UNIT – 3

CLOUD PLATFORM ARCHITECTURE:-

Cloud Computing and service Models, Public Cloud Platforms, Service Oriented Architecture, Programming on Amazon AWS and Microsoft Azure, Programming on Amazon AWS and Microsoft Azure.

1. CLOUD COMPUTING AND SERVICE MODELS:

- Over the past two decades, the world economy has rapidly moved from manufacturing to more service-oriented.
- Cloud computing benefits the service industry most and advances business computing with a new paradigm.
- Developers of innovative cloud applications no longer acquire large capital equipment in advance. They just rent the resources from some large datacenters that have been automated for this purpose.

1.1 PUBLIC CLOUD:

- A public cloud is built over the Internet and can be accessed by any user who has paid for the service. Public clouds are owned by service providers and are accessible through a subscription.
- The providers of the aforementioned clouds are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure.
- A public cloud delivers a selected set of business processes. The application and infrastructure services are offered on a flexible price-per-use basis.

Examples:

- Google App Engine (GAE)
- Amazon Web Services (AWS)
- Microsoft Azure
- IBM Blue Cloud
- Salesforce.com's Force.com.

Advantages:

- Standardization
- Preserves Capital Investment
- Offers Application Flexibility

1.2 PRIVATE CLOUD:

- A private cloud is built within the domain of an intranet owned by a single organization. It is client owned and managed, and its access is limited to the owning clients and their partners.

- Its deployment was not meant to sell capacity over the Internet through publicly accessible interfaces.
- Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains.
- A private cloud is supposed to deliver more efficient and convenient cloud services. It may impact the cloud standardization, while retaining greater customization and organizational control.

Examples:

- IBM RC2
- Amazon Virtual Private Cloud
- VMware Private Cloud
- Rackspace Private Cloud (Powered by OpenStack)
- CloudBees

Advantages:

- Customization & offers higher efficiency
- Resiliency
- Security
- Privacy

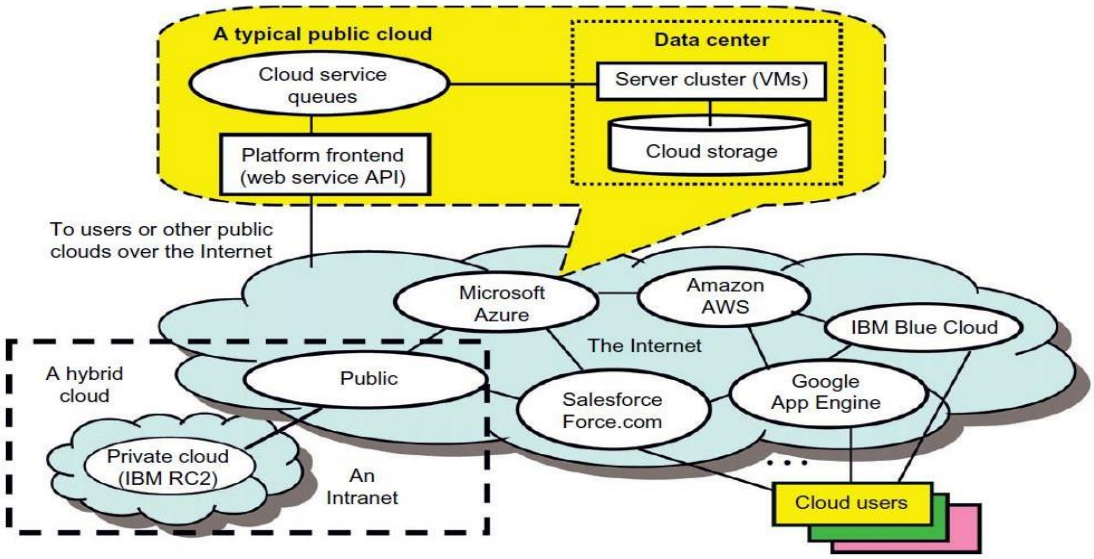


FIGURE 4.1

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

1.3 HYBRID CLOUDS:

- A hybrid cloud is built with both public and private clouds Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud.

- A hybrid cloud provides access to clients, the partner network, and third parties. Hybrid clouds operate in the middle, with many compromises in terms of resource sharing.

Example:

- Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, Europe, and Asia.

1.4 CLOUD SERVICES:

1.4.1 Infrastructure as a Service(IaaS):

- This model allows users to use virtualized IT resources for computing, storage, and networking. The service is performed by rented cloud infrastructure.
- The user can deploy and run his applications over his chosen OS environment. The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components.
- This IaaS model encompasses storage as a service, compute instances as a service, and communication as a service.
- Many startup cloud providers have appeared in recent years. GoGrid, FlexiScale, and Aneka are good examples.

Table 4.1 Public Cloud Offerings of IaaS [10,18]

Cloud Name	VM Instance Capacity	API and Access Tools	Hypervisor, Guest OS
Amazon EC2	Each instance has 1–20 EC2 processors, 1.7–15 GB of memory, and 160–1.69 TB of storage.	CLI or web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30–480 GB of storage.	REST, Java, PHP, Python, Ruby	Xen, Linux, Windows
Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10–620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
FlexiScale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20–270 GB of storage.	web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30–480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, OpenSolaris

1.4.2 .Platform as a Service (PaaS):

- To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment.

- Such a platform includes operating system and runtime library support. This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications.
- The platform cloud is an integrated computer system consisting of both hardware and software infrastructure.
- The user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET). The user does not manage the underlying cloud infrastructure. The cloud provider supports user application

Table 4.2 Five Public Cloud Offerings of PaaS [10,18]

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

1.4.3 Software as a Service (SaaS):

- This refers to browser-initiated application software over thousands of cloud customers. Services and tools offered by PaaS are utilized in construction of applications and management of their deployment on resources offered by IaaS providers.
- The SaaS model provides software applications as a service. As a result, on the customer side, there is no upfront investment in servers or software licensing.
- On the provider side, costs are kept rather low, compared with conventional hosting of user applications. Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS.

Examples of SaaS:

- Google Gmail and docs
- Microsoft SharePointa
- CRM software from Salesforce.com.

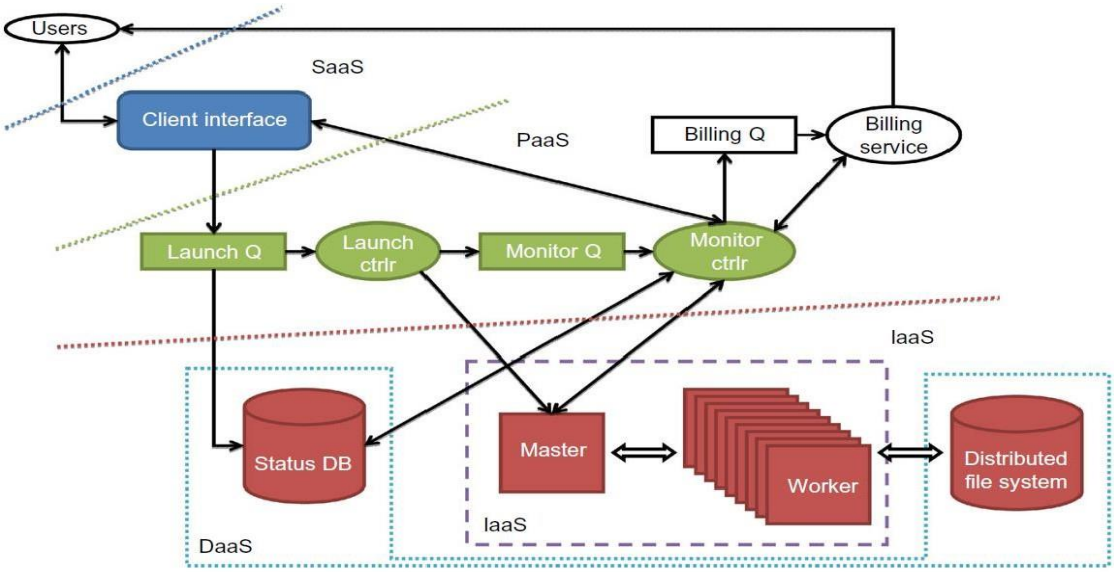


FIGURE 4.5

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

1.5 CLOUD ECOSYSTEM AND ENABLING TECHNOLOGIES:

Classical Computing	**Cloud Computing**
(Repeat the following cycle every 18 months)	(Pay as you go per each service provided)
Buy and own	**Subscribe**
Hardware, system software, applications to meet peak needs	- - - -
Install, configure, test, verify, evaluate, manage	**Use** (Save about 80-95% of the total cost)
- - - -	- - - -
Use	(Finally)
- - - -	**$ - Pay for what you use**
Pay $$$$$ (High cost)	based on the QoS

1.6 CLOUD DESIGN OBJECTIVES:

The following list highlights six design objectives for cloud computing:

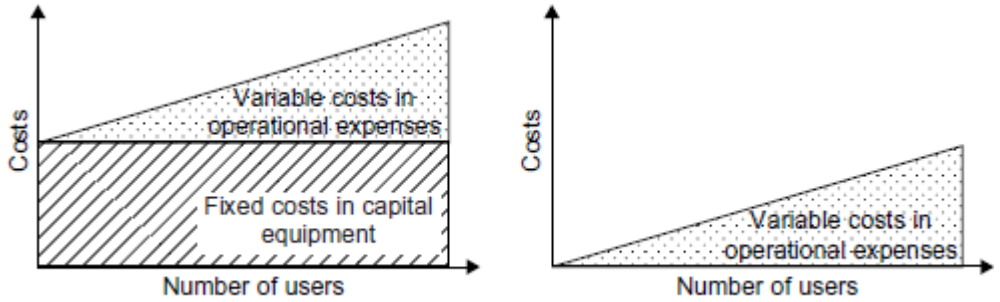
- Shifting computing from desktops to data centers Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- Service provisioning and cloud economics Providers supply cloud services by signing SLAs with consumers and end users. The services must be efficient in terms of computing, storage,and power consumption. Pricing is based on a pay-as-you-go policy.
- Scalability in performance The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.

- Data privacy protection Can you trust data centers to handle your private data and records? This concern must be addressed to make clouds successful as trusted services.
- High quality of cloud services The QoS of cloud computing must be standardized to make clouds interoperable among multiple providers.
- New standards and interfaces This refers to solving the data lock-in problem associated with data centers or cloud providers. Universally accepted APIs and access protocols are needed to provide high portability and flexibility of virtualized applications

1.7 COST MODEL:

- ✓ In traditional IT computing, users must acquire their own computer and peripheral equipment as capital expenses. In addition, they have to face operational expenditures in operating and maintaining the computer systems, including personnel and service costs. The addition of variable operational costs on top of fixed capital investments in traditional IT. The fixed cost is the main cost, and that it could be reduced slightly as the number of users increases.The operational costs may increase sharply with a larger number of users. Therefore, the total cost escalates quickly with massive numbers of users.

- ✓ Cloud computing applies a pay-per-use business model, in which user jobs are outsourced to data centers. To use the cloud, one has no up-front cost in hardware acquisitions. Only variable costs are experienced by cloud users, Overall, cloud computing will reduce computing costs significantly for both small users and large enterprises. Computing economics does show a big gap between traditional IT users and cloud users. The savings in acquiring expensive computers up front releases a lot of burden for startup companies.

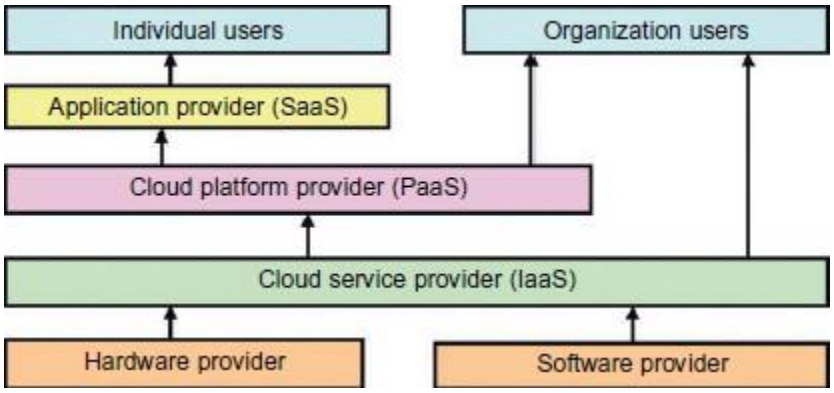


(a) Traditional IT cost model (b) Cloud computing cost model

2. PUBLIC CLOUD PLATFORMS: GAE, AWS, AND AZURE:-

- ➢ Cloud services are demanded by computing and IT administrators, software vendors, and end users. We have five levels of cloud players.

- At the top level, individual users and organizational users demand very different services. The application providers at the SaaS level serve mainly individual users. Most business organizations are serviced by IaaS and PaaS providers.
- The infrastructure services (IaaS) provide compute, storage, and communication resources to both applications and organizational users. The cloud environment is defined by the PaaS or platform providers. Note that the platform providers support both infrastructure services and organizational users directly.
- Cloud services rely on new advances in machine virtualization, SOA, grid infrastructure management, and power efficiency. Consumers purchase such services in the form of IaaS, PaaS, or SaaS as described earlier.
- Many cloud entrepreneurs are selling value-added utility services to massive numbers of users. The cloud industry leverages the growing demand by many enterprises and business users to outsource their computing and storage jobs to professional providers. The provider service charges are often much lower than the cost for users to replace their obsolete servers frequently.



2.1 GOOGLE APP ENGINE (GAE)

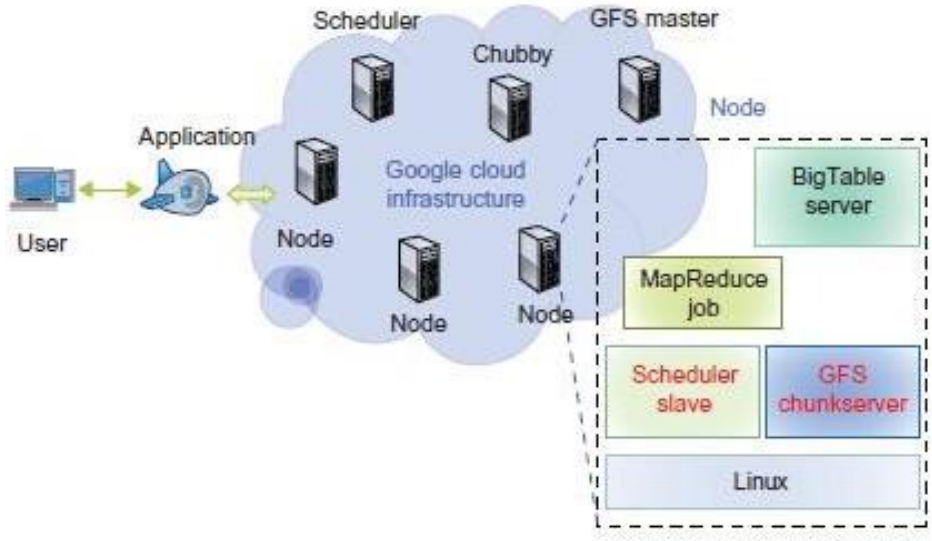
- Google has the world's largest search engine facilities. The company has extensive experience in massive data processing that has led to new insights into data-center design and novelprogramming models that scale to incredible sizes.
- The Google platform is based on its search engine expertise, but as discussed earlier with MapReduce, this infrastructure is applicable to many other areas.
- Google has hundreds of data centers and has installed more than 460,000 servers worldwide.

- Google Cloud Infrastructure. Google has pioneered cloud development by leveraging the large number of data centers it operates.
- Google pioneered cloud services in Gmail, Google Docs, and Google Earth, among other applications. These applications can support a large number of users

simultaneously with HA. Notable technology achievements include the Google File System (GFS), MapReduce, BigTable, and Chubby

2.2 GAE ARCHITECTURE:

- The major building blocks of the Google cloud platform are mentioned below and with these building blocks, Google has built many cloud applications
- The building blocks of Google's cloud computing application include the Google File System for storing large amounts of data, the MapReduce programming framework for application developers, Chubby for distributed application lock services, and BigTable as a storage service for accessing structural or semistructural data.



- Users can interact with Google applications via the web interface provided by each application. Third-party application providers can use GAE to build cloud applications for providing services. The applications all run in data centers under tight management by Google engineers.

2.3 FUNCTIONAL MODULES OF GAE

The GAE platform comprises the following five major components. The GAE is not an infrastructure platform, but rather an application development platform for users.

1. The **datastore** offers object-oriented, distributed, structured data storage services based on BigTable techniques. The datastore secures data management operations.
2. The **application runtime environment** offers a platform for scalable web programming

andexecution. It supports two development languages: Python and Java.

3. The **software development kit** (SDK) is used for local application development. The SDK allows users to execute test runs of local applications and upload application code.
4. The **administration console** is used for easy management of user application development cycles, instead of for physical resource management.
5. The **GAE web service infrastructure** provides special interfaces to guarantee flexible use and management of storage and network resources by GAE

2.4 GAE APPLICATIONS

Well-known GAE applications are

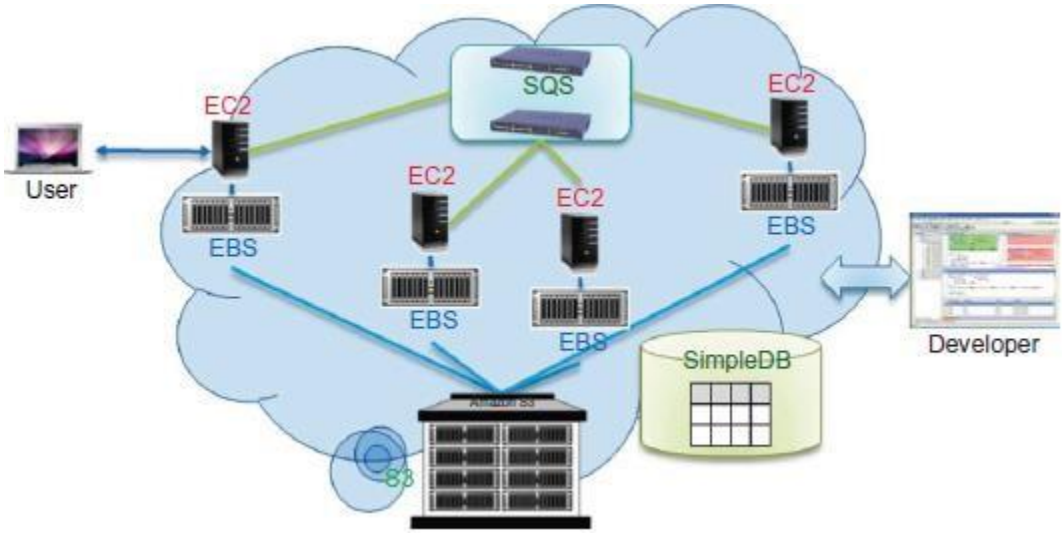
- ✓ Google Search Engine
- ✓ Google Docs
- ✓ Google Earth
- ✓ Gmail.

- These applications can support large numbers of users simultaneously. Users can interact with Google applications via the web interface provided by each application. Third-party application providers can use GAE to build cloud applications for providing services.
- The applications are all run in the Google data centers. Inside each data center, there might be thousands of server nodes to form different clusters. Each cluster can run multipurpose servers.
- GAE supports many web applications. One is a storage service to store application-specific data in the Google infrastructure.
- The data can be persistently stored in the backend storage server while still providing the facility for queries, sorting, and even transactions similar to traditional database systems.
- GAE also provides Google-specific services, such as the Gmail account service (which is the login service, that is, applications can use the Gmail account directly).
- This can eliminate the tedious work of building customized user management components in web applications. Web applications built on top of GAE can use the APIs authenticating users and sending e-mail using Google accounts.

2.5 AMAZON WEB SERVICES (AWS)

Amazon has been a leader in providing public cloud services. Amazon applies the IaaS model in providing its services. The following figure shows the AWS architecture.

1. EC2 provides the virtualized platforms to the host VMs where the cloud application can run.
2. S3 (Simple Storage Service) provides the object-oriented storage service for users.
3. EBS (Elastic Block Service) provides the block storageinterface which can be used to support traditional applications.



- SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes. The message can be kept reliably even when the receiver processes are not running.
- Users can access their objects through SOAP with either browsers or other client programs which support the SOAP standard.
- Different from Google, Amazon provides a more flexible cloud computing platform for developers to build cloud applications. Small and medium-size companies can put their business on the Amazon cloud platform.
- Using the AWS platform, they can service large numbers of Internet users and make profits through those paid services. ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances and allows user to avoid nonoperating nodes and to equalize load on functioning images.
- Both autoscalingand ELB are enabled by CloudWatch which monitors running instances. CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2.
- It provides customers with visibility into resource utilization, operational performance, and overall demand patterns, including metrics such as CPU utilization, disk reads and writes, and network traffic. Amazon (like Azure) offers a Relational Database Service (RDS) with a messaging interface. The Elastic MapReduce capability is equivalent to Hadoop running on the basic EC2 offering.
- AWS Import/Export allows one to ship large volumes of data to and from EC2 by shipping physical disks; it is well known that this is often the highest bandwidth connection between geographically distant systems. Amazon CloudFront implements a content distribution

Table 4.6 AWS Offerings in 2011

Service Area	Service Modules and Abbreviated Names
Compute	Elastic Compute Cloud (EC2), Elastic MapReduce, Auto Scaling
Messaging	Simple Queue Service (SQS), Simple Notification Service (SNS)
Storage	Simple Storage Service (S3), Elastic Block Storage (EBS), AWS Import/Export
Content Delivery	Amazon CloudFront
Monitoring	Amazon CloudWatch
Support	AWS Premium Support
Database	Amazon SimpleDB, Relational Database Service (RDS)
Networking	Virtual Private Cloud (VPC) (Example 4.1, Figure 4.6), Elastic Load Balancing
Web Traffic	Alexa Web Information Service, Alexa Web Sites
E-Commerce	Fulfillment Web Service (FWS)
Payments and Billing	Flexible Payments Service (FPS), Amazon DevPay
Workforce	Amazon Mechanical Turk

(Courtesy of Amazon, http://aws.amazon.com [3])

2.6 MICROSOFT WINDOWS AZURE

- In 2008, Microsoft launched a Windows Azure platform to meet the challenges in cloud computing. This platform is built over Microsoft data centers.
- The following figure shows the overall architecture of Microsoft's cloud platform. The platform is divided into three major component platforms.
- Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology.
- Applications are installed on VMs deployed on the data-center servers. Azure manages all servers, storage, and network resources of the data center. On top of the infrastructure are the various services for building different cloud applications. Cloud-level services provided by the Azure platform are

  1. **Live service**: Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.

  2. **.NET service:** This package supports application development on local hosts and execution on cloud machines.

  3. **SQL Azure** : This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.

  4. **SharePoint service:** This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.

  5. **Dynamic CRM service:** This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

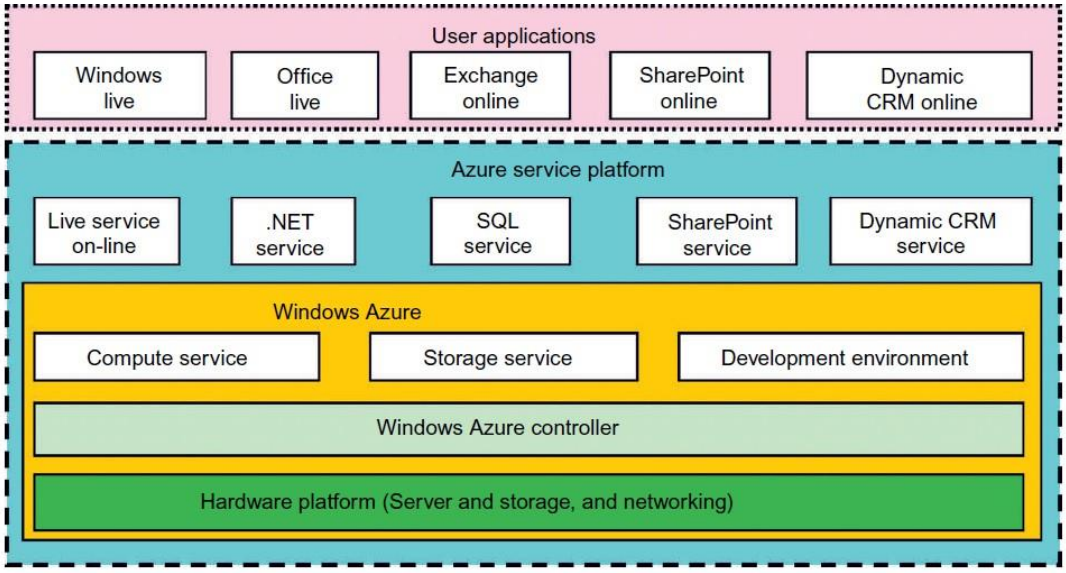


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

- All these cloud services in Azure can interact with traditional Microsoft software applications, such as Windows Live, Office Live, Exchange online, SharePoint online, and dynamic CRM online.
- The Azure platform applies the standard web communication protocols SOAP and REST. The Azure service applications allow users to integrate the cloud application with other platforms or third-party clouds.

SERVICES AND SERVICE-ORIENTED ARCHITECTURE(SOA):-

In general, SOA is about how to design a software system that makes use of services of new or legacy applications through their published or discoverable interfaces. These applications are often distributed over the networks. SOA also aims to make service interoperability extensible and effective. It prompts architecture styles such as loose coupling, published interfaces, and a standard communication model in order to support this goal.

- The World Wide Web Consortium (W3C) defines SOA as a form of distributed systems architecture characterized by the following properties:

1. **Logical view**: The SOA is an abstracted, logical view of actual programs, databases, business processes, and so on, defined in terms of what it does, typically carrying out a business-level operation. The service is formally defined in terms of the messages exchanged between provider agents and requester agents.
2. **Message orientation:** The internal structure of providers and requesters include the implementation language, process structure, and even database structure. These

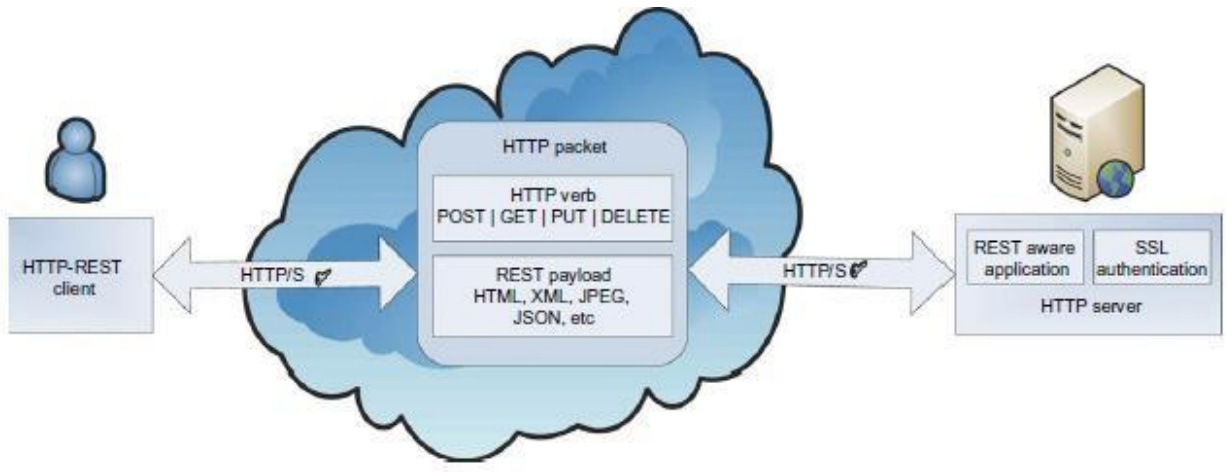
features are deliberately abstracted away in the SOA: Using the SOA discipline one does not and should not need to know how an agent implementing a service is constructed. A key benefit of this concerns legacy systems. By avoiding any knowledge of the internal structure of an agent, one can incorporate any software component or application to adhere to the formal service definition.

3. **Description orientation:** A service is described by machine-executable metadata. The description supports the public nature of the SOA: Only those details that are exposed to the public and are important for the use of the service should be included in the description. The semantics of a service should be documented, either directly or indirectly, by its description.
   - ✓ **Granularity** Services tend to use a small number of operations with relatively large and complex messages.
   - ✓ **Network orientation** Services tend to be oriented toward use over a network, though this is not an absolute requirement.
   - ✓ **Platform-neutral** Messages are sent in a platform-neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.

2.7 REST AND SYSTEMS OF SYSTEMS:

REST is a software architecture style for distributed systems, particularly distributed hypermedia systems, such as the World Wide Web. It has recently gained popularity among enterprises such as Google, Amazon, Yahoo!, and especially social networks such as Facebook and Twitter because of its simplicity, and its ease of being published and consumed by clients.

2.8 REST ARCHITECTURE:



The REST architectural style is based on four principles:

1. **Resource Identification through URIs:**
   - ✓ The RESTful web service exposes a set of resources which identify targets of

interaction with its clients. The key abstraction of information in RESTis a resource.

- ✓ Any information that can be named can be a resource, such as a document or image or a temporal service. A resource is a conceptual mapping to a set of entities. Each particular resource is identified by a unique name, or more precisely, a Uniform Resource Identifier (URI) which is of type URL, providing a

global addressing space for resources involved in an interaction between components as well as facilitating service discovery.

2. **Uniform, Constrained Interface:** Interaction with RESTful web services is done via the HTTP standard, client/server cacheable protocol. Resources are manipulated using a fixed set of four CRUD (create, read, update, delete) verbs or operations:
   - ✓ PUT - creates a new resource
   - ✓ GET - retrieves the current state of a resource
   - ✓ POST - transfers a new state onto a resource
   - ✓ DELETE – destroys a resource which was created by using PUT.
3. **Self-Descriptive Message:**A REST message includes enough information to describe how to process the message. This enables intermediaries to do more with the message without parsing the message contents. In REST, resources are decoupled from their representation so that their content can be accessed in a variety of standard formats (e.g., HTML, XML, MIME, plain text, PDF, JPEG, JSON, etc.). REST provides multiple/alternate representations of each resource. Metadata about the resource is available and can be used for various purposes, such as cache control, transmission error detection, authentication or authorization, and access control.
4. **Stateless Interactions:**The REST interactions are "stateless" in the sense that the meaning of a message does not depend on the state of the conversation. Stateless communications improve visibility, since a monitoring system does not have to look beyond a single request data field in order to determine the full nature of the request reliability as it facilitates the task of recovering from partial failures, and increases scalability as discarding state between requests allows the server component to quickly free resources

2.9 SERVICES AND WEB SERVICES:

- In an SOA paradigm, software capabilities are delivered and consumed via loosely coupled, reusable, coarse-grained, discoverable, and self-contained services interacting via a message-based communication model.
- The web has becomes a medium for connecting remote clients with applications for years, and more recently, integrating applications across the Internet has gained in popularity. The term "web service" is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web. Once a web service is deployed, other applications and other web services can discover and invoke the deployed service.
- The W3C working group defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network.

- According to this definition, a web service has an interface described in a machine-executable format (specifically Web Services Description Languageor WSDL).

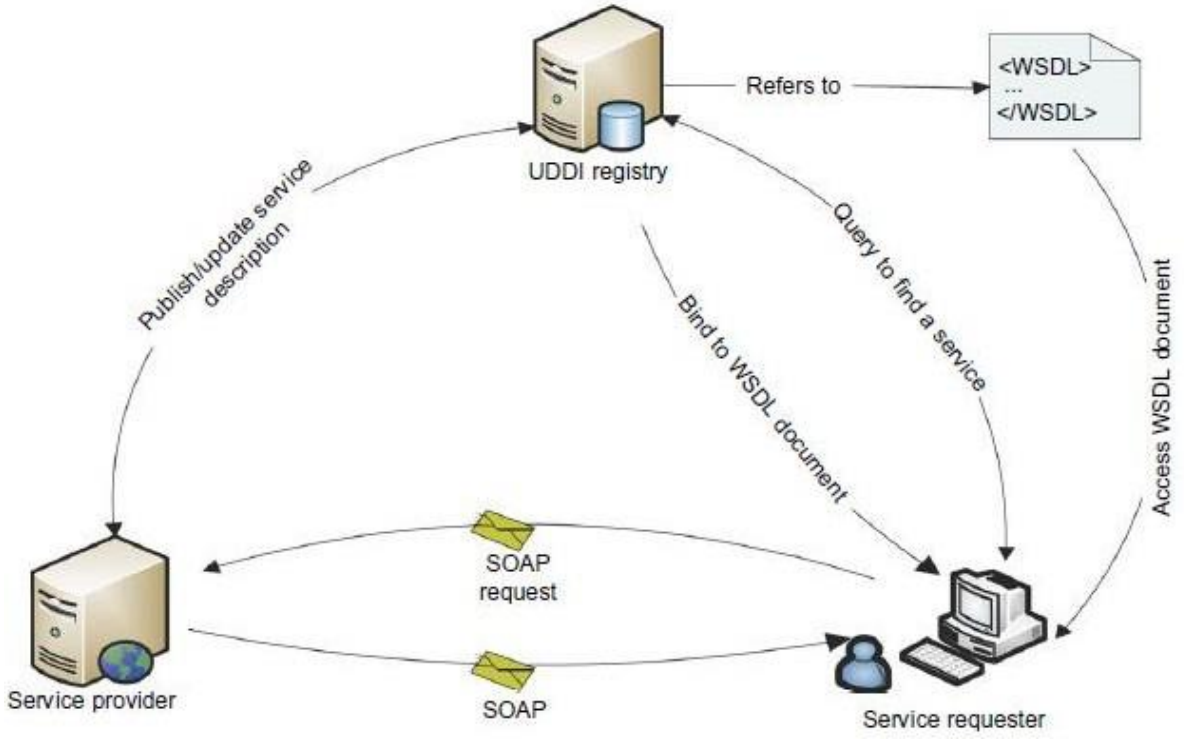
- Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization inconjunction with other web-related standards.

The technologies that make up the core of today's web services are as follows:

1. Simple Object Access Protocol (SOAP):

- SOAP provides a standard packaging structure for transmission of XML documents over various Internet protocols, such as SMTP, HTTP, and FTP. Hence, heterogeneous middleware systems can achieve interoperability.

- A SOAP message consists of a root element called envelope, which contains a header: a container that can be extended by intermediaries with additional application-level elements such as routing information, authentication, transaction management, message parsing instructions, and Quality of Service (QoS) configurations, as well as a body element that carries the payload of the message.
- The content of the payload will be marshaled by the sender's SOAP engine and unmarshaled at the receiver side, based on the XML schema that describes the structure of the SOAP message.



2. **Web Services Description Language (WSDL):**

WSDL describes the interface, a set of operations supported by a web service in a standard format. It standardizes the representation of input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire. Using WSDL enables disparate clients to automatically understand how to interact with a web service.

3. **Universal Description, Discovery, and Integration (UDDI):**

UDDI provides a global registry for advertising and discovery of web services, by searching for names, identifiers, categories, or the specification implemented by the web service.

Programming on Amazon AWS and Microsoft Azure : -

Programming on Amazon Web Services (AWS) and Microsoft Azure involves using various programming languages and tools to build and deploy applications on their respective cloud platforms.

Both Platforms offer a wide range of services and functionalities that developers can leverage to create scalable, secure and reliable cloud-based applications.

Programming on Amazon AWS and Microsoft Azure in cloud computing involves developing and deploying applications on the respective cloud platforms provided by Amazon Web Services (AWS) and Microsoft Azure. This process typically involves using tools and services provided by the cloud providers to create and manage scalable, secure, and reliable applications that can leverage the flexibility and scalability of cloud infrastructure. Developers use programming languages such as Java, Python, C#, and JavaScript to write code that interacts with the cloud services and resources, enabling functions such as data storage, computation, networking, and more. By utilizing the features and functionalities of AWS and Azure, developers can build cutting-edge solutions and meet the increasing demand for cloud-based applications and services.

Azure	AWS
Azure was launched in 2010	AWS was launched in 2006
In the Cloud, For computation, virtual machines are used.	In AWS, For computation, Elastic Compute Cloud is used.
Azure uses blocks to store.	While it uses Simple Storage Service to store.
Azure is a virtual network.	While AWS is a virtual private cloud.
Azure Cloud spans 140 availability zones. (as of Feb 2023).	AWS cloud spans 61 availability zones. (as of Feb 2023).
SQL databases, MySQL, Cosmos DB, etc., are used in Azure for databases.	In AWS for database, RDS and DynamoDB are used.
The pricing model offered by Microsoft is less flexible.	The pricing model offered by AWS is more flexible.
There are four levels of certification in Azure.	AWS has six levels of specialty certifications.
Microsoft Azure has a 22% market share.	Amazon Web Services has a 33% market share
Some famous clients of Azure are: Nike, Dell, Starbucks, etc	Some famous clients of AWS area include: Netflix, Adobe, Spotify, etc

Key aspects of programming on AWS and Azure :-

1) Programming Languages:-

Both Platforms support popular programming languages like Java,Python,C# and Javascript.

2) Cloud Services:-

   Developers use AWS and Azure services for tasks like data storage (E.g, S3 on AWS,Blob Storage on Azure),compute (e.g EC2 on AWS,Virtual Machines on Azure),databases (e.g. RDS on AWS,SQL Database on Azure) and networking.

3) Tools and SDKs:-

   Both platforms provide SDKs (Software Development Kits) and other tools to simplify development and deployment of applications on their respective cloud environments.

4) Scalability and Reliability :-

   Cloud Platforms enable developers to build applications that can scale up or down as needed,and can be designed for high availability and redundancy.

5) Serverless Computing :-

   Both AWS (E.g., Lambda) and Azure (E.g Azure functions) offer serverless computing services,allowing developers to write code that runs only when triggered by an event,without managing underlying infrastructure).

Similarities and differences:-

Similarities:-

Both AWS and Azure offer similar services like compute,storage,databases and networking.

Differences:-

They have different service names and implementation details and their resource organization may differ.

For example, AWS user accounts , while Azure uses subscriptions and resource groups.