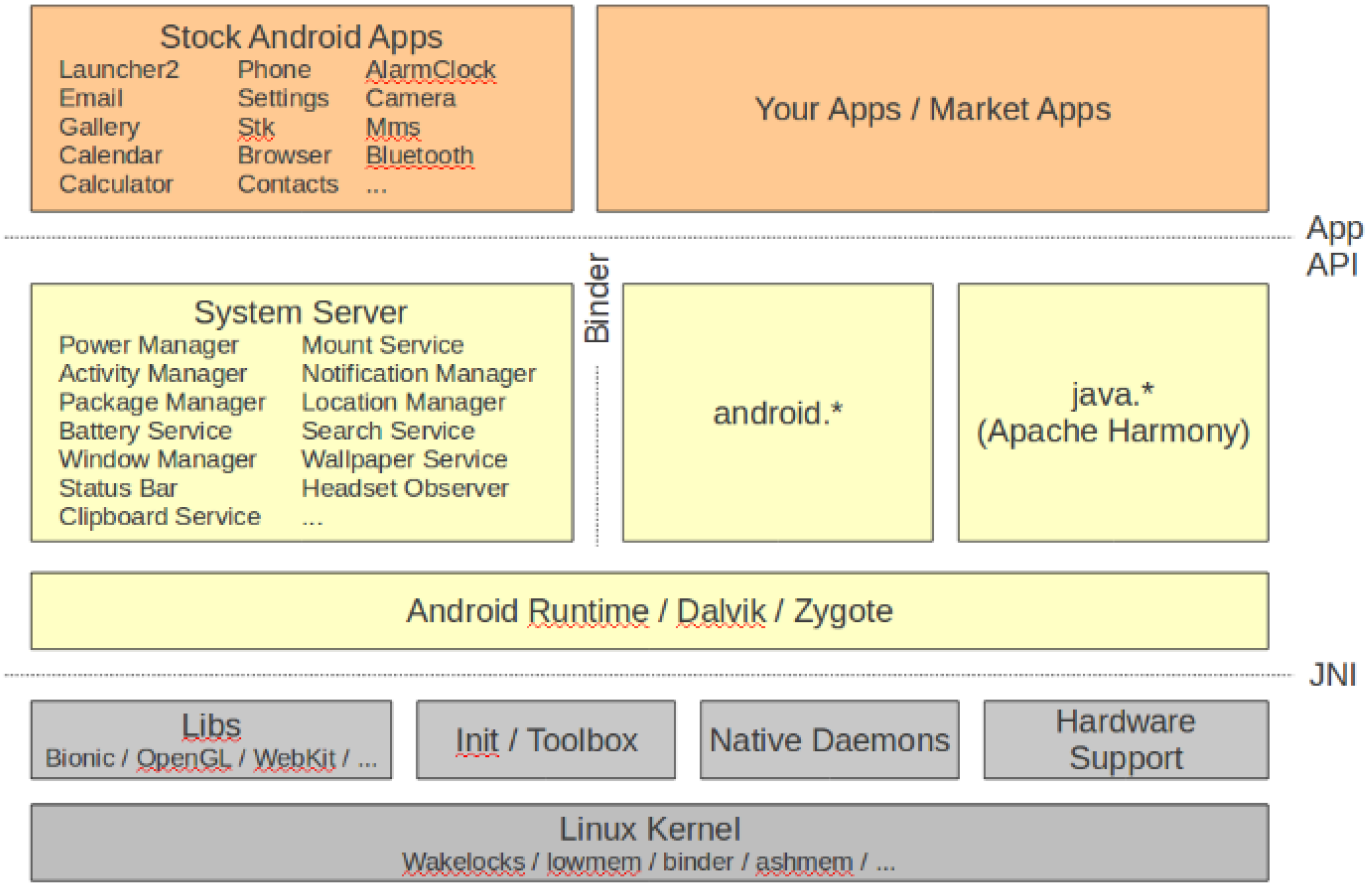
G54MDP
Mobile Device Programming

Lecture 20 – Cross-platform, Revision

Cross Platform?

- Apps developed for one system won't work on another
- Would need to port it over
  - This can actually be desirable
    - Can tailor our app to the look and feel of the target device
    - Apple encourage the use of iOS "metaphors"
      - Sliding on/off switches, spinning picker wheels
  - Significant coding effort
- However, there are times when it is desirable to target multiple platforms
  - In-house apps
  - Games (Platform chrome usually irrelevant)
- What are the issues behind cross-platform support?



SDK, Eclipse, .apk

Manifest:
Perms / SDK ver.

.dex, ddms

NDK, rootfs, initrc, adb

GNU toolchain

(fastboot)

Language

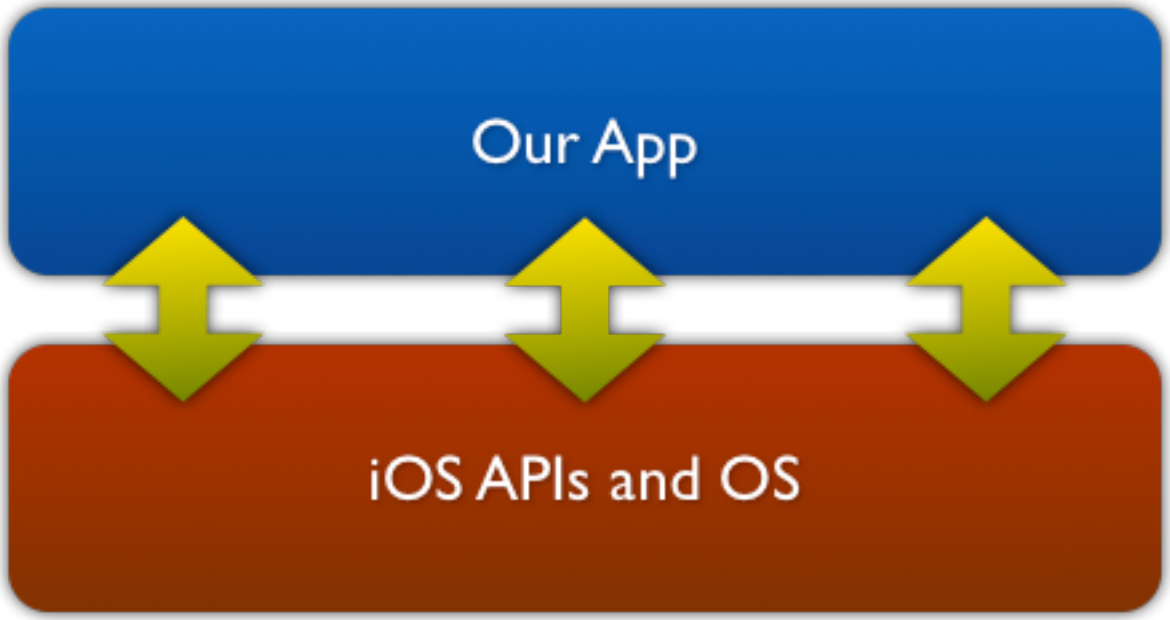
- Compile code for multiple platforms?
  - Can compile Java into native code (gcj)
    - Android supports native code libraries
  - C++ can be compiled to CLR (for WinPhone 7)
- This would work at a technical level
- The code would execute
  - But the app wouldn't run
  - Each platform has different APIs…
    - Android — Activitys, Intents, Services…
    - iOS — Views, ViewControllers,
- Could port the app logic relatively easil;y
  - But would still need to rewrite the UI
  - This maybe enough for games
    - OpenGL ES supported across several platforms
  - End up rewriting boilerplate UI setup stuff

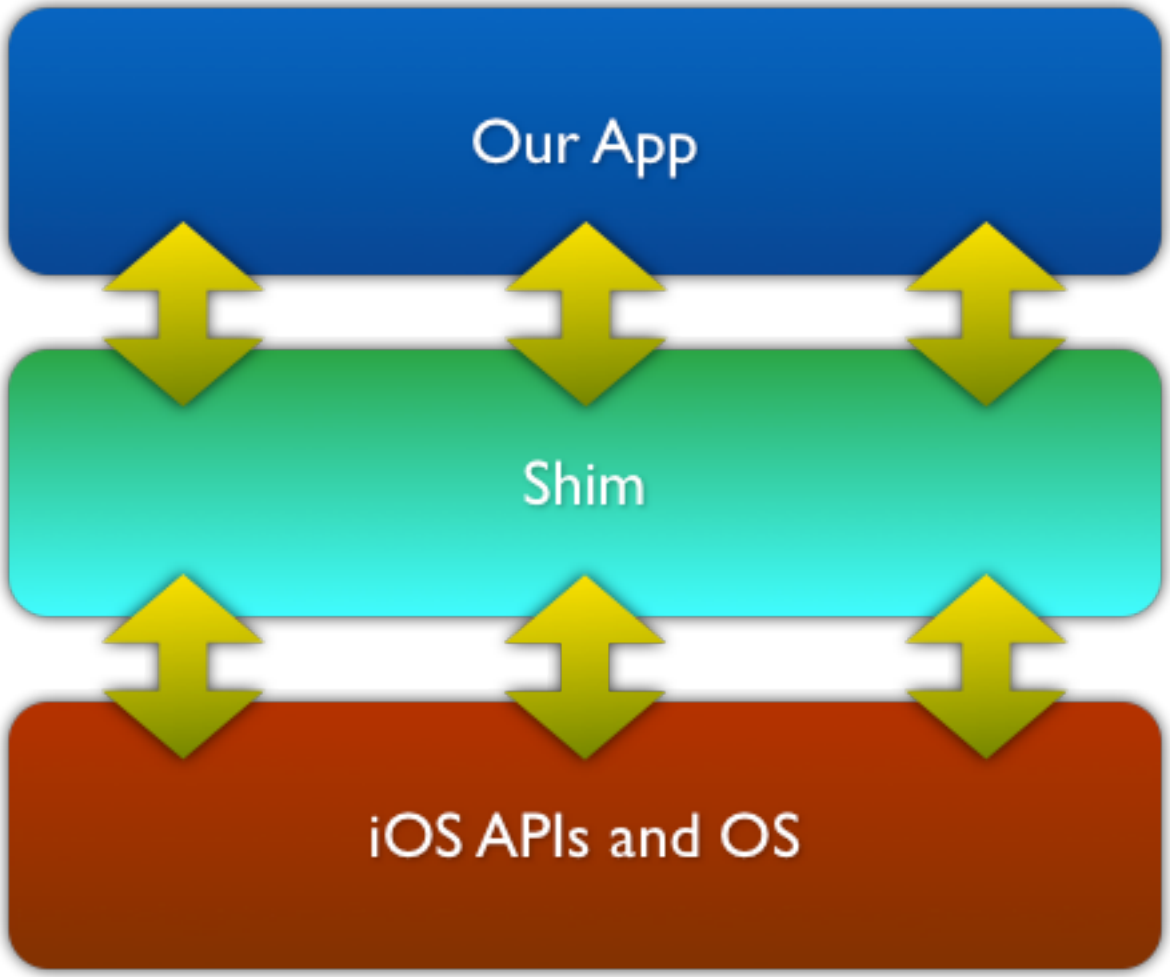
Language

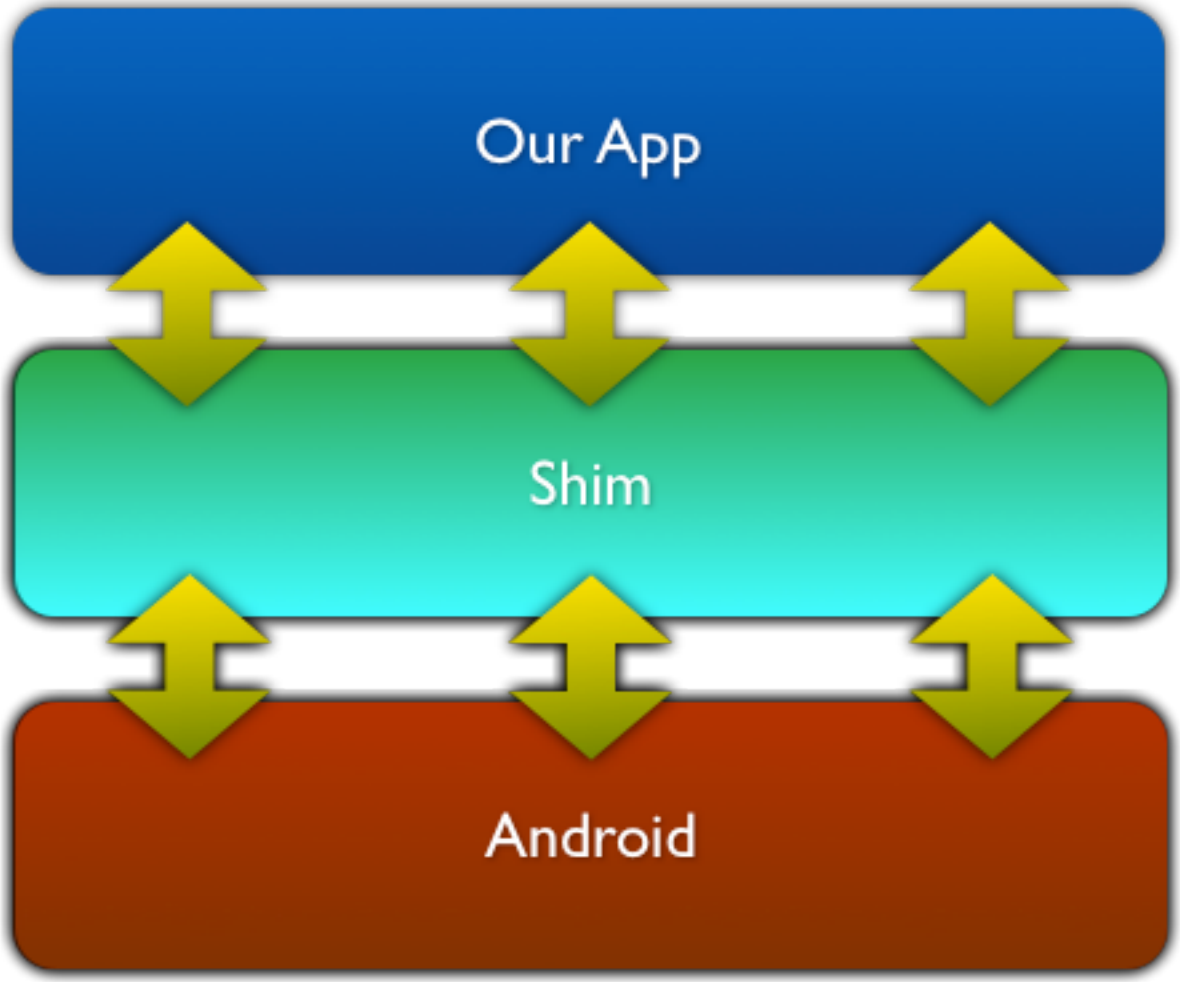
Platform	Language
Android	Java / C++
iOS	Objective C
Blackberry	Java, some Android support
Windows Phone	C#
webOS	C/C++ or HTML/ Javascript

Truly Cross Platform

- Assuming that we can compile code for each device
- To be completely cross platform we can insert a **shim** between our code and the APIs
  - Effectively abstracting our code from the original APIs
  - Our code calls our abstraction
- To port to another device, change the shim and recompile with the appropriate tool-chain
  - New shim provides the same interface to our app
  - But implements it using the native APIs of the new platform


Our App
iOS APIs and OS


Our App
Shim
iOS APIs and OS


Our App
Shim
Android

Adobe AIR

- Developed to let web developers leverage their existing skills to develop desktop apps
- AIR apps can be written in either Flash or HTML+JScript
- Additional libraries allow support for native APIs
  - Windows, widgets, sensors
- App packaged up using Adobe tools
  - Executed by the Adobe AIR runtime
  - Multi-platform support
  - One app — multiple runtimes
- Used for the BBC iPlayer Desktop app and TweetDeck

AIR on mobile

- Android
  - Apps are written using ActionScript
  - Using standard Flash/ActionScript libraries
  - Compiled into a SWF file as normal
    - Then packaged as an AIR app
  - Packaged into a .apk file
    - Need the AIR runtime installed
- iOS
  - Apple forbid the use of VMs on their platform
    - **2.8** Apps that install or launch other executable code will be rejected
  - .swf -> llvm -> ARM native code
  - Linked with Flash runtime written on top of iOS APIs
  - Packaged as a static app executable

Other cross-platform offerings

- Unity
  - Primarily for game development
- Phonegap, Appcelerator, App Furnace
  - Develop application as HTML / Javascript pages
    - Primarily hosted in native web component
  - Integration with native components via a shim

References

- http://phonegap.com/
- http://unity3d.com/
- http://appfurnace.com/

Exam

- 1 hr
- Answer 2 of 3 questions
- Worth 50% of your module mark
- 15 marks per question
- “State…”
- “Describe…”
- “Giving an example…”

Topics

- Mobile Device Characteristics
- Device Architecture
- Android
  - Application Lifecycle
  - Activities
  - Layout and Widgets
  - Threads and Services
  - IPC
  - Data Storage and Content Providers
  - Broadcast Receivers
  - Security and Permissions
- Touch
- Battery / Power Management

Example Question 1

- This question concerns the application components used in an Android application.
  - Name four most significant application components used in an Android application. For each component, describe its role and responsibilities, and give an example of how it might be used in a typical Android application. [12 marks]
  - Android applications are single-threaded by default. Describe the implications of this model when developing an application that must appear to be responsive to the user. [3 marks]

Example Question 2

- This question concerns the Activity Lifecycle.
  - The Android system maintains a stack of Activity objects. Outline the lifecycle of an Android Activity and the methods that are called to inform it of its current state. Describe under what conditions each of these methods would be called and the different stages of the lifecycle. [10 marks]
  - Describe the role of Intent objects in an Android application, and state how one would be used to start a new Activity or a Service. [5 marks]

Example Question 3

- This question concerns the capabilities of the Android operating system.
  - Android applications are single-threaded by default. State the implications of this model when developing an application that must appear to be responsive to the user [3 marks]
  - Mobile devices by their nature are resource limited, particularly with regard to battery life. Giving an example, describe how badly written code that makes use of the network can cause excessive battery drain, and describe how you might optimise network communication to reduce this drain [5 marks]
  - The Android operating system makes use of the Linux kernel, which enforces the "sandboxing" of individual applications, and process isolation. Describe how Android applications make use of the *binder* to support inter-process communication (IPC) and state one advantage and one disadvantage of this architecture. [7 marks]

Example Question 4

- **This question concerns Android Activities.**
  - A simple Android application consists of two Activity components. The first, MainActivity, displays a button and a text entry field. Pressing the button launches SecondActivity, which displays an OK button and a second text entry field. Pressing the OK button results in the application returning to MainActivity, which appends the text entered in the second Activity to the contents of the first text entry field.
  - Using pseudo-code or code fragments define the Java classes for the two Activities. You should include the relevant API calls that you would use to launch the second Activity, and how you would return data from the second Activity to the first in an appropriate manner. You should handle the relevant lifecycle events as appropriate for the task.
  - [12 marks]