Cloud

Enabling Technologies

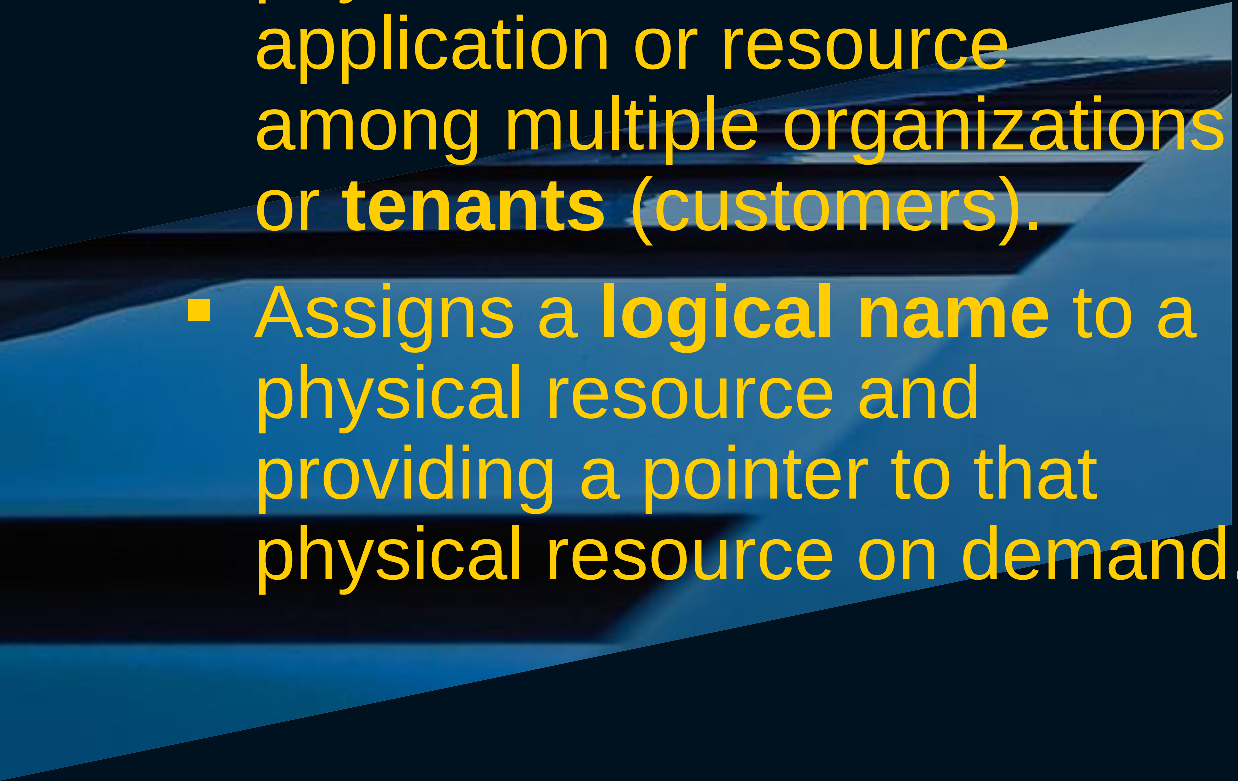
Dr. Syed Imtiyaz Hassan
Assistant Professor, Deptt. of CSE,
Jamia Hamdard
(Deemed to be University),
New Delhi, India.
https://syedimtiyazhassan.org
s.imtiyaz@jamiahamdard.ac.in

