Cloud Computing Environments and Technologies

Salvatore Filippone, PhD

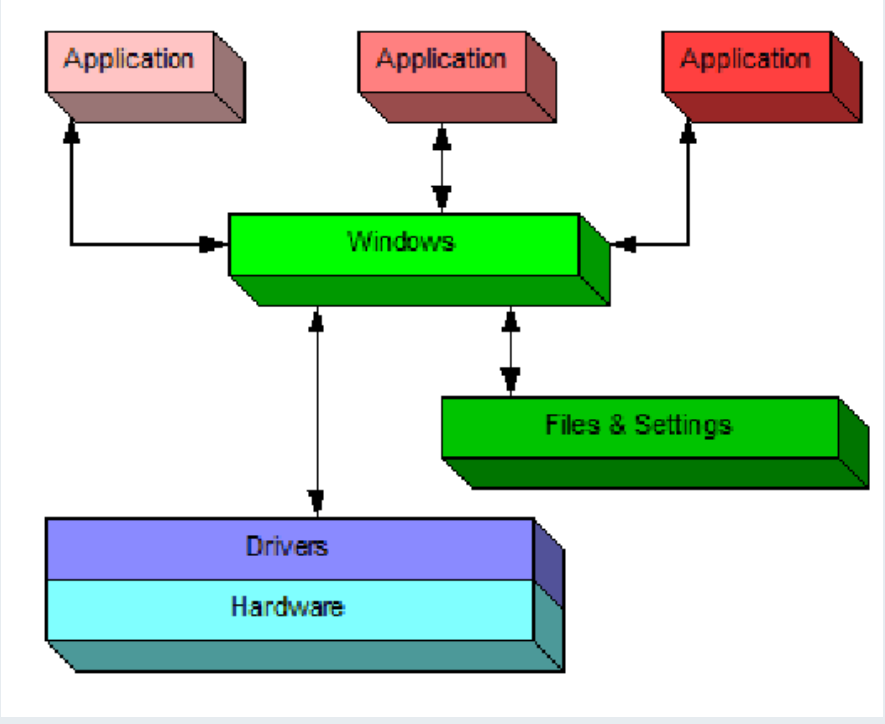
School of Aerospace, Transport and Manufacturing
salvatore.filippone@cranfield.ac.uk

Google App Engine — Introduction

- Provides a PaaS to developers of web based applications
- Allows your web based applications to run on Google's infrastructure
- Only handles HTTP/HTTPS requests
- Applications are based around this web-based model
- No low level access to resources
- Resources packaged and offered to developers via API
- Services to manage the availability of in-house applications to users, with an associated one-off monthly charge (App Engine for Business)

Google App Engine — Sandbox

Windows on its own: conceptual view on how Windows and Windows applications operate at a very high level:

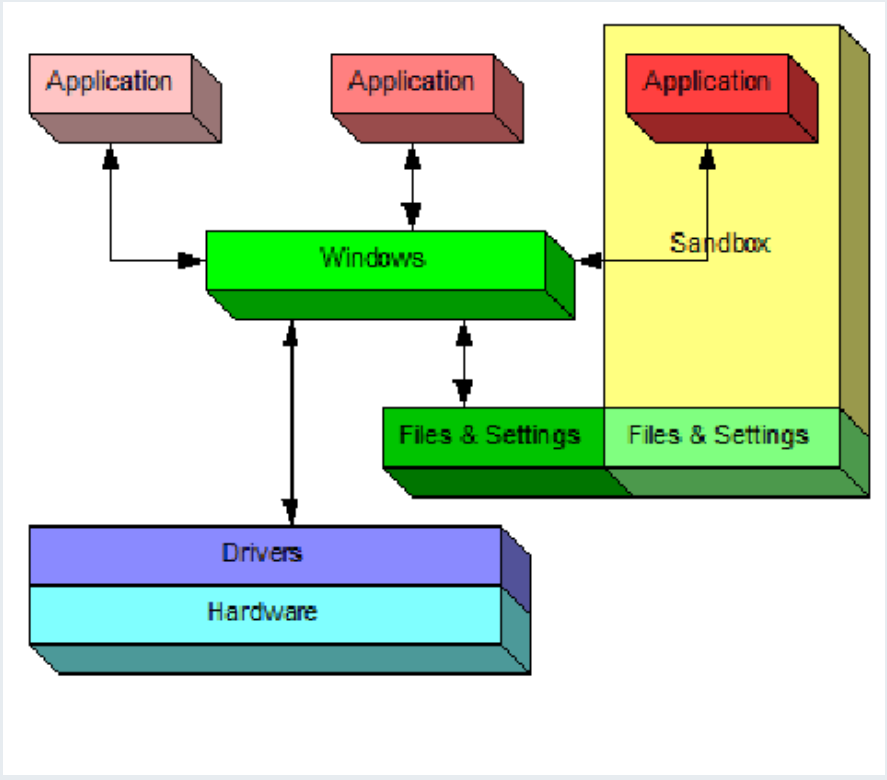


- Applications running in Windows interact with the machine and with you through Windows.
- Windows manages access to the files and on-disk resources
- Also manages access to the hardware through the device drivers that are installed for your machine's specific hardware configuration.

Source: `ask-leo.com`

Google App Engine — Sandbox

A sandbox under Windows: a sandbox is a container placed around an application running within Windows

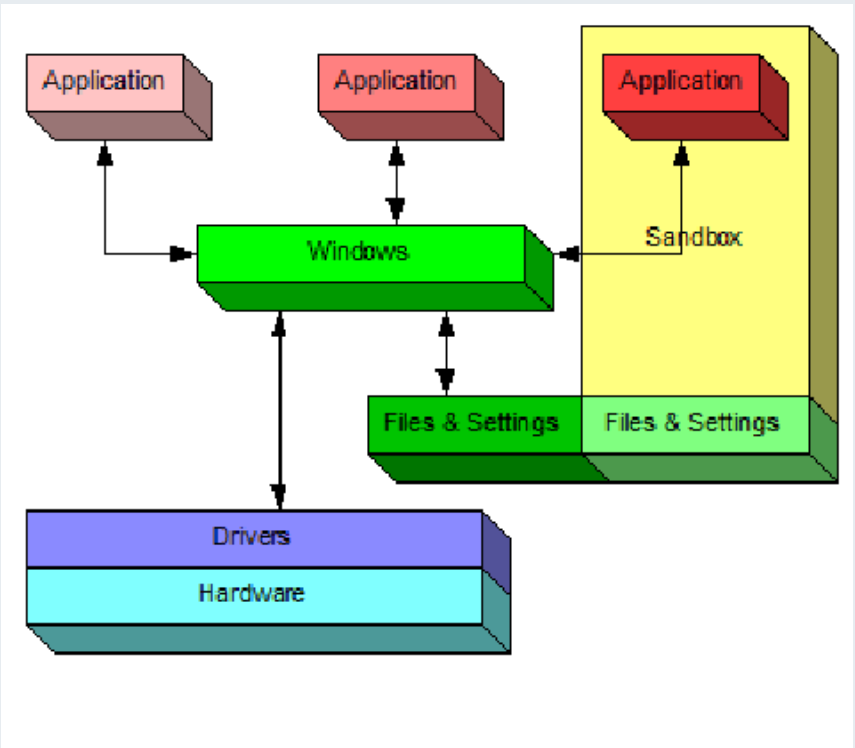


- One of the 3 applications in this ex. is drawn as being within a sandbox.
- "Files & Settings" used by that application are also placed in that sandbox.
- When you run an application within a sandbox, it continues to have access to everything that it would if it was not sandboxed.

Source: `ask-leo.com`

Google App Engine — Sandbox

A sandbox under Windows: a sandbox is a container placed around an application running within Windows



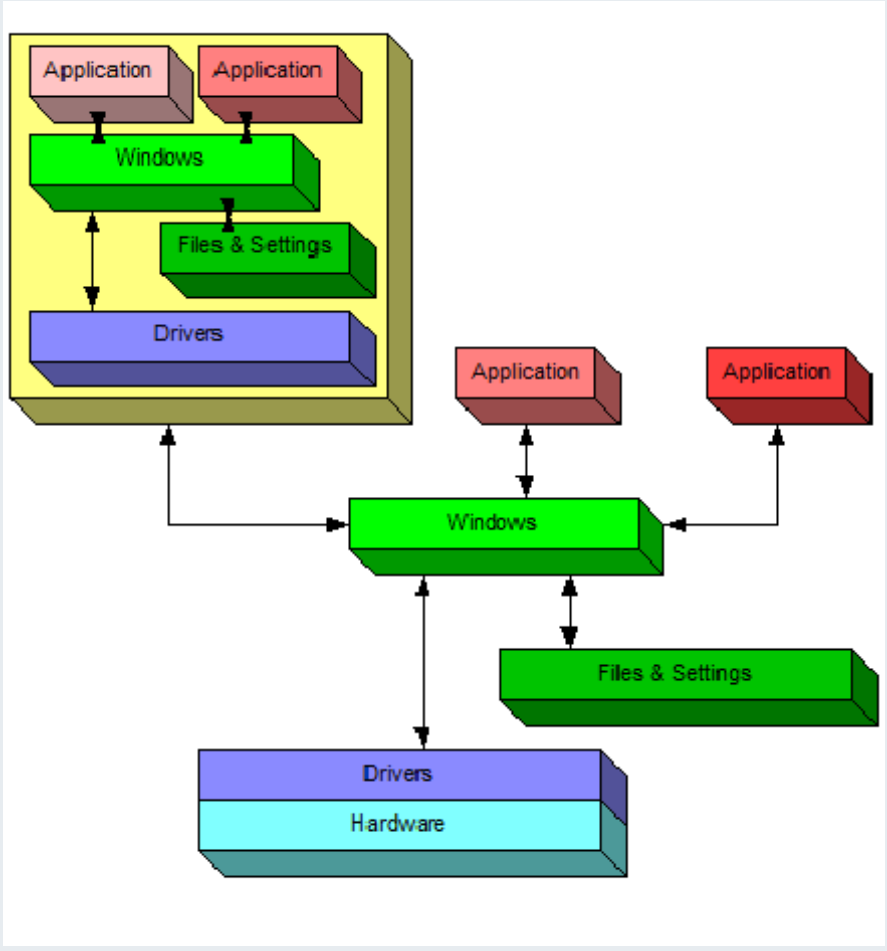
The primary difference is that anything created or changed by the sandboxed application is:

- Not visible outside of the sandbox; other Windows applications don't see it.
- Not saved when the sandboxed application exits

The best example: any malware downloaded and "installed" by the sandboxed application is discarded when the application exits.

Google App Engine — Sandbox

A virtual machine under Windows: an application that creates an environment simulating a completely separate computer.



- Left application is a VM running a completely separate copy of Windows
- "machine within a machine"
- Windows on the actual PC — Host OS
- VMs on it — Guest OSs
- Within a VM, applications continue to access the world around them through that VM's copy of Windows

Source: `ask-leo.com`

Google App Engine — Sandbox

A virtual machine under Windows:

- Includes that VM's own virtual hard disk on which files and settings are stored.
- Also includes its own set of virtual device drivers that behave as if they're interfacing to actual hardware
- In reality, they're mimicking the presence of actual hardware and talking to the host copy of Windows to gain access to the real hardware.
- Everything that happens in the VM stays within the VM.
- It behaves exactly as if it were a completely separate physical machine.
- So any downloads, changes, updates, installations . . . anything . . . that is created or saved within the virtual machine is only accessible through the VM in some way
- If we delete the VM, it is like getting rid of a PC- Everything on the virtual hard disk is erased

Source: `ask-leo.com`

Google App Engine — Sandbox

Sandbox vs VM

- Sandboxing is lightweight, easy to set up and use. But complexities, like changes that we want to preserve while in sandbox is complicated
  $\Rightarrow$ Ex. If your browser is sandboxed, getting a downloaded file you want to use outside the sandbox may take a few extra steps.
- VMs are not lightweight. Need disk space to allocate to the virtual hard drive, decisions like how much of RAM to allocate to the VM while it's running, needs installing an OS from scratch.
- A VM and its host isolated from each other, to copy files to and from the VM needs setting up network access on that machine —- just as if it was separate physical machine.
- When to go for VM— If you want a completely isolated "virtual" 2nd, 3rd, 4th, etc machine, each one running a different OS.

Source: `ask-leo.com`

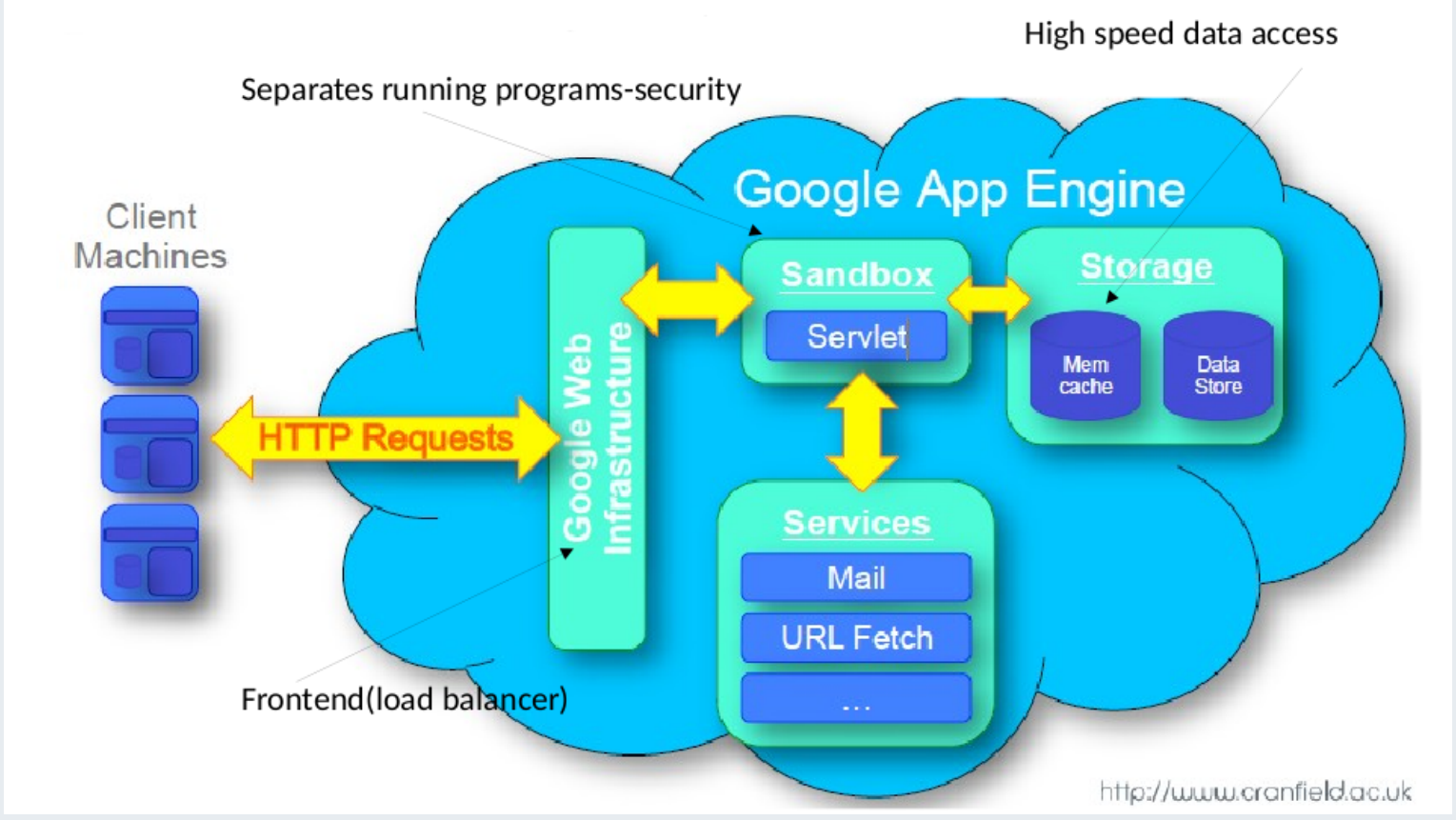
Google App Engine

Advantages of sandbox environment

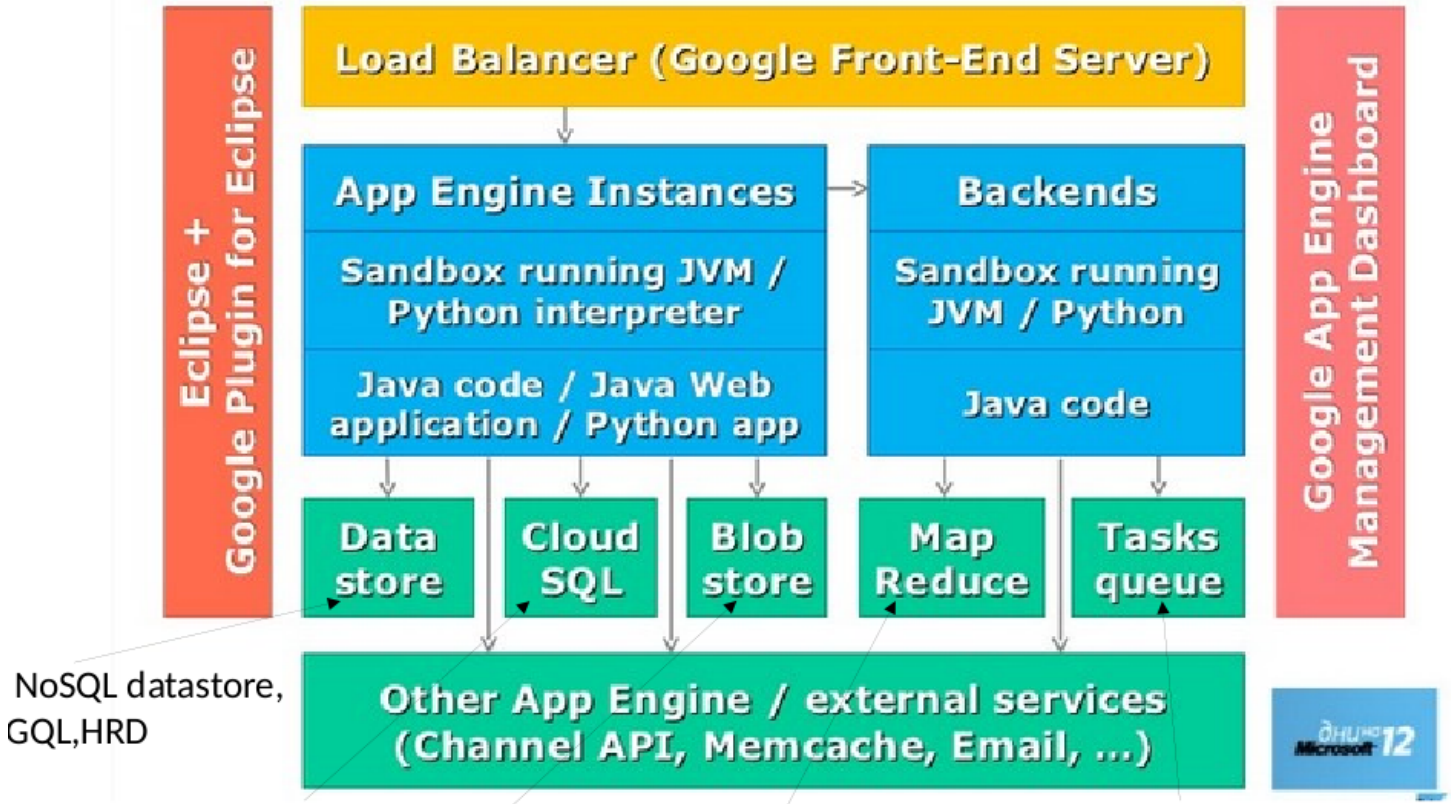
- Limited access to underlying physical resources, which instead are offered through a managed API
- Sandboxing allows App Engine to run multiple applications on the same server without the behavior of one application affecting another
- Sandbox ensures that the application is isolated from other applications, increasing security
- Applications can only access other computers via HTTP/HTTPS requests
- There is no local block storage. Persistence is offered through one of App Engine's storage services
- Applications respond to a web request, and must respond with data within 30 seconds
- Web based requests for the application are handled by Google's infrastructure, resources can be dynamically scaled as the requests change

Typical Application Architecture

Google Web Infrastructure scales out as required to handle web requests



Google App Engine — Typical Application Architecture



Google App Engine

Services

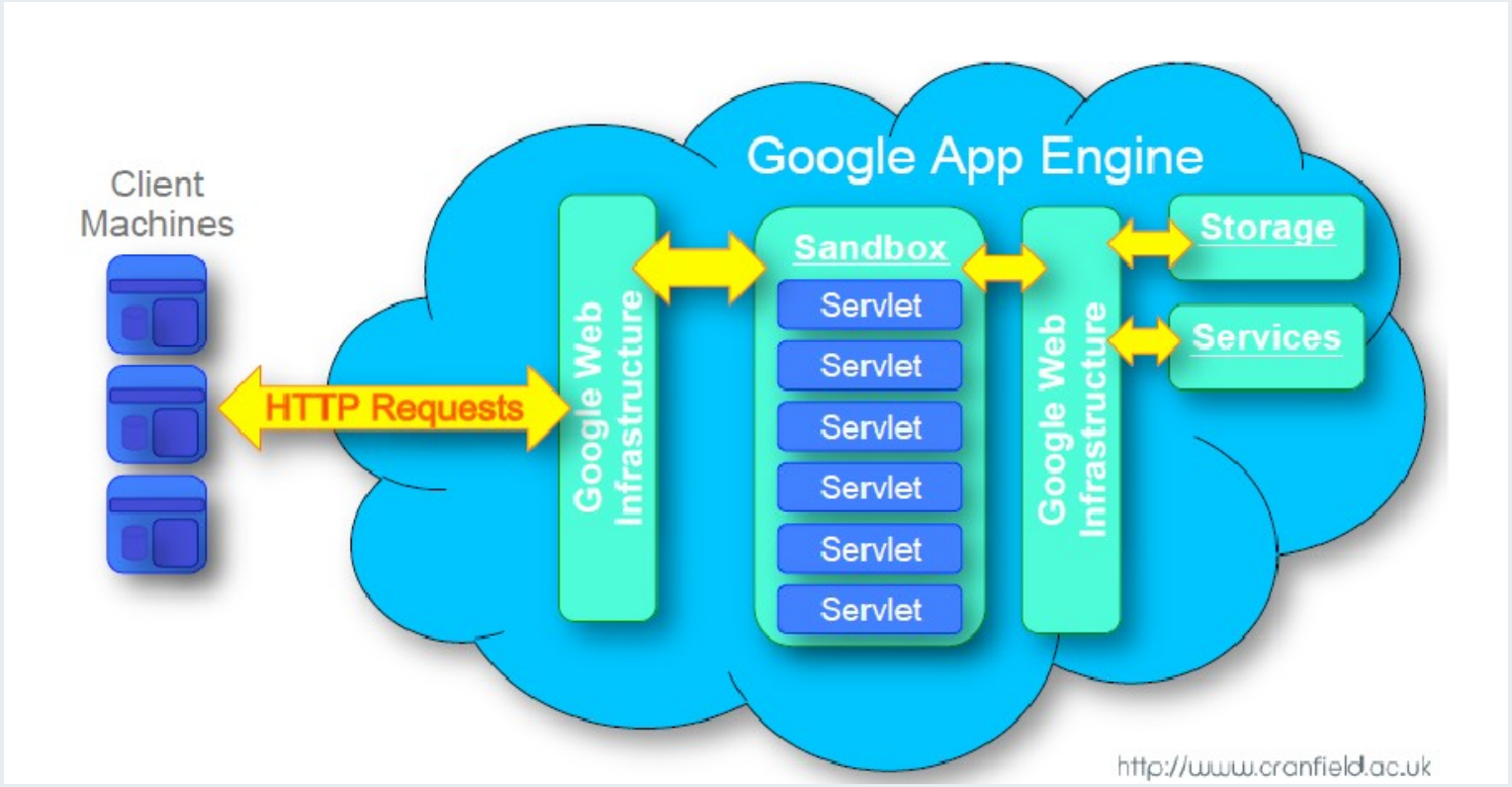
Instances	Computing units hosting applications
Backends	Like instances, but higher computing resources used for background processing
Datastores	NoSQL schemaless object db, GQL, new High Replication Datastore, master slave datastore $\Rightarrow$ M-S Datastore - quick, strongly consistent reads; Master/Slave replication topology, fast writes, but applications can see data immediately after it is written
Cloud SQL	Managed MySQL in GAE
Blobstore/ Cloud Storage	for files/blobs
MapReduce API	$\Rightarrow$ Highly-scalable parallel computing API for heavy computing tasks (based on Hadoop)
Channel API	$\Rightarrow$ Push notifications for JavaScript applications

Google App Engine

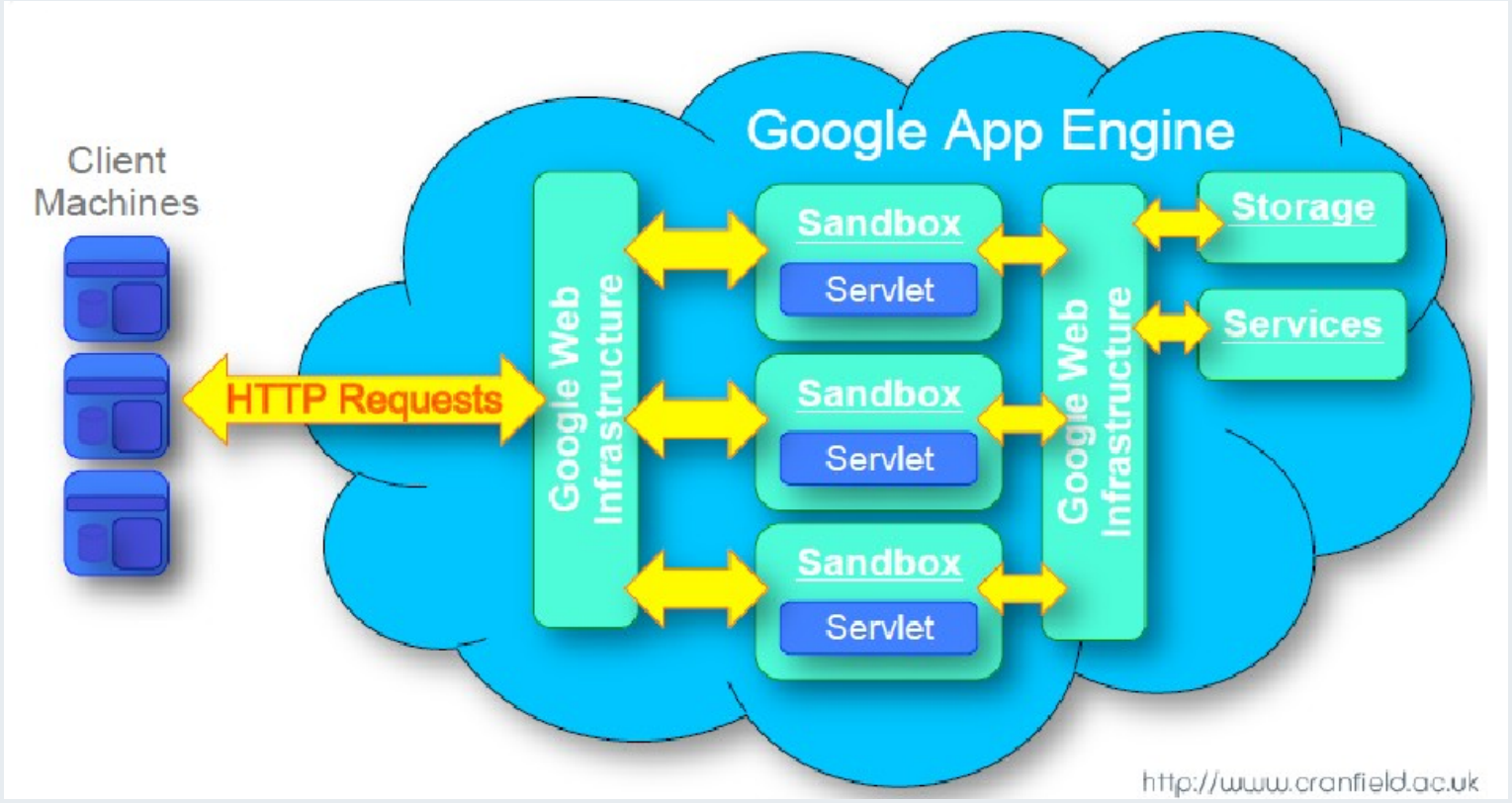
Services

Task Queues	Services for execution of background work
URL Fetch	access resources from other internet sites or services
Mail	send or receive messages using Googles e-mail infrastructure
Memcache	Distributed in-memory data cache; key-value cache available to all instances of the application
Image Manipulation	Basic image processing capabilities
Scheduled Tasks and Task Queues	allows performance of tasks other than responses to web requests

Typical Application Architecture: Individual Apps Scale Out



Typical Application Architecture: Infrastructure Scales As Required



Google App Engine — Development Tools

Limited development environments for applications:

- App Engine supports applications that will run in either Java or Python run time environments
- Java programming supports the Java Servlet standard — a Java application that runs on the server side
- Extends the functionality of a web server by allowing the implementation of application specific functionality to web pages
- Common Java storage APIs are available — JDO and JPA
- Similar functionality available for Python based applications
- Provides IDE extensions to Eclipse to aid development

Google App Engine — Typical Costs

Charges are based on usage and are relatively straight forward (after free quotas)

- Outgoing Bandwidth — $0.12 / GB
- Incoming Bandwidth — Free
- CPU Time — $0.05 / CPU hour
- Datastore Storage — $0.18 / GB per month
- Cloud Storage - $0.026/GB per month
- Recipients Emailed — Free

Source: Google (`http://code.google.com/appengine/docs/billing.html`) November 2017

Google App Engine — Typical Costs

App Engine presents mechanisms for limiting costs and resource usage

Google charges for transaction bandwidth and data storage, so how can a vendor ensure that the costs do not exceed their ability to pay?

Billable quotas a maximum daily spend defined by the application vendor

Per-minute quotas — ensures that app doesn't consume daily quota in a short period of time

Fixed Quota an absolute maximum of resource usage defined by Google to limit the impact of high usage applications

Once a quota has been depleted, the application will not run until the quota has been replenished

Google App Engine

Pros

- Generous free quota
- Billed for actual CPU usage, not "live" hours
- Scaling is (relatively) easy
- "Free" built-in Google authentication
- Great integration with many of the Google offerings — calendars, mail

Cons

- (Pretty much) locked in to Google App Engine
- No support for C#/.NET
- Non-relational datastore - more restrictive on how data can be accessed
- (Relatively) high rate of datastore errors, need to code to handle them
- No local storage, so unable to use libraries that rely on writing to disk

Microsoft Azure vs Amazon AWS

Azure

- **IaaS** Azure Virtual Machines
- **PaaS**
  - **Web Roles** for hosting front end Web Applications
  - **Web Sites** - for developing and running web apps

Amazon

- **IaaS** Amazon EC2 (Elastic Compute 2)
- **PaaS** Elastic Beanstalk (Beta) - to create web applications

Microsoft Azure — Introduction

Microsoft Azure: PaaS platform offering a range of services

Compute Services Azure Virtual Machine

Storage Services SQL Azure

Content Distribution Network

AppFabric connects local applications with applications running on Azure

- Windows Azure Platform Appliance – Local Azure installation
- Supports a range of development platforms, environments and tools
- More choice over the available compute resources than App Engine
- Also PaaS services — Azure Web Roles, Azure Web Sites, Azure SQL Database

Microsoft Azure — Introduction

Microsoft Windows Azure;

- Fast-growing public cloud
- Provides rich PaaS platform
- Mainly for .NET developers
- Supports all major NET technologies:
  - ASP.NET MVC, WCF, ADO.NET EF, WWF
  - Also Java, PHP and Node.js APIs
- No free version, only 3 months trial

Microsoft Azure — Introduction

PaaS platform offering a range of services:

Compute Services Azure

Storage Services SQL Azure

Content Distribution Network

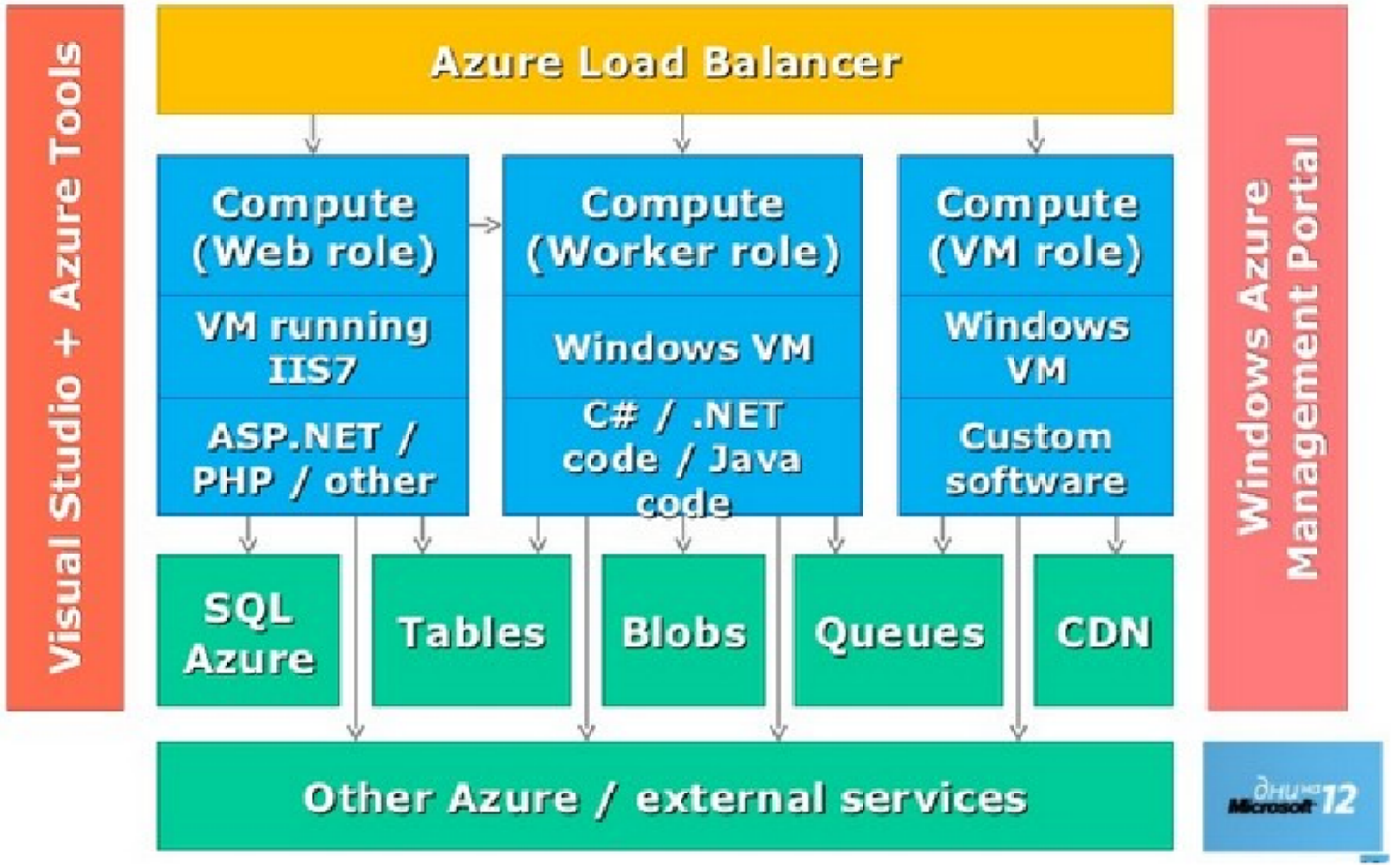
AppFabric connects local applications with applications running on Azure

Windows Azure Platform Appliance Local Azure installation

$\Rightarrow$ Supports a range of development platforms, environments and tools

$\Rightarrow$ More choice over the available compute resources than App Engine

Microsoft Azure — Architecture



Source: `slideshare.net`

Microsoft Azure — Services

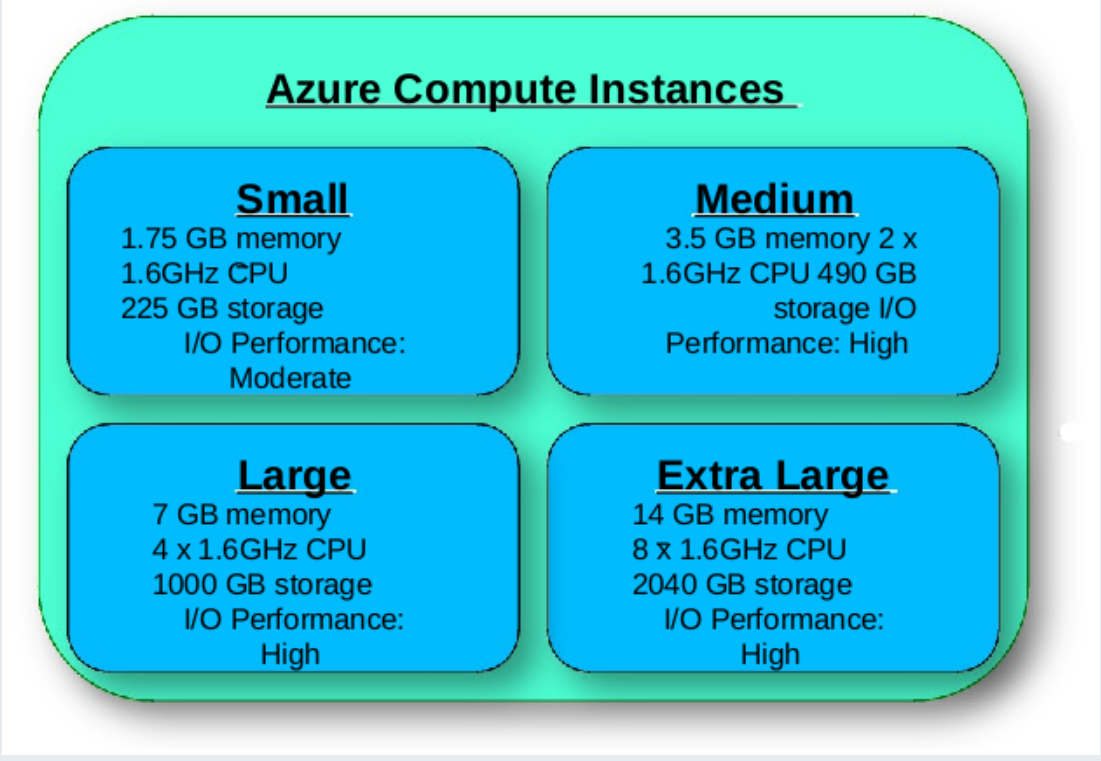
Compute Services

- Hosting environment provides a run time environment for managed code
- Azure platform offers a basic unit of execution called a role, several of which form a service and execute in the run-time environment
- Roles can be of two types:

  Web roles web application running ASP.NET

  Worker roles general application that may be used to perform background processing for the web roles
- The run-time platform will execute code based on the .Net Framework
- Roles can access web resources via HTTP/HTTPS
- Web roles can communicate with a single external end-point via HTTP/ HTTPS, whilst worker roles can communicate with up to 5 end-points via HTTP/HTTPS/TCP

Azure Compute Instance Configurations



Each instance is a Virtual Machine running in a hypervised environment

- Compare with AWS IaaS
- Azure offers specific instance types but you cannot log into them (ssh)
- Multi-tenancy based system, so physical resources are shared between instances

Microsoft Azure — Typical Costs

General purpose compute instance pricing rates, basic tier, Windows; Linux available

Small instance (A1) (1 Core, 1.75GB RAM)	$0.024/hr
Medium instance A2	$0.068/hr
Large instance A3	$0.176/hr
Extra large instance A4	$0.352/hr

Source: Microsoft (`https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/`)

November 2017

Microsoft Azure — Services

Storage Services

Within Azure, multiple levels of storage are available

Binary Object Large Object (Blob) text or binary data storage

Queue Service persistent messaging between services

Table Service structured storage

Azure Drive mountable virtual storage for compute instances

DocumentDB NoSQL document database

Services accessed by compute instances or directly (HTTP/HTTPS)

SQL Azure

- Fully featured SQL relational database running on the Azure cloud platform
- Available to both Compute instances and external applications
- High scalability, availability and reliability

Microsoft Azure — Costs

Data Transfers: Free

Data Operations: Write $0.05, Read $0.004 per 10,000 operations

Standard instance storage charges

First 50 TB	$0.0184 GB/month
Next 450TB	$0.0177 GB/month
Over 500TB	$0.017 GB/month

SQL database

50 DTU, max 500 GB	$0.1511 /hour
100 DTU, max 750 GB	$0.3021 /hour
200 DTU, max 1 TB	$0.6042 /hour
300 DTU, max 1.25 TB	$0.9063 /hour

Source: Microsoft (`https://azure.microsoft.com/en-us/pricing/`) November 2017

Microsoft Azure — Development Tools

Development based on .Net environment

- Allows existing skill sets to be utilised without needing to re-skill the entire development team (VB.Net, C#, .Net)
- Existing applications may be more easily ported to the cloud
- Better integration between local and cloud based services
- AppFabric provides a robust and secure communications mechanism to connect services running locally, remotely and on the cloud

Simulation of the main storage services

- Allows the developer to test programming interfaces and data access methods
- Performance testing may not be realistic

Microsoft Azure — Development Tools

The "development fabric" simulates the full Azure run-time environment:

- Services can be executed and tested locally before deployment to the Azure platform
- Full debugging available in the development fabric
- Cannot dynamically change the number of deployed instances
- Some aspects of Azure cannot be fully simulated, such as the Load Balancer service

Microsoft Azure vs Amazon AWS

Azure

IaaS	Azure Virtual Machines
PaaS	Web Roles — for hosting front end Web Applications
	Web Sites — for developing and running web apps

Amazon

IaaS	Amazon EC2 (Elastic Compute 2)
PaaS	Elastic Beanstalk (Beta) — to create web applications

Microsoft Azure vs Amazon AWS

Azure Web Role (PaaS)

Scalability

Up Instance size
⇒ Requires a new deployment

Out Easily add more instances in the management console

Availability & Disaster Recovery

- 99.90% for single instance
- 99.95% for load balanced instances (2 or more)

Elasticity
⇒ Enterprise Library Auto-scaling Application Block (WASABi)

Microsoft Azure vs Amazon AWS

Web Application -Amazon Elastic Beanstalk (PaaS)

Scalability:

Up Instance size (management console)

Out Easily add more instances (management console / Auto-Scaling pane)

Availability & Disaster Recovery
$\Rightarrow$ 99.95% for single instance

Elasticity:

- Easy (in management console)
- Use CloudWatch + Command line procedure for advanced settings

Web Applications — Conclusion

Azure Web Roles vs Amazon Elastic Beanstalk

Azure Web Roles

- Simpler & focus on application
- Less Control

Amazon Beanstalk

- More Features
- More Control
- Supports larger Compute & Memory needs

Microsoft Azure vs Amazon AWS

DB Storage

Azure

- IaaS — Azure Virtual Machines
- PaaS — Azure SQL Database

Amazon

- Iaas — Amazon EC2(Elastic Cloud Compute)
- PaaS — Amazon RDS

Storage- Azure SQL

Scalability

- Up — No control on instance type (Host)
- Up — DB size (Automatic scaling)
- Out — SQL Azure Federations
  - Based on sharding pattern
  - Massive scalability
  - Scales with no down time
  - Certain design constraints
  - Issues & Limitations with Entity Framework

Storage- Azure SQL Contd.

Availability & Disaster Recovery:

- 2 "Standby" Replicas
  $\Rightarrow$ Same Data Center - 1 Synchronous and 1 Asynchronous
- 99.9% Monthly Availability
- Internal daily backups (Azure)
  - Min 14 days retention
  - Every 5 minutes: Backup the Log
- Import & Export functionality
  - Creates a .BACPAC file
  - Does not guarantee consistency!
  - Needs to be combined with DB Copy

References

- http://w3facility.org/question/windows-azure-vs-amazon-ec2-vs-google-app-engine/
- http://www.infoq.com/news/2008/11/Comparing-EC2-App-Engine-Azure
- http://www.cloudscaling.com/blog/cloud-computing/grid-cloud-hpc-whats-the-diff/
- https://computing.llnl.gov/tutorials/mpi/
- http://www.rackspace.com/knowledge_center/article/high-performance-computing-cluster-in-a-cloud-environme
- http://patricklam.ca/p4p/2014/notes/pdf/L22-slides.pdf
- Book — Cloud Computing: Methodology, Systems, and Applications:
  - Chapter 11 High Performance Computing Clouds
  - Chapter14 Auto-Scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds