
Modulo 2

Piattaforme e paradigmi di
sviluppo

-
- Recap (chi siamo e perché)
 - Piattaforme
 - Apple iOS
 - Microsoft Windows 8
 - Microsoft Windows Phone
 - BlackBerry
 - Bada
 - Tizen
 - Multipiattaforma
 - PhoneGap
 - Xamarin
 - Android
-

1.

Modalità e strumenti (recap)

1.1 Modalità di lezione

- **Lezioni frontali**

Aspetti e concetti della programmazione mobile.

- **Dimostrazioni**

Una o più demo dei concetti.

- **Esercizi di verifica**

Seguendo la traccia delle demo.

1.2 Strumenti

- **Repository mercurial**

<https://code.google.com/p/sviluppo-apps-per-dispositivi-smart/>

- Slide
- Materiali
- Progetti sviluppati
- Esercizi

- **Google Group**

<https://groups.google.com/forum/?hl=it&fromgroups=#!forum/corso-smart-apps>

2.

Piattaforme e caratteristiche

2.1

Apple iOS

2.1 iPhone - L'inizio della storia

Chiamato anche iPhone 2G o iPhone EDGE.

Rilasciato il **29 giugno 2007**.

Integra **tre dispositivi in uno**:

- computer palmare,
- telefono,
- iPod.



Sistema operativo **iPhone OS 1.0**

È il primo smartphone.

2.1 iPhone - L'inizio della storia

Prodotto di **fascia alta**.

Costava:

- **499\$** nella versione 4GB
- **599\$** nella versione da 8 GB

Venduti 270 000 telefoni in 30 ore.

Punta su **semplicità di utilizzo** e **design** a scapito di funzionalità avanzate.



Non fa moltissime cose
ma le fa meglio di chiunque altro

2.1 iPhone OS 1.0-2.0



2.1 iPhone OS 1.0-2.0

iPhone OS 1.0 06/2007 Disponibile su iPhone 2G

iPhone OS 1.1.2 09/2007 Multilanguage

JAILBRAKE

iPhone OS 2.0 08/2008 AppStore + iPhone Developer Program
iPhone 3G

2.1 iPhone OS 3.0



Springboard

2.1 iPhone OS 3.0

iPhone OS 3.0	06/2009	Molte nuove API Notifiche Push Copy & Paste (& Cut) Tethering Device Search iPhone 3GS
iPhone OS 3.1	09/2009	Abilitazione MMS API editing video Miglioramenti in OpenGL e Quartz
iPhone OS 3.2	04/2010	Lancio iPad (versione non rilasciata per iPhone)

2.1 iOS 4.0



2.1 iOS 4.0

<u>iOS 4.0</u>	06/2010	Cambio del nome (drop di alcuni device) Multitasking per le 3rd-party App Maggiori possibilità di personalizzazione Performance migliorate GameCenter, iAd, iBooks iPhone 4
<u>iOS 4.2.1 (4.2)</u>	11/2010	per iPhone/iPodTouch/iPad AirPlay Aggiunto la font Helvetica (ora default)

2.1 iOS 5.0



2.1 iOS 5.0

<u>iOS 5.0</u>	10/2011	Nuovo centro delle notifiche Integrazione con Twitter Compatibilità con iCloud Sync delle App su più dispositivi Importanti migliorie grafiche Siri (solo iPhone 4S ing/fra/ted) iPhone 4S XCode 4.2 Gcc->LLVM (ARC)
<u>iOS 5.1</u>	03/2012	Limite dimensione App tramite 3G (50 MB) Rilevamento dei volti nell'app Fotocamera

2.1 iOS 6.0



2.1 iOS 6.0

<u>iOS 6.0</u>	09/2012	Integrazione con Facebook Siri in italiano Mappe usa mappe proprietarie (torna 12/12) Miglioramento supporto alla lingua cinese App YouTube rimossa dalla home PassBook iPhone 5
<u>iOS 6.1</u>	01/2013	Supporto a LTE carrier Eu/As/Af/Am(Sud) "Report a Problem" più largo in Mappe
<u>iOS 6.1.3</u>	21/02/13	Ultimo aggiornamento ad oggi.

2.1 Situazione attuale

Più di 365 milioni di device iOS.

Più di 250 milioni di iPhone venduti.

Più di 100 milioni di iPad.

Un azione Apple Inc. vale \$433.89 (14/03/13).

Più di 500 milioni di account iTunes.

Più di 807 mila App su AppStore in 155 paesi.

Più di 40 miliardi di download (di cui 20 solo nel 2012).

50 mila download per App (avg).

Prezzo medio \$1.48.

Più di 7 miliardi di \$ agli sviluppatori.

2.1 Caratterizzazione - Contro

Software & Hardware strettamente legati

Hardware solo di fascia alta

WWDC - L'unico evento ufficiale

Sistema tendenzialmente chiuso (Codice sotto NDA)

Necessità di device particolari per essere programmato

2.1 Caratterizzazione - Pro

iOS SDK

WWDC - Unico evento ma grande risonanza

Strumenti curati, integrati e performanti (XCode, iOS Simulator, ecc.)

Ottima documentazione

Revenue alte per i programmatori

2.1 Sviluppare per iOS - Requisiti

Apple Id (gratuito)

Computer Apple (MacBook, iMac, ecc.)