Enabling Technologies

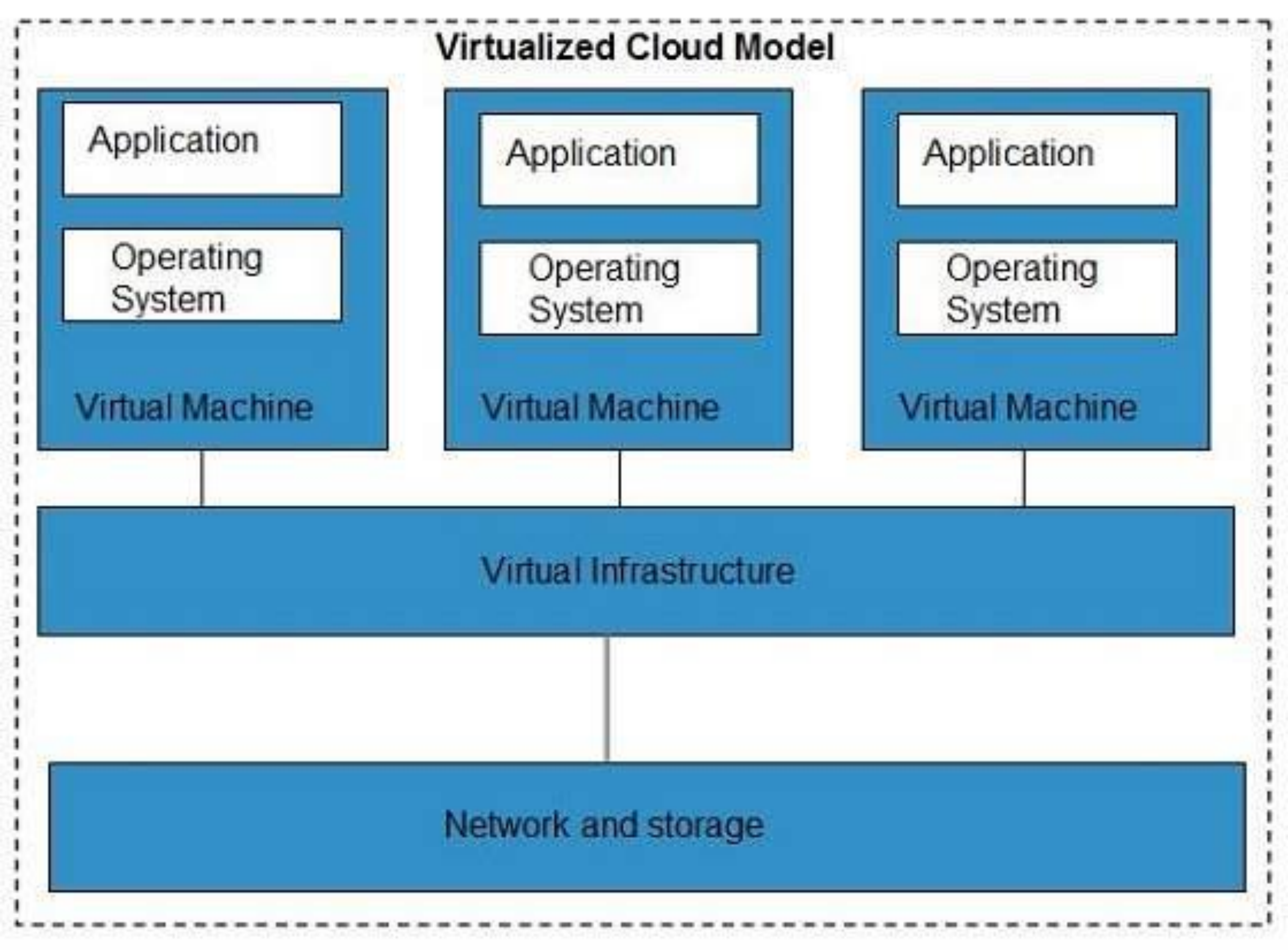
- Virtualization & Multitenancy
- Service-Oriented Architecture (SOA)
- Grid Computing
- Utility Computing

Virtualization

- Allows to **share** single physical instance of an application or resource among multiple organizations or **tenants** (customers).
- Assigns a **logical name** to a physical resource and providing a pointer to that physical resource on demand.



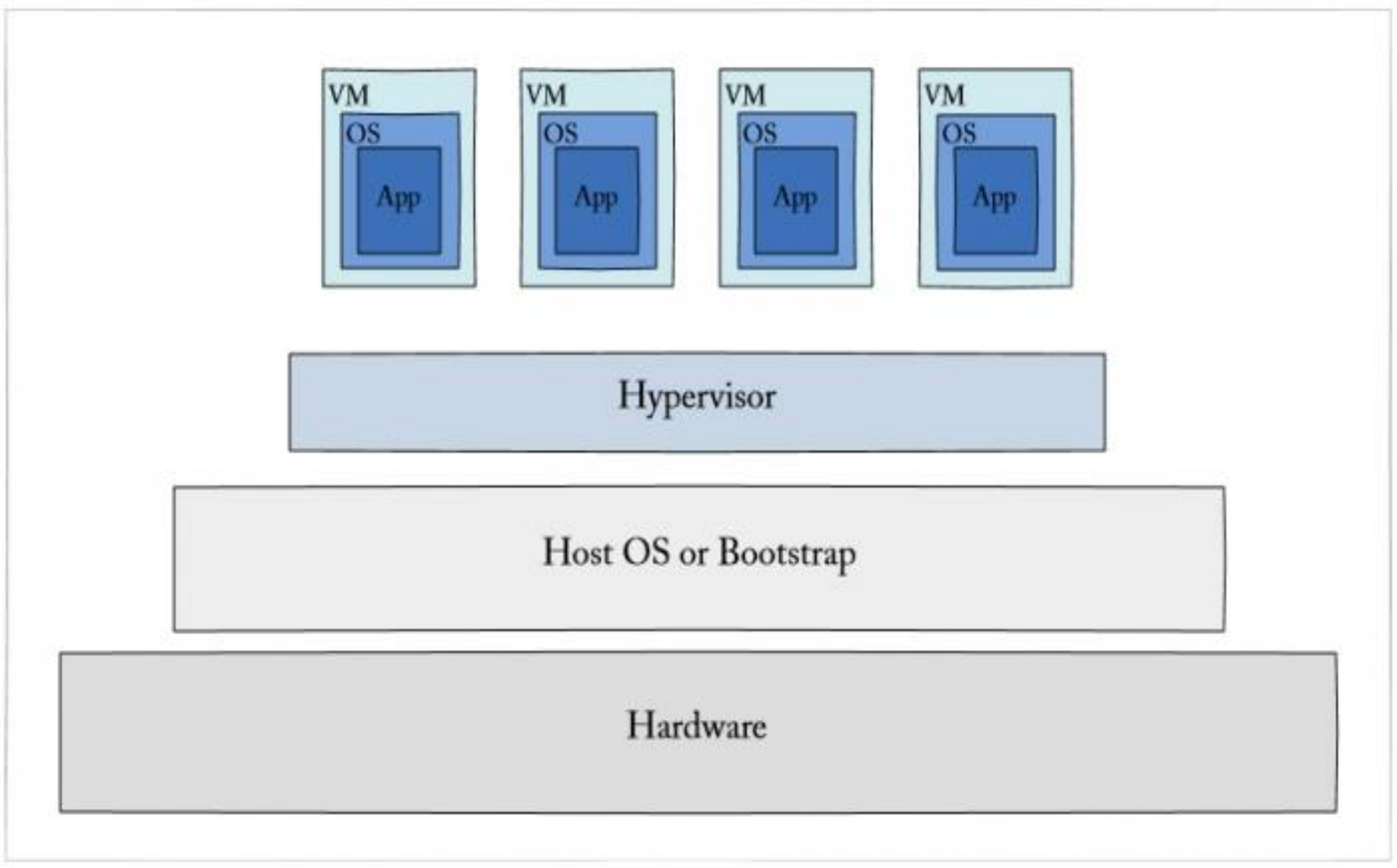
Virtualization



Virtualization

- Creating a virtual machine over existing operating system and hardware is referred as **Hardware Virtualization.**
- The machine on which the virtual machine is created is known as **host machine.**
- The virtual machine is referred as **guest machine**.
- This virtual machine is managed a software or firmware, which is known as **hypervisor**.

Virtualization

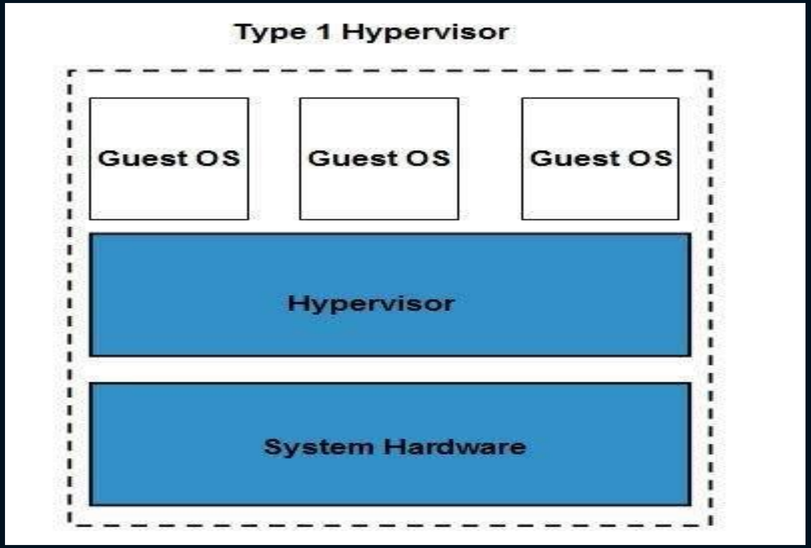


Hypervisor

- The hypervisor is a firmware or low-level program that acts as a **Virtual Machine Manager**.
    - Type 1
    - Type 2

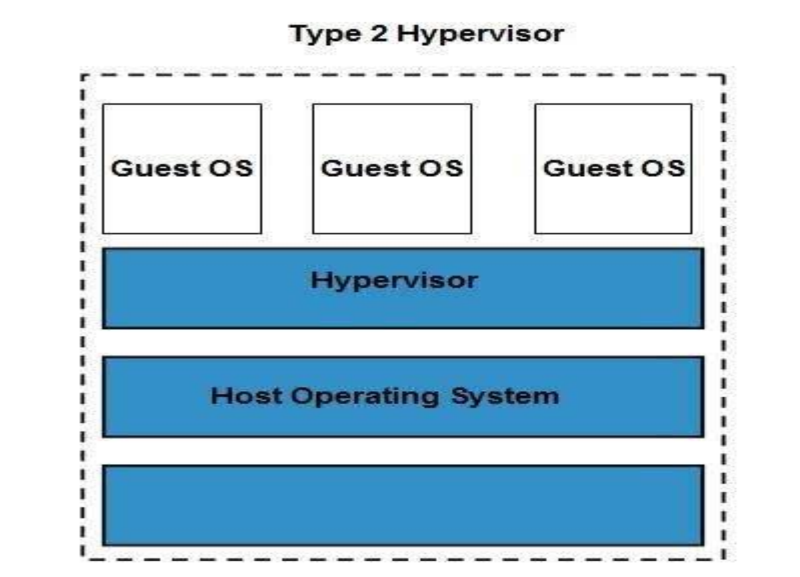
Type 1 Hypervisor

- Executes on **bare system**.
- Does **not** have any **host** operating system.
- LynxSecure, RTS Hypervisor, Oracle VM, VirtualLogic VLX



Type 2 Hypervisor

- **Software interface** that emulates the devices with which a system normally interacts.
- Hyper V, VMWare Fusion, Virtual Server 2005 R2, Windows Virtual PC



Types of Virtualization

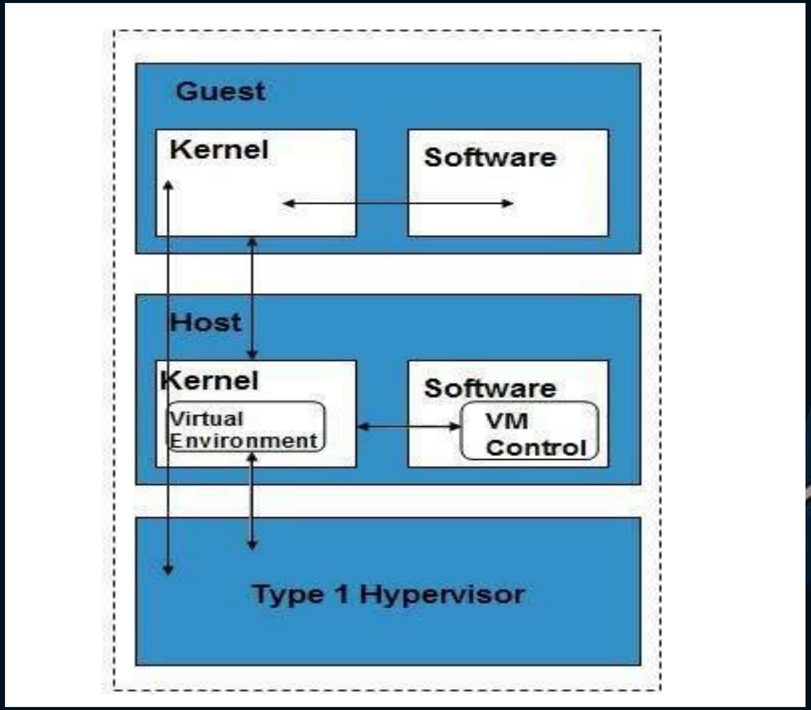
- Hardware Virtualization
- Software Virtualization
- Memory Virtualization
- Storage Virtualization
- Network Virtualization

Hardware Virtualization

- Involves embedding virtual machine software into the **server's hardware** components.
- **Types**
  - Full Virtualization
  - Emulation Virtualization
  - Para-virtualization

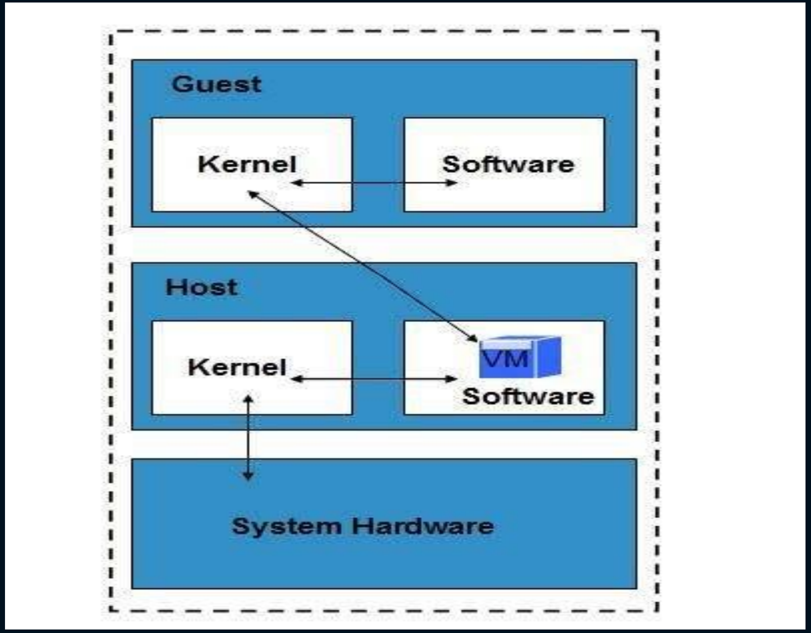
Full Virtualization

- The underlying hardware is **completely simulated.**



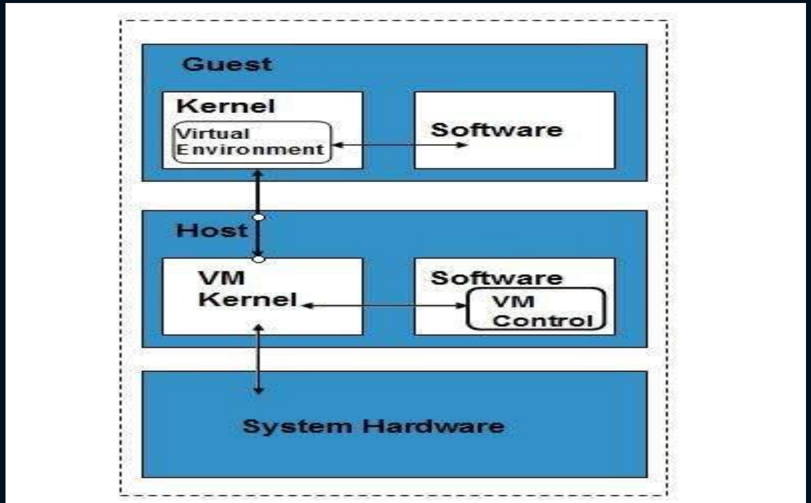
Emulation Virtualization

- The **virtual machine** simulates the hardware.



Para Virtualization

- The hardware is **not simulated**.
- The **guest software** run their own isolated domains.
- The guest and the host are aware of one another.



Software Virtualization

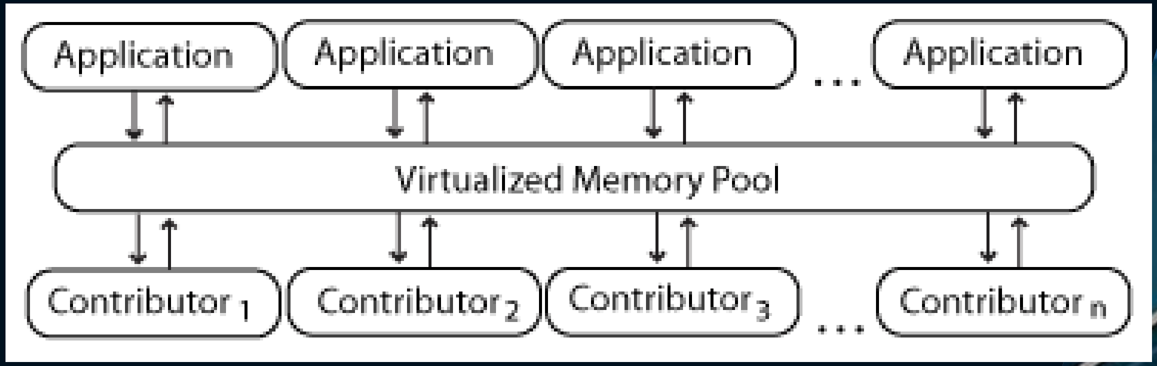
- Operating system
- Application virtualization
- Service virtualization

Memory Virtualization

- Introduces a way to **decouple** memory from the server to provide a shared, distributed or networked function.
- Enhances performance by providing greater memory **capacity** without any addition to the main memory.
- **Types**
  - Application-level integration
  - Operating System Level Integration

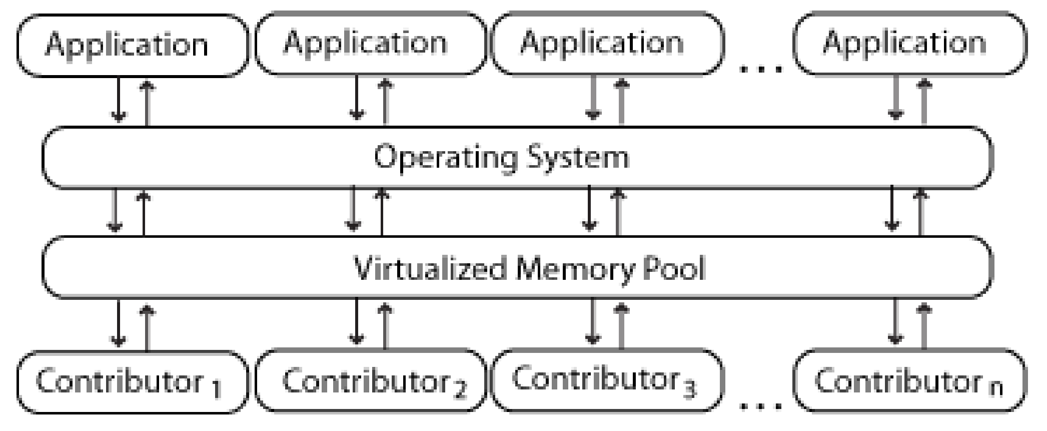
Application-level Integration

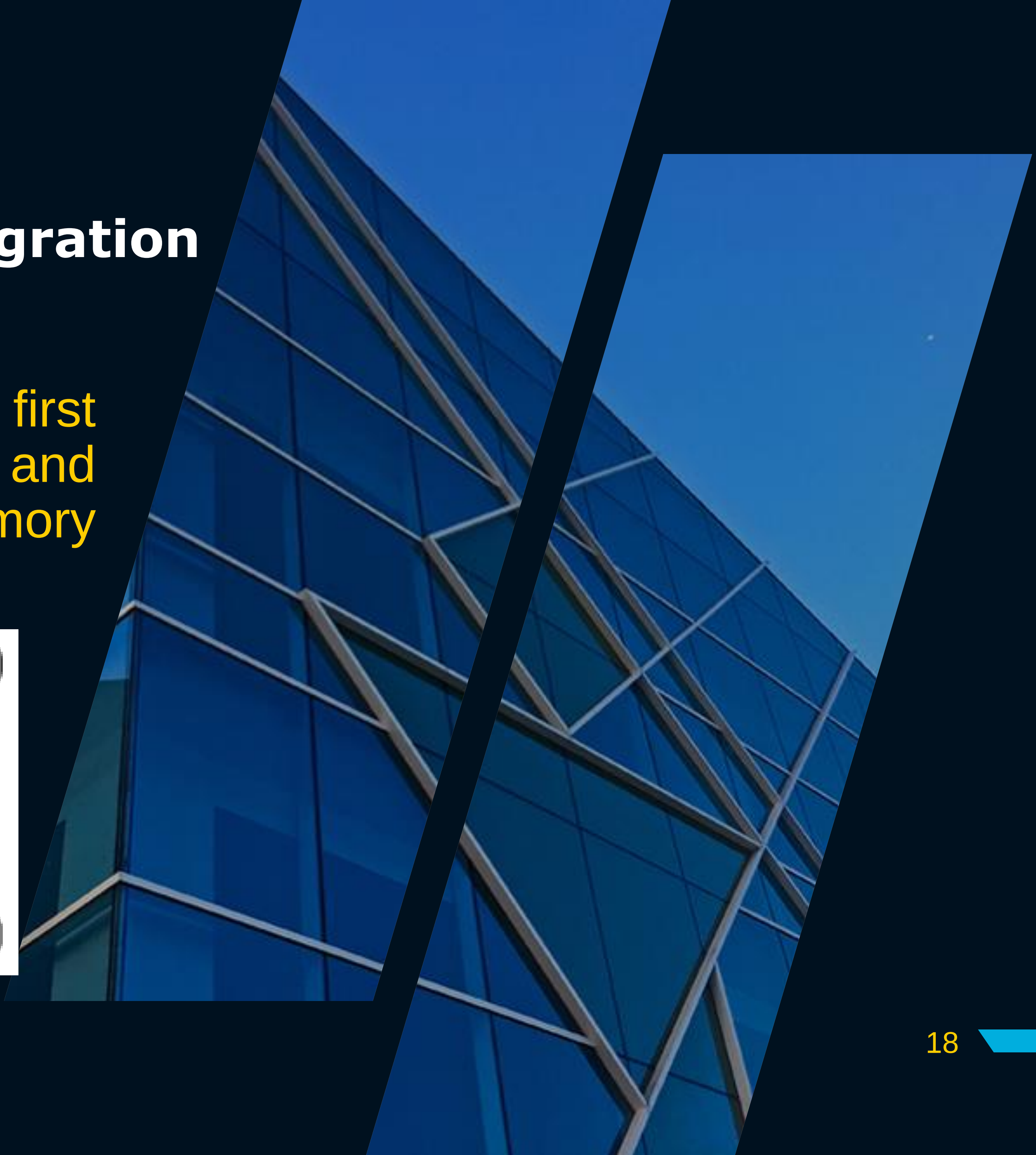
- Applications running on connected computers **directly connect** to the memory pool through an **API** or the file system.



Operating System Level Integration

- The **operating system** first connects to the memory pool, and makes that pooled memory available to applications.





Storage Virtualization

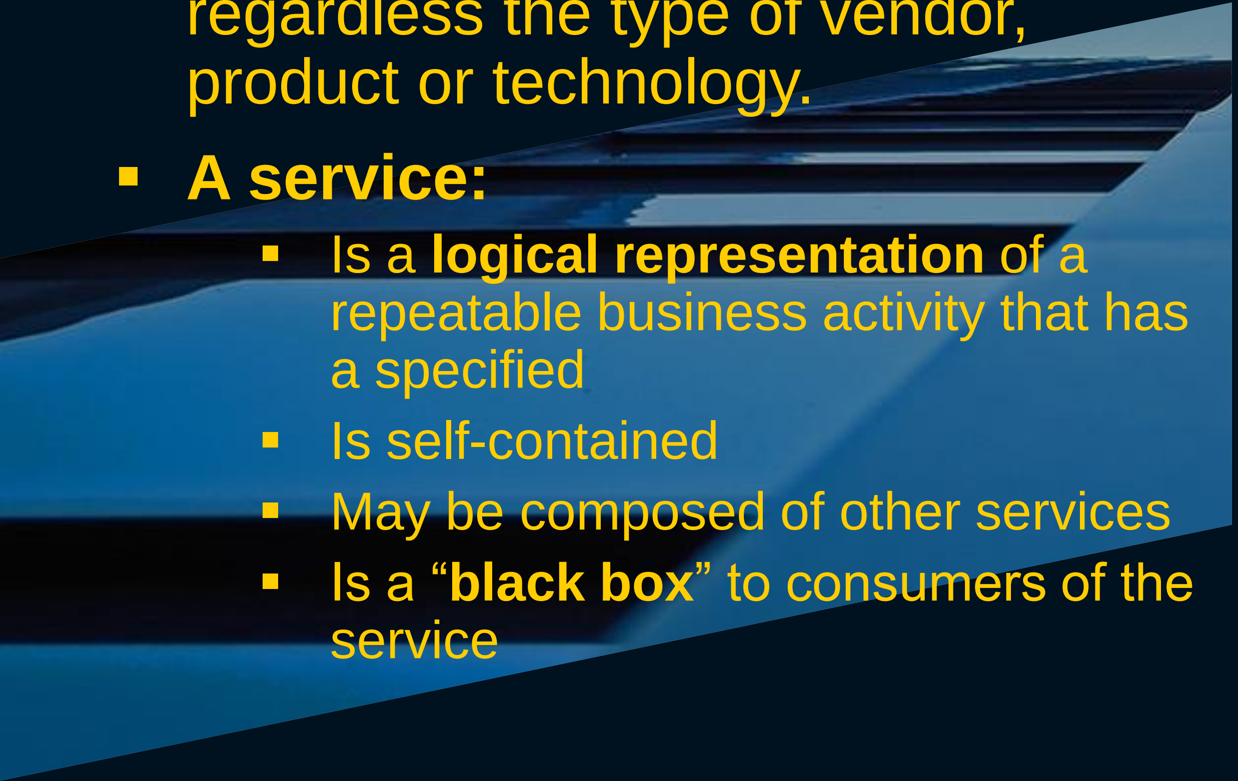
- Multiple physical storage devices are **grouped together**, which then appear as a single storage device.
- Types
  - **Block Virtualization**: It works before the file system exists. It replaces controllers and takes over at the disk level.
  - **File Virtualization:** Storage system grants access to files that are stored over multiple hosts.

Network Virtualization

- The management and monitoring of a computer network as a **single managerial entity** from a single software-based administrator’s console.
- Types
  - **Internal network**: Enables a single system to function like a network.
  - **External network**: Consolidation of multiple networks into a single one, or segregation of a single network into multiple ones.

Service-Oriented Architecture

- Helps to use applications as a service for other applications regardless the type of vendor, product or technology.
- **A service:**
  - Is a **logical representation** of a repeatable business activity that has a specified
  - Is self-contained
  - May be composed of other services
  - Is a “**black box**” to consumers of the service



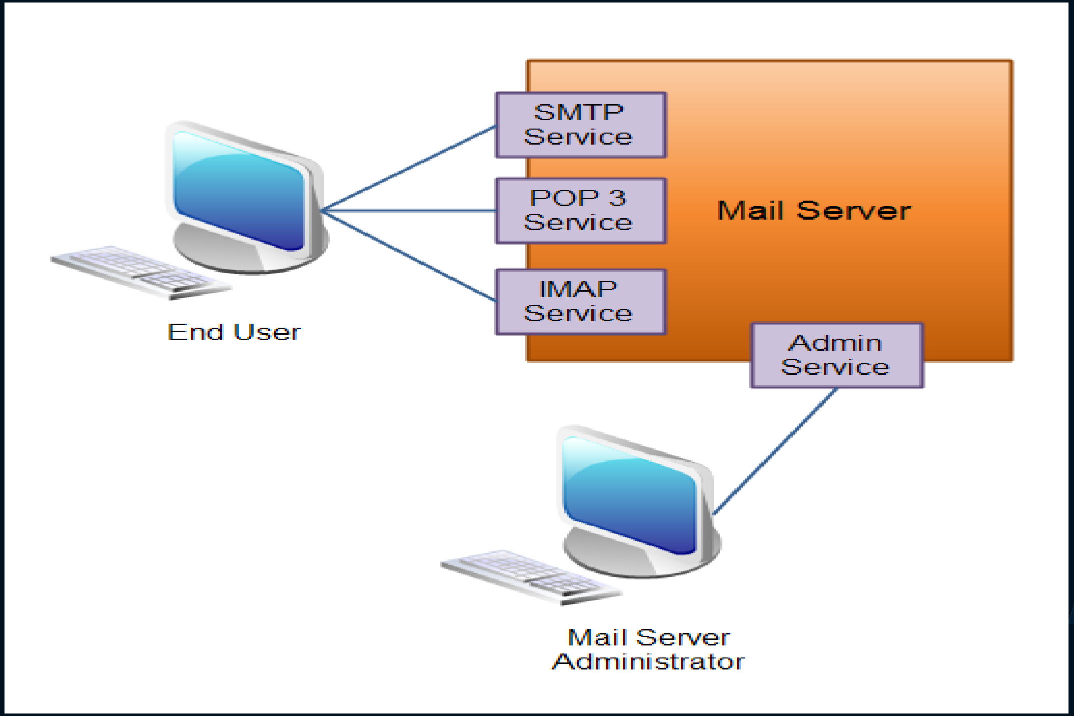
Service-Oriented Architecture

- Services vs. Applications

Services	Applications
Perform a single or a few specialized operations.	Perform a wide range of operations, and may even expose some of these operations as services.
Most often accessed by other programs.	Often (but not always) accessed by humans.
Often (but not always) targets part of a larger problem domain.	Often (but not always) targets a whole problem domain.

Service-Oriented Architecture

- A mail server (application) exposing 4 different services.



Service-Oriented Architecture

Advantages

- SOA allows **reuse** the service of an existing system alternately building the new system.
- It allows **plugging** in new services or upgrading existing services to place the new business requirements.
- It can **enhance the performance**, functionality of a service and easily makes the system upgrade.
- SOA has capability to **adjust or modify** the different external environments and large applications can be managed easily.
- It provides **reliable** applications in which you can test and debug the independent services easily as compared to large number of code.

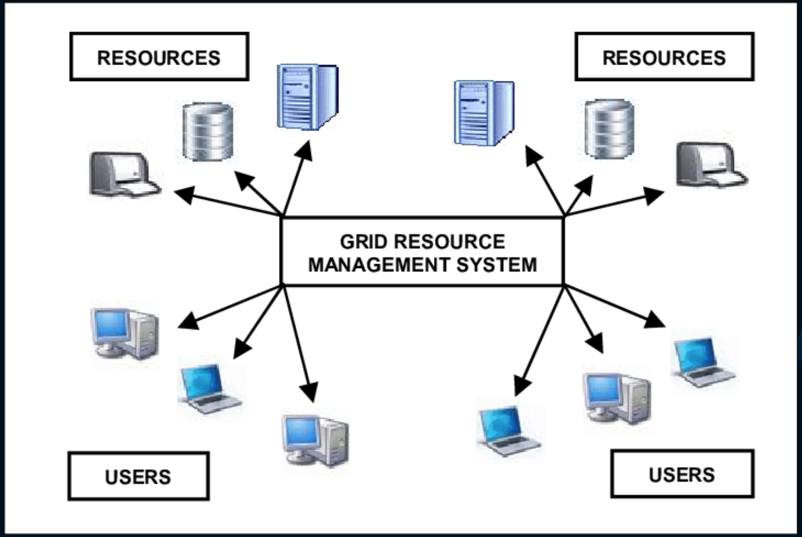
Service-Oriented Architecture

Limitations

- SOA requires **high investment cost** (means large investment on technology, development and human resource).
- There is **greater overhead** when a service interacts with another service which increases the response time and machine load while validating the input parameters.
- SOA is **not** suitable for **GUI** (graphical user interface) applications which will become mor complex when the SOA requires the heavy d exchange.

Grid Computing

- Grid Computing refers to **distributed computing**, in which a group of computers from multiple locations are **connected** with each other to achieve a common objective.
- In a basic grid computing system, every computer can access the resources of every other computer belonging to the network.

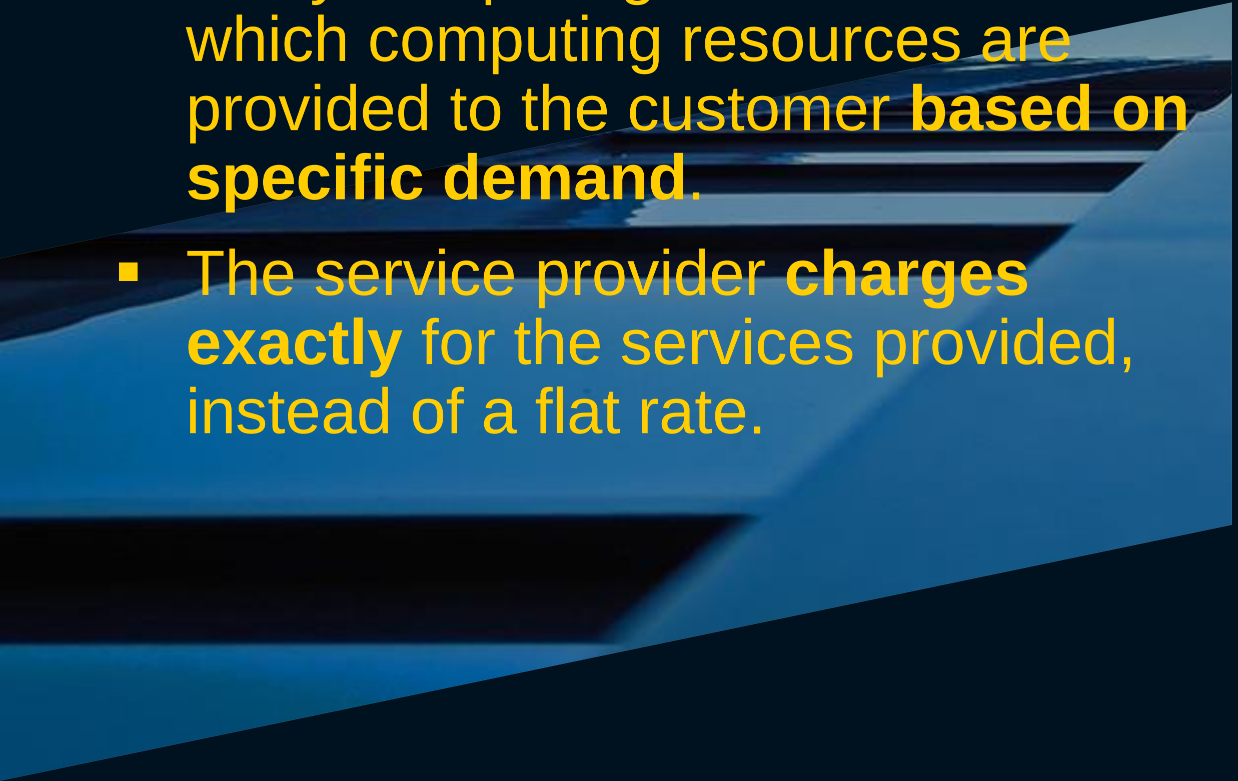


Grid Computing

BASIS FOR COMPARISON	CLOUD COMPUTING	GRID COMPUTING
Application focus	business and web-based applications.	Collaborative purposes.
Architecture used	Client-server	Distributed computing
Management	Centralized	Decentralized
Business model	Pay per use	No defined business model
Accessibility of services	High because it is real-time	Low because of scheduled services.
Resource usage patterns	Centralized manner	Collaborative manner
Flexibility	High	Low
Interoperability	Vendor lock-in and integration are some issues	Easily deals with interoperability between providers.

Utility computing

- **Pay-per-Use** model
- Utility computing is a model in which computing resources are provided to the customer **based on specific demand**.
- The service provider **charges exactly** for the services provided, instead of a flat rate.



THANK YOU!