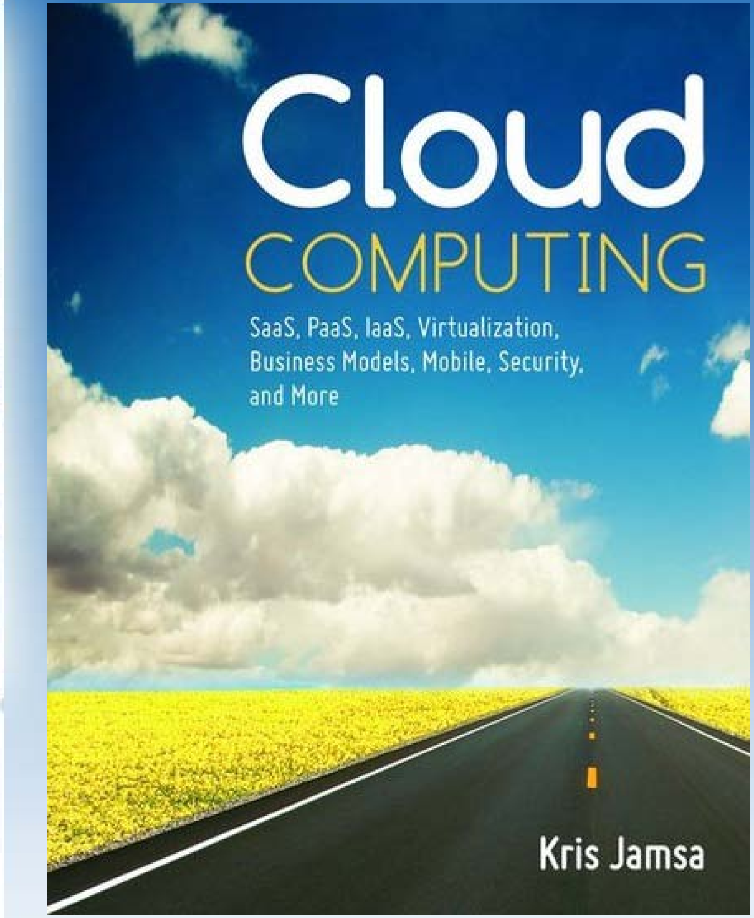
Cloud Computing

Chapter 3

Platform as a Service (PaaS)



Cloud
COMPUTING

SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security, and More

Kris Jamsa

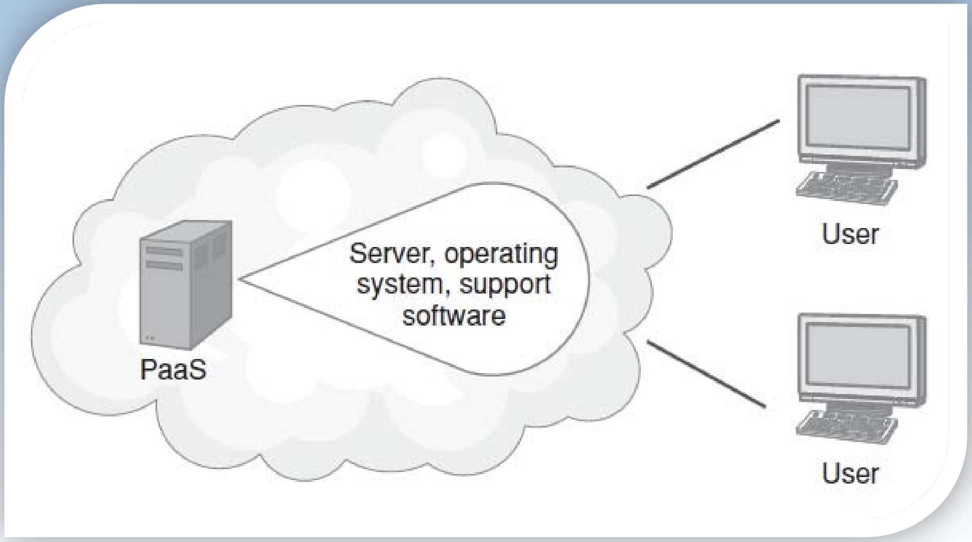
JONES & BARTLETT LEARNING
An Ascend Learning Company

Learning Objectives

- Define and describe the PaaS model.
- Describe the advantages and disadvantages of PaaS solutions.
- List and describe several real-world PaaS solutions.
- List and describe cloud-based database solutions and describe their advantages.
- Discuss the development history that led to PaaS.

Platform as a Service (PaaS)

- Provide a **collection of hardware and software resources** that developers can use to **build and deploy applications** within the cloud.
- Depending on their needs, developers may use a Windows-based PaaS solution or a Linux-based PaaS.



Advantages

- Developers do not need to buy and maintain hardware, and install and manage operating system and database software.
- Computing resources no longer reside in the data center, but rather in the cloud,
  - the resources can scale on demand
  - the company can pay for only resources it consumes.
- Further, because PaaS eliminates the developers' need to worry about servers, they can more quickly deploy their web-based solutions.

Disadvantages

- Some developers and administrators want finer control over the underlying systems (versions, patch releases/applications, …)

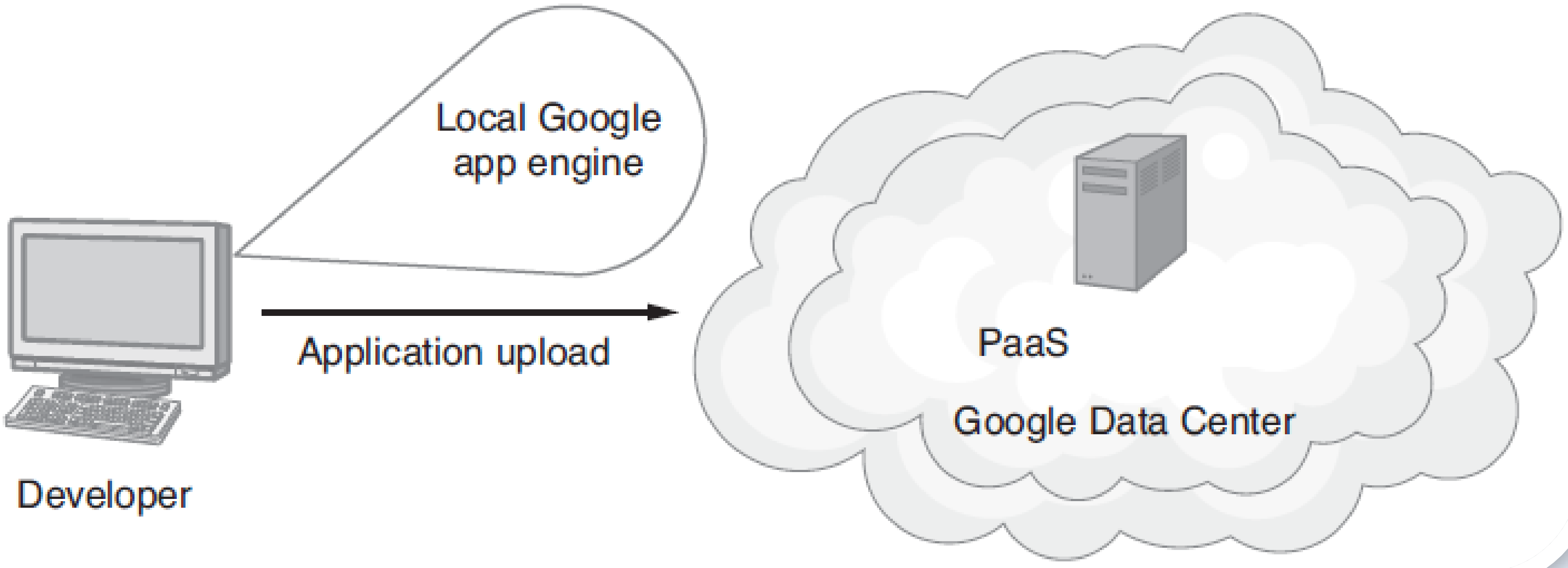
Real World: Google App Engine

- **Google App Engine (GAE)**, is a PaaS solution.
  - Developers create and host web-based applications that reside and run on services managed by Google.
- Google App Engine provides platform support for a variety of programming languages
  - Java, Python, and Go.
- Google App Engine is a free service.

Google App Engine Continued

- Google App Engine features include the following:
  - Support for dynamic web pages
  - Data storage and query support
  - Load balancing for application scalability
  - Application program interface (API) support for application-based e-mail through Google services
  - A local development environment that simulates Google App Engine on the developer's computer
  - Support for event scheduling and triggering
  - An application sandbox that limits access to the underlying operating system
  - An administrative console for managing applications

Google App Engine



Google App Engine (Supplement)

GUIDO VAN ROSSUM

STANFORD EE380 COLLOQUIUM, NOV 5, 2008

Features

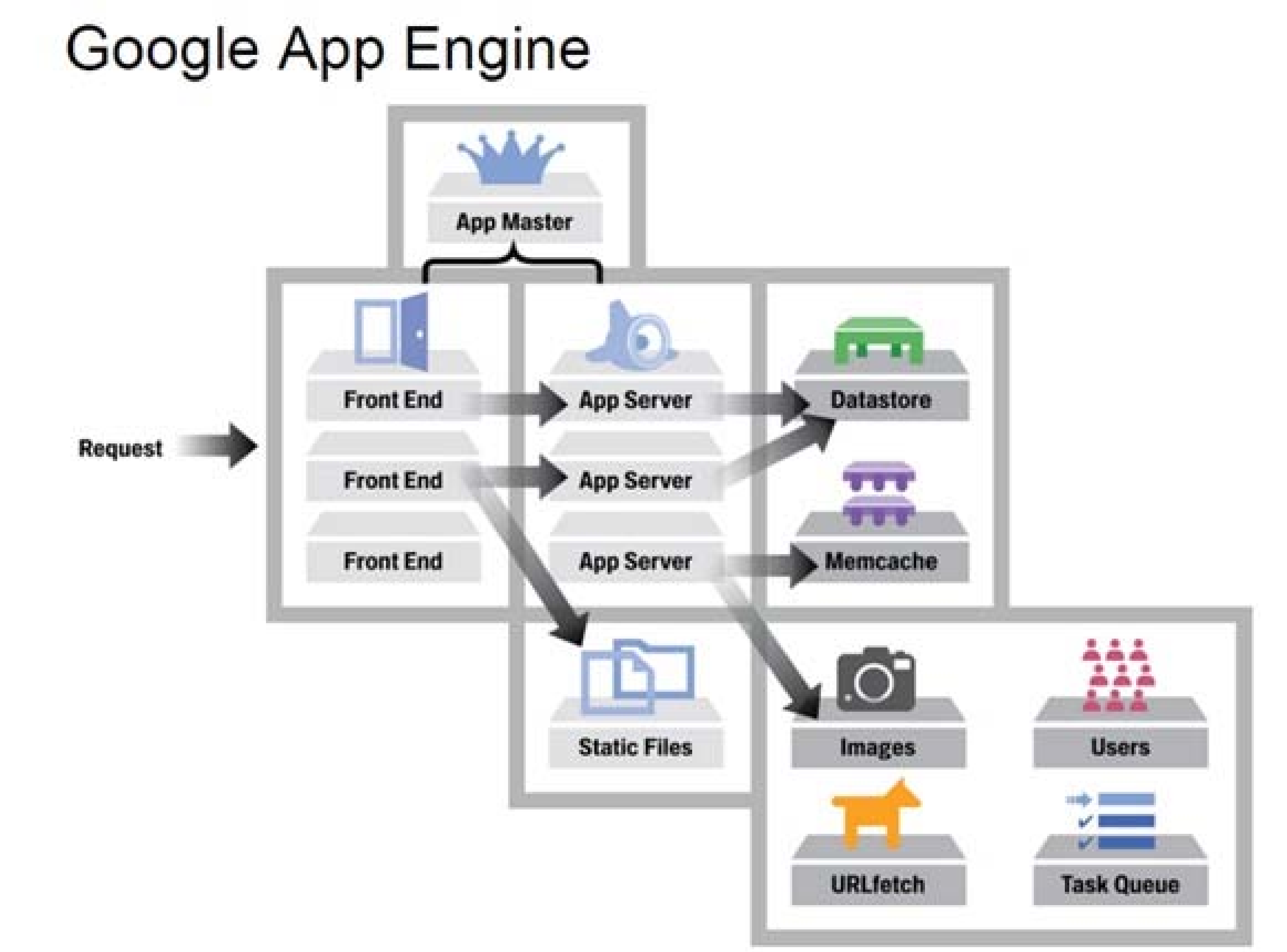
- Does one thing well: running web apps
- Simple app configuration
- Scalable
- Secure

GAE Does One Thing Well

- App Engine handles HTTP(S) requests, nothing else
  - Request in, processing, response out
  - Works well for the web and AJAX; also for other services
- App configuration is very simple
  - No performance tuning needed
- Everything is built to scale
  - “infinite” number of apps, requests/sec, storage capacity
  - APIs are simple

AJAX: Asynchronous JavaScript and XML

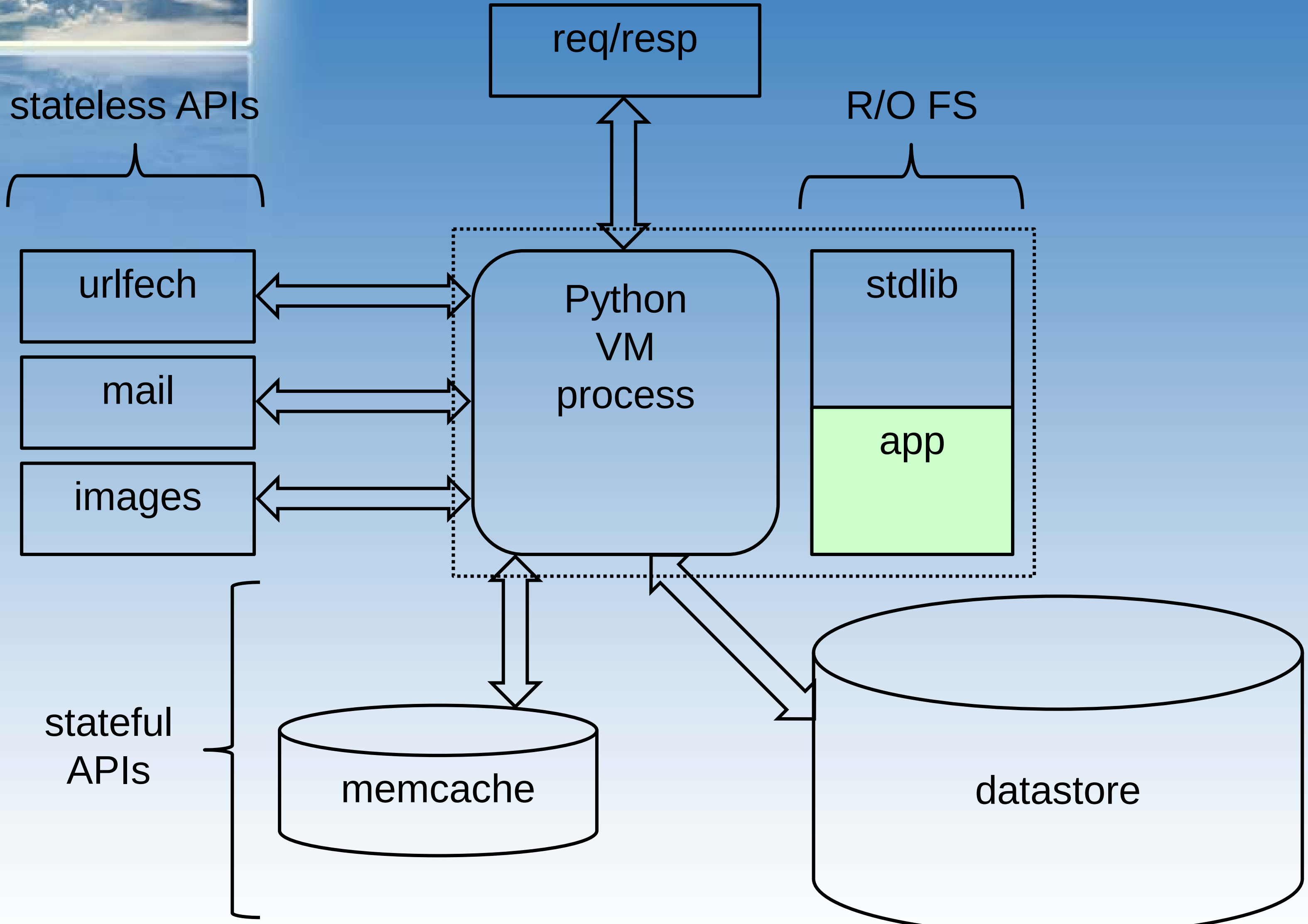
GAE Architecture



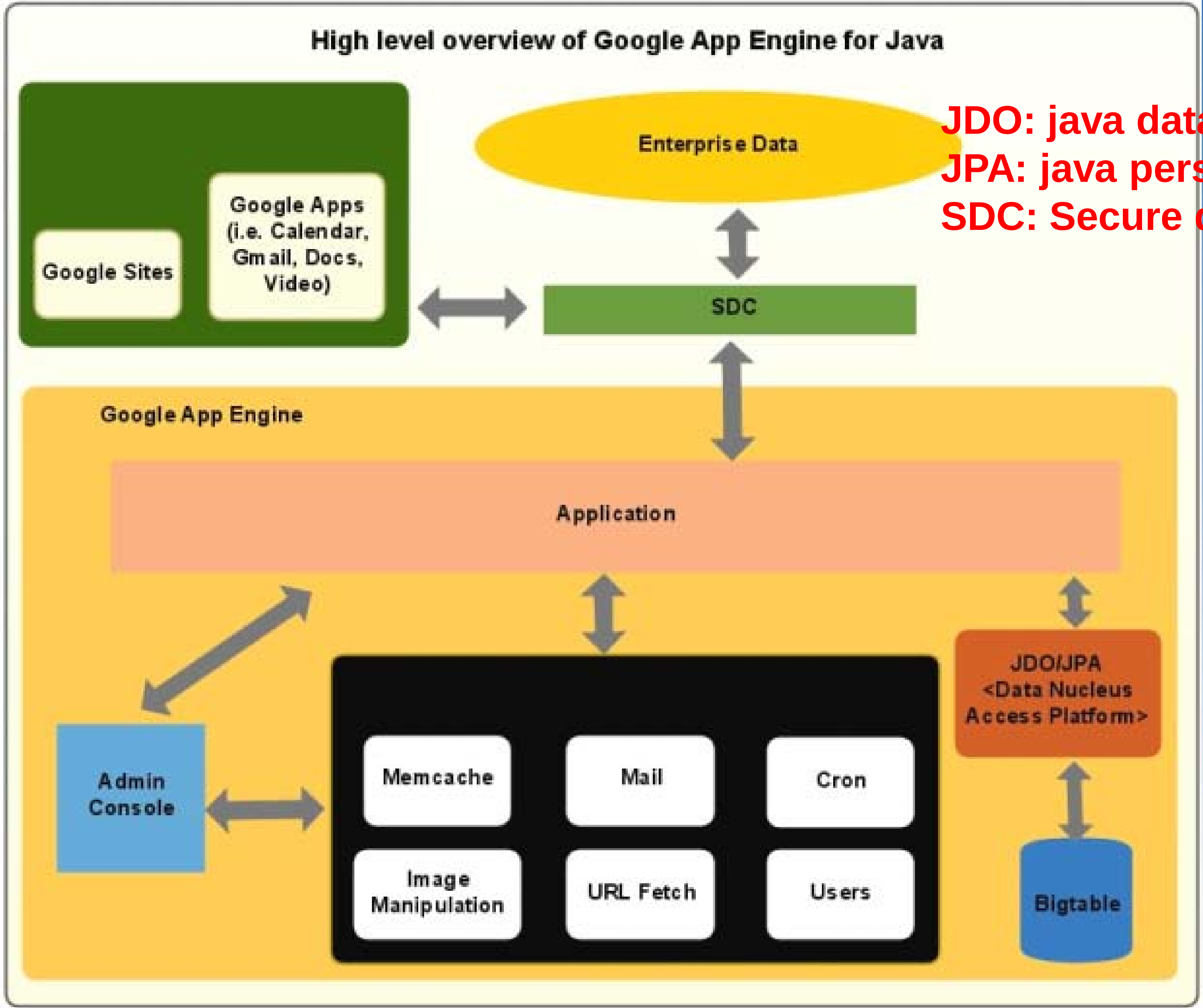
Services

- URLFetch: fetch web resources/services
- Images: manipulate images; resize, rotate, flip, crop
- Google Accounts
- Mail
- Extensible Messaging and Presence Protocol (XMPP): instant messages
- Task Queue: message queue; allow integration with non-GAPPs (Google Apps)
- Datastore: managing data objects
- Blobstore: large files, much larger than objects in datastore, use <key, object> to access

GAE Architecture (python)



GAE Architecture (Java)


High level overview of Google App Engine for Java
Google Sites
Google Apps (i.e. Calendar, Gmail, Docs, Video)
Enterprise Data
SDC
Google App Engine
Application
Admin Console
Memcache
Mail
Cron
Image Manipulation
URL Fetch
Users
JDO/JPA <Data Nucleus Access Platform>
Bigtable

JDO: java data object
JPA: java persistent API
SDC: Secure data connector

Java or python?

- Python: powerful python syntax, library, shorter code
- Java: can use JDO/JPA
  - Better portability if you need to use Bigtable to store data

Java Data Objects (JDO)
JavaPersistence API (JPA)

Why Not LAMP?

- Linux, Apache, MySQL/PostgreSQL (LAMP), Python/Perl/PHP/Ruby
- LAMP is the industry standard
- But management is a hassle:
  - Configuration, tuning
  - Backup and recovery, disk space management
  - Hardware failures, system crashes
  - Software updates, security patches
  - Redesign needed once your database exceeds one box

“**We carry pagers so you don’t have to**”

Scaling

- Low-usage apps: many apps per physical host
- High-usage apps: multiple physical hosts per app

- Stateless APIs are trivial to replicate
- **Datastore built on top of Bigtable**; designed to scale well
  - Abstraction on ***top*** of Bigtable
  - API influenced by scalability

Automatic Scaling to Application Needs

- You don’t need to configure your resource needs
- One CPU can handle many requests per second
- Apps are hashed onto CPUs:
  - One process per app, many apps per CPU
  - Creating a new process clones a generic “model” process and then loading the application code (in fact the clones are pre-created and sit in a queue)
  - The process (handle process) hangs around to handle more requests (reuse)
  - Eventually old processes are killed (recycle)
- Busy apps (many QPS (query per sec)) get assigned to multiple CPUs

Preserving **Fairness** Through Quotas

- An app is limited by **quotas**, for example:
  - request count, bandwidth used, CPU usage, datastore call count, disk space used, emails sent, even errors!
- If you **run out of quota** that **particular operation is blocked** (raising an exception) for a while (~10 min) until replenished
- Free quotas are tuned so that a well-written app (light CPU/datastore use) can survive a moderate "***slashdotting***"
  - **Slashdotting:** when a popular website links to a smaller site, causing a massive increase in traffic.
  - This overloads the smaller site, causing it to slow down or even temporarily become unavailable.

Preserving **Fairness** Through Quotas

- The point of quotas is to be able to support a very large number of small apps (analogy: baggage limit in air travel)

- Large apps need raised quotas
  - currently this is a manual process (search FAQ for “quota”)
  - in the future you can buy more resources

FAQ(Frequently Asked Questions)

Datastore (storage organization)

- Data model
  - Property, entity, entity group
  - Schemeless: properties can have different types/meanings for different objects
  - Allow (1) object query (2) SQL-like query
- Transaction
  - Can be applied to a group of operations
- Persistent store (check BigTable)
  - Strongly consistent
  - Not relational database
  - Index built-in
- Memcache
  - Caches objects from bigtable to improve performance

Hierarchical Datastore

- ***Entities*** have a ***Kind***, a ***Key***, and ***Properties***
  - Entity --> Record --> Python dict --> Python class instance
  - Key --> structured foreign key; includes Kind
  - Kind --> Table --> Python class
  - Property --> Column or Field; has a type

- Key has either *id* or *name*
  - id is auto-assigned; name is set by app

- Dynamically typed: Property types are recorded per Entity

- Paths define *entity groups* which limit *transactions*

Indexes

- Properties are automatically indexed by **type + value**
  - There is an index for each Kind / property name combo
  - Whenever an entity is written all relevant indexes are updated
  - However Blob and Text properties are never indexed
- This supports basic queries: AND on property equality
- For more advanced query needs, create *composite indexes*
  - SDK auto-updates **index.yaml** based on queries executed
  - These support inequalities (<, <=, >, >=) and result ordering
  - Index building has to scan *all* entities due to parent keys

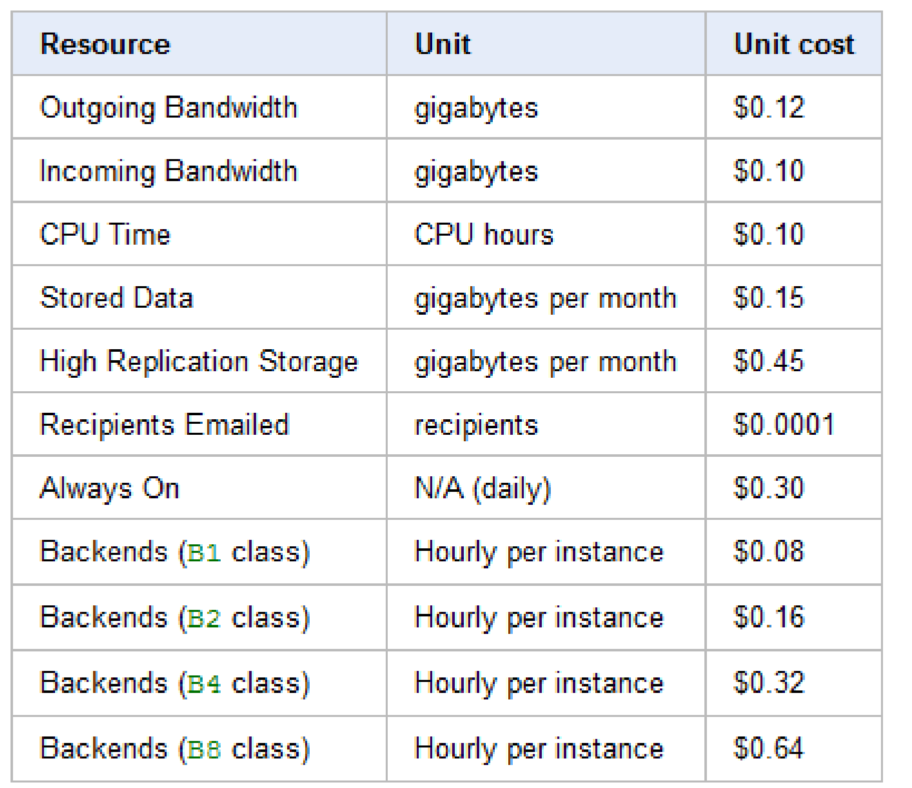
- For more info, see video of Ryan Barrett's talk at Google I/O
  - https://www.youtube.com/watch?v=tx5gdoNpcZM

index.yaml

- Every datastore query made by an application needs a corresponding index.
- Indexes for simple queries, such as queries over a single property, are created automatically.
- Indexes for complex queries must be defined in a configuration file named index.yaml.
  - This file is uploaded with the application to create indexes in the datastore.

Pricing

- Free quota
  - – 1 GB of persistent storage
  - – Enough CPU and bandwidth for about 5 million page views a month.
- User defined budget



Resource	Unit	Unit cost
Outgoing Bandwidth	gigabytes	$0.12
Incoming Bandwidth	gigabytes	$0.10
CPU Time	CPU hours	$0.10
Stored Data	gigabytes per month	$0.15
High Replication Storage	gigabytes per month	$0.45
Recipients Emailed	recipients	$0.0001
Always On	N/A (daily)	$0.30
Backends (B1 class)	Hourly per instance	$0.08
Backends (B2 class)	Hourly per instance	$0.16
Backends (B4 class)	Hourly per instance	$0.32
Backends (B8 class)	Hourly per instance	$0.64

Security

- Prevent the bad guys breaking into your app

- Constrain direct OS functionality
  - no processes, threads, dynamic library loading
  - no sockets (use urlfetch API)
  - can't write files (use datastore)
  - disallow unsafe Python extensions (e.g. ctypes)

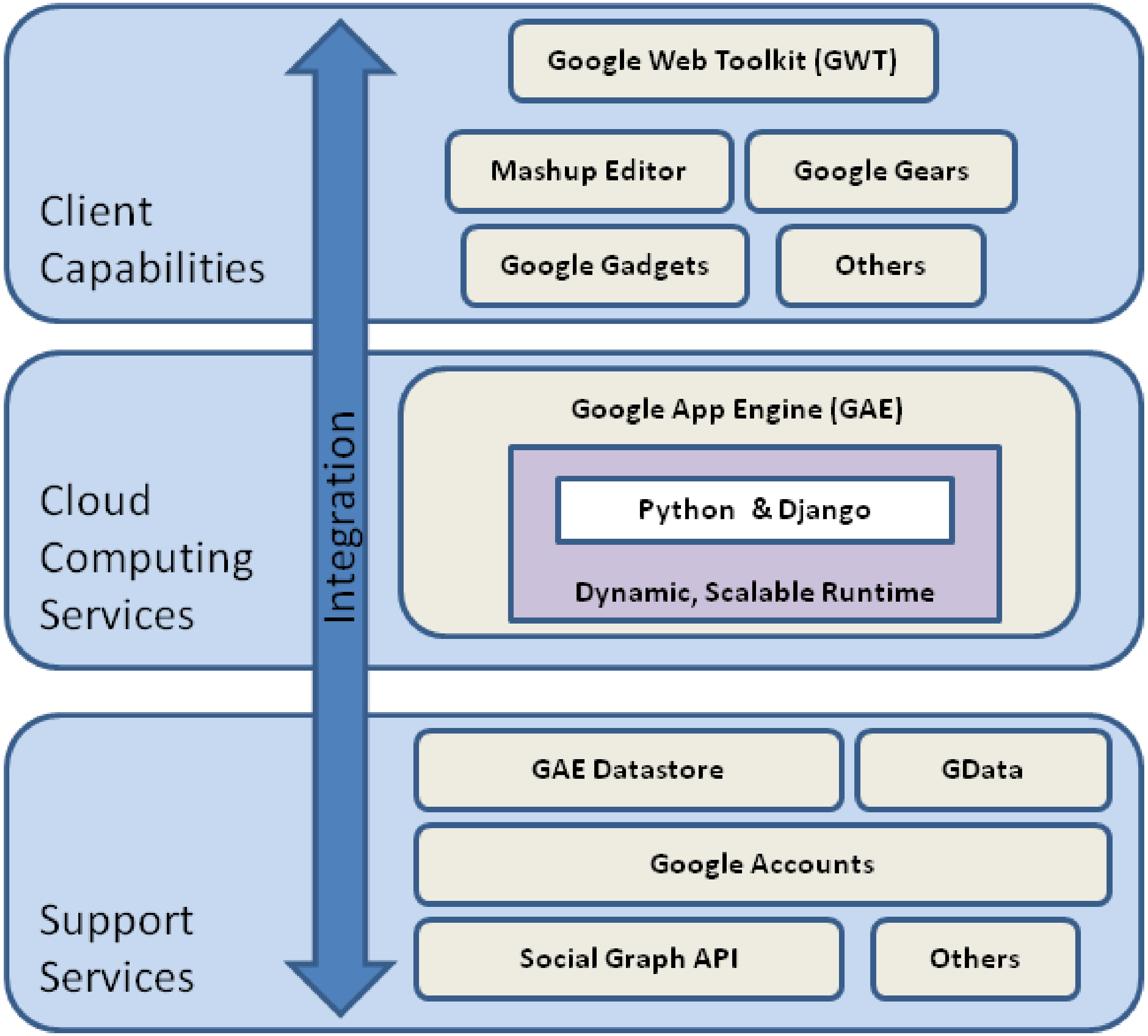
- Limit resource usage
  - Hard time limit of 30 seconds per request
  - Most requests must use less than 300 msec CPU time
  - Hard limit of 1MB on request/response size, API call size, etc.
  - Quota system for number of requests, API calls, emails sent, etc
  - Free use for 500MB data and 5M requests per month
  - 10 applications per account

The Future

- Big things we’re working on:
  - Large file uploads and downloads
  - Datastore import and export for large volumes
  - Pay-as-you-go billing (for resource usage over free quota)
  - More languages
  - Uptime monitoring site

- No published timeline – agile development process

Google App Engine



Client Capabilities

Google Web Toolkit (GWT)

Mashup Editor

Google Gears

Google Gadgets

Others

Integration

Cloud Computing Services

Google App Engine (GAE)