Mac OSX

XCode (gratuito dal Mac App Store)

Apple Developer Program

Device Apple (iPhone, iPad, iPod Touch)

2.1 Sviluppare per iOS - Requisiti

Apple Id (gratuito)

Computer Apple (MacBook, iMac, ecc.)

Mac OSX

XCode (gratuito dal Mac App Store)

Apple Developer Program

Device Apple (iPhone, iPad, iPod Touch)

2.1 Apple Developer Program

Iscrizione necessaria per:

pubblicare app nel App Store

testare le proprie app su device reali

Ha **validità annuale**

A pagamento per Singoli e Aziende

Gratuito per enti universitari

2.1 Apple Developer Program



iOS Developer Program (\$99/year)

Test su dispositivi reali + Distribuzione su App Store +
Supporto nel forum ufficiale + test su 100 device



iOS Developer Enterprise Program (\$299/year)

Come sopra + distribuzione in-house delle proprie app +
test su infiniti device



iOS Developer University Program (Free)

Come sopra + limite a 200 sviluppatori

2.1 Sviluppare per iOS - Requisiti

Apple Id (gratuito)

Computer Apple (MacBook, iMac, ecc.)

Mac OSX

XCode (gratuito dal Mac App Store)

Apple Developer Program

Device Apple (iPhone, iPad, iPod Touch)

2.1 Device Apple - non solo iPhone



2.1 Device Apple



iPhone 3GS



iPod 4 th



iPad 2



iPhone 4



iPod 5th



iPad 3



iPhone 4s



iPad 4



iPhone 5



iPad Mini

2.1 Device Apple



iPhone 3G

320x480 px

116 ppi



iPhone 4 / 4S
iPod 4 th

640x480 px

326 ppi



iPhone 5
iPod 5 th

640x1136 px

332 ppi

2.1 Device Apple



iPad 1 / 2

1024x768 px

132 ppi



iPad 3 / 4

2048x1536 px

264 ppi



iPad mini

1024x768 px

162 ppi

2.1 Device Apple



ARMv7

ARMv7s

1 GB

iPad 3

iPhone 5

iPad 4

512 MB

iPhone 4

iPhone 4S
iPad 2
iPod 5th

256 MB

iPhone 3Gs
iPod 3rd

iPad 1
iPod 4th

2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guideline Apple

Conoscere il linguaggio Objective-C (v 2.0)

2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guideline Apple

Conoscere il linguaggio Objective-C (v 2.0)

2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guideline Apple

Conoscere il linguaggio Objective-C (v 2.0)

2.1 Strumenti

XCode

iOS Simulator

Instruments

Demo

2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guideline Apple

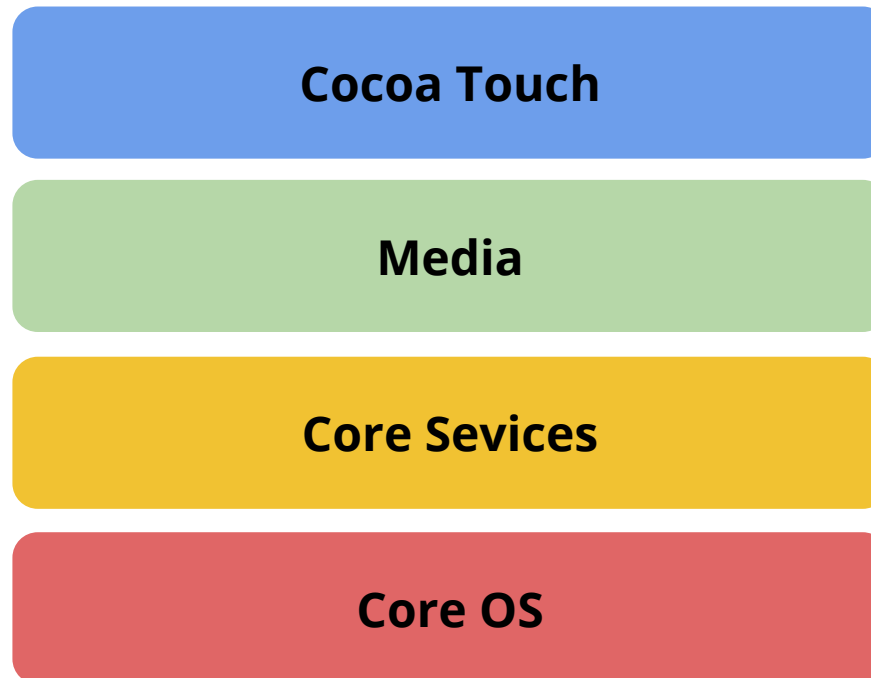
Conoscere il linguaggio Objective-C (v 2.0)

2.1 iOS

Come Mac OS X è una derivazione di UNIX (famiglia **BSD**).

Usa un microkernel XNU Mach basato su Darwin OS.

Ha quattro livelli di astrazione:



2.1 iOS Software Stack

Cocoa Touch

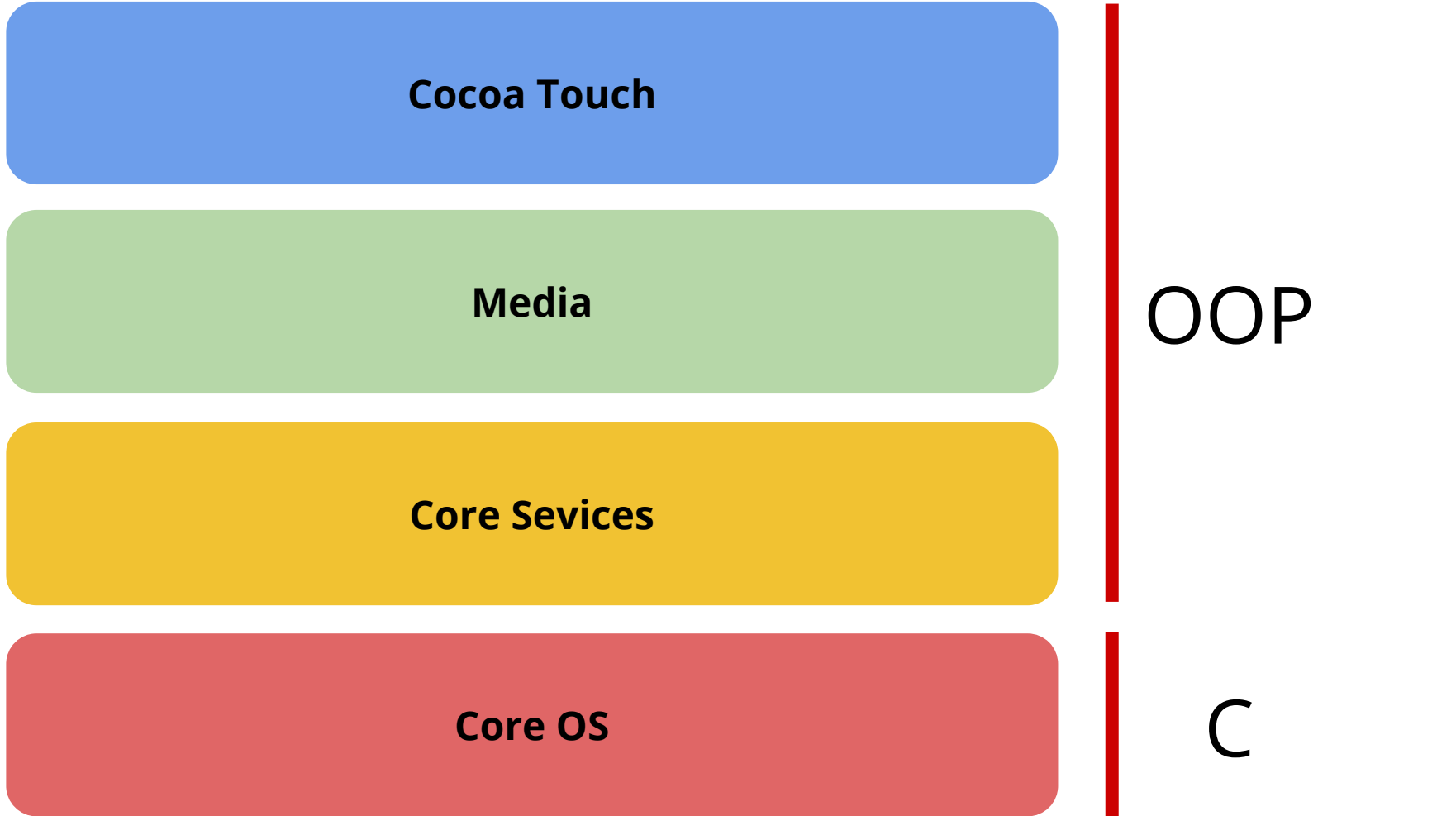
Media

Core Services

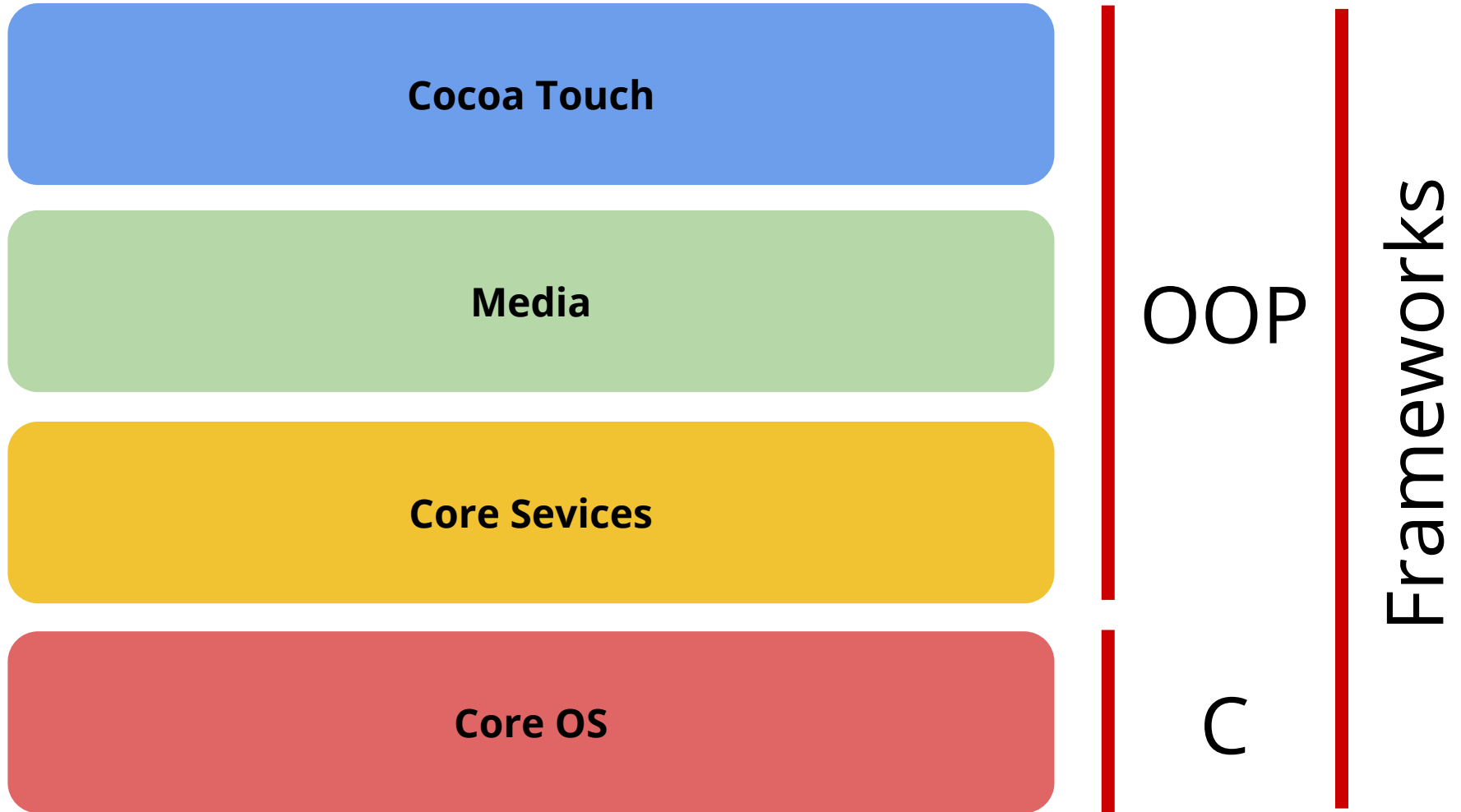
Core OS

OOP

C



2.1 iOS Software Stack



2.1 iOS Frameworks

Pacchetti di risorse relazionate fra loro
Grande aiuto per il programmatore

Table B-1 Device frameworks

Name	First available	Prefixes	Description
<code>Accelerate.framework</code>	4.0	<code>cbblas</code> , <code>vDSP</code>	Contains accelerated math and DSP functions.
<code>Accounts.framework</code>	5.0	<code>AC</code>	Contains interfaces for managing access to Accounts Framework .
<code>AddressBook.framework</code>	2.0	<code>AB</code>	Contains functions for accessing the user's Address Book Framework .
<code>AddressBookUI.framework</code>	2.0	<code>AB</code>	Contains classes for displaying the system address book. See "Address Book UI Framework" .
<code>AdSupport.framework</code>	6.0	<code>AS</code>	Contains a class for gathering analytics. See "Ad Support Framework" .
<code>AssetsLibrary.framework</code>	4.0	<code>AL</code>	Contains classes for accessing the user's photo library. See "Assets Library Framework" .
<code>AudioToolbox.framework</code>	2.0	<code>AU</code> , <code>Audio</code>	Contains the interfaces for handling audio processing. See "Core Audio" .

2.1 iOS Software Stack

Cocoa Touch

Media

Core Services

Core OS

API in C
di OSX Kernel

2.1 iOS Software Stack

Cocoa Touch

Media

Core Services

Core OS

Object
Wrapper
basato su ↓

2.1 iOS Software Stack

Cocoa Touch

Media

Playback
+ Editing

Core Services

Core OS

2.1 iOS Software Stack

Cocoa Touch

Media

Core Services

Core OS

Infrastruttura di base delle app: multitasking, touch-based input, push notifications, ecc.

2.1 iOS - quale versione?

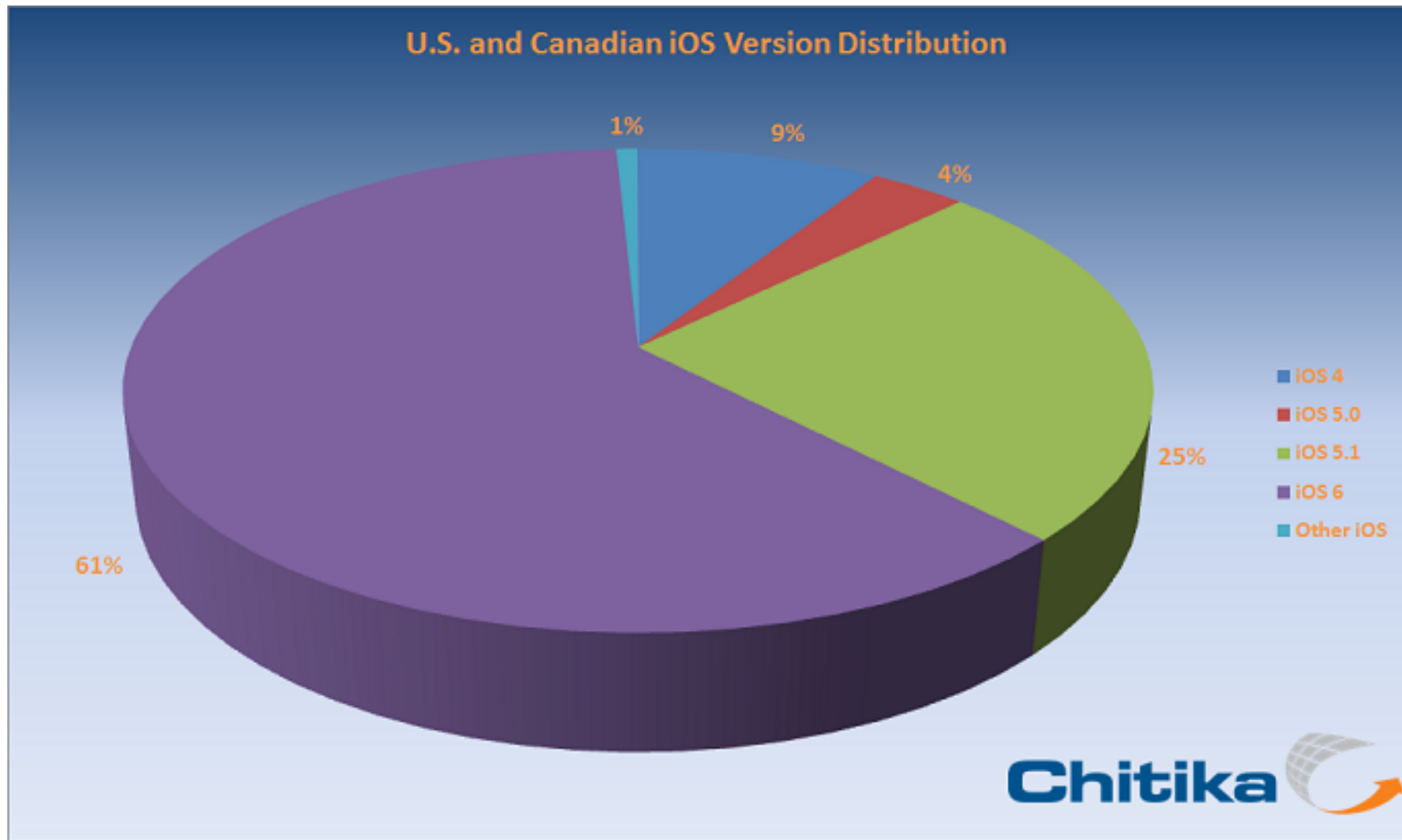
Current versions

[\[edit\]](#)

[Obsolete](#) [Discontinued](#) [Current](#) [Beta](#)

Version	Build	Release date	Highest version for
3.1.3	7E18	February 2, 2010; 3 years ago	iPhone (original) ; iPod Touch (1st generation)
4.2.1	8C148	November 22, 2010; 2 years ago	iPhone 3G ; iPod Touch (2nd generation)
5.1.1	9B206	May 7, 2012; 9 months ago	iPod Touch (3rd generation) ; iPad (original)
6.1.2	10B146	February 19, 2013; 20 days ago	iPhone 3GS , iPhone 4 , iPhone 4S , iPhone 5 ; iPod Touch (4th & 5th generation) ; iPad 2 , iPad (3rd generation) , iPad (4th generation) , iPad mini
6.1.3	10B318	February 21, 2013; 18 days ago	iPhone 3GS , iPhone 4 , iPhone 4S , iPhone 5 ; iPod Touch (4th & 5th generation) ; iPad 2 , iPad (3rd generation) , iPad (4th generation) , iPad mini

2.1 iOS - Versions distribution

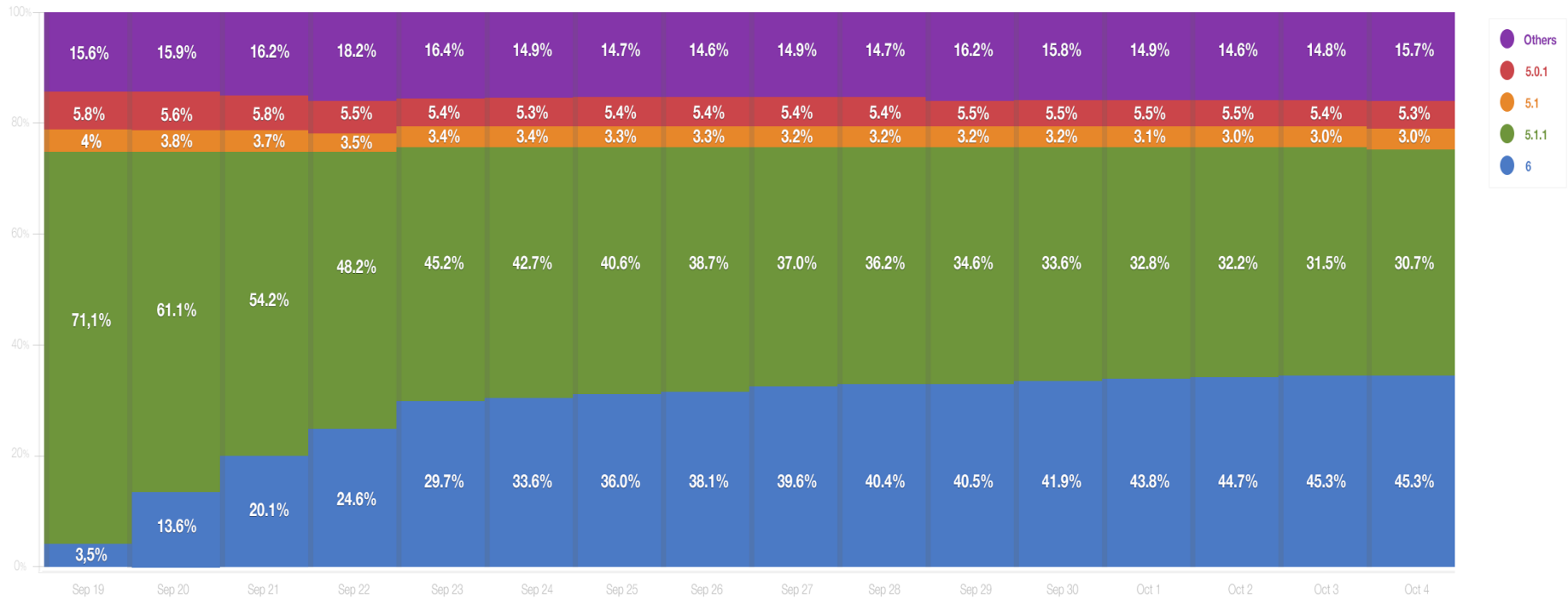


2 ottobre 2012

2.1 iOS - Adoption rate



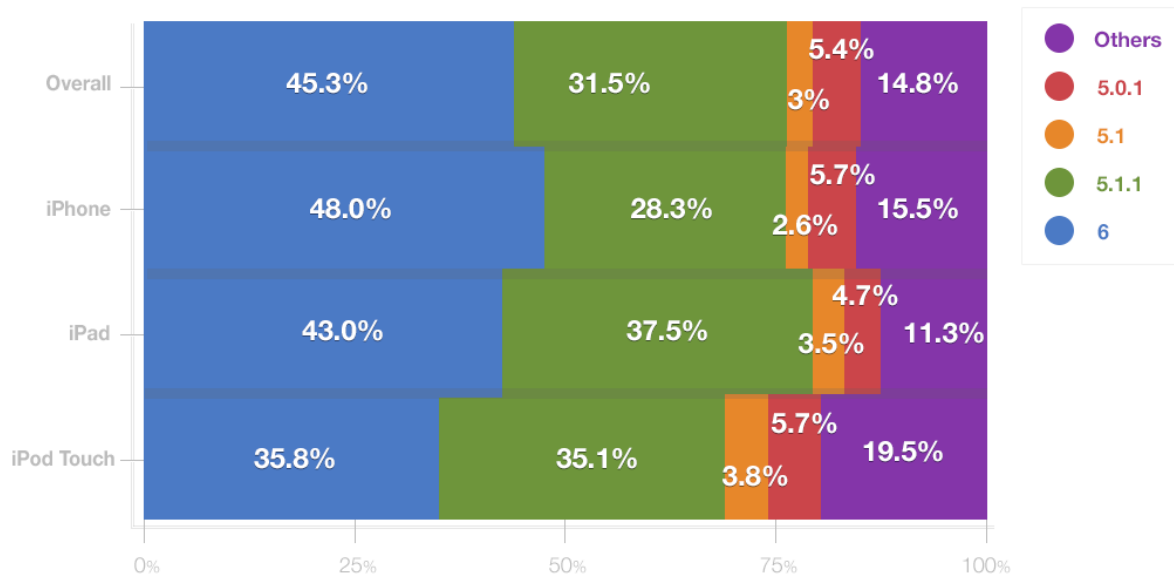
Unique Users by iOS version : Daily



2.1 iOS - Adoption rate



Unique Users by Device and iOS Version



2.1 iOS - quale versione?

L'ultima!
(iOS 6)

2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guidelines Apple

Conoscere il linguaggio Objective-C (v 2.0)

2.1 Apple Guidelines

Si tratta principalmente line guida che sfruttano pattern di programmazione ad oggetti per assicurare:

un ottimo riuso del codice

basso accoppiamento fra le classi

Principali pattern:

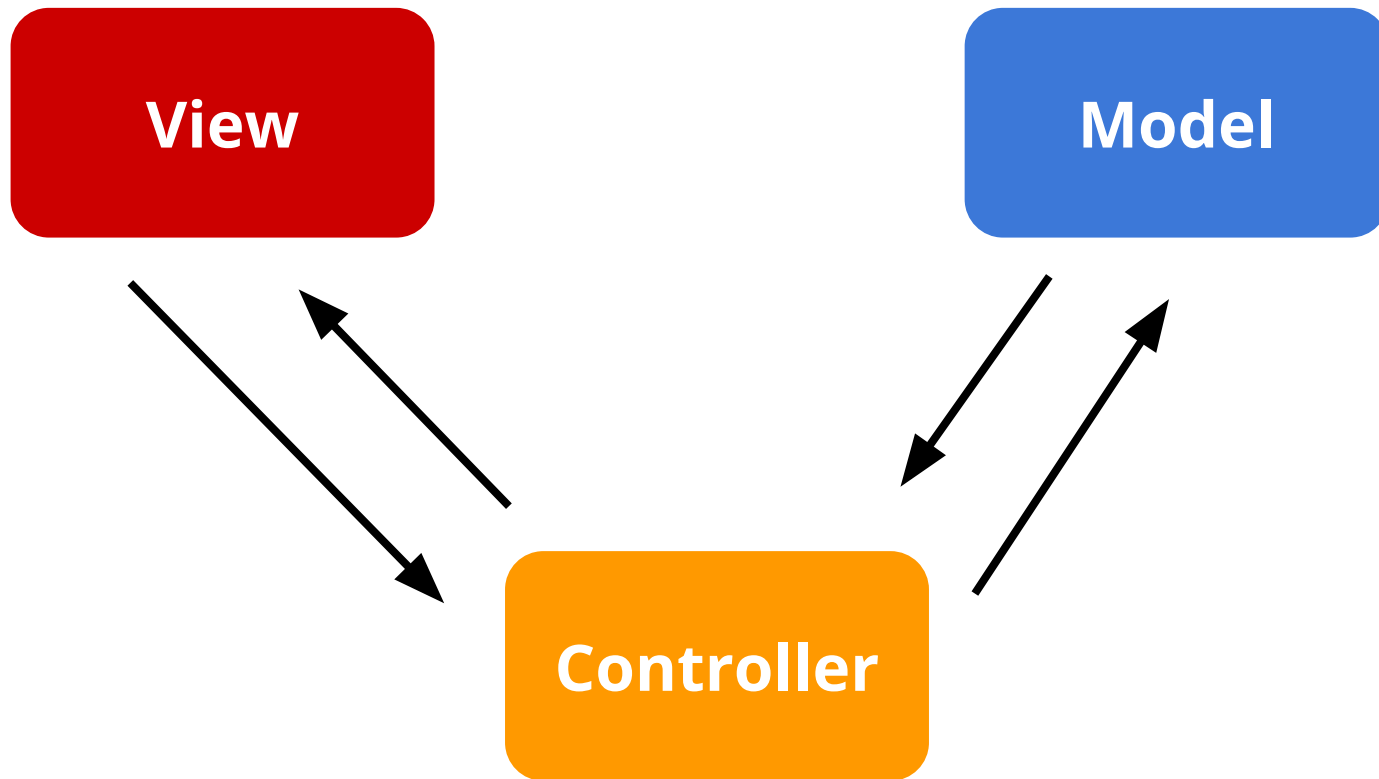
MVC

Target/Action

Delegates

Notification

2.1 Apple Guidelines - MVC



2.1 Apple Guidelines - Target/Action



Target

```
/*
 File: RecipeAddViewController.h
 Abstract: View controller to allow the user to add a
 new recipe
 |
 Version: 1.4
 */
@class Recipe;

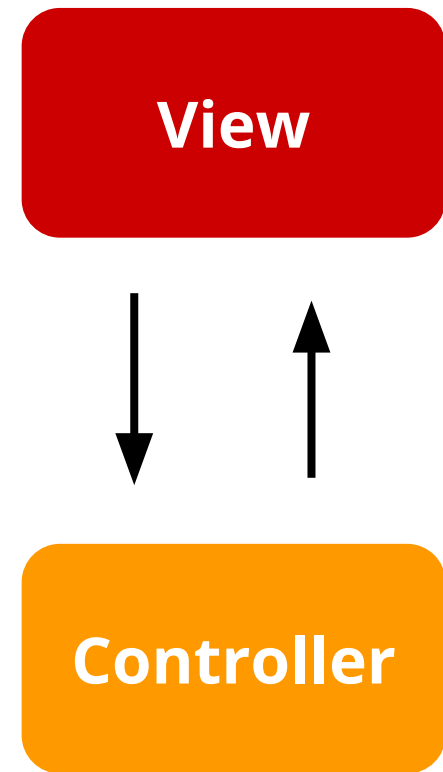
@interface RecipeAddViewController : UIViewController
{
    Recipe *recipe;
    UITextField *nameTextField;
}

@property(n nonatomic, retain) Recipe *recipe;
@property(n nonatomic, retain) IBOutlet UITextField *
nameTextField;

- (void)saveRecipe:(id)sender;
- (void)cancel;

@end
```

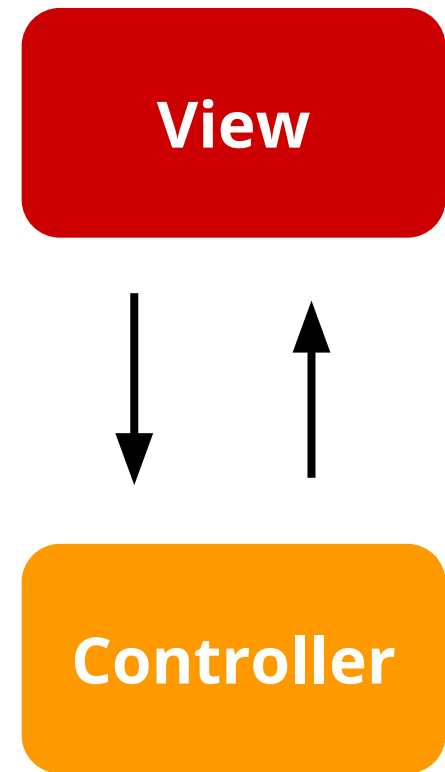
Action Method



2.1 Apple Guidelines - Delegates

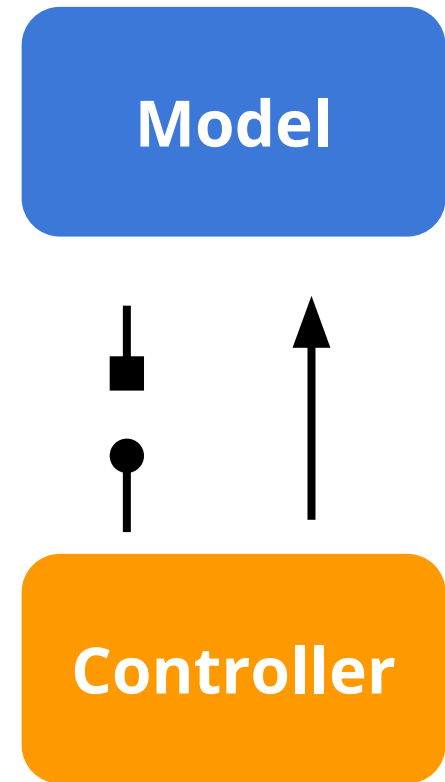
Un delegato reagisce alle modifiche avvenute su un altro oggetto influenzandone il comportamento.

Basato sui costrutti Objective-C dei Blocks e dei Protocol



2.1 Apple Guidelines - Notification

La traduzione in Design Patter del
Hollywood Principle:
"Don't call us, we'll call you"



2.1 Sviluppare per iOS - Requisiti (2)

Conoscere il linguaggio C

Conoscere i principi base di OOP

Sapere utilizzare gli strumenti messi a disposizione e l'SDK

Conoscere le basi di iOS

Conoscere e saper applicare le principali guideline Apple

Conoscere il linguaggio Objective-C (v 2.0)

2.1 Objective-C in a (μ)nutshell

È un superset di C

- Supporta la stessa sintassi del C

- Usa gli stessi tipi primitivi

- Mantiene le strutture di controllo

- Puntatori, Funzioni, Struct

- Offre l'accesso alle librerie standard C `stdlib.h` e `stdio.h`

2.1 Objective-C in a (μ)nutshell

È Object Oriented

- Definizione dei metodi di classe e di istanza

- Getter/Setter sintetizzati automaticamente

- Typing statico e dinamico

- Definizione dei Blocks

- Protocols e categories

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextViewDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextViewDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
```

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextFieldDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextFieldDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextViewDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```


2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextViewDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```

2.1 Objective-C in a (μ)nutshell

Le classi vengono dichiarate nei **file .h**

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

@interface InsertViewController : UIViewController <UITextViewDelegate> {
    UITextField *_nameTextField;
    int count;
}

-(void)addButtonTapped;
+(void)resetButtonTapped;

@end
|
```

2.1 Objective-C in a (μ)nutshell

Dichiarazione dei metodi

```
-(NSInteger) calcolaSomma:(int) addendo1 piu:(int) addendo2
```

2.1 Objective-C in a (μ)nutshell

Dichiarazione dei metodi

```
-(NSInteger) calcolaSomma:(int) addendo1 piu:(int) addendo2
```

2.1 Objective-C in a (μ)nutshell

Dichiarazione dei metodi

```
-(NSInteger) calcolaSomma:(int) addendo1 piu:(int) addendo2
```

2.1 Objective-C in a (μ)nutshell

Dichiarazione dei metodi

```
-(NSInteger) calcolaSomma:(int) addendo1 piu:(int) addendo2
```

2.1 Objective-C in a (μ)nutshell

Dichiarazione dei metodi

```
-(NSInteger) calcolaSomma:(int) addendo1 piu:(int) addendo2
```

```
-calcolaSomma:piu:
```

2.1 Objective-C in a (μ)nutshell

e implementate nei **file .m**

```
@implementation ListViewController
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        self.view.backgroundColor = [UIColor blueColor];
    }
    return self;
}
```


2.1 Objective-C in a (μ)nutshell

e implementate nei **file .m**

```
@implementation ListViewController
```

```
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil  
{  
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];  
    if (self) {  
        self.view.backgroundColor = [UIColor blueColor];  
    }  
    return self;  
}
```

2.1 Objective-C in a (μ)nutshell

e implementate nei **file .m**

```
@implementation ListViewController
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        self.view.backgroundColor = [UIColor blueColor];
    }
    return self;
}
```

2.1 Objective-C in a (μ)nutshell

e implementate nei **file .m**

```
@implementation ListViewController
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        self.view.backgroundColor = [UIColor blueColor];
    }
    return self;
}
```

2.1 Hello World

Vediamo un semplicissimo progetto per iPhone

Demo

2.1 References

<http://developer.apple.com/library/ios/#documentation>

<http://148apps.biz/app-store-metrics>

<http://www.nasdaq.com>

<http://chitika.com/insights/2012/ios-6-adoption-one-month/>

<http://techcrunch.com/2012/10/05/after-early-boom-new-numbers-suggest-ios-6-adoption-may-be-reaching-a-plateau/>

[http://en.wikipedia.org/wiki/App_Store_\(iOS\)](http://en.wikipedia.org/wiki/App_Store_(iOS))

http://en.wikipedia.org/wiki/IOS_version_history#iOS_4.x_2

<http://www.styletechblog.com/2012/07/ios-history>

<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>

<https://itunes.apple.com/us/itunes-u/corso-ios/id434824250>

<http://www.slideshare.net/MassimoOliviero/ios-getting-started>

Beginning iOS 5 Development, APress, D.Mark, J.Nutting, J.LaMarche - 2011 ISBN:9781430236054

Design Patterns: Elements of Reusable Object-Oriented Software: Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides.

2.2

Ecosistema Microsoft

2.2.1

Windows 8 (Windows RT)

2.2.1 Windows RT ?

Risposta al fenomeno tablet/iPad.

Un nuovo Windows "senza compromessi".



2.2.1 Windows Store

Da un Windows tradizionale ad un OS "app-centric":

- Windows Store
- Single (dual) tasking
- Nuovo *Start menu*
- Pervasivo uso del touch
- Nuovