Department of Computer Science and Engineering

CLOUD COMPUTING (BCS601)

Module-03

Cloud Platform Architecture over Virtualized Datacenters

4.1 Cloud Computing and Service Models

- Over the past 20 years, the global economy has shifted from manufacturing to service industries.
- By 2010, 80% of the U.S. economy was service-based, with only 15% in manufacturing and 5% in agriculture and other sectors.
- Cloud computing is especially beneficial for the service industry and brings a new way of doing business computing.
- In 2009, the global cloud market was worth $17.4 billion; it was predicted to grow to $44.2 billion by 2013.
- Cloud application developers rent resources from large automated data centers instead of buying expensive hardware.
- Cloud platforms allow users to access and use applications from anywhere at low costs.
- These platforms are usually built on top of large data centers using virtualization.
- Cloud computing turns data centers into virtual systems with automated management of hardware, databases, interfaces, and applications.
- The goal of cloud computing is to improve data centers using automation and efficient resource use.

4.1.1 Public, Private, and Hybrid Clouds

- Cloud computing came from older systems like cluster, grid, and utility computing.
- Cluster and grid computing use many computers working together.
- Utility computing and SaaS let you pay only for what you use.
- Cloud computing gives services using big data centers.
- People can use cloud services from anywhere, anytime.

- Instead of moving data around, cloud sends programs to the data.
- This saves time and improves internet speed.
- Virtualization helps use resources better and cuts costs.
- Companies don't need to set up or manage servers themselves.
- Cloud provides hardware, software, and data only when needed.
- The goal is to replace desktop computing with online services.
- Cloud can run many different apps at the same time easily.

4.1.1.1 Centralized versus Distributed Computing

- ✓ Cloud computing is **distributed using virtual machines** in big data centers.
- ✓ Public and private clouds work over the **Internet**.
- ✓ Big companies like **Amazon, Google, and Microsoft** build **distributed cloud systems** for speed, reliability, and legal reasons.
- ✓ **Private clouds** (within companies) can connect to **public clouds** to get more resources.
- ✓ People may worry about using clouds in other countries unless strong **agreements (SLAs)** are made.

4.1.1.2 Public Clouds

- ✓ A **public cloud** is available to anyone who pays for it.
- ✓ It is run by cloud providers (like **Google, Amazon, Microsoft, IBM, Salesforce**).
- ✓ Users **subscribe** to use services like storage or computing power.
- ✓ Public clouds let users create and manage virtual machines online.
- ✓ Services are charged **based on usage** (pay-as-you-go).

4.1.1.3 Private Clouds

- ✓ A **private cloud** is built and used **within one organization** (not public).
- ✓ It is **owned and managed** by the company itself.
- ✓ Only the **organization and its partners** can access it — not the general public.
- ✓ It does **not sell services** over the Internet like public clouds do.
- ✓ Private clouds give **flexible, secure, and customized services** to internal users.
- ✓ They allow the company to keep **more control** over data and systems.
- ✓ Private clouds may affect **cloud standard rules**, but offer better **customization** for the company.

4.1.1.4 Hybrid Clouds

- ✓ A **hybrid cloud** combines both **public and private clouds**.
- ✓ It allows a company to use its **private cloud** but also get extra power from a **public cloud** when needed.
- ✓ Example: **IBM's RC2** connects private cloud systems across different countries.
- ✓ Hybrid clouds give access to the **company, partners, and some third parties**.
- ✓ **Public clouds** offer flexibility, low cost, and standard services.
- ✓ **Private clouds** give more **security, control, and customization**.
- ✓ **Hybrid clouds** balance the two, making **compromises** between sharing and privacy.

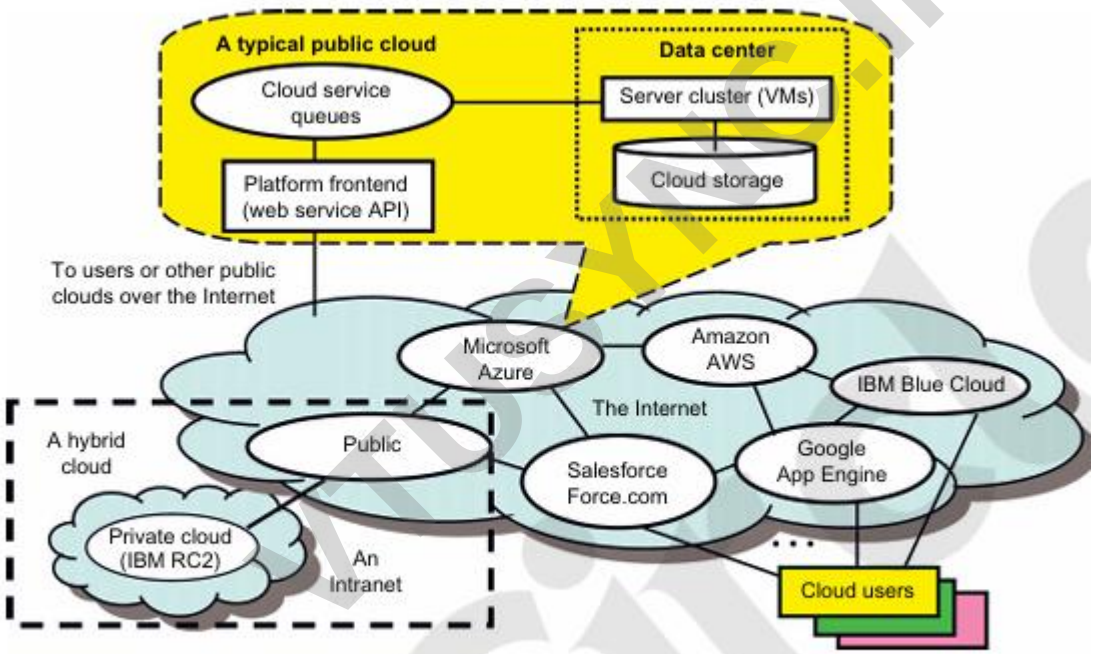

FIGURE 4.1
Public, private, and hybrid clouds illustrated by functional architecture and connectivity of representative clouds available by 2011.

4.1.1.5 Data-Center Networking Structure

✓ The **core of a cloud** is a **server cluster** made of many virtual machines (VMs).
- ✓ **Compute nodes** do the work; **control nodes** manage and monitor cloud tasks.
- ✓ **Gateway nodes** connect users to the cloud and handle **security**.
- ✓ Clouds create **virtual clusters** for users and assign jobs to them.
- ✓ Unlike old systems, clouds handle **changing workloads** by adding or removing resources as needed.
- ✓ **Private clouds** can support this flexibility if well designed.

- ✓ **Data centers** use **thousands to millions** of servers (e.g., Microsoft's Chicago center has 100,000 servers).
- ✓ **Supercomputers** and data centers are similar but have key differences.
    - Supercomputers use **separate storage** and **custom networks** (like fat trees or 3D torus).
    - Data centers use **server disks, memory**, and **standard IP networks** (like 10 Gbps Ethernet).

- ✓ Data center networks have **layers**:
    - **Layer 2**: server racks connected by fast switches
    - **Layer 3**: access and border routers that link to the Internet.

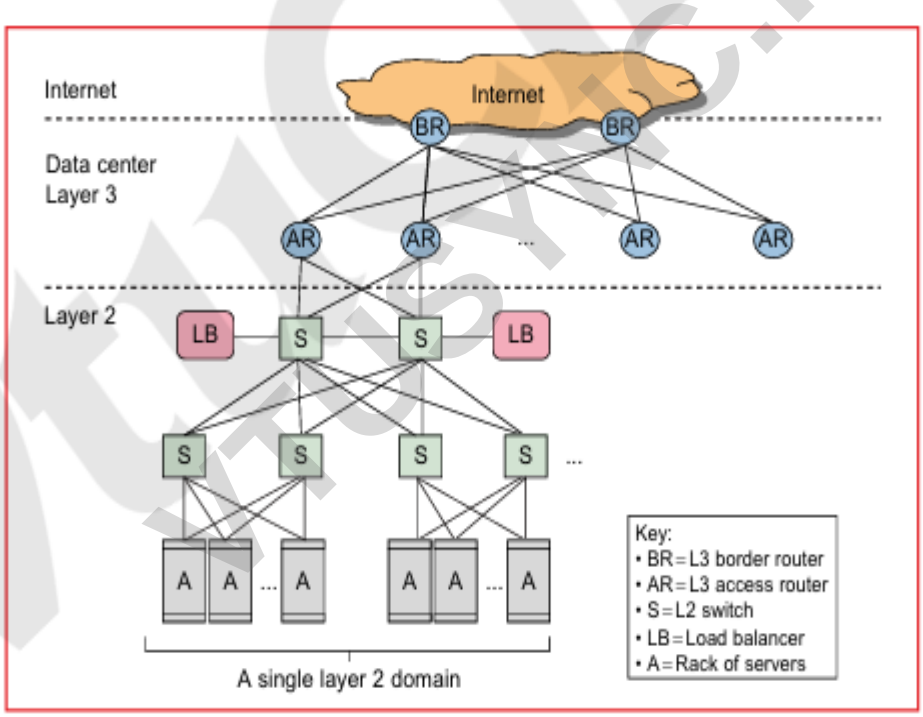

FIGURE 4.2

Standard data-center networking for the cloud to access the Internet.

4.1.1.6 Cloud Development Trends

- ✓ **Private clouds** will grow faster than **public clouds** in the future.
- ✓ **Private clouds** are more **secure** and **trustworthy** for companies.

