

## GOOGLE APP ENGINE

Greg Cousin – 18204188 Practical 3

22.10.2018 03:38:34 PM

gregoire.cousin@ucdconnect.ie

### 1. What is Google App Engine?

- Google app engine is a deployment platform for developer to run their applications seamlessly with the cloud using any languages they want. This would mean that you would need to include real time processing technologies. As long as you can provide the code they will do the rest (infrastructure).

### 2. Describe briefly the services offered by Google App Engine.

1- app engine is able to let you use any languages that you want.

2- you have application versioning technologies, that lets you easily host different versions of your code making AB testing and rollout seamless.  
- (split traffic)

3- Security, App engine has a strong customizable firewall and can guarantee manageable SSL/TLS Certificates by default

4- Google app engine can let you see potential issues with your code including  
+ monitoring, logging, debugging and tracing capabilities = google stack diagnostics

5- modular with traffic to the app - scalable. this means that the infrastructure will scale depending on volume and traffic

6- google app engine lets has instance balancing capabilities on multiple availability zones so your app can be up and available.

### 3. What is Google Compute Engine?

Google compute engine is the infrastructure as a service that google cloud has.

It has capabilities to let you deploy virtual machines on demand.

This is the technology used for google's search engine,

Gmail, youtube and all other google related apps.

1: high-performance, scalable VM's

2: industry leading price & performance

3: fast and efficient networking

4: environmentally friendly global network

5: flexibility for every workload

<!-- PART TWO -->

### 1. Do you think that the homogeneity of a large-scale distributed systems is an advantage?

Justify your answer.

Yes, I believe that with a homogeneous Distributed system to be beneficial for the following. the most obvious reason being that if all sites are aware of each other and with identical instances you will have a more secure and reliable system. Since they are aware of each other due to their homogeneity, it will become easier to process proper requests.

However, after adding a heterogeneous system to your cloud service, this could also just as well augment the complexity of an already complex ecosystem.

### 2. Which components of hardware homogeneity are the most relevant? why?

GPUs, memory, multi-core CPUs and other like threaded based hardware. This is because these components use homogeneity at its base technology in its parallel complexity.

### 3. What components of software homogeneity do you think are the most relevant? why?

Parallel Virtualization, Network along Virtual machine infrastructures are the closest to software homogeneity-based technologies.

For instance, since the goal of a homogeneity Network system is to combine multiple computers together. Network systems rely highly on the homogeneity between its systems.

<!-- PART THREE -->

### 1. Compare the three cloud computing delivery models, SaaS, PaaS, and IaaS, from the point of view of the application developers and users.

To developers SaaS has a way of managing software and its deployment to third party services. Making coding and integration simple. common examples of SAAS include Google Apps, Salesforce, Concur and GoToMeeting. It lets developers integrate a client's work, datasets and project seamlessly between machines. Whereas PAAS would be the communicator between SaaS and other cloud services: utilizing a life cycle based

on os, database, software, hosting, network access, security and scripting. A developer has to be wearier of hardware processes when dealing with IaaS. He has to take in consideration how much virtual space he is going to need to partition, the cloud hosting types he can provide based on hardware, what Data center to use, bandwidth and load balancing.

2. Discuss the security and the reliability of each one of them.

In SaaS -

SECURITY and RELIABILITY: there are a lot to take in account for SaaS and security and reliable protocols. I would say build

a: an access management control identity system where it will ensure that the SaaS application are only accessible by the correct user.

b: A data control system including encryption, data loss technologies, tokenization and more. This would also make sure that the user knows that he is using his own credentials.

c: monitoring system. where you can see where and what is happening while you are using the front end of the application if necessary.

In PaaS -

There are many aspects to take in consideration between building robust applications and utilizing proper PaaS in a secure and reliable way.

PaaS Security depends on its protocols instance lifecycles and application threat modeling. While its reliability is based on the algorithm aspect of these instances while also making sure that even with a high amount of traffic there is a balanced power consumption. You can use technologies such as SQL injection to verify the robust quality and real time issues with your platform. Utilizing a good monitoring system is key to finding bugs within a PaaS hyper-scale system.

3. Analyze the differences between the PaaS and the IaaS.

In an IaaS environment, users are responsible for managing data, runtime, applications and middleware but the IaaS providers still have to provide virtualization, servers, space and other networking services. These company providers are now also stating to provide database type services. however, in a PaaS you are more dealing with the back end, how SaaS is being transmitted and its transmitted, and its architecture.

4. An organization is trying to decide whether to adopt private cloud or a

public cloud, e.g., the AWS, for its computational and storage needs, asks your advice. What information will you require to base your recommendation on, and how will you use each one of the following items:

a. the description of the algorithms and the type of the applications the organization will run

- Since we are a database company we need to have a very tight secure way of monitoring our client's data. We are heavy on SQL injections and provide users to save their data on our storage. Rather than outsourcing the placement of our infrastructure It would be more practical for us to keep our client's data private within our company. this would mean converting into a private cloud ecosystem. This would also be due to creating monitoring applications to make it easier and faster for us to reach out back to them.

b. the system software used by these applications

- the system software we will use will be a combination of SaaS and IaaS. We need to invest in our own hardware but because we are building a UI for our clients it would be the most pragmatic solution.

c. the resources needed by each application

- the resources will de determine of the consummation of our application. We would need to put in place the fallowing: network for communication, machines for distribution, a full cloud-based ecosystem, task scheduling, monitoring and fault prevention.

d. the size of the user population

- if the business has already a lot of left over hardware and are interested in adopting a private cloud system. They might run out of space quickly since they will be responsible for the client's populations. This is why using a public cloud might be the better option.

e. the relative experience of the user population

- a user will feel safer with his data keeping a one on one relationship with the company. Which is why I would recommend utilizing a private cloud.

f. the costs involved.

- in a public cloud the cloud provider will take control of all hardware, software and other infrastructure. Which is a benefit, but if the company needs to priorities privacy and have adopted a system that is faster for their customers, a private cloud would be better.