Cloud Platforms in Industry



Concepts

I.Amazon Web Services- Compute Services

II.Storage Services

III.Communication Services and Additional Services.

IV.Google App Engine-Architecture and Core Concepts

V.Application Life-Cycle

VI.Cost model.

VII.Microsoft Azure- Azure Core Concepts

VIII.SQL Azure.

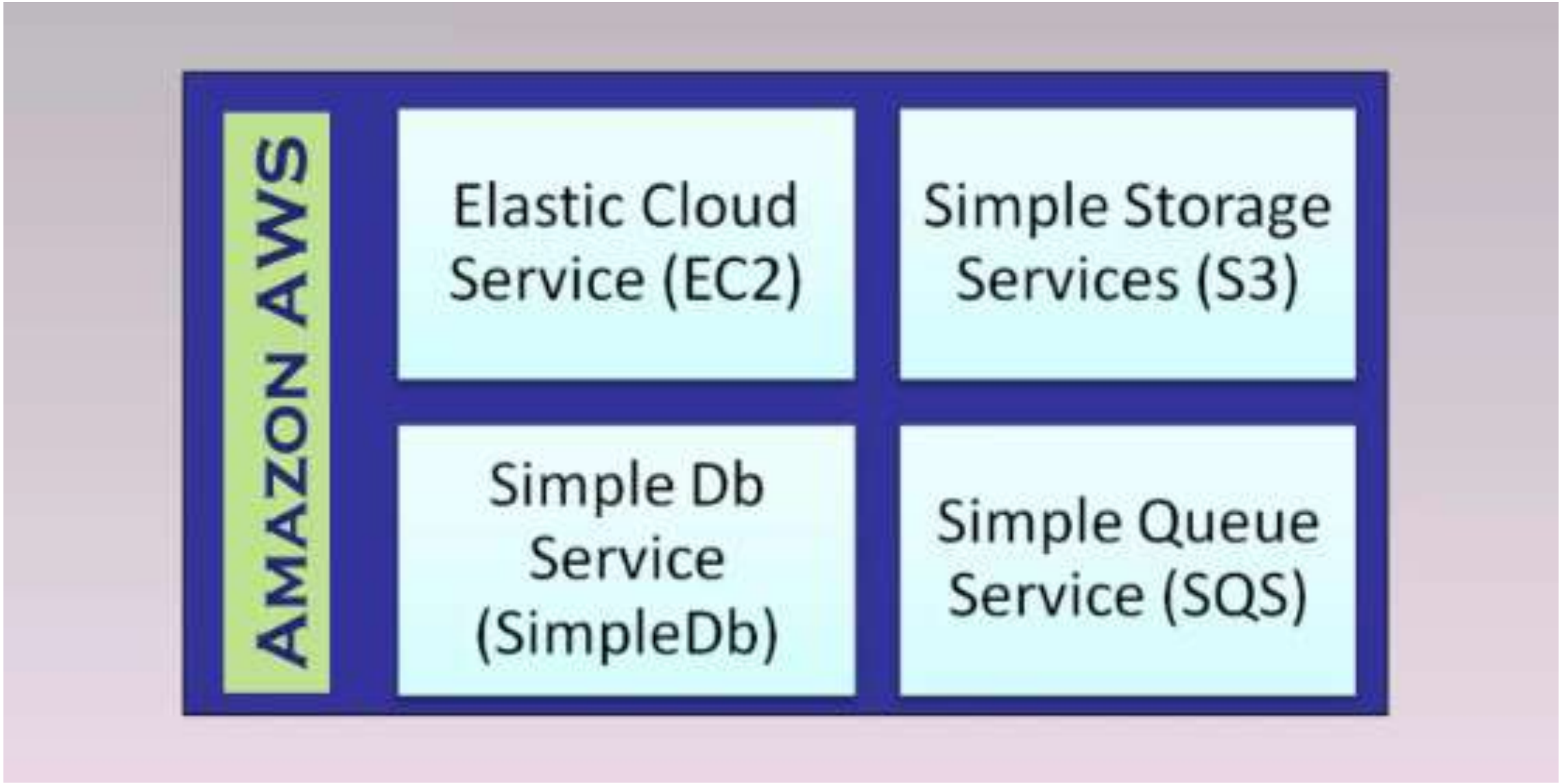
1.Amazon Web Services- Compute Services

- Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, data storage.
- where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

- **Amazon Elastic Compute** (EC2) and **Amazon Simple Storage Service** (S3) are services of AWS.

- The higher level, Elastic MapReduce and AutoScaling provide additional capabilities
- Elastic Block Store (EBS), Amazon SimpleDB, Amazon RDS, and Amazon ElastiCache provide solutions for reliable data snapshots and the management of structured and semistructured data.

AWS Services



Vendor/Product	Service Type	Description
Amazon Web Services	IaaS, PaaS, SaaS	Amazon Web Services (AWS) is a collection of Web services that provides developers with compute, storage, and more advanced services. AWS is mostly popular for IaaS services and primarily for its elastic compute service EC2.
Google AppEngine	PaaS	Google AppEngine is a distributed and scalable runtime for developing scalable Web applications based on Java and Python runtime environments. These are enriched with access to services that simplify the development of applications in a scalable manner.
Microsoft Azure	PaaS	Microsoft Azure is a cloud operating system that provides services for developing scalable applications based on the proprietary Hyper-V virtualization technology and the .NET framework.
SalesForce.com and Force.com	SaaS, PaaS	SalesForce.com is a Software-as-a-Service solution that allows prototyping of CRM applications. It leverages the Force.com platform, which is made available for developing new components and capabilities for CRM applications.
Heroku	PaaS	Heroku is a scalable runtime environment for building applications based on Ruby.
RightScale	IaaS	Rightscale is a cloud management platform with a single dashboard to manage public and hybrid clouds.

Services available in the AWS ecosystem

- At the base of the solution stack are services that provide raw compute and raw storage:

  Amazon Elastic Compute (EC2)

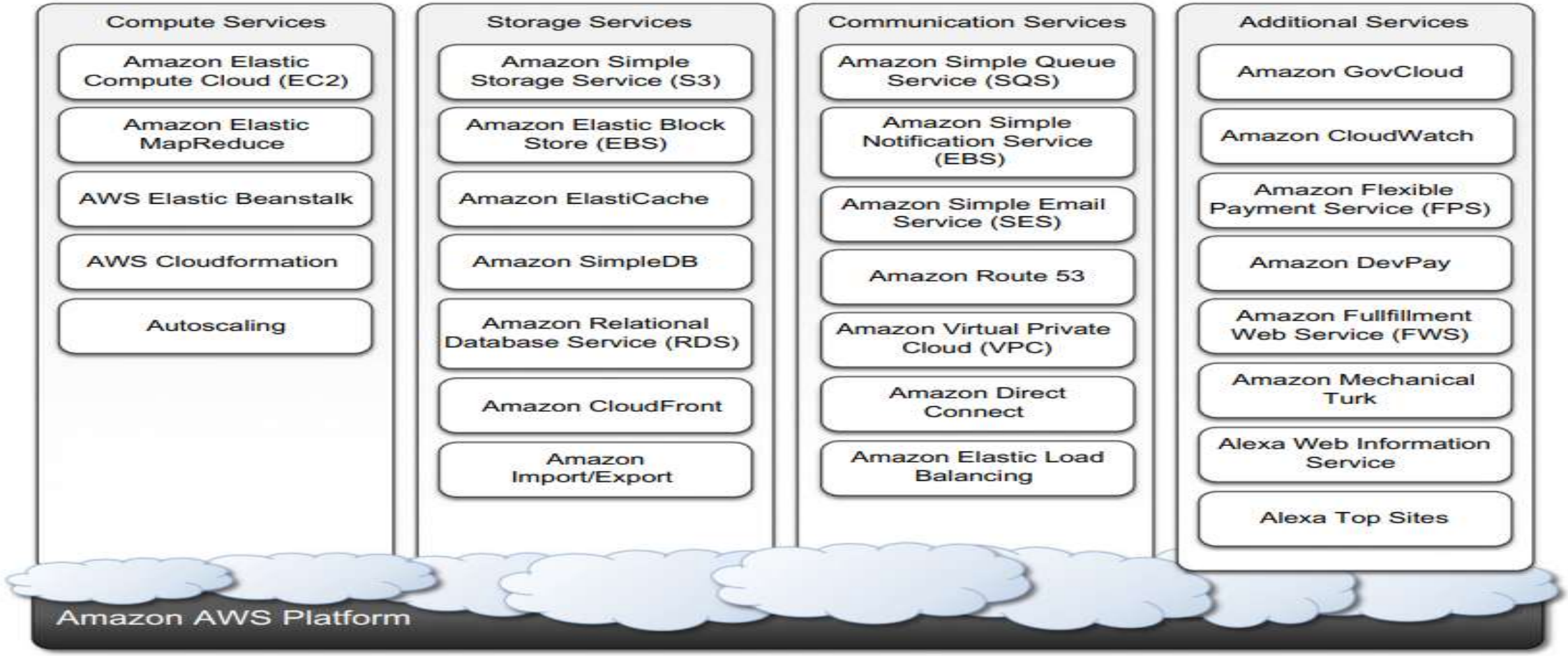
  Amazon Simple Storage Service (S3).

- These are the two most popular services, which are generally complemented with other offerings for building a complete system.
- At the higher level, Elastic MapReduce and AutoScaling provide additional capabilities for building smarter and more elastic computing systems.
- On the data side, Elastic Block Store (EBS), Amazon SimpleDB, Amazon RDS, and Amazon ElastiCache provide solutions for reliable data snapshots.
- Communication needs are covered at the networking level by Amazon Virtual Private Cloud (VPC), Elastic Load Balancing, Amazon Route 53, and Amazon Direct Connect.

Compute Services

Compute services constitute the fundamental element of cloud computing systems. The fundamental service in this space is Amazon EC2, which delivers an IaaS solution that has server

1.Amazon machine images



2.EC2 instances

EC2 provides six different categories of increasing computing power

- **Standard instances**. This class offers a set of configurations that are suitable for most applications such as computing power, storage, and memory.
- **Micro instances**. This class is suitable for those applications that consume a limited amount of computing power and memory ,Micro instances can be used for small Web applications with limited traffic.
- **High-memory instances**. This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile.
- **High-CPU instances**. This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.
- **Cluster Compute instances**. This class is used to provide virtual cluster services. Instances in this category are characterized by high CPU compute power and large memory and an extremely high I/O and network performance,
- **Cluster GPU instances.** This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance.

3.EC2 environment

- The EC2 environment is in charge of allocating addresses, attaching storage volumes, and configuring security in terms of access control and network connectivity.
- **General purpose.** These instances can be used for many different workloads, especially workloads that depend upon a balance of compute, memory and networking resources.
- **Compute-optimized.** These instances are meant for compute-bound or compute-intensive applications, such as batch processing, media transcoding and HPC.
- **Memory-optimized.** This set of instances delivers fast performance for workloads that process large memory data sets.
- **Accelerated computing.** These instances use hardware accelerators to perform certain functions more efficiently compared to CPU software.
- **Storage-optimized.** These instances deliver a high amount of low-latency, random I/O operations per second (IOPS) for workloads that require high sequential read/write access to large local data sets.
- **HPC-optimized.** These instances offer the best price performance for running HPC workloads at scale on AWS, such as complex simulations and deep learning workloads.

4. Advanced compute services

1.AWS CloudFormation

2.Amazon elastic MapReduce

1.**AWS CloudFormation** constitutes an extension of the simple deployment model that characterizes EC2 instances. CloudFormation introduces the concepts of templates. CloudFormation allows easily and explicitly linking EC2 instances.

2.Amazon elastic MapReduce: Amazon Elastic MapReduce provides AWS users with a cloud computing platform for MapReduce applications. It utilizes Hadoop as the MapReduce engine, deployed on a virtual infrastructure composed of EC2 instances

II.Storage Services

- AWS provides a collection of services for data storage and information management.

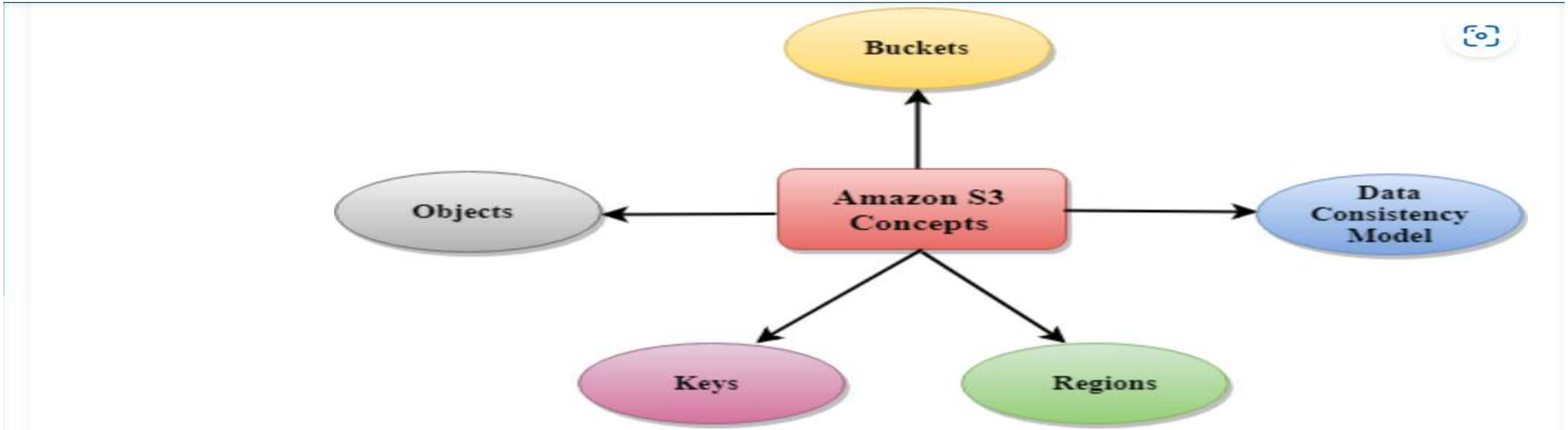
1.S3 key concepts
2.Amazon elastic block store

3.Amazon ElastiCache

4. Structured storage solutions

5 Amazon CloudFront

1.S3 key concepts



- Buckets
- Objects
- Keys
- Regions
- Data Consistency Model

- **Data Storage:** Amazon s3 acts as the best option for scaling both small and large storage applications. It helps in storing and retrieving the Data.

- **Backup and Recovery:** Many Organizations are using Amazon S3 to backup their critical data and maintain the data durability and availability for recovery needs.

- **Hosting Static Websites:** Amazon S3 facilitates in storing HTML, CSS and other web content from Users/developers allowing them for hosting Static Websites benefiting with low-latency access and cost-effectiveness.

- **Data Archiving:** service integration helps as a cost-effective solution for long-term data storing which are less frequently accessed applications.

- **Big Data Analytics:** Amazon S3 is often considered as data lake because of its capacity to store large amounts of both structured and unstructured data offering seamless integration with other AWS Analytics and AWS Machine Learning Services.

2 . Amazon Elastic Block Store (Amazon EBS)

- provides scalable, high-performance block storage resources that can be used with Amazon Elastic Compute Cloud (Amazon EC2) instances. With Amazon Elastic Block Store, you can create and manage the following block storage resources

- **Amazon EBS volumes** — These are storage volumes attached to Amazon EC2 instances. for example, to store files or to install applications.
- **Amazon EBS snapshots** — These are point-in-time backups of Amazon EBS volumes that persist independently from the volume itself. we can create snapshots to back up the data on your Amazon EBS volumes. We can restore new volumes from those snapshots at any time.

3.ElastiCache

- ElastiCache is an implementation of an elastic in-memory cache based on a cluster of EC2 instances.
- It provides fast data access from other EC2 instances through a Memcached-compatible protocol so that existing applications

4.Structured storage solutions

- Amazon provides applications with structured storage services in three different forms: preconfigured EC2 AMIs, Amazon Relational Data Storage (RDS), and Amazon SimpleDB

1.Preconfigured EC2 AMIs

2. Amazon RDS

3. Amazon SimpleDB

1.Preconfigured EC2 AMIs

1.EC2 instances created from these AMIs :can be completed with an EBS volume for storage persistence. Available AMIs include installations of IBM DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Sybase, and Vertica. Instances are priced hourly according to the EC2 cost model. This solution poses most of the administrative burden on the EC2 user,

3.**Amazon RDS** :RDS is relational database service that relies on the EC2 infrastructure and is managed by Amazon. Developers do not have to worry about configuring the storage for high availability.

The second option provides users with increased performance for applications that are heavily based on database reads.

In this case, Amazon deploys copies of the primary service that are only available for database reads, thus cutting down the response time of the service .

4.Amazon SimpleDB Amazon SimpleDB is a lightweight, highly scalable, and flexible data storage solution for applications that do not require a fully relational model for their data. provides support for semistructured data, the model for which is based on the concept of domains, items, and attributes.

5 . Amazon CloudFront

- CloudFront is an implementation of a content delivery network on top of the Amazon distributed storage infrastructure.
- It leverages a collection of edge servers strategically located around the globe to better serve requests for static and streaming Web content so that the transfer time is reduced as much as possible.

III.Communication services

- Amazon provides facilities to structure and facilitate the communication among existing applications and services residing within the AWS infrastructure.
- These facilities can be organized into .two major categories: virtual networking and messaging.

❖**Virtual networking** :Virtual networking comprises a collection of services that allow AWS users to control the connectivity to and between compute and storage services. Amazon Virtual Private Cloud (VPC) and Amazon Direct Connect provide connectivity solutions in terms of infrastructure. Amazon Direct Connect allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports

❖**Messaging services :**constitute the next step in connecting applications by leveraging AWS capabilities. The three different types of messaging services offered are Amazon Simple Queue Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple Email Service (SES).The service provides subscribers with different notification models (HTTP/HTTPS, email/email JSON, and SQS)

IV .Google App Engine-Architecture and Core Concepts

- Is s a cloud computing Platform as a Service (PaaS) which provides Web app developers and businesses with access to Google's scalable hosting data centers and tier 1 Internet service.
- It enables developers to take full advantage of its serverless platform.
- These applications are required to be written in, namely: **Java, Python, PHP, Go, Node.JS, . NET, and Ruby.**
- Applications in the Google App Engine require the use of Google query language and store data in Google Bigtable.



1.Infrastructure

2.Run time environment

3.Storage

4. Application services

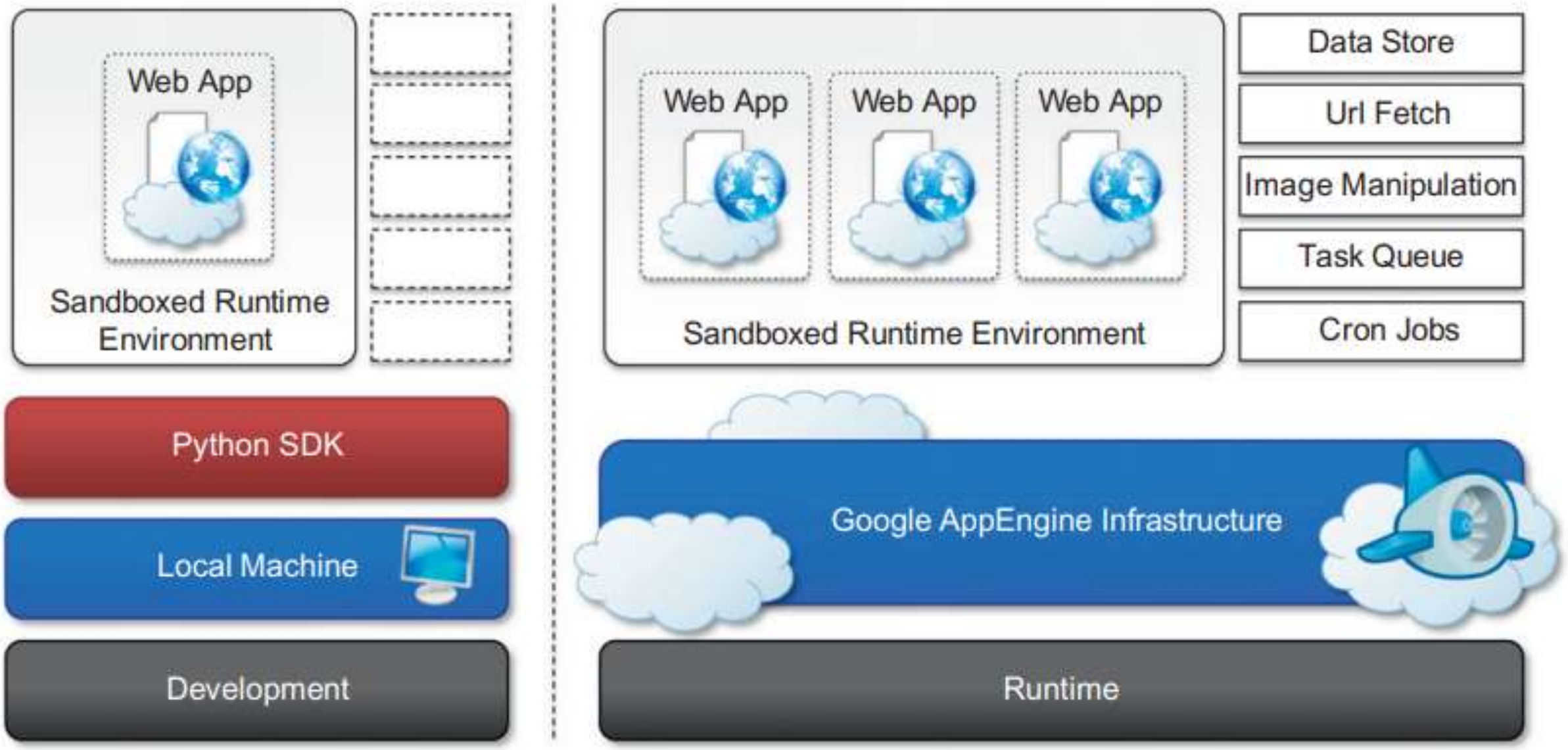
a)MemCache

b)Mail and instant messaging

c) Account management

5.Compute services

1.Infrastructure

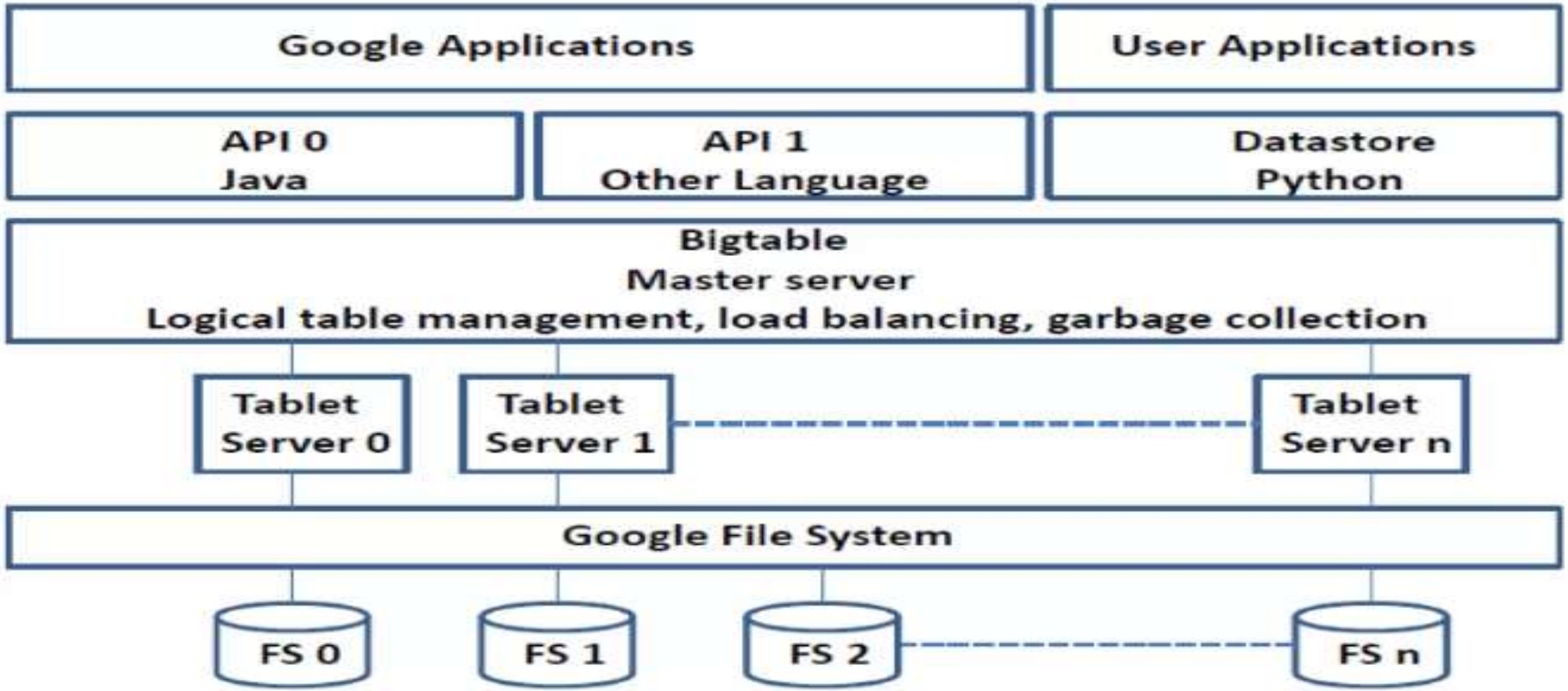


- AppEngine’s infrastructure takes advantage of many servers available within Google datacenters.
- For each HTTP request, AppEngine locates the servers hosting the application that processes the request, evaluates their load, and, if necessary, allocates additional resources (i.e., servers) or redirects the request to an existing server.
- Particular design of applications, which does not expect any state information to be implicitly maintained between requests to the same application.

- The infrastructure is also responsible for monitoring application performance and collecting statistics on which the billing is calculated


Language Runtime
python™
Java
Google App Engine

Google Data Store Architecture



Advantages of Google App Engine

- Infrastructure for Security
- Scalability
- Performance and Reliability
- Cost Savings
- Platform Independence

2.Runtime environment

- **Sandboxing:** One of the major responsibilities of the runtime environment is to provide the application environment with an isolated and protected context in which it can execute without causing a threat to the server and without being influenced by other applications. In other words, it provides applications with a sandbox.
- **Support runtimes** :AppEngine currently supports Java 6, and developers can use the common tools for Web application development in Java, such as the Java Server Pages (JSP), and the applications interact with the environment by using the Java Servlet standard.

3.Data store

- AppEngine provides various types of storage, which operate differently depending on the volatility of the data.
- There are different levels of storage: in memory-cache, storage for semistructured data, and long-term storage for static data
- Data Store is a service that allows developers to store semistructured data. The service is designed to scale and optimized to quickly access data.
- Data Store also provides facilities for creating indexes on data and to update data within the context of a transaction.

4.Application services

- These services simplify most of the common operations that are performed in Web applications: access to data, account management, integration of external resources, messaging and communication, image manipulation, and asynchronous computation.
- **UrlFetch** :Web 2.0 has introduced the concept of composite Web applications. Different resources are put together and organized as meshes within a single Web page. Meshes are fragments of HTML generated in different ways.
- They can be directly obtained from a remote server or rendered from an XML document retrieved from a Web service. Therefore, it is fundamental for Web applications to be able to retrieve remote resources

MemCache AppEngine

- MemCache AppEngine :provides developers with access to fast and reliable storage, which is DataStore.
- Despite this, the main objective of the service is to serve as a scalable and long-term storage, where data are persisted to disk redundantly.
- This design poses a limit on how much faster the store can be compared to other solutions, especially for objects that are frequently accessed
- for example, at each Web request. AppEngine provides caching services by means of MemCache. This is a distributed in

Mail and instant messaging



- Mail and instant messaging Communication is another important aspect of Web applications.
- It is common to use email for following up with users about operations performed by the application.
- Email can also be used to trigger activities in Web applications. To facilitate the implementation of such tasks, AppEngine provides developers with the ability to send and receive mails through Mail.
- The service allows sending email on behalf of the application to specific user accounts.
- It is also possible to include several types of attachments and to target multiple recipients. Mail operates asynchronously,

Account management

- Google account management by means of Google Accounts.
- The integration with the service also allows Web applications to offload the implementation of authentication capabilities to Google's authentication system.
- Using Google Accounts, Web applications can conveniently store profile settings in the form of key-value pairs, attach them to a given Google account, and quickly retrieve them once the user authenticates.
- With respect to a custom solution, the use of Google Accounts requires users to have a Google account, but it does not require any further implementation.

5.Compute services

- Web applications are mostly designed to interface applications with users by means of a ubiquitous channel, that is, the Web
- Task queues
- Cron jobs

Task Queues

- Task Queues allow applications to submit a task for a later execution. This service is particularly useful for long computations that cannot be completed within the maximum response time of a request handler.
- The service allows users to have up to 10 queues that can execute tasks at a configurable rate.
- In fact, a task is defined by a Web request to a given URL, and the queue invokes the request handler by passing the payload as part of the Web request to the handler.
- It is the responsibility of the request handler to perform the “task execution,”

V.Application Life-Cycle

- AppEngine provides support for almost all the phases characterizing the life cycle of an application: testing and development, deployment, and monitoring.

1.Application development and testing

a) Java SDK

b) Python SDK

The Java SDK provides developers with the facility for building applications with the Java 5 and Java 6 runtime environments. it is possible to develop applications within the Eclipse development environment by using the Google AppEngine plug-in, which integrates the features of the SDK within the powerful Eclipse environment. These three components allow developers to program powerful and rich Java applications for AppEngine.

Python SDK The Python SDK allows developing Web applications for AppEngine with Python 2.5. It provides a standalone tool, called GoogleAppEngineLauncher, for managing Web applications locally and deploying them to AppEngine. The tool provides a convenient user interface that lists all the available Web applications

2.Application deployment and management:

An application identifier is mandatory because it allows unique identification of the application while it's interacting with AppEngine.

- Developers use an app identifier to upload and update applications.
- This is the primary tool used for application monitoring and provides users with insight into resource usage (CPU, bandwidth) and services and other useful counters.
- It is also possible to manage multiple versions of a single application

VI.Cost model.

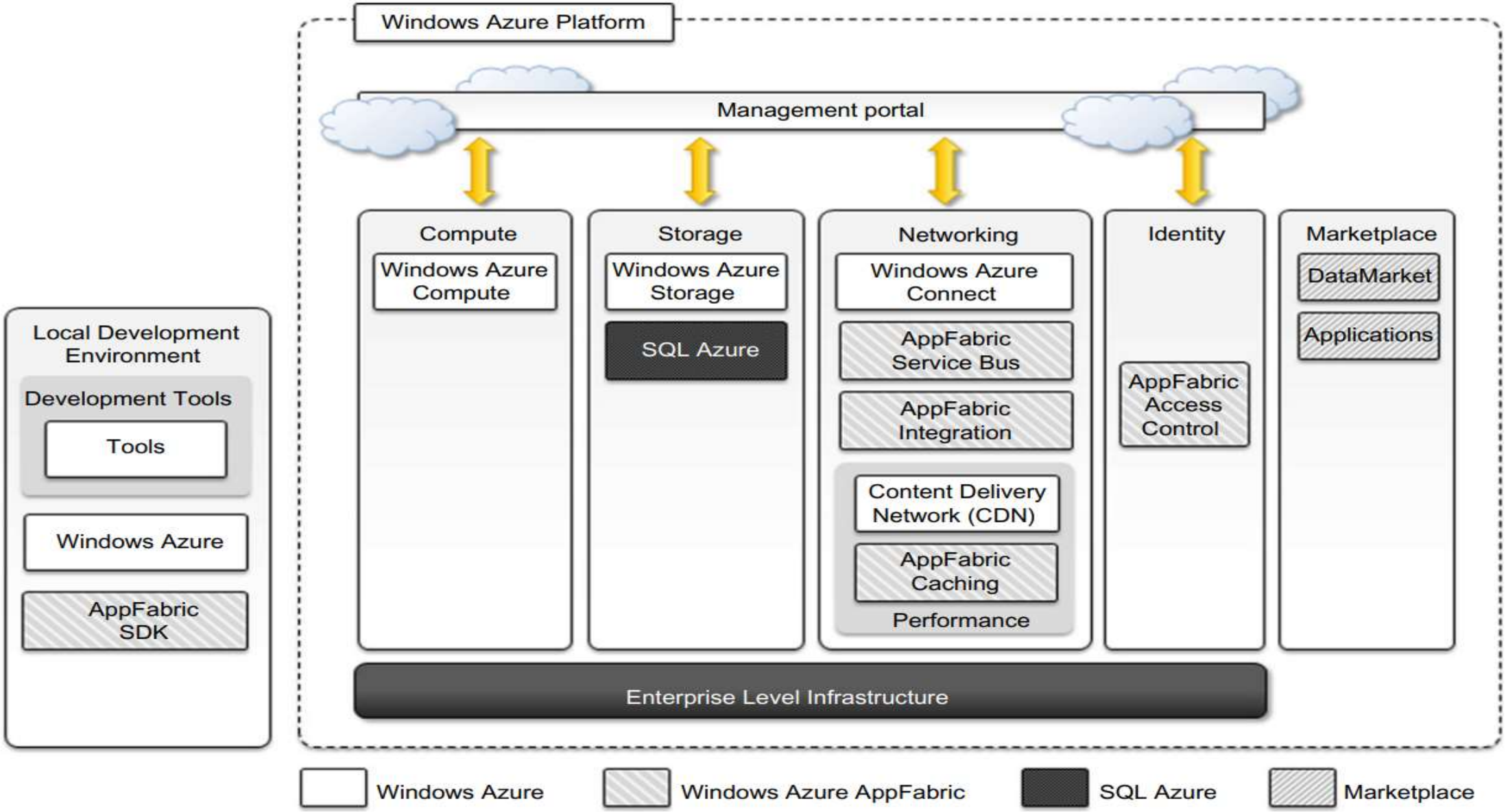
- AppEngine provides a free service with limited quotas that get reset every 24 hours.
- Once the application has been tested and tuned for AppEngine, it is possible to set up a billing account and obtain more allowance and be charged on a pay-per-use basis.
- This allows developers to identify the appropriate daily budget that they want to allocate for a given application.
- Once an application reaches the quota for a given resource, the resource is depleted and will not be available to the application until the quota is replenished.
- Once a resource is depleted, subsequent requests to that resource will generate an error or an exception.
- Resources such as CPU time and incoming or outgoing bandwidth will return an “HTTP 403” error page to users
- All the other resources and services will generate an exception that can be trapped in code to provide more useful feedback to users.

Cloud Cost Management Strategies in Companies

- **Budget control** —as a first step, organizations must establish budgets for cloud services and ensure teams are aware of them, and cannot exceed the budget for their specific project.
- **Right sizing** —ensuring compute instances, storage volumes and other services are provisioned at the level actually required by the business. It is very common for cloud resources to be provisioned and not fully utilized.
- **Auto scaling** —scale resources up and down dynamically according to application demand, ensuring you only pay for extra cloud resources during peak usage.
- **Scheduling** —many cloud services are not required 24/7, and can be scheduled to shut down when not needed. For example, services used by a US-based team can be shut down outside US business hours.
- **Detecting unused resources** —compute instances, storage volumes, load balancers, snapshots, and many other resources can easily be created and forgotten.
- **Smartly applying discounts** —discounted pricing models like reserved instances and spot instances can significantly drive down cloud costs, but they must be used appropriately.

VII.Microsoft Azure- Azure Core Concepts

Microsoft Windows Azure is a cloud operating system built on top of Microsoft datacenters' infrastructure and provides developers with a collection of services for building applications with cloud technology.



Microsoft Windows Azure Platform Architecture.

1.Azure core concepts

- The Windows Azure platform is made up of a foundation layer and a set of developer services that can be used to build scalable applications.
- These services cover compute, storage, networking, and identity management, which are tied together by middleware called AppFabric.
- This scalable computing environment is hosted within Microsoft data centers and accessible through the Windows Azure Management Portal.

a)**Compute services**: Compute services are the core components of Microsoft Windows Azure, and they are delivered by means of the abstraction of roles. **A role is a runtime environment that is customized for a specific compute task.**

Cntd.

- **Web role :**The Web role is designed to implement scalable Web applications. Web roles represent the units of deployment of Web applications within the Azure infrastructure.
- They are hosted on the IIS 7 Web Server, which is a component of the infrastructure that supports Azure.
- **Worker role** Worker roles are designed to host general compute services on Azure. They can be used to quickly provide compute power or to host services that do not communicate with the external world through HTTP.
- A common practice for Worker roles is to use them to provide background processing for Web applications developed with Web roles

- **Virtual machine role** :The Virtual Machine role allows developers to fully control the computing stack of their compute service by defining a custom image of the Windows Server 2008 R2 operating system and all the service stack required by their applications.

Compute Instance Type	CPU	Memory	Instance Storage	I/O Performance	Hourly Cost (USD)
Extra Small	1.0 GHz	768 MB	20 GB	Low	$0.04
Small	1.6 GHz	1.75 GB	225 GB	Moderate	$0.12
Medium	2 × 1.6 GHz	3.5 GB	490 GB	High	$0.24
Large	4 × 1.6 GHz	7 GB	1,000 GB	High	$0.48
Extra Large	8 × 1.6 GHz	14 GB	2,040 GB	High	$0.96

- provides an overview of the options offered during 20112012.
- **Storage services:**Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage.
- Compared to local storage, these services can be accessed by multiple clients

Block blobs. Block blobs are composed of blocks and are optimized for sequential access; therefore they are appropriate for media streaming.

Page blobs. Page blobs are made of pages that are identified by an offset from the beginning of the blob.

- **Azure drive Page blobs** can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file.
- This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage.
- A page blob mounted as part of an NTFS tree is called an Azure Drive.

- **Tables** Tables constitute a semistructured storage solution, allowing users to store information in the form of entities with a collection of properties.
- Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table.
- Users can insert, update, delete, and select a subset of the rows stored in the table.
- Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relationships among entities.
- For this reason, tables are more similar to spreadsheets rather than SQL tables.

- **Queues** :Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages.
- Applications enter messages into a queue, and other applications can read them in a first-in, first-out (FIFO) style.
- This two-phase process ensures that messages get processed before they are removed from the queue, and the client failures do not prevent messages from being processed.
- At the same time, this is also a reason that the queue does not enforce a strict FIFO model.
- All the services described are geo-replicated three times to ensure their availability in case of major disasters.
- Geo-replication involves the copying of data into a different datacenter that is hundreds or thousands of miles away from the original data center

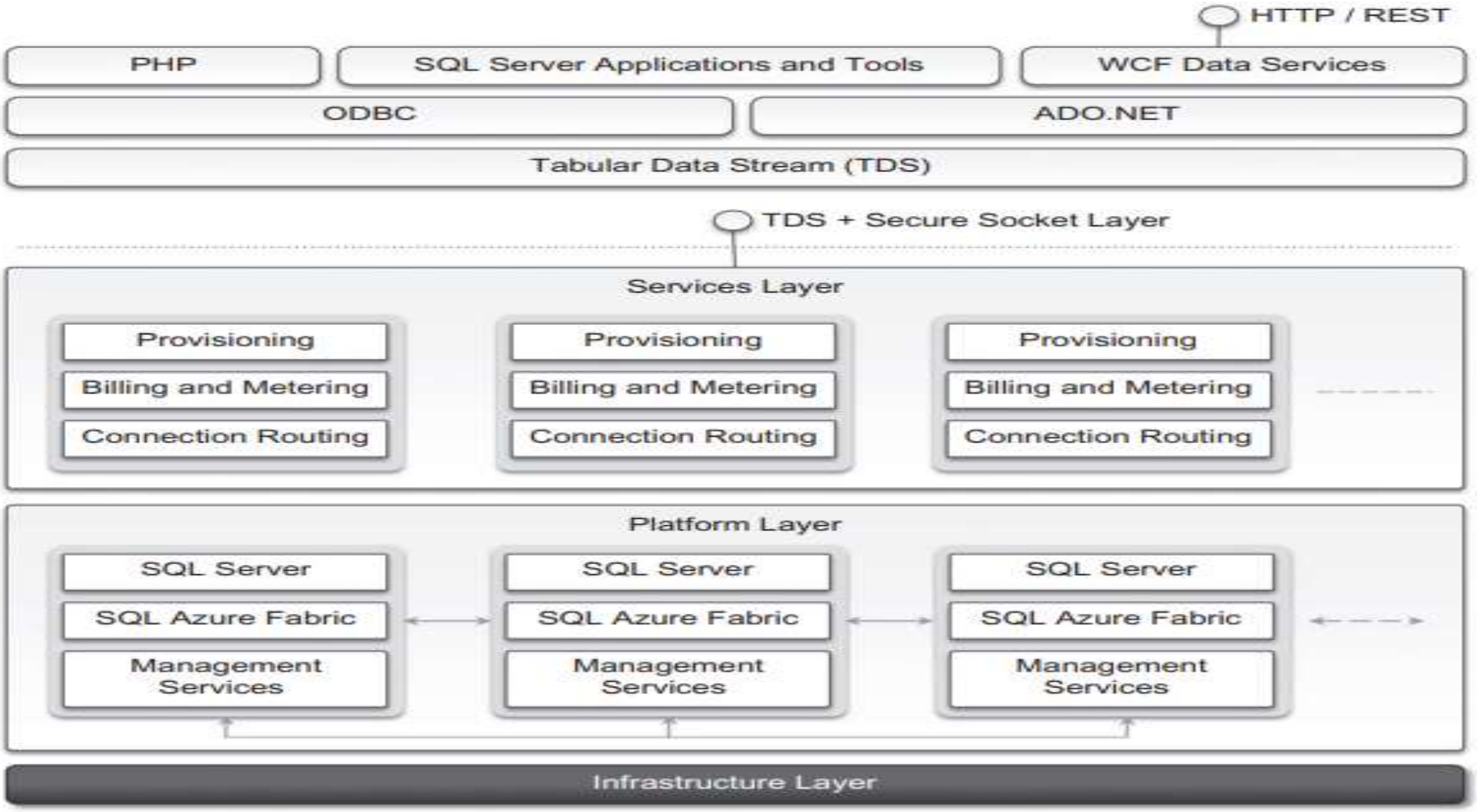
VIII.SQL Azure

- SQL Azure is accessible from either the Windows Azure Cloud or any other location that has access to the Azure Cloud.

- It is fully compatible with the interface exposed by SQL Server, so applications built for SQL Server can transparently migrate to SQL Azure.

- Moreover, the service is fully manageable using REST APIs, allowing developers to control databases deployed in the Azure Cloud as well as the firewall rules set up for their accessibility.

Architecture



- These services are logically part of server instances, which are managed by SQL Azure Fabric.
- This is the distributed database middleware that constitutes the infrastructure of SQL Azure and that is deployed on Microsoft datacenters.
- Developers have to sign up for a Windows Azure account in order to use SQL Azure.
- Once the account is activated, they can either use the Windows Azure Management Portal or the REST APIs to create servers and logins and to configure access to servers.
- SQL Azure servers are abstractions that closely resemble physical SQL Servers.
- SQL Azure ensures that multiple copies of each server are maintained within the Azure Cloud and that these copies are kept synchronized when client applications insert, update, and delete data on them


Thank you