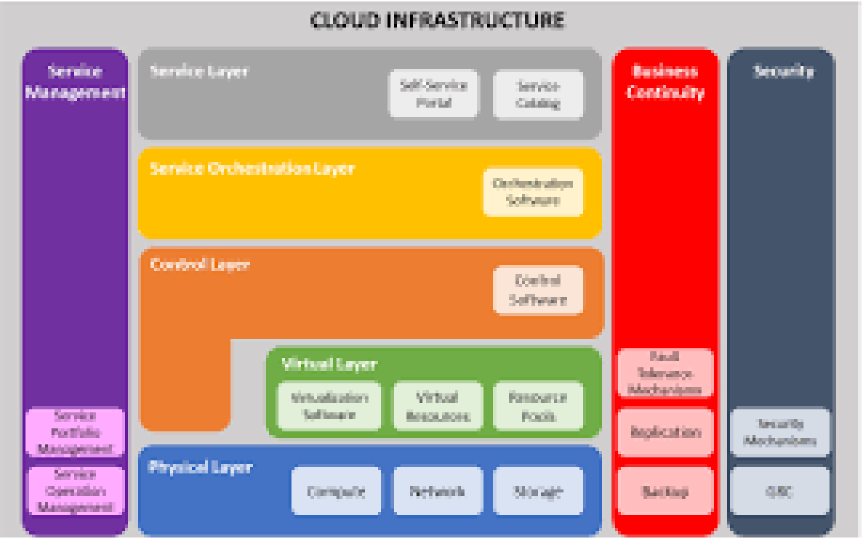
Unit 2: Cloud Computing Architecture

Owner	Saugat Tiwari
Tags	BCA Cloud Computing Notes
Creation Date	@December 4, 2024

2.1 Cloud reference model

The cloud computing reference model is an abstract framework that defines and standardizes cloud computing functions by organizing them into distinct layers and cross-layer operations.

The cloud computing reference model organizes functions and activities into five logical layers and three cross-layer functions.



The five layers are: `physical layer`, `virtual layer`, `control layer`, `service orchestration layer`, and `service layer`.

Each layer encompasses specific entities within the cloud computing environment, including compute systems, network devices, storage devices, virtualization software, security mechanisms, control software, orchestration software, and management software. These entities work together in defined relationships.

The three cross-layer functions are `business continuity`, `security`, and `service management`.

Business continuity and security functions manage the activities and processes needed to deliver reliable and secure cloud services to consumers.

Service management coordinates the activities and processes that enable cloud infrastructure administration, balancing provider business requirements with consumer expectations.

Cloud computing layers

1. **Physical Layer:**
    - The foundation layer of cloud infrastructure.
    - This layer contains compute systems, network devices, storage devices, operating environments, protocols, tools, and processes.
    - Its main function is executing requests from the virtualization and control layers.
2. **Virtual Layer:**
    - Built on top of the physical layer.
    - Key components include virtualization software, resource pools, and virtual resources.
    - Primary functions: Creates virtual resources by abstracting physical resources (enabling multi-tenant environments) and executes control layer requests.
3. **Control Layer:**
    - Can be deployed on either the virtual or physical layer.
    - Uses control software as its main operating entity.
    - Primary functions include:
        - Configuring resources and resource pools
        - Managing resource provisioning
        - Processing service layer requests
        - Supporting and exposing resources to the service layer
        - Working with virtualization software to enable resource pooling, virtual resource creation, dynamic allocation, and optimal resource

utilization

4. **Service Orchestration Layer:**
    - The main entity operating at this layer is orchestration software.
    - Its primary functions include providing workflows for automated tasks and interacting with various entities to handle provisioning tasks.
5. **Service Layer:**
    - This is where consumers interact with and consume cloud resources.
    - Key entities include the service catalog and self-service portal.
    - The service catalog stores and presents information about available cloud services to consumers.
    - The self-service portal enables consumers to access and manage their cloud services.

Cross-layer function

1. Business Continuity

Implements proactive and reactive measures to minimize service downtime.
Ensures service availability according to Service Level Agreements (SLAs).
Maintains uninterrupted services across all layers.

2. Security

Implements both administrative mechanisms (security policies, personnel policies, and standard operating procedures) and technical mechanisms (firewalls, intrusion detection systems, and antivirus software).
Deploys security measures to meet governance, risk, and compliance requirements.
Ensures secure service delivery across all layers.

3. Service Management

Oversees service portfolio management and service operation management activities.

Service Portfolio Management:

- Develops service roadmaps, features, and service levels
- Evaluates and prioritizes service portfolio investments
- Manages budgeting and pricing
- Handles customer operations including orders, billing, and payment collection

Service Operation Management:

Manages infrastructure configuration and resource provisioning

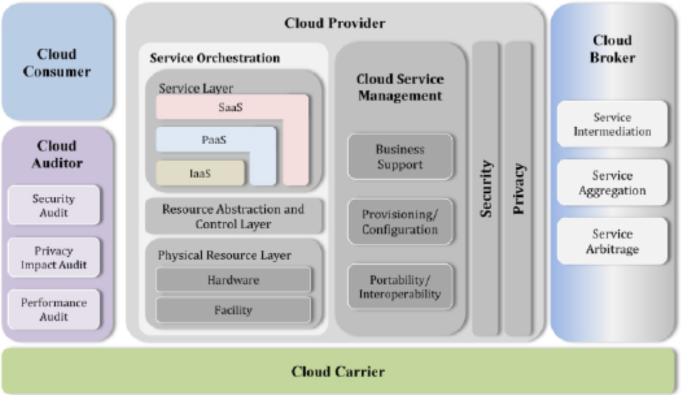
Resolves technical problems

Oversees capacity and availability

Ensures compliance

Monitors cloud services and components

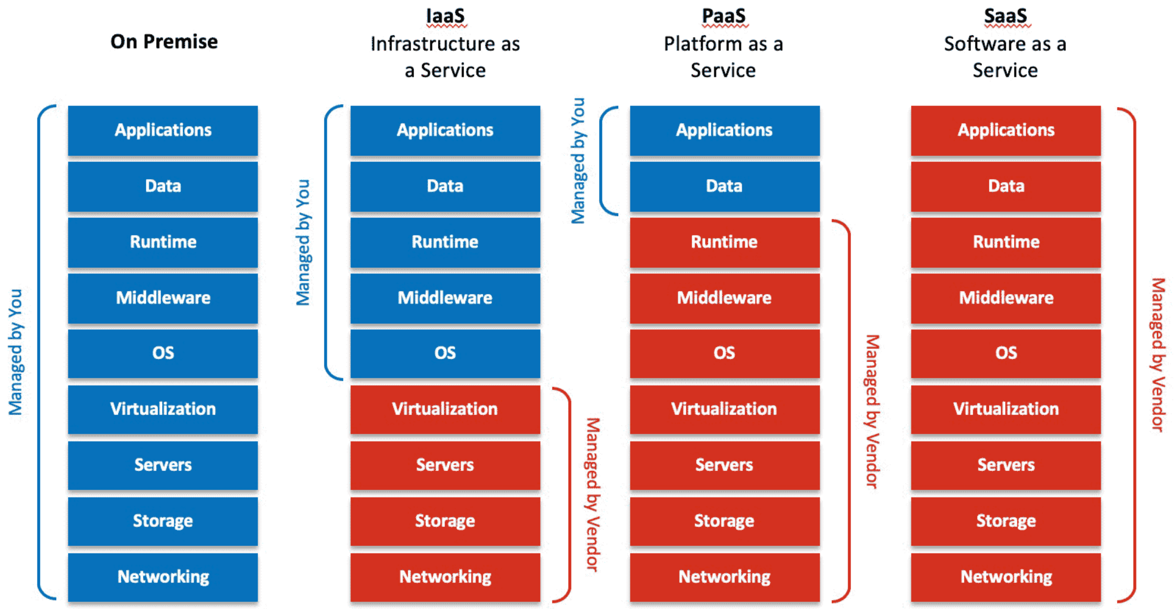
Cloud computing conceptual reference model identifies the major actors, their activities and functions in cloud computing.



The NIST cloud computing reference architecture defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor, and cloud broker. Each actor is an entity (a person or organization) that participates in transactions or processes and performs tasks in cloud computing. The table below lists these actors and their roles in the NIST cloud computing reference architecture.

SNO	Actor	Definition
1	Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.
2	Cloud Provider	A person, organization, or entity responsible for making a service available to interested parties.
3	Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
4	Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
5	Cloud Carrier	An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

Comparison of Cloud Computing Service Models vs On-Premises



The image shows a comparative visualization of different cloud service models: On-Premise, `Infrastructure as a Service (IaaS)`, `Platform as a Service (PaaS)`, and `Software as a Service (SaaS)`. Using blue boxes for user-managed components and red boxes for vendor-managed components, it illustrates how management responsibilities shift from the user to the service provider across these models.

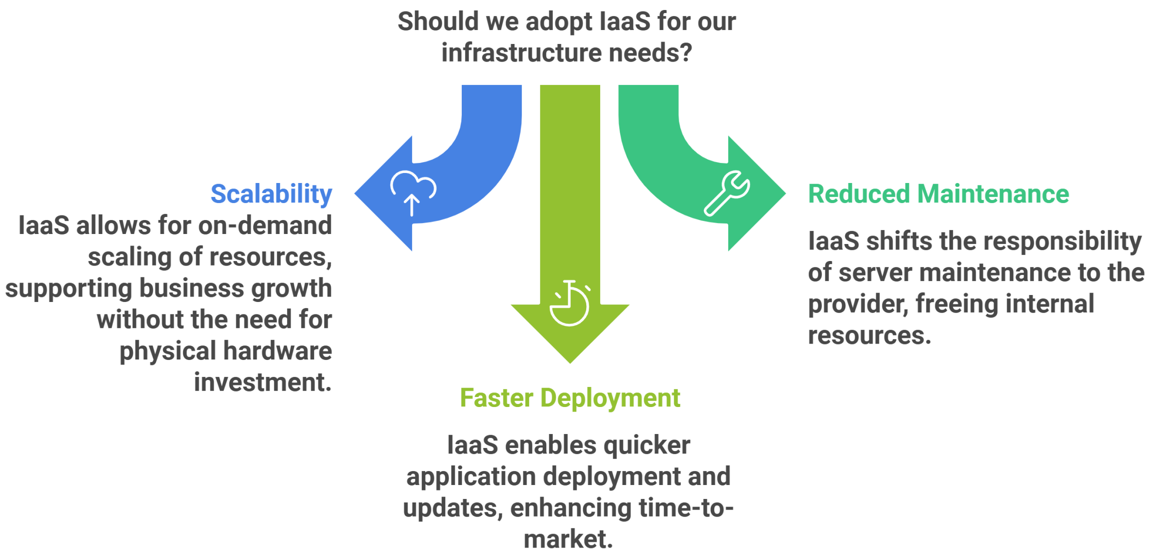
The stack consists of layers including `Networking`, `Storage`, `Servers`, `Virtualization`, `OS`, `Middleware`, `Runtime`, `Data`, and `Applications`.

In On-Premise, users manage everything; in IaaS, providers manage only the infrastructure layers; in PaaS, providers manage up to the runtime environment; and in SaaS, providers manage the entire stack, leaving users to only interact with the application level.

2.1.1 Infrastructure as service

In computing, infrastructure comprises the computers and servers that run code and store data, along with the wires and appliances that connect these machines. This includes hardware like servers, hard drives, and routers. Historically, before cloud computing emerged, businesses maintained their own infrastructure and ran all applications on-premises.

`Infrastructure-as-a-Service (IaaS)` is a model where cloud computing vendors host infrastructure for their customers in data centers: `the cloud`. Customers access this infrastructure through the Internet, using it to build and host web applications, store data, and run business logic—essentially performing any task possible with traditional on-premises infrastructure, but with greater flexibility.


Should we adopt IaaS for our infrastructure needs?
Scalability
IaaS allows for on-demand scaling of resources, supporting business growth without the need for physical hardware investment.
Faster Deployment
IaaS enables quicker application deployment and updates, enhancing time-to-market.
Reduced Maintenance
IaaS shifts the responsibility of server maintenance to the provider, freeing internal resources.

ADVANTAGES:

1. **Scalability:** IaaS provides a foundation for easier business expansion. Rather than purchasing, installing, and maintaining new servers for growth,

businesses can instantly add servers through their IaaS provider. This on-demand scalability is a key advantage across all cloud service models.

2. **Fewer resources dedicated to server maintenance:** IaaS transfers server purchasing, maintenance, and updating responsibilities to the provider. This approach typically costs less and demands less time and effort from internal teams compared to self-hosted infrastructure.
3. **Faster time to market:** With IaaS, companies can deploy and update applications more rapidly since providers can instantly scale infrastructure to meet demand.

2.1.2 Platform as service

Platform-as-a-Service (PaaS) is a cloud computing model where developers rent everything needed to build applications, including development tools, infrastructure, and operating systems from a cloud provider. As one of the three main cloud service models, PaaS significantly simplifies web application development by handling all backend management behind the scenes. While PaaS shares some features with serverless computing, they remain distinctly different approaches.

PaaS platforms are accessible through any internet connection, allowing developers to build applications entirely through a web browser. This accessibility enables worldwide developer collaboration since the development environment isn't tied to local machines. While developers trade some control over their development environment for this flexibility, they benefit from significantly reduced operational overhead.

PaaS vendors typically provide these core services:

- Development tools
- Middleware
- Operating systems
- Database management
- Infrastructure

While vendors may offer additional services, these form the essential foundation of PaaS offerings.

1. **Development tools**
   PaaS vendors provide essential software development tools including source code editors, debuggers, compilers, and other utilities. These tools

are often bundled into a comprehensive framework. While specific offerings vary by vendor, each PaaS platform includes all core tools developers need to build applications.

2. **Middleware**
   PaaS platforms include pre-built middleware, eliminating the need for custom development. This software layer connects user-facing applications with the operating system—managing tasks like keyboard and mouse input processing. While middleware is essential for application functionality, it operates invisibly to end users.
3. **Operating systems**
   PaaS vendors handle both the provision and maintenance of operating systems where developers work and applications run.
4. **Databases**
   PaaS providers manage database administration and maintenance, typically including a database management system for developers.
5. **Infrastructure**
   As the layer above IaaS in the cloud computing stack, PaaS incorporates all IaaS components. PaaS providers either maintain their own servers, storage, and data centers, or source these from an IaaS provider.

Why do developers use PaaS?

1. **Faster time to market**

   PaaS enables rapid application development by eliminating the need for developers to manage platforms and backend infrastructure. Developers can focus solely on writing and testing code while the vendor manages everything else.

2. **One environment from start to finish**

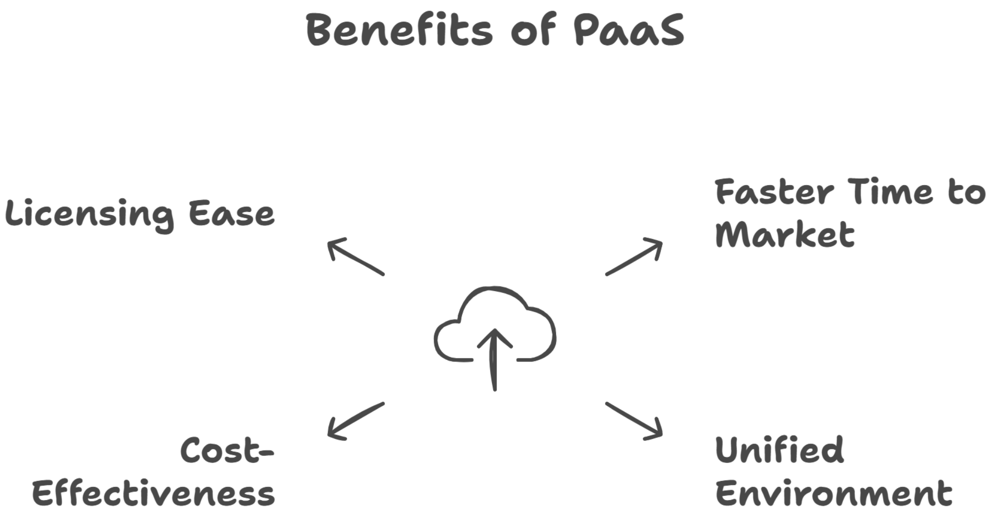
   PaaS provides a unified environment for building, testing, debugging, deploying, hosting, and updating applications. This integrated approach ensures applications will function properly in production and streamlines the development lifecycle.

3. **Price**

   PaaS often proves more cost-effective than IaaS since customers don't need to manage virtual machines, reducing overhead costs. Many providers offer pay-as-you-go pricing, charging only for actual computing resources used—though some platforms opt for monthly flat fees.

4. **Ease of licensing**

   PaaS providers manage all licensing requirements for operating systems, development tools, and other platform components.



What are the potential drawbacks of using PaaS?

