

Module 3.2

Google AppEngine

Google App Engine (GAE)

- Google App Engine (often referred to as GAE or simply App Engine) is a cloud computing **platform as a service for developing and hosting web applications in Google-managed data centers.**

-Definition

Google App Engine (GAE)

- Applications are **sandboxed and run across multiple servers**.
- App Engine offers **automatic scaling for web applications**—as the number of requests increases for an application,
- App Engine **automatically allocates more resources** for the web application to handle the additional demand.

-Definition

Google App Engine (GAE)

- Google has hundreds of data centers and has installed more than 460,000 servers worldwide.
- For example, 200 Google data centers are used at one time for a number of cloud applications.
- Data items are stored in text, images, and video and are replicated to tolerate faults or failures.
- Here we discuss Google's App Engine (GAE) which offers a PaaS platform supporting various cloud and web applications.

Architecture

- AppEngine is a platform for developing scalable applications accessible through the Web.
- The platform is logically divided into four major components:
 - 1) Infrastructure,
 - 2) The runtime environment,
 - 3) The underlying storage, and
 - 4) The set of scalable services that can be used to develop applications

Architecture

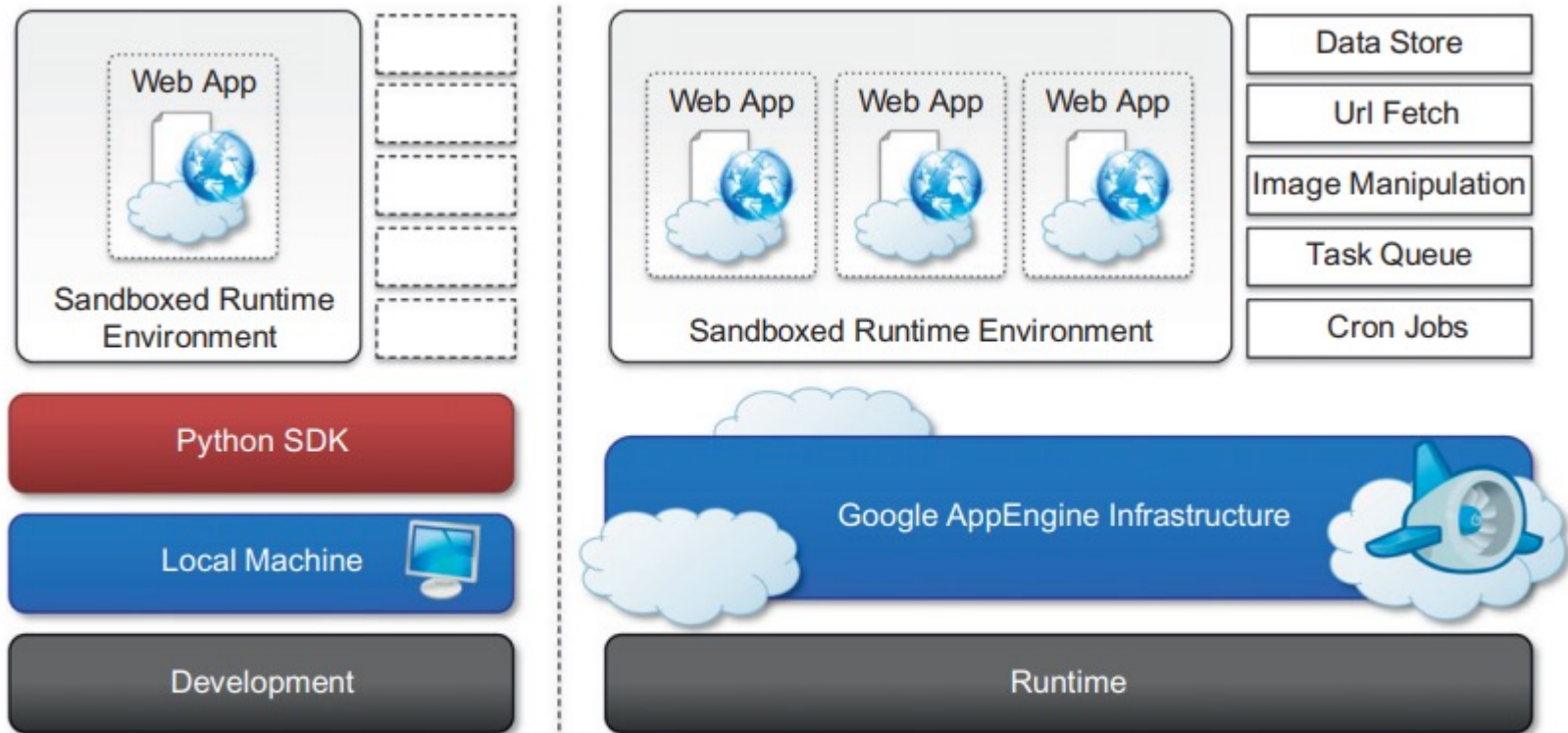


FIGURE 9.2

Google AppEngine platform architecture.

Architecture

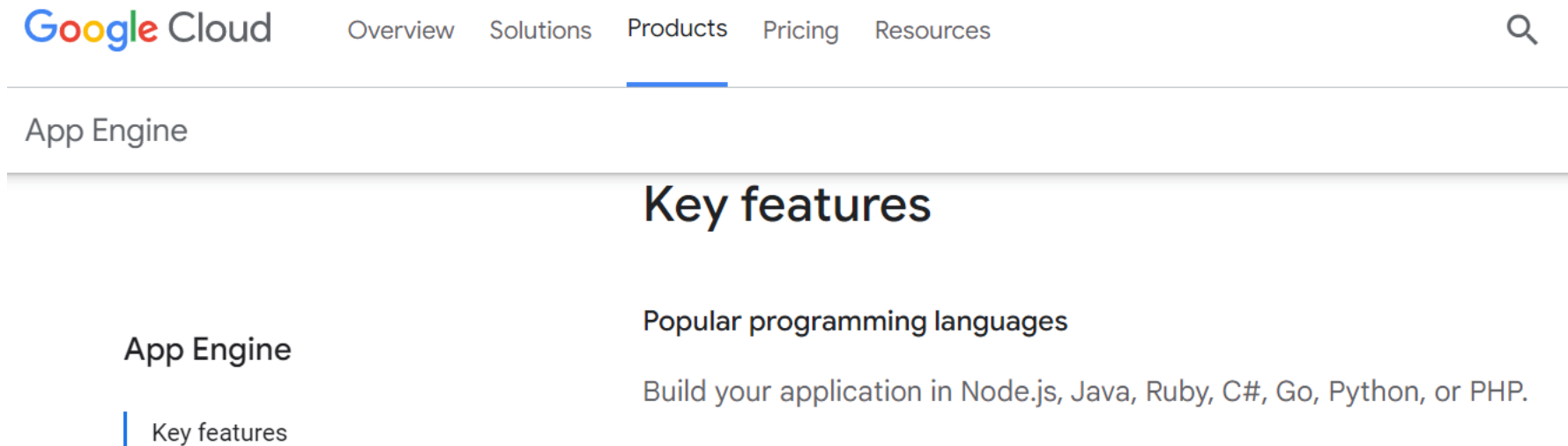
- AppEngine **hosts Web applications, and its primary function is to serve users requests efficiently.**
- To do so, AppEngine's infrastructure takes advantage of **many servers available within Google datacenters.**
- For each HTTP request, **AppEngine locates the servers hosting the application that processes the request, evaluates their load, and, if necessary, allocates additional resources (i.e., servers) or redirects the request to an existing server**

Run time Environment

- The runtime environment represents **the execution context of applications hosted on AppEngine.**
- With reference to the **AppEngine infrastructure code, which is always active and running, the runtime comes into existence when the request handler starts executing and terminates** once the handler has completed.

Supported runtimes

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



The screenshot shows the Google Cloud App Engine 'Key features' page. At the top is the Google Cloud logo and a navigation bar with links for Overview, Solutions, Products (which is underlined), Pricing, and Resources. A search icon is on the far right. Below the navigation bar is a header for 'App Engine'. The main content area is titled 'Key features' and includes a sub-section 'Popular programming languages' with the text 'Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP.' On the left side of the page, there is a sidebar with 'App Engine' and 'Key features' (indicated by a vertical blue line).

Google Cloud Overview Solutions **Products** Pricing Resources

App Engine

Key features

App Engine

Key features

Popular programming languages

Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP.

Supported runtimes

- Support for Python is provided by **an optimized Python interpreter.**
- The environment supports the Python standard library, but **some of the modules that implement potentially harmful operations have been removed, and attempts to import such modules or to call specific methods generate exceptions.**

Supported runtimes

- 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.**

Supported runtimes

- **The Go runtime environment allows applications developed with the Go programming language to be hosted and executed in AppEngine.**

Supported runtimes

App Engine

Key features

Documentation

- All features

Pricing

Popular languages

Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP.

Fully managed

A fully managed environment lets you focus on code while App Engine manages infrastructure concerns.

Sandboxing

- 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.
- GAE provides applications with a sandbox.

Sandboxing

- The isolation metaphor is taken from the idea of children who do not play well together, so each is given their own sandbox to play in alone.

Sandboxing

- Currently, AppEngine **supports applications that are developed only with interpreted or managed languages, which by design require a runtime** for translating their code into executable instructions.
- Therefore, **sandboxing is achieved by means of modified runtimes for applications that disable some of the common features normally available with their default implementations**

Sandboxing

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

Sandboxing

- Some of the operations that are not allowed in the sandbox include
- **writing to the server's file system**
- **accessing computer through network besides using Mail,UrlFetch, and XMPP**
- **executing code outside the scope of a request**
- **a queued task**
- **processing a request for more than 30 seconds.**

Storage

- AppEngine provides **various types of storage, which operate differently depending on the volatility of the data.**
- There are three different levels of storage: in memory-cache, storage for semistructured data, and long-term storage for static 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 file servers

- Static data often are mostly constituted of the **components that define the graphical layout of the application (CSS files, plain HTML files, JavaScript files, images, icons, and sound files) or data files.**
- These files **can be hosted on static file servers, since they are not frequently modified.**

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.**
- **Both the type of the key and the structure of the object can vary.**

DataStore

- With respect to the traditional Web applications backed by a relational database,
- **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 (such as reference constraints and join operations).**
- These design **decisions originated from a careful analysis of data usage patterns for Web applications** and were taken in order to obtain a more scalable and efficient data store.

DataStore

- **Datastore is a highly scalable NoSQL database for your web and mobile applications.**
- Use a managed, NoSQL, **schemaless database for storing non-relational data.**
- Cloud Datastore **automatically scales as you need it and supports transactions as well as robust, SQL-like queries- GQL(Graph Query Language)**
- The underlying infrastructure of **DataStore is based on Bigtable , a redundant, distributed, and semistructured data store that organizes data in the form of tables**

MemCache

- AppEngine provides caching services by means of 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.**

MemCache

- 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

- 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, and **in case of failed delivery the sending address is notified through an email detailing the error**.

XMPP

- AppEngine provides also another way to communicate with the external world
- **Extensible Messaging and Presence Protocol (XMPP).**

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.
- Even though the chat is a communication medium mostly used for human interactions, **XMPP can be conveniently used to connect the Web application with chat bots** or to implement a small administrative console.

Account management

- Web applications often keep various data that customize their interaction with users.
- These data normally go under the user profile and are attached to an account.
- AppEngine simplifies account management by allowing **developers to leverage Google account management by means of Google Accounts.**

Account management

- 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.**

Account management

- The use of Google Accounts is particularly **advantageous for developing Web applications within a corporate environment using Google Apps.**
- In this case, the applications **can be easily integrated with all the other services (and profile settings) included in Google Apps**

Image manipulation

- Web applications render pages with graphics. Often simple operations, **such as adding watermarks or applying simple filters, are required.**
- AppEngine allows applications to perform **image resizing, rotation, mirroring, and enhancement by means of Image Manipulation, a service that is also used in other Google products.**
- Image Manipulation is mostly designed for lightweight image processing and is optimized for speed

Google App Engine (GAE)

Google App Engine Front Page: <http://code.google.com/appengine/>

Signing up for an account or use your gmail account name : <https://appengine.google.com/>

Downloading GAE SDK : <http://code.google.com/appengine/downloads.html>

Python Getting Started Guide: <http://code.google.com/appengine/docs/python/gettingstarted/>

Java Getting Started Guide: <http://code.google.com/appengine/docs/java/gettingstarted/>

Quota page for free service: <http://code.google.com/appengine/docs/quotas.html#Resources>

Billing page if you go over the quota:

[http://code.google.com/appengine/docs/billing.html#Billable Quota Unit Cost](http://code.google.com/appengine/docs/billing.html#Billable_Quota_Unit_Cost)

GAE

- 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.

GAE Applications

- Well-known GAE applications include the [Google Search Engine, Google Docs, Google Earth, and Gmail](#).
- These applications can support large numbers of users simultaneously.
- 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 also provides [Google-specific services, such as the Gmail account service \(which is the login service, that is, applications can use the Gmail account directly\)](#).