) or hypervisor software is directly installed on the hardware system is known as hardware virtualization.

Software virtualization :

software virtualization is just like a virtualization but able to abstract the software installation procedure and create virtual software installations.

Types of Virtualization:

- Application virtualization
- Network virtualization
- Desktop virtualization
- Server Virtualization.
- Storage Virtualization.
- Data Virtualization.

Application Virtualization

- This can be defined as the type of Virtualization that enables the end-user of an application to have remote access.
- This is achieved through a server. This server has all personal information and other applicable characteristics required to use the application.
- The server is accessible through the internet, and it runs on a local workstation. With Application virtualization, an end-user can run two different versions of the same software or the same application.
- Application virtualization is offered through packaged software or a hosted application.

Network Virtualization

- This kind of virtualization can execute many virtual networks, and each has a separate control and data plan.
- It co-occurs on the top of a physical network, and it can be run by parties who are not aware of one another.
- Network virtualization creates virtual networks, and it also maintains a provision of virtual networks.
- Through network virtualization, logical switches, firewalls, routers, load balancers, and workload security management systems can be created.

Desktop Virtualization

- This can be defined as the type of Virtualization that enables the operating system of end-users to be remotely stored on a server or data center.
- It enables the users to access their desktops remotely and do so by sitting in any geographical location.
- They can also use different machines to virtually access their desktops.
- With desktop virtualization, an end-user can work on more than one operating systems basis the business need of that individual.

Server Virtualization:

- When the virtual machine software or virtual machine manager (VMM) is directly installed on the Server system is known as server virtualization.

Usage:

- Server virtualization is done because a single physical server can be divided into multiple servers on the demand basis and for balancing the load.

Storage Virtualization:

- Storage virtualization is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device.
- Storage virtualization is also implemented by using software applications.

Usage:

- Storage virtualization is mainly done for back-up and recovery purposes.

Data Virtualization

- This can be defined as the type of Virtualization wherein data are sourced and collected from several sources and managed from a single location.
- There is no technical knowledge from where such data is sourced and collected, stored, or formatted for such data.
- The data is arranged logically, and the interested parties and stakeholders then access the virtual view of such data.
- These are reports are also accessed by end-users on a remote basis.

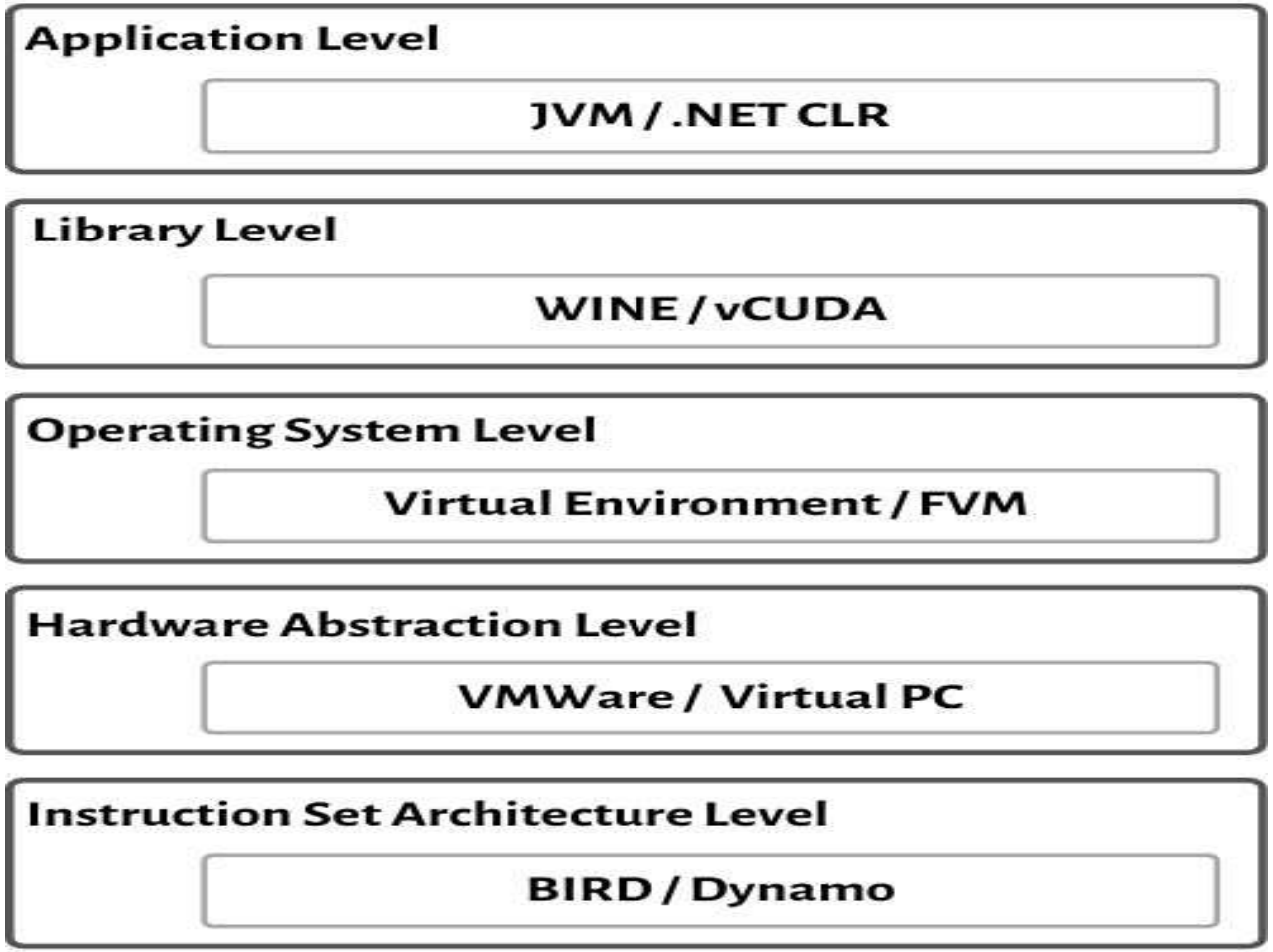
Usage of virtualization

- The **main usage of Virtualization Technology** is to provide the applications with the standard versions to their cloud users, suppose if the next version of that application is released, then cloud provider has to provide the latest version to their cloud users and practically it is possible because it is more expensive.
- To overcome this problem we use basically virtualization technology, By using virtualization, all severs and the software application which are required by other cloud providers are maintained by the third party people, and the cloud providers has to pay the money on monthly or annual basis.

Implementation Levels of Virtualization In Cloud Computing

- It is not simple to set up virtualization computer runs on an operating system that gets configured on some particular hardware. It is not feasible or easy to run a different operating system using the same hardware.
- To do this, we need a hypervisor. It is a bridge between the hardware and the virtual operating system, which allows smooth functioning.
- Talking of the Implementation levels of virtualization in Cloud Computing., there are a total of five levels that are commonly used.

Five Levels of Virtualization



1) Instruction Set Architecture Level (ISA)

- It is located at the microprocessor and microcontroller levels.
- It includes the basic sets of instructions.
- In the physical system, we write some code logic and emulate it as software.
- The host machine has a CPU and other devices. To communicate with the machine, instructions are required.
- The native-level instruction is converted to a higher level.
- **Eg:bird**

.

2)Hardware Abstraction Level (HAL)

- To convert a lower level to a higher level certain processing power and computing cycles are required; it finds the similarities between host os and guest os suitable for x86 architecture.
- It finds similarities to reduce the computation cycles.
- This will be successful only if we capture PI (privileged Instructions) and send it to VMM.
- **Eg:vmware**

3) Operating System Level

- It keeps track of the libraries and DS used to avoid redundancy.
- It also maintains the prerequisite of the host machine.
- **Eg:Jail**

- **4)Library Level**

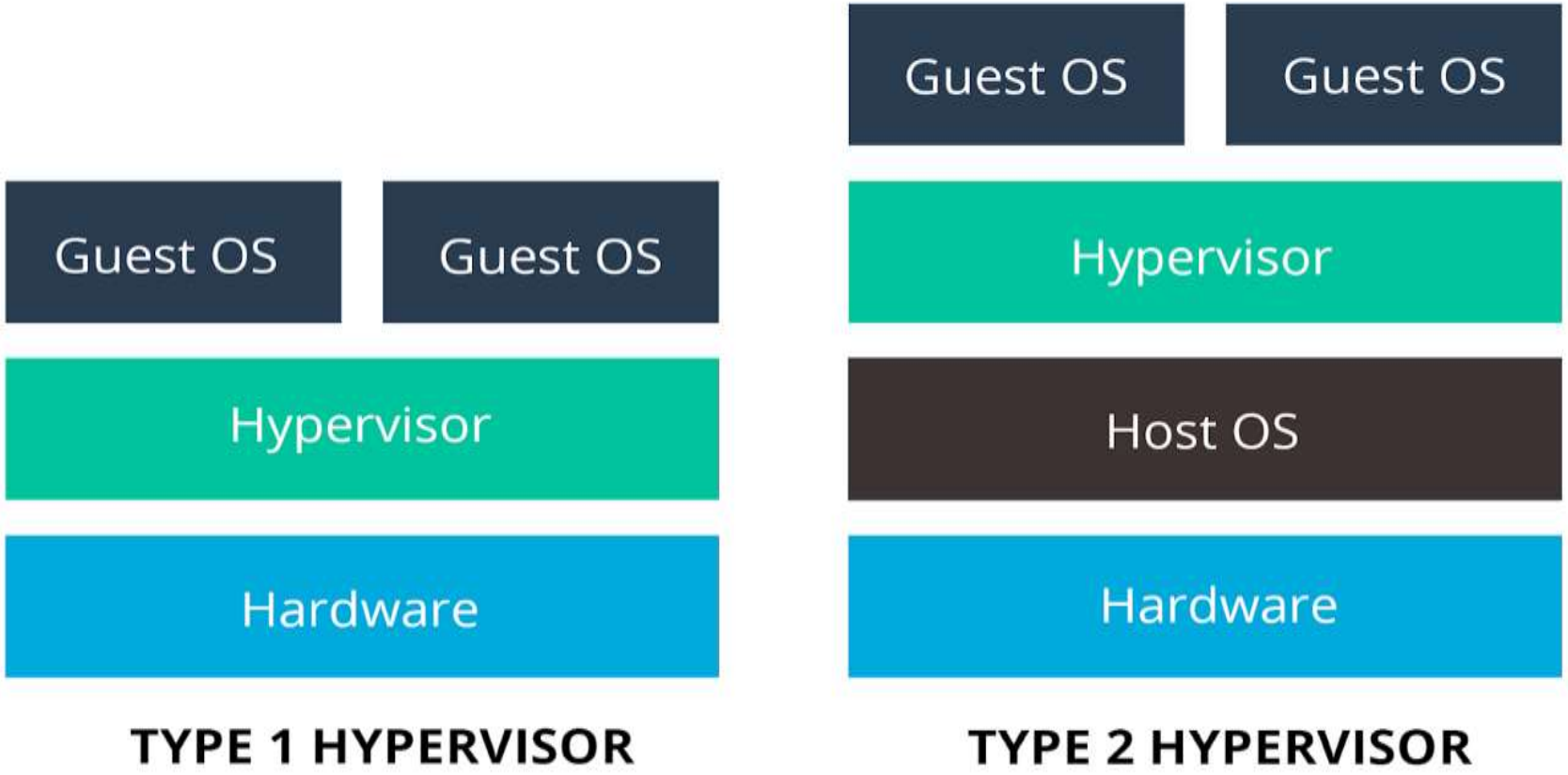
- It is the programming API used by the application level for the execution of some code. At this level, API can be called ABI (Application Binary Interface).
- It is automated to convert programmatic logic to the binary format which cut downs the emulation process.
- **Eg:Wine**

5) Application Level

The code is written and executed at this level only after the successful completion of all other four levels.

Eg: JVM/.Net

Virtualisation Structure



- Virtualization is performed with the help of a hypervisor or VMM.
- Hypervisors are of two types - Type 1 and Type 2

Type 1

- Hypervisor runs directly on the host with simple programming.
- This doesn’t require an individual operating system to compute.
- It is called bare metal or native. It lies between the OS and hardware.
- This type of Hypervisor is suited for the enterprise.

Type 2

- Hypervisor functions on top of the OS.
- It is called a host machine.
- This implies that there will be no direct point of access to the hardware.
- And the VM running in this type will be managed by the Virtual Machine Monitor (VMM).

- Before virtualization --> OS manages the hardware.ð
- After virtualization --> a virtualization layer is inserted between the hardware and the OS.ð
- The virtualization layer converts portions of the real hardware into virtual Hardware.
- Thus different operating systems such as Linux and Windows can run on the same physical machine, simultaneously.
- Depending on the position of the virtualization layer, there are several classes of VM architectures

❑ the hypervisor architecture

❑ paravirtualization

❑ host-based virtualization

1.Hypervisor and XEN Architecture

- Hardware virtualization technique allowing multiple OS called guests to run on a host machine.
- Also called the Virtual Machine Monitor (VMM).
- Supports hardware-level virtualization on bare metal devices like CPU, memory, disk and network interfaces
- Sits directly between the physical hardware and its OS.

- "So the size of the Xen hypervisor is kept rather small.
- "Xen provides a virtual environment located between the hardware and the OS.

1.1 The Xen Architecture

- Xen is an open source hypervisor program developed by Cambridge University.
- Xen is a micro-kernel hypervisor, which separates the policy from the mechanism.
- The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain 0.

The core components of a Xen system are the

- ❑Hypervisor
- ❑ kernel
- ❑ Applications.

- The organization of the three components is important.
- Like other virtualization systems, many guest OSes can run on top of the hypervisor.
- However, not all guest OSes are created equal, and one in

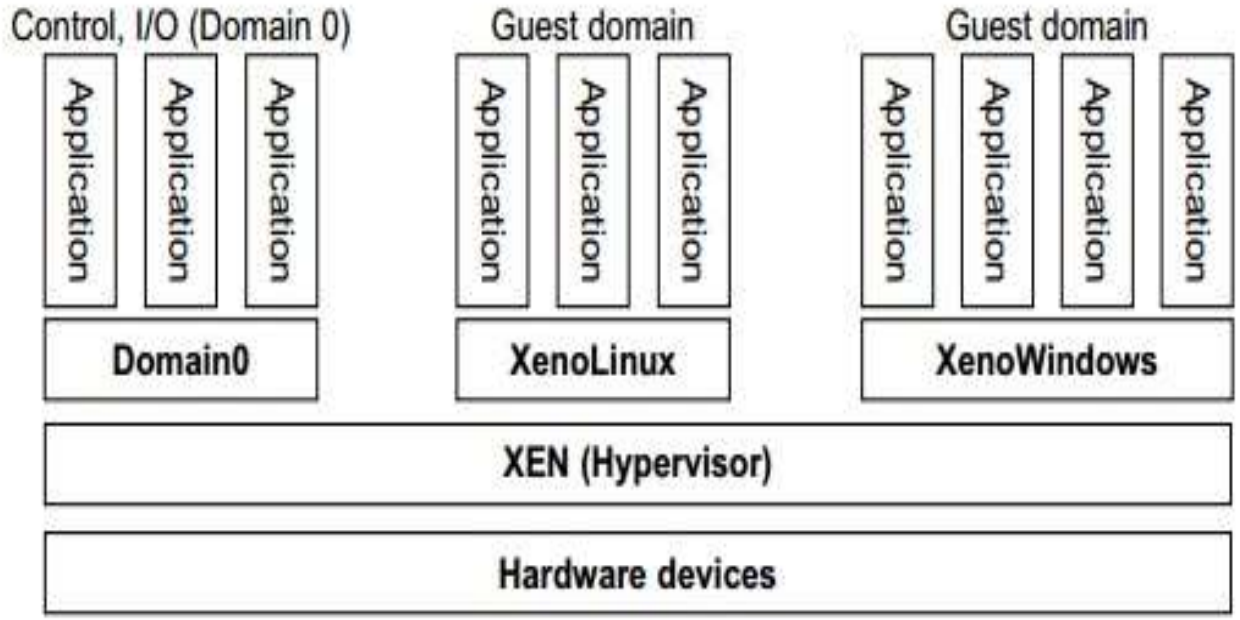


FIGURE 3.5

The Xen architecture's special domain 0 for control and I/O, and several guest domains for user applications.

2.Binary Translation with Full Virtualization

- " Depending on implementation technologies, hardware virtualization can be classified into two categories
- " full virtualization
- " host-based virtualization.
- " Full virtualization no need to modify the host OS.
- " Relies on binary translation to trap and to virtualize the execution of certain sensitive, non-virtualizable instructions.

2.1 Full Virtualization

- " With full virtualization, noncritical instructions run on the hardware directly while critical
- instructions are discovered and replaced with traps into the VMM to be emulated by software.
- " Binary translation can incur a large performance overhead.
- " Noncritical instructions do not control hardware or threaten the security of the system, but critical instructions do.
- " Therefore, running noncritical instructions on hardware not only can promote efficiency, but also can ensure system security.

2.2 Binary Translation of Guest OS Requests Using a VMM

- " VMware: puts the VMM at Ring 0 and the guest OS at Ring 1.
- " The VMM scans the instruction stream and identifies the privileged, control- and behavior-
- sensitive instructions.
- " Once identified trapped into the VMM emulates the behavior of these instructions.ð ð
- " The method used in this emulation is called binary translation.

2.3 Host Based Virtualization

- Alternative VM architecture install a virtualization layer on top of the host OS.ð
- This host OS is still responsible for managing the hardware.
- The guest OSes are installed and run on top of the virtualization layer.
- Dedicated applications may run on the VMs.

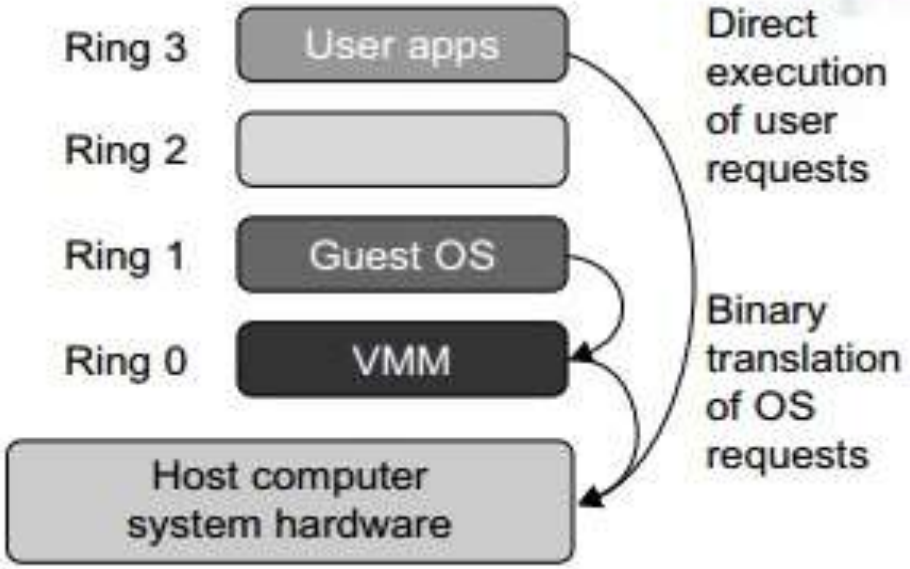


FIGURE 3.6

Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

3.Para-Virtualization with Compiler Support

- Para-virtualization needs to modify the guest operating systems.
- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.
- Performance degradation is a critical issue of a virtualized system.
- Para-virtualization attempts to reduce the virtualization overhead - performance is improved by modifying only the guest OS kernel.

3.1Para-Virtualization Architecture

- When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.
- According to the x86 ring definition, the virtualization layer should also be installed at Ring 0.
- Different instructions at Ring 0 may cause some problems.
- Para-virtualization replaces non virtualizable instructions with hypercalls that communicate directly with the hypervisor or VMM.
- However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.

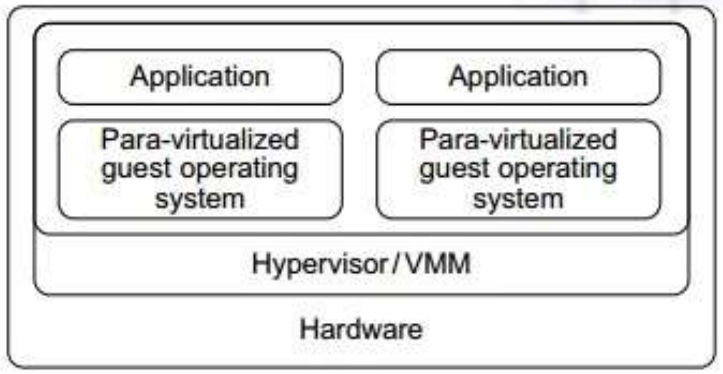


FIGURE 3.7

Para-virtualized VM architecture, which involves modifying the guest OS kernel to replace nonvirtualizable instructions with hypercalls for the hypervisor or the VMM to carry out the virtualization process (See Figure 3.8 for more details.)

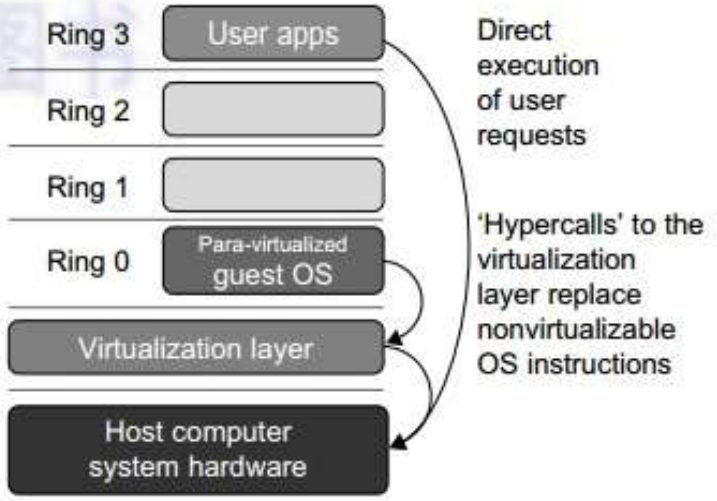


FIGURE 3.8

The use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls.

(*Courtesy of VMWare [71]*)

3.2 KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel.
- Memory management and scheduling activities are carried out by the existing Linux kernel.
- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine.
- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

3.3 Para-Virtualization with Compiler Support

- Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, para-virtualization handles these instructions at compile time.
- The guest OS kernel is modified to replace the privileged and sensitive instructions with hypercalls to the hypervi-sor or VMM.
- Xen assumes such a para-virtualization architecture.

VIRTUALIZATION OF CPU, MEMORY, AND I/O DEVICES

- To support virtualization, processors such as the x86 employ a special running mode and instructions, known as hardware-assisted virtualization.
- In this way, the VMM and guest OS run in different modes and all sensitive instructions of the guest OS and its applications are trapped in the VMM.

1. Hardware Support for Virtualization

- Modern operating systems and processors permit multiple processes to run simultaneously.
- If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash.
- Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware.
- Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions.

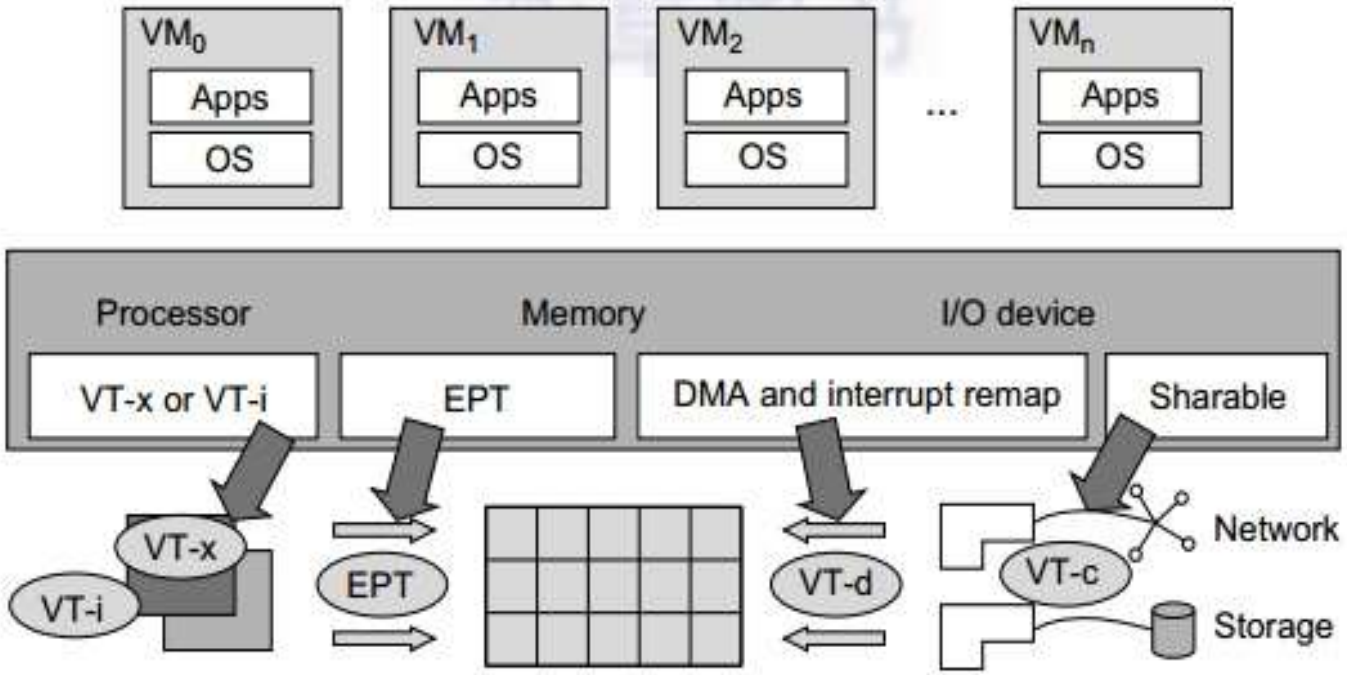


FIGURE 3.10

Intel hardware support for virtualization of processor, memory, and I/O devices.

2. CPU Virtualization

- A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode.
- Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency.
- Other critical instructions should be handled carefully for correctness and stability.
- The critical instructions are divided into three categories:

❑ privileged instructions,

❑ control-sensitive instructions, and

❑ behavior-sensitive instructions.

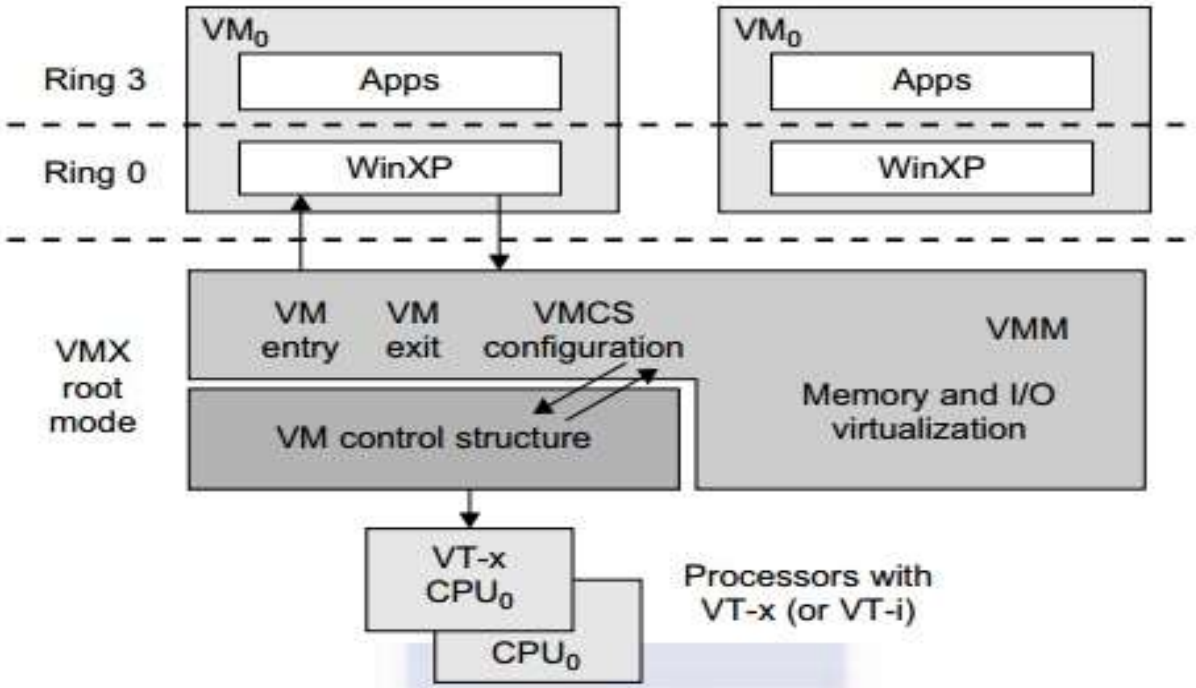


FIGURE 3.11

Intel hardware-assisted CPU virtualization.

3. Memory Virtualization

- Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems.
- In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory.
- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.

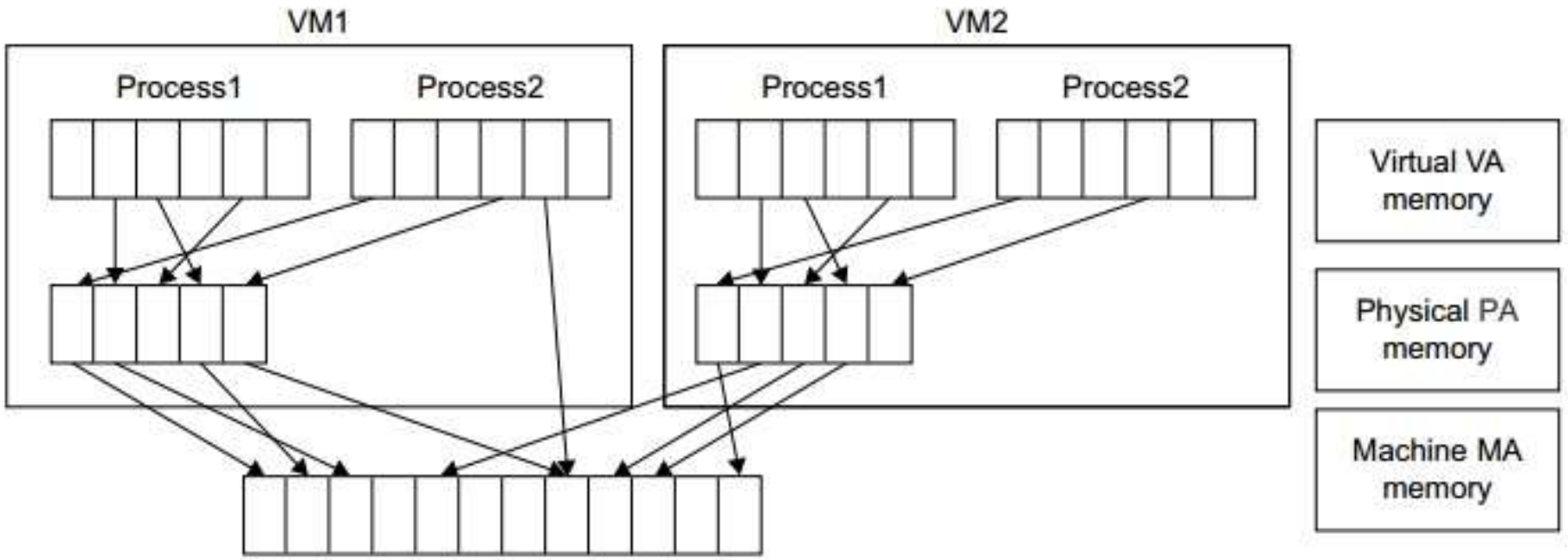


FIGURE 3.12

Two-level memory mapping procedure.

4. I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
- At the time of this writing, there are three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O.
- Full device emulation is the first approach for I/O virtualization.
- Generally, this approach emulates well-known, real-world devices.

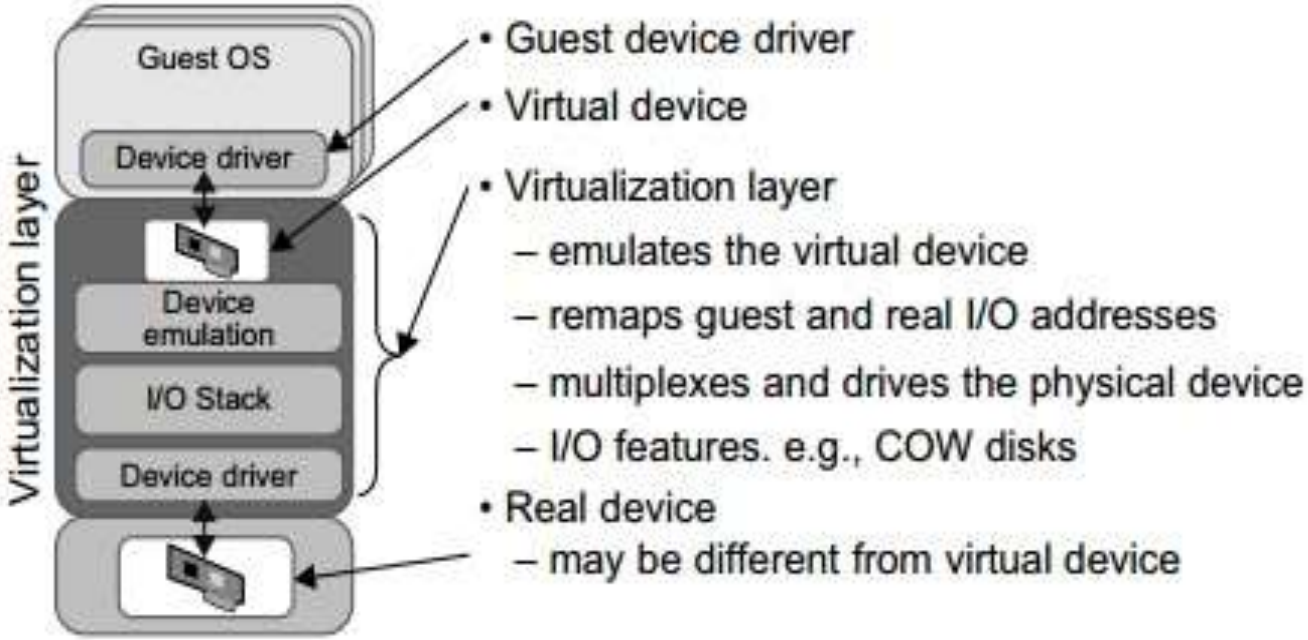


FIGURE 3.14

Device emulation for I/O virtualization implemented inside the middle layer that maps real I/O devices into the virtual devices for the guest device driver to use.

Virtualization in Multi-Core Processors

- Virtualizing a multi-core processor is relatively more complicated than virtualizing a uni-core processor.
- Though multicore processors are claimed to have higher performance by integrating multiple processor cores in a single chip, muti-core virtualiuzation has raised some new challenges to computer architects, compiler constructors, system designers, and application programmers.

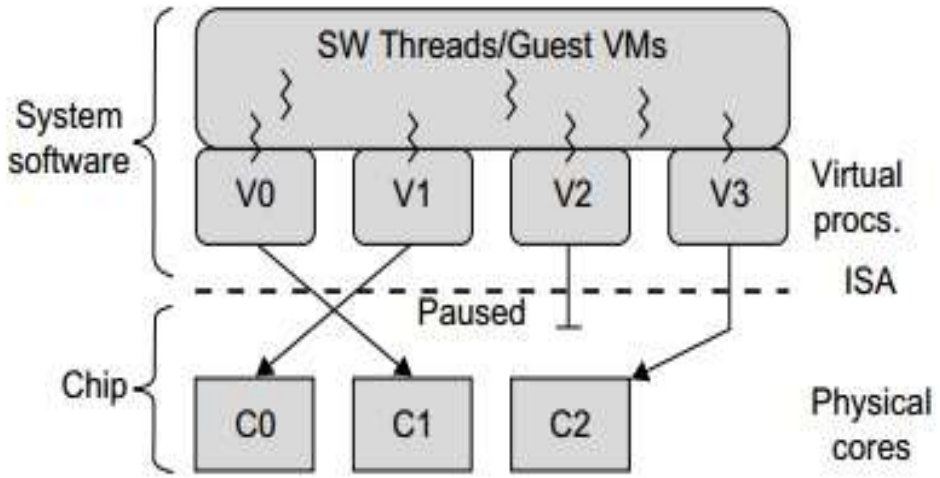


FIGURE 3.16

Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.

Virtualization support

- Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines.
- Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine
- Virtualization support for RISC-based systems
- Operating system native clustering, partitioning, or virtualization are supported for all RISC-based platforms that Sterling B2B Integrator supports: AIX® on P5 and above:
  - Partitioning/Virtualization – PowerVM, LPAR, dLPAR
  - Clustering/Failover - HACMP

VIRTUALIZATION DISASTER RECOVERY

DISASTER RECOVERY

- DR relies upon the replication of data and computer processing in an off-premises location not affected by the disaster.
- When servers go down because of a natural disaster, equipment failure or cyber attack, a business needs to recover lost data from a second location where the data is backed up.

VIRTUAL DISASTER RECOVERY

- Virtual disaster recovery refers to the use of virtualized workloads for disaster recovery planning and **failover**.
- To achieve this, organizations need to regularly replicate workloads to an offsite virtual disaster recovery site.
- For enterprises in virtual environments, **VM replication** provided by hypervisor vendors may be enough.
- As for physical or hybrid environments, **physical to virtual conversion (P2V)** is still needed before replication.


USERS
NORMAL
FAILOVER
SOURCE SERVER
TARGET SERVER
Local or Remote
REPLICATION
RESTORE
HOST-BASED
SOFTWARE
WAN / LAN
RECOVERY MADE EASY

Virtual disaster recovery vs physical disaster recovery

- Virtualization brings great flexibility to traditional physical disaster recovery.
- It is much easier to create an image-level backup of a virtual machine and start it on **another bare-metal host** than migrating workloads to a remote physical site that needs to be rebuilt.

Disaster recovery type	Physical disaster recovery	Virtual disaster recovery
Infrastructure	physical	virtual
Space utilization	lower	higher
Recovery points	less	more
Recovery time	more	less
Costs	higher	lower
Test difficulty	higher	lower
Instant undo feature	no	snapshots

- A good disaster recovery plan is the one that is best suited to the enterprises' actual situation. In practice, the following factors are most often taken into consideration:

➢ **Recovery Time Objective (RTO) --** the measure of downtime

➢ **Recovery Point Objective (RPO) --** the measure of data loss

➢ **Test Time Objective (TTO) --** the measure of testing ease

- A disaster recovery plan in virtualization technology with ideal RTO and RPO allows enterprises to quickly switch to disaster recovery mode, but it can also be costly.
- Therefore, it is important to make bold assumptions about maximum tolerable downtime and data loss.
- In addition, to further reduce costs, affordable third-party software can also be taken into consideration.
- For example, **AOMEI Cyber Backup** allows you to automate multiple VMware ESXi or Hyper-V **VMs backup and restore** for free.