G54MDP
Mobile Device Programming

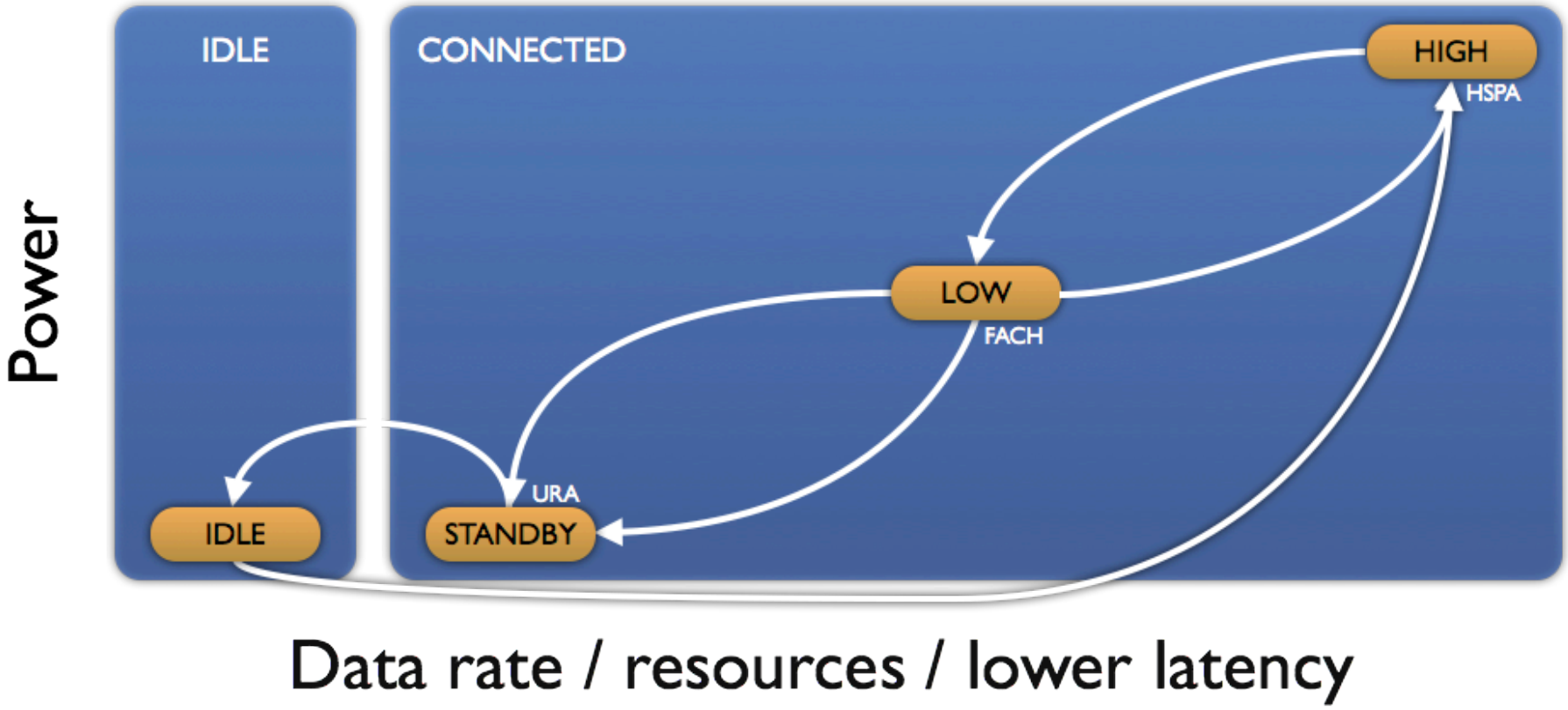
Lecture 19 – iOS, Cross-platform

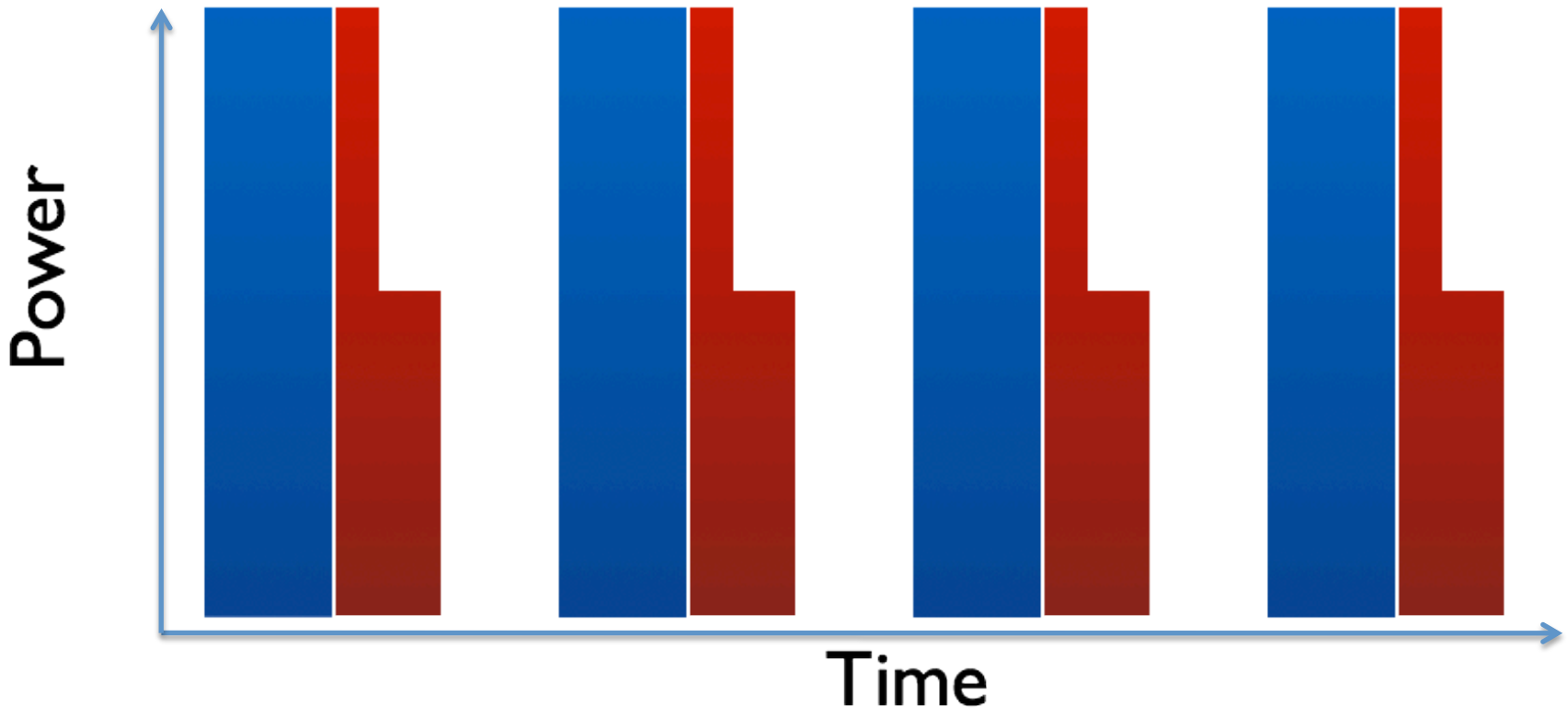
Other Chips

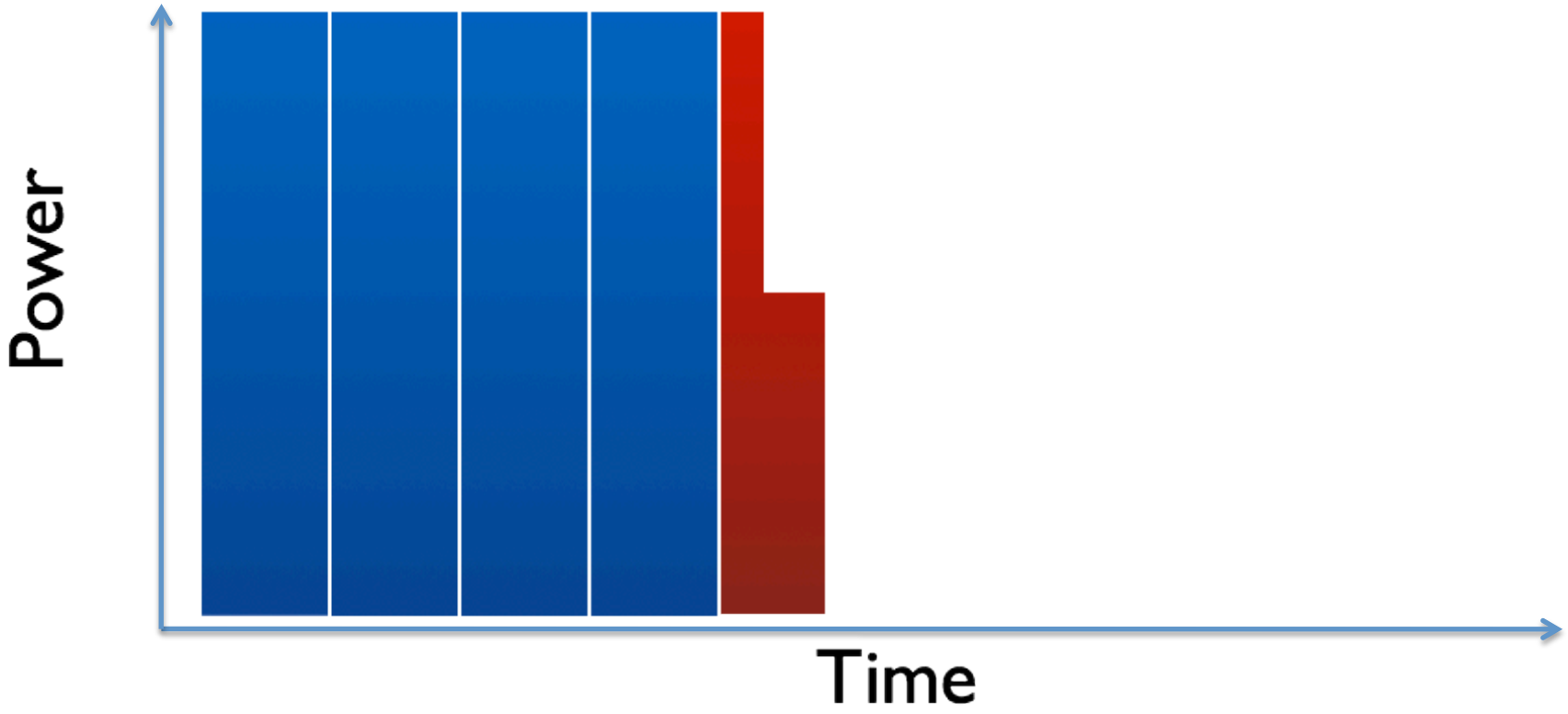
- CPU ~100mAh
- Other components use power too
  - Accelerometer
    - 10mA normal use – 80mA fastest / finest measurements
      - Choose most appropriate frequency
  - Location
    - Wifi basestations (~100m)
    - Cellphone tower triangulation (~500m – 3km)
    - GPS (~1-5m)
    - Select the most appropriate accuracy
      - GPS is very expensive in terms of battery usage, especially cold start
    - Register for updates appropriately
  - Radios (network connectivity, phone calls)

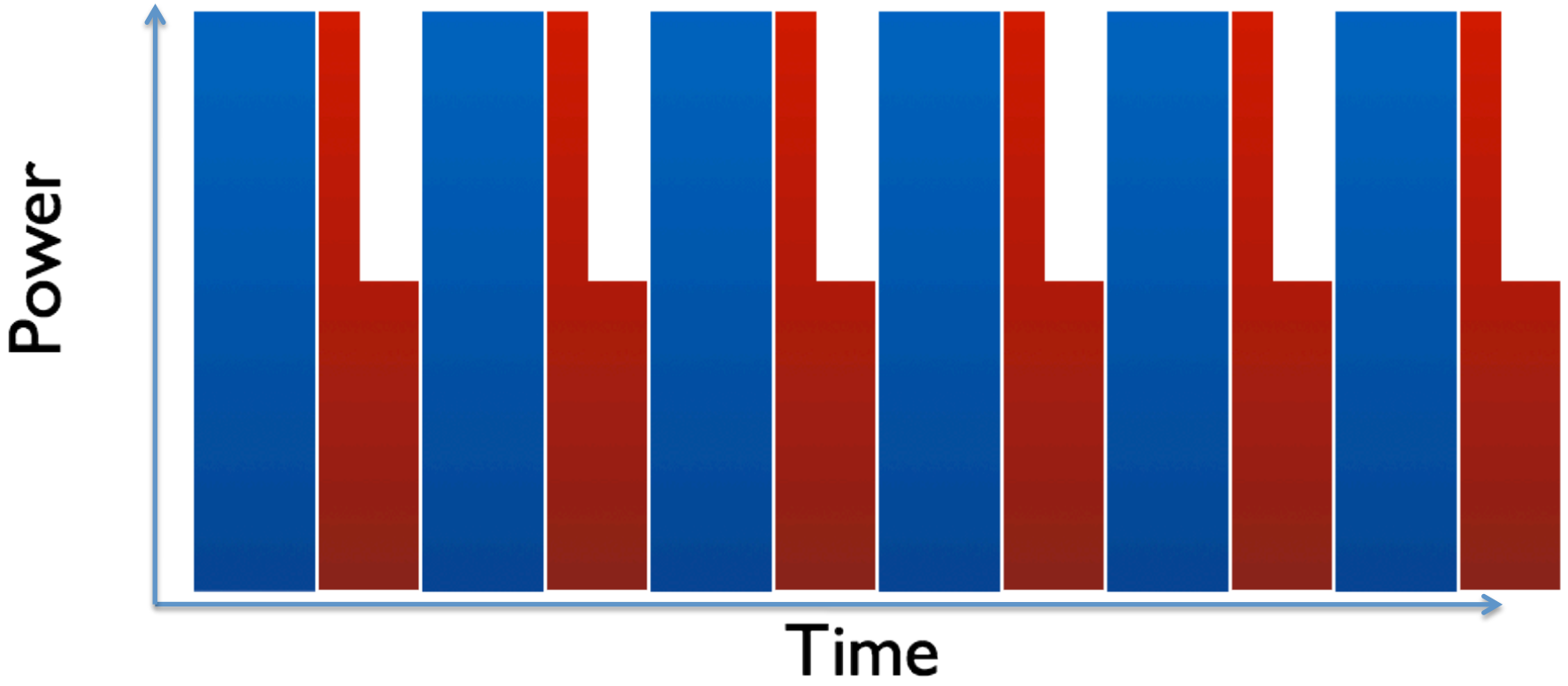
Radio / Network

- 3G chip has a number of states
  - URA – Connected but not sending data
  - FACH – Half power, small amount of data
  - HSPA – Full power, dedicated channel
- Cost / time to transition upwards
  - Ramp up power, negotiate channel
- In high power radio state
  - Delay to transmit is shorter
  - Device stays in high state for a short period of time following communication
- Regular polling keeps the radio transition between states
  - Pay the battery cost even if we transfer nothing
    - Synchronize polling – inExactAlarms
    - Coalesce data into large chunks
    - Small transfers will only transition up to low / FACH power state (~256 – 512 bytes)
  - Be careful of reusing libraries
    - Were they designed for 3G, or do they assume Ethernet


Power
IDLE
CONNECTED
HIGH
HSPA
LOW
FACH
URA
IDLE
STANDBY
Data rate / resources / lower latency

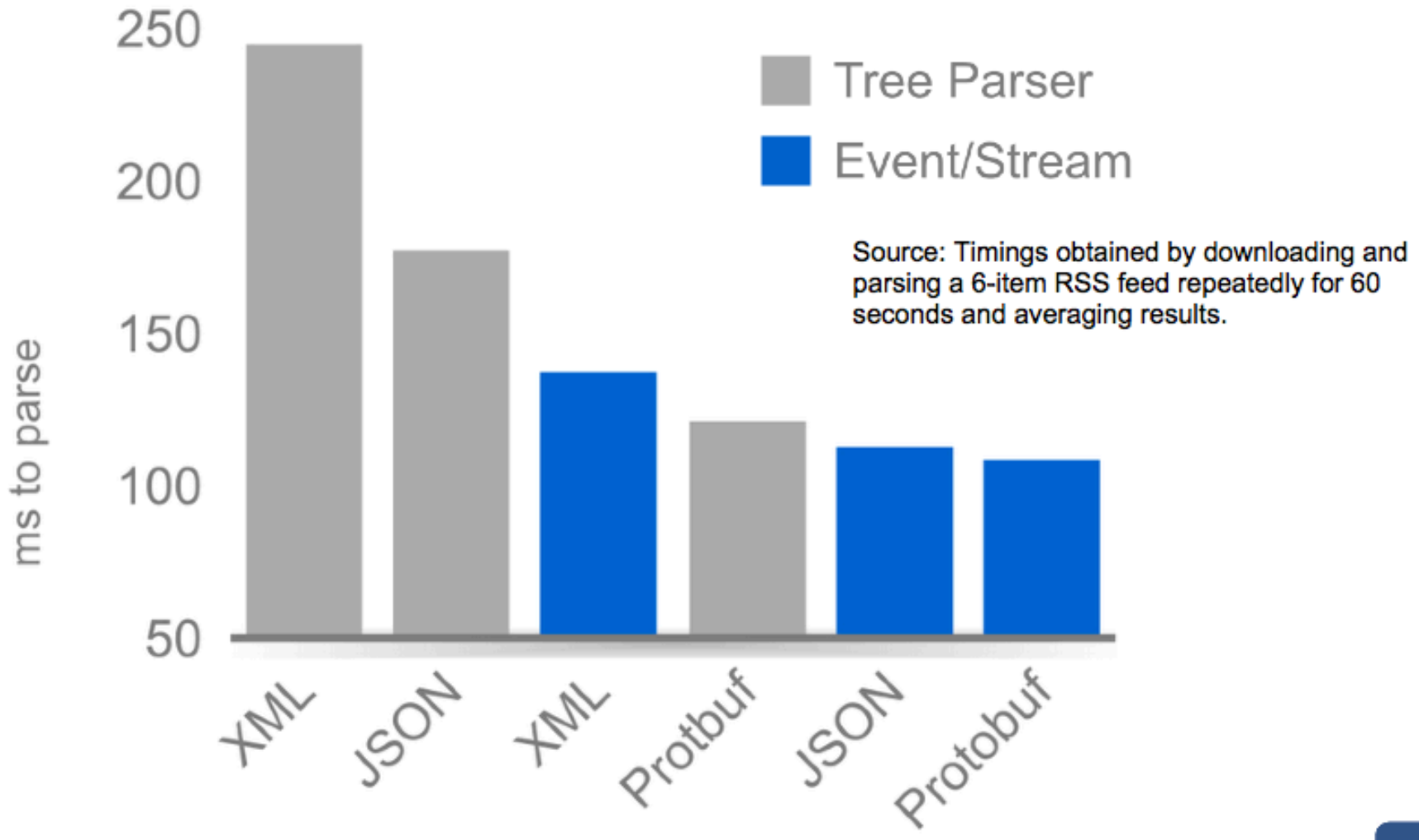

Power
Time


Power
Time


Power
Time

Data Transfer

- Battery cost per byte
  - Radio usage, CPU usage
  - Minimise the amount of data transferred
- Reduce signal-to-noise ratio
  - How much of the data describes the structure and not the data?
  - XML is bulkier than JSON
  - JSON is bulkier than binary
- Use Gzip compression where possible
  - Decompressor is native code
    - *Cost* to decompress is less than *cost* to send uncompressed
- Consider time taken to parse


250
200
150
100
50
ms to parse
Tree Parser
Event/Stream
Source: Timings obtained by downloading and parsing a 6-item RSS feed repeatedly for 60 seconds and averaging results.
XML
JSON
XML
Protbuf
JSON
Protobuf

Google 09

References

- http://www.google.com/events/io/2009/sessions/CodingLifeBatteryLife.html
- http://developer.sonymobile.com/2010/08/23/android-tutorial-reducing-power-consumption-of-connected-apps/
- http://www.slideshare.net/EricssonLabs/droidcon-understanding-smartphone-traffic

iOS

- OS of the iPhone/iPad/iPod Touch
  - Originally called iPhoneOS
  - Based (heavily) on MacOS X
- App support added in v2 — 2008
- Closed Source
  - Tools, deployment, app ecosystem controlled by Apple
- Apps can only be installed from an App Store
  - Cryptographically signed
  - Apple runs iTunes App Store
  - Approves all apps available from it
  - It is possible to set up an internal to enterprise app store

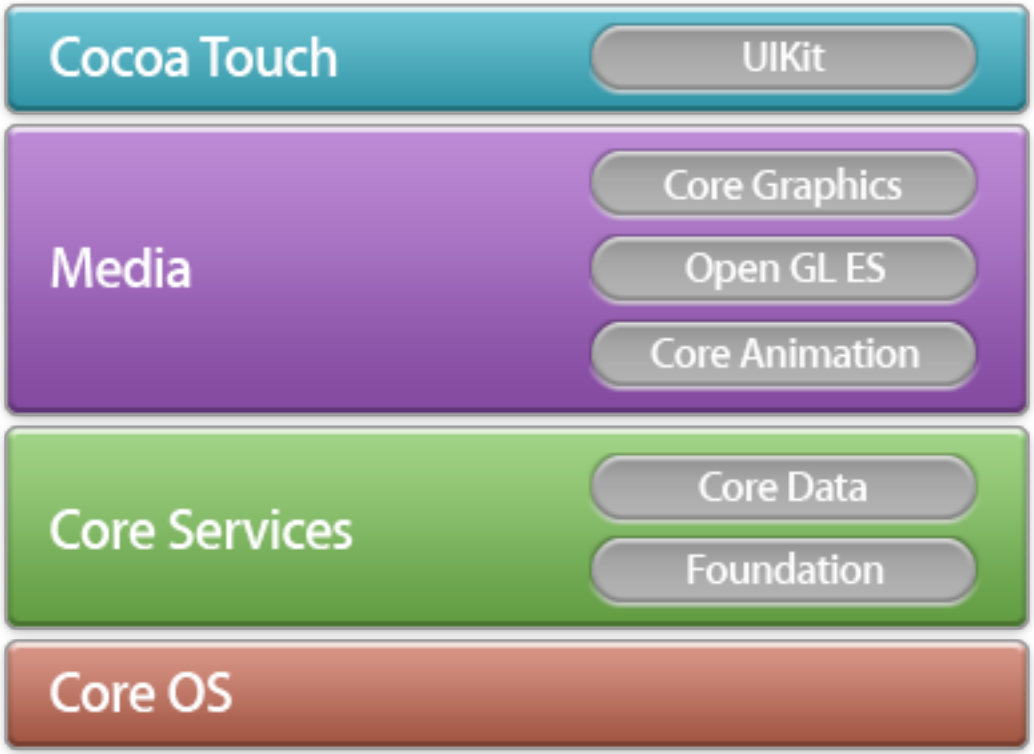
iOS Apps

- Written in Objective-C (ObjC)
  - Using the Cocoa Touch UI framework
  - Can also use C/C++ libraries
  - Compiles to native code
    - Not interpreted/JITted as on Android
- iOS uses Objective-C as its main language
  - Extension of C to add support for OO
  - Developed around the same time as C++

iOS Frameworks

- iOS comes with several frameworks that can help us with development
  - Foundation framework provides support for strings, files, collections etc
  - Other Frameworks provide support for Audio, video, animation, location etc
  - At the top is the UI framework, CocoaTouch
    - Widgets, buttons, views
- iOS is very much an evolution of PC GUI programming into the mobile space
  - Particularly MacOS X GUI programming
  - Almost every class in CocoaTouch has an equivalent in OS X
  - Vs Android major components

iOS Frameworks

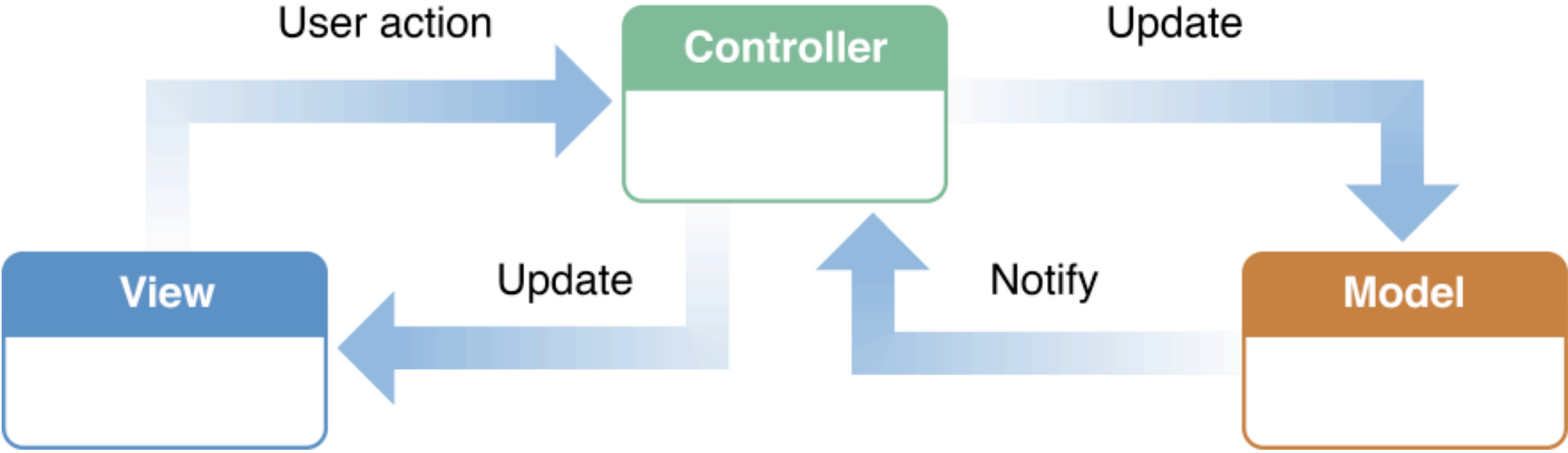


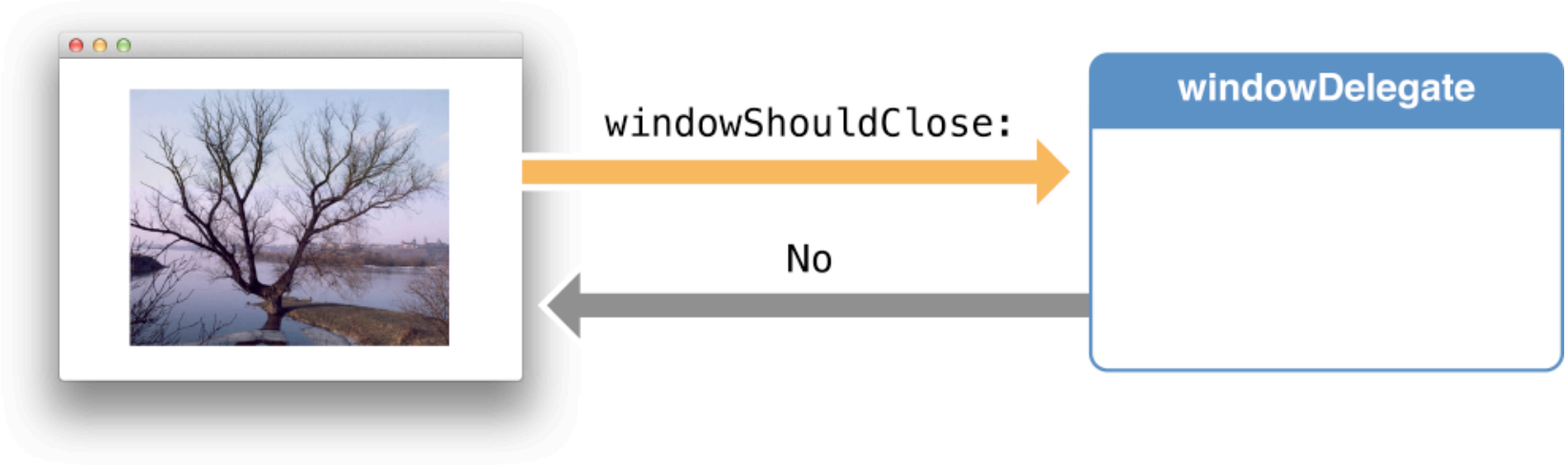
Design Patterns

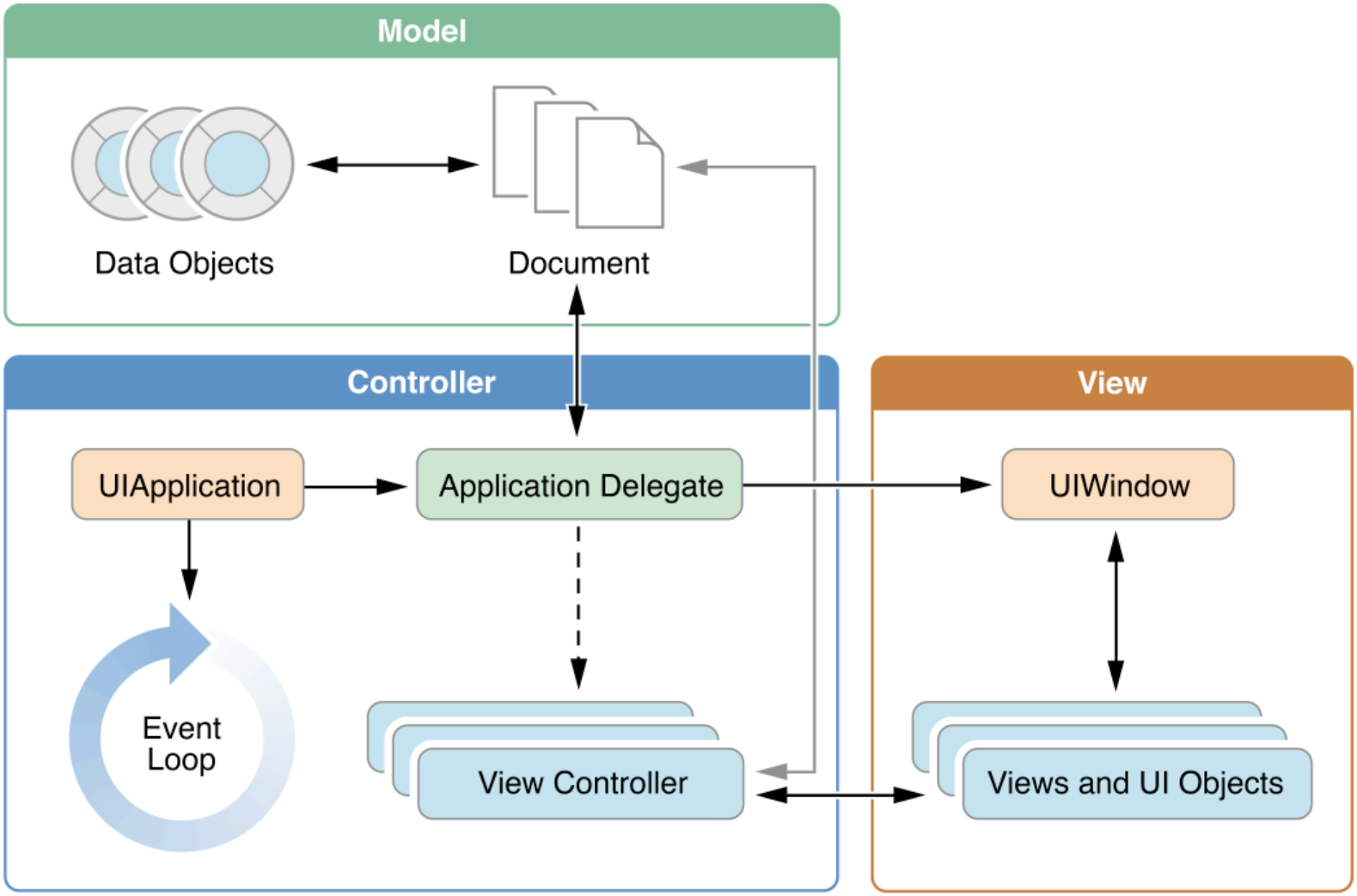
- iOS / Cocoa framework strongly suggest use of certain design patterns
  - **Model View Controller**
  - Delegation
  - Protocols
  - Notification
  - Target-Action
  - Key-Value Observation

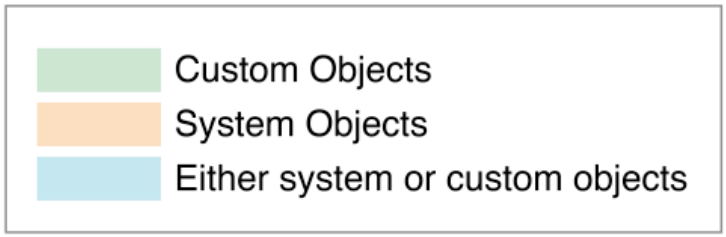
Model View Controller

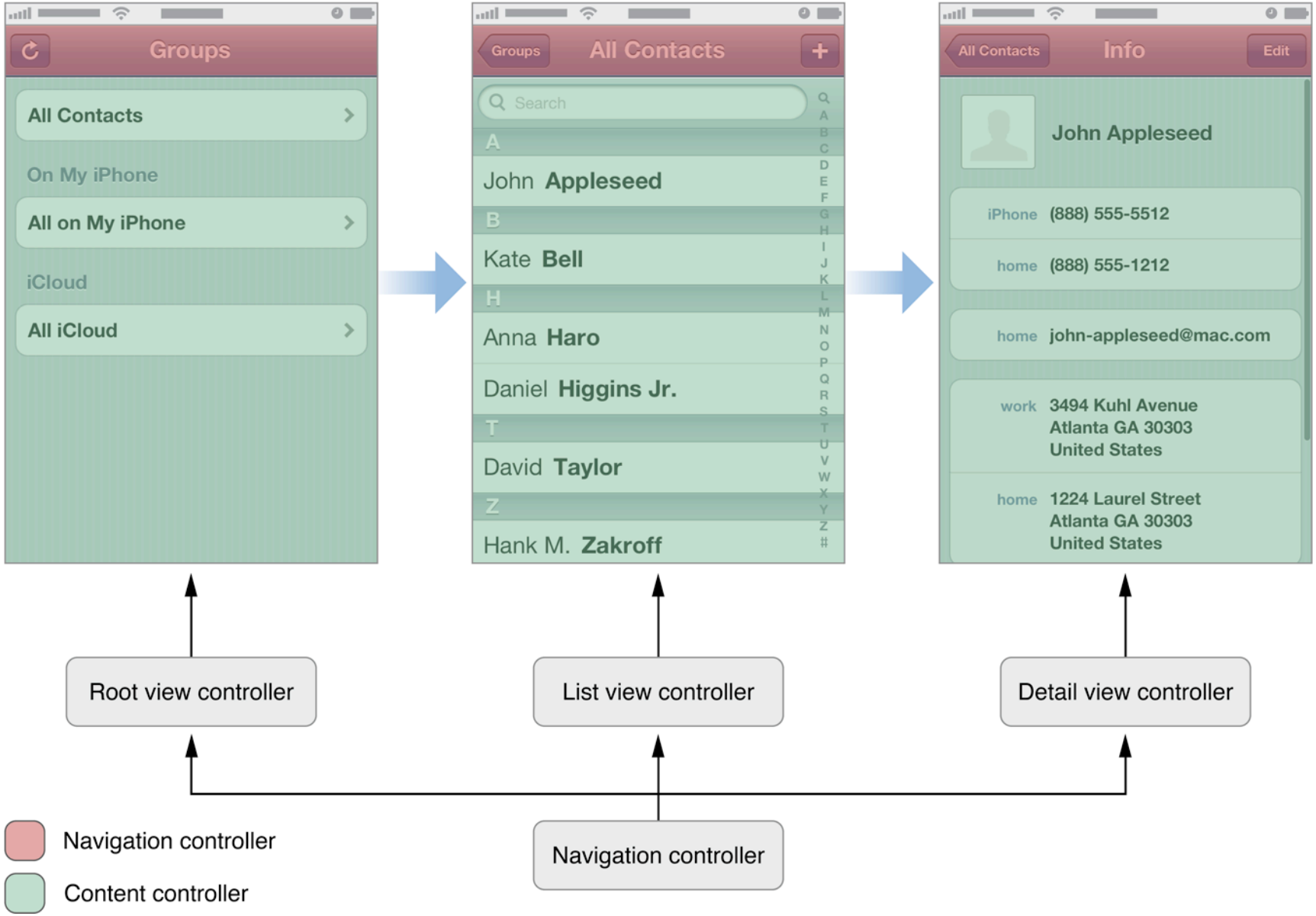
- Divide objects into three types
- Model
  - What the application **is**, but not how it is displayed
  - A contact in an address book
- Controller
  - **How** the model is presented to and manipulated by the user
  - Add / read / modify a contact
- View
  - Drawing things on the screen
  - Render a text view containing the contact
- MVC design pattern determines how these components should communicate
  - The model and view are typically decoupled


User action
Controller
Update
View
Update
Notify
Model


windowShouldClose:
No
windowDelegate

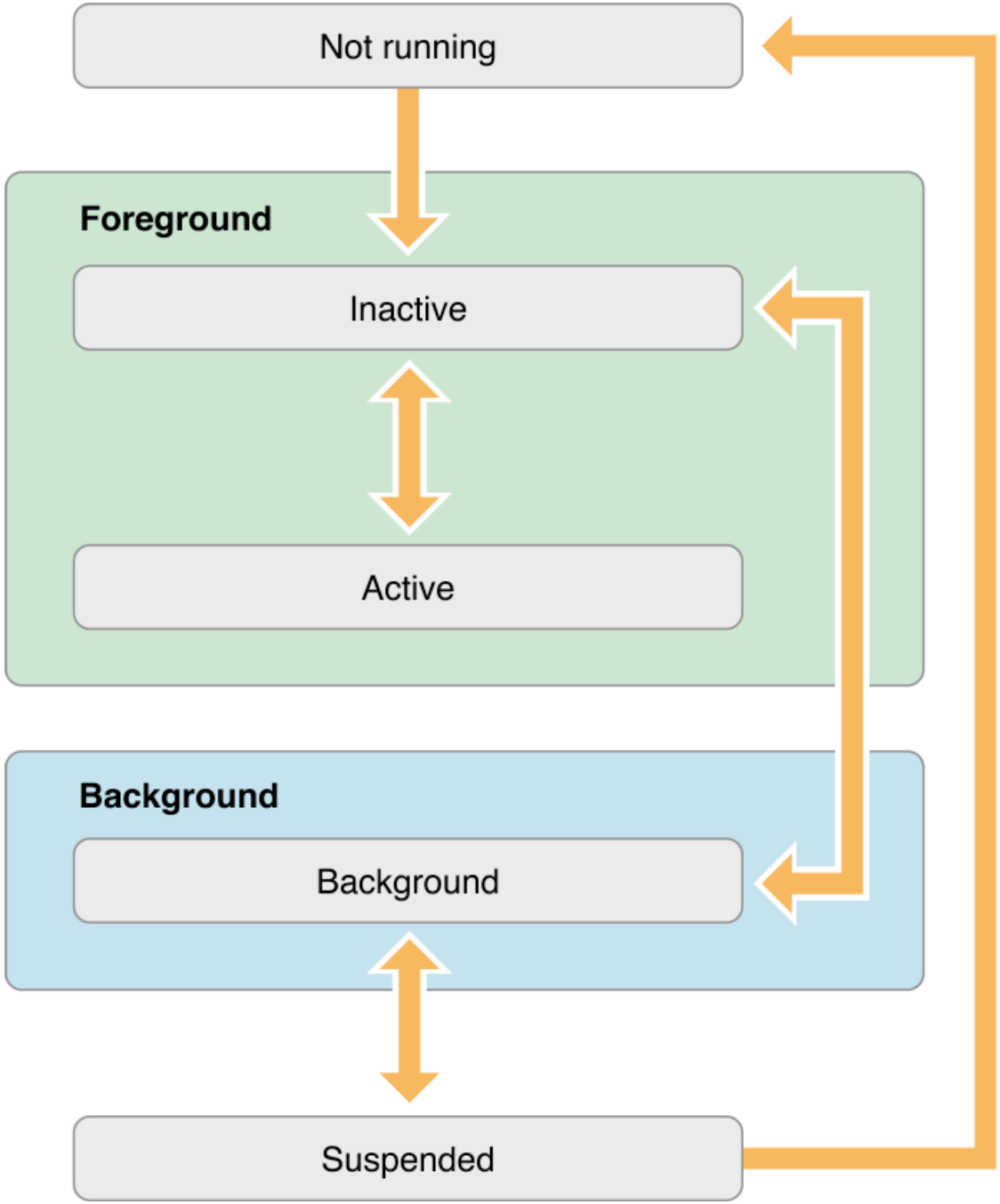

Model
Data Objects
Document
Controller
UIApplication
Application Delegate
Event Loop
View Controller
View
UIWindow
Views and UI Objects


Custom Objects
System Objects
Either system or custom objects


Groups
All Contacts
On My iPhone
All on My iPhone
iCloud
All iCloud
Groups
All Contacts
Search
A
John Appleseed
B
Kate Bell
H
Anna Haro
Daniel Higgins Jr.
T
David Taylor
Z
Hank M. Zakroff
All Contacts
Info
Edit
John Appleseed
iPhone (888) 555-5512
home (888) 555-1212
home john-appleseed@mac.com
work 3494 Kuhl Avenue
Atlanta GA 30303
United States
home 1224 Laurel Street
Atlanta GA 30303
United States
Root view controller
List view controller
Detail view controller
Navigation controller
Navigation controller
Content controller

iOS Lifecycle

- Analogous to Android lifecycle
  - Only one application in the foreground / visible at any one time
- A main loop processes events for the application
- An app can be a number of significant states
  - Active – foreground
  - Inactive – foreground but interrupted
    - By a phonecall, notification etc
  - Background – can remain in this state to perform long running tasks
    - Analogous to Services
  - Suspended
    - Main loop no longer running, potentially killed by the operating system
- iOS 3.2 and earlier
  - No support for suspended / background states
    - No long running tasks


Not running
Foreground
Inactive
Active
Background
Background
Suspended

App Store

- “We will reject Apps for any content or behavior that we believe is over the line. What line, you ask? Well, as a Supreme Court Justice once said, "I'll know it when I see it". And we think that you will also know it when you cross it.”
- Pre-moderation
  - Apple approves all applications in advance
  - Vs Android – publish then revoke
- A long list of guidelines as to what is appropriate
  - Correct use of interface components
  - Substantial content

App Store Restrictions

- **2.5** Apps that use non-public APIs will be rejected
- **2.8** Apps that install or launch other executable code will be rejected
- **2.10** iPhone Apps must also run on iPad without modification, at iPhone resolution, and at 2X iPhone 3GS resolution
- **2.16** Multitasking Apps may only use background services for their intended purposes: VoIP, audio playback, location, task completion, local notifications, etc.
- **2.17** Apps that browse the web must use the iOS WebKit framework and WebKit Javascript
- **13.2** Apps that rapidly drain the device's battery or generate excessive heat will be rejected

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?

Language

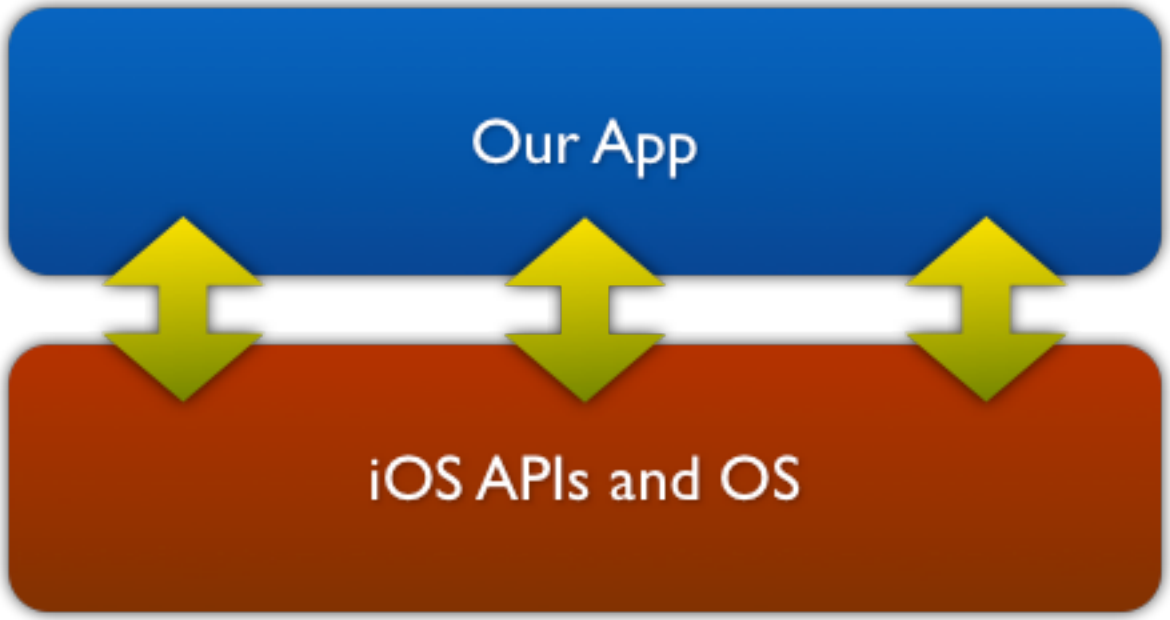
Platform	Language
Android	Java / C++
iOS	Objective C
Blackberry	Java, some Android support
Windows Phone	C#
webOS	C/C++ or HTML/ Javascript

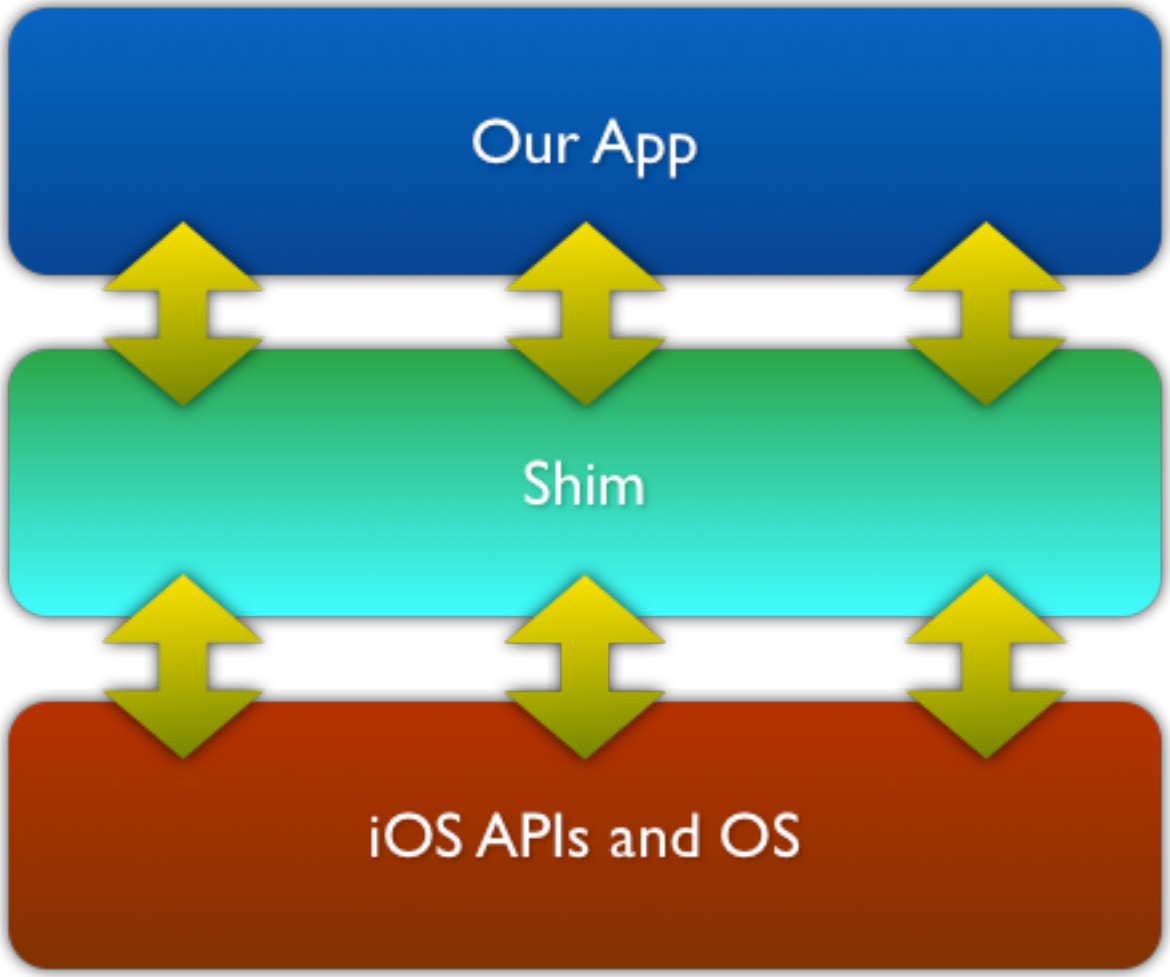
Language

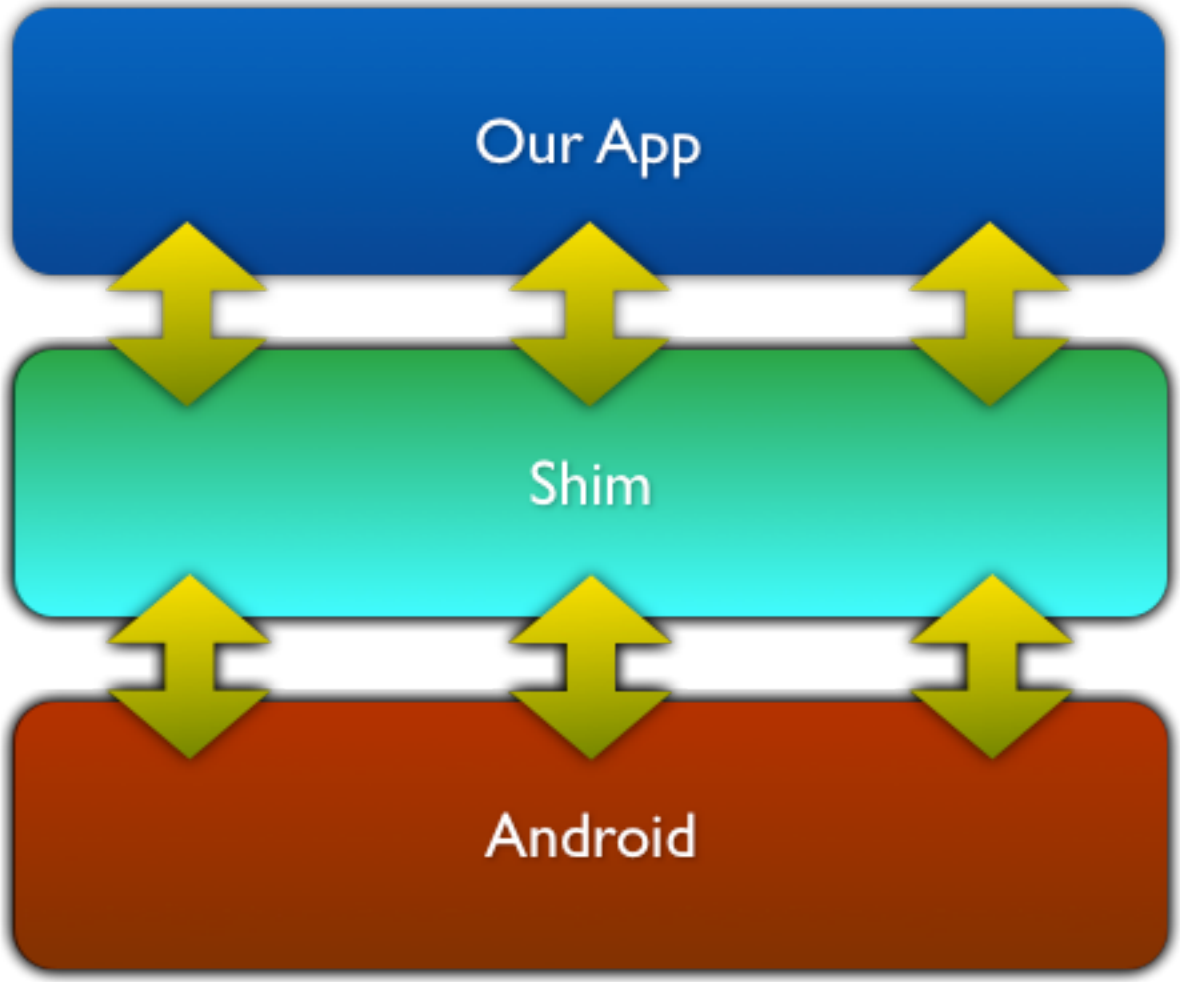
- Compile code for multiple platforms?
  - Can compile Java into native code (gcj)
    - Android supports native code libraries
  - C++ can be compiled to CLR (for WinPhone 7)
- This would work at a technical level
- The code would execute
  - But the app wouldn't run
  - Each platform has different 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

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://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html
- http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/RoadMapiOS/chapters/DesignPatterns.html
- https://developer.apple.com/appstore/resources/approval/guidelines.html
- http://phonegap.com/
- http://unity3d.com/
- http://appfurnace.com/