1. **Vendor lock-in**

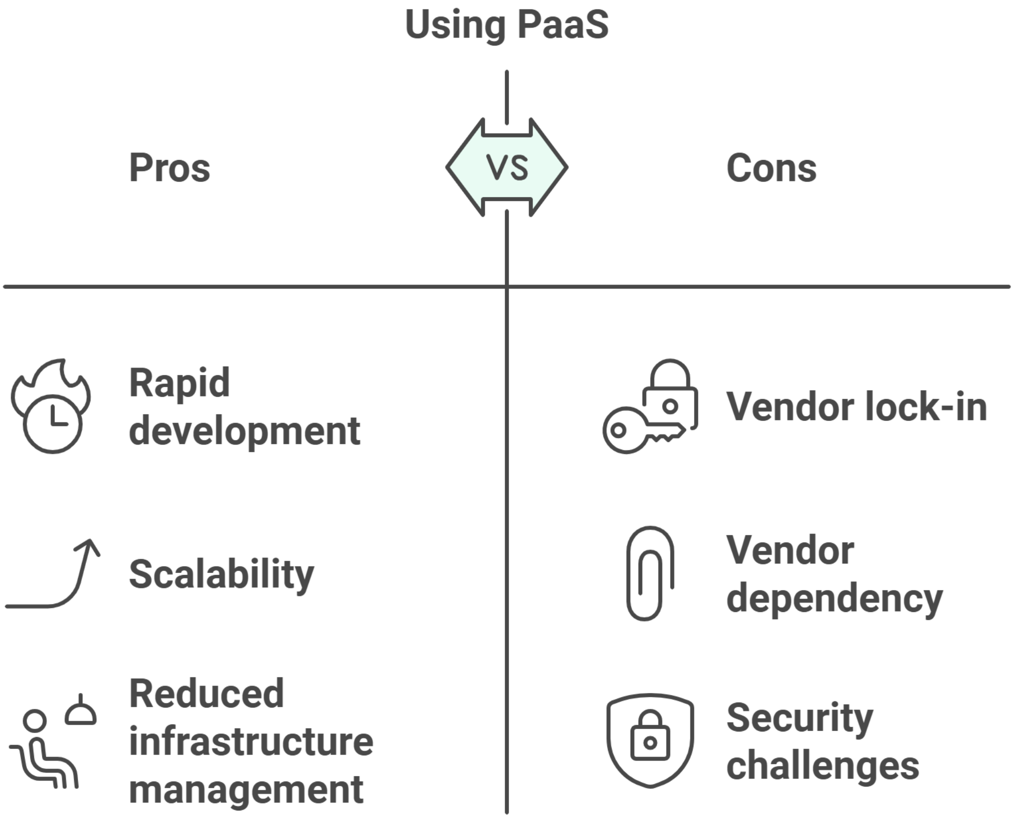
   Switching PaaS providers can be challenging because applications are built using vendor-specific tools and platforms. Each vendor has unique architecture requirements and may support different programming languages, libraries, APIs, and operating systems. When switching vendors, developers often need to significantly modify or completely rebuild their applications.

2. **Vendor dependency**

   The complexity of changing PaaS vendors can create strong dependency on the current provider. Even minor changes in the vendor's infrastructure or processes can significantly impact applications optimized for their platform. Furthermore, changes in the vendor's pricing model can unexpectedly increase operational costs.

3. **Security and compliance challenges**

   PaaS vendors host both application code and data, sometimes using third-party IaaS providers for storage. Despite major PaaS vendors maintaining robust security measures, this architecture makes it challenging to thoroughly evaluate and test security protocols. Organizations subject to strict data security regulations face additional complexity in verifying compliance across multiple external vendors, potentially delaying deployment.

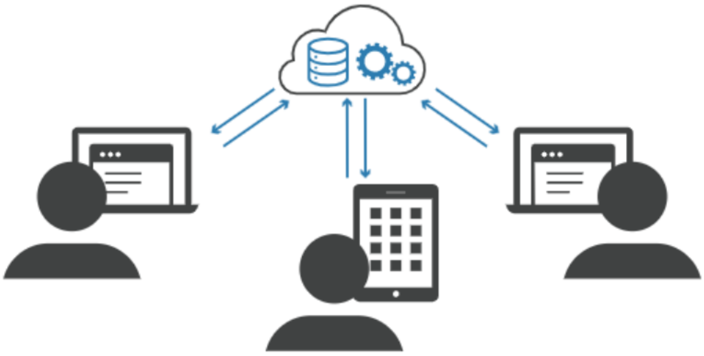


2.1.3 Software as a service

Software-as-a-Service (SaaS) is a cloud-based method of delivering software to users through a subscription model rather than traditional one-time purchases and installations. Users can access SaaS applications from any compatible device with an internet connection, while the software itself runs on remote cloud servers.

Users typically access SaaS applications through web browsers or dedicated apps. Common examples include web-based email services like Gmail and Office 365.

The relationship between SaaS and traditional software installation is analogous to streaming a TV show versus buying DVD box sets—one offers flexible, on-demand access while the other requires local storage and management.



Users can access SaaS applications on any device

The SaaS model offers several advantages and disadvantages, though its benefits typically outweigh the drawbacks for modern businesses and users. Here are the key advantages of using SaaS applications:

Advantages:

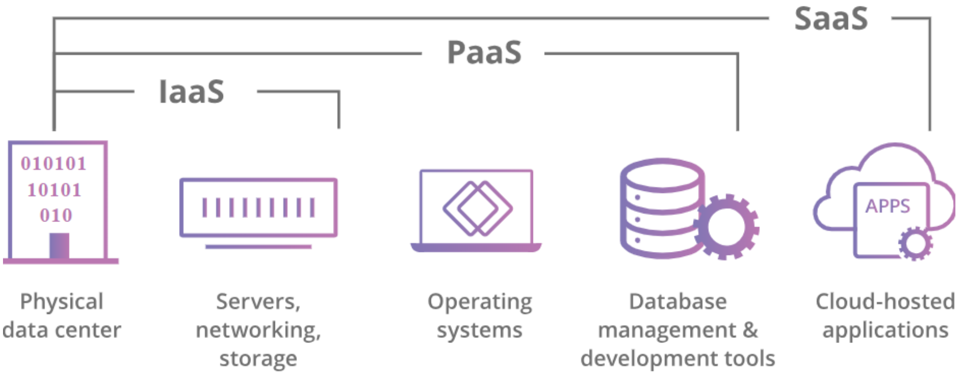
- **Access from anywhere, on any device:** Users can log into SaaS applications from any device and location, providing exceptional flexibility. This enables businesses to support worldwide employees while ensuring users can access their files anywhere. Furthermore, since users often work across multiple devices and upgrade them frequently, they won't need to reinstall applications or purchase new licenses with each device change.
- **No need for updates or installations:** The SaaS provider handles all application updates and patches automatically.
- **Scalability:** The SaaS provider manages all scaling requirements, including expanding database storage or increasing compute power as usage grows.
- **Cost savings:** SaaS reduces internal IT costs and overhead since the provider maintains all servers and infrastructure. Businesses only pay the application subscription cost.

Disadvantages:

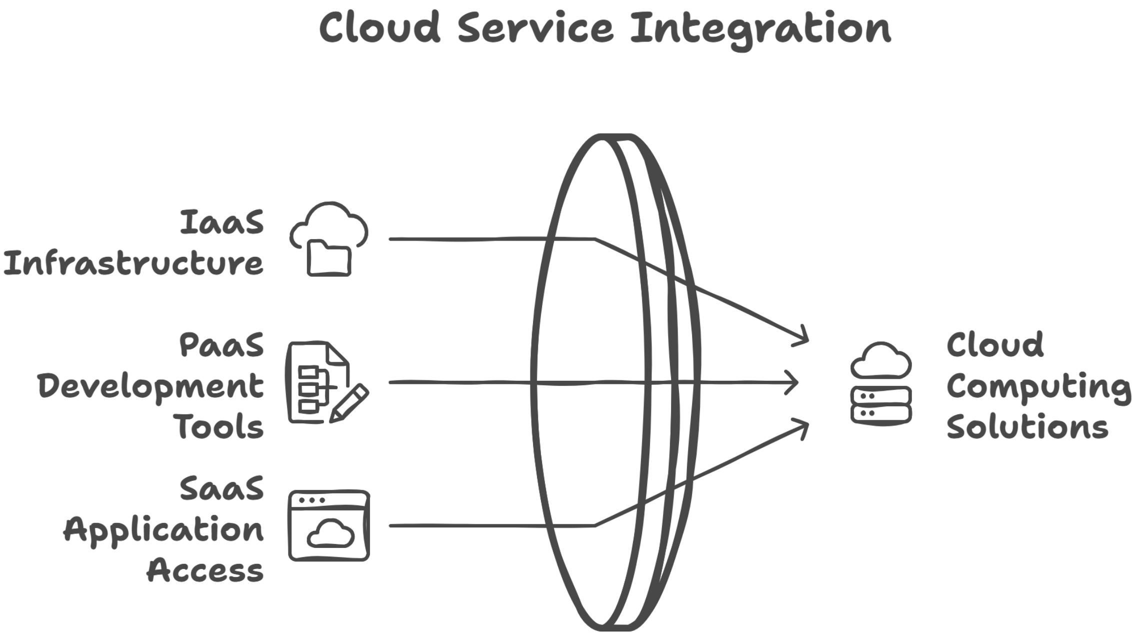
- Enhanced Access Control Requirements
  SaaS applications' broad accessibility means user identity verification and access control are critical. Since organizational assets reside outside the internal network, access depends solely on login credentials. This makes robust identity verification essential.
- Vendor Lock-in
  Organizations often become dependent on their SaaS provider. Migrating to

a different application can be costly and time-consuming, especially when an organization's entire database is stored within the current application.

- Security and Compliance Challenges (particularly for enterprises)
  SaaS shifts security responsibility from internal IT teams to external providers. While this benefits smaller businesses: as major cloud providers typically have robust security resources. It can pose challenges for large enterprises with strict security and regulatory requirements. Organizations may lack the ability to independently verify application security through methods like penetration testing, forcing them to rely on the provider's security assurances.


SaaS
PaaS
IaaS
010101
10101
010
Physical data center
Servers, networking, storage
Operating systems
APPS
Database management & development tools
Cloud-hosted applications

IaaS vs. PaaS vs. SaaS



IaaS provides infrastructure hosted in the cloud, including virtual servers, storage, security, and managed data center resources.

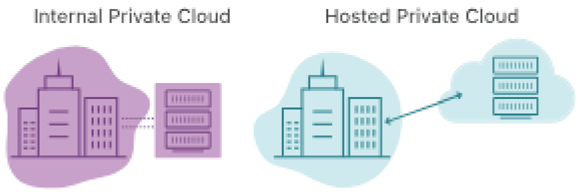
Platform-as-a-Service (PaaS) builds on IaaS to offer developers a complete platform for application development. It bundles essential components like development tools, middleware, operating systems, and database management. PaaS providers either maintain their own infrastructure or leverage IaaS services.

Software-as-a-Service (SaaS) delivers complete applications hosted and managed in the cloud. Instead of traditional software installation, users access these applications through internet-based subscriptions.

2.2 Cloud deployment models

2.2.1 Private Cloud

A private cloud is a cloud service exclusively dedicated to a single organization. It allows organizations to leverage cloud computing benefits while maintaining complete resource isolation. Private clouds can be hosted either on-premises within an organization or managed remotely by a third party via the Internet; but unlike public clouds, the resources remain exclusively dedicated to one organization.



What is a hosted private cloud?
A hosted private cloud is managed and hosted remotely by a third party, rather than being located on an organization's premises.

For example, if a large company in Chicago wants to run a private cloud, they have two options: they can set up an internal private cloud in their office building, or they can have a third-party provider host it. This provider could be located anywhere—in another part of Chicago, elsewhere in Illinois, or even in a different state. The key distinction is that the cloud servers are not physically located at the organization's facility, but are instead maintained off-site by the external provider.

What is an internal private cloud?

- An internal private cloud is hosted and managed on an organization's own premises.
- Unlike a hosted private cloud, the organization handles all aspects of operation themselves—purchasing and maintaining servers, ensuring system uptime, and managing the software infrastructure.

How is an internal private cloud different from a traditional on-premises data center?

- While an internal private cloud shares some similarities with a traditional data center—being hosted, managed, and accessed by a single company on its premises—there are key differences.
- The main distinction lies in architecture: internalh private clouds use cloud technology and virtual machines to optimize hardware utilization.
- This cloud-based approach makes private clouds significantly more efficient, powerful, and scalable than traditional data centers.
- To use an analogy: if a private cloud is like owning a modern, efficient laundromat, a traditional data center is like having a single washer and dryer at home.

Internal private clouds offer several key advantages over traditional data centers beyond virtualization:

• **Scalability:** Resources expand automatically as needed without manual IT configuration

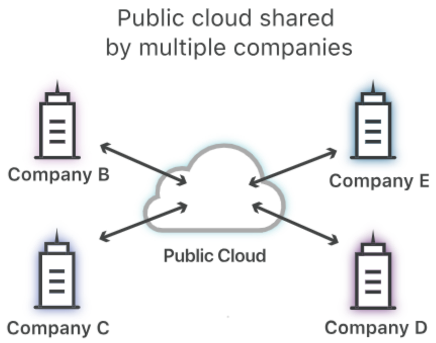
• **Self-service:** Users can access resources independently without IT support intervention

• **Broad access:** Teams across the organization enjoy seamless access to cloud resources

• **Measurability:** Detailed usage metrics for storage, bandwidth, and active users enable precise resource allocation

2.2.2 Public clouds

A public cloud is a cloud service that providers offer to multiple customers simultaneously. The term distinguishes between the traditional cloud computing model—where services are accessed over the Internet—and private clouds. Public clouds encompass all major service types: SaaS, PaaS, and IaaS.



Like all cloud services, public clouds operate on remote servers managed by the provider, with customers accessing these services through the Internet.

📓 ***Notes:***

What is multitenancy?

In a public cloud, multiple organizations often share the same physical server simultaneously—this is called multitenancy. It means that one server may store data from different companies or run processes from separate applications at the same time. For example, when two companies use the same cloud provider, their data and applications might coexist on a single server while remaining securely separated.

Advantages:

- **Cost savings:** Public cloud adoption significantly reduces IT operations costs. Organizations outsource infrastructure management to third-party providers who operate more efficiently through economies of scale. Public clouds are typically more cost-effective than private clouds since providers maximize hardware utilization by serving multiple customers simultaneously.
- **Less server management:** Organizations using public clouds eliminate the burden of server maintenance that comes with traditional on-premises data centers or internal private clouds. This frees up internal IT teams to focus on core business objectives.

- **Security:** Small and medium-sized businesses often lack resources for comprehensive security infrastructure. Public cloud providers offer robust security measures and expertise that would be otherwise unattainable for these organizations.

Disadvantages:

- **Security and compliance concerns:** Multitenancy poses challenges for businesses with strict regulatory requirements. While the risk of data leakage is minimal—as cloud providers maintain rigorous security standards—some specialized businesses may find even this small risk unacceptable. Additionally, organizations often struggle to maintain consistent security policies across their internal resources and public cloud infrastructure, particularly during migration.
- **Vendor lock-in:** This represents an ongoing challenge with cloud technology. While organizations benefit from cost savings and flexibility, they often become dependent on their cloud vendor's specific services including virtual machines, storage, applications, and technologies for critical business operations.

2.2.3 Hybrid clouds

A **hybrid cloud** combines two or more types of cloud environments. It typically includes both public and private clouds, and may also involve on-premises legacy systems. To be truly hybrid, these environments must be connected and work together as a unified infrastructure. Most hybrid cloud setups include at least one public cloud.

Hybrid clouds serve multiple purposes. They allow organizations to:

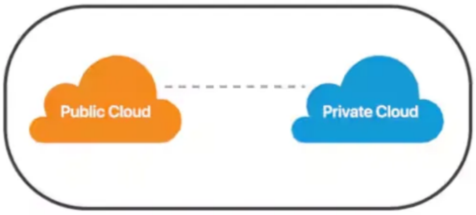
- Distribute workloads between private and public clouds.
- Use public clouds as backups for private infrastructure.
- Handle traffic spikes using public clouds while keeping core operations in a private cloud.

What types of environments are in hybrid clouds?

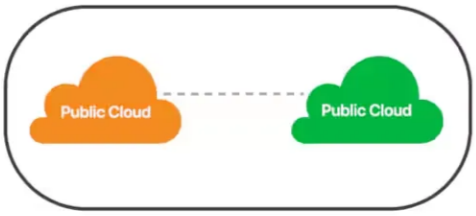
Hybrid clouds can mix any of the following environments:

- **Public Cloud:** A cloud service operated by an external vendor, shared by multiple organizations through virtual machines on the same physical servers.

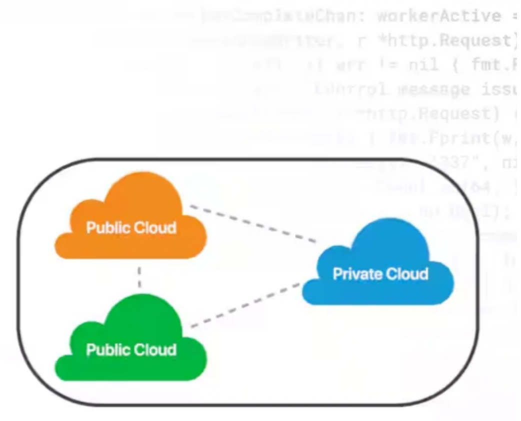
- **On-Premises Private Cloud:** A private data center dedicated to one organization, where all maintenance and security are handled internally.
- **Hosted Private Cloud:** Similar to an on-premises private cloud, but hosted by a third-party provider in remote data centers, accessed via the internet.
- **On-Premises (Legacy):** Traditional IT infrastructure where an organization owns the hardware and software, installed on-site, without cloud technology.



Hybrid cloud



Multicloud



Multi/hybrid cloud

A **multi-cloud** uses multiple public clouds, whereas a **hybrid cloud** combines a public cloud with another environment type. Hybrid clouds mix different cloud types (like apples and oranges), while multi-cloud setups involve multiple public clouds (like different varieties of apples).

A multi-cloud setup can be a hybrid cloud if it uses both public and other types of environments. It's similar to how a rectangle can be a square, but not all rectangles are squares. Likewise, not all multi-clouds are hybrid clouds.

For a hybrid cloud to function properly, the different clouds must be connected, using tools like:

- **APIs (Application Programming Interfaces)**
- **VPNs (Virtual Private Networks)**
- **WANs (Wide Area Networks)**

Without these connections, you are just running separate clouds in parallel, not a hybrid cloud.

Advantages of Hybrid Cloud Architecture

- **Flexibility**: Hybrid clouds make it easier to move between different cloud deployment styles (e.g., switching to a public cloud).
- **Variety of Technology:** Public clouds can support technologies (e.g., big data) that are difficult to run in a private cloud.
- **Backup & Redundancy:** If one cloud fails, you can rely on another cloud to avoid downtime.
- **Handling Demand Spikes:** Hybrid clouds can handle spikes in demand (e.g., during sales events) by "bursting" workloads into a public cloud for additional power.
- **Cost Savings**: Moving some operations to a public cloud can reduce the need for expensive on-premises infrastructure.
- **Sensitive Data Security:** Sensitive data (e.g., credit card or health information) can stay on a private cloud, while other apps run in the public cloud.

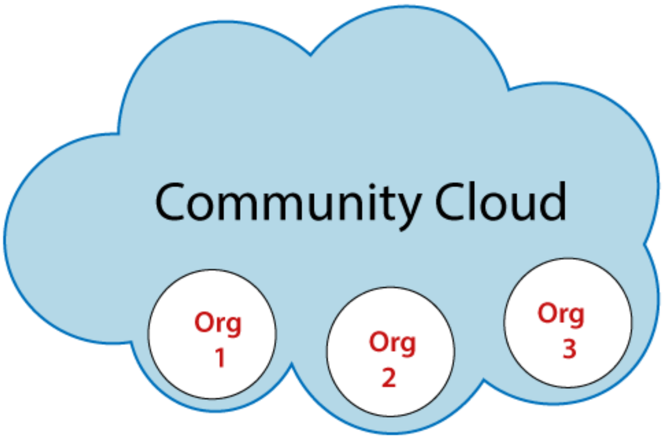
Drawbacks of Hybrid Cloud Architecture

- **Increased Attack Surface:** More complex networks can lead to more security vulnerabilities, as not all clouds may have the same security standards.
- **Complex Integrations:** Connecting and managing different types of clouds (e.g., setting up VPNs) is more complicated than using just one cloud.
- **Security Challenges:** Securing data across multiple clouds can be difficult. Different security tools may be needed for private and public clouds, and managing access control across them can be tricky.

2.2.4 Community cloud

A **community cloud** is a shared cloud infrastructure used by several organizations with similar interests, such as security, compliance, or jurisdiction. This cloud can be managed internally or by a third-party and hosted either on-site or remotely. The costs are shared among the organizations, which makes it more affordable than a private cloud, but not as cost-efficient as a public cloud.

Community clouds are designed for organizations with common needs (e.g., security, policies, or application demands). These organizations can share infrastructure and resources, making it a collaborative effort.



Key Scenarios Where Community Cloud is Beneficial:

1. **Government Agencies:** Multiple government departments can share cloud infrastructure for processing transactions and storing data. This setup is cost-effective and reduces data traffic.
2. **Federal Agencies:** U.S. government agencies with similar security, audit, and privacy requirements can use a community cloud to share resources in a secure and efficient manner.
3. **Businesses with Similar Needs:** Multiple companies can share a cloud system or application, reducing the need for separate infrastructure while keeping their sessions logically separated.

Benefits of Community Cloud:

- **Openness and Impartiality:** Community clouds are open systems, reducing organizations' dependency on specific cloud providers. They offer many advantages while avoiding some of the limitations of public and private clouds.
- **Flexibility and Scalability:** Community clouds are flexible and can adapt to different use cases. They support various devices (smartphones, tablets, etc.) and allow organizations to scale their resources (hardware, services, and manpower) as their user base grows.

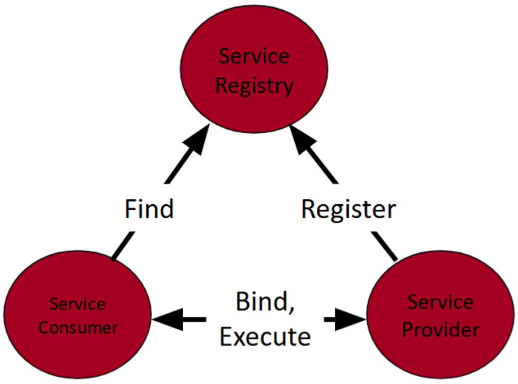
- **High Availability and Reliability:** Community clouds ensure that data and applications are always available by replicating them across secure locations. This redundancy protects against unexpected events and ensures the cloud remains reliable.
- **Security and Compliance:**
  - **Data Security:** Each organization in the community cloud has authorized access to its own data, and security measures prevent unauthorized access.
  - **Compliance:** Compliance with regulatory requirements is crucial. Organizations in a community cloud must ensure they follow the necessary rules while respecting the privacy and security of others' data. Agreements on compliance responsibilities help avoid confusion and streamline processes.
  - **Shared Access Control:** While multiple organizations have access to audit logs, agreements must clarify who handles specific tasks (e.g., reviewing logs) to save time and resources.

2.3 Cloud design and implementation using SOA

Service-Oriented Architecture (SOA) is a programming style where software is built using independent services that work together. These services can exchange data and coordinate actions, supporting business processes. SOA emphasizes flexibility and reusability, allowing developers to create systems that are efficient and scalable.

What is Service-Oriented Architecture (SOA)?

- SOA is a programming style, not a specific architecture.
- Applications are composed of independent "services."
- Each **service** is a building block that encapsulates a specific business process.
- Services are reusable and accessible over a network.
- The process involves three main steps: **Find**, **Bind**, and **Execute**.
- A well-known example of SOA is **Web Services.**



Key Actors in SOA

1. **Service Provider**
    - The owner of the service.
    - Manages the platform that offers the service.
2. **Service Registry**
    - Stores service descriptions.
    - Helps consumers find services that meet their needs.
3. **Service Consumer**
    - The client application that uses the service to fulfill specific functions.

Core Principles of SOA

1. **Formal Contract:**

    Every service must have a standard contract that defines:

    - Its functionality (what it does).
    - Data and policies involved.
    - How users interact with it.
2. **Loose Coupling:**
    - Services are designed independently of specific users or systems.
    - Enables reuse and flexibility.
3. **Abstraction:**
    - Hides internal details of the service.

- Allows users to interact without needing to understand the underlying technology.

4. **Reusability:**
    - Services are designed to be used across multiple applications.
    - This reduces development time and increases return on investment (ROI).
5. **Autonomy:**
    - Services have control over their processes.
    - Autonomy exists at both **runtime** (execution) and **design-time** (development).
6. **Statelessness:**
    - Services do not retain any data or state after processing a request.
    - This improves scalability and concurrent processing.
7. **Discoverability:**
    - Services are easy to find using registries or directories.
    - Metadata helps users understand their capabilities.
8. **Composability:**
    - Services can be combined to create new applications, saving time and resources.

Statelessness in SOA and REST

- **SOA:**
    - Can use both stateful and stateless communication.
    - Supports transactions and message sequences.
- **REST (Representational State Transfer):**
    - Always uses stateless communication.
    - Relies on document transfers with no awareness of client or server state.

SOA Properties – Self-Management

To reduce the need for manual management, SOA systems often include **self-properties**:

1. **Self-Configuration:**

   Services automatically adapt to new environments and tasks.

2. **Self-Organization:**

   Systems adjust themselves dynamically to achieve goals efficiently.

3. **Self-Healing:**

   Services remain available, even during partial outages.

4. **Self-Optimization:**

   Resources are allocated efficiently to optimize performance.

5. **Self-Protection:**

   Services detect and handle malicious events, acting as an "immune system."

Benefits of SOA

Business Benefits

- Focuses on solving business problems.
- Uses existing infrastructure effectively.
- Increases agility in responding to changes.

Technical Benefits

- Enables loose coupling for flexibility.
- Promotes autonomous services that are independent and reusable.
- Supports location transparency, allowing services to work regardless of physical location.
- Facilitates late binding, enabling decisions to be made at runtime.

2.4 Security, Trust and Privacy

Several security challenges exist in cloud computing. These include:

1. **Misconfiguration**

Incorrect security settings are a major cause of cloud data breaches. Many organizations lack proper strategies to secure their cloud-based systems.

2. **Unauthorized Access**

   Cloud infrastructure is accessible over the public internet, which makes it vulnerable to unauthorized access. While it benefits employees and customers, this open access increases the risk of cyberattacks.

3. **Insecure Interfaces/APIs**

   Cloud Service Providers (CSPs) offer APIs to help customers interact with services. While these are often well-documented, they can still be vulnerable if not properly secured.

4. **Account Hijacking**

   Weak passwords and password reuse make it easier for attackers to hijack accounts. Once an account is compromised, it can be used to access multiple services, which increases the impact of data breaches.

5. **Lack of Visibility**

   Since cloud resources are hosted off-site, traditional security tools may not be effective. This lack of visibility can hinder an organization's ability to monitor and secure cloud infrastructure.

6. **Malicious Insiders**

   Employees or contractors with authorized access can pose a significant security threat. Unlike external attackers, insiders already have access to sensitive data, making their actions harder to detect.

7. **Cyberattacks**

   Cloud infrastructure is often targeted by cybercriminals due to its high value and sometimes inadequate security measures.

Trust in Cloud Computing

Trust is the confidence users have in a system to deliver expected results. Building and maintaining trust in cloud services is essential for their success.

Factors Influencing Trust

- **Control**

Users feel more confident when they have control over their data. For example, when withdrawing cash from an ATM, users trust the process because they control the transaction.

- **Ownership**

  Trust can vary depending on who owns the data. For instance, an employer may trust a cloud service for company data but feel less secure when it involves employee data.

- **Prevention**

  Contracts, like Service Level Agreements (SLAs), are important for establishing trust. However, cloud systems should focus more on preventing failures than compensating for breaches, as data loss and reputational damage are often irreversible.

- **Security**

  Robust security measures are vital for maintaining trust. These include:

  - Access control
  - Data leakage prevention
  - Identity management
  - Protection against virtual machine (VM) attacks

Privacy in Cloud Computing

Privacy is the right of individuals to control their personal data, know how it is used, and prevent its misuse. Privacy concerns are critical in the online world, influencing user confidence and economic growth.

Key Privacy Challenges in Cloud Computing

- Who owns and operates the cloud services?
- Where is the data stored and how is it replicated?
- What legal rules govern data processing?
- How can providers ensure security and privacy?

Essential Aspects of Privacy

1. **Data Integrity**

Protects data from unauthorized deletion, modification, or misuse. Ensures only authorized users can access sensitive information.

2. **Data Confidentiality**

   Ensures private data remains secure through authentication and access control strategies.

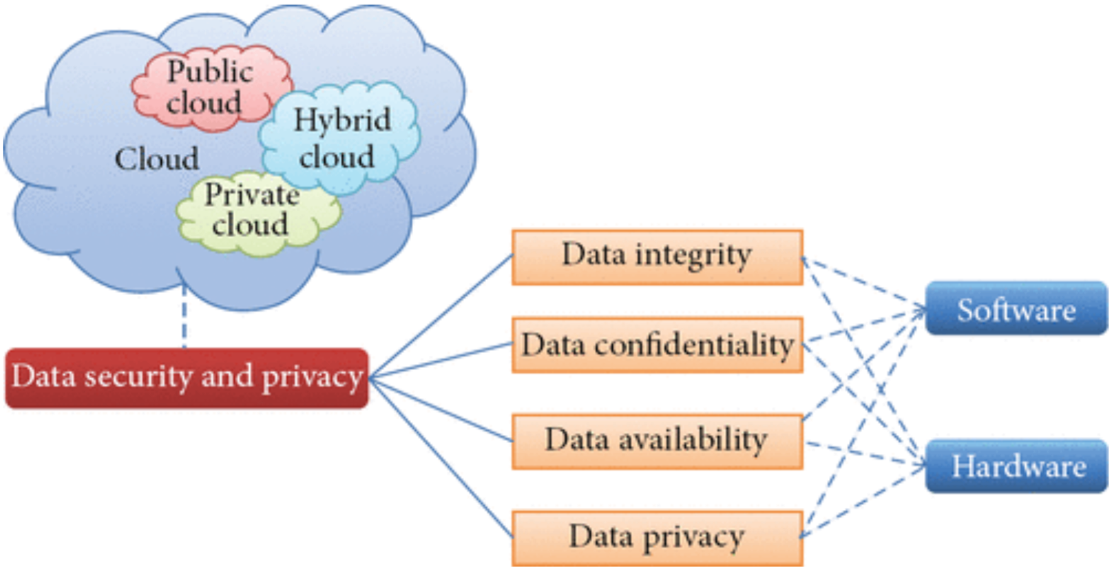
3. **Data Availability**

   Refers to the ability to recover and access data during unexpected events like hardware failures or disasters.

4. **Data Privacy**

   Allows individuals to control how and when their information is shared.

   - **When**: Concern over current or future data being shared.
   - **How**: Users may prefer manual sharing over automated alerts.
   - **Extent**: Users might prefer general data (e.g., regions) over specific details.



Security, Trust, and Privacy Elements

- **Data Integrity:** Protect data from unauthorized changes.
- **Data Confidentiality:** Use strong authentication to protect sensitive information.

- **Data Availability:** Ensure data is recoverable and accessible even during disasters.
- **Data Privacy:** Give users control over their data sharing preferences.