iPhone

Developer's overview and comparison of the iPhone and the Android platforms

Erlend Stav
SINTEF ICT
June 2009

mailto:erlend.stav@sintef.no

Overview

iPhone

android

- Vendors and licensing
- Application distribution
- Platform architecture
- Application architecture
- GUI and other components
- Inter-app communication
- Development tools

Devices, licensing , and openness

iPhone

- Apple is the only device vendor
- Device OS is not open source

- iPhone Developer Program membership has an annual fee ($99 to $299)
- Regular iPhone and iPod Touch retail devices are used for development, but each device must first be registered by the developer under the iPhone Developer Programme

android

- Open Handset Alliance
- Devices from multiple vendors, including HTC, Samsung, Sony Ericsson, Motorola, Acer, …
- Free and open source mobile platform, including OS, middleware and key applications
- Regular retail devices can be used to test and use applications
- Android Dev Phone 1, SIM-unlocked and hardware-unlocked development device, but is currently only ship to 18 countries worldwide.

Distributing your applications

iPhone

- iPhone Developer Programme members can sell or provide their applications for free through App Store
- Applications have to be approved by Apple before the become available in App Store
- With the Store Kit API developers can integrate purchase of new content and application extensions into their applications
- In-house distribution available for enterprises with 500+ employees

android

- Developers can publish their applications through Android Market
- To become a publisher of applications in the Android Market requires registration and a fee ($25)
- Currently only developers from 7 countries can sell applications
- Currently free applications can be distributed to 24 countries, and paid applications to 9 countries

Acquiring and deploying applications

iPhone

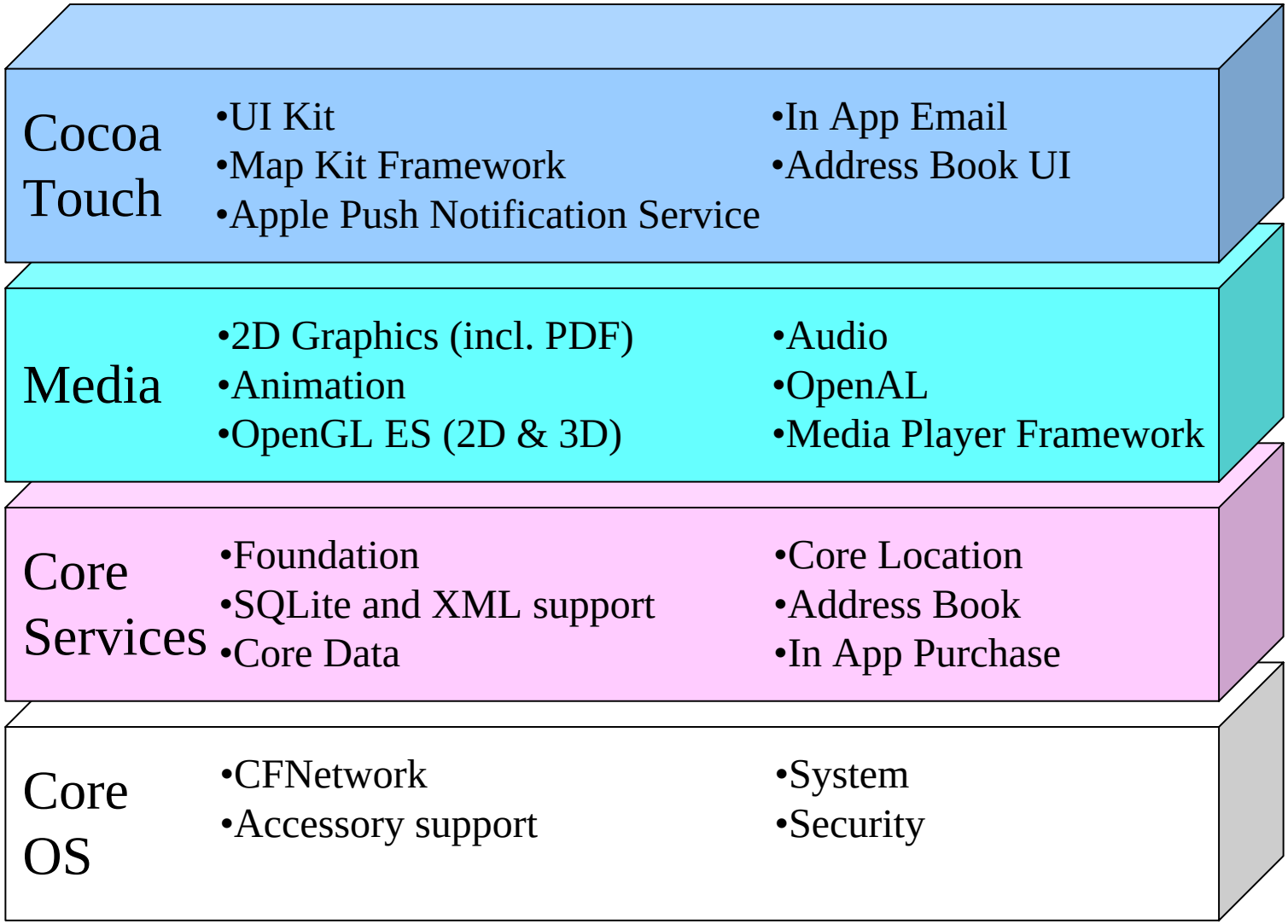
- Applications are acquired by users through Apple's official App Store (both free and paid)
- Applications are acquired directly from AppStore to the device, or using AppStore from iTunes on PC / Mac and then synchronizing the device
- Updates to installed applications are visible when visiting AppStore or using iTunes, and selected updates are deployed similar to new applications

android

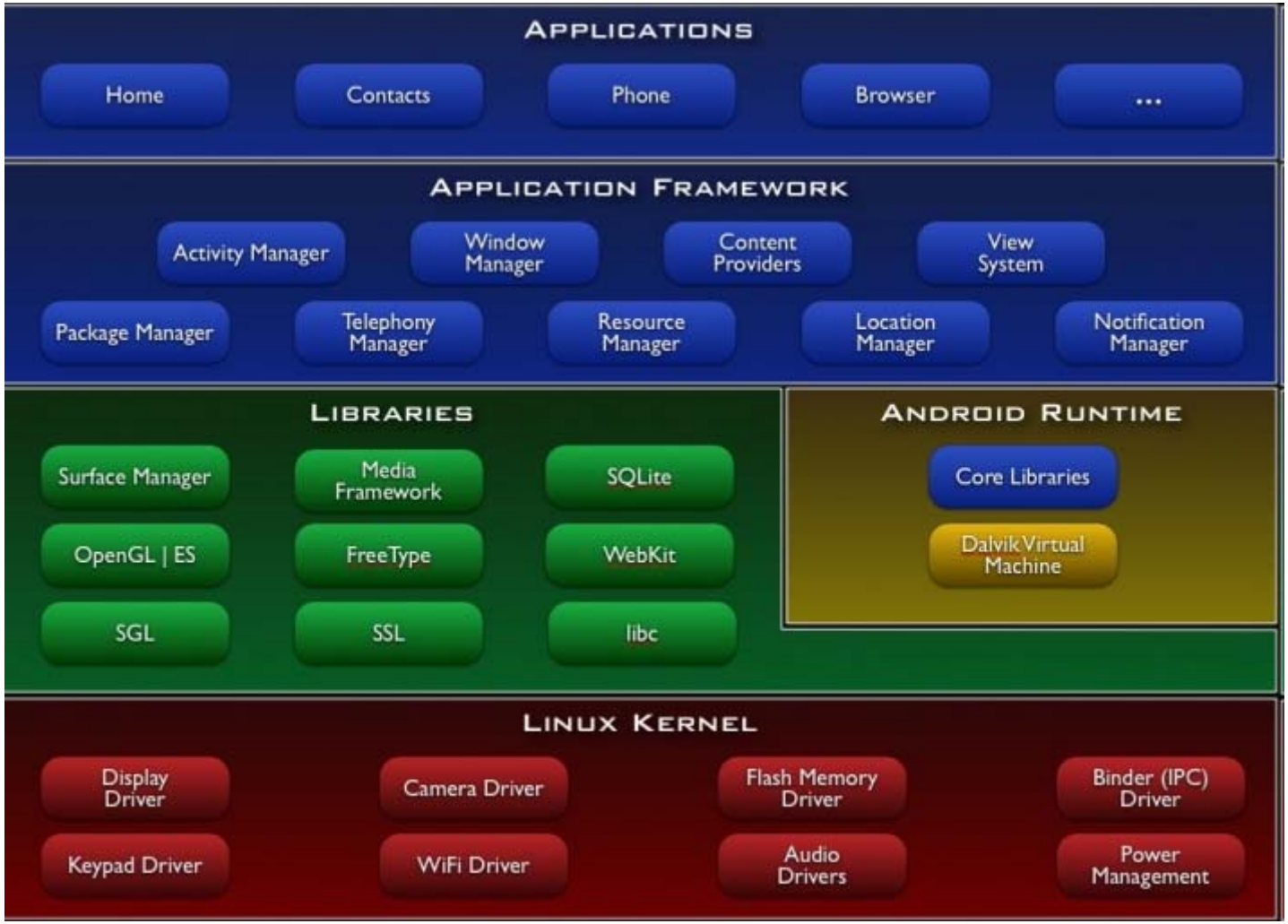
- Applications can be acquired by users through Android Market
- Applications are acquired directly from Android Market to the device. Currently Market is not viewable outside the device.
- All code for an application is bundled as an Android package for distribution and deployment
- Updates to installed applications are available when visiting My Downloads under Android Market.

iPhone

Architecture Overview



Architecture overview



Source: http://developer.android.com/guide/basics/what-is-android.html

Runtime model / processes for application

iPhone

- Single 3rd party application running
- Design rule: fast launch and fast shutdown, store state to allow users to continue where they left
- Application launch image – gives user impression of quick launch
- This Apple video includes an example of the basic interaction (time interval 1:12-1:22)

android

- Full multitasking
- Default: every application has separate Linux process
- Each process has its own Java VM
- To conserve resources it is possible to allow applications to run in the same process, sharing the same VM
- These YouTube videos shows how home screen and home button is used to switch between applications

Application Architecture

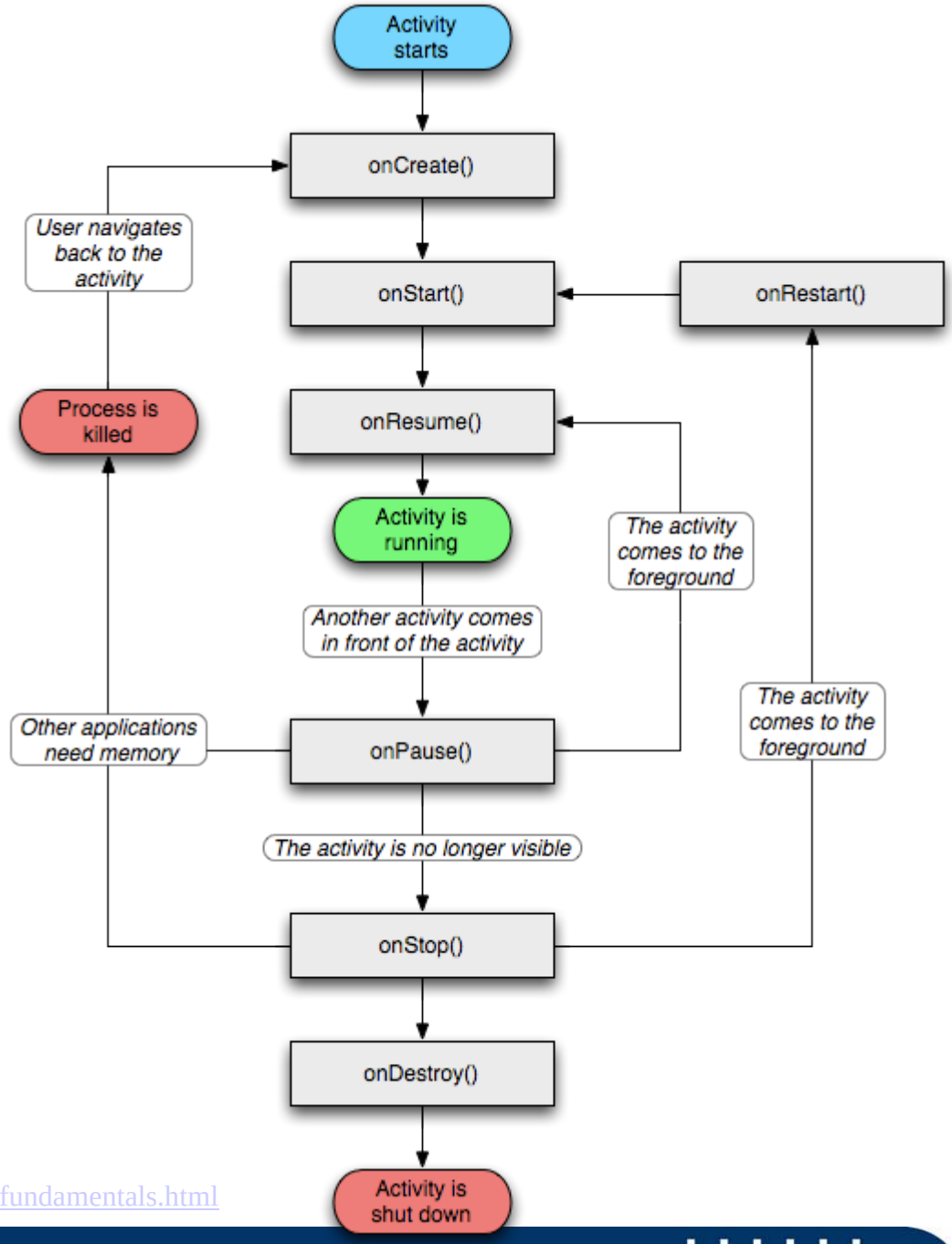
- No main() function for applications
- Four kinds of components, an application may have multiple of each.

- Activity:
  - presents user interface
- Service:
  - runs in background without user interface
- Broadcast receivers:
  - only receives and reacts to broadcast announcements
- Content providers:
  - makes set of application data available to other applications

Activities, tasks, intents

- A task is a group of activities arranged as a stack (seen from the user as an application)
- One activity can start other activities
- Activities from different applications can part of the same task (but belong to different processes)
- Activities, services and broadcast receivers are activated through asynchronous messages called *intents*
- An intent names an action and a data URI
- Intents supported by components are declared in the application's manifest
- See this YouTube video for illustration

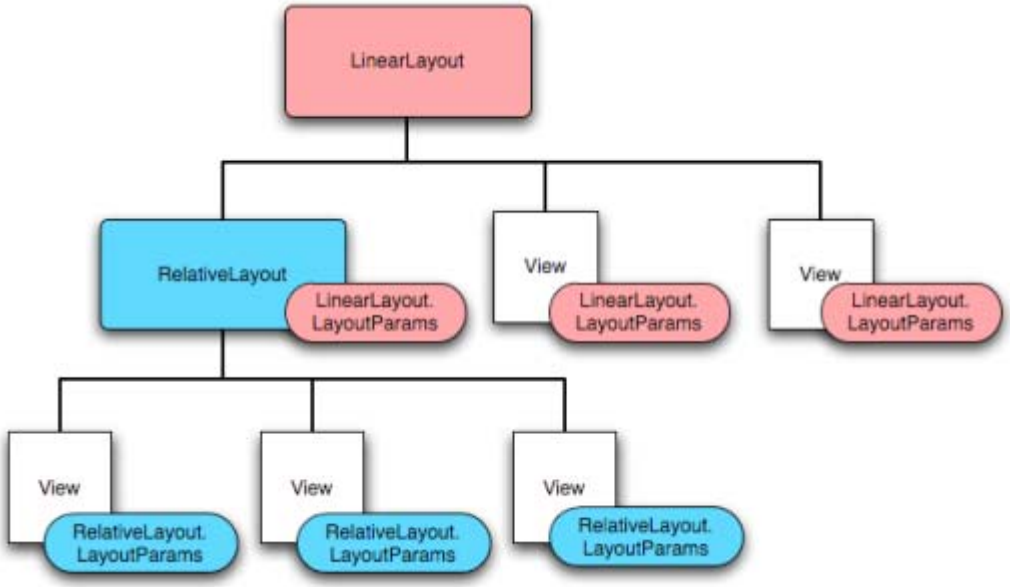
Lifecycle of activities



Source: http://developer.android.com/guide/topics/fundamentals.html

User interfaces

- Views are basic units of UI, view groups
- View groups, such as layouts, provide hierarchy and organization such as linear, tabular, etc.
- Rich set of widgets (concrete views such as buttons, text view, image view, etc)
- Activities have a content view



Source: http://developer.android.com/guide/topics/ui/declaring-layout.html

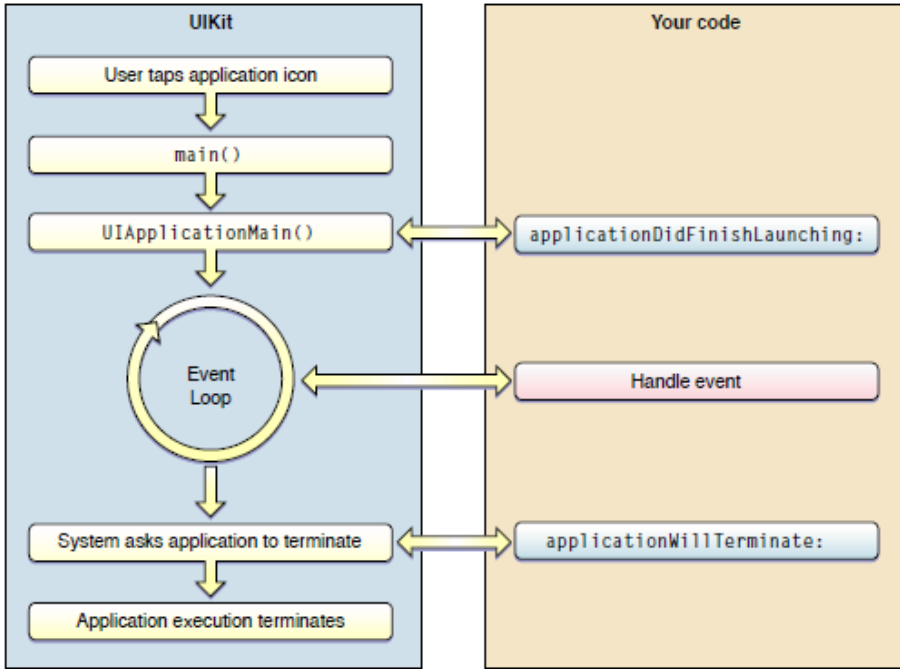
iPhone

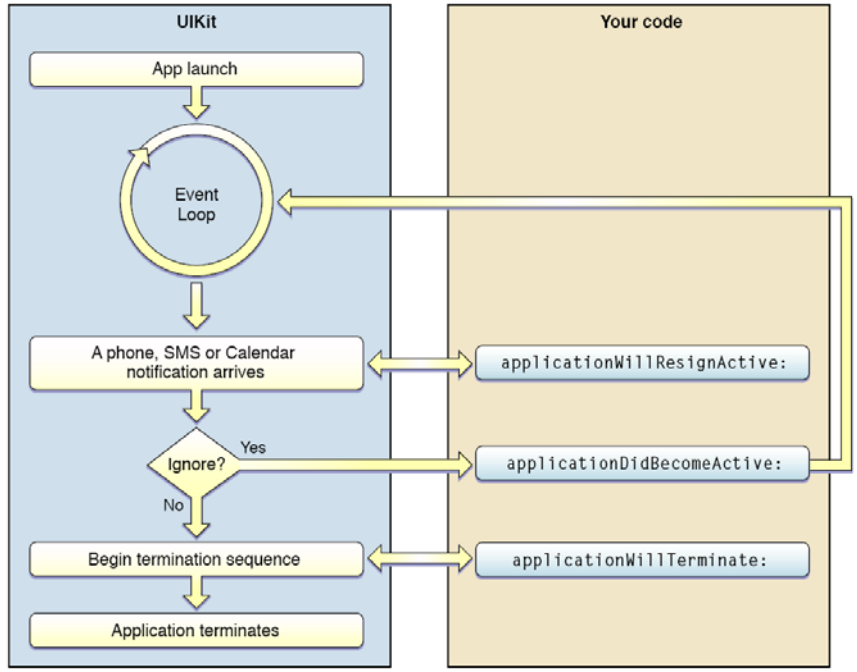
Application architecture

- Applications have a main() function (usually follows a template)
- Design patterns
  - Model-View-Controller
  - Delegation
  - Trigger-action
- Core data
  - Manages changes to model objects
  - Allows keeping just a subset of model objects in memory
  - Schema describe model objects

iPhone

Application lifecycle

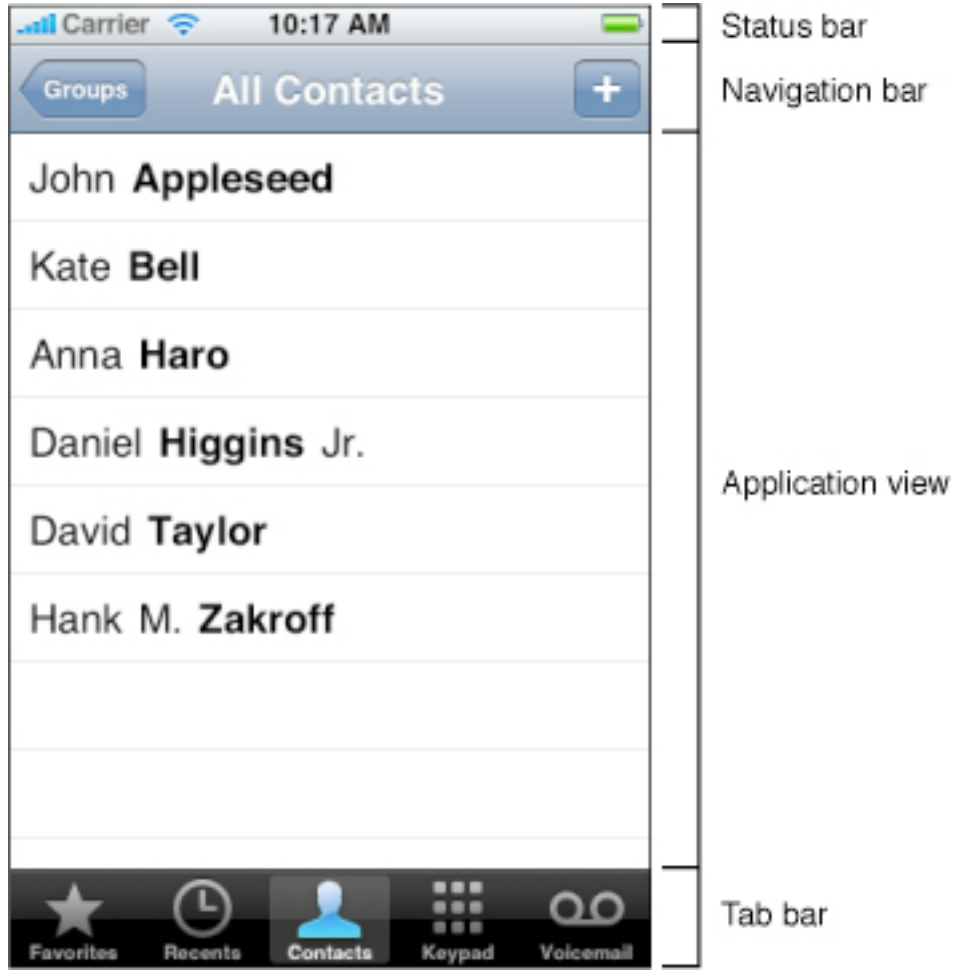




Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/index.html

iPhone GUI

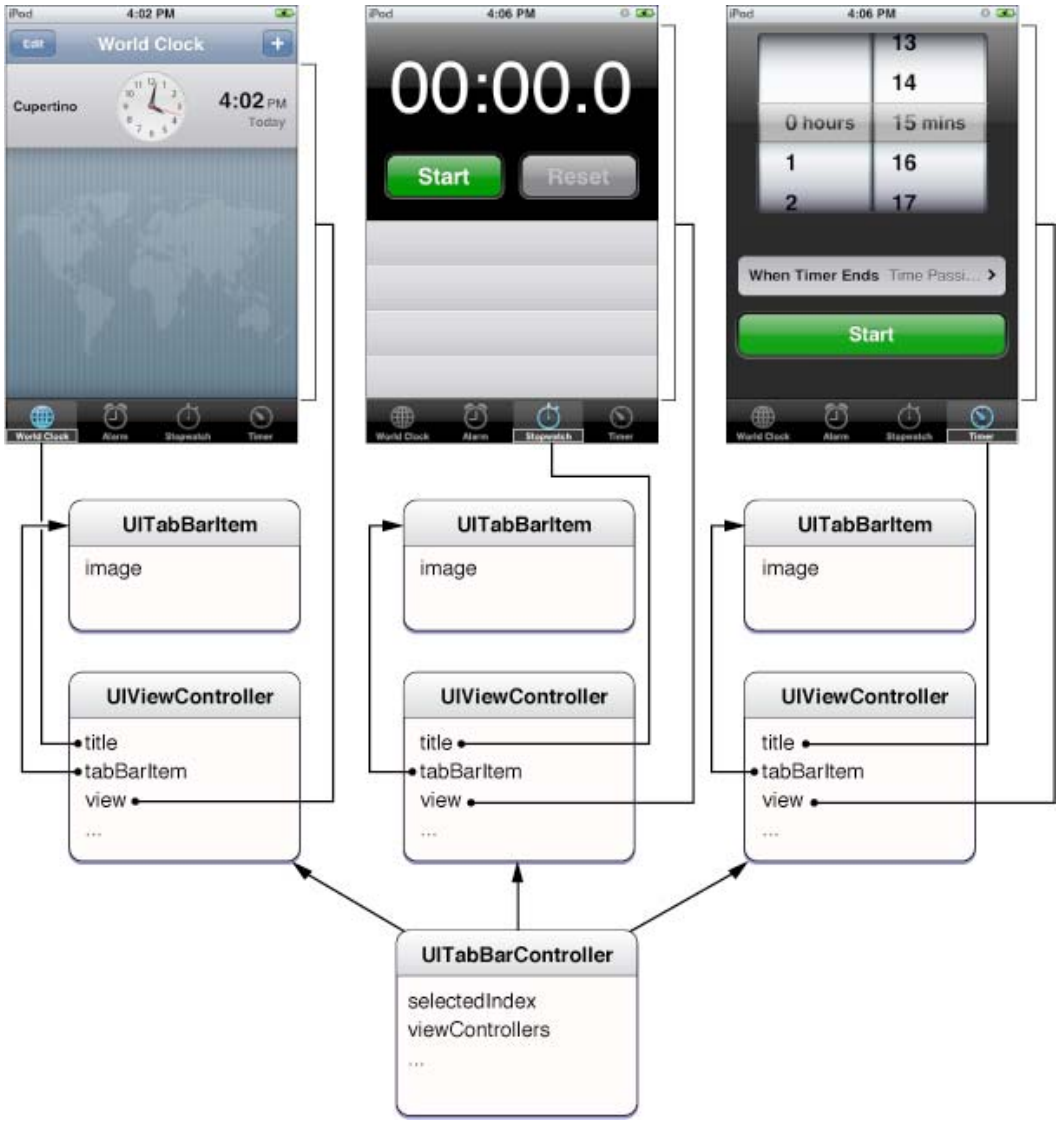
- Rich set of views designed for touch interface
- Multi-touch events
- View animation



Source: http://developer.apple.com/iphone/library/featuredarticles/ViewControllerPGforiPhoneOS/index.html

iPhone

Tab Bar GUI and controllers



Source: http://developer.apple.com/iphone/library/featuredarticles/ViewControllerPGforiPhoneOS/index.html

iPhone Navigation GUI

Source: http://developer.apple.com/iphone/library/featuredarticles/ViewControllerPGforiPhoneOS/index.html

High level components

iPhone

- Map component (based on Google Maps), can add annotations
- HTML-viewer (including HTML, CSS and JavaScript content)
- Contact / mail
- Music
- Photo library
- Camera – standard UI for taking pictures

android

- Map View (as add-on), supports overlays
- Web View
- Video View
- Media Controller

- In addition activities from other applications are integrated through intents

Inter-application communication

iPhone

- Launch other application using an URL to communicate with it
- Application can register custom URL schemes
- Copy / paste between applications

android

- Launch other activities using intents with action and URI
- Components register intent filters they react to
- Copy / paste between applications
- Content Providers gives access to data from other applications
- Components can bind to services also of other applications, and perform remote procedure calls to interfaces declared using an IDL

Examples of URLs and intent actions

iPhone

- mailto:frank@example.com
- tel:1-408-555-5555
- sms:1-408-555-1212
- http://maps.google.com/maps?daddr=San+Francisco,+CA&saddr=cupertino (Driving directions between San Francisco and Cupertino)

android

- ACTION_SEND
- CALL tel:2125551212
- ACTION_EDIT content://contacts/1
- VIEW http://web_address
- VIEW geo:0,0?q=my+street+address
- ACTION_GET_CONTENT
- ACTION_REBOOT

Device access

iPhone

- Accelerometer
  - Shake events
- Positioning (GPS)
- Camera
- External accessories / BT
  - Made for iPod licensing program

android

- Sensors including accelerometer, proximity, light, temperature, …
- Positioning, location provides, GPS status
- Camera including settings, preview, snap picture
- Currently no Bluetooth API
- API for accessing current state and listening to state changes for in telephone (network type, cell ids, calls, data traffic…)

File management

iPhone

- Sandbox
- Each application has its own home directory
- Backup: automatic when synched using iTunes, all of home directory except cache and tmp directory

android

- Default: each application has a unique Linux user id, and application files are only visible to that user id

Programming languages

iPhone

- C and Objective C
- Compiled to native code
- No garbage collection, but uses reference counters and reference pools.
- No (official) Java version

android

- All applications are developed in Andoid's Java
- Dalvik VM specific to Android
- Android-specific bytecode optimized for minimal memory footprint
- Libraries do not match any of the standard Java profiles, but covers most of J2ME CDC

iPhone

Objective C code Example

Header file and implementation of one method

```
#import <UIKit/UIKit.h>

@interface MyViewController : UIViewController <UITextFieldDelegate> {
    UITextField *textField;
    UILabel *label;
    NSString *string;
}
@property (nonatomic, retain) IBOutlet UITextField *textField;
@property (nonatomic, retain) IBOutlet UILabel *label;
@property (nonatomic, copy) NSString *string;
- (IBAction)changeGreeting:(id)sender;
@end


- (IBAction)changeGreeting:(id)sender {
    self.string = textField.text;
    NSString *nameString = string;
    if ([nameString length] == 0) {
        nameString = @"World";
    }
    NSString *greeting = [[NSString alloc]
        initWithFormat:@"Hello, %@!", nameString];
    label.text = greeting;
    [greeting release];
}
```

Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhone101/index.html

Development tools and profiling

iPhone

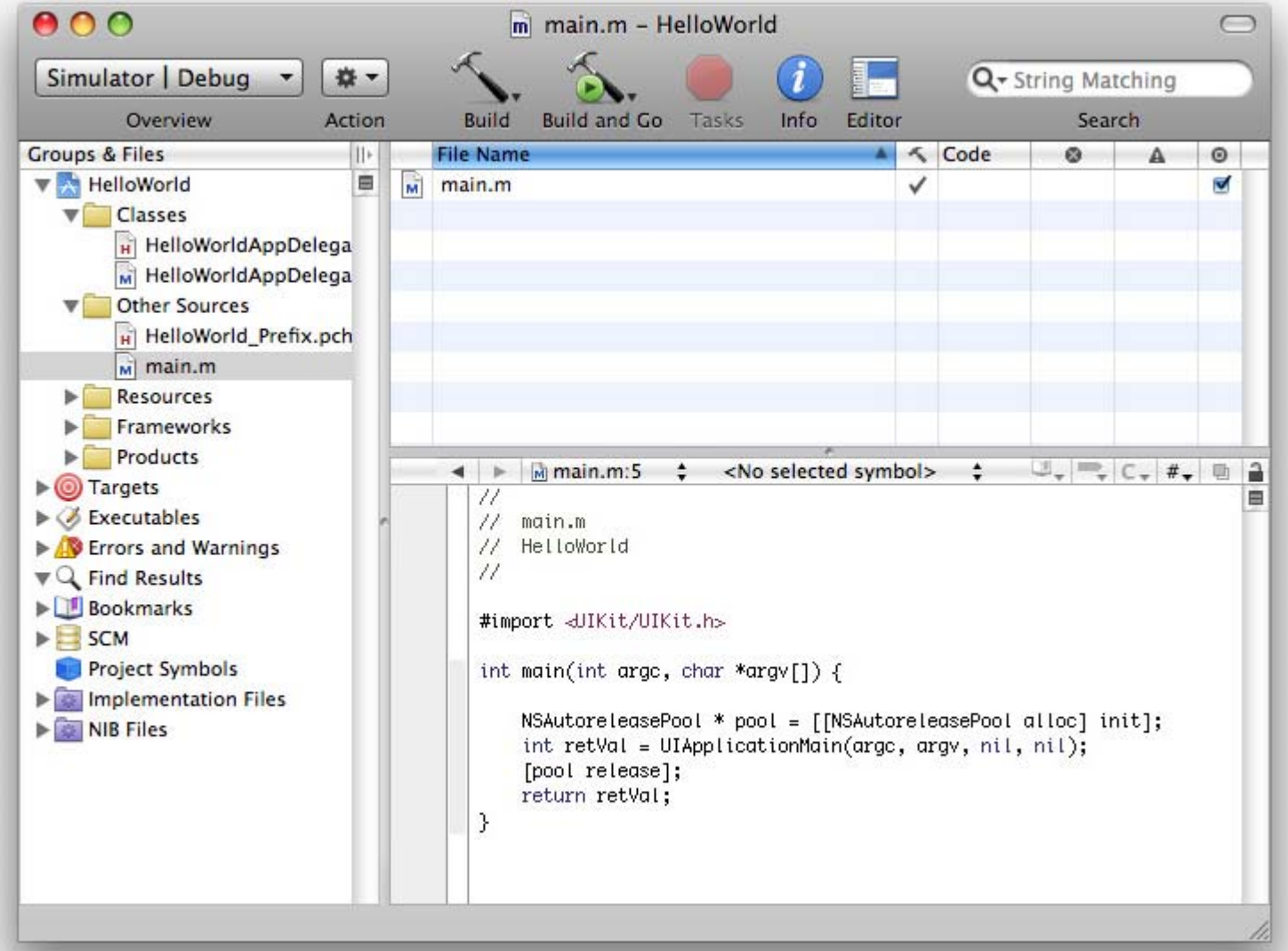
- Development requires a Mac
- Free tools include:
  - XCode IDE
  - iPhone emulator
  - Instruments (for profiling)
  - Interface Builder

ANDROID

- Free SDK available for Windows, Mac, and Linux development
- Any Java development environment can be used
- Free tools include:
  - Android Development Tools plugin for Eclipse
  - Android Emulator
  - Memory and performance profiling
  - Debugging tools
  - UI stress-test tools

iPhone

XCode IDE

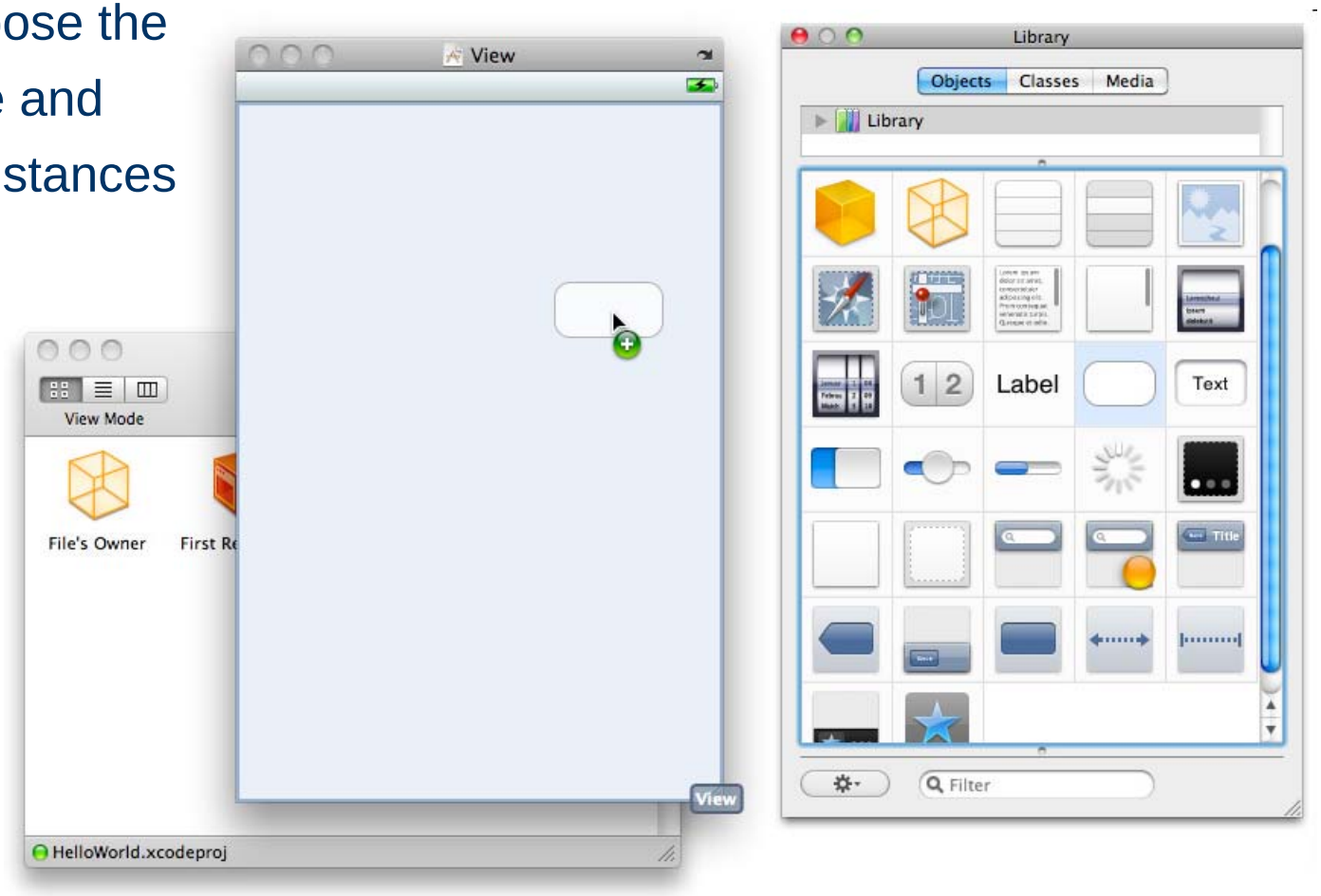


Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhone101/index.html

iPhone

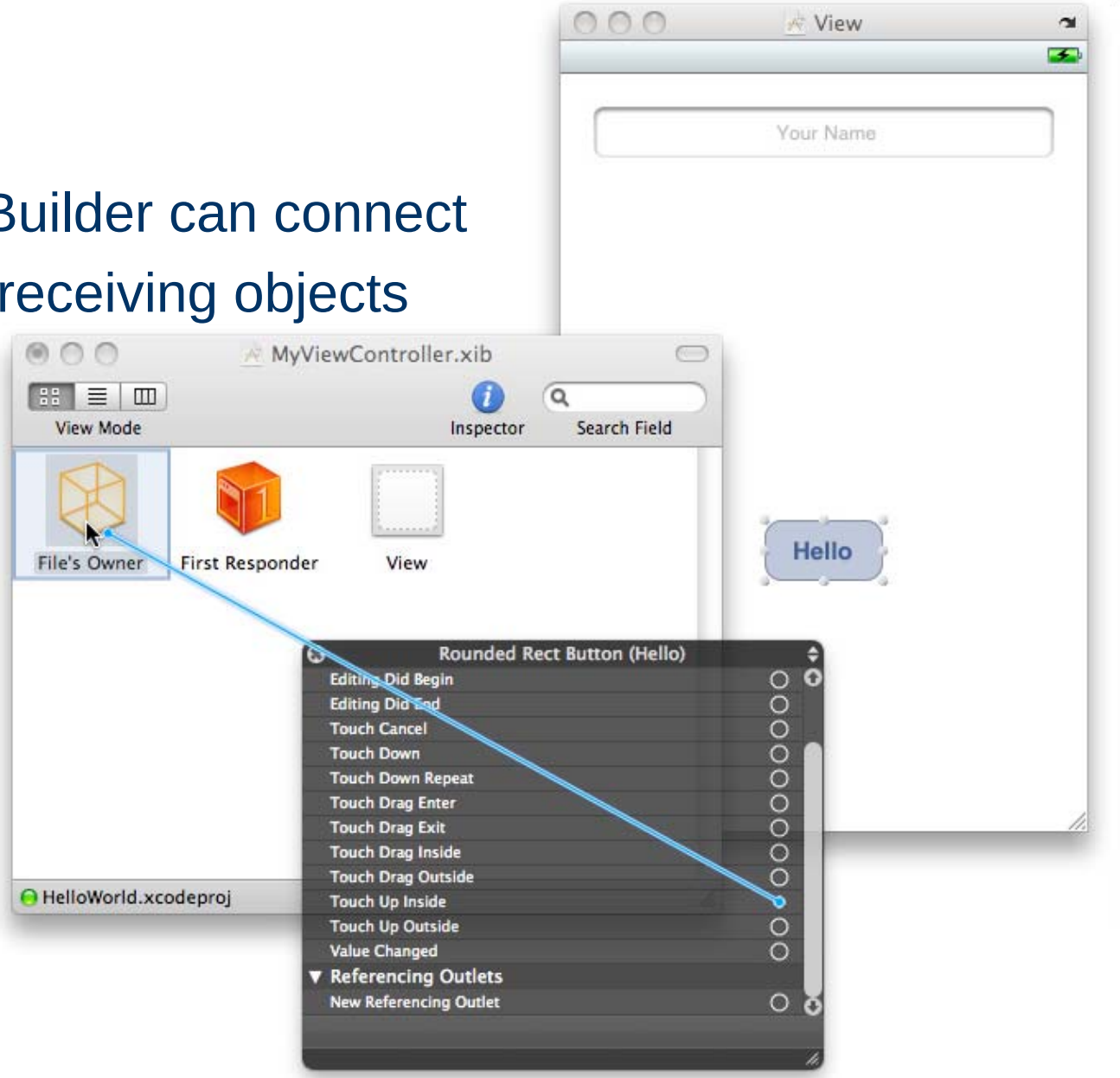
Interface Builder

Used to compose the User interface and main object instances of the iPhone application



Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhone101/index.html

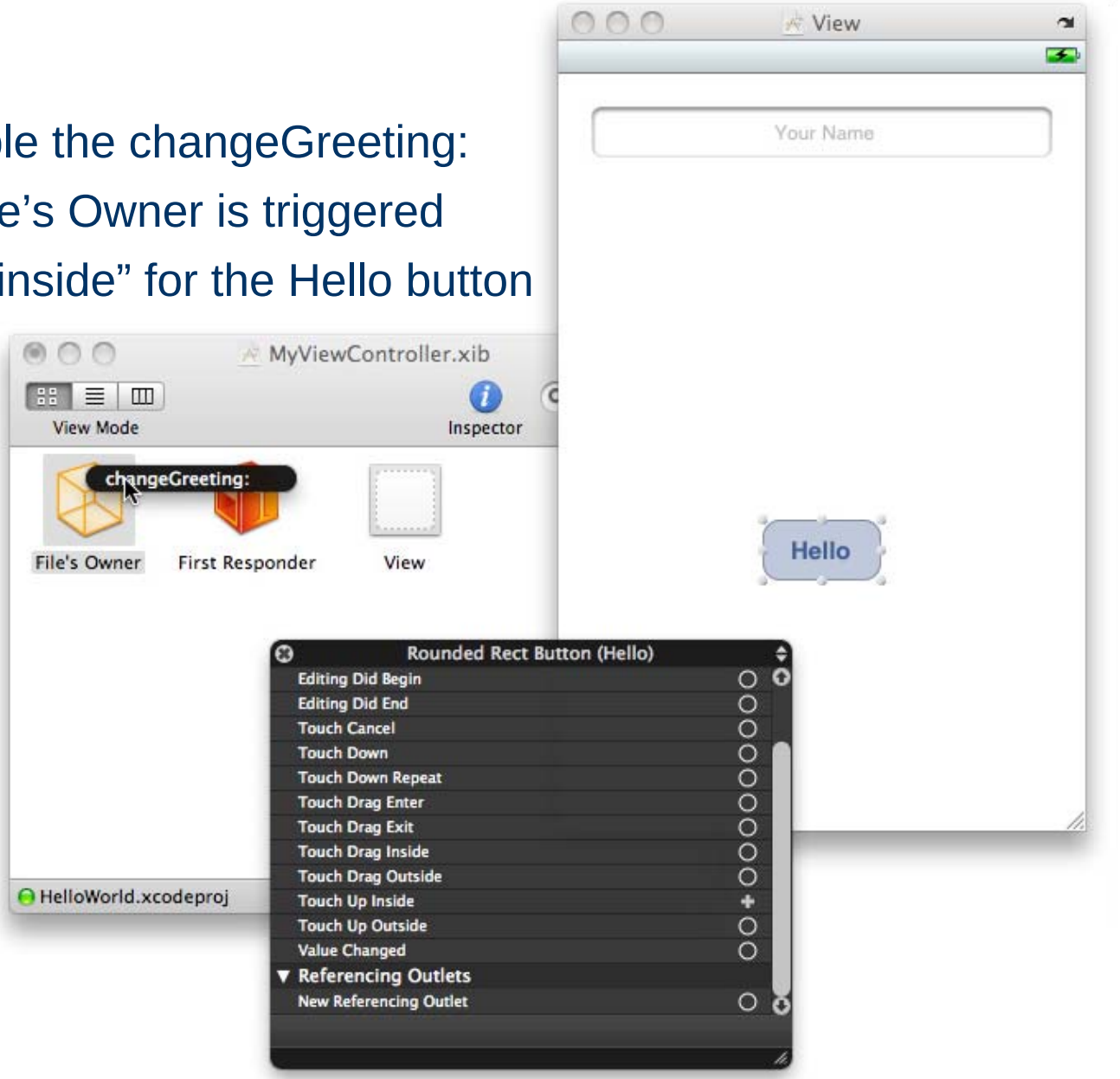
Interface Builder can connect Events to receiving objects



Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhone101/index.html

iPhone

In this example the changeGreeting:
method of File's Owner is triggered
by "touch up inside" for the Hello button



Source: http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhone101/index.html
http://developer.android.com/guide/topics/ui/declaring-layout.html

References

iPhone

- Apple's iPhone Dev Center
  - Site: http://developer.apple.com/iphone/
  - Illustrations from:
    - iPhone Application Programming Guide
    - View Controller Programming Guide for iPhone OS
    - Your First iPhone Application
- Android's developer guide
  - Site: http://developer.android.com/guide/
  - Illustrations from:
    - What is Android?
    - Application Fundamentals
    - User Interface – Declaring Layouts