Module – 5

Cloud Platforms in Industry. (Chapter - 9)

9. Cloud Platforms in Industry

An overview of a few prominent cloud computing platforms and a brief description of the types of service they offer.

A cloud computing system can be developed using either a single technology and vendor or a combination of them.

9.1 Amazon web services

Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage.

The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web-based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

Figure 9.1 shows all the 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) and Amazon Simple Storage Service (S3).

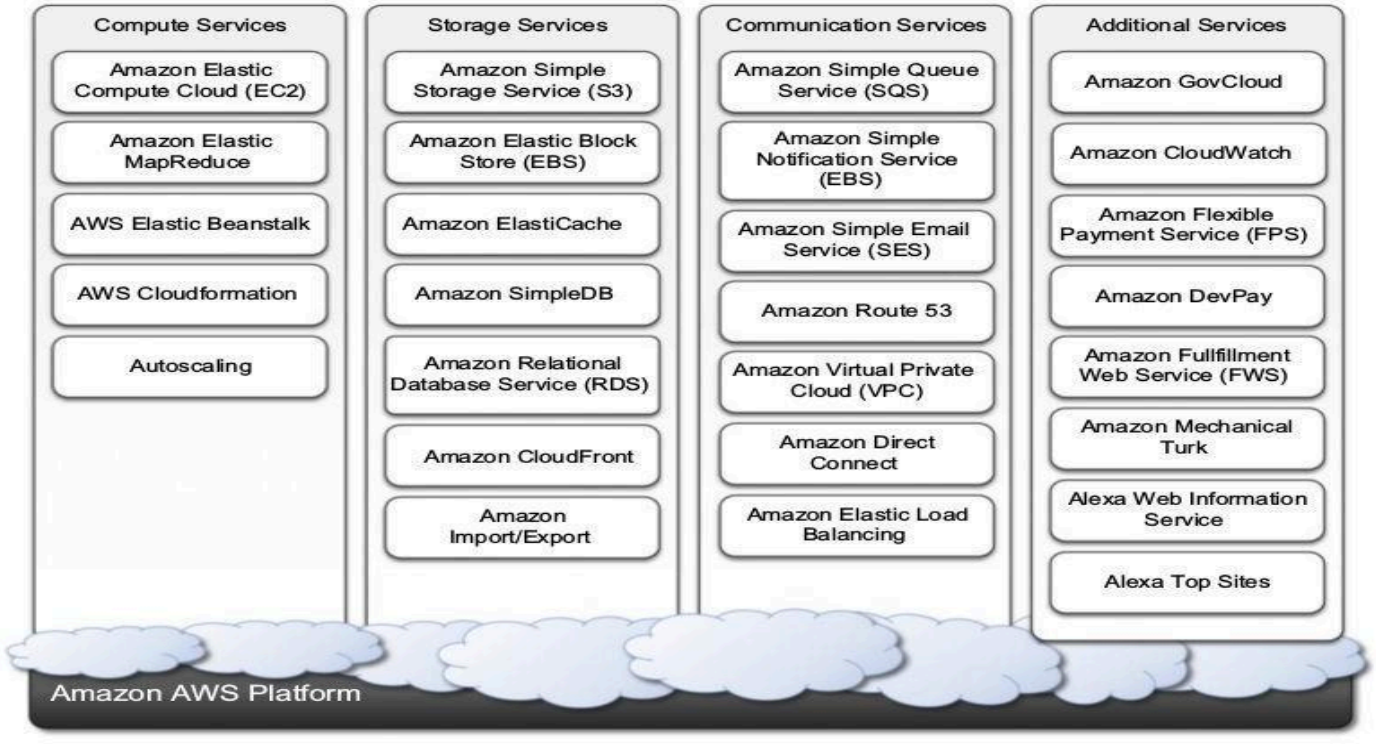

FIGURE 9.1

Amazon Web Services ecosystem.

9.1.1 Compute services

The fundamental service in this space is Amazon EC2, which delivers an IaaS solution that has served as a reference model for several offerings from other vendors in the same market segment.

Amazon EC2 allows deploying servers in the form of virtual machines created as instances of a specific image. Images come with a preinstalled operating system and a software stack, and instances can be configured for memory, number of processors, and storage.

Users are provided with credentials to remotely access the instance and further configure or install software if needed.

1. Amazon machine images
2. EC2 instances
3. EC2 environment
4. Advanced compute services

1. Amazon machine images

- Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine.
- They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxx and a manifest XML file.
- AMI contains physical file system layout with a predefined operating system installed as Amazon Ramdisk Image (ARI, id: ari-yyyyyy) and the Amazon Kernel Image (AKI, id: aki-zzzzzz)

A common practice is

- to prepare new AMIs to create an instance from a preexisting AMI,
- log into it once it is booted and running, and
- install all the software needed.

Using the tools provided by Amazon, we can convert the instance into a new image. Once an AMI is created, it is stored in an S3 bucket and the user can decide whether to make it available to other users or keep it for personal use.

2. EC2 instances

EC2 instances represent virtual machines. They are created using AMI as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs). The use of ECUs helps give users a consistent view of the performance offered by EC2 instances.

One ECU is defined as giving the same performance as a 1.0 - 1.2 GHz 2007 Opteron or 2007 Xeon processor.

We can identify six major configurations for EC2 instances.

- **Standard instances.** offers set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.
- **Micro instances.** suitable for those applications that consume limited amount of computing power and memory and need bursts in CPU cycles to process surges in workload.
- **High-memory instances.** 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.** targets compute-intensive applications.
- **Cluster Compute instances.** 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, which makes it suitable for HPC applications.
- **Cluster GPU instances.** provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance. This class is particularly suited for cluster applications that perform heavy graphic computations.

3. EC2 environment

EC2 instances are executed within a virtual environment. The EC2 environment is in charge of allocating addresses, attaching storage volumes, and configuring security in terms of access control and network connectivity.

It is possible to associate an Elastic IP to each instance. Elastic IPs allows instances running in EC2 to act as servers reachable from the Internet.
EC2 instances are given domain name in the form 0ec2 –
xxx-xxx-xxx.compute-x.amazonaws.com, where

1. xxx-xxx-xxx normally represents the four parts of the external IP address separated by a dash,
2. compute-x gives information about the availability zone where instances are deployed.

Currently, there are five availability zones: two in the United States (Virginia and Northern California), one in Europe (Ireland), and two in Asia Pacific (Singapore and Tokyo).

3. Advanced compute services

AWS CloudFormation constitutes an extension of the simple deployment model that characterizes EC2 instances. CloudFormation introduces the concepts of templates, which are JSON formatted text files that describe the resources needed to run an application or a service in EC2 together.
Templates provide a simple and declarative way to build complex systems and integrate EC2 instances with other AWS services such as S3, SimpleDB, SQS, SNS, Route 53, Elastic Beanstalk, and others.

AWS Elastic Beanstalk constitutes a simple and easy way to package applications and deploy them on the AWS Cloud. This service simplifies the process of provisioning instances and deploying application code and provides appropriate access to them.
Currently, this service is available for Web applications developed with the Java/Tomcat technology stack. Beanstalk simplifies tedious tasks without removing the user's capability of accessing—and taking over control of—the underlying EC2 instances.

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, and uses Amazon S3 for storage needs.
Elastic MapReduce introduces elasticity and allows users to dynamically size the Hadoop cluster according to their needs, as well as select the appropriate configuration of EC2 instances to compose the cluster.

9.1.2 Storage services

The core service is represented by Amazon Simple Storage Service (S3). This is a distributed object store that allows users to store information in different formats. The core components of S3 are two: buckets and objects. Buckets represent virtual containers in which to store objects; objects represent the content that is actually stored. Objects can also be enriched with metadata that can be used to tag the stored content with additional information.

1 S3 key concepts
2 Amazon elastic block store
3 Amazon ElastiCache
4 Structured storage solutions
5 Amazon CloudFront

S3 key concepts

S3 has been designed to provide a simple storage service that's accessible through a Representational State Transfer (REST) interface.

- The storage is organized in a two-level hierarchy.
- Stored objects cannot be manipulated like standard files.
- Content is not immediately available to users.
- Requests will occasionally fail.

Access to S3 is provided with RESTful Web services. These express all the operations that can be performed on the storage in the form of HTTP requests (GET, PUT, DELETE, HEAD, and POST).

Resource naming

Buckets, objects, and attached metadata are made accessible through a REST interface. Therefore, they are represented by uniform resource identifiers (URIs) under the s3.amazonaws.com domain.

Amazon offers three different ways of addressing a bucket:

1. Canonical form: http://s3.amazonaws.com/bukect_name/
2. Subdomain form: http://bucketname.s3.amazon.com/
3. Virtual hosting form: http://bucket-name.com/ Buckets

A bucket is a container of objects. It can be thought of as a virtual drive hosted on the S3 distributed storage, which provides users with a flat store to which they can add objects. Buckets are top-level elements of the S3 storage architecture and do not support nesting. That is, it is not possible to create "subbuckets" or other kinds of physical divisions.

Objects and metadata

Objects constitute the content elements stored in S3. Users either store files or push to the S3 text stream representing the object's content. An object is identified by a name that needs to be unique within the bucket in which the content is stored. The name cannot be longer than 1,024 bytes when encoded in UTF-8, and it allows almost any character. Buckets do not support nesting.

Access control and security

Amazon S3 allows controlling the access to buckets and objects by means of Access Control Policies (ACPs). An ACP is a set of grant permissions that are attached to a resource expressed by means of an XML configuration file.

A policy allows defining up to 100 access rules, each of them granting one of the available permissions to a grantee.

Currently, five different permissions can be used:

A. READ allows the grantee to retrieve an object and its metadata and to list the content of a bucket as well as getting its metadata.
B. WRITE allows the grantee to add an object to a bucket as well as modify and remove it.
C. READ_ACP allows the grantee to read the ACP of a resource.
D. WRITE_ACP allows the grantee to modify the ACP of a resource.
E. FULL_CONTROL grants all of the preceding permissions.

Amazon elastic block store

The Amazon Elastic Block Store (EBS) allows AWS users to provide EC2 instances with persistent storage in the form of volumes that can be mounted at instance startup. They accommodate up to 1 TB of space and are accessed through a block device interface, thus allowing users to format them according to the needs of the instance they are connected to.

EBS volumes normally reside within the same availability zone of the EC2 instances that will use them to maximize the I/O performance. It is also possible to connect volumes located in different availability zones. Once mounted as volumes, their content is lazily loaded in the background and according to the request made by the operating system. This reduces the number of I/O requests that go to the network.

Amazon ElastiCache

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

ElastiCache is based on a cluster of EC2 instances running the caching software, which is made available through Web services.
An ElastiCache cluster can be dynamically resized according to the demand of the client applications.

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.

Preconfigured EC2 AMIs are predefined templates featuring an installation of a given database management system. 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.
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, designing failover strategies, or keeping the servers up-to-date with patches. Moreover, the service provides users with automatic backups, snapshots, point-in-time recoveries, and facilities for implementing replications.

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. SimpleDB provides support for semistructured data, the model for which is based on the concept of domains, items, and attributes.
SimpleDB uses domains as top-level elements to organize a data store. These domains are roughly comparable to tables in the relational model. Unlike tables, they allow items not to have all the same column structure; each item is therefore represented as a collection of attributes expressed in the form of a key-value pair.

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.
AWS provides users with simple Web service APIs to manage CloudFront. To make available content through CloudFront, it is necessary to create a distribution. This identifies an origin server, which contains the original version of the content being distributed, and it is referenced by a DNS domain under the Cloudfront.net domain name.
The content that can be delivered through CloudFront is static (HTTP and HTTPS) or streaming (Real Time Messaging Protocol, or RMTP).

9.1.3 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:
1. **Virtual networking** and
2. **Messaging**.

1. 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 VPC provides flexibility in creating virtual private networks within the Amazon

infrastructure and beyond. The service providers prepare templates for network service for advanced configurations. Templates include public subnets, isolated networks, private networks accessing Internet through network address translation (NAT), and hybrid networks including AWS resources and private resources.

Amazon Direct Connect allows AWS users to create dedicated networks between the user private network and Amazon Direct Connect locations, called ports. This connection can be further partitioned in multiple logical connections and give access to the public resources hosted on the Amazon infrastructure. The advantage is the consistent performance of the connection between the users premises and the Direct Connect locations.

Amazon Route 53 implements dynamic DNS services that allow AWS resources to be reached through domain names different from the amazon.com domain. By leveraging the large and globally distributed network of Amazon DNS servers.

2. **Messaging**.

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).

Amazon SQS constitutes disconnected model for exchanging messages between applications by means of message queues, hosted within the AWS infrastructure. Using the AWS console or directly the underlying Web service AWS, users can create an unlimited number of message queues and configure them to control their access. Applications can send messages to any queue they have access to. These messages are securely and redundantly stored within the AWS infrastructure for a limited period of time, and they can be accessed by other (authorized) applications.

Amazon SNS provides a publish-subscribe method for connecting heterogeneous applications. Amazon SNS allows applications to be notified when new content of interest is available. This feature is accessible through a Web service whereby AWS users can create a topic, which other applications can subscribe.

Amazon SES provides AWS users with a scalable email service that leverages the AWS infrastructure. Once users are signed up for the service, they have to provide an email that SES will use to send emails on their behalf. To activate the service, SES will send an email to verify the given address and provide the users with the necessary information for the activation.

9.2 Google AppEngine

Google AppEngine is a PaaS implementation.

Distributed and scalable runtime environment that leverages Google's distributed infrastructure to scale out applications.

9.2.1 Architecture and core concepts

AppEngine is a platform for developing scalable applications accessible through the Web.

The platform is logically divided into four major components: infrastructure, the runtime environment, the underlying storage, and the set of scalable services.

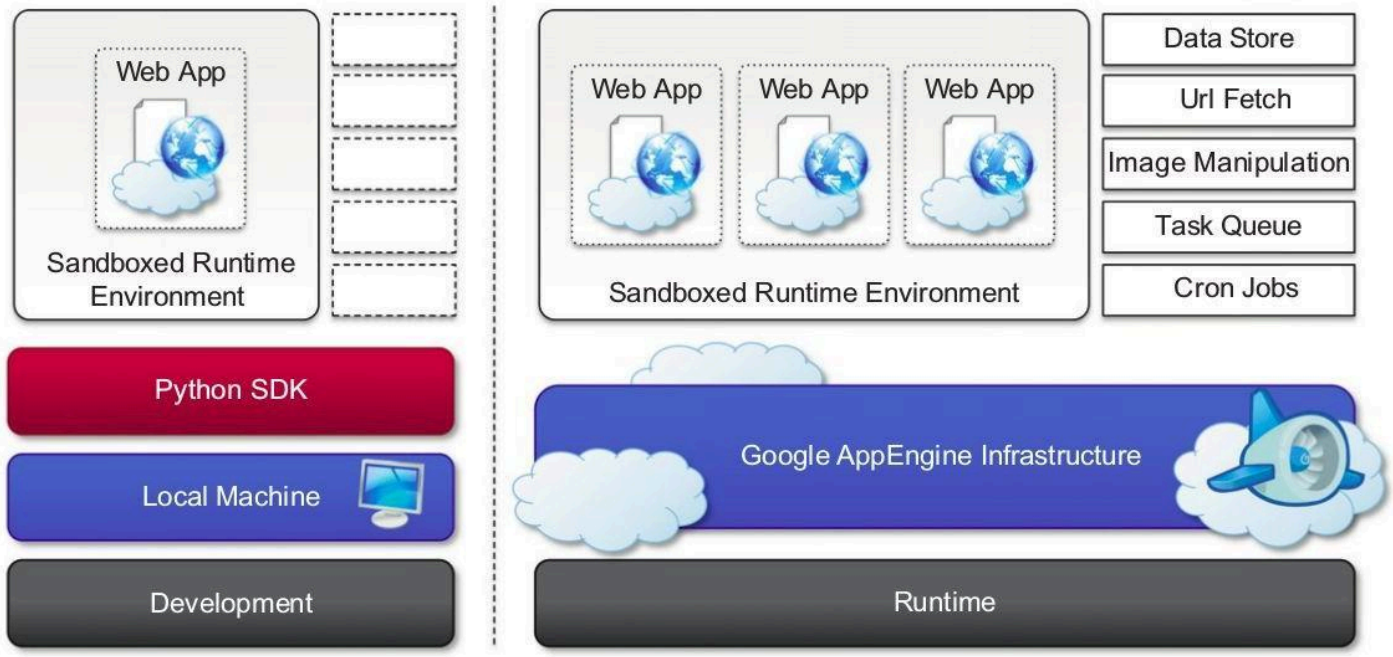

FIGURE 9.2

Google AppEngine platform architecture.

1 Infrastructure

AppEngine hosts Web applications, and its primary function is to serve users requests efficiently.

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 or redirects the request to an existing server.

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

2 Runtime environment

The runtime environment represents the execution context of applications hosted on AppEngine.

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.

If an application tries to perform any operation that is considered potentially harmful, an exception is thrown and the execution is interrupted.

Supported runtimes- Currently, it is possible to develop AppEngine applications using three different languages and related technologies: Java, Python, and Go.

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.

Support for Python is provided by an optimized Python 2.5.2 interpreter. As with Java, the runtime environment supports the Python standard library.

Developers can use a specific Python Web application framework, called webapp, simplifying the development of Web applications.

The Go runtime environment allows applications developed with the Go programming language to be hosted and executed in AppEngine. Currently the release of Go that is supported by AppEngine is r58.1. The SDK includes the compiler and the standard libraries for developing applications in Go and interfacing it with AppEngine services.

3 Storage

AppEngine provides various types of storage, which operate differently depending on the volatility of

the data. **Static file servers**- Web applications are composed of dynamic and static data. Dynamic data are a result of the logic of the application and the interaction with the user. Static data often are mostly constituted of the components that define the graphical layout of the application or data files.

DataStore-

- DataStore is a service that allows developers to store semistructured data. The service is designed to scale and optimized to quickly access data. DataStore can be considered as a large object database in which to store objects that can be retrieved by a specified key.
- DataStore imposes less constraint on the regularity of the data but, at the same time, does not implement some of the features of the relational model.
- The underlying infrastructure of DataStore is based on Bigtable, a redundant, distributed, and semistructured data store that organizes data in the form of tables.
- DataStore provides high-level abstractions that simplify interaction with Bigtable. Developers define their data in terms of entity and properties, and these are persisted and maintained by the service into tables in Bigtable.
- DataStore also provides facilities for creating indexes on data and to update data within the context of a transaction. Indexes are used to support and speed up queries. A query can return zero or more objects of the same kind or simply the corresponding keys.

4 Application services

Applications hosted on AppEngine take the most from the services made available through the runtime environment. These services simplify most of the common operations that are performed in Web applications

UrlFetch - The sandbox environment does not allow applications to open arbitrary connections through sockets, but it does provide developers with the capability of retrieving a remote resource through HTTP/HTTPS by means of the UrlFetch service. Applications can make synchronous and asynchronous Web requests and integrate the resources obtained in this way into the normal request-handling cycle of the application.

UrlFetch is not only used to integrate meshes into a Web page but also to leverage remote Web services in accordance with the SOA reference model for distributed applications.

MemCache- This is a distributed in-memory cache that is optimized for fast access and provides developers with a volatile store for the objects that are frequently accessed. The caching algorithm implemented by MemCache will automatically remove the objects that are rarely accessed. The use of MemCache can significantly reduce the access time to data; developers can structure their applications so that each object is first looked up into MemCache and if there is a miss, it will be retrieved from DataStore and put into the cache for future lookups.

Mail and instant messaging- 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.

AppEngine provides also another way to communicate with the external world: the Extensible Messaging and Presence Protocol (XMPP). Any chat service that supports XMPP, such as Google Talk, can send and receive chat messages to and from the Web application, which is identified by its own address.

Account management- AppEngine simplifies account management by allowing developers to leverage Google account management by means of Google Accounts.

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.

5 Compute services

AppEngine offers additional services such as Task Queues and Cron Jobs that simplify the execution of computations.

Task queues- 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," which is seen from the queue as a simple Web request.

Cron jobs- the required operation needs to be performed at a specific time of the day, which does not coincide with the time of the Web request. In this case, it is possible to schedule the required operation at the desired time by using the Cron Jobs service.

9.2.2 Application life cycle

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

1 Application development and testing

Developers can start building their Web applications on a local development server.

The development server simulates the AppEngine runtime environment by providing a mock implementation of DataStore, MemCache, UrlFetch, and the other services leveraged by Web applications.

AppEngine builds indexes for each of the queries performed by a given application in order to speed up access to the relevant data. This capability is enabled by a priori knowledge about all the possible queries made by the application; such knowledge is made available to AppEngine by the developer while uploading the application.

Java SDK- The Java SDK provides developers with the facility for building applications with the Java 5 and Java 6 runtime environments. Using the Eclipse software installer, it is possible to download and install Java SDK, Google Web Toolkit, and Google AppEngine plug-ins into Eclipse. 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 Python implementation of the SDK also comes with an integrated Web application framework called webapp that includes a set of models, components, and tools that simplify the development of Web applications and enforce a set of coherent practices.

The webapp framework has been reimplemented and made available in the Python SDK so that it can be used with AppEngine.

2 Application deployment and management

Once application has been developed and tested, it can be deployed on AppEngine with simple click or command-line tool.

Before performing such task, it is necessary to create application identifier, which will be used to locate application from Web browser by typing the address http://<application-id>.appspot.com.

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.

Once an application identifier has been created, it is possible to deploy an application on AppEngine. This task can be done using either respective development environment (GoogleAppEngineLauncher and Google AppEngine plug-in) or the command-line tools.

9.2.3 Cost model

Once 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 appropriate daily budget that they want to allocate for given application.

An application is measured against **billable quotas, fixed quotas, and per-minute quotas.**

Google AppEngine uses these quotas to ensure that users do not spend more than the allocated budget and that applications run without being influenced by each other from a performance point of view.

Billable quotas identify the daily quotas that are set by application administrator and are defined by daily budget allocated for application.

Free quotas are part of the billable quota and identify the portion of the quota for which users are not charged. **Fixed quotas** are internal quotas set by AppEngine that identify infrastructure boundaries and define operations that application can carry out on infrastructure.

9.2.4 Observations

AppEngine, a framework for developing scalable Web applications, leverages Google's infrastructure.

The core components of service are scalable and sandboxed runtime environment for executing applications and a collection of services that implement most of the common features.

One of the characteristic elements of AppEngine is use of simple interfaces that allow applications to perform specific operations that are optimized and designed to scale.

Building on top of these blocks, developers can build applications and let AppEngine scale them out when needed.

Cloud Applications. (Chapter - 10)

Cloud computing has gained huge popularity in industry due to its ability to host applications for which the services can be delivered to consumers rapidly at minimal cost.

This chapter discusses some application case studies, detailing their architecture and how they leveraged various cloud technologies.

Applications from a range of domains, from scientific to engineering, gaming, and social networking, are considered.

10.1 Scientific applications

10.1.1 Healthcare: ECG analysis in the cloud

10.1.2 Biology: protein structure prediction

10.1.3 Biology: gene expression data analysis for cancer diagnosis

10.1.4 Geoscience: satellite image processing

10.2 Business and consumer applications

10.2.1 CRM and ERP

1 Salesforce.com

2 Microsoft dynamics CRM

3 NetSuite

10.2.2 Productivity

1 Dropbox and iCloud

2 Google docs

3 Cloud desktops: EyeOS and XIOS/3

10.2.3 Social networking 1 Facebook

10.2.4 Media applications

1 Animoto

2 Maya rendering with Aneka

3 Video encoding on the cloud: Encoding.com

10.2.5 Multiplayer online gaming

10.1 Scientific Applications

Scientific applications are a sector that is increasingly using cloud computing systems and technologies.

Cloud computing systems meet the needs of different types of applications in the scientific domain: high-performance computing (HPC) applications, high-throughput computing (HTC) applications, and data-intensive applications.

The opportunity to use cloud resources is even more appealing because minimal changes need to be made to existing applications in order to leverage cloud resources.

10.1.1 Healthcare: ECG analysis in the cloud

Healthcare is a domain in which computer technology has found several and diverse applications: from supporting the business functions to assisting scientists in developing solutions to cure diseases.

An illustration of the ķpatient's heartbeat. Such information is transmitted to the patient's mobile device, which will eventually forward it to the cloud-hosted Web service for analysis.

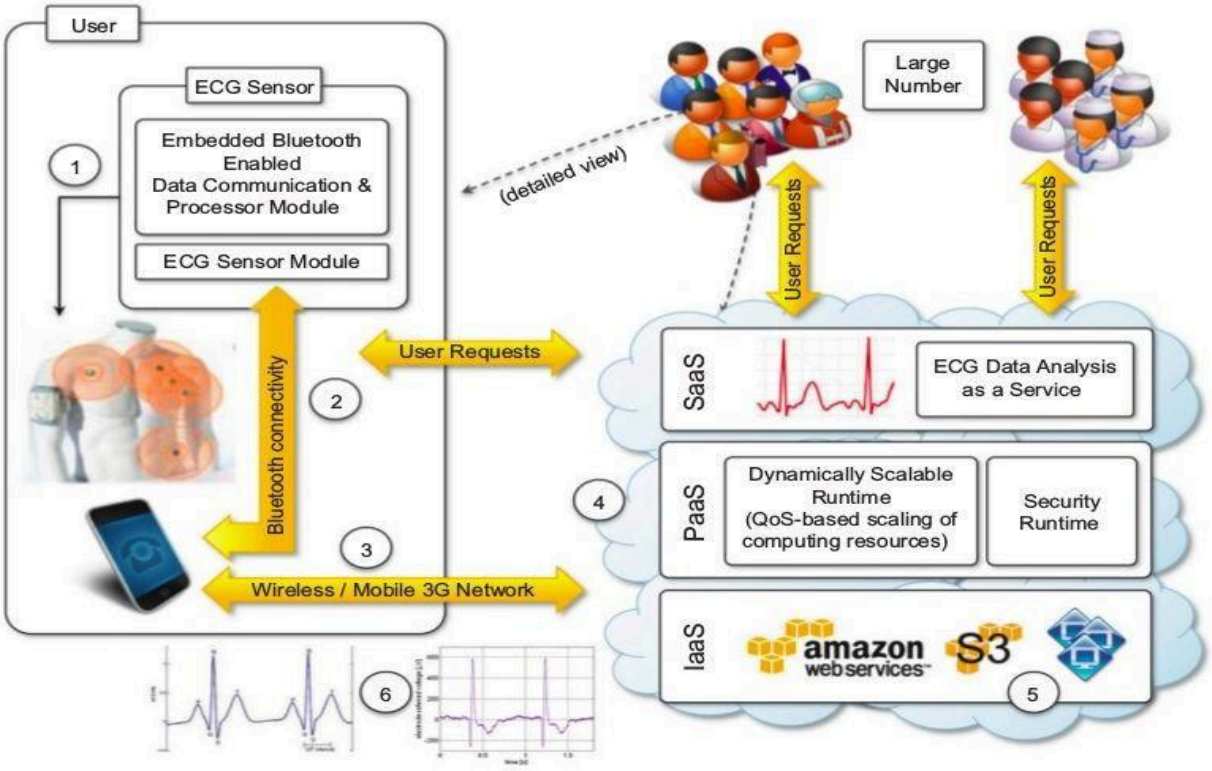

FIGURE 10.1

An online health monitoring system hosted in the cloud.

The Web service forms the front-end of a platform that is hosted in cloud and leverages three layers of cloud computing stack: SaaS, PaaS, and IaaS.

The Web service constitute SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud platform.

The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka.

The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances used to execute the single tasks defined by the workflow engine for a single ECG processing job.

Advantages

1. The elasticity of cloud infrastructure that can grow and shrink according to the requests served. As a result, doctors and hospitals do not have to invest in large computing infrastructures designed after capacity planning, thus making more effective use of budgets.
2. Ubiquity. Cloud computing technologies are easily accessible and promise to deliver systems with minimum or no downtime. Computing systems hosted in cloud are accessible from any Internet device through simple interfaces (such as SOAP and REST-based Web services). This makes systems easily integrated with other systems maintained on hospital's premises.
3. Cost savings. Cloud services are priced on a pay-per-use basis and with volume prices for large numbers of service requests.

10.1.2 Biology: protein structure prediction

Applications in biology require high computing capabilities and operate on large data-sets that cause extensive I/O operations.

Therefore biology applications have made use of supercomputing and cluster computing infrastructures. Similar capabilities can be leveraged using cloud computing technologies in a more dynamic fashion, thus opening new opportunities for bioinformatics applications.

Protein structure prediction is a computationally intensive task that is fundamental to different types

of research in the life sciences.

The geometric structure of a protein cannot be directly inferred from the sequence of genes that compose its structure, but it is the result of complex computations aimed at identifying the structure that minimizes the required energy.

This task requires the investigation of a space with a massive number of states, consequently creating a large number of computations for each of these states.

One project that investigates the use of cloud technologies for protein structure prediction is **Jeeva** - an integrated Web portal that enables scientists to offload the prediction task to a computing cloud based on Aneka (**Figure 10.2**).

