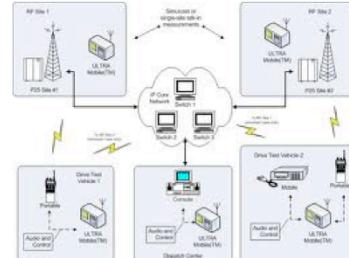




# Module Structure

- \* Where are you now?
  - \* Programming Languages?
  - \* Web Services?
  - \* Mobile Development Kit?
  - \* Design Skills?
  - \* Infrastructure?
    - \* Web Applications
    - \* Networking
    - \* XML/JSON



Good to start a route by knowing where you are.



The slide has a blue header bar with a small image of a mobile device on the left. The main title "Module Structure" is in white text on the right. Below the title is a bulleted list of requirements. To the right of the list are logos for Google App Engine, Android, Amazon Web Services, and iOS.

## Module Structure

- \* Where do you need to be?
  - \* At least one platform
    - \* Possibly many, using X-Platform tools
  - \* Web Service development
    - \* Build services using standard kit (e.g.) Java, .Net, Node.js, Python+Django
    - \* Build cloud apps using (e.g.) Azure, Amazon EC3, Google App Engine
  - \* UI Design for 'limited' devices
    - \* Android, WinMobile, iOS, jQuery Mobile (HTML5+JS)



Google App Engine logo (blue and white text with a globe icon).  
Android logo (green robot icon).  
Amazon Web Services logo (yellow cubes and text).  
iOS logo (silver text).

I'm not being prescriptive about what technologies you use in this module.  
You'll be given goals to meet, you choose the tech that can meet them.

## Module Structure



- \* Getting there
  - \* Module Lectures cover principles
    - \* What we need to create to provide various facilities
    - \* Core practices in development
  - \* Labs are not “led by the nose”
    - \* Given a basic requirement (e.g. create a simple user-interface for a mobile device)
      - \* YOU select development kit
      - \* YOU find suitable examples and tutorials
      - \* i.e. You are in the driving seat in this module – I’m just the driving examiner

Often this will involve you going through online tutorials for a specific technology (e.g. the jQuery Mobile ones).

## Module Structure

- \* What Resources?
- \* In the labs...
  - \* WebStorm (for web-apps/jQuery Mobile/Bootstrap etc.)
  - \* Android Development Kit
  - \* Google App Engine Dev Kit
  - \* Azure Dev. Kit and Licences
- \* Good online tutorials for each of these is available
- \* In the absence of other experience, you would be best to concentrate on Web Apps/jQuery/ jQuery Mobile for client app development
- \* My preferred kit for service development is Google App Engine
  - \* If you're happier with Apache+PHP/IIS+.NET or whatever, feel free to use that

See online sources:

[www.jquerymobile.com](http://www.jquerymobile.com)

[www.google.com \(app engine\)](http://www.google.com/app_engine)

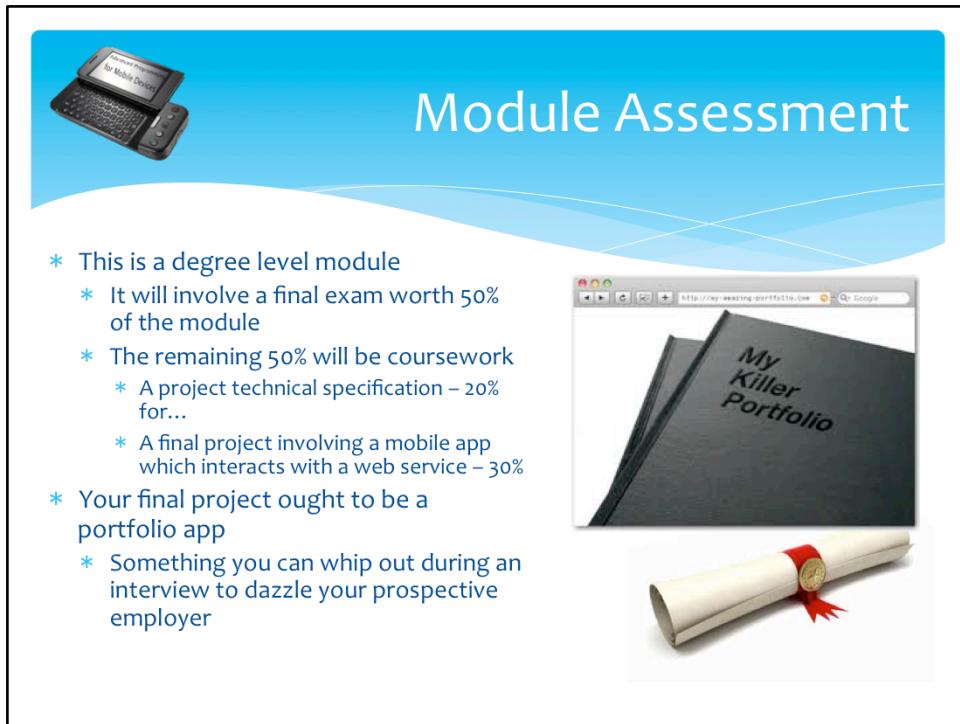
etc.

# Challenges

- \* Each week (starting next week), I'll give you a simple task e.g.
  - \* Build a simple 4-function calculator for a mobile device
  - \* Create an online data storage system
  - \* Integrate mapping into a simple app
- \* I'll point you to various online tutorials to get you going (you don't need to use them, but then you need to find your own – unless you already know how to do it)
- \* You'll get started on building a mobile app (sometimes even a desktop one – depends on the problem) or a cloud service to do the job
  - \* I'll help where help is needed
- \* Some of these tasks will be critical to your success in the module, and will be assessed the following week

The screenshot shows a mobile application development environment. On the left is a smartphone icon displaying a news article from 'ESPN Mobile Web' about 'Barry And Advance'. On the right is a window titled 'Mobile' with a menu bar. The menu bar includes 'File', 'Edit', 'View', 'Project', 'Mobile', 'Web Runtime', 'Community', 'Settings', 'Help', and '(iPhone Services)'. The main area of the window shows some code or configuration options.

Aaarrggghhh..



The slide has a blue header with a mobile phone icon and the title "Module Assessment". The main content area contains a bulleted list of assessment details and an image of a portfolio and scroll.

**Module Assessment**

- \* This is a degree level module
  - \* It will involve a final exam worth 50% of the module
  - \* The remaining 50% will be coursework
    - \* A project technical specification – 20% for...
    - \* A final project involving a mobile app which interacts with a web service – 30%
- \* Your final project ought to be a portfolio app
  - \* Something you can whip out during an interview to dazzle your prospective employer



Even more Aaaarrrrggghhhh!!!!

The slide has a blue header bar with the text "Questions?" and a small icon of a mobile device (a flip phone) in the top left corner.

**\* Any questions about**

- \* Lectures?**
- \* Labs?**
- \* Assessment?**

**HELP WANTED**  
SEPT 2, 2005

NINTENDO EXPERT NEEDED  
\$50,000 Salary + Bonus  
Equal opportunity employer  
LOOKING FOR GOOD MARIO  
BROTHERS PLATER \$100,000  
plus your own car. \$5000\*

IF YOU HAVE 5000 HRS  
OR MORE OF VIDEO GAME  
EXPERIENCE WE NEED YOU  
DO YOU LAUGH IN THE  
FACE OF KILLER GOOMBAS?  
CATCH US A \$80,000  
plus a free house now  
CAN YOU SAVE THE PRINCESS?  
WE NEED SKILLED MEN & WOMEN  
\$75,000 + Retirement + Benefits  
Expanding Company needs Skilled  
Computer games operator Call  
521...

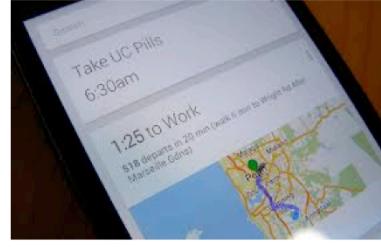
VIDEO PLAY  
PIGIGA ADVENTURE  
\$120,000  
NINTENDO  
OFFER  
WE  
SUPER MARIO BROS EXPERT  
\$45,000 yr. Four day  
work week + Ferrari.  
DO YOU KNOW A NINTENDO  
EXPERT? Please read  
him or her this ad

**Hopeful parents**

The cartoon shows two parents standing in front of a child who is sitting on the floor playing a video game. The father is holding a newspaper with a large "HELP WANTED" ad for Nintendo. The ad lists various job positions and their requirements, including a "NINTENDO EXPERT" position with a salary of \$50,000 and a "Super Mario Bros. Expert" position with a salary of \$45,000 per year and a four-day work week plus a Ferrari. The parents are looking at the ad with expressions of hope and concern.

## Mobile Apps

- \* Mobile Application Characteristics
- \* NOT – small versions of desktop apps
- \* Mobility is the key characteristic
  - \* Apps respond to geographical location, diary entries, messages,
  - \* Assistive technology is common (PDA=Personal Digital “Assistant”)
    - \* Location awareness
    - \* Information retrieval
    - \* Integration of local information with online
  - \* Apps can be always there
    - \* Business assistance
    - \* Medical assistance
    - \* Communications
    - \* Short-term storage
    - \* Collaboration at a distance



### Mobile Application Characteristics

#### NOT – small versions of desktop apps

Even when they are – they generally have different design objectives

#### Mobility is the key characteristic

Apps respond to geographical location, diary entries, messages,

Assistive technology is common (PDA=Personal Digital “Assistant”)

Location awareness is an obvious feature

Information retrieval

Integration of local information (on device) with online data -

e.g. where am I? (local) and where is the nearest station?  
(online)

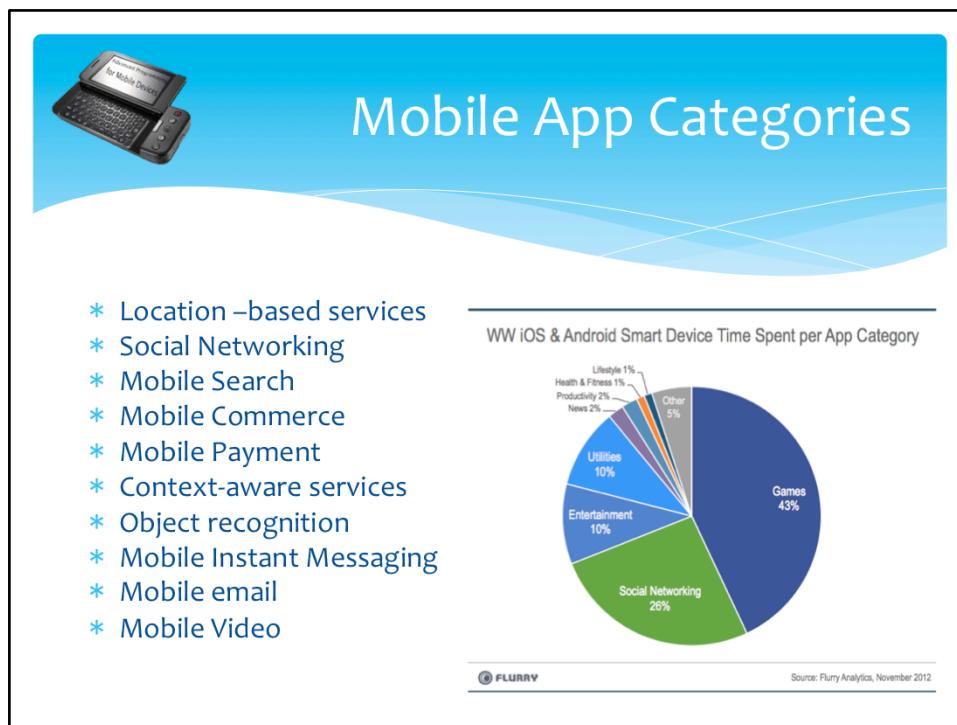
Apps can be always there – a useful feature for many tasks

Business assistance – reminders, info on tap, calculations, push notifications

Medical assistance (e.g. remember to take your pills, check blood pressure, etc.)

Communications – exchange key info with colleagues

Short-term storage – e.g. store information gathered on the road for later upload



### Location –based services

Mapping, Directions

Use device GPS to direct downloading of map-segments, calculation of routes

### Social Networking

Facebook, Twitter

Simplified mobile client apps

### Mobile Search

Around me

Use locale information (from GPS) to perform online database search

### Mobile Commerce

Amazon mobile app

Shopping anywhere

### Mobile Payment

NFC

Interact with NFC terminals (e.g. fares, small item purchases)

### Context-aware services

Google Now

Integrate device data (e.g. diary, contacts) with locale and online data (e.g. current location, train timetables) to provide timely information

### Object recognition

Google Goggles, Augmented Reality



## Apps use of data connections

- \* Currently popular mobile apps fall into several categories of 'connectedness'
  - \* Portable assistance – e.g. calculator
  - \* Portable entertainment – e.g. music player
  - \* Online information sources – e.g. share prices, locale data, online banking, weather, photo uploading
  - \* Online resource access – e.g. eBook reader, astronomical data
  - \* Online information interpretation – e.g. Google Goggles, Google Now



Currently popular mobile apps fall into several categories of 'usefulness'

Portable assistance – e.g. calculators (normal or task –specific). e.g. a calculator, a tax calculator, photo editing or management

No (or little) need to access online data

Portable entertainment apps – e.g. music/video player - no need for a permanent connection

Online information sources – e.g. share prices, locale data, online banking, weather, photo uploading (e.g. to Instagram/Flickr)

An internet connection is essential for the app to do its job

The app maintains an online presence on the move

Online resource access – e.g. eBook reader, astronomical data

An internet connection is needed to access information, but this can then be stored locally on the device

The app accesses online data stores, but caches the data for mobile use

Online information interpretation – e.g. Google Goggles, Google Now

The app integrates online data with device information (e.g. locale, PIM etc.) to assist in personal scheduling/information retrieval



Infrastructure is whatever external resource is needed to make a mobile app possible.

At its most minimal, this can be a delivery site (e.g. the Apple App Store or Google Play)

Many apps have needs beyond this – e.g. maps need constant updating as the locale changes

In all apps of this type, there is a need to balance the locus of the processing demand

Device app (i.e. the client) does presentation, but also local processing (e.g. integrating diary or contacts app)

Online processing does things like looking up data-sets, filtering data sent to device (to cut down needed bandwidth)

In most cases, a sweet-spot is where the local and online processing is balanced so that each performs the processing that it is best placed to do.

## External Data

- \* Integrating external data can bring complexities to a mobile app
- \* Sporadic connections (e.g. train goes into a tunnel)
- \* External data formats (e.g. a web service can provide XML, JSON, JSONP, some proprietary format, HTML..)
- \* Type of app can impose limitations
  - \* e.g. a web-app is limited by the single origin policy
  - \* Can be got-around using JSONP, but only if an external data source provides it
- \* In most situations, mobile web-apps will benefit from the provision of a “delivery” site that also provides data access

| ID | Type    | Display format    |
|----|---------|-------------------|
| 1  | String  | (A/_)             |
| 2  | Integer | [+ -] (9)         |
| 3  | Real    | [+ -] (9) (. (9)) |
| 4  | Data    | yyyy-mm-dd        |
| 5  | Time    | hh/mm/ss          |
| 6  | Data    |                   |

For a device to use external data, there must be a protocol (or format contract) between the device and its info source.

Although this can be at quite a high level (e.g. Map tiles are complete graphics), often the need is to consider

the data exchanges at a lower level. What is the raw data (at the online end), and how should this be

packaged for effective and efficient transfer to devices?

There are several standard formats (XML, JSON, JSONP, csv, proprietary formats)

There are technical barriers to providing a device with data

Apps are not free to just grab data from a source, even though this data is freely available to a browser

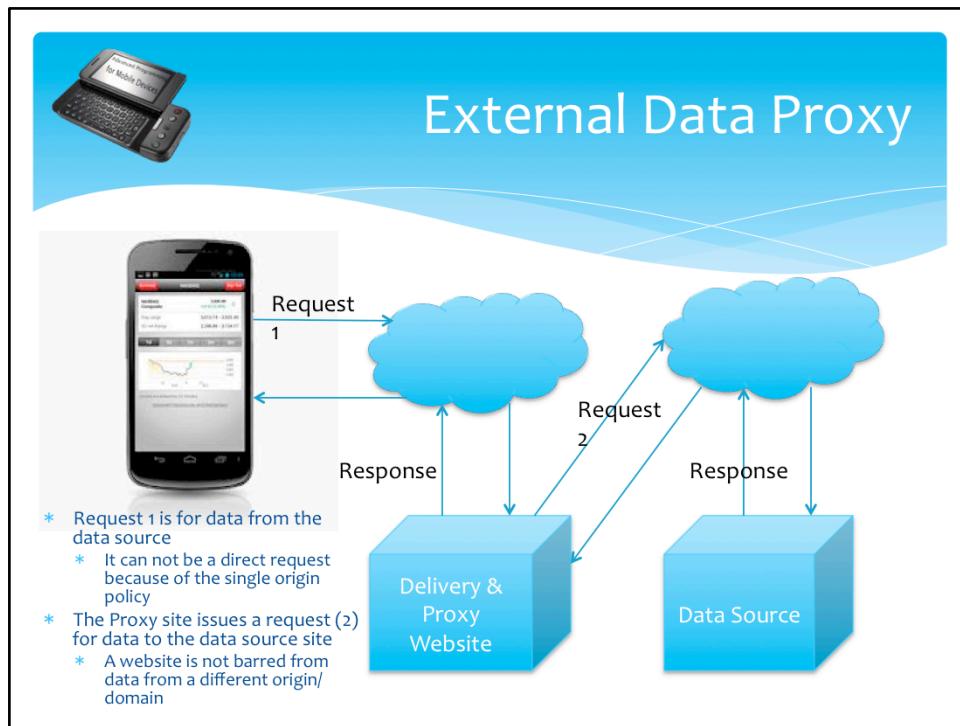
Permission needed (for data access, from the device owner etc.)

The Single Origin Policy is a particular problem – this says that a web application can not simply access data from

other sources – the browser restricts it to access data only where the data origin is the same as the app origin

in terms of Protocol (e.g. http/ftp), Domain (e.g. www.uws.ac.uk) and Port (e.g. 80 for web)

Developers need to work around this (in a variety of creative ways we'll look into later).



One route around the Single Origin Policy is to create a native app which must request permission either to be hosted (at the Apple app store) or when it is installed (like Android apps).

Another is to provide a proxy so that a web app can access online data via its original delivery site.

A third is to use a format that is allowed to cross domains – JSON/JSONP does this.



## Software Development Kits

You may (should) have a preference for some SDKs – feel free to use whichever you want

- \* Client-side development
  - \* Native SDKs
    - \* XCode for iOS
    - \* ADK for Android
    - \* Visual Studio for WinPhone
    - \* Blackberry DK for (guess)
  - \* Cross-platform SDKs
    - \* PhoneGap for any
    - \* Ionic Creator
  - \* Web-App
    - \* jQuery Mobile (HTML5/JS)
    - \* Sencha Touch (HTML5/JS)
  - \* Server-side development
    - \* LAMP (HTML + PHP)
    - \* Ruby-on-Rails
    - \* Python + Django
    - \* Java Server Faces
    - \* Google App Engine SDK
    - \* IIS + Visual Studio for ASP.NET
    - \* IIS (+ notepad??) for ASP
    - \* Node.JS (Javascript)

This is a short list of kits. There are many more, usually a bit more obscure or limited.



## Examples in the module

- \* There will be example/demo implementations to cover all of the core components of a distributed mobile app
- \* There are too many SDKs around to provide examples of all of these, so I'll limit the examples to...
  - \* jQuery/jQuery Mobile for HTML5 + JS client apps
    - \* Advantages: cross platform, single language, optimised for mobile, zero cost
    - \* Disadvantages: some native features not available, UIs less slick
  - \* Google App Engine (Python version) for server apps
    - \* Advantages: Quick and simple development, in-built distributed database, free deployment to Google's servers (for up to 6 million requests per month), built-in templates engine, also available for Java and Google Go languages
    - \* Disadvantages: Google.

In some cases, I'll use apps already deployed on App Engine for you to practice  
Build a mobile app that access some data fed from a JQM source  
Find these in Moodle

Why am I still using jQuery Mobile???

One good reason is that jQM strongly supports the idea of a true Single Page Application (SPA), and uses AJAX to switch between multiple pages in a single HTML file. Other Mobile Web App platforms I've examined tend to promote the idea of converting standard web content into web apps, and typical examples from their websites involve multiple HTML files making up an app. That is an exception in jQM (the recommendation is that all pages are in a single HTML file for normal web-app functionality, with separate HTML files used only for seldom accessed functionality – registration, app configuration etc. The result is that jQM apps tend to work faster once they are installed on a device (provided they have been programmed well). Once Ionic decides that this is a more sensible approach, I'll consider adopting it.