Python & Django

Dynamic, Scalable Runtime

Support Services

GAE Datastore

GData

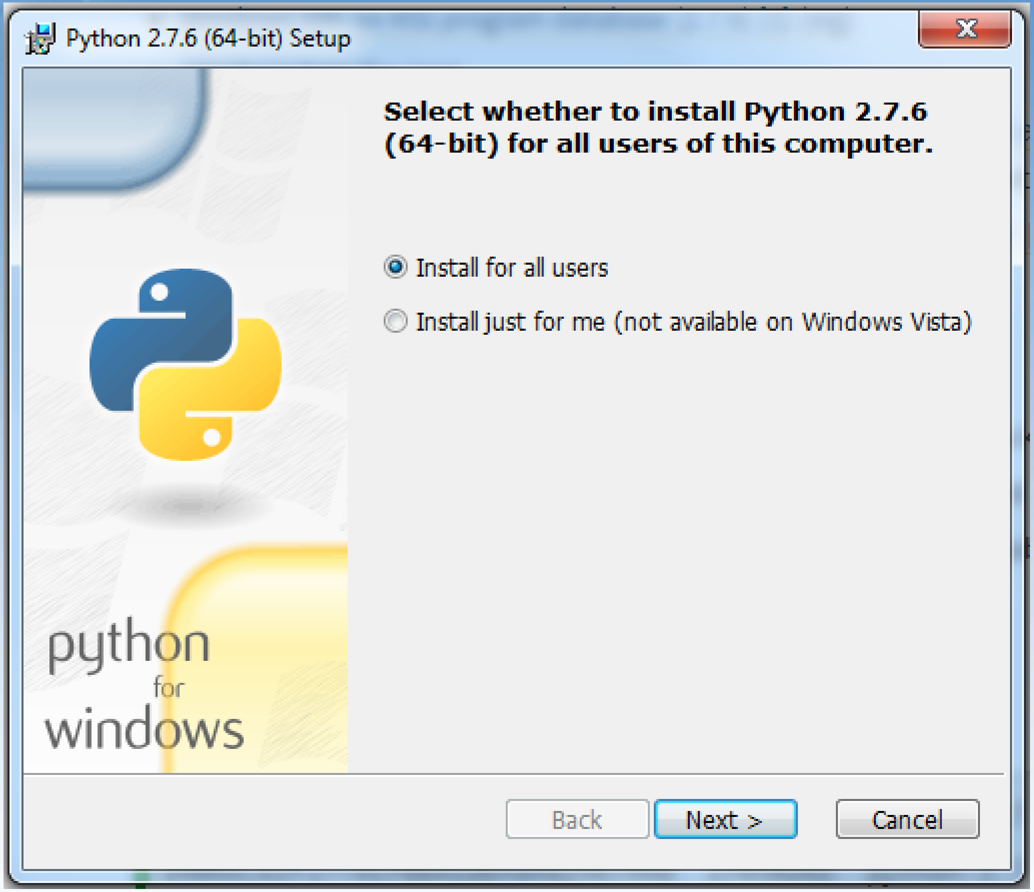
Google Accounts

Social Graph API

Others

Install Python

- http://www.python.org/download/



Python 2.7.6 (64-bit) Setup

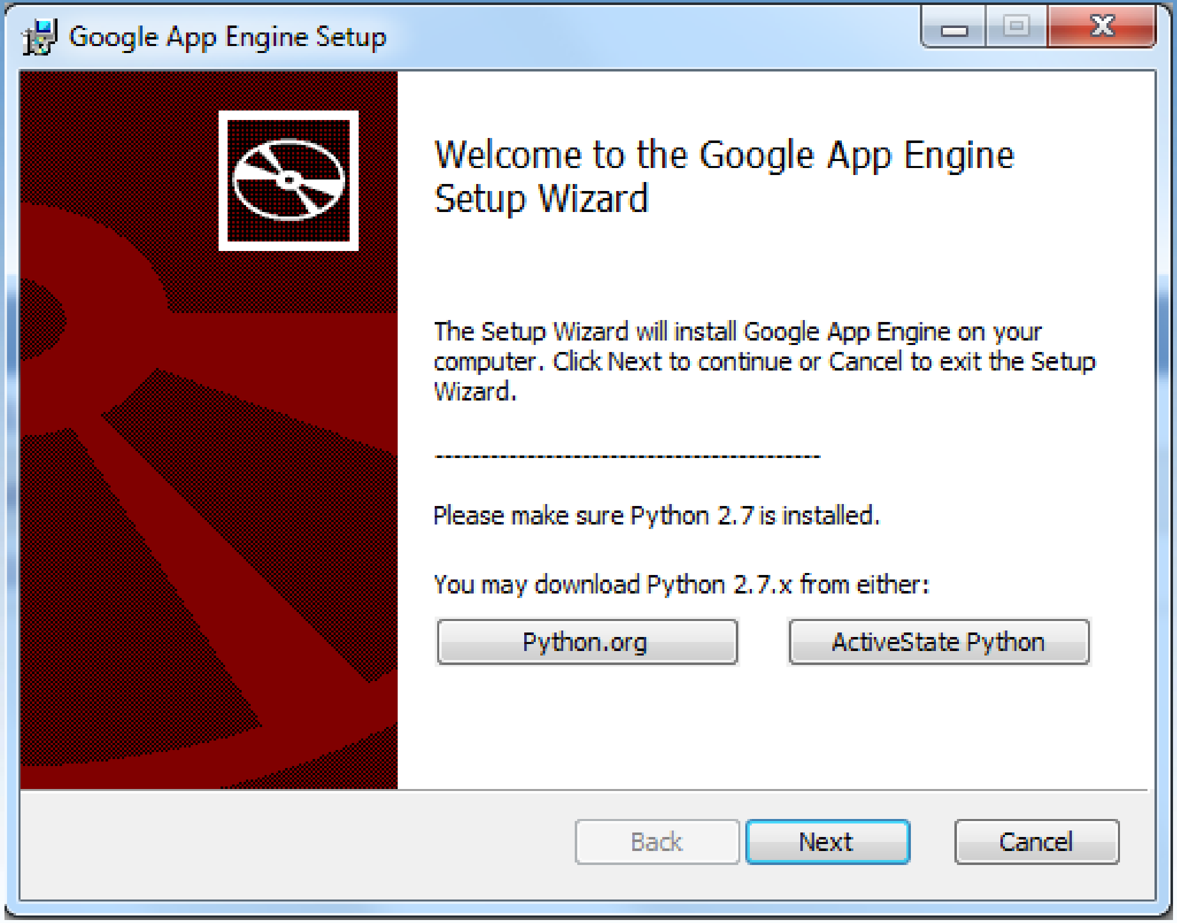
Select whether to install Python 2.7.6 (64-bit) for all users of this computer.

● Install for all users

○ Install just for me (not available on Windows Vista)

Install Google App Engine

- https://developers.google.com/appengine/downloads?csw=1



Google App Engine Setup

Welcome to the Google App Engine Setup Wizard

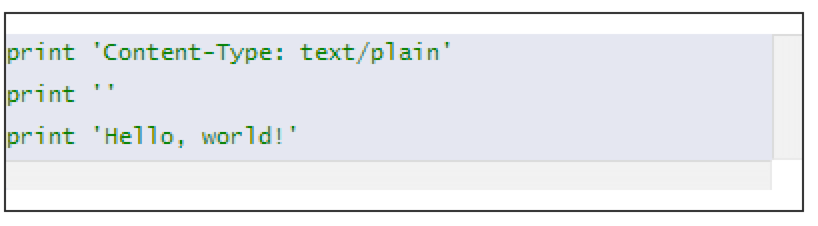
The Setup Wizard will install Google App Engine on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Please make sure Python 2.7 is installed.

You may download Python 2.7.x from either:

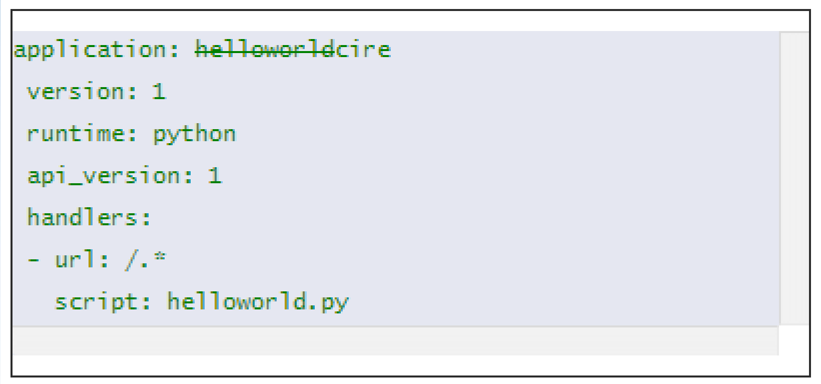
Hello World

- helloworld.py

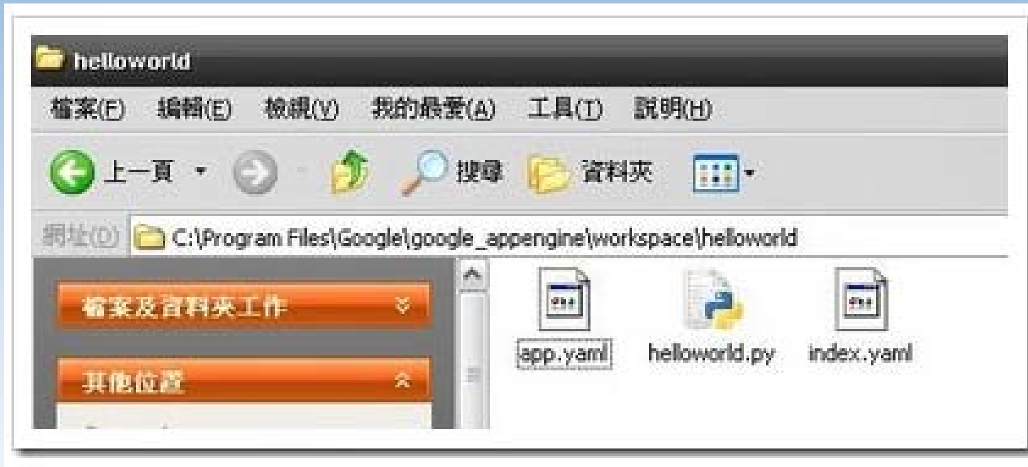


```
print 'Content-Type: text/plain'
print ''
print 'Hello, world!'
```

- app.yaml

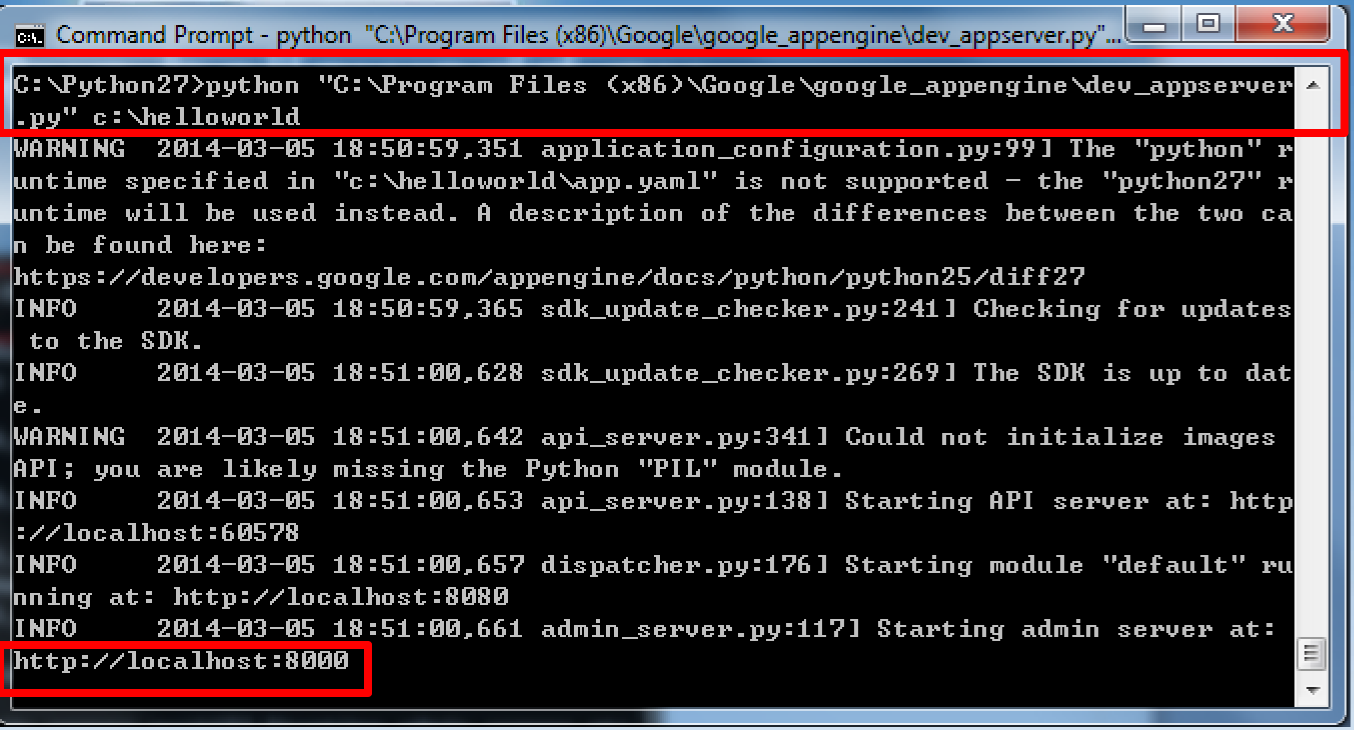


```
application: ~~helloworld~~cire
 version: 1
 runtime: python
 api_version: 1
 handlers:
 - url: /.*
   script: helloworld.py
```



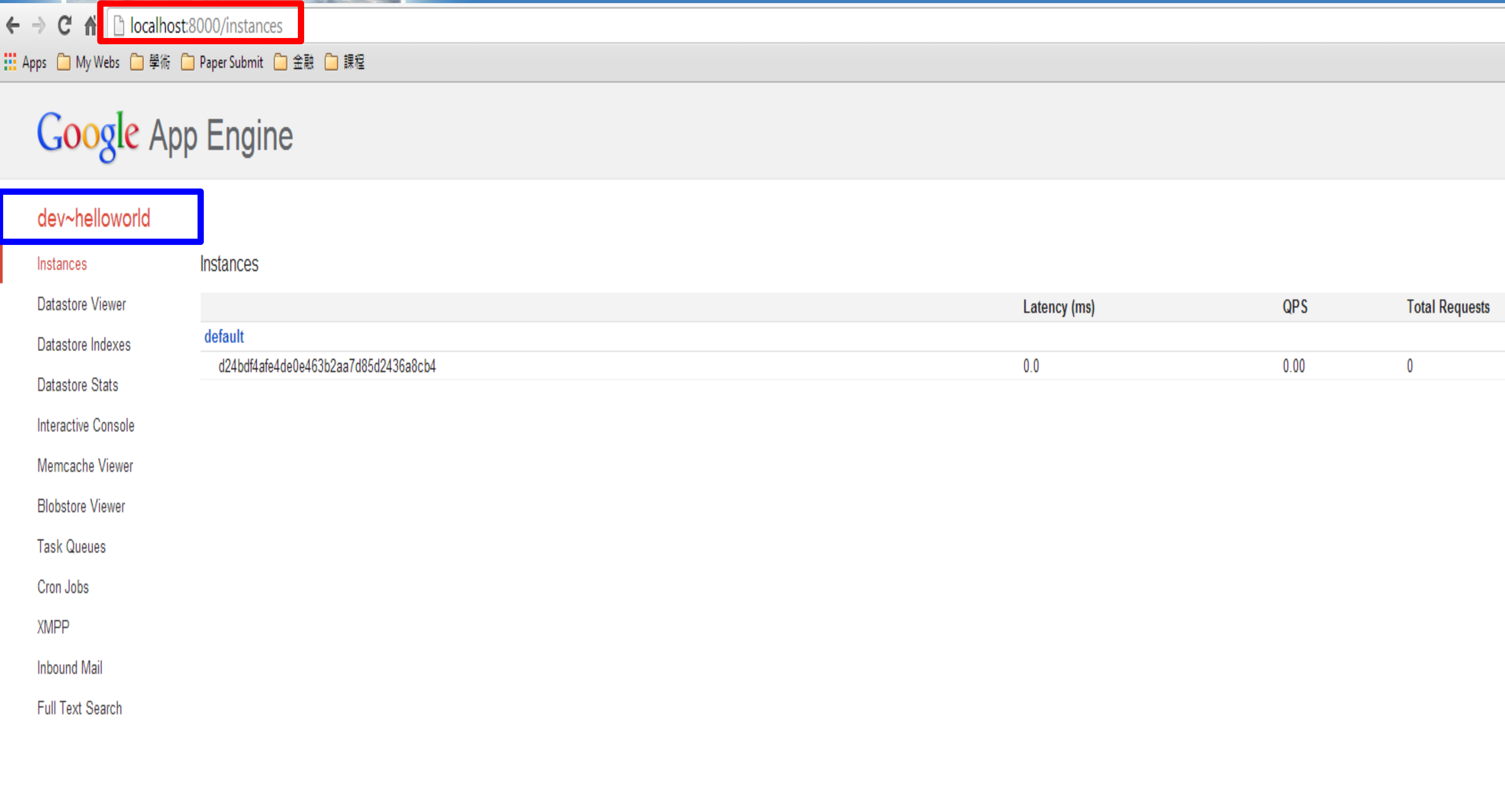
網址(D) C:\Program Files\Google\google_appengine\workspace\helloworld

Run helloworld.py


Command Prompt - python "C:\Program Files (x86)\Google\google_appengine\dev_appserver.py"...

```
C:\Python27>python "C:\Program Files (x86)\Google\google_appengine\dev_appserver
.py" c:\helloworld
WARNING  2014-03-05 18:50:59,351 application_configuration.py:99] The "python" r
untime specified in "c:\helloworld\app.yaml" is not supported - the "python27" r
untime will be used instead. A description of the differences between the two ca
n be found here:
https://developers.google.com/appengine/docs/python/python25/diff27
INFO     2014-03-05 18:50:59,365 sdk_update_checker.py:241] Checking for updates
 to the SDK.
INFO     2014-03-05 18:51:00,628 sdk_update_checker.py:269] The SDK is up to dat
e.
WARNING  2014-03-05 18:51:00,642 api_server.py:341] Could not initialize images
API; you are likely missing the Python "PIL" module.
INFO     2014-03-05 18:51:00,653 api_server.py:138] Starting API server at: http
://localhost:60578
INFO     2014-03-05 18:51:00,657 dispatcher.py:176] Starting module "default" ru
nning at: http://localhost:8080
INFO     2014-03-05 18:51:00,661 admin_server.py:117] Starting admin server at:
http://localhost:8000
```

Run helloworld.py



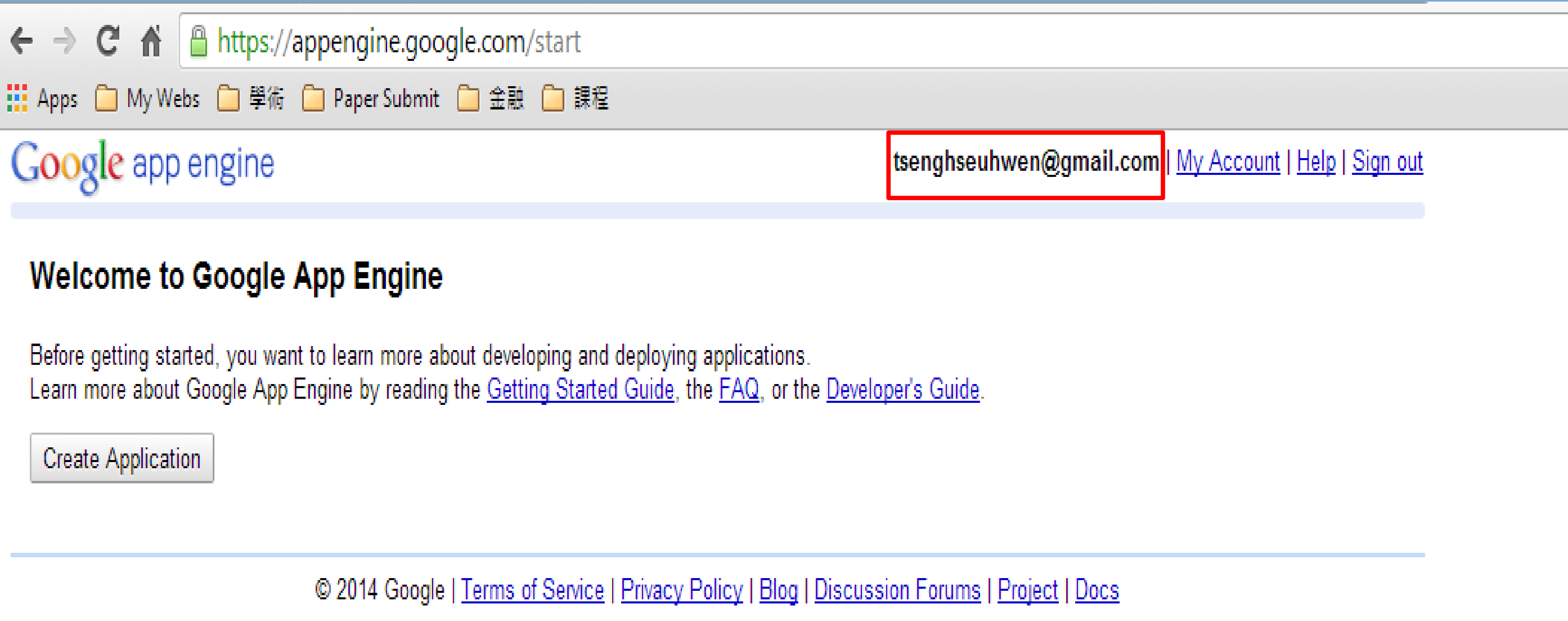
localhost:8000/instances

Instances

	Latency (ms)	QPS	Total Requests
default			
d24bdf4afe4de0e463b2aa7d85d2436a8cb4	0.0	0.00	0

Create an Application

- https://sites.google.com/site/gdevelopercodelabs/app-engine/creating-your-app-engine-account



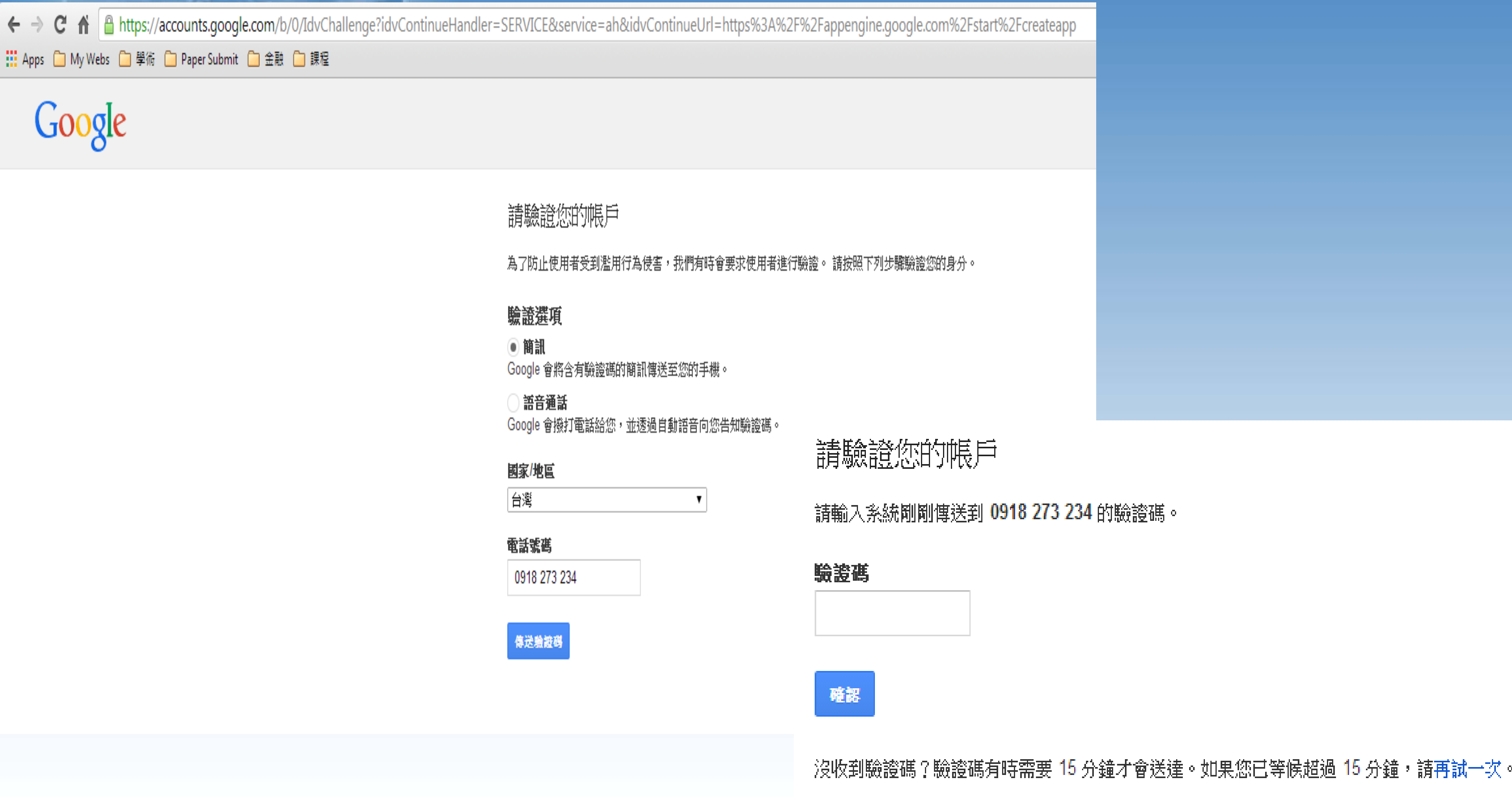
tsenghseuhwen@gmail.com | My Account | Help | Sign out

Welcome to Google App Engine

Before getting started, you want to learn more about developing and deploying applications.
Learn more about Google App Engine by reading the Getting Started Guide, the FAQ, or the Developer's Guide.

© 2014 Google | Terms of Service | Privacy Policy | Blog | Discussion Forums | Project | Docs

Create an Application



https://accounts.google.com/b/0/IdvChallenge?idvContinueHandler=SERVICE&service=ah&idvContinueUrl=https%3A%2F%2Fappengine.google.com%2Fstart%2Fcreateapp

請驗證您的帳戶

為了防止使用者受到濫用行為侵害，我們有時會要求使用者進行驗證。 請按照下列步驟驗證您的身分。

驗證選項

● **簡訊**
Google 會將含有驗證碼的簡訊傳送至您的手機。

○ **語音通話**
Google 會撥打電話給您，並透過自動語音向您告知驗證碼。

國家/地區
台灣 ▾

電話號碼
0918 273 234

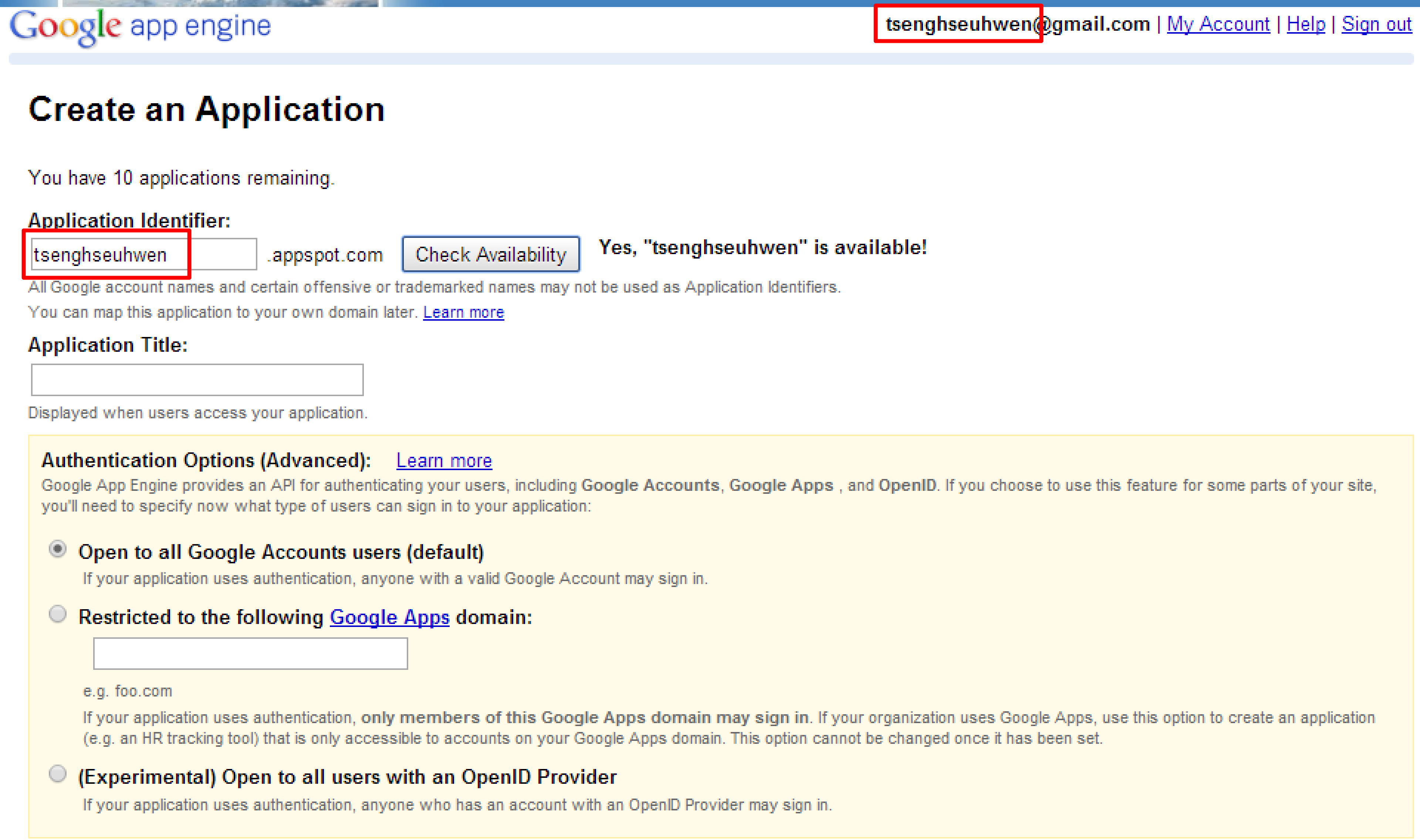
請驗證您的帳戶

請輸入系統剛剛傳送到 0918 273 234 的驗證碼。

驗證碼

沒收到驗證碼？驗證碼有時需要 15 分鐘才會送達。如果您已等候超過 15 分鐘，請再試一次。

Create an Application



Google app engine

tsenghseuhwen@gmail.com | My Account | Help | Sign out

Create an Application

You have 10 applications remaining.

Application Identifier:

tsenghseuhwen .appspot.com [Check Availability] **Yes, "tsenghseuhwen" is available!**

All Google account names and certain offensive or trademarked names may not be used as Application Identifiers.

You can map this application to your own domain later. Learn more

Application Title:

Displayed when users access your application.

Authentication Options (Advanced): Learn more

Google App Engine provides an API for authenticating your users, including **Google Accounts**, **Google Apps** , and **OpenID**. If you choose to use this feature for some parts of your site, you'll need to specify now what type of users can sign in to your application:

● **Open to all Google Accounts users (default)**

If your application uses authentication, anyone with a valid Google Account may sign in.

○ **Restricted to the following Google Apps domain:**

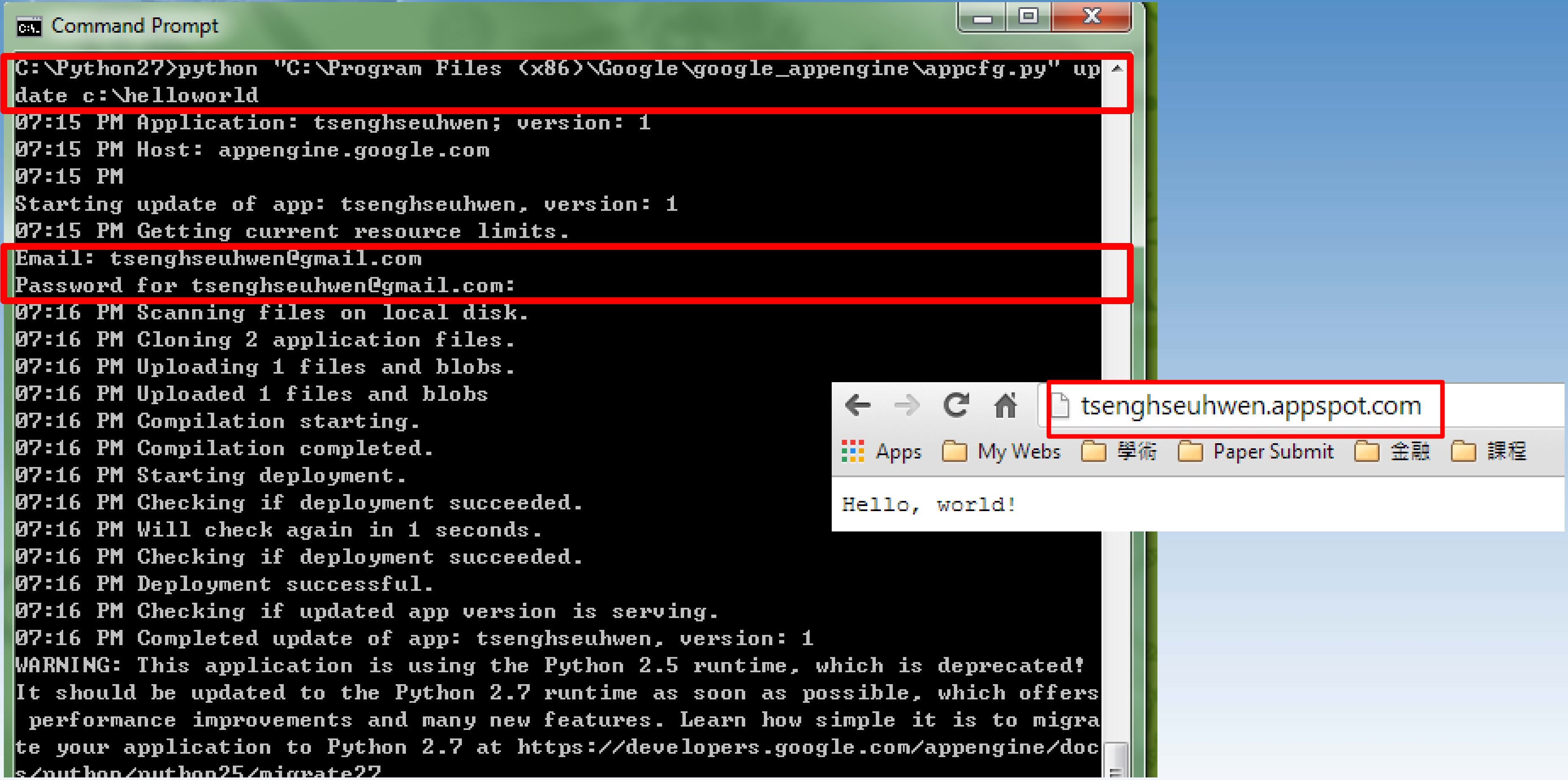
e.g. foo.com

If your application uses authentication, **only members of this Google Apps domain may sign in**. If your organization uses Google Apps, use this option to create an application (e.g. an HR tracking tool) that is only accessible to accounts on your Google Apps domain. This option cannot be changed once it has been set.

○ **(Experimental) Open to all users with an OpenID Provider**

If your application uses authentication, anyone who has an account with an OpenID Provider may sign in.

Upload helloworld

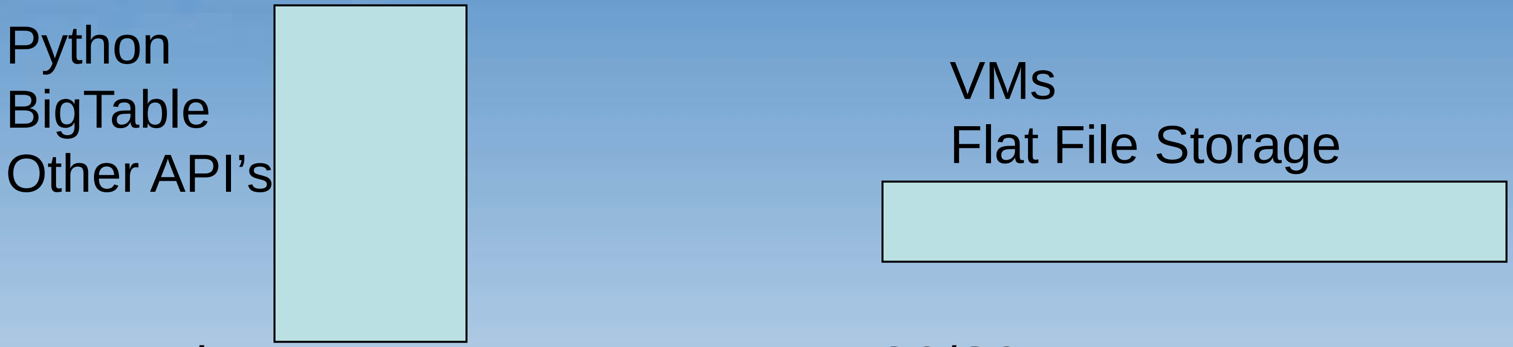


```
C:\Python27>python "C:\Program Files (x86)\Google\google_appengine\appcfg.py" up
date c:\helloworld
07:15 PM Application: tsenghseuhwen; version: 1
07:15 PM Host: appengine.google.com
07:15 PM
Starting update of app: tsenghseuhwen, version: 1
07:15 PM Getting current resource limits.
Email: tsenghseuhwen@gmail.com
Password for tsenghseuhwen@gmail.com:
07:16 PM Scanning files on local disk.
07:16 PM Cloning 2 application files.
07:16 PM Uploading 1 files and blobs.
07:16 PM Uploaded 1 files and blobs
07:16 PM Compilation starting.
07:16 PM Compilation completed.
07:16 PM Starting deployment.
07:16 PM Checking if deployment succeeded.
07:16 PM Will check again in 1 seconds.
07:16 PM Checking if deployment succeeded.
07:16 PM Deployment successful.
07:16 PM Checking if updated app version is serving.
07:16 PM Completed update of app: tsenghseuhwen, version: 1
WARNING: This application is using the Python 2.5 runtime, which is deprecated!
It should be updated to the Python 2.7 runtime as soon as possible, which offers
 performance improvements and many new features. Learn how simple it is to migra
te your application to Python 2.7 at https://developers.google.com/appengine/doc
s/python/python25/migrate27
```

tsenghseuhwen.appspot.com

```
Hello, world!
```

Comparing Google AppEngine and Amazon EC2



Python
BigTable
Other API's

VMs
Flat File Storage

AppEngine:

- Higher-level functionality (e.g., automatic scaling)
- More restrictive (e.g., respond to URL only)
- Proprietary lock-in

EC2/S3:

- Lower-level functionality
- More flexible
- Coarser billing model

Will The Two Models Converge?

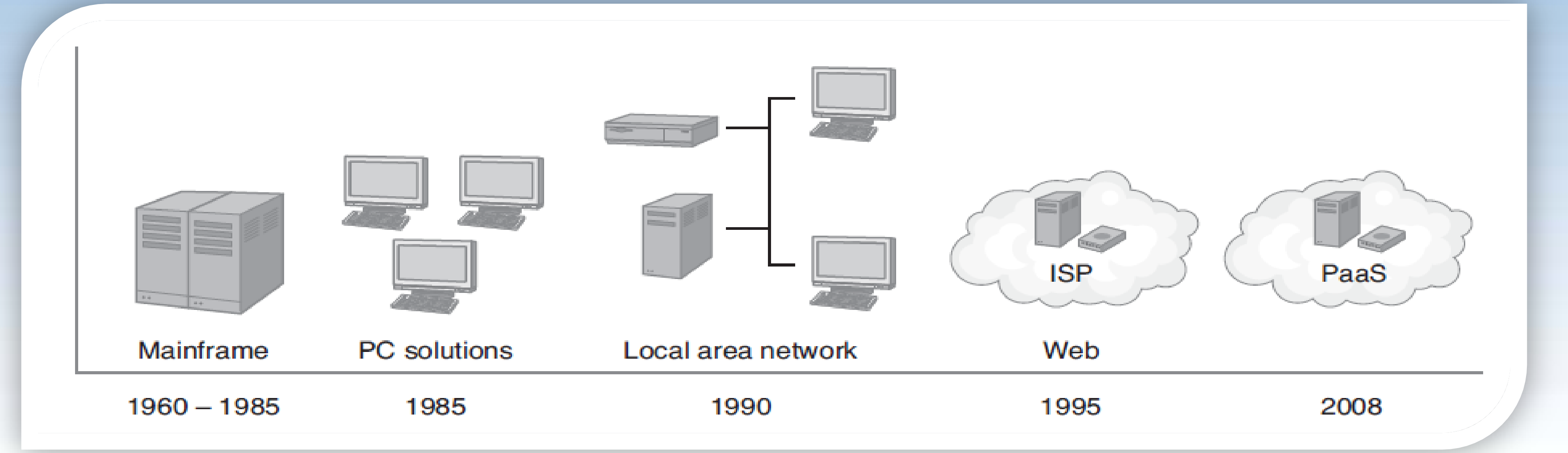
- Amazon:
  - Add more proprietary APIs?
- Google:
  - Support more languages, storage mechanisms?

Making a Choice

- Researchers will pick Amazon:
  - Fewer restrictions
  - Easier to try out new ideas
- Application developers:
  - If AppEngine meets all your needs, it will probably be easier to use.
  - If AppEngine doesn’t meet your needs, it may be hard to extend.

Evolution to the Cloud

- Mainframe Computers
- Personal Computers
- Local-Area Networks
- Internet Service Providers (ISPs)
- PaaS



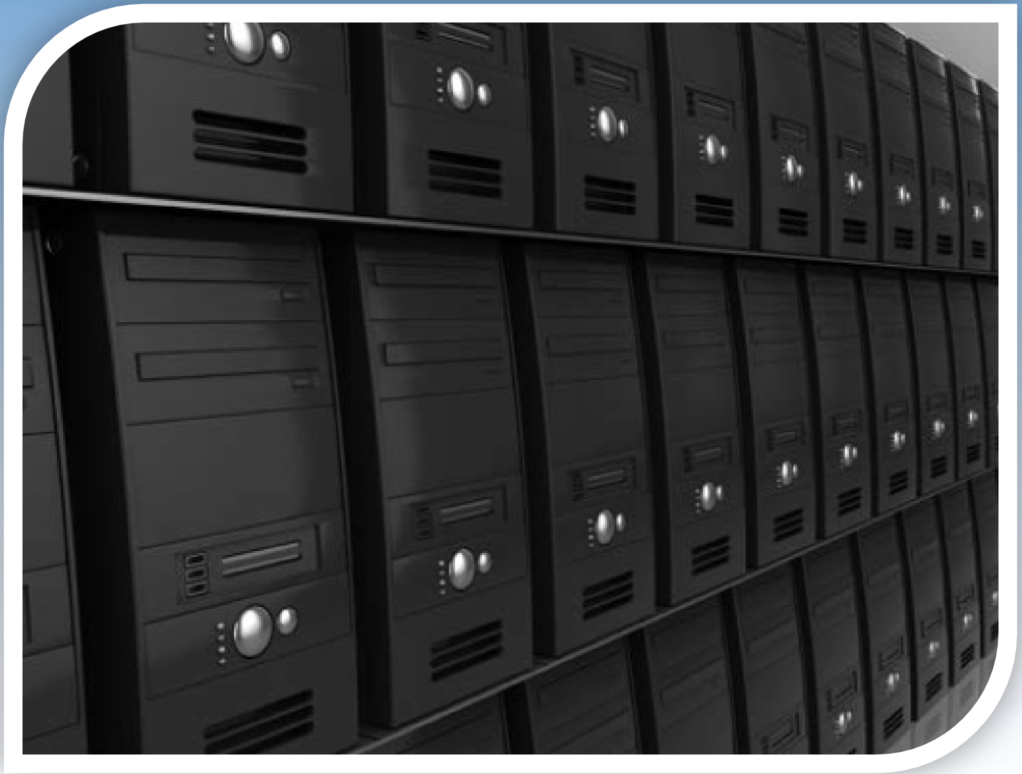
Mainframe Computing

- Large capital investment for data-center-based computers
- Large, expensive disk and tape storage systems that often provided only limited storage capacity
- User interface to the system provided through dumb terminals
- Limited computer–network interconnectivity
- System security maintained through physical security (few users had direct access to the computer hardware)

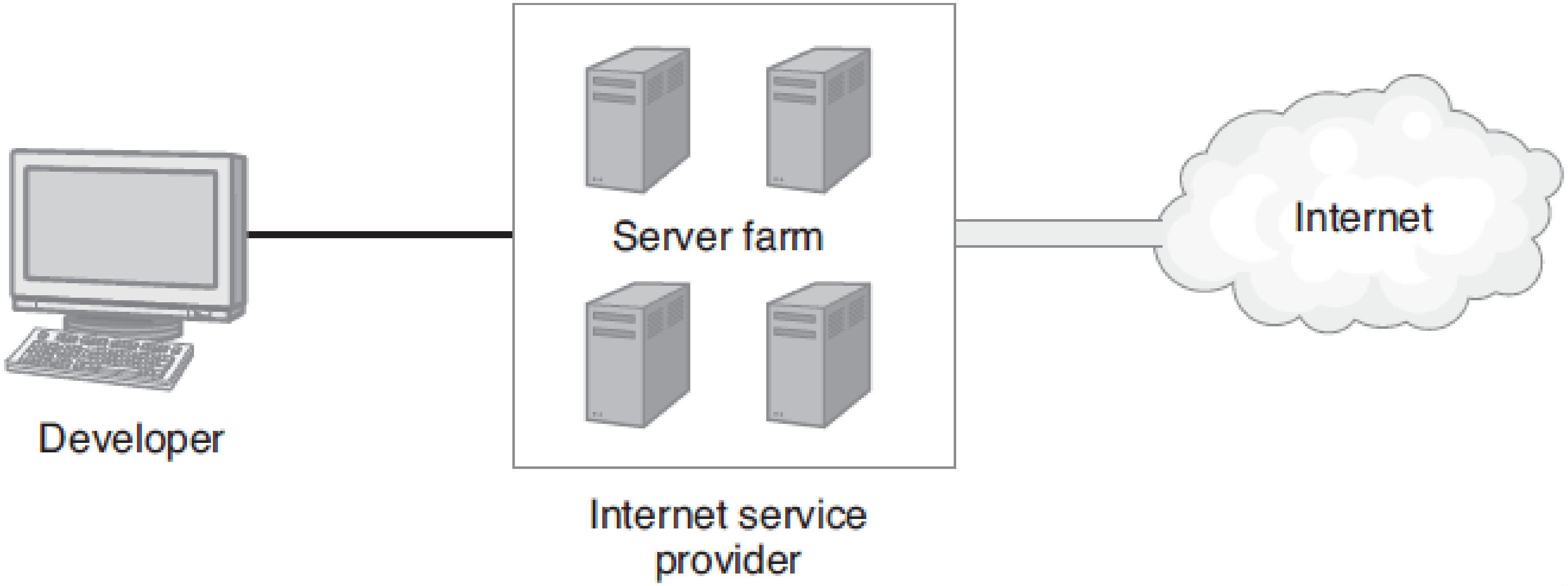
Mainframe Computer

Tower-Based Servers

- Large physical footprint
- Considerable heat generation and power consumption



Internet Service Providers (ISPs)



ISP Advantages

- **Reduced cost**: The ISP provided the high-speed, high-bandwidth Internet connection, which it shared across several companies.
- **Less server administration**: The ISP managed the servers to which developers uploaded their solutions.
- **Less hardware to purchase and maintain**: The ISP purchased and managed the hardware and managed the infrastructure software, such as the operating system.

ISP Advantages Continued

- **Greater system uptime**: Through the use of redundant hardware resources, the ISP provided high system uptime.

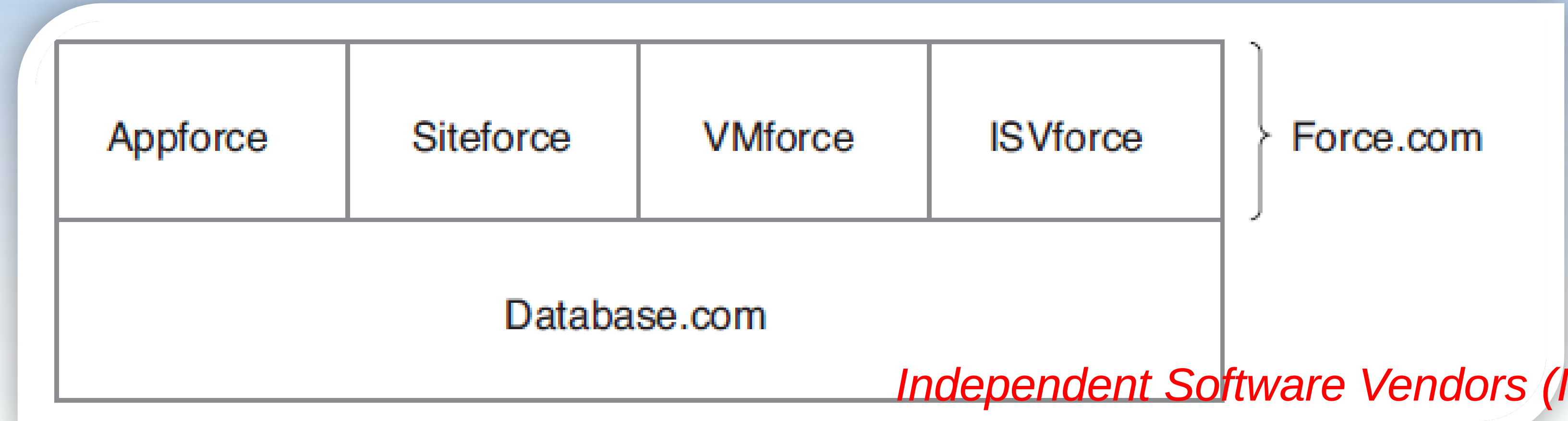
- **Potential scalability**: The ISP had the ability to move a high-demand application to a faster bandwidth connection.

Blade Computers

- Reduced server footprint
- Reduced power consumption and heat generation

Real World: Force.com PaaS

- To extend its cloud capabilities to application developers, Salesforce.com has released the Force.com PaaS.
- Originally developed to provide a home for business applications,
  - Force.com now runs applications across most sectors.



Independent Software Vendors (ISVs)

Benefits of PaaS

- In order to shift computing resources from an on-site data center to the cloud, PaaS solutions offer:
  - **Lower total cost of ownership**: Companies no longer need to purchase and maintain expensive hardware for servers, power, and data storage.
  - **Lower administration overhead**: Companies shift the burden of system software administration from in-house administration to employees of the cloud provider.

Benefits of PaaS Continued

- **More current system software**: The cloud administrator is responsible for maintaining software versions and patch installations.
- **Increased business and IT alignment**: Company IT personnel can focus on solutions as opposed to server-related issues.
- **Scalable solutions**: Cloud-based solutions can scale up or down automatically based on application resource demands.
  - Companies pay only for the resources they consume.

Disadvantages of PaaS

- Potential disadvantages of PaaS solutions include:
  - **Concerns about data security**: Some companies are hesitant to move their data storage off-site.
  - **Challenges to integrating cloud solutions with legacy software**:
    - A company may need to support on-site solutions as well as cloud-based solutions.
    - Communication between the two application types may be difficult to impossible.
  - **Risk of breach by the PaaS provider**: If the company providing the PaaS service fails to meet agreed-upon service levels, performance, security, and availability may be at risk, and moving the application may be difficult.

Real World: Windows Azure as a PaaS

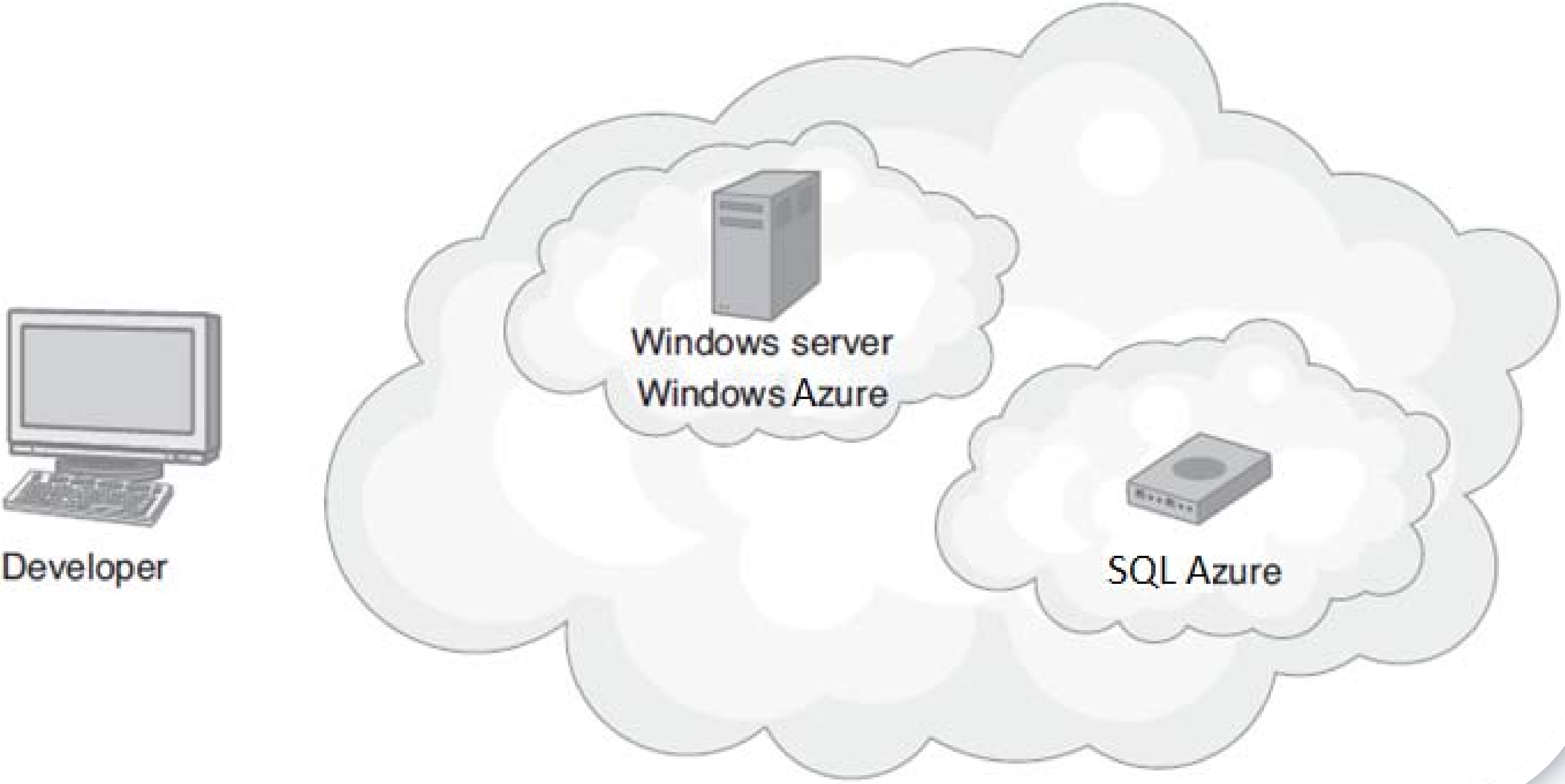
- Microsoft .NET has driven the development of many dynamic web solutions and web services.
- Windows Azure is a PaaS running within Microsoft data centers.
  - Users pay only for the scalable processor resources that they consume.
- SQL Azure provides a cloud-based database solution for applications running within Windows Azure.

Windows Azure Continued

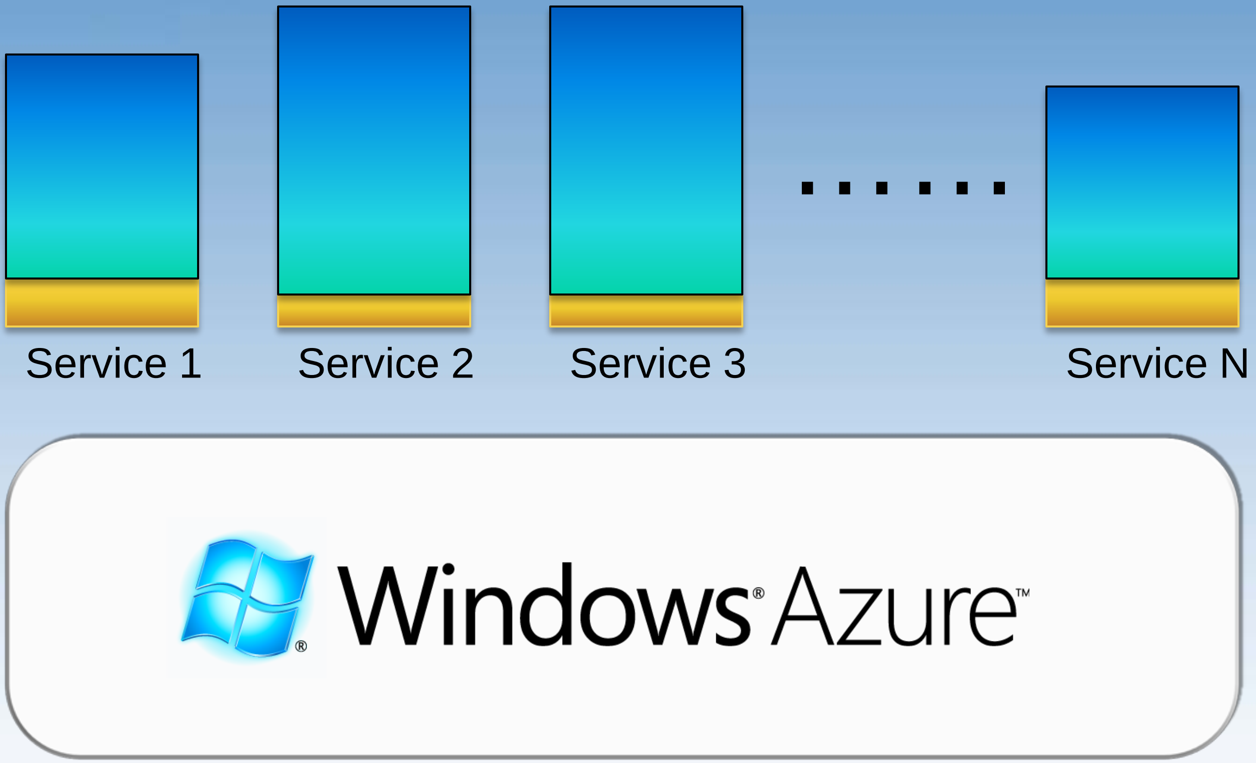
- Windows Azure goes beyond .NET and includes support for Java, PHP, and Ruby.
  - Developers can build and deploy their solutions to Azure using an IDE such as Visual Studio or Eclipse.

- Developers can interface to SQL Azure using much of the same code they would use to access a local database.

Windows Azure Continued



What's Missing?



Service 1

Service 2

Service 3

. . . . . .

Service N

Windows® Azure™

Windows Azure (Supplement)

- Platform as a Service
  - Application Platform in the Cloud
- Provides:
  - *Compute*
    - Web, Worker & VM Role
  - *Storage*
    - *Blob, Table, Queue & Azure SQL Server*
  - Application *Fabric*
    - *Service Bus, Access Control, (Future: Cache, Integration & Composite)*

Blob :basic large object

Windows Azure

- Windows Azure is an OS for the data center
  - Model: Treat the data center as a machine
  - Handles resource management, provisioning, and monitoring
  - Manages application lifecycle
  - Allows developers to concentrate on business logic
- Provides shared pool of compute, disk and network
  - Virtualized storage, compute and network
  - Illusion of boundless resources
- Provides common building blocks for distributed applications
  - Reliable queuing, simple structured storage, SQL storage
  - Application services provide access control and connectivity

Windows Azure Components

	Windows Azure PaaS
Applications	Windows Azure Service Model
Runtimes	.NET 3.5/4, ASP .NET, PHP
Operating System	Windows Server 2008/R2-Compatible OS
Virtualization	Windows Azure Hypervisor
Server	Microsoft Blades
Database	SQL Azure
Storage	Windows Azure Storage (Blob, Queue, Table)
Networking	Windows Azure-Configured Networking

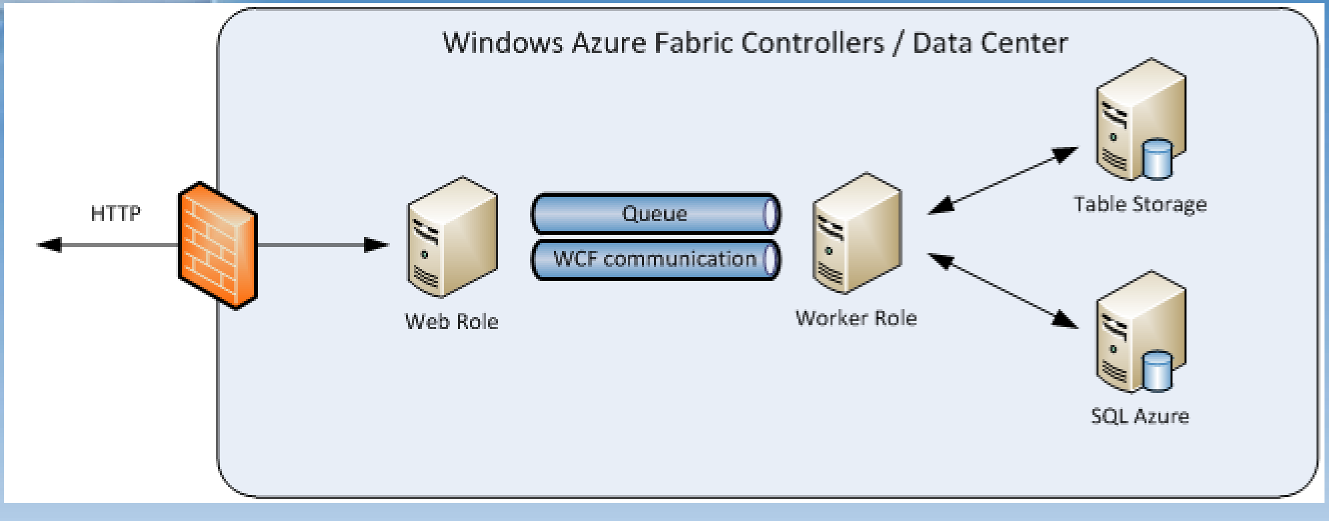
Modeling Cloud Applications

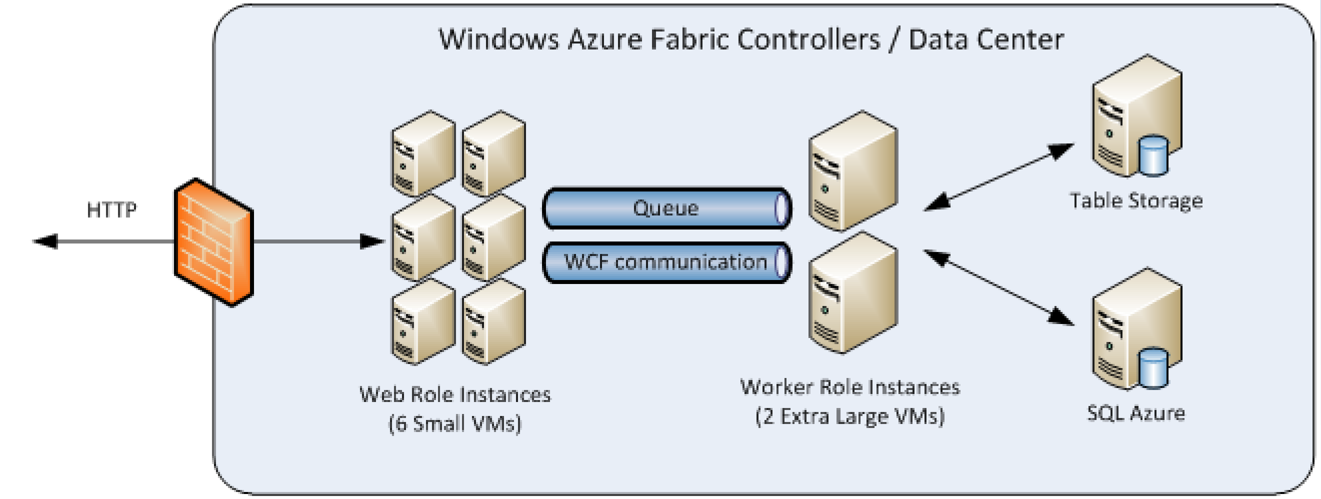
- A cloud application is typically made up of different components
  - **Front end:** e.g. load-balanced stateless web servers
  - **Middle worker tier:** e.g. order processing, encoding
  - **Backend storage:** e.g. SQL tables or files
  - **Multiple instances of each for scalability and availability**

The Windows Azure Service Model

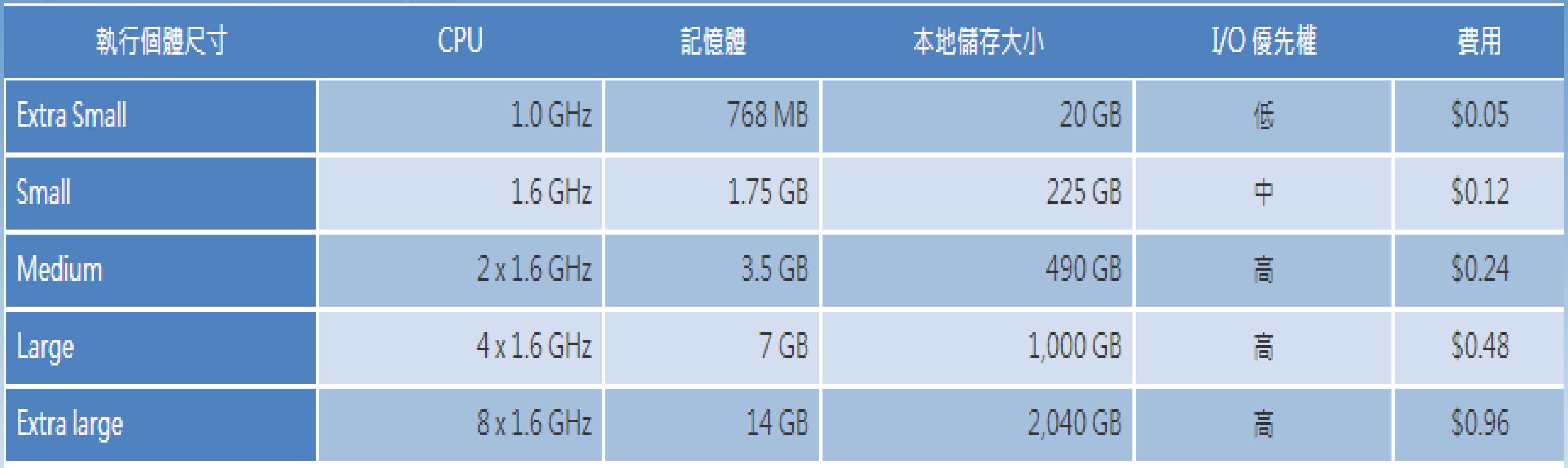
- A Windows Azure application is called a “**service**”
  - Definition information
  - Configuration information
  - At least one “role”
- Roles are like DLLs in the service “**process**”
  - Collection of code with an entry point that runs in its own virtual machine
- There are currently three role types:
  - **Web Role**: IIS7 and ASP.NET in Windows Azure-supplied OS
  - **Worker Role**: arbitrary code in Windows Azure-supplied OS
  - **VM Role**: uploaded virtual hard disk (VHD) with customer-supplied OS