- ✓ **Private clouds** may become **public clouds** or **hybrids** as they mature.
- ✓ **Hybrid clouds** (mix of private and public) will be common in the future.
- ✓ **Applications** (like email) use **service-access nodes** to connect to **internal cloud services**.
- ✓ **Supporting service nodes** help manage cloud tasks (like locking services).
- ✓ **Independent service nodes** provide specific services, like **geographical data**.
- ✓ **Clouds** improve **network efficiency** by reducing **data movement**.
- ✓ **Clouds** help solve the **petascale I/O problem** (handling large data).
- ✓ **Cloud performance** and **Quality of Service (QoS)** are still being tested in real use.

4.1.2 Cloud Ecosystem and Enabling Technologies

- ✓ T**raditional computing**: Buy and manage hardware every 18 months.
- ✓ **Cloud computing**: **Rent resources**, only pay for what you use.
- ✓ **Cloud saves 80-95%** compared to traditional computing.
- ✓ Small businesses can avoid **high upfront costs** with cloud.
- ✓ **Cloud market** may reach **$126 billion** by 2012.
- ✓ Clouds work as **service factories** with many data centers.
- ✓ Cloud computing reduces **costs and barriers**.

4.1.2.1 Cloud Design Objectives

- ✓ **Shift from desktops to data centers**: Move computing, storage, and software from local devices to data centers over the Internet.
- ✓ **Service provisioning and economics**: Cloud providers offer services with **SLAs**, focusing on **efficiency** and **pay-as-you-go pricing**.
- ✓ **Scalability**: Cloud platforms must be able to **scale** as the number of users grows.
- ✓ **Data privacy protection**: Ensure data privacy to build **trust** in cloud services.
- ✓ **High quality of services**: Standardize **Quality of Service (QoS)** to ensure **interoperability** between different cloud providers.

- ✓ **New standards and interfaces**: Solve the **data lock-in problem** with **universal APIs** for **flexibility** and **portability**.

4.1.2.2 Cost Model

Traditional IT costs:

- Users buy hardware and face **fixed costs** and **operational costs** (e.g., maintenance).
- **Costs increase with more users**.

Cloud computing costs:

- **Pay-as-you-go** model: No upfront cost, only pay for **operational expenses**.
- **Cloud computing is cheaper** because it avoids large initial investments and scales with demand.

Cloud computing is great for small businesses:

- No need to buy expensive equipment, just pay for what you use.
- Ideal for businesses that need **flexibility** and want to avoid heavy capital expenses.

Cloud Ecosystems:

- Cloud computing has created an ecosystem of **providers**, **users**, and **technologies**.
- **Private and hybrid clouds** are growing, offering flexible resources with public cloud involvement.

Private Cloud Ecosystem:

- **Four levels**: Consumer demands, cloud management, virtual infrastructure management, and VM management.
- Ecosystem tools help manage cloud resources but lack **standard interfaces**.

Cloud tools:

- **VM management tools** like **vSphere**, **oVirt**, and **OpenNebula** help manage VMs and virtualized resources.
- **Public cloud tools** like **Amazon EC2** and **Eucalyptus** support cloud infrastructure.

4.1.2.4 Surge of Private Clouds

- ✓ **Private clouds** use a company's **existing IT setup and staff**.
- ✓ **Both public and private clouds** handle changing workloads.
- ✓ **Public clouds** work best with **independent tasks** (no communication between jobs).

- ✓ **Private clouds** balance workloads **within the company's network** for better efficiency.
- ✓ **Private clouds** offer better **security, privacy**, and **testing environments**.
- ✓ **Public clouds** help avoid **big upfront costs** in hardware, software, and staff.
- ✓ Companies often start by **virtualizing their systems** to reduce operating costs.
- ✓ Big companies (like **Microsoft, Oracle, SAP**) use **policy-based IT management** to improve services.
- ✓ **IT as a service** boosts flexibility and avoids replacing servers often.
- ✓ This leads to better **IT efficiency and agility** for companies.

4.1.3 Infrastructure-as-a-Service (IaaS)

- ✓ **IaaS** means renting IT infrastructure like servers, storage, and networks over the Internet.
- ✓ It provides **virtual machines**, **storage**, **networks**, and **firewalls**.
- ✓ Users can choose their **own operating system and software**.
- ✓ Users **don't manage physical hardware**, only virtual resources.
- ✓ It's a **pay-as-you-go** model – no need to buy expensive equipment.
- ✓ IaaS is **scalable** – add or remove resources anytime.
- ✓ Great for **startups**, **developers**, and **large businesses**.
- ✓ Helps with **testing**, **hosting apps**, **data backup**, and **disaster recovery**.
- ✓ Examples of IaaS providers:
    - **Amazon EC2, S3**
    - **Microsoft Azure VMs**
    - **Google Compute Engine**
    - **IBM Cloud**
    - **Oracle Cloud Infrastructure (OCI)**
- ✓ Saves money and time by avoiding hardware setup.
- ✓ Ideal for companies needing **flexible and powerful IT resources**.

4.1.3 Platform-as-a-Service (PaaS)

- ✓ **Paas** provides a **platform** to build, test, and deploy applications.
- ✓ It includes **tools, libraries, databases**, and **runtime environments**.

- ✓ Developers **don't manage servers, storage, or infrastructure**.
- ✓ Focus is only on **writing and running code**.
- ✓ It handles **app hosting, scaling, updates**, and **security** automatically.
- ✓ **Faster development** because everything is ready to use.
- ✓ Great for **developers** and **software teams**.
- ✓ Useful for **web apps, mobile apps**, and **APIs**.
- ✓ Examples of PaaS providers:

  - **Google App Engine**
  - **Microsoft Azure App Service**
  - **Heroku**
  - **IBM Cloud Foundry**
  - **Red Hat OpenShift**

- ✓ Pay only for what you use – **no upfront setup or hardware costs**.
- ✓ Helps teams **collaborate** easily and **launch apps faster**.

4.1.4 Software-as-a-Service (SaaS)

- ✓ SaaS means using **software over the internet**.
- ✓ No need to **install or update** anything.
- ✓ **Accessible from any device** with internet.
- ✓ Software is managed by the **provider**, not the user.
- ✓ Users **pay monthly or yearly** (subscription model).
- ✓ **No hardware or server needed** by the user.
- ✓ Used for **email, file sharing, CRM, video calls**, etc.
- ✓ **Data is stored in the cloud** by the provider.
- ✓ Saves **time, money, and effort**.
- ✓ Great for **businesses and individuals**.
- ✓ **Examples of SaaS:**

  - **Gmail**
  - **Google Docs**
  - **Microsoft 365**
  - **Salesforce**
  - **Zoom**

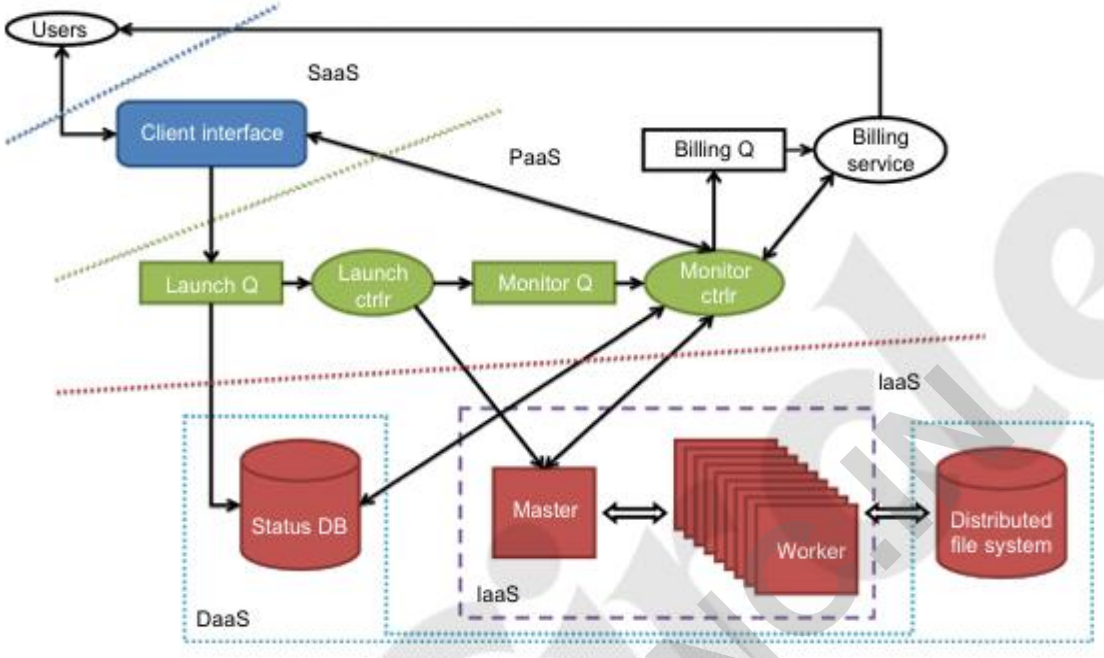

FIGURE 4.5
The IaaS, PaaS, and SaaS cloud service models at different service levels..

4.2 DATA-CENTER DESIGN AND INTERCONNECTION NETWORKS

A **data center** is a facility that houses a large number of servers connected through a high-speed network to provide computing, storage, and service resources. A **modular data center** is a smaller, portable version that fits inside a 40-foot container and can be easily deployed and expanded, offering flexibility and scalability for remote or rapid deployment needs.

4.2.1 Warehouse-Scale Data-Center Design

- **Large Data Centers**: Cloud computing uses huge data centers, sometimes as big as shopping malls, with hundreds of thousands of servers.
- **Lower Costs for Bigger Centers**: Bigger data centers are cheaper to run per server due to economies of scale.
- **Cost Breakdown**: Operating a large data center is cheaper, with lower network and storage costs compared to small ones.
- **Microsoft's Global Data Centers**: Microsoft runs around 100 data centers worldwide to support cloud services.

4.2.1.1 Data-Center Construction Requirements

- ✓ **Data Center Components**: Data centers use off-the-shelf components, such as servers with multi-core CPUs, DRAM, and disk drives. Servers are connected via switches for resource access.
- ✓ **Bandwidth and Latency**: Disk bandwidth varies between local and off-rack storage. Large data centers often face challenges due to discrepancies in latency, bandwidth, and capacity.
- ✓ **Failure and Reliability**: Hardware and software failures are common in large data centers. Redundant hardware and data replication ensure reliability, preventing data loss during failures.
- ✓ **Cooling System**: Data centers use raised floors to distribute cool air to server racks. Cold air is pumped into the floor and escapes through perforated tiles in front of racks. Hot air is then returned to the cooling units for recirculation. Advanced systems may also use cooling towers for efficiency.

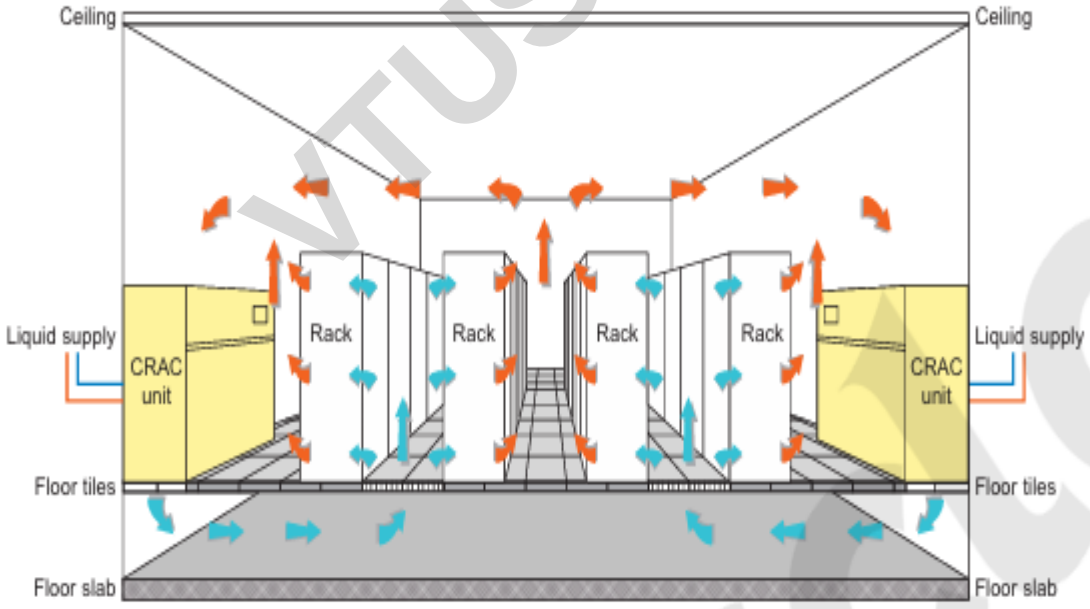

FIGURE 4.9

The cooling system in a raised-floor data center with hot-cold air circulation supporting water heat exchange facilities.

4.2.2 Data-Center Interconnection Networks

The interconnection network in a data center connects all servers in the cluster and is a vital component for performance.

The network must meet five key requirements:

1. **Low Latency**: Ensure fast communication between servers.
2. **High Bandwidth**: Enable data transfer with high throughput.
3. **Low Cost**: Keep the network affordable for large-scale implementations.
4. **MPI Support**: Support efficient message-passing interface (MPI) communication for parallel processing.
5. **Fault Tolerance**: Maintain network functionality even in the event of failures.

The network must handle both point-to-point (direct server-to-server) and collective communication (group communication) among servers.

4.2.1 Warehouse-Scale Data-Center Design

- Cloud computing runs on big buildings called **data centers**.
- Some data centers are huge—**as big as shopping malls** or **11 football fields**.
- A large data center can hold **400,000 to 1 million servers**.
- Small data centers might only have **1,000 servers**.
- **Bigger data centers are cheaper to run per server** (this is called "economies of scale").
- Running a **large data center** is cheaper than running a small one.
- For example, in a 400-server data center:
  - **Network cost** is $13 per Mbps
  - **Storage cost** is $0.40 per GB
- A small data center has:
  - **Network costs 7 times higher**
  - **Storage costs 5.7 times higher**
- **Microsoft has about 100 data centers** around the world.

4.2.1.1 Data-Center Construction Requirements

- **Most data centers use standard, off-the-shelf components** (not custom-built supercomputing parts).
- Each server has:
  - Multiple CPUs (with multiple cores)
  - 8 GB of memory (DRAM)
  - Four 1 TB disk drives

- **Servers are grouped in racks** (40 servers per rack) and connected through **rack-level switches**.
- All racks are connected to a **cluster-level switch**.
- **Disk speed differences**:
  - **Local disk**: 200 MB/s
  - **Off-rack disk (via shared link)**: 25 MB/s
- The **total disk storage** is about **10 million times larger than the memory (DRAM)**.
- Big applications must handle **differences in speed and size between memory and storage**.
- **Large-scale data centers are more cost-efficient**, but...
- **Failures are common** (about 1% of servers can fail at any time):
  - Hardware issues (CPU, disk, network)
  - Software bugs
- Power failures can shut down the entire data center
- To **avoid losing data**, systems must:
  - Use **redundant hardware**
  - **Store multiple copies of data** in different places
  - Keep data available even during failures (high reliability).

4.2.1.2 Cooling System of a Data-Center Room

- **Data centers use raised floors** (2–4 feet above concrete) to **hide cables and distribute cool air**.
- **Cold air is blown under the floor** by **CRAC units** (Computer Room Air Conditioning).
- Cold air **comes out through perforated tiles** in front of server racks.
- **Server racks are arranged in cold and hot aisles**:
  - **Cold aisle**: where cold air enters servers
  - **Hot aisle**: where hot air exits
- **Hot air is pulled back** into the CRAC units, **cooled**, and **recycled** into the floor.
- Coolant enters at **12–14°C**, warms up, and then returns to a **chiller**.
- **Newer data centers use cooling towers** to help cool the system more efficiently:
  - **Water absorbs heat** from the coolant in a heat exchanger
  - The heated water is **cooled in towers** and reused
- In short, data centers use a carefully designed air and water system to **keep servers from overheating** by constantly recycling and cooling air and liquid.

4.2.2 Data-Center Interconnection Networks

A critical part of data center design is the interconnection network that links all servers within the cluster. This network must be carefully designed to meet five key requirements: low latency for fast communication, high bandwidth to handle large volumes of data, low cost to maintain affordability, support for the Message Passing Interface (MPI) used in parallel processing, and fault tolerance to ensure the system keeps running even when some components fail. The inter-server network must efficiently handle both point-to-point communication between

individual servers and collective communication among multiple servers at once. Meeting these needs is essential for the smooth and reliable operation of large-scale data centers.

4.2.2.1 Application Traffic Support

- The network must support **all MPI communication types** (point-to-point and collective).
- It should have **high bisection bandwidth** to handle large data flow across the network.
- **One-to-many communication** is needed for tasks like **distributed file access**.
- **Metadata master servers** must talk to many **slave servers**.
- The network should support **MapReduce operations** at high speed.
- It must handle **different types of network traffic** used by various applications.

4.2.2.2 Network Expandability

- The **network must be expandable** to support thousands or more servers in the future.
- **Topology should allow easy restructuring** as more servers are added.
- It should support **load balancing and smooth data movement** to avoid bottlenecks.
- **Fat-tree and crossbar networks** can be built using **low-cost Ethernet switches**.
- As server numbers grow, **network design becomes more complex**.
- **Modular growth** is important—**server containers** are used as building blocks.
- Each **data-center container** can hold **hundreds to thousands of servers**.
- Containers are **pre-built units**—just connect power, network, and cooling to start.
- This **reduces setup and maintenance costs**.
- Networks can be built by:
  - Creating a **core connection backbone** first, then linking end servers.
  - Using **external switching and cabling** to connect multiple containers.

4.2.2.3 Fault Tolerance and Graceful Degradation

- The **network must handle link or switch failures** without crashing.
- There should be **multiple paths between any two servers** for reliability.
- **Redundancy** (extra hardware/software) helps tolerate failures.
- **Data and computing** are replicated across servers for fault tolerance.
- Similar **redundancy should apply to the network** (hardware and software).
- **Software should detect failures** and reroute traffic around broken links.
- This should happen **automatically and without affecting cloud services**.
- The network should **degrade gracefully** during small failures.

- **Hot-swappable components** (easy to replace while running) are preferred.
- The design should avoid any **single point of failure**.
- Networks are usually built in **two layers**:
  - **Lower layer** connects to end servers
  - **Upper layer** connects server groups (backbone network)
- This **hierarchical structure** supports modular container-based data centers.

4.2.2.4 Switch-centric Data-Center Design

- There are **two approaches** to building data-center-scale networks:
  1. **Switch-centric**: Uses switches to connect servers, no changes needed on the servers themselves.
  2. **Server-centric**: Modifies the operating system on servers with special drivers to handle traffic.
- **Fat-tree network design** is an example used in data centers:
- **Two layers**:
  - **Bottom layer**: Server nodes connected by **edge switches**.
  - **Upper layer**: Aggregates the edge switches through **aggregation switches**.
- **Pods**: Each pod contains edge switches, aggregation switches, and their connected nodes.
- **Core switches** link different pods.
- **Multiple paths** between any two servers provide **fault tolerance** if links fail.
- Key benefits of the fat-tree design:
  - **Switch failure** doesn't affect the entire network (alternate paths exist).
  - **Edge switch failure** affects only a small number of servers.
  - **Higher bandwidth** within pods supports massive data movement for cloud applications.
  - **Low-cost Ethernet switches** are used, making the design more affordable.
  - **Routing algorithms** inside switches help find alternate paths if there's a failure.
  - **Server nodes** aren't affected by switch failures as long as alternate paths are intact.

4.2.3 Modular Data Center in Shipping Containers

- **Modern data centers** are often made up of **truck-towed containers** that house server clusters.
- **SGI ICE Cube modular data centers** have containers that hold hundreds of blade servers in racks, with fans circulating heated air through a **heat exchanger** to cool it for the next rack.
- A single container can hold **46,080 processing cores** or **30 PB of storage**.

- **Container-based data centers** offer benefits like:
  - **Lower power consumption**
  - **Higher computing density**
  - **Mobility** (easy to move to areas with cheaper electricity or better cooling)
- **Cooling costs** can be reduced by **80%** compared to traditional data centers through efficient **chilled air and cold water circulation**.
- These data centers are often built in **locations with cheaper utilities** and more efficient cooling.
- **Modular containers** can form a large-scale data center, like a **shipping yard** of containers.
- **Centralized management** (in a single building) is important for handling **data integrity**, **server monitoring**, and **security management**.

4.2.3.1 Container Data-Center Construction

- **Modular data centers** are housed in **truck-towable containers** that include all necessary components like network, computer, storage, and cooling systems.
- **Cooling efficiency** can be improved by adjusting **water and airflow** with better airflow management.
- Data center construction evolves in stages: starting with a **single server**, moving to a **rack system**, and finally to a **container system**.
- Building a **rack of 40 servers** might take half a day, but expanding to a **container system** with 1,000 servers requires more time for layout, power, networking, cooling, and testing.
- The container must be **weatherproof** and **easily transportable**.
- **Modular construction** can be completed quickly (a few days) if all components and supplies are ready.
- This design is suitable for many cloud applications, like in the **healthcare industry**, where **data centers** can be installed at **clinic sites**.
- **Information exchange** with a central database and ensuring **data consistency** becomes a challenge in **hierarchical data center** structures.
- **Security** in cloud services may require the use of **multiple data centers**.

FIGURE 4.11
A modular data center built in a truck-towed ICE Cube container, that can be cooled by chilled air circulation with cold-water heat exchanges.

4.2.4 Interconnection of Modular Data Centers

- **Container-based data-center modules** can be combined to build large-scale data centers using multiple container modules.
- One example of such a network design is the **server-centric BCube network** for modular data centers.
- The **BCube network** uses a **layered structure**:
  - **Level 0** consists of server nodes.
  - **Level 1** contains switches connecting to Level 0 servers.
  - The structure is recursively built, where each **BCubek** is made up of **n BCubek-1** networks with **n-port switches**.
- In **BCube1**:
  - Each server in **BCube0** connects to the corresponding port of a **Level 1 switch**.
  - Servers have multiple ports attached, allowing additional devices to connect.
- The BCube network provides **multiple paths** between any two nodes, offering:
  - **Extra bandwidth** for different cloud application communication patterns.
- **Routing support** via a **kernel module** in the server's OS, allowing packet forwarding without modifying upper layer cloud applications.
- This design allows cloud applications to run on top of the BCube network without requiring changes.

4.2.4.1 Inter-Module Connection Networks

- **BCube** networks are used inside **server containers** in modular data centers.
- However, to connect multiple containers, a new level of **networking** is required, leading to the design of **MDCube** (Modularized Datacenter Cube).

- **MDCube** connects multiple **BCube containers** using **high-speed switches** inside each BCube, creating a network topology for inter-container communication.
- **MDCube** builds a **virtual hypercube** at the container level, complementing the internal **cube structure** of each BCube.
- The **MDCube** network is scalable, supporting large-scale data centers and cloud applications.
- The architecture supports **cloud application communication patterns** and helps create a **high-performance inter-container network**.
- For detailed information, readers can refer to the full study by Wu et al. [82], which includes implementation and simulation results of this design.

4.2.5 Data-Center Management Issues

- **Quality Service for Users**: The data center should ensure quality service for users for at least 30 years.
- **Controlled Information Flow**: The system should streamline information flow and focus on high availability and consistent services.
- **Multiuser Manageability**: The system must handle all data center functions, such as traffic flow, database updates, and server maintenance.
- **Scalability**: The system should grow with increasing workload, with scalable storage, processing, I/O, power, and cooling.
- **Reliability in Virtualized Infrastructure**: Features like failover, fault tolerance, and VM live migration should be integrated to ensure recovery from failures or disasters.
- **Low Cost**: Minimize costs for both users and providers, including operational expenses.
- **Security and Data Protection**: Strong security measures should be in place to protect against attacks, ensure data privacy, and maintain integrity.
- **Green Technology**: Data centers should focus on energy efficiency and reducing power consumption.

4.2.5.1 Marketplaces in Cloud Computing Services

- **Factory Racking and Packing**: Data centers should be built efficiently in factories, avoiding complex packaging layers at the customer site.
- **Custom-Crafted vs. Prefabricated**: While modular designs are more space-efficient, data centers are still typically custom-crafted rather than being prefab units.
- **High Power Density**: Modular data centers can support high power densities, exceeding **1250 W/sq ft**.

- **Flexible Installation Locations**: These data centers can be installed on rooftops or parking lots.
- **Future Upgrades**: It's important to include enough **redundancy** and flexibility in the design to allow for future upgrades as needs grow.

4.3 ARCHITECTURAL DESIGN OF COMPUTE AND STORAGE CLOUDS

4.3.1 A Generic Cloud Architecture Design

An Internet cloud is envisioned as a public cluster of servers provisioned on demand to perform collective web services or distributed applications using data-center resources.

4.3.1.1 Cloud Platform Design Goals

- **Scalability**: The system should easily expand by adding more servers and network capacity as needed, supporting growing workloads and user demands.
- **Virtualization**: Cloud management must support both physical and virtual machines, allowing flexible resource allocation and efficient operation.
- **Efficiency**: The platform should be optimized for performance and resource use, ensuring cost-effectiveness and smooth operations.
- **Reliability**: Data should be replicated across multiple locations, so even if one data center fails, the data remains accessible. This ensures system uptime and continuous service.

4.3.1.2 Enabling Technologies for Clouds

Driving Forces Behind Cloud Computing:

1. **Ubiquitous Networking**: Widespread broadband and wireless networking allow seamless access to cloud services.
2. **Falling Storage Costs**: Decreasing prices for storage make it more affordable to build and maintain large-scale data centers.
3. **Improvements in Internet Software**: Advancements in software have enhanced cloud capabilities, making services more reliable and efficient.

Benefits to Cloud Users and Providers:

- **Users** can easily scale their resources, reduce costs, experiment with new services, and discard unused capacity.
- **Service Providers** can increase resource utilization through technologies like **multiplexing**, **virtualization**, and **dynamic provisioning**.

Technologies Enabling Cloud Computing:

1. **Hardware Advancements**: Improvements in multicore CPUs, memory chips, and disk arrays allow for faster and more powerful data centers with vast storage capacity.
2. **Resource Virtualization**: Virtualization enables quick cloud deployment and helps with disaster recovery.
3. **Service-Oriented Architecture (SOA)**: Supports flexible cloud service design and integration.
4. **Software as a Service (SaaS) & Web 2.0 Standards**: These have enabled cloud applications and services to be easily accessible and scalable.
5. **Internet Performance**: Better network infrastructure ensures fast and reliable cloud access.
6. **Large-Scale Distributed Storage**: A foundational technology for handling vast amounts of data across cloud environments.
7. **License Management and Billing**: Advances in managing licenses and automating billing help streamline cloud service operations.

4.3.1.3 A Generic Cloud Architecture

- **Cloud Platform Structure**:
  - The cloud is made up of many servers that are added or removed as needed.
  - These servers can either be physical machines or virtual ones (VMs).
- **Key Components**:
  - **User Interfaces**: These are used by people to request cloud services.
  - **Provisioning Tool**: It allocates the needed resources to fulfill those requests.
  - **Storage**: Data is stored in distributed systems, like SANs.
  - **Database Systems**: Manage the cloud's data.
  - **Security Devices**: Protect the cloud with things like firewalls.
  - **APIs**: Allow developers to connect with cloud services.
  - **Monitoring Tools**: Keep track of how resources are used and if everything is working well.

- **Management of Resources**:
  - The software in the cloud automatically handles resources, like adding or removing servers.
  - Big companies like Google and Microsoft have data centers around the world to make their clouds work efficiently.
- **Cloud Types**:
  - **Private Clouds**: For internal use by a single organization.
  - **Public Clouds**: Open to everyone, managed by third-party companies.
  - **Hybrid Clouds**: A mix of private and public clouds for special needs.
- **Security Concerns**:
  - **Trust & Reputation**: Systems that ensure resources are safe and reliable.
  - **Security Monitoring**: Keeps the cloud safe from attacks and breaches.
  - **Privacy Issues**: Protecting the data and ensuring only authorized access.

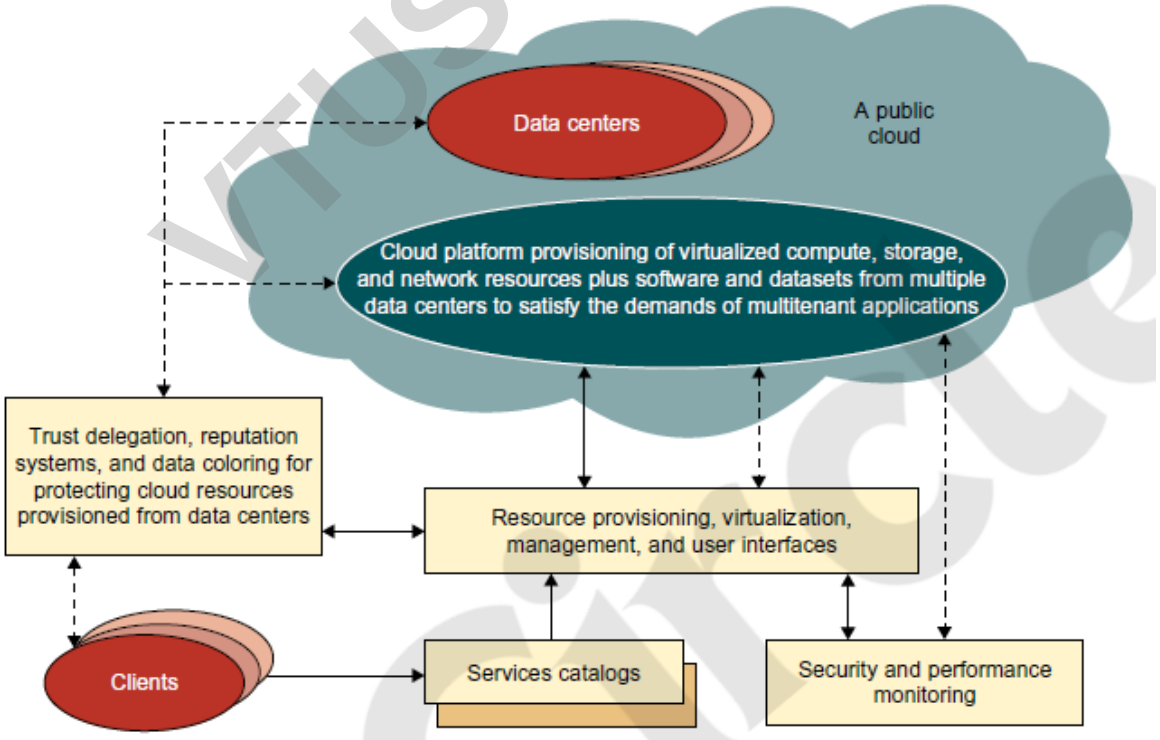

FIGURE 4.14

A security-aware cloud platform built with a virtual cluster of VMs, storage, and networking resources over the data-center servers operated by providers.

4.3.2 Layered Cloud Architectural Development

cloud Architecture – 3 Main Layers

1. Infrastructure Layer (IaaS)

- Base layer of cloud architecture.
- Provides virtual servers, storage, and networking.
- Uses virtualization to manage physical hardware.
- Example: Amazon EC2.

2. Platform Layer (PaaS)

- Middle layer built on top of infrastructure.
- Offers tools for developing, testing, and running apps.
- Ensures scalability, security, and reliability.
- Acts like a software environment for developers.

3. Application Layer (SaaS)

- Top layer where users access software directly.
- Includes apps for office work, CRM, finance, etc.
- Example: Salesforce.com.

Cloud Deployment Types

- **Public Cloud** – via Internet (e.g., Google Cloud).
- **Private Cloud** – via internal network.
- **Hybrid Cloud** – mix of both.

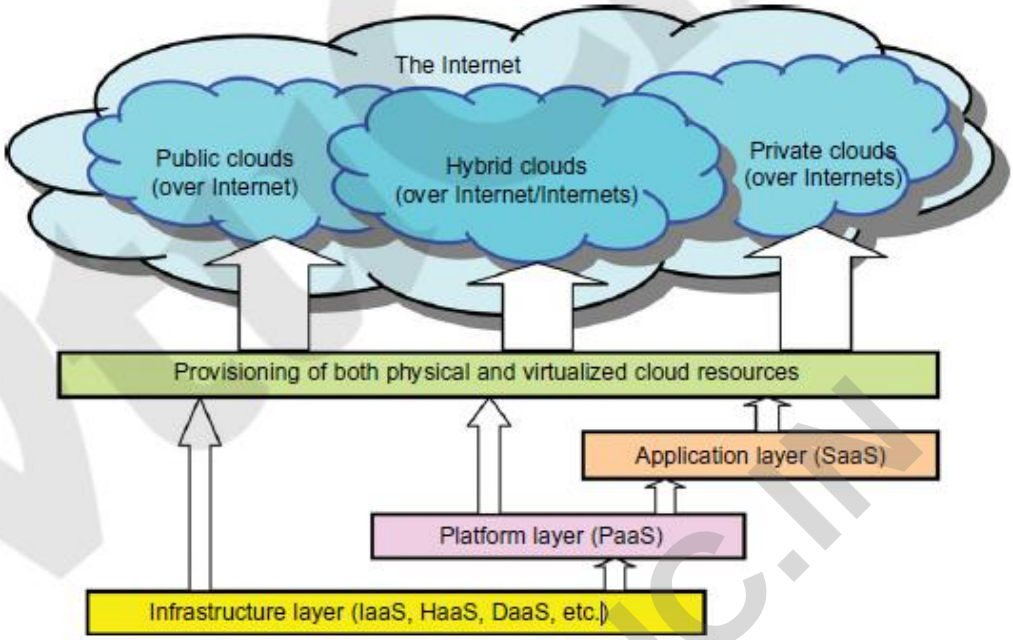

FIGURE 4.15

Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

4.3.2.1 Market-Oriented Cloud Architecture

- Users need guaranteed Quality of Service (QoS) from cloud providers.
- QoS expectations are defined in Service Level Agreements (SLAs).
- Traditional resource management is not suitable for cloud environments.
- A market-oriented approach balances resource supply and demand.
- Economic incentives help reduce costs and improve service quality.
- Users or brokers send service requests to the cloud from anywhere.
- SLA Resource Allocator manages interaction between users and data centers.
- Service Request Examiner checks QoS needs and accepts/rejects requests.
- VM Monitor tracks availability and usage of virtual machines (VMs).
- Service Request Monitor observes the progress of running tasks.
- Pricing mechanism sets charges based on time, demand, and availability.
- Accounting mechanism logs actual usage and calculates final costs.
- Dispatcher runs accepted tasks on assigned VMs.
- Multiple VMs can run on one physical machine with different OS environments.

- VMs can be created or stopped on demand for flexibility.
- Resource allocation is based on real-time needs and historical data.
- The system prevents overload and ensures efficient use of resources.
- This architecture supports fair pricing, scalability, and QoS delivery.

4.3.2.2 Quality of Service Factors

- Data centers contain multiple servers to meet cloud service demands.
- Cloud services support business operations and must meet QoS needs like time, cost, reliability, and security.
- QoS requirements are **not fixed** and can change over time.
- Cloud systems must **prioritize customer needs** since customers pay for the services.
- Current cloud systems lack strong support for **dynamic SLA negotiation**.
- SLA negotiation mechanisms are needed to handle changing user demands and alternate offers.
- Clouds must support **customer-driven service management** based on individual profiles and needs.
- **Computational risk management** helps identify and manage risks in service execution.
- Resource management should use **market-based strategies** to balance user needs and system efficiency.
- **Autonomic resource management** allows the system to adapt to changes without manual intervention.
- Virtual machines (VMs) enable flexible and dynamic resource allocation.
- Resources are shared and adjusted in real-time based on service demands.
- The system must handle **multiple competing service requests** effectively.
- Overall goal: maintain SLA commitments while adapting to evolving conditions and customer expectations.

4.3.3 Virtualization Support and Disaster Recovery

One very distinguishing feature of cloud computing infrastructure is the use of system virtualization and the modification to provisioning tools. Virtualization of servers on a shared cluster can consolidate web services. As the VMs are the containers of cloud services, the provisioning tools will first find the corresponding physical machines and deploy the VMs to those nodes before scheduling the service to run on the virtual nodes.

In addition, in cloud computing, virtualization also means the resources and fundamental infrastructure are virtualized. The user will not care about the computing resources that are used for providing the services. Cloud users do not need to know and have no way to discover physical resources that are involved while processing a service request. Also, application developers do not care about some infrastructure issues such as scalability and fault tolerance (i.e., they are virtualized). Application developers focus on service logic.

4.3.3.1 Hardware Virtualization

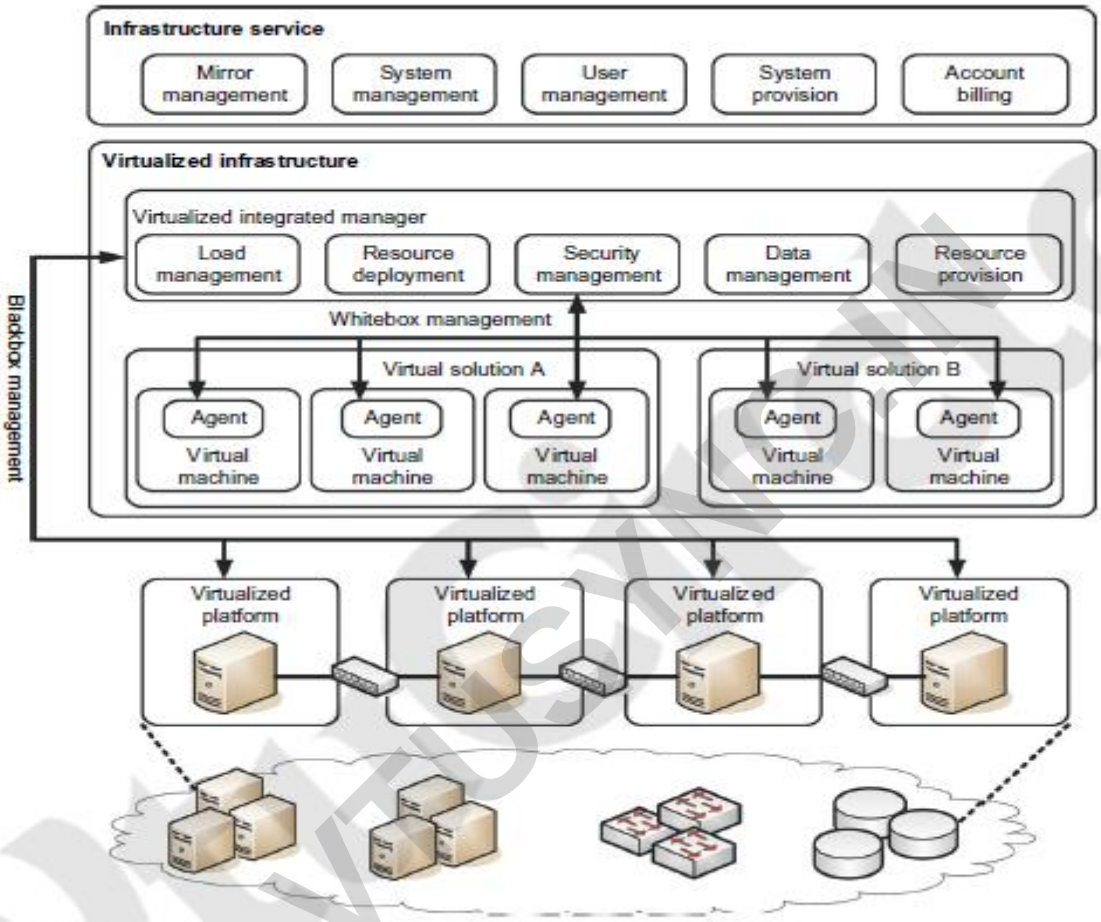

FIGURE 4.17
Virtualized servers, storage, and network for cloud platform construction.

- Virtualization software simulates hardware to run unmodified operating systems.
- It enables legacy software and older OSes to run in modern cloud environments.
- Developers can use any OS or programming language, increasing flexibility.
- Development and deployment environments can now be identical, reducing runtime errors.
- Virtualization supports running third-party programs on cloud platforms.
- VMs offer flexible runtime environments, freeing users from hardware/software setup worries.
- Traditional resource sharing (users/groups) lacks flexibility and customization.
- VMs allow users full privileges and system customization without affecting others.

- Multiple VMs can run different OSes on the same physical server.
- Virtual disk storage and virtual networks are created for each VM.
- Virtualized resources are pooled and managed by special servers.
- Virtualized integration managers handle system tasks like load balancing, security, and provisioning.
- Each VM platform can deliver custom virtual solutions for different user needs.
- Virtualization ensures complete separation and isolation among users.
- Resource management includes mirror, system, user, data, and security management.
- Recovery from failures is faster with VMs compared to traditional disaster recovery methods.
- AWS uses Xen VMs; Azure uses common language runtime; Google App Engine uses Python handlers.
- Storage models vary—AWS has block and blob stores, Azure uses SQL Data Services, GAE uses BigTable.
- Network configurations are mostly automated and hidden from users, with scaling managed internally.

4.3.3.2 Virtualization Support in Public Clouds

- **AWS** offers **full VM-level virtualization**, giving users high flexibility to run custom applications.
- **GAE (Google App Engine)** offers **limited, application-level virtualization**, restricting users to Google's predefined services.
- **Microsoft Azure** provides **programming-level virtualization** through the **.NET framework**.
- **VMware tools** support virtualization on **workstations, servers, and data centers**.
- **Microsoft virtualization tools** are designed for **PCs and some specific servers**.
- **XenEnterprise** tools are used for **Xen-based server virtualization** only.
- The IT industry is widely adopting cloud computing due to its benefits.
- **Virtualization** supports **high availability (HA)**, **disaster recovery**, **dynamic load balancing**, and **resource provisioning**.
- **Cloud computing** and **utility computing** both rely on virtualization for scalability and automation.

4.3.3.3 Storage Virtualization for Green Data Centers

- IT power consumption in the U.S. has more than doubled, now using 3% of the nation's total energy.
- A major cause is the high number of power-hungry data centers.
- Over half of Fortune 500 companies are adopting new energy-saving policies.
- Virtualization significantly reduces energy costs by lowering physical server usage.
- Surveys by IDC and Gartner confirm virtualization's role in cutting power consumption.

- The IT industry is becoming more energy-conscious due to rising power concerns.
- There is a growing need to save energy because alternative energy options are limited.
- **Virtualization** and **server consolidation** help reduce the number of physical machines needed.
- **Green data centers** aim to use energy-efficient infrastructure.
- **Storage virtualization** adds to energy savings by optimizing storage usage.
- Together, these efforts support the goal of **green computing**.

4.3.3.4 Virtualization for IaaS

- VM technology allows users to build **custom environments** on physical infrastructure.
- **Underutilized servers** can be consolidated into fewer machines, saving resources.
- VMs can run **legacy code** without affecting other system interfaces or APIs.
- VMs improve **security** by using **sandboxes** to isolate risky applications.
- VMs support **performance isolation**, enabling **better QoS guarantees** for customers.

4.3.3.5 VM Cloning for Disaster Recovery

- Traditional recovery (physical-to-physical) is **slow, complex, and costly**.
- Recovery of one VM by another **eliminates OS and agent installation**, reducing recovery time by **about 60%**.
- **VM encapsulation** enables **faster disaster recovery**.
- **Cloning VMs** on remote servers offers an effective recovery solution.
- **Only one clone VM is active**; others stay **suspended** until needed.
- **Cloud control centers** activate the clone VM in case of failure.
- **Snapshots** allow **live migration** with minimal downtime.
- Only **updated data and states** are sent to keep clone VMs current.
- **Recovery Point Objective (RPO)** and **Recovery Time Objective (RTO)** depend on **snapshot frequency**.
- **Security** must be maintained during **live VM migration**.

4.3.4 Architectural Design Challenges

There are **six open challenges** in cloud architecture development.These challenges are identified by **Armbrust et al.** as both **obstacles and opportunities.**

4.3.4.1 Challenge 1–Service Availability and Data Lock-in Problem

- Relying on a single cloud provider creates a single point of failure.
- Using multiple cloud providers improves high availability (HA) and fault tolerance.
- DDoS attacks pose threats to cloud service availability.
- Some utility computing services help SaaS providers scale quickly to defend against DDoS.
- Interoperability among cloud platforms has improved, but APIs are still proprietary.
- Proprietary APIs cause data lock-in, making it hard to move data or applications.
- Standardizing APIs allows easier migration across cloud providers.
- Standard APIs support both public and private cloud usage (hybrid model).
- This enables "surge computing" — using public cloud resources when private cloud capacity is exceeded.

4.3.4.2 Challenge 2–Data Privacy and Security Concerns

- Public cloud networks are more exposed to attacks than private networks.
- Existing technologies like encrypted storage, VLANs, and firewalls can help secure cloud systems.
- Encrypting data before storing it in the cloud adds protection.
- Some laws require SaaS providers to store data within national borders.
- Traditional network attacks include DoS, buffer overflows, spyware, and malware.
- Cloud-specific threats include hypervisor malware, guest VM hopping, and VM rootkits.
- VM migrations are vulnerable to man-in-the-middle attacks.
- Passive attacks focus on stealing data or passwords.
- Active attacks can damage servers by altering kernel data structures.

4.3.4.3 Challenge 3–Unpredictable Performance and Bottlenecks

- Multiple VMs can share CPU and memory easily, but sharing I/O causes performance issues.
- Example: 75 EC2 instances show good memory bandwidth (1,355 MB/s) but poor disk write bandwidth (55 MB/s).
- I/O interference occurs when VMs compete for disk access.
- A solution is to improve I/O architecture and OS support for virtualizing interrupts and I/O channels.
- Internet applications are increasingly data-intensive and distributed across cloud boundaries.
- This complicates data placement and transfer, increasing cost and latency.

- Effective data traffic and placement strategies are essential to avoid bottlenecks.
- Amazon's CloudFront is one example addressing these data transfer challenges.
- Bottleneck links should be widened and underperforming servers removed to improve overall performance.

4.3.4.4 Challenge 4–Distributed Storage and Widespread Software Bugs

- Cloud application databases are constantly growing and need scalable storage solutions.
- The goal is to design distributed SANs that scale up or down on demand.
- Data centers must support scalability, durability, and high availability (HA).
- Ensuring data consistency in SAN-connected cloud data centers is a significant challenge.
- Large-scale distributed bugs are hard to reproduce, requiring debugging in live production environments.
- Data centers typically do not allow in-production debugging.
- Virtual machines (VMs) can help capture valuable debugging information.
- Well-designed simulators offer another method for debugging distributed systems.

4.3.4.5 Challenge 5–Cloud Scalability, Interoperability, and Standardization

- Storage and network bandwidth are billed based on the number of bytes used.
- Computation charges vary by virtualization level; GAE charges by CPU cycles and auto-scales with load, while AWS charges hourly per VM instance, even if idle.
- There is an opportunity to scale resources up and down rapidly based on load variation to save costs without violating SLAs.
- Open Virtualization Format (OVF) provides a secure, portable, and platform-independent way to package and distribute VMs.
- OVF supports packaging software to run on any virtualization platform, regardless of host or guest OS.
- It includes transport mechanisms for VM templates and supports multi-VM virtual appliances.
- Research is needed to create hypervisor-agnostic VMs and enable live migration between x86 Intel and AMD systems.
- Further efforts are required to support legacy hardware and achieve effective load balancing across heterogeneous platforms.

4.3.4.6 Challenge 6–Software Licensing and Reputation Sharing

- Many cloud providers initially use open source software due to commercial software licensing not fitting utility computing well.
- There is an opportunity for open source to stay popular or for commercial vendors to adapt licensing models for cloud use.
- Combining pay-for-use and bulk-use licensing schemes can broaden business reach.
- One customer's bad behavior can damage the cloud's reputation, such as EC2 IP blacklisting by spam-prevention services affecting VM installation.
- Creating reputation-guarding services, similar to trusted email services, could protect cloud providers and users.
- Legal liability transfer between cloud providers and customers is a challenge that must be addressed in SLAs.

4.4 PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE

4.4.1 Public Clouds and Service Offerings

- Cloud services are requested by IT admins, software vendors, and end users.
- Five levels of cloud players exist, with individual and organizational users at the top level demanding different services.
- SaaS application providers mainly serve individual users.
- IaaS and PaaS providers primarily serve business organizations.
- IaaS offers compute, storage, and communication resources to both applications and organizations.
- PaaS providers define the cloud environment and support infrastructure services and organizational users.
- Cloud services depend on advances in virtualization, SOA, grid management, and power efficiency.
- Consumers buy cloud services as IaaS, PaaS, or SaaS.
- Many entrepreneurs offer value-added utility services to large user bases.
- The cloud industry grows as enterprises outsource computing and storage to professional providers.
- Provider service charges are usually much lower than the cost of frequent server replacements.
- Table 4.5 summarizes profiles of five major cloud providers as of 2010.

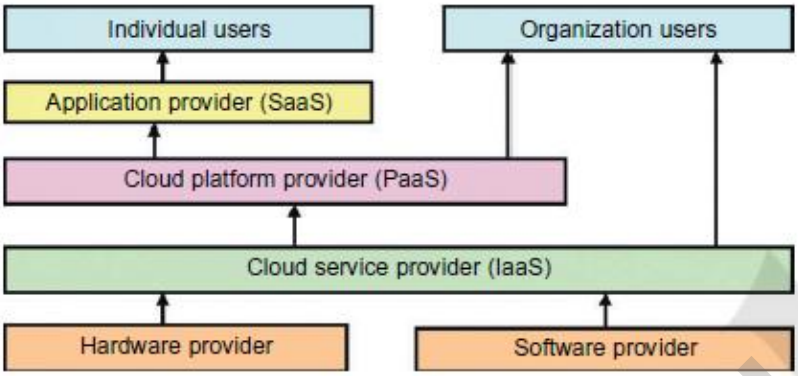

FIGURE 4.19

Roles of individual and organizational users and their interaction with cloud providers under various cloud service models.

4.4.2 Google App Engine (GAE)

- Google runs the world's largest search engine infrastructure.
- They have expertise in processing massive amounts of data.
- This led to innovations in data-center design and scalable programming models like MapReduce.
- Google operates hundreds of data centers worldwide with over 460,000 servers.
- Around 200 data centers run cloud applications simultaneously.
- Data stored includes text, images, and videos, all replicated for fault tolerance.
- Google's App Engine (GAE) is a PaaS platform for various cloud and web applications.

4.4.2.1 Google Cloud Infrastructure

- Google has led cloud development by using many data centers worldwide.
- Popular cloud services from Google include Gmail, Google Docs, and Google Earth, all supporting many users with high availability.
- Google's major technologies include Google File System (GFS), MapReduce, BigTable, and Chubby.
- In 2008, Google introduced Google App Engine (GAE), a platform for scalable web applications used by many smaller cloud providers.
- GAE runs applications on Google's extensive data center network linked to its search infrastructure.
- Key Google cloud components:

- GFS for storing large data sets,
- MapReduce for developing applications,
- Chubby for managing distributed locks,
- BigTable for structured and semi-structured data storage.

- Google's data centers have thousands of servers organized into clusters running these services.
- GAE hosts third-party applications, so developers don't manage servers directly.
- GAE's frontend is an application framework like ASP or J2EE, supporting Python and Java.

4.4.2.2 GAE Architecture

- The Google cloud platform's main components include Google File System (GFS) for large data storage, MapReduce for application development, Chubby for distributed lock services, and BigTable for structured data storage.
- These technologies are used together inside Google data centers, which contain thousands of servers organized into clusters.
- Users access Google applications through web interfaces; third-party developers can build cloud apps using Google App Engine (GAE).
- Google's core infrastructure is private and not open for external service building.
- GAE runs third-party applications on Google's infrastructure, removing the need for developers to manage servers.
- GAE combines several software components, with a frontend framework similar to ASP, J2EE, or JSP.
- GAE supports Python and Java environments, functioning like web application containers and providing full web technology support.

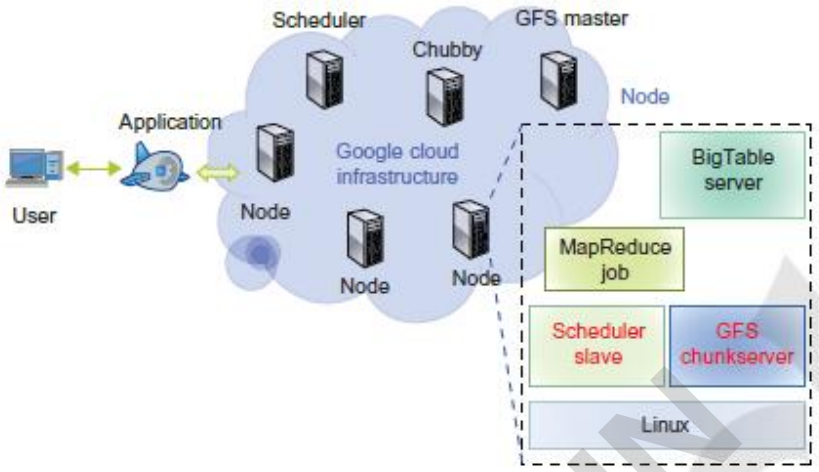

FIGURE 4.20

Google cloud platform and major building blocks, the blocks shown are large clusters of low-cost servers.

4.4.2.3 Functional Modules of GAE

- GAE helps you build apps but doesn't give you control of the actual servers.
- It supports only Python, Ruby, and Java programming languages.
- You can use GAE for free with limits, and pay if you use more.
- Google takes care of all the hard server stuff for you.

1. **Datastore**: Stores data safely using Google's BigTable system.
2. **App Runtime**: Lets you build and run web apps using Python or Java.
3. **SDK**: A tool you use to create and test apps on your own computer before uploading them.
4. **Admin Console**: Helps you manage your apps easily.
5. **Web Service Infrastructure**: Manages storage and network smoothly for your apps.

4.4.2.4 GAE Applications

- Popular Google apps like Search, Docs, Earth, and Gmail run on GAE and support many users at once.
- Users access these apps through web browsers.
- Third-party developers can use GAE to build their own cloud apps.
- These apps run on thousands of servers inside Google's data centers.
- GAE offers storage services for apps to save data securely and perform database-like operations (queries, sorting, transactions).

- It also provides Google services like Gmail login, so developers don't have to build their own user management.
- Apps on GAE can easily authenticate users and send emails using Google accounts.

4.4.3 Amazon Web Services (AWS)

- Amazon AWS offers public cloud services mainly using the Infrastructure-as-a-Service (IaaS) model.
- EC2 provides virtual machines (VMs) where cloud apps run.
- S3 is Amazon's object storage service for storing data like files.
- EBS offers block storage that works like a hard drive for traditional apps.
- SQS (Simple Queue Service) ensures reliable message delivery between processes, even if one is offline.

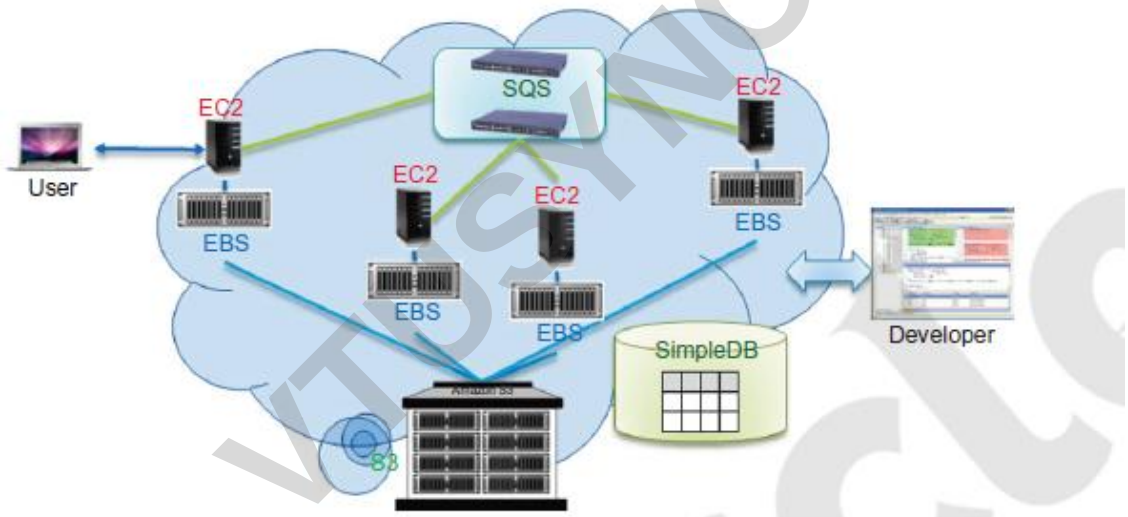

FIGURE 4.21
Amazon cloud computing infrastructure (Key services are identified here; many more are listed in Table 4.6).

- Users access AWS services via standard protocols like SOAP through browsers or client programs.
- AWS offers many services across different application areas (12 tracks summarized in a table).
- AWS also provides SQS and SNS for messaging and notifications.
- ELB (Elastic Load Balancer) distributes incoming app traffic across EC2 instances to balance load and avoid failing servers.

- Auto Scaling and ELB are managed by CloudWatch, which monitors resource usage and performance (CPU, disk, network, etc.).
- AWS offers a Relational Database Service (RDS) similar to Azure's.
- Elastic MapReduce is AWS's big data service, similar to Hadoop, running on EC2.
- AWS Import/Export lets users physically ship disks to transfer huge data volumes quickly between distant locations.
- Amazon CloudFront is a content delivery network to distribute content efficiently worldwide.
- AWS is flexible and popular with small and medium businesses to build and run cloud applications, supporting large user bases and paid services.

4.4.4 Microsoft Windows Azure

- Microsoft launched Windows Azure in 2008 as a cloud platform built on Windows OS and Microsoft virtualization.
- Azure runs applications on virtual machines (VMs) hosted in Microsoft data centers.
- Azure manages all data center resources: servers, storage, and networks.
- The platform has three main components and provides various cloud-level services:
- Live Service: lets users access Microsoft Live apps and work on data across multiple machines.
- .NET Service: supports app development locally and execution on the cloud.
- SQL Azure: cloud-based relational database service using Microsoft SQL Server.
- SharePoint Service: platform to build scalable business web applications.
- Dynamic CRM Service: platform to build and manage customer relationship management (CRM) apps for finance, marketing, sales, etc.
- Azure services integrate well with other Microsoft apps like Windows Live, Office Live, Exchange Online, SharePoint Online, and Dynamic CRM Online.
- It uses standard web protocols like SOAP and REST for communication.
- Users can integrate Azure cloud apps with other platforms and third-party clouds.
- The Azure SDK allows developers to build and test Azure apps locally on Windows before deploying to the cloud.

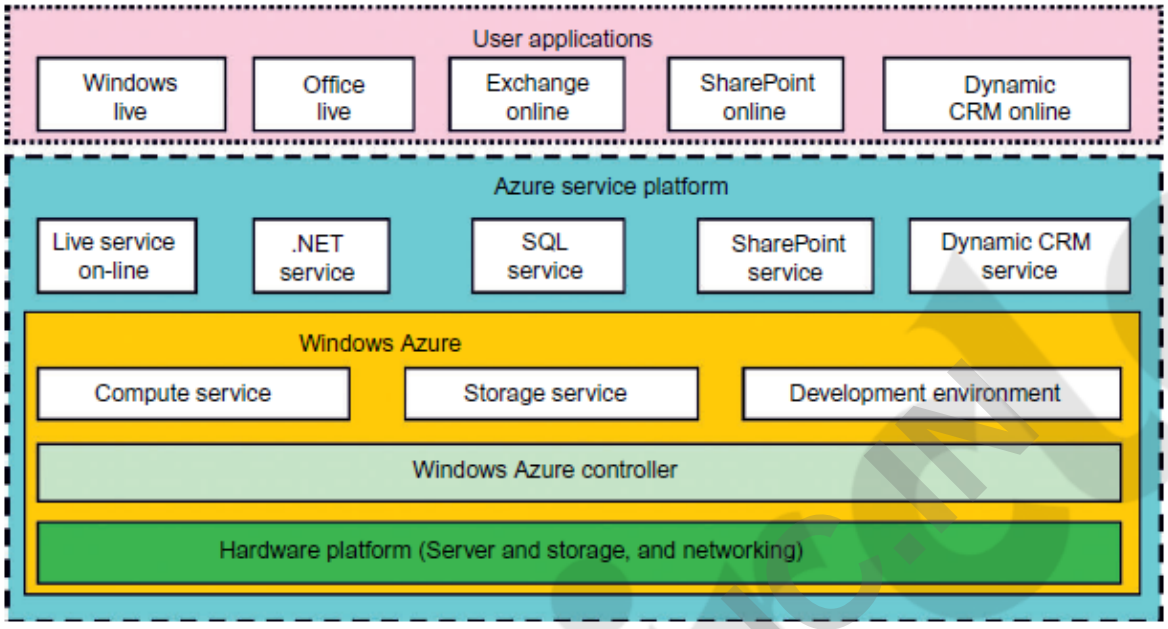

FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

4.5 INTER-CLOUD RESOURCE MANAGEMENT

4.5.1 Extended Cloud Computing Services

- **Six layers of cloud services (from bottom to top):**
  - **Hardware as a Service (HaaS):** Physical hardware resources like servers.
  - **Network as a Service (NaaS):** Networking components connecting hardware (includes Virtual LANs).
  - **Location as a Service (LaaS):** Data center space, power, and security for hardware; sometimes called Security as a Service.
  - **Infrastructure as a Service (IaaS):** Compute, storage, and communication resources. Can be divided further into Data as a Service (DaaS) and Communication as a Service (CaaS).
  - **Platform as a Service (PaaS):** Platforms for developing and running applications, built on IaaS.
  - **Software as a Service (SaaS):** Software applications delivered over the cloud, built on PaaS.
- **Layer dependency:**
  - SaaS depends on PaaS, which depends on IaaS, which depends on the lower physical layers. You cannot run SaaS without having the underlying infrastructure in place.

- **Cloud players divided into three groups:**
  - Cloud service providers and IT admins.
  - Software developers or vendors.
  - End users or business users.

- **Different views on cloud models:**
  - Software vendors care most about application performance on the cloud platform.
  - Providers focus on the cloud infrastructure itself.

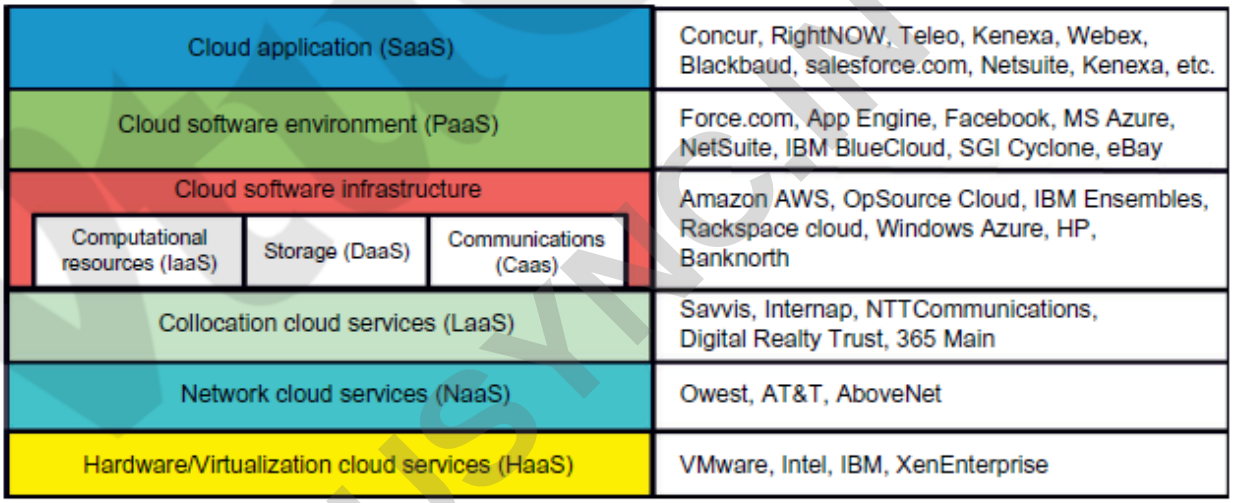

FIGURE 4.23
A stack of six layers of cloud services and their providers.

4.5.1.1 Cloud Service Tasks and Trends

The five layers of cloud services and some examples:

1. **Five layers of cloud services:**
   - **SaaS (Software as a Service):** Top layer with business apps like CRM (Customer Relationship Management), collaboration tools, finance, and HR management. CRM was the first successful SaaS.
   - **PaaS (Platform as a Service):** Platforms to develop apps, offered by Google, Salesforce.com, Facebook, etc.
   - **IaaS (Infrastructure as a Service):** Cloud infrastructure like servers and storage, offered by Amazon, Windows Azure, Rackspace, etc.
   - **Collocation services:** Physical hosting and support across multiple cloud providers, often for manufacturing supply chains.

- **Network cloud services:** Communication services provided by companies like AT&T, Qwest, AboveNet.

2. **Vertical cloud services:**
   - These are connected, mutually supportive cloud services often combined together (called cloud mashups) for specific industries or workflows.
3. **Cloud service growth:**
   - Cloud services, especially SaaS for business applications, are rapidly growing.

4.5.1.2 Software Stack for Cloud Computing

- Cloud software is built in layers, each with a specific job.
- Lower layers like the OS are often customized for cloud needs.
- Cloud systems must be fast, always available, and handle failures well.
- Cloud platforms run on physical servers or virtual machines (VMs).
- VMs make cloud platforms flexible and not tied to specific hardware.
- The bottom layer stores huge amounts of data like a file system.
- Above storage are layers for databases, programming, and data queries.
- Each layer talks to the one above to help build cloud applications.

4.5.1.3 Runtime Support Services

- Cloud clusters use **cluster monitoring** to check the status of all nodes.
- A **scheduler** manages and assigns tasks to nodes based on availability.
- Cloud schedulers support special programming styles like **MapReduce**.
- **Runtime support software** helps keep the cloud system running smoothly and efficiently.
- In the **SaaS model**, users access software via browsers without buying or installing it.
- SaaS users don’t need to invest in servers or software licenses upfront.
- Cloud providers save costs compared to traditional hosting.
- User data is stored in cloud servers, which may be private or public clouds.