Virtualization

Mr. Pathan Bilalkhan R.

Assistant Professor

Computer Science & Engineering Department

CHAPTER-2

Virtualization

Topics to be Covered

- **Virtualization of Computing,**
- **Storage and Resources.**
- **Cloud Services:**

  Introduction to Cloud Services IaaS, PaaS and SaaS

Virtualization of Computing

Cloud infrastructure is the tools used to build a cloud environment, while cloud architecture is the blueprint for how it's built.

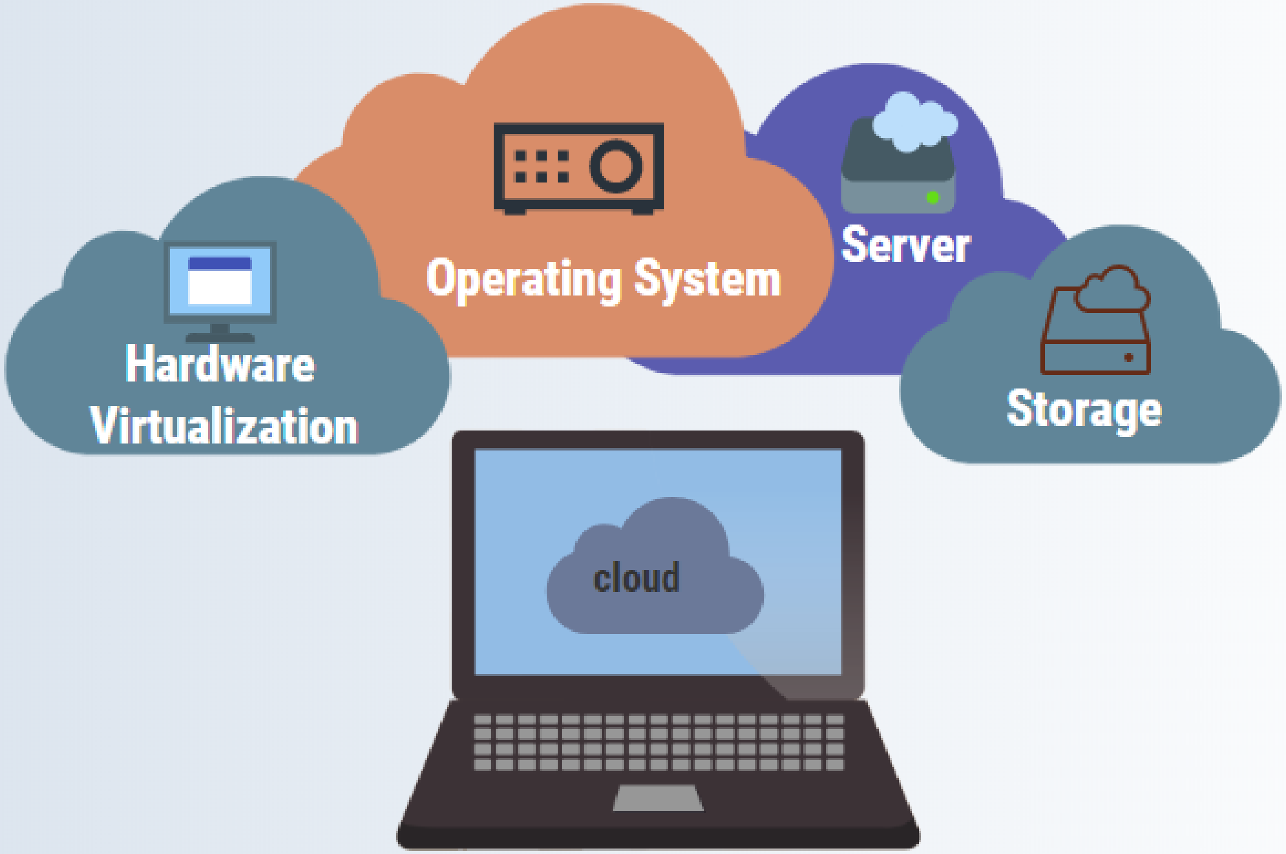
Cloud infrastructure

- The tools used to build a cloud environment
- Includes the data center building, equipment, and systems that keep it running
- Includes back-up power equipment, HVAC systems, and fire suppression equipment

Cloud architecture

- The concept or blueprint for how a cloud environment is built
- Includes the combination of components that make up a cloud environment
- Includes hardware, networks, operating systems, virtual resources, automation software, management tools, and container technologies

Virtualization in Cloud Computing



Cloud architecture components

Cloud architecture components include:

- A frontend platform
- A backend platform
- A cloud-based delivery model
- A network (internet, intranet, or intercloud)

