

# WPAM

## W4 - Apple iPhone OS

**Piotr Gawrysiak**

*pgawrysiak@supermedia.pl*

**Politechnika Warszawska  
Instytut Informatyki  
Zakład Systemów Informatycznych**



2011

# Reinventing the phone?

Apple iPhone został po raz pierwszy zaprezentowany publicznie podczas MacWorld Expo w 2007 roku - *czyli zaledwie (?) trzy i pół roku temu.*

Typowe „flagowe” modele telefonów i urządzeń Nokia (koniec 2006) – Nokia E61, N770 (Internet Tablet)



# Steve Jobs i efekt RDF

Oczywiście pamiętać należy iż Steve Jobs otaczany jest przez pole zakłóceń rzeczywistości („*reality distortion field*”)

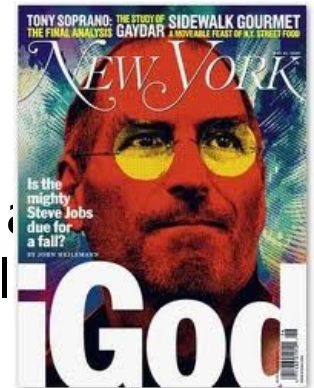
...

Wiele z technologii wymienionych przez Jobsa w keynote jako „invented by Apple” istniało

- Ekran dotykowy multitouch
- Gestures (pinch to zoom itd.)
- Wykorzystanie akcelerometru w urządzeniu mobilnym
- HTML email client
- ...



Pierwsze wersje iPhone cechowały także zastanawiające braki funkcjonalne (np. copy-paste, multitasking, brak aplikacji itp.).



Ale to akurat okazało się nie mieć żadnego znaczenia 😊

# Rodzina iOS

System operacyjny iOS (poprzednio iPhone OS) oraz koncepcja dotykowego interfejsu użytkownika to nie tylko iPhone!

iPod Touch  $\approx$  iPhone bez modułu GSM

iPad, iPad2  $\approx$  iPod Touch „nadmuchany” do rozmiarów kartki A4 (z ew. modułem GSM)

Różnice (pomiędzy iPod Touch i iPhone, szczególnie w przypadku nowszych generacji, są dla twórcy aplikacji niewielkie) – pierwsza wersje iPod Touch nie posiadała głośnika i akceleracji



# iPhone - ewolucja platformy sprzętowej

Model	iPhone	iPhone 3G	iPhone 3GS	iPhone 4
<b>Initial operating system</b>	iPhone OS 1.0	iPhone OS 2.0	iPhone OS 3.0	iOS 4.0
<b>Highest Supported operating system</b>	iPhone OS 3.1.3	iOS 4.1		
<b>Display</b>	3.5 in (89 mm), 3:2 aspect ratio, scratch-resistant <sup>[21]</sup> glossy glass covered screen, 262,144-color LCD, 480 × 320 px (HVGA) at 163 ppi		In addition to previous, features a fingerprint-resistant oleophobic coating <sup>[75]</sup>	3.5 in (89 mm), 3:2 aspect ratio, aluminosilicate glass covered IPS LCD screen, 960 × 640 px at 326 ppi, 800:1 contrast ratio
<b>Storage</b>	4, 8 and 16 GB	8 and 16 GB	8, 16 and 32 GB	16 and 32 GB
<b>Processor</b>	620 MHz (underclocked to 412 MHz) Samsung 32-bit RISC ARM 1176JZ(F)-S v1.0 <sup>[76][77]</sup>		833 MHz (underclocked to 600 MHz) ARM Cortex-A8 <sup>[78][79]</sup> Samsung S5PC100 <sup>[78][80]</sup>	ARM Cortex-A8 Apple A4 <sup>[81]</sup>
<b>Graphics</b>	PowerVR MBX Lite 3D GPU <sup>[82]</sup>		PowerVR SGX535 GPU <sup>[78]</sup>	PowerVR SGX535 GPU <sup>[83]</sup>
<b>Memory</b>	128 MB DRAM <sup>[84]</sup>		256 MB DRAM <sup>[78][79]</sup>	512 MB DRAM <sup>[85]</sup>
<b>Connectivity</b>	Wi-Fi (802.11b/g), USB 2.0/Dock connector, Quad band GSM/GPRS/EDGE (850, 900, 1800, 1900 MHz) Bluetooth 2.0 + EDR Cambridge Bluecore4 <sup>[86]</sup>	In addition to previous: Assisted GPS, Tri-band UMTS/HSDPA (850, 1900, 2100 MHz) <sup>[46]</sup> , Includes earphones with mic	In addition to previous: 7.2 Mbit/s HSDPA, Voice Control, Digital compass, Nike+, Bluetooth 2.1 + EDR Broadcom 4325 <sup>[87]</sup> , Includes earphones with remote and mic	In addition to previous: Penta-band UMTS/HSDPA (800, 850, 900, 1900, 2100 MHz) <sup>[35][88]</sup> , 5.76 Mbit/s HSUPA, 2.4 GHz 802.11n, 3-axis gyroscope, Dual-mic noise suppression, microSIM

# iPhone - ewolucja platformy sprzetowej

Camera	2.0 MP with geotagging		3.0 MP with <a href="#">VGA video</a> at 30 fps, <a href="#">geotagging</a> , <a href="#">tap to focus</a> , and <a href="#">focus</a> , <a href="#">white balance</a> , <a href="#">macro focus</a> & <a href="#">exposure</a>	Rear 5.0 MP backside <a href="#">illuminated CMOS image sensor</a> with <a href="#">720p HD video</a> at 30 fps and <a href="#">LED flash</a>
	N/A			Front 0.3 MP ( <a href="#">VGA</a> ) with <a href="#">geotagging</a> , <a href="#">tap to focus</a> , and <a href="#">480p SD video</a> at 30 fps
Audio codec	<a href="#">Wolfson Microelectronics WM8758BG</a> <sup>[18]</sup>	<a href="#">Wolfson Microelectronics WM6180C</a> <sup>[19]</sup>	<a href="#">Cirrus Logic CS42L61</a> <sup>[20][21]</sup>	
Materials	<a href="#">Aluminum</a> , glass and plastic	Glass and plastic; black or white (white not available for 8 GB models)		<a href="#">Aluminosilicate glass</a> and <a href="#">stainless steel</a> ; black (white announced, but as of October 2010 not available)
Power	Built-in non removable rechargeable <a href="#">lithium-ion polymer battery</a> <sup>[22][23][24]</sup>			
	3.7 V 1400 mA·h <sup>[citation needed]</sup>	3.7 V 1150 mA·h <sup>[23]</sup>	3.7 V 1219 mA·h <sup>[25]</sup>	3.7 V 1420 mA·h <sup>[26]</sup>
Rated battery life (hours)	<i>audio: 24</i> <i>video: 7</i> <i>Talk over 2G: 8</i> <i>Browsing internet: 6</i> <i>Standby: 250</i>	<i>audio: 24</i> <i>video: 7</i> <i>Talk over 3G: 5</i> <i>Browsing over 3G: 5</i> <i>Browsing over Wi-Fi: 9</i> <i>Standby: 300</i>	<i>audio: 30</i> <i>video: 10</i> <i>Talk over 3G: 5</i> <i>Browsing over 3G: 5</i> <i>Browsing over Wi-Fi: 9</i> <i>Standby: 300</i>	<i>audio: 40</i> <i>video: 10</i> <i>Talk over 3G: 7</i> <i>Browsing over 3G: 6</i> <i>Browsing over Wi-Fi: 10</i> <i>Standby: 300</i> <sup>[27]</sup>
Dimensions	115 × 61 × 11.6 mm (4.5 × 2.4 × 0.46 in)	115.5 × 62.1 × 12.3 mm (4.5 × 2.4 × 0.48 in)		115.2 × 58.6 × 9.3 mm (4.5 × 2.31 × 0.37 in)
Weight	135 g (4.8 oz)	133 g (4.7 oz)	135 g (4.8 oz)	137 g (4.8 oz)
Released	4 and 8 GB: June 29, 2007 16 GB: February 5, 2008	July 11, 2008	16 and 32 GB: June 19, 2009 Black 8 GB: June 24, 2010	June 24, 2010
Discontinued	4 GB: September 5, 2007 8 and 16 GB: July 11, 2008	16 GB: June 8, 2009 Black 8 GB: June 4, 2010	16 and 32 GB: June 24, 2010 Black 8 GB: In production	In production

# iPhone z punktu widzenia użytkownika

iPhone jaki jest każdy widzi..., niemniej jednak warto podkreślić kilka cech użytkowych:

Urządzenie jest elementem większej infrastruktury opartej o produkty Apple, w tym w szczególności program iTunes (który m.in. niezbędny jest do uruchomienia telefonu)

iTunes jest właściwie jedynym mechanizmem (pomijając *jailbreaking* itp.) pozwalającym na transfer danych zawierających treści cyfrowe (*content*) - DRM

Interfejs użytkownika oparty jest o koncepcję indywidualnych aplikacji, które w niewielkim jedynie stopniu współpracują ze sobą:

- do wersji 4.0 systemu jedynie niektóre wbudowane aplikacje (np. iPod) mogły działać współbieżnie, od wersji 4.0 pozostałe aplikacje mogą „udawać” iż działają współbieżnie
- system plików urządzenia jest niedostępny dla użytkownika, każda aplikacja ma dostęp jedynie do własnego katalogu danych w ramach *sandbox*
- wymiana danych pomiędzy aplikacjami jest zadaniem nietrywialnym

4.1



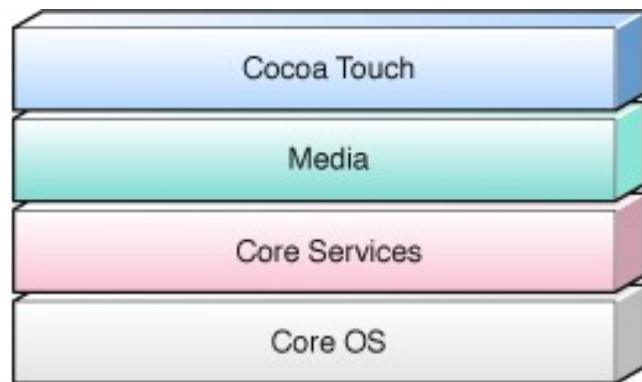
2.0



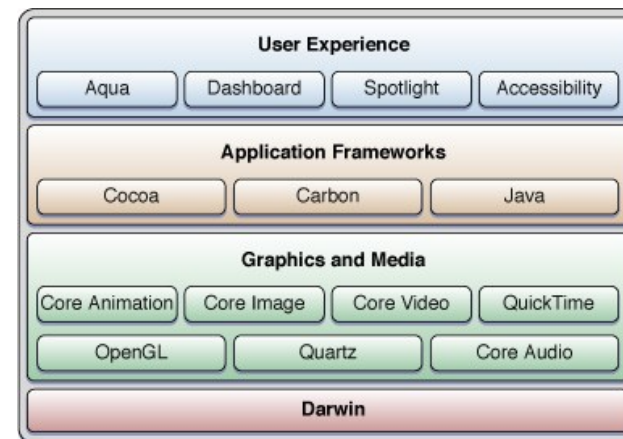
Aplikacje firm trzecich mogą być instalowane tylko

# iOS - z punktu widzenia twórcy aplikacji

iOS został zaprojektowany przez twórców systemu Mac OS X – dla twórców aplikacji wykorzystujących standardowe biblioteki Mac OS (Objective-C, Cocoa) jest to środowisko „naturalne”



iOS



Mac OS X

Ponieważ iOS tworzono od podstaw, można było zrezygnować z tych elementów, które w przypadku systemu Mac OS X służą zapewnieniu zgodności z istniejącymi już aplikacjami (np. Carbon, Java).



# CoreOS

Jądro systemu operacyjnego i podstawowa warstwa HAL

Podstawowe usługi systemu operacyjnego (nie związane z interfejsem użytkownika):

- **System** (LibSystem)

*Wątki, obsługa sieci, system plików, locale, zarządzanie pamięcią itp.*

- **Security Framework** (Security.framework)

*Certyfikaty, hasła, keychain, szyfrowanie danych itp.*

- **External Accessory Framework**

(ExternalAccessory.framework, iOS 3)

*Komunikacja z urządzeniami zewnętrznymi (w praktyce DDK) – poprzez Bluetooth lub złącze dock*

# Core Services

Większość „użytkowych” usług systemu operacyjnego

Dużo różnorodnych funkcji – przykłady:

- **Address Book** (obsługa bazy kontaktów)
- **CFNetwork** (obsługa sieci - HTTP, HTTPS, FTP, SLL/TLS)
- **Core Data** (wsparcie dla wzorca projektowego MVC)
- **Core Foundation** (zarządzanie danymi - strings, streams, arrays, sets itp.)
- **Core Location** (geolokalizacja)
- **Core Media** (niskopoziomowa obsługa dźwięku i wideo)
- **Core Telephony** (dostęp do danych z modułu GSM)
- **Event Kit** (dostęp do danych kalendarza)
- **In App Purchase** (transakcje poprzez konto iTunes użytkownika)
- **SQLite** (relacyjna baza danych)
- **System Configuration** („reachability interfaces” - ocena stanu połączeń sieciowych i dostępności usług)

# Media

Usługi wspierające odtwarzanie wideo, tworzenie grafiki (2D i 3D) oraz odtwarzanie dźwięku:

**Grafika:** wsparcie dla grafiki wektorowej (Core Graphics aka Quartz) i animacji 2D (Core Animation), OpenGL ES (sprzętowo akcelerowana grafika 3D/2D), obsługa czcionek wektorowych (Core Text), biblioteki do odczytu i zapisu popularnych formatów graficznych, dostęp do zdjęć i filmów w „galerii” użytkownika

**Wideo:** odtwarzanie i nagrywanie filmów (obsługa niskopoziomowa – Core Media), formaty: H264 (640x480, 320x240), MPEG-4 (640x480) – *mov, mp4, m4v, 3gp*

**Dźwięk:** nagrywanie i odtwarzanie dźwięku oraz sterowanie modułem wibracji, zarówno wysokopoziomowe jak i niskopoziomowe, zgodność z OpenAL (gry), wsparcie dla

# Cocoa Touch

Przede wszystkim obsługa interfejsu użytkownika

UIKit (UIKit.framework):

- Tworzenie i obsługa elementów interfejsu (view controllers)
- Obsługa zdarzeń
- Cut, copy, paste
- Obsługa kamery
- Dane z sensorów (akcelerometr)
- itd.

MapKit – mapa, „wbudowywalna” w aplikacji

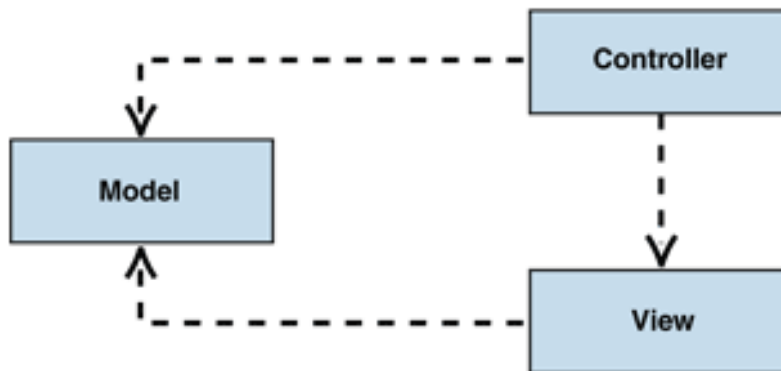
GameKit – wsparcie dla gier sieciowych

iAd Framework (iOS 4) – wsparcie dla reklam „bannerowych”

Address Book Framework – tworzenie, edycja i wybór kontaktów

# Intermedium - Model, View, Controller

MVC (Model, View, Controller) - wzorzec projektowy pozwalający na separację interfejsu użytkownika od „logiki biznesowej” czy też logiki przetwarzania danych



**Model** – dane i logika biznesowa aplikacji

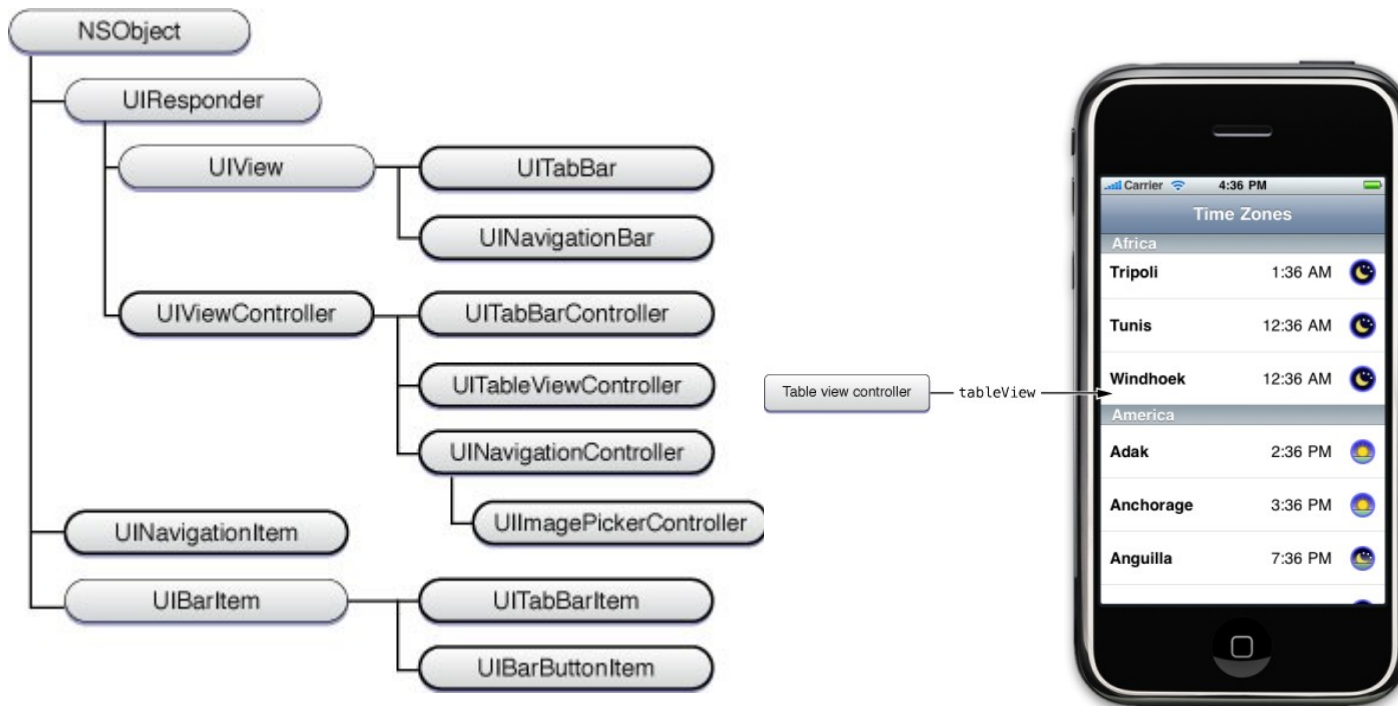
**Controller** – interakcja z użytkownikiem (odbieranie zdarzeń, informowanie modelu o zmianach, odpowiednie modyfikowanie interfejsu – view)

**View** – elementy interfejsu użytkownika

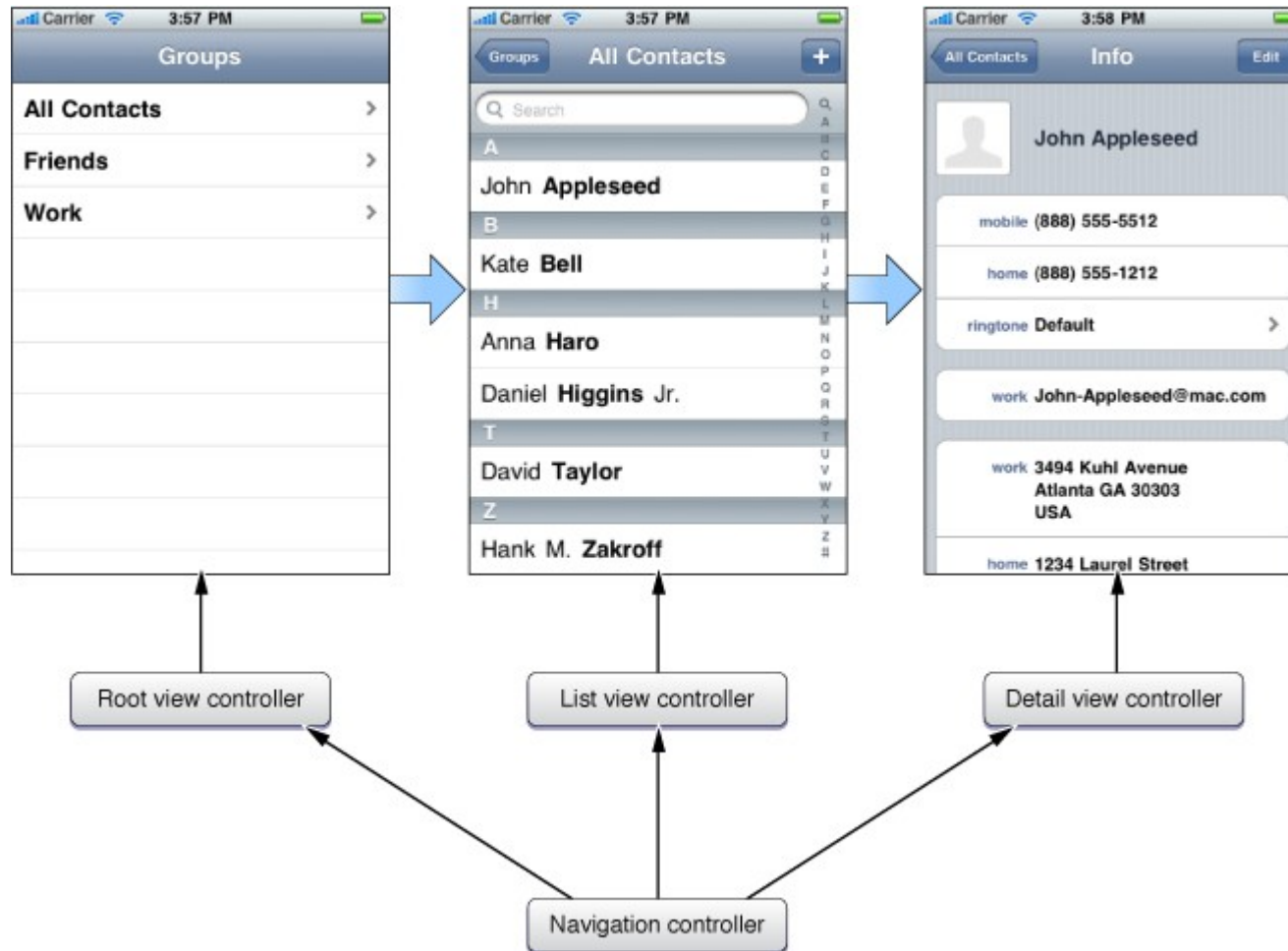
Bezpośrednia komunikacja nie zachodzi pomiędzy *view* i *model*

# View Controllers

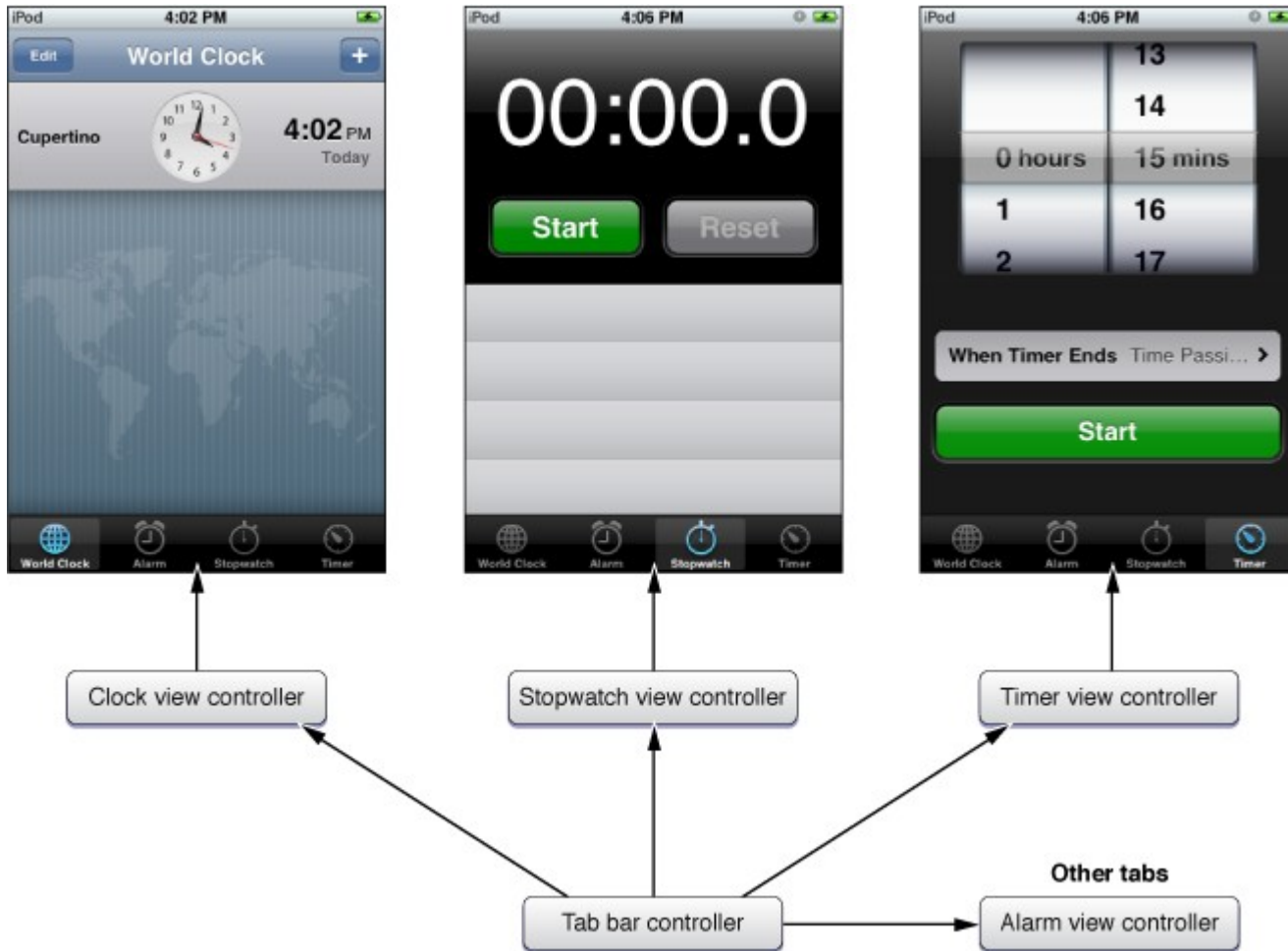
Standardowe obiekty UINavigationController zapewniają wsparcie dla typowych operacji związanych z przeglądaniem danych (np. przewijanie, zaznaczanie wierszy tabeli itp.) zajmujących pełny ekran urządzenia np.:



# View Controllers



# View Controllers





# Tworzenie aplikacji

Narzędzia jakie stoją do naszej dyspozycji zależą od tego, czy mamy do czynienia z urządzeniem *jailbroken* czy też nie...

Jeśli tak – dostępnych jest wiele narzędzi i języków programowania, także „on device” ... ale nasza aplikacja dystrybuowana może być jedynie nieoficjalnymi kanałami (np. Cydia)

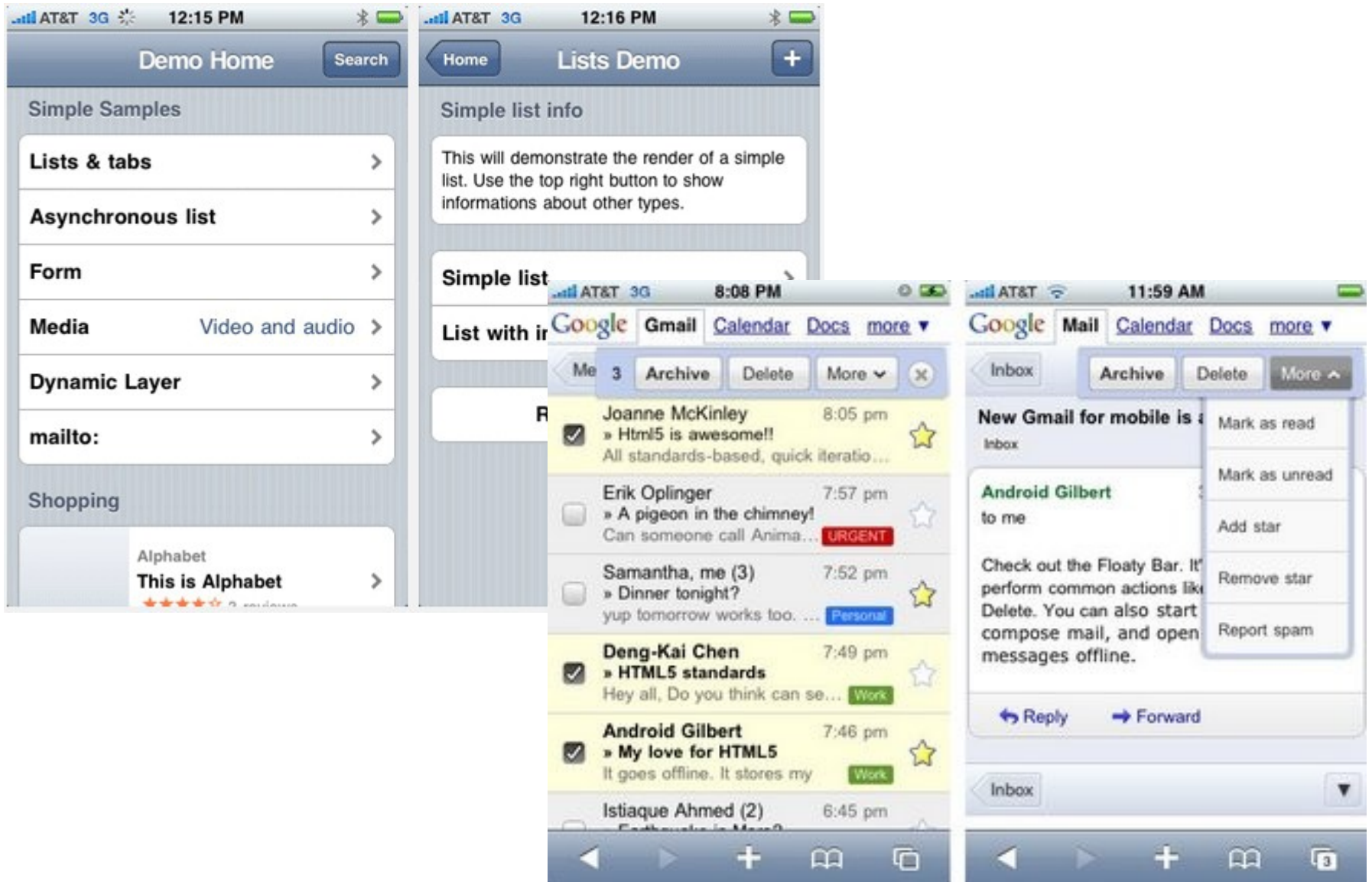
Jeśli nie – pozostaje nam oficjalny SDK, a to oznacza w praktyce Xcode + iPhone Simulator + Objective-C (ew. C / C++ *jeśli nasza aplikacja nie będzie intensywnie wykorzystywać Cocoa Touch: czytaj – będzie to gra*)

*A co jeśli chcemy jednocześnie przygotować wersję np. dla Androida i Windows Phone 7?*

*Jest jeszcze jedno rozwiązanie (nb. początkowo było to jedyne rozwiązanie przed wprowadzeniem SDK i AppStore) – aplikacje sieciowe (WebApps)*

Przeglądarka iOS (Safari) posiada zestaw rozszerzeń pozwalających na dostęp do wielu natywnych funkcjonalności

# iOS Web Apps



# Objective-C

Dziedziczy wszystkie cechy języka C bez wprowadzania zawichości znanych z C++

Model obiektowy oparty o komunikaty (Smalltalk) – typy nie są sprawdzane podczas kompilacji

Ułatwienia (szczególnie w Objective-C 2.0) znane z języków takich jak Java (iteratory, wyjątki)

Dość niskopoziomowy (np. zarządzanie pamięcią nie w pełni

automatyczne )

```
#import <Foundation/NSObject.h>

@interface Fraction: NSObject {
    int numerator;
    int denominator;
}

-(void) print;
-(void) setNumerator: (int) n;
-(void) setDenominator: (int) d;
-(int) numerator;
-(int) denominator;
@end
```

składnia np.

fraction.m

```
#import "Fraction.h"
#import <stdio.h>

@implementation Fraction
-(void) print {
    printf( "%i/%i", numerator, denominator );
}

-(void) setNumerator: (int) n {
    numerator = n;
}

-(void) setDenominator: (int) d {
    denominator = d;
}
```

```
#import <stdio.h>
#import "Fraction.h"

int main( int argc, const char *argv[] ) {
    // create a new instance
    Fraction *frac = [[Fraction alloc] init];
    Fraction *frac2 = [[Fraction alloc] init];
    Fraction *frac3 = [[Fraction alloc] initWithNumerator: 3 denominator: 10];

    // set the values
    [frac setNumerator: 1];
    [frac setDenominator: 3];

    // combined set
    [frac2 setNumerator: 1 andDenominator: 5];

    // print it
    printf( "The fraction is: " );
    [frac print];
    printf( "\n" );

    printf( "Fraction 2 is: " );
    [frac2 print];
    printf( "\n" );

    printf( "Fraction 3 is: " );
    [frac3 print];
    printf( "\n" );

    // free memory
    [frac release];
    [frac2 release];
    [frac3 release];
}
```

# iOS SDK

- Do tworzenia aplikacji potrzebujemy SDK – **Xcode** (środowisko IDE wraz z kompilatorem i edytorem kodu źródłowego), **Interface Builder** (narzędzie projektowania elementów widoków), **Simulator** (środowisko wykonania aplikacji – technicznie rzecz biorąc nie jest to emulator, albowiem wykonywany jest kod natywny x86)
- Wszystkie powyższe narzędzia wymagają komputera Mac z procesorem Intel
- SDK dostępne za darmo ze stron Apple (*developer.apple.com*) ale... aby móc uruchamiać aplikacje na rzeczywisty *developer* – to



jestrować się jako

# Stanford Programming Course

Bodajże najlepszy dostępny obecnie kurs online tworzenia aplikacji dla iOS

**<http://www.stanford.edu/class/cs193p>** (slajdy, materiały pomocnicze, kod źródłowy)

Sam wykład (w postaci nagrań wideo) dostępny poprzez iTunes w ramach usługi iTunesU uniwersytetu Stanforda (licencja CC-BY-SA)



**Cdn.**