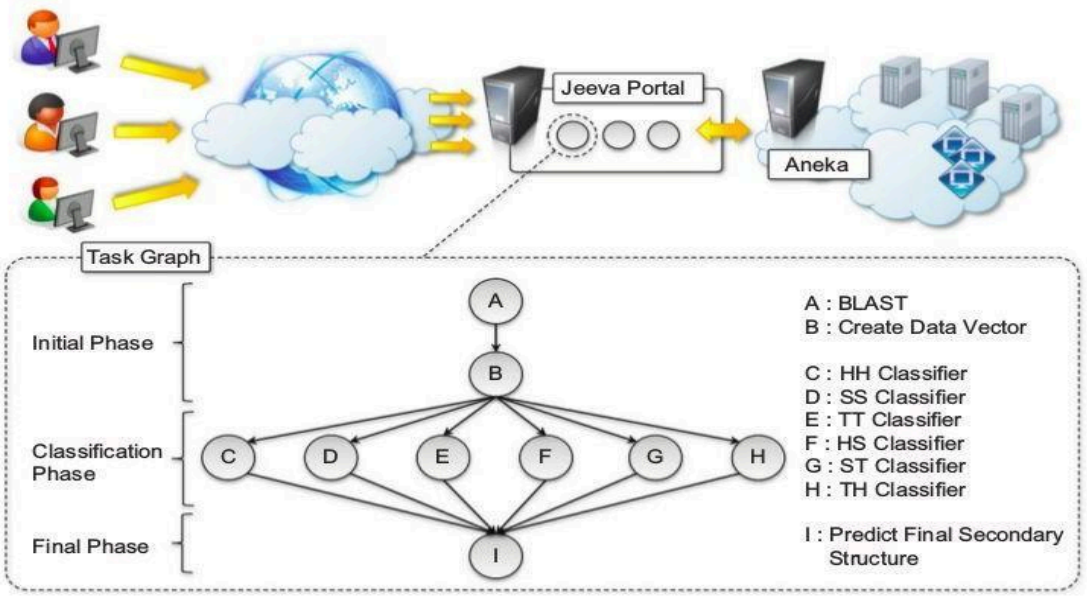

FIGURE 10.2

Architecture and overview of the Jeeva Portal.

The prediction task uses machine learning techniques (support vector machines) for determining the secondary structure of proteins.

These techniques translate problem into one of pattern recognition, where a sequence has to be classified into one of three possible classes (E, H, and C).

A popular implementation based on support vector machines divides the pattern recognition problem into three phases: initialization, classification, and a final phase.

These three phases have to be executed in sequence, we can perform parallel execution in the classification phase, where multiple classifiers are executed concurrently.

This reduces computational time of the prediction.

The prediction algorithm is then translated into a task graph that is submitted to Aneka.

Once the task is completed, the middleware makes the results available for visualization through the portal.

The advantage of using cloud technologies is the capability to leverage a scalable computing infrastructure that can be grown and shrunk on demand.

10.1.3 Biology: gene expression data analysis for cancer diagnosis

Gene expression profiling is the measurement of the expression levels of thousands of genes at once. It is used to understand the biological processes that are triggered by medical treatment at a cellular level.

important application of gene expression profiling is cancer diagnosis and treatment.

Cancer is a disease characterized by uncontrolled cell growth and proliferation. This behavior occurs because genes regulating the cell growth mutate.

Gene expression profiling is utilized to provide a more accurate classification of tumors.The dimensionality of typical gene expression datasets ranges from several thousands to over tens of thousands of genes.

This problem is approached with learning classifiers, which generate a population of condition-action rules that guide the classification process. The eXtended Classifier System (XCS) has been utilized for classifying large datasets in bioinformatics and computer science domains.

A variation of this algorithm, CoXCS [162], has proven to be effective in these conditions. CoXCS divides the entire search space into subdomains and employs the standard XCS algorithm in each of these subdomains. Such a process is computationally intensive but can be easily parallelized because the classifications problems on the subdomains can be solved concurrently.

Cloud-CoXCS (**Figure 10.3**) is a cloud-based implementation of CoXCS that leverages Aneka to solve the classification problems in parallel and compose their outcomes. The algorithm is controlled by strategies, which define the way the outcomes are composed together and whether the process needs to be iterated.

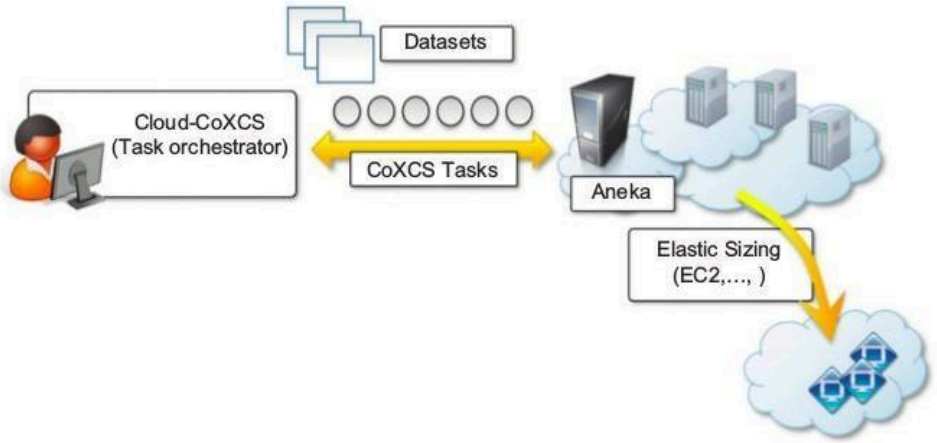

FIGURE 10.3

Cloud-CoXCS: An environment for microarray data processing on the cloud.

10.1.4 Geoscience: satellite image processing

Geoscience applications collect, produce, and analyze massive amounts of geospatial and nonspatial data. As the technology progresses and our planet becomes more instrumented, volume of data that needs to be processed increases significantly.

Geographic information system (GIS) applications capture, store, manipulate, analyze, manage, and present all types of geographically referenced data.

Cloud computing is an attractive option for executing these demanding tasks and extracting meaningful information to support decision makers.

Satellite remote sensing generates hundreds of gigabytes of raw images that need to be further processed to become the basis of several different GIS products. This process requires both I/O and compute-intensive tasks. Large images need to be moved from a ground station's local storage to compute facilities, where several transformations and corrections are applied.

The system shown in **Figure 10.4** integrates several technologies across the entire computing stack.

A SaaS application provides a collection of services for such tasks as geocode generation and data visualization.

At the PaaS level, Aneka controls the importing of data into the virtualized infrastructure and the execution of image-processing tasks that produce the desired outcome from raw satellite images.

The platform leverages a Xen private cloud and the Aneka technology to dynamically provision the required resources.

FIGURE 10.4
A cloud environment for satellite data processing.

10.2 Business and consumer applicatons

The business and consumer sector is the one that benefits the most from cloud computing technologies.

On one hand, the opportunity to transform capital costs into operational costs makes clouds an attractive option for all enterprises that are IT-centric.

On the other hand, the sense of ubiquity that the cloud offers for accessing data and services makes it interesting for end users.

The combination of all these elements has made cloud computing the preferred technology for a wide range of applications.

10.2.1 CRM and ERP

Customer relationship management (CRM) and enterprise resource planning (ERP) applications are market segments that are flourishing in the cloud

Cloud CRM applications constitute a great opportunity for small enterprises and start-ups to have fully functional CRM software without large up-front costs and by paying subscriptions.

Your business and customer data from everywhere and from any device, has fostered the spread of cloud CRM applications.

ERP solutions on the cloud are less mature and have to compete with well-established in-house solutions. ERP systems integrate several aspects of an enterprise: finance and accounting, human resources, manufacturing, supply chain management, project management, and CRM.

1 Salesforce.com

Salesforce.com is most popular and developed CRM solution available today.

As of today more than 100,000 customers have chosen Safesforce.com to implement their CRM solutions.

The application provides customizable CRM solutions that can be integrated with additional features developed by third parties.

Salesforce.com is based on the Force.com cloud development platform.

This represents scalable and high-performance middleware executing all operations of all Salesforce.com applications.

The architecture of the Force.com platform is shown in Figure 10.5.

At the core of the platform resides its metadata architecture, which provides the system with flexibility and scalability.

Application core logic and business rules are saved as metadata into the Force.com store.

Both application structure and application data are stored in the store. A runtime engine executes application logic by retrieving its metadata and then performing the operations on the data.

A full-text search engine supports the runtime engine. This allows application users to have an effective user experience The search engine maintains its indexing data in a separate store.

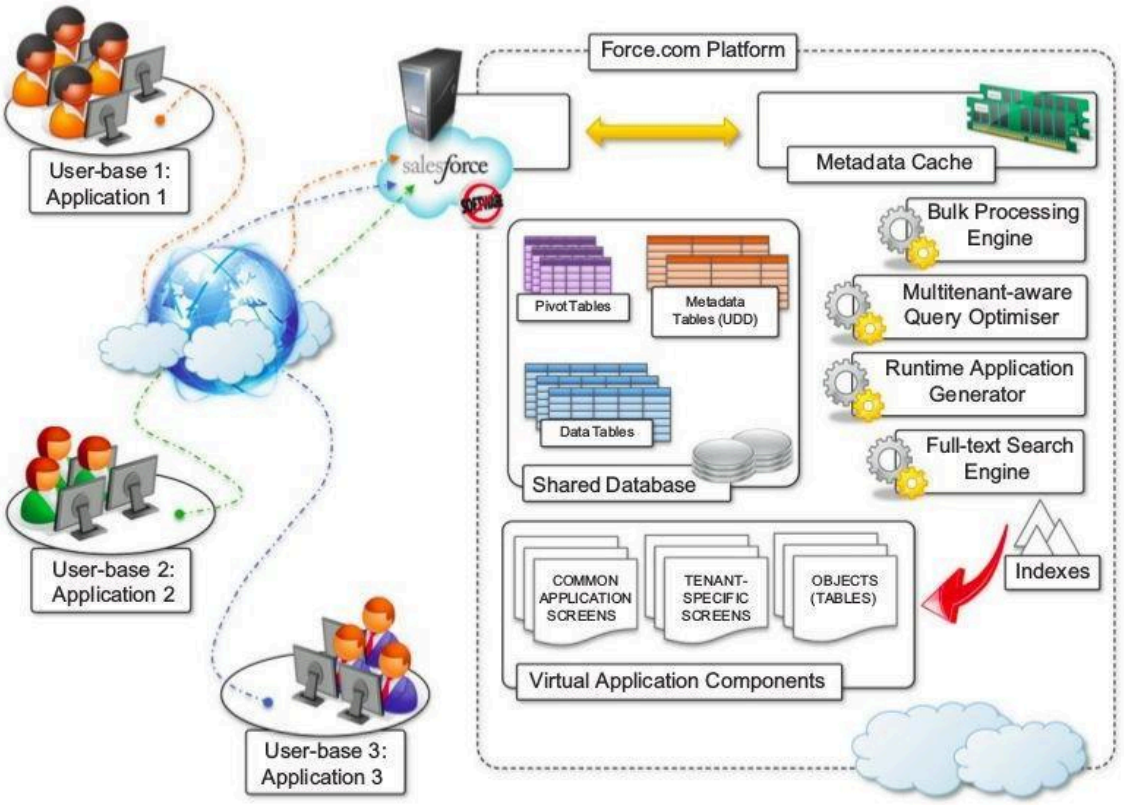

FIGURE 10.5

Salesforce.com and Force.com architecture.

2 Microsoft dynamics CRM

The system is completely hosted in Microsoft's datacenters across the world and offers to customers a 99.9% SLA.

Each CRM instance is deployed on a separate database, and application provides users with facilities for marketing, sales, and advanced customer relationship management.

Dynamics CRM Online features can be accessed either through a Web browser interface or by means of SOAP and RESTful Web services.

This allows Dynamics CRM to be easily integrated with both other Microsoft products and line-of-business applications.

Dynamics CRM can be extended by developing plug-ins that allow implementing specific behaviors triggered on the occurrence of given events.

3 NetSuite

NetSuite provides a collection of applications that help customers manage every aspect of the business enterprise.

Its offering is divided into three major products: NetSuite Global ERP, NetSuite Global CRM1 , and NetSuite Global Ecommerce.

Moreover, an all-in-one solution: NetSuite One World, integrates all three products together.

The services NetSuite delivers are powered by two large datacenters on the East and West coasts of the United States, connected by redundant links.

This allows NetSuite to guarantee 99.5% uptime to its customers.

The NetSuite Business Operating System (NS-BOS) is a complete stack of technologies for building SaaS business applications that leverage the capabilities of NetSuite products.

On top of the SaaS infrastructure, the NetSuite Business Suite components offer accounting, ERP, CRM, and ecommerce capabilities.

10.2.2 Productivity

Productivity applications replicate in cloud. The most common tasks that we are used to performing on our desktop: from document storage to office automation and complete desktop environments hosted in the cloud.

1 Dropbox and iCloud

Online storage solutions have turned into SaaS applications and become more usable as well as more advanced and accessible.

The most popular solution for online document storage is **Dropbox**, that allows users to synchronize any file across any platform and any device in a seamless manner (**Figure 10.6**). Dropbox provides users with a free storage that is accessible through the abstraction of a folder. Users can either access their Dropbox folder through a browser or by downloading and installing a Dropbox client, which provides access to the online storage by means of a special folder. All the modifications into this folder are silently synched so that changes are notified to all the local instances of the Dropbox folder across all the devices.

Another interesting application in this area is **iCloud**, a cloud-based document-sharing application provided by Apple to synchronize iOS-based devices in a completely transparent manner.

Documents, photos, and videos are automatically synched as changes are made, without any explicit operation. This allows the system to efficiently automate common operations without any human intervention.

This capability is limited to iOS devices, and currently there are no plans to provide iCloud with a Web-based interface that would make user content accessible from even unsupported platforms.

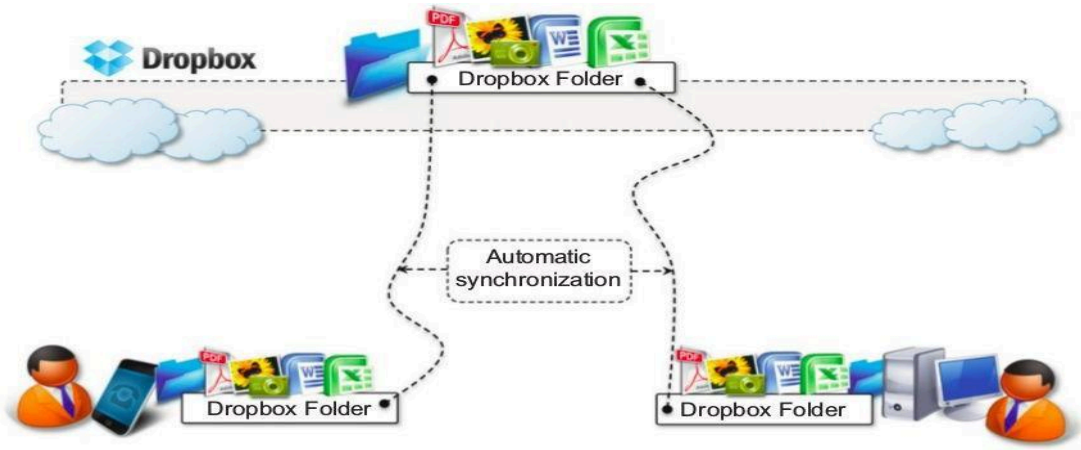

FIGURE 10.6

Dropbox usage scenario.

2 Google docs

Google Docs is a SaaS application that delivers the basic office automation capabilities with support for collaborative editing over the Web.

Google Docs allows users to create and edit text documents, spreadsheets, presentations, forms, and

drawings. It aims to replace desktop products such as Microsoft Office and OpenOffice and provide similar interface and functionality as a cloud service.

By being stored in the Google infrastructure, these documents are always available from anywhere and from any device that is connected to the Internet.

Google Docs is a good example of what cloud computing can deliver to end users: ubiquitous access to resources, elasticity, absence of installation and maintenance costs, and delivery of core functionalities as a service.