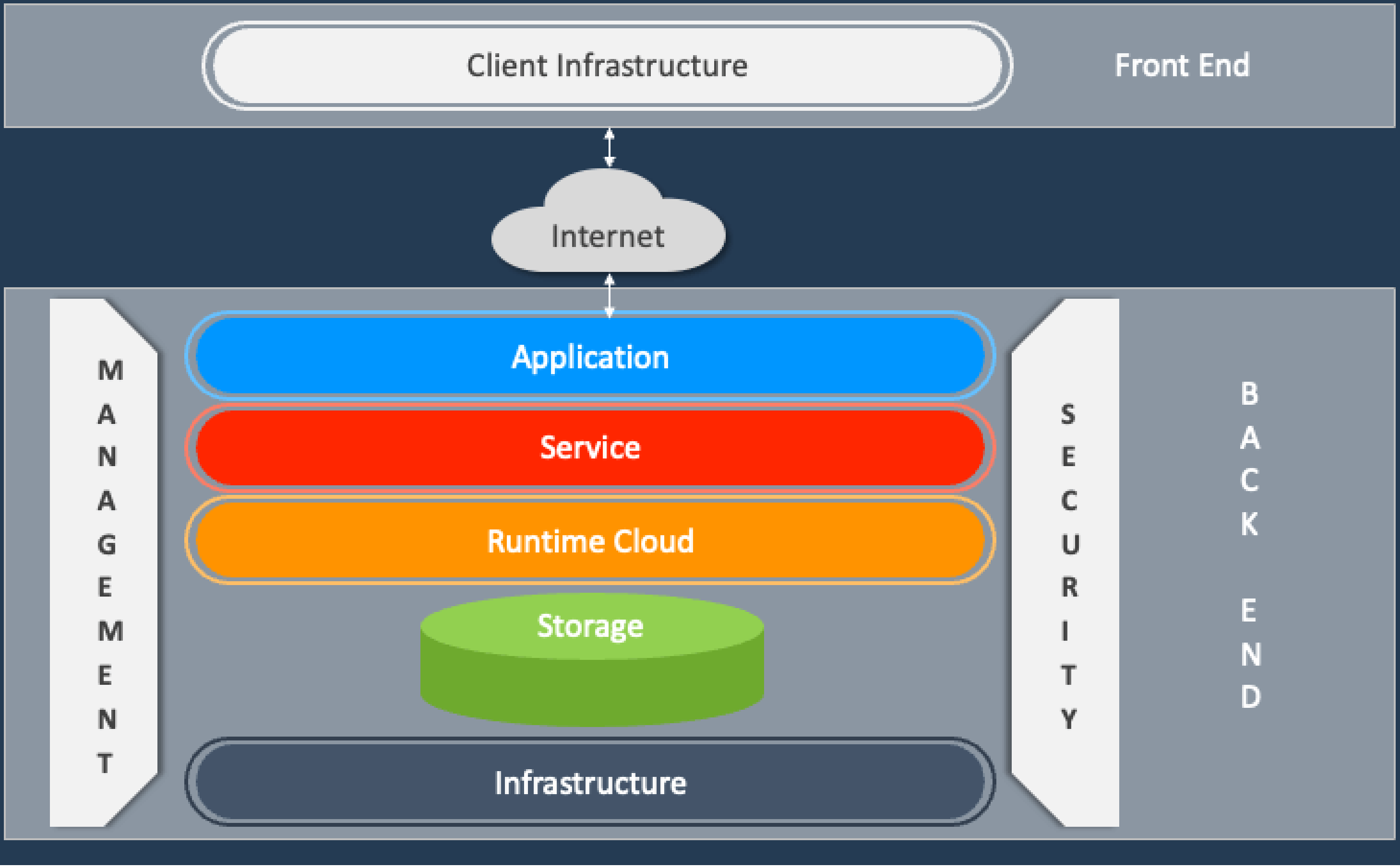
In cloud computing, frontend platforms contain the client infrastructure— user interfaces, client-side applications, and the client device or network that enables users to interact with and access cloud computing services. For example, you can open the web browser on your mobile phone and edit a Google Doc. All three of these things describe frontend cloud architecture components.

On the other hand, the back end refers to the cloud architecture components that make up the cloud itself, including computing resources, storage, security mechanisms, management, and more.

Below is a list of the main backend components:

CLOUD ARCHITECTURE

Architecture of Cloud Computing



Application: The backend software or application the client is accessing from the front end to coordinate or fulfill client requests and requirements.

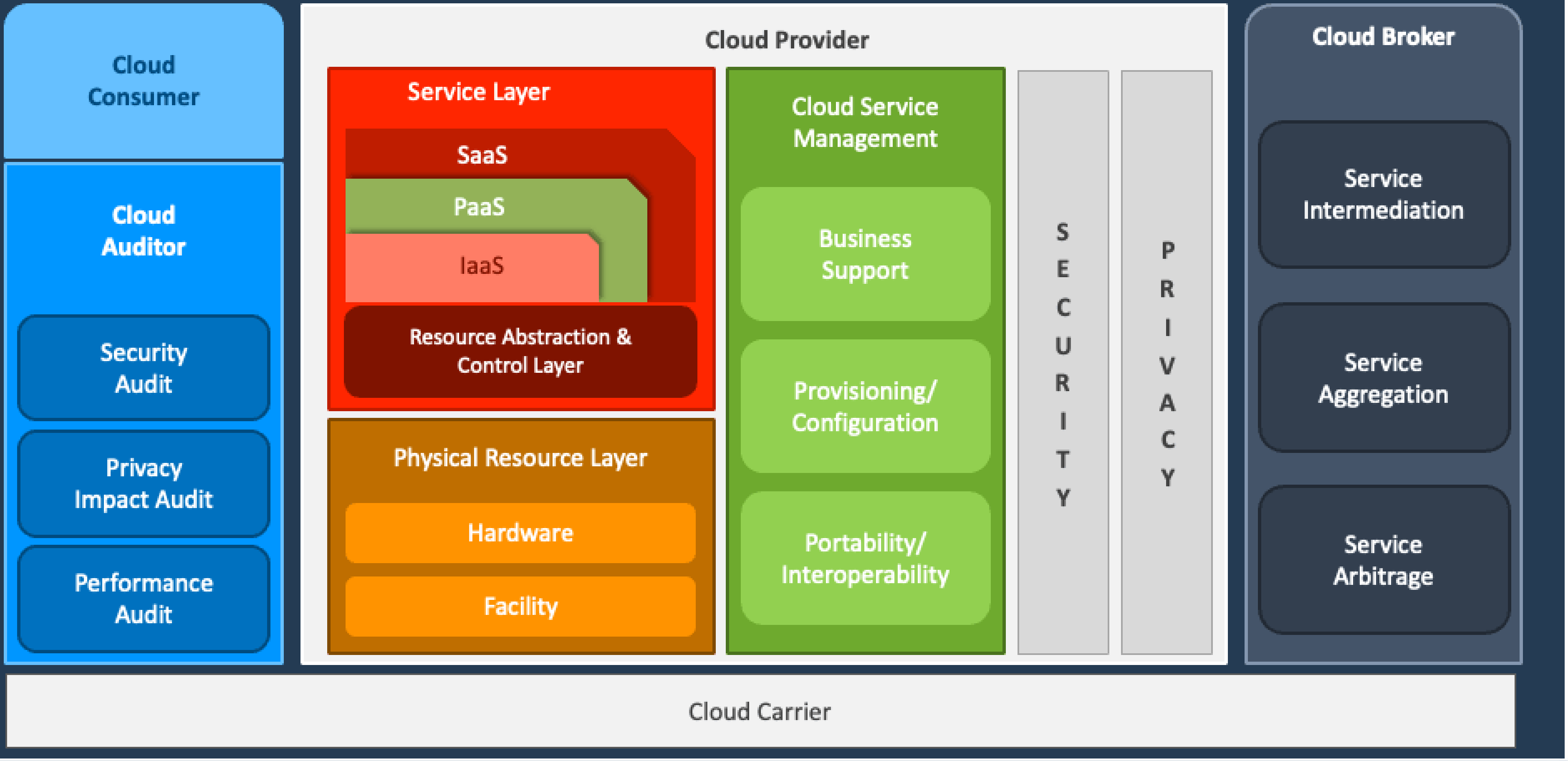
Service: The service is the heart of cloud architecture, taking care of all the tasks being run on a cloud computing system. It manages which resources you can access, including storage, application development environments, and web applications.

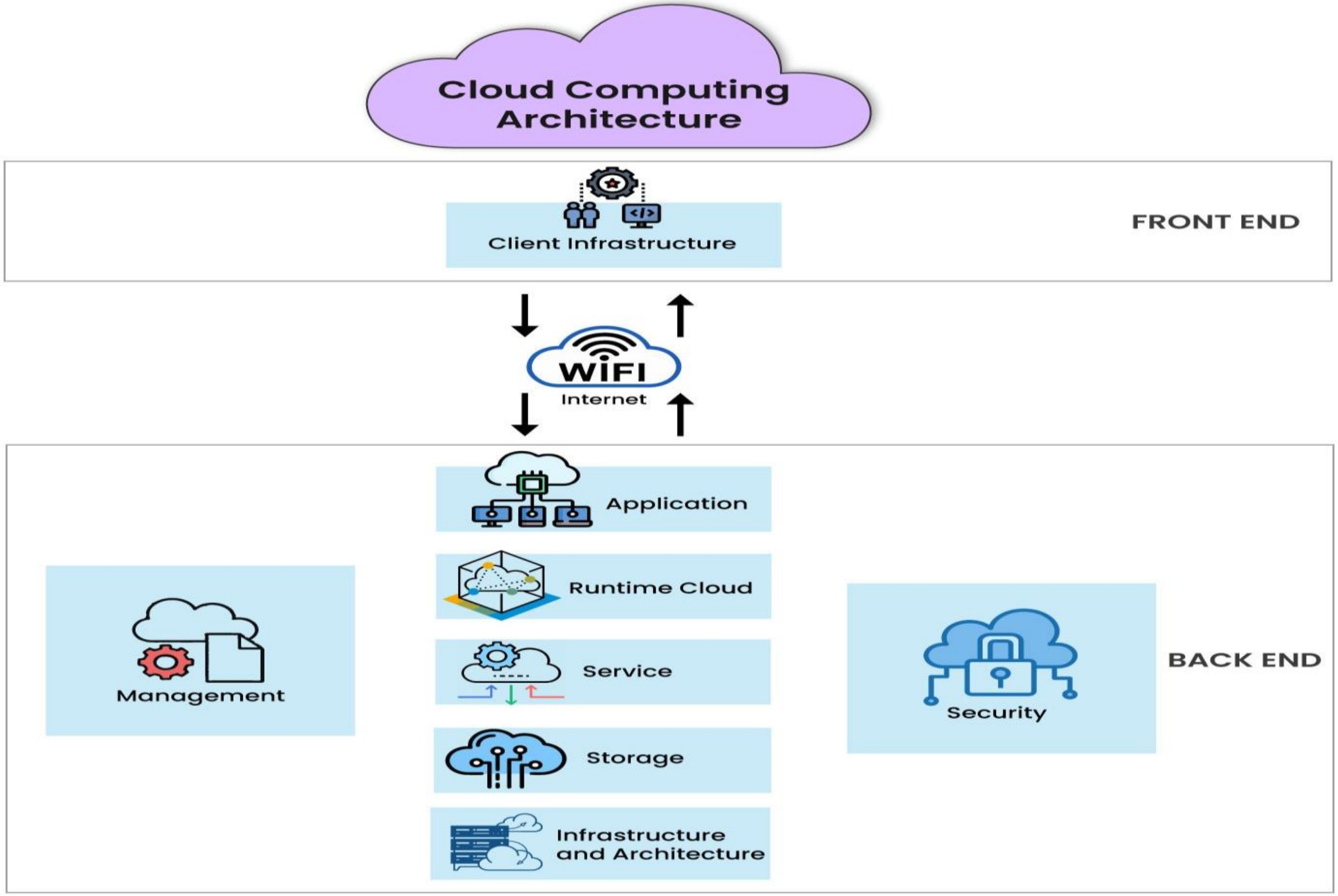
Runtime cloud: Runtime cloud provides the environment where services are run, acting as an operating system that handles the execution of service tasks and management. Runtimes use virtualization technology to create hypervisors that represent all your services, including apps, servers, storage, and networking.

Storage: The storage component in the back end is where data to operate applications is stored. While cloud storage options vary by provider, most cloud service providers offer flexible scalable storage services that are designed to store and manage vast amounts of data in the cloud. Storage may include hard drives, solid-state drives, or persistent disks in server bays.

CLOUD ARCHITECTURE

Cloud Reference Architecture





Cloud Computing Architecture

Client Infrastructure

FRONT END

WiFi

Internet

Application

Runtime Cloud

Service

Storage

Infrastructure and Architecture

Management

Security

BACK END

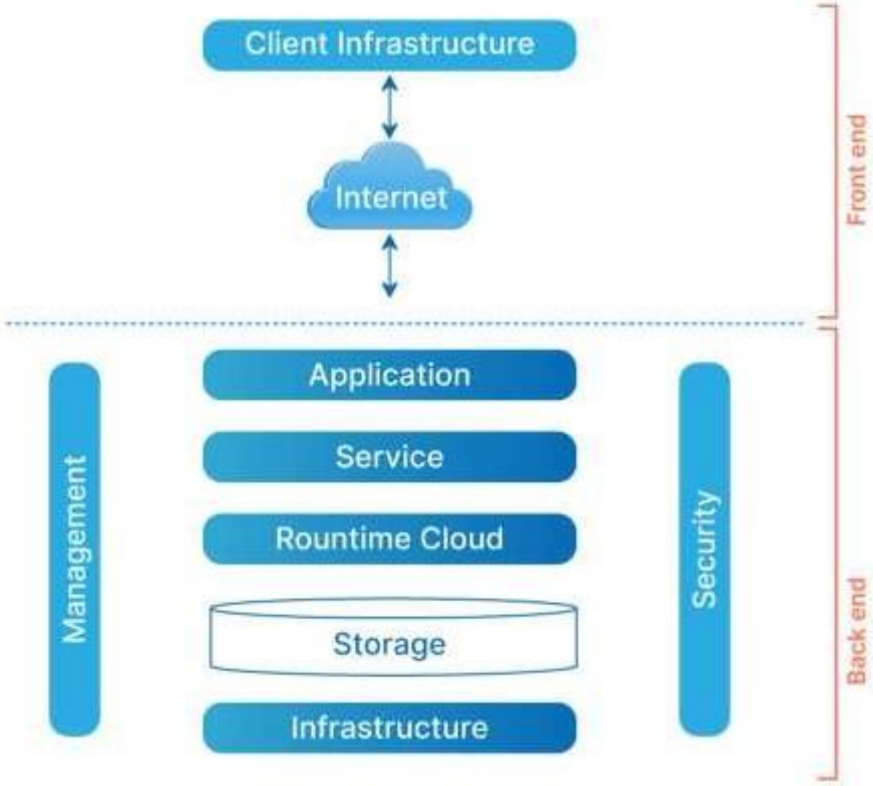
Infrastructure: Infrastructure is probably the most commonly known component of cloud architecture. In fact, you might have thought that cloud infrastructure *is* cloud architecture. However, cloud infrastructure comprises all the major hardware components that power cloud services, including the CPU, graphics processing unit (GPU), network devices, and other hardware components needed for systems to run smoothly. Infrastructure also refers to all the software needed to run and manage everything.

Cloud architecture, on the other hand, is the plan that dictates how cloud resources and infrastructure are organized.

Management: Cloud service models require that resources be managed in real time according to user requirements. It is essential to use management software, also known as middleware, to coordinate communication between the backend and frontend cloud architecture components and allocate resources for specific tasks. Beyond middleware, management software will also include capabilities for usage monitoring, data integration, application deployment, and disaster recovery.

Security: As more organizations continue to adopt cloud computing, implementing cloud security features and tools is critical to securing data, applications, and platforms. It's essential to plan and design data security and network security to provide visibility, prevent data loss and downtime, and ensure redundancy. This may include regular backups, debugging, and virtual firewalls.

ARCHITECTURE OF CLOUD COMPUTING



Types of Virtualization

1.Application Virtualization

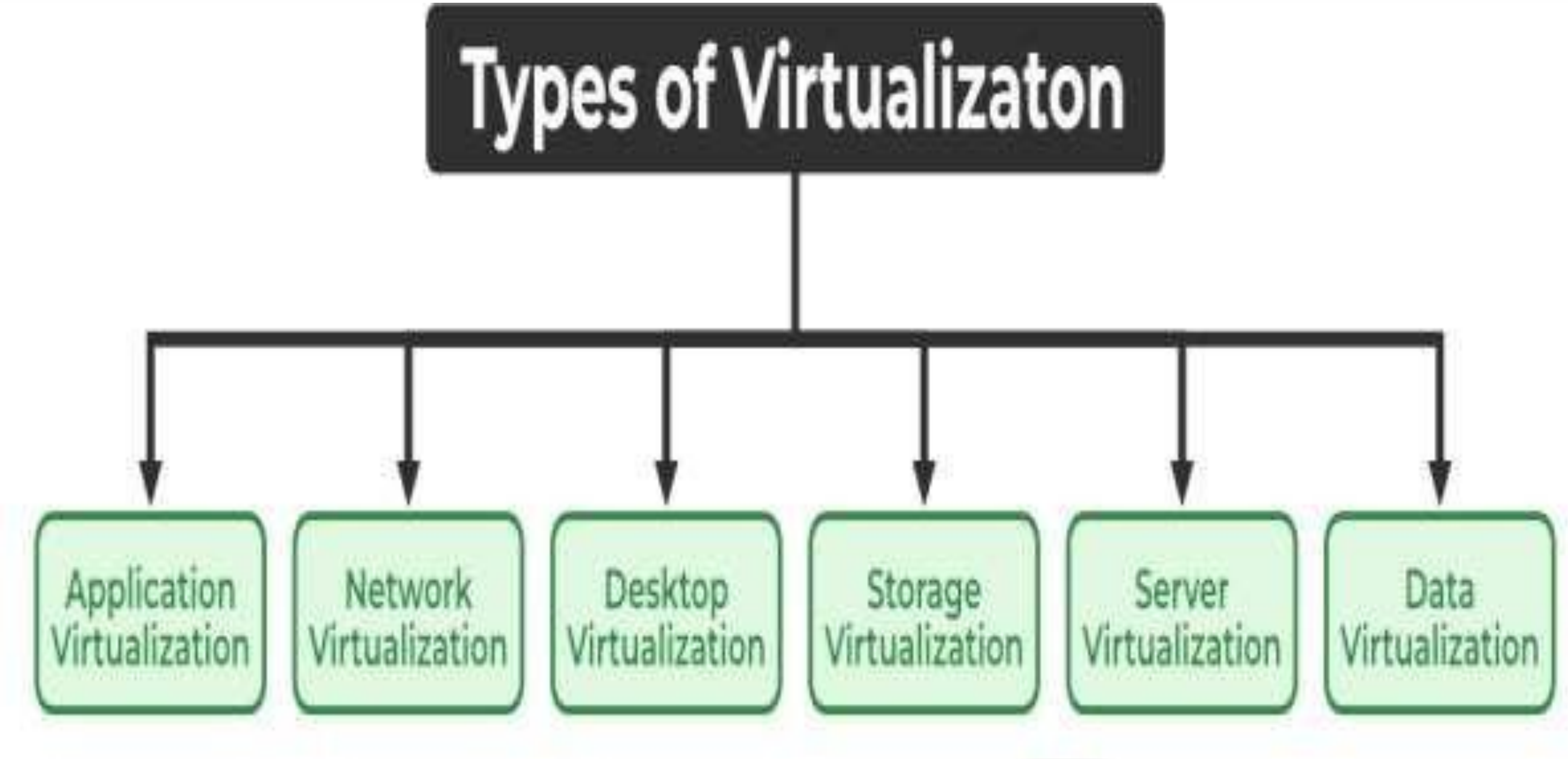
2.Network Virtualization

3.Desktop Virtualization

4.Storage Virtualization

5.Server Virtualization

6.Data virtualization


Types of Virtualizaton

Application Virtualization

Network Virtualization

Desktop Virtualization

Storage Virtualization

Server Virtualization

Data Virtualization

- **Server Virtualization**:
  - **Description**: Dividing a physical server into multiple virtual servers (VMs).
  - **Technologies**: Hypervisors (e.g., VMware ESXi, Microsoft Hyper-V, KVM).
  - **Real-World Application**: Data centers use server virtualization to run multiple applications on fewer physical servers, reducing costs and energy consumption.
- **Storage Virtualization**:
  - **Description**: Pooling physical storage from multiple devices into a single virtual storage device.
  - **Technologies**: Storage Area Networks (SAN), Network Attached Storage (NAS).
  - **Real-World Application**: Companies like Amazon use storage virtualization to provide scalable storage solutions for their cloud services.
  - **Network Virtualization**: Creating a virtualized network that can be managed and configured independently of the physical network.
    - **Technologies**: Software-Defined Networking (SDN), Virtual Private Networks (VPNs).
    - **Real-World Application**: Enterprises use network virtualization to create isolated environments for different departments while sharing the same physical infrastructure.

- **Desktop Virtualization**:
  - **Description**: Running desktop environments on a centralized server, allowing users to access their desktops remotely.
  - **Technologies**: Virtual Desktop Infrastructure (VDI), Remote Desktop Services (RDS).
  - **Real-World Application**: Organizations use desktop virtualization to enable remote work and improve security by centralizing data.
- **Benefits of Virtualization**
- **Resource Efficiency**: Maximizes the use of physical resources.
- **Cost Savings**: Reduces hardware costs and energy consumption.
- **Scalability**: Easily scale resources up or down based on demand.
- **Isolation**: Provides security and stability by isolating applications and workloads.
- **Disaster Recovery**: Simplifies backup and recovery processes.

Software-Defined Networking (SDN) is an approach to networking that uses software-based controllers or application programming interfaces (APIs) to communicate with underlying hardware infrastructure and direct traffic on a network.

This model differs from that of traditional networks, which use dedicated hardware devices (i.e., routers and switches) to control network traffic. SDN can create and control a virtual network – or control a traditional hardware – via software.

While network virtualization allows organizations to segment different virtual networks within a single physical network, or to connect devices on different physical networks to create a single virtual network, software-defined networking enables a new way of controlling the routing of data packets through a centralized server.

Why Software-Defined Networking is important?

SDN represents a substantial step forward from traditional networking, in that it enables the following:

- **Increased control with greater speed and flexibility:** Instead of manually programming multiple vendor-specific hardware devices, developers can control the flow of traffic over a network simply by programming an open standard software-based controller. Networking administrators also have more flexibility in choosing networking equipment, since they can choose a single protocol to communicate with any number of hardware devices through a central controller.
- **Customizable network infrastructure:** With a software-defined network, administrators can configure network services and allocate virtual resources to change the network infrastructure in real time through one centralized location. This allows network administrators to optimize the flow of data through the network and prioritize applications that require more availability.

- **Robust security:** A software-defined network delivers visibility into the entire network, providing a more holistic view of security threats. With the proliferation of smart devices that connect to the internet, SDN offers clear advantages over traditional networking. Operators can create separate zones for devices that require different levels of security, or immediately quarantine compromised devices so that they cannot infect the rest of the network.

  The key difference between SDN and traditional networking is infrastructure: SDN is software-based, while traditional networking is hardware-based. Because the control plane is software-based, SDN is much more flexible than traditional networking. It allows administrators to control the network, change configuration settings, provision resources, and increase network capacity — all from a centralized user interface, without the need for more hardware.

  There are also security differences between SDN and traditional networking. Thanks to greater visibility and the ability to define secure pathways, SDN offers better security in many ways. However, because software-defined networks use a centralized controller, securing the controller is crucial to maintaining a secure network.

How does Software-Defined Networking (SDN) work?

Here are the SDN basics: In SDN (like anything virtualized), the software is decoupled from the hardware. SDN moves the control plane that determines where to send traffic to software, and leaves the data plane that actually forwards the traffic in the hardware. There are three parts to a typical SDN architecture, which may be located in different physical locations:

Applications, which communicate resource requests or information about the network as a whole

Controllers, which use the information from applications to decide how to route a data packet

Networking devices, which receive information from the controller about where to move the data

Physical or virtual networking devices actually move the data through the network. In some cases, virtual switches, which may be embedded in either the software or the hardware, take over the responsibilities of physical switches and consolidate their functions into a single, intelligent switch. The switch checks the integrity of both the data packets and their virtual machine destinations and moves the packets along.

Benefits of Software-Defined Networking (SDN)

Many of today’s services and applications, especially when they involve the cloud, could not function without SDN. SDN allows data to move easily between distributed locations, which is critical for cloud applications.

Additionally, SDN supports moving workloads around a network quickly. For instance, dividing a virtual network into sections, using a technique called network functions virtualization (NFV), allows telecommunications providers to move customer services to less expensive servers or even to the customer’s own servers. Service providers can use a virtual network infrastructure to shift workloads from private to public cloud infrastructures as necessary, and to make new customer services available instantly. SDN also makes it easier for any network to flex and scale as network administrators add or remove virtual machines, whether those machines are on-premises or in the cloud.

Finally, because of the speed and flexibility offered by SDN, it is able to support emerging trends and technologies such as edge computing and the Internet of Things, which require transferring data quickly and easily between remote sites.

•Computing Virtualization

Refers to the creation of virtual machines (VMs) that emulate physical computers.

Components:

Hypervisor: Software layer enabling virtualization.

Type 1 (Bare Metal): Runs directly on hardware (e.g., VMware ESXi, Microsoft Hyper-V).

Type 2 (Hosted): Runs on an operating system (e.g., Oracle VirtualBox, VMware Workstation)

Guest OS: Operating system installed on a VM.

Virtual CPU (vCPU) and memory allocated to VMs.

Running multiple operating systems on one physical machine.

- Development and testing environments.
- High availability and disaster recovery.

Image source: Roger Pressman, Software engineering: A practitioner's Approach, McGraw-Hill International Editions

Pooling and abstracting physical storage devices into a single, unified storage resource.

Types:

Block Storage Virtualization: Logical storage blocks are presented to hosts (e.g., SAN - Storage Area Network).

File Storage Virtualization: Files are abstracted and managed via a network (e.g., NAS - Network Attached Storage).

Components:

Storage Controllers: Manage physical storage.

Virtual Storage Layer: Interfaces with applications. **Storage Pools**: Grouping of virtualized storage resources. **Use Cases**:

Simplified storage management. Dynamic allocation of storage capacity. Disaster recovery and backup.

Cloud Services: IaaS – Infrastructure as a Service

Cloud provider gives **infrastructure resources**:

- Virtual machines
- Storage
- Networks
- Load balancers

You manage: OS, apps, data, runtime

Provider manages: Hardware, networking, virtualization

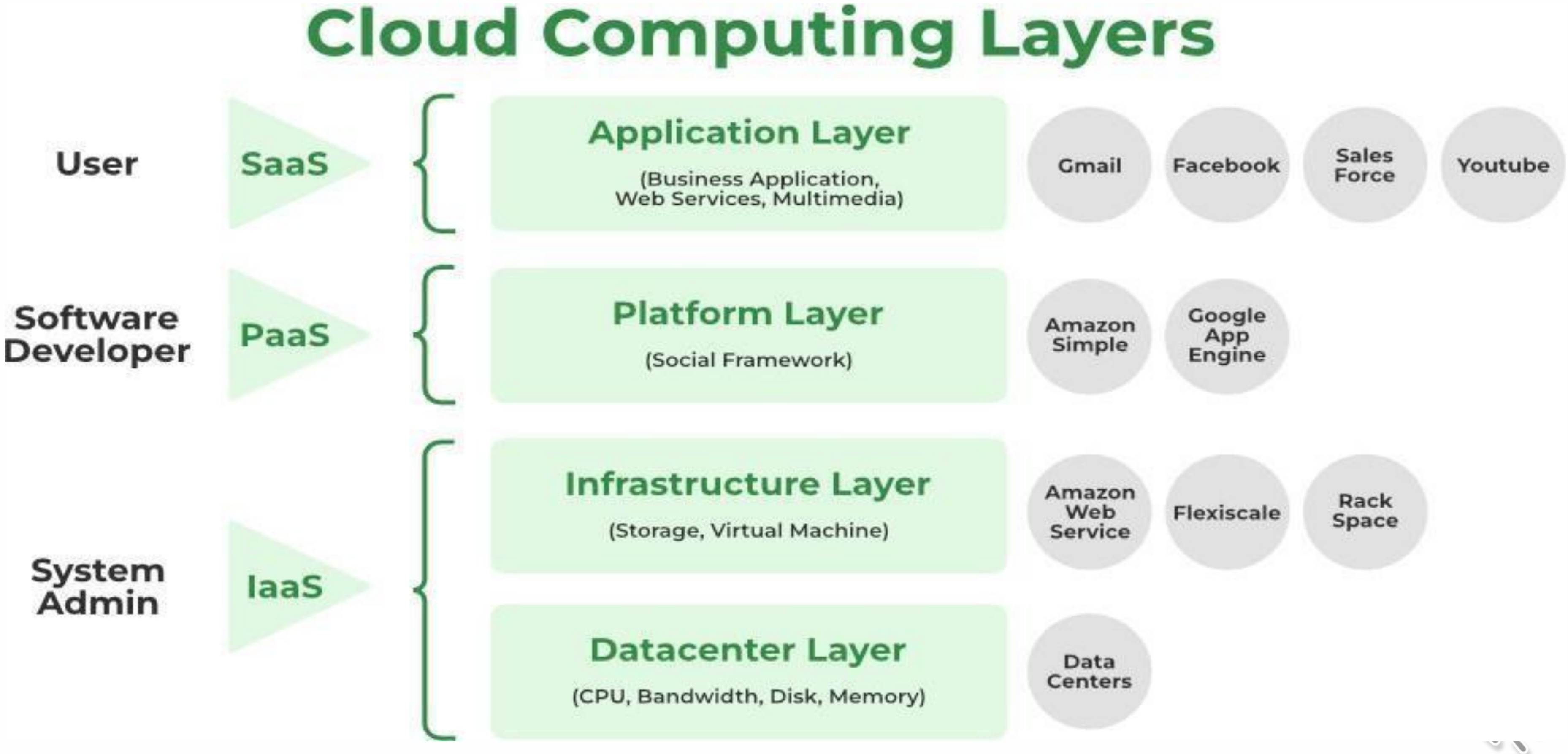
Examples

- AWS EC2
- Google Compute Engine
- Microsoft Azure VM

Use Cases

- Hosting websites
- Running custom applications
- Creating virtual networks

Layers of Clouds



Cloud Computing Layers

User — SaaS — **Application Layer** (Business Application, Web Services, Multimedia) — Gmail, Facebook, Sales Force, Youtube

Software Developer — PaaS — **Platform Layer** (Social Framework) — Amazon Simple, Google App Engine

System Admin — IaaS — **Infrastructure Layer** (Storage, Virtual Machine) — Amazon Web Service, Flexiscale, Rack Space

Datacenter Layer (CPU, Bandwidth, Disk, Memory) — Data Centers

Cloud Services: PaaS – Platform as a Service

Cloud provider gives a **ready-to-use platform** for application development.

Includes:

- OS
- Runtime environment
- Databases
- Development tools

You manage: Only code & application

Provider manages: Servers, OS, storage, network, platform

Examples

- Google App Engine
- Microsoft Azure App Services
- Heroku

Use Cases

- App development
- API hosting
- Automated deployment pipelines

Cloud Services: SaaS – Software as a Service

Cloud provider delivers **complete software applications** over the internet.
You manage: Only usage and data
Provider manages: Everything (infrastructure + platform + software)

Examples

Gmail
Google Docs
Salesforce
Microsoft 365

Use Cases

Email services
Document collaboration
CRM systems

Feature	IaaS	PaaS	SaaS
What you get	Infrastructure	Development Platform	Ready-made Software
You manage	OS, Apps, Data	Apps & Data	Only use the app
Examples	AWS EC2	Google App Engine	Gmail, Office 365
Skill needed	High	Medium	Very low

IAAS

Infrastructure as a Service (IaaS)

Definition:

IaaS provides virtualized computing resources like servers, storage, and networks over the internet.

It acts as the foundational layer where users control and manage the infrastructure themselves.

Scalability: Resources can be scaled up or down based on demand.

Cost Efficiency: Pay-as-you-go model eliminates upfront hardware costs.

Flexibility: Complete control over the operating system, applications, and middleware.

Where It Is Used:

Disaster Recovery: Storing and retrieving critical data during emergencies.

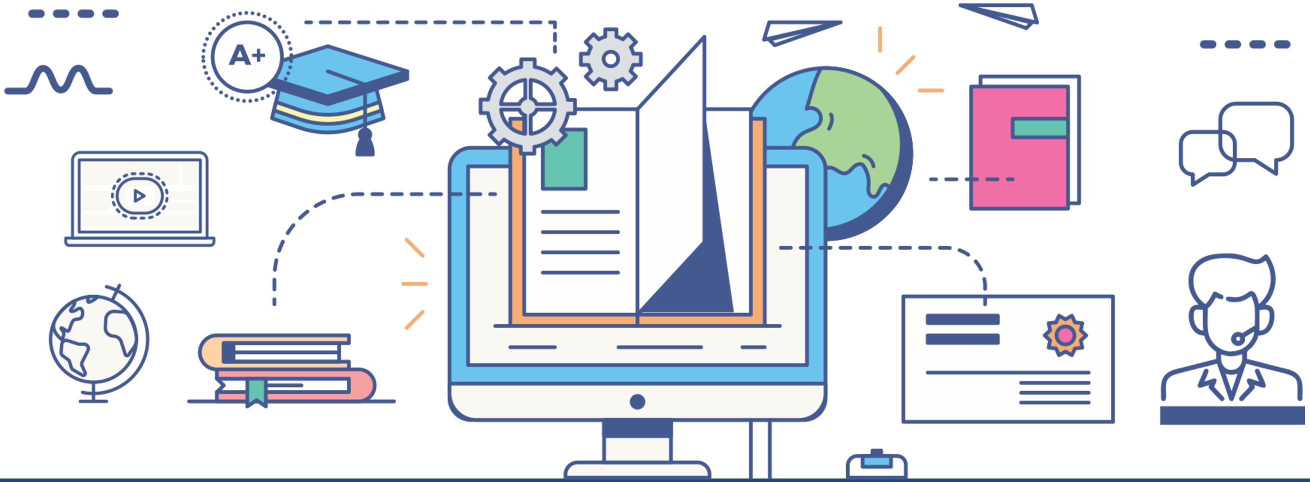
Big Data Analysis: Companies like Netflix use IaaS to process and analyze massive datasets.

Web Hosting: Hosting applications and websites with high traffic demands.

Examples:

Amazon EC2, Microsoft Azure Virtual Machines, Google Compute Engine.

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in