Role Types


Windows Azure Fabric Controllers / Data Center
HTTP
Web Role
Queue
WCF communication
Worker Role
Table Storage
SQL Azure


Windows Azure Fabric Controllers / Data Center
HTTP
Web Role Instances
(6 Small VMs)
Queue
WCF communication
Worker Role Instances
(2 Extra Large VMs)
Table Storage
SQL Azure

VM Sizes



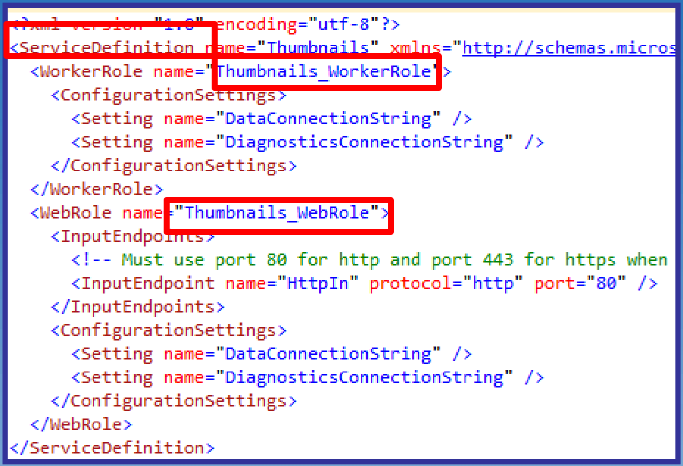
執行個體尺寸	CPU	記憶體	本地儲存大小	I/O 優先權	費用
Extra Small	1.0 GHz	768 MB	20 GB	低	$0.05
Small	1.6 GHz	1.75 GB	225 GB	中	$0.12
Medium	2 x 1.6 GHz	3.5 GB	490 GB	高	$0.24
Large	4 x 1.6 GHz	7 GB	1,000 GB	高	$0.48
Extra large	8 x 1.6 GHz	14 GB	2,040 GB	高	$0.96

Role Contents

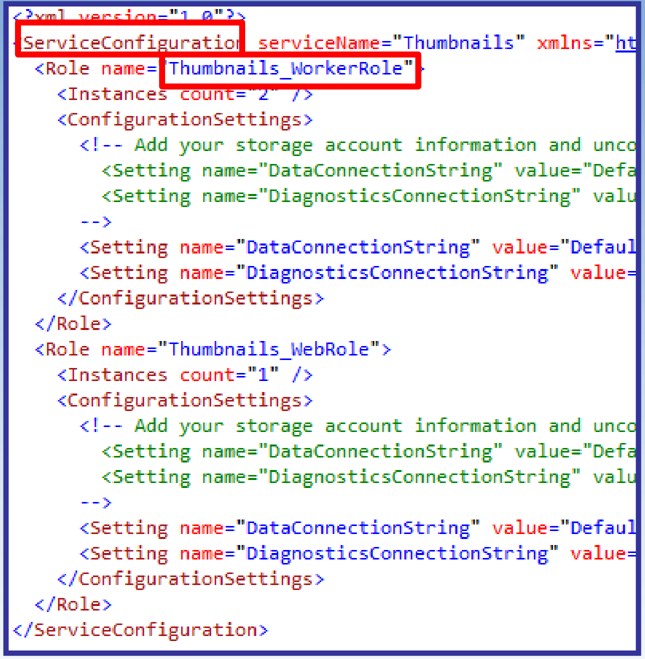
- Definition:
  - Role name
  - Role type
  - VM size (e.g. small, medium, etc.)
  - Network endpoints
- Code:
  - Web/Worker Role: Hosted DLL and other executables
  - VM Role: virtual hard disk (VHD)
- Configuration:
  - Number of instances
  - Number of **update and fault domains**

Service Model Files

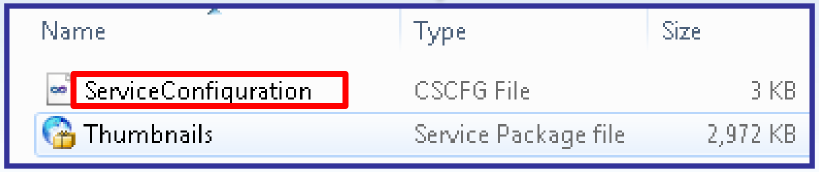
1. Service definition is in ServiceDefinition.csdef
2. Service configuration is in ServiceConfiguration.cscfg
3. CSPack program Zips service binaries and definition into service package file (service.cscfg)



```
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="Thumbnails" xmlns="http://schemas.micros
  <WorkerRole name="Thumbnails_WorkerRole">
    <ConfigurationSettings>
      <Setting name="DataConnectionString" />
      <Setting name="DiagnosticsConnectionString" />
    </ConfigurationSettings>
  </WorkerRole>
  <WebRole name="Thumbnails_WebRole">
    <InputEndpoints>
      <!-- Must use port 80 for http and port 443 for https when
      <InputEndpoint name="HttpIn" protocol="http" port="80" />
    </InputEndpoints>
    <ConfigurationSettings>
      <Setting name="DataConnectionString" />
      <Setting name="DiagnosticsConnectionString" />
    </ConfigurationSettings>
  </WebRole>
</ServiceDefinition>
```



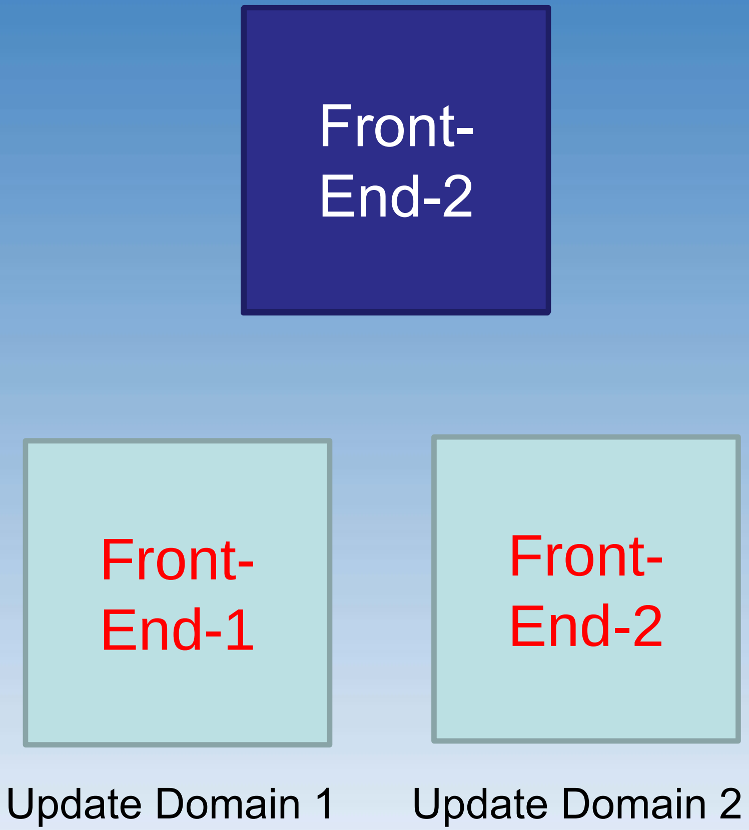
```
<?xml version="1.0"?>
<ServiceConfiguration serviceName="Thumbnails" xmlns="ht
  <Role name="Thumbnails_WorkerRole">
    <Instances count="2" />
    <ConfigurationSettings>
      <!-- Add your storage account information and unco
        <Setting name="DataConnectionString" value="Defa
        <Setting name="DiagnosticsConnectionString" valu
      -->
      <Setting name="DataConnectionString" value="Defaul
      <Setting name="DiagnosticsConnectionString" value=
    </ConfigurationSettings>
  </Role>
  <Role name="Thumbnails_WebRole">
    <Instances count="1" />
    <ConfigurationSettings>
      <!-- Add your storage account information and unco
        <Setting name="DataConnectionString" value="Defa
        <Setting name="DiagnosticsConnectionString" valu
      -->
      <Setting name="DataConnectionString" value="Defaul
      <Setting name="DiagnosticsConnectionString" value=
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>
```



Name	Type	Size
ServiceConfiguration	CSCFG File	3 KB
Thumbnails	Service Package file	2,972 KB

Availability: Update Domains

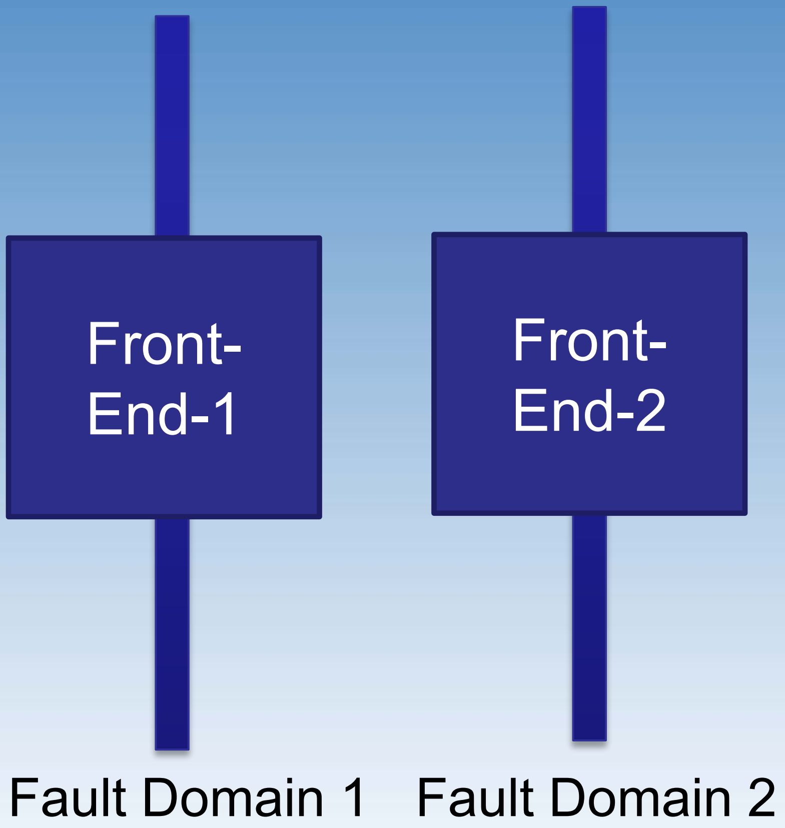
- Purpose: Ensure service stays up while updating service and Windows Azure OS
- System considers update domains when upgrading a service
  - percent of service = Update domains/Instance count
    - they will be offline
  - Default and max is 5, but you can override with upgradeDomainCount service definition element
- The Windows Azure SLA is based on at least two update domains and two role instances in each role



Service Level Agreement (SLA)

Availability: Fault Domains

- Purpose: Avoid single points of failures
  - Similar concept to update domains
  - But you don’t control the updates
- Unit of failure based on data center topology
  - E.g. top-of-rack switch on a rack of machines
- Windows Azure considers fault domains when allocating service roles
  - E.g. don’t put all roles in same rack



Front-End-1

Front-End-2

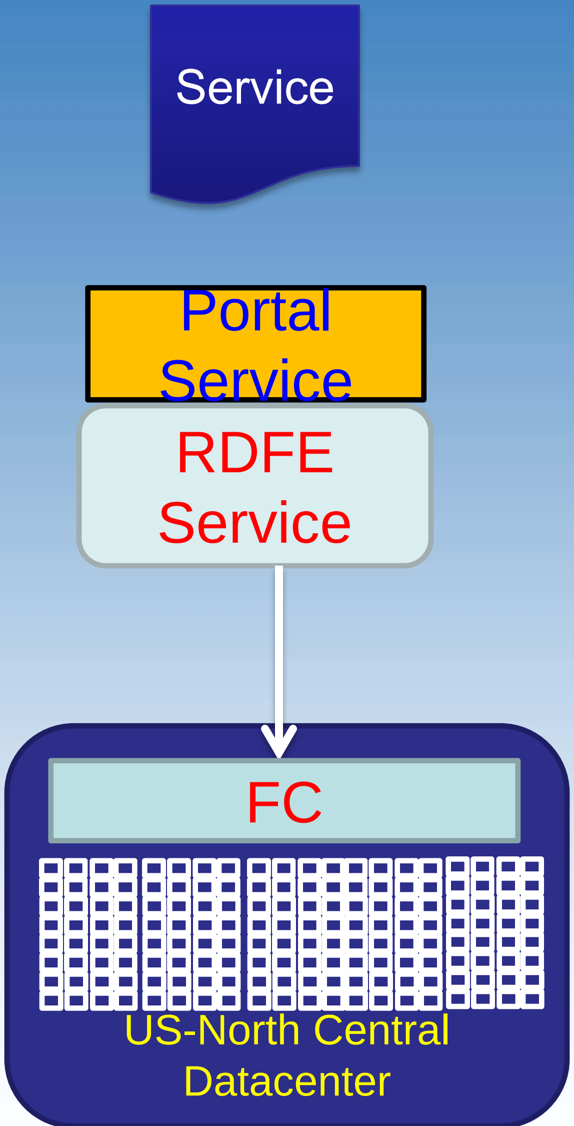
Fault Domain 1

Fault Domain 2

Deploying a Service

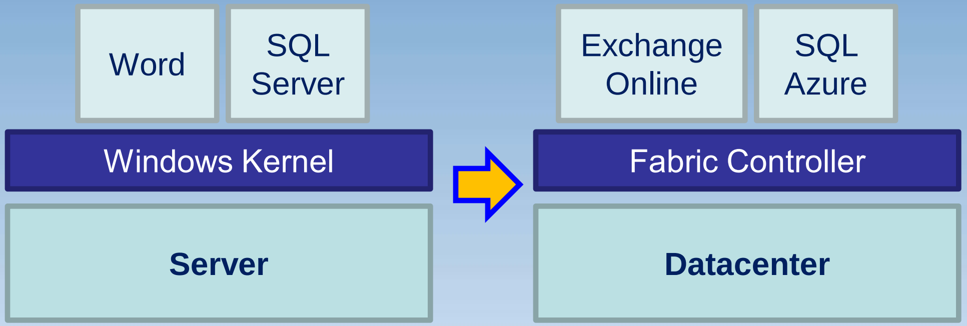
The 10,000 foot view

- Service package uploaded to portal
  - Windows Azure Portal Service passes service package to “**Red Dog Front End**” (RDFE) Azure service
  - RDFE converts service package to native “RD” version
- RDFE sends service to Fabric Controller (FC) based on target region
- FC stores image in repository and deploys and activates service



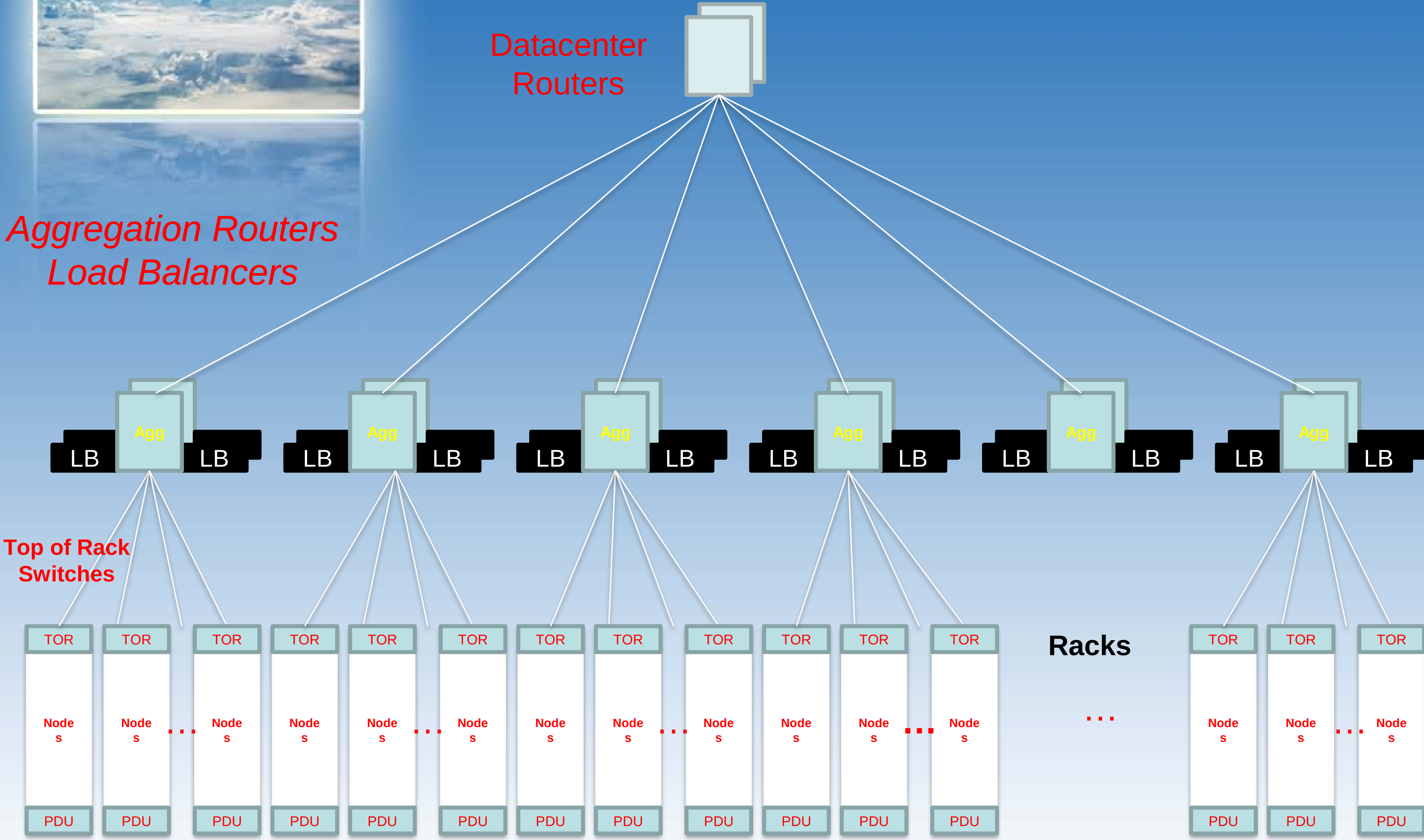
The Fabric Controller (FC)

- The “**kernel**” of the cloud operating system
  - Manages datacenter hardware
  - Manages Windows Azure services
- Four main responsibilities:
  1. Datacenter resource allocation
  2. Datacenter resource provisioning
  3. Service lifecycle management
  4. Service health management



- Inputs:
  - Description of the hardware and network resources it will control
  - Service model and binaries for cloud applications

Datacenter Architecture



Datacenter Routers

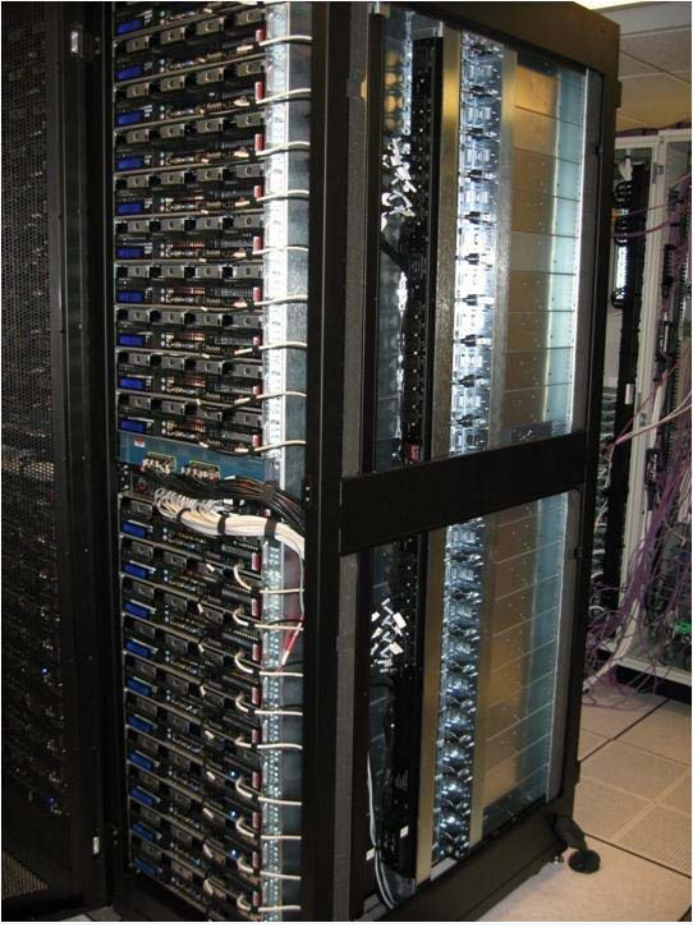
Aggregation Routers
Load Balancers

Top of Rack Switches

Racks

PDU(Power Distribution Units)

Windows Azure Datacenters

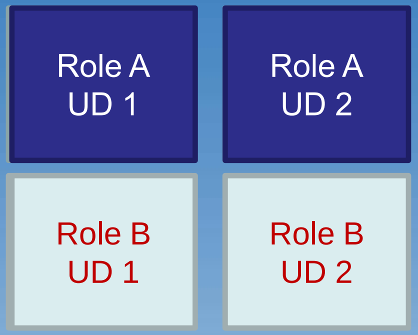


DIP-Directed IP
Virtual IP Swap

Update Types

- There are **two update types**:
  - **In-place update:**
    - Supports changes to configuration or binaries, not service definition
    - Role instances upgraded one update domain at a time
    - Two modes: automatic and manual
  - **VIP swap update:**
    - Service definition can change, but external endpoints must remain the same
    - New version of service deployed, external VIP/DIP mapping swapped with old

In-Place Update



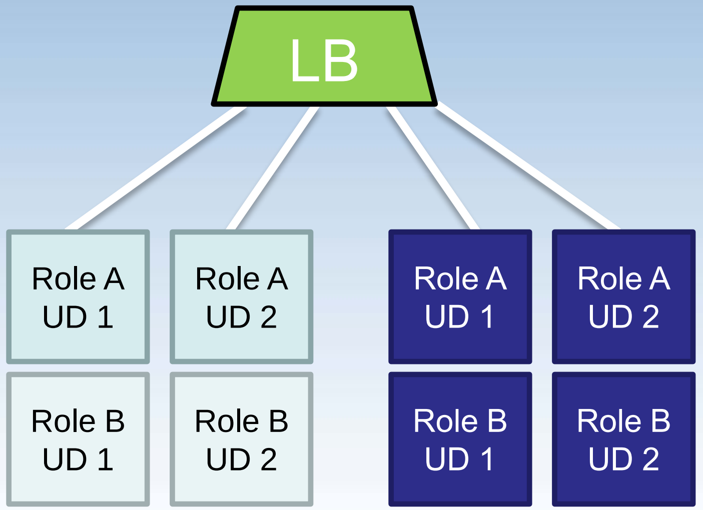
Role A UD 1

Role A UD 2

Role B UD 1

Role B UD 2

VIP Swap Update



LB

Role A UD 1

Role A UD 2

Role A UD 1

Role A UD 2

Role B UD 1

Role B UD 2

Role B UD 1

Role B UD 2

Node and Role Health Maintenance

- FC maintains service availability by monitoring the software and hardware health
  - Based primarily on heartbeats
  - Automatically “heals” affected roles

Problem	How Detected	Fabric Response
Role instance crashes	FC guest agent monitors role termination	FC restarts role
Guest VM or agent crashes	FC host agent notices missing guest agent heartbeats	FC restarts VM and hosted role
Host OS or agent crashes	FC notices missing host agent heartbeat	Tries to recover node FC reallocates roles to other nodes
Detected node hardware issue	Host agent informs FC	FC migrates roles to other nodes Marks node “out for repair”

Summary

- Platform as a Service is all about reducing management and operations overhead
- The Windows Azure Fabric Controller is the foundation for Windows Azure’s PaaS
  - Provisions machines
  - Deploys services
  - Configures hardware for services
  - Monitors service and hardware health
  - Performs service healing

Windows Azure Platform Purchasing Models

Consumption

"Pay as you go and grow"

Available Jan 2010

- Low barrier to entry & flexibility
- Optimized for cloud elasticity

Subscription

"Value for a commitment "

Select offers available Jan 2010

- Discounts for commitment
- Plans for payment predictability

Additional Licensing

"Coordinated purchasing"

Planned for post PDC

- Centralized purchasing experience
- Introduction to volume discounts

Promotional Offers

Development Pricing

Partner Discount

Integration with Programs

Windows Azure Platform Consumption Prices

Pay as you go and grow for only what you use when you use it

Windows Azure

Elastic, scalable, secure, & highly available automated service platform

Microsoft SQL Azure

Highly available, scalable, and self managed distributed database service

Compute

Per service hour

$0.12/hour
+ Variable Instance Sizes

Storage

Per GB stored & transactions
$0.15 GB/month
$0.01/10K transactions

Web Edition

Per database/month

$9.99/month
(up to 1 GB DB/month)

Business Edition

Per database/month

$99.99/month
(up to 10 GB DB/month)

Windows Azure platform AppFabric Service Bus & Access Control

Scalable, automated, highly available services for secure connectivity

Access Control

Per Message Operation

$0.015/10k Message Operations

Service Bus

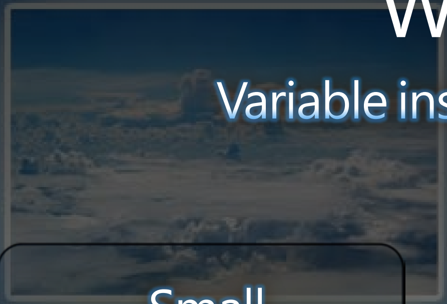
Per Message Operation

$0.015/10k Message Operations

Prices shown in USD only

International prices are available

Windows Azure Instance Sizes



Variable instance sizes to handle complex workloads of any size

Small	Medium	Large	X Large
$0.12	$0.24	$0.48	$0.96
Per service hour	Per service hour	Per service hour	Per service hour

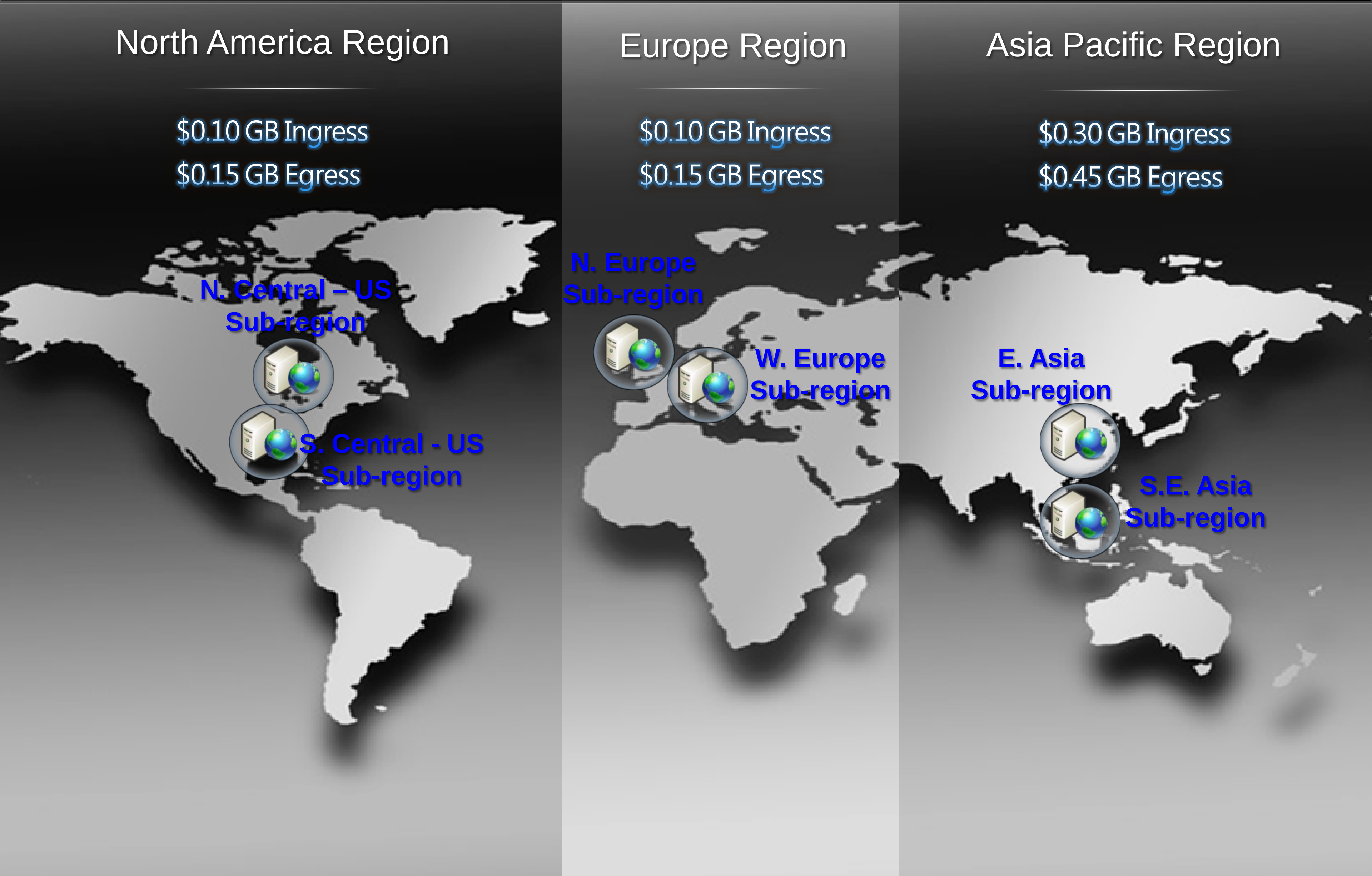
Unit of Compute Defined

Equivalent compute capacity of a 1.6Ghz processor (on 64bit platform)

Small	Medium	Large	X-Large
1 x 1.6Ghz	2 x 1.6Ghz	4 x 1.6Ghz	8 x 1.6Ghz
(moderate IO)	(high IO)	(high IO)	(high IO)
1.75 GB memory	3.5 GB memory	7.0 GB memory	14 GB memory
250 GB storage (instance storage)	500 GB storage (instance storage)	1000 GB storage (instance storage)	2000 GB (instance storage)

Windows Azure Platform Data Transfer

Priced per GB transferred/month (prices shown in USD)



North America Region

$0.10 GB Ingress

$0.15 GB Egress

Europe Region

$0.10 GB Ingress

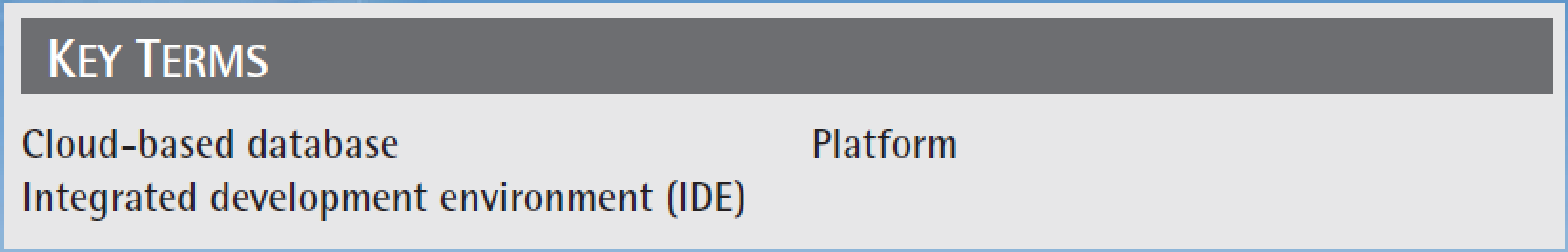
$0.15 GB Egress

Asia Pacific Region

$0.30 GB Ingress

$0.45 GB Egress

Key Terms



KEY TERMS

Cloud-based database

Integrated development environment (IDE)

Platform

Chapter Review

1. Define and describe PaaS.

2. List the benefits of PaaS solutions.

3. Describe potential disadvantages of PaaS.

4. Describe how a cloud-based database management system differs from an on-site database.

5. List the computing resources normally provided with a PaaS.

Chapter Review Continued

6. Assume your company must deploy a .NET solution to the cloud. Discuss the options available to developers. Research the web and estimate the costs associated with deploying a PaaS solution.

7. Assume your company must deploy a PHP or Java solution to the cloud. Discuss the options available to developers. Research the web and estimate the costs associated with deploying a PaaS solution.