3 Cloud desktops: EyeOS and XIOS/3

EyeOS is one of the most popular Web desktop solutions based on cloud technologies. It replicates the functionalities of a classic desktop environment and comes with pre-installed applications for the most common file and document management tasks (**Figure 10.7**). Single users can access the EyeOS desktop environment from anywhere and through any Internet-connected device, whereas organizations can create a private EyeOS Cloud on their premises to virtualize the desktop environment of their employees and centralize their management.

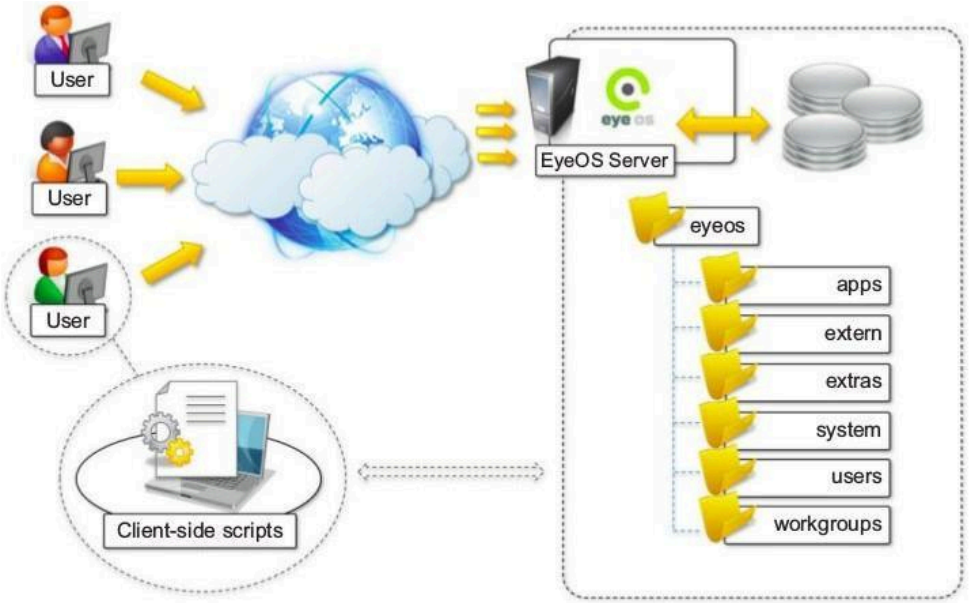

FIGURE 10.7

EyeOS architecture.

The EyeOS architecture is quite simple: On the server side, the EyeOS application maintains the information about user profiles and their data, and the client side constitutes the access point for users and administrators to interact with the system. EyeOS stores the data about users and applications on the server file system. Once the user has logged in by providing credentials, the desktop environment is rendered in the client's browser by downloading all the JavaScript libraries required to build the user interface and implement the core functionalities of EyeOS.

EyeOS also provides APIs for developing new applications and integrating new capabilities into the system. EyeOS applications are server-side components that are defined by at least two files (stored in the eyeos/apps/appname directory): appname.php and appname.js. The first file defines and implements all the operations that the application exposes; the JavaScript file contains the code that needs to be loaded in the browser in order to provide user interaction with the application.

Xcerion XML Internet OS/3 (XIOS/3) is another example of a Web desktop environment. The service is delivered as part of the CloudMe application, which is a solution for cloud document storage. The key differentiator of XIOS/3 is its strong leverage of XML, used to implement many of the tasks of the OS: rendering user interfaces, defining application business logics, structuring file

system organization, and even application development.

XIOS/3 is released as open-source software and implements a marketplace where third parties can easily deploy applications that can be installed on top of the virtual desktop environment. It is possible to develop any type of application and feed it with data accessible through XML Web services: developers have to define the user interface, bind UI components to service calls and operations, and provide the logic on how to process the data.

10.2.3 Social networking

Social networking applications have grown considerably in the last few years to become the most active sites on the Web.

To sustain their traffic and serve millions of users seamlessly, services such as Twitter and Facebook have leveraged cloud computing technologies.

1 Facebook

Facebook is probably the most evident and interesting environment in social networking. With more than 800 million users, it has become one of the largest Websites in the world.

To sustain this incredible growth, it has been fundamental that Facebook be capable of continuously adding capacity and developing new scalable technologies and software systems while maintaining high performance to ensure a smooth user experience.

Currently, the social network is backed by two data centers that have been built and optimized to reduce costs and impact on the environment.

On top of this highly efficient infrastructure, built and designed out of inexpensive hardware, a completely customized stack of opportunely modified and refined open-source technologies constitutes the back-end of the largest social network.

The reference stack serving Facebook is based on LAMP (Linux, Apache, MySQL, and PHP). This collection of technologies is accompanied by a collection of other services developed in-house.

These services are developed in a variety of languages and implement specific functionalities such as search, news feeds, notifications, and others.

While serving page requests, the social graph of the user is composed.

The social graph identifies a collection of interlinked information that is of relevance for a given user.

Most of the user data are served by querying a distributed cluster of MySQL instances, which mostly contain key-value pairs.

10.2.4 Media applications

Media applications has taken a considerable advantage from leveraging cloud computing technologies. The computationally intensive tasks can be easily offloaded to cloud computing infrastructures.

1 Animoto

Animoto is the most popular example of media applications on the cloud. The Website provides users with a very straightforward interface for quickly creating videos out of images, music, and video fragments submitted by users. Users select a specific theme for a video, upload the photos and videos and order them in the sequence they want to appear, select the song for the music, and render the video. The process is executed in the background and the user is notified via email once the video is rendered.

A proprietary artificial intelligence (AI) engine, which selects the animation and transition effects according to pictures and music, drives the rendering operation. Users only have to define the storyboard by organizing pictures and videos into the desired sequence.

The infrastructure of Animoto is complex and is composed of different systems that all need to scale (**Figure 10.8**). The core function is implemented on top of the Amazon Web Services infrastructure. It uses Amazon EC2 for the Web front-end and worker nodes; Amazon S3 for the storage of pictures, music, and videos; and Amazon SQS for connecting all the components.

The system's auto-scaling capabilities are managed by Rightscale, which monitors the load and controls the creation of new worker instances.

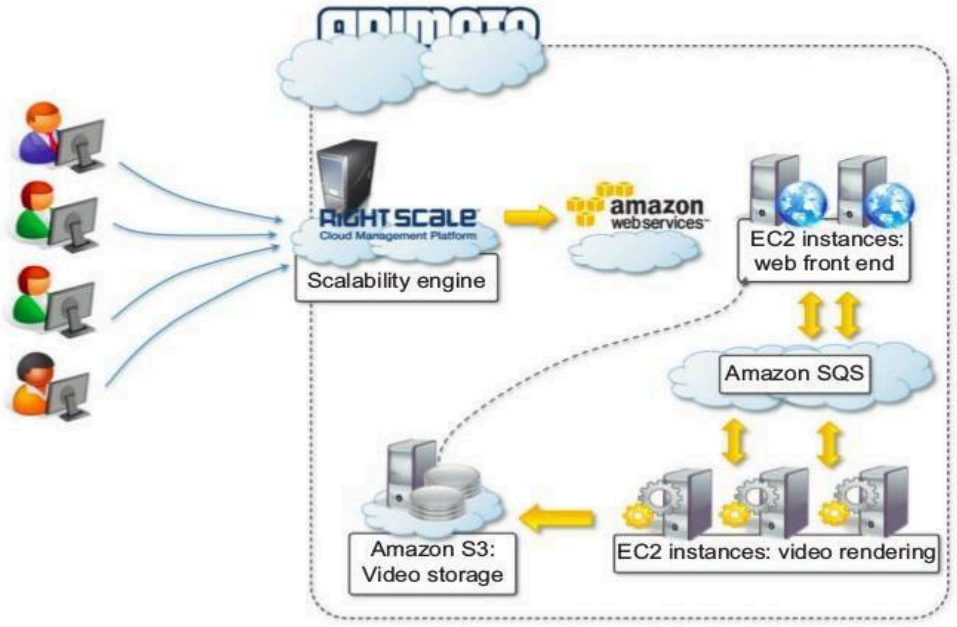

FIGURE 10.8

Animoto reference architecture.

2 Maya rendering with Aneka

A private cloud solution for rendering train designs has been implemented by the engineering department of GoFront group, a division of China Southern Railway (**Figure 10.9**). The department is responsible for designing models of high-speed electric locomotives, metro cars, urban transportation vehicles, and motor trains. The design process for prototypes requires high-quality, three-dimensional (3D) images. The analysis of these images can help engineers identify problems and correct their design.

Three-dimensional rendering tasks take considerable amounts of time, especially in the case of huge numbers of frames, but it is critical for the department to reduce the time spent in these iterations. This goal has been achieved by leveraging cloud computing technologies, which turned the network of desktops in the department into a desktop cloud managed by Aneka.

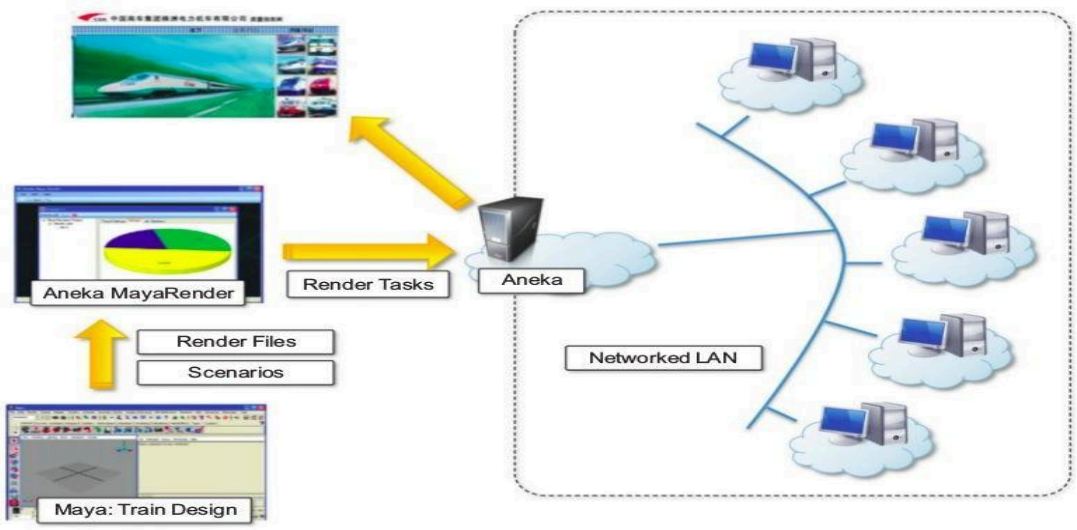

FIGURE 10.9
3D rendering on private clouds.

3 Video encoding on the cloud: Encoding.com

Video encoding and transcoding are operations that can greatly benefit from using cloud technologies: They are computationally intensive and potentially require considerable amounts of storage.

Encoding.com is a software solution that offers video-transcoding services on demand and leverages cloud technology to provide both the horsepower required for video conversion and the storage for staging videos. The service integrates with both Amazon Web Services technologies (EC2, S3, and CloudFront) and Rackspace (Cloud Servers, Cloud Files, and Limelight CDN access).

To use the service, users have to specify the location of the video to transcode, the destination format, and the target location of the video. Encoding.com also offers other video-editing operations such as the insertion of thumbnails, watermarks, or logos. Moreover, it extends its capabilities to audio and image conversion.

10.2.5 Multiplayer online gaming

Online multiplayer gaming attracts millions of gamers around the world who share a common experience by playing together in a virtual environment that extends beyond the boundaries of a normal LAN. Online games support hundreds of players in the same session, made possible by the specific architecture used to forward interactions, which is based on game log processing.

Players update the game server hosting the game session, and the server integrates all the updates into a log that is made available to all the players through a TCP port. The client software used for the game connects to the log port and, by reading the log, updates the local user interface with the actions of other players.

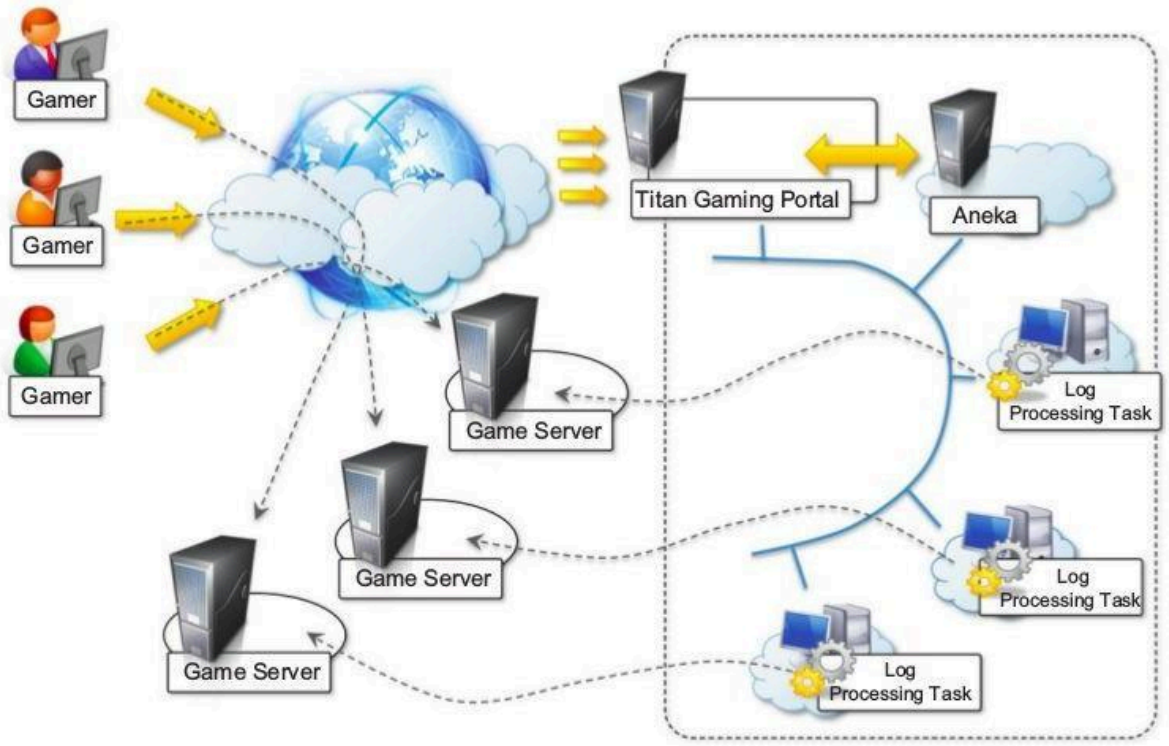

FIGURE 10.10

Scalable processing of logs for network games.

Game log processing is also utilized to build statistics on players and rank them. These features constitute the additional value of online gaming portals that attract more and more gamers. The

processing of game logs is a potentially compute-intensive operation that strongly depends on the number of players online and the number of games monitored.

The use of cloud computing technologies can provide the required elasticity for seamlessly processing these workloads and scale as required when the number of users increases. A prototype implementation of cloud-based game log processing has been implemented by Titan Inc. (**Figure 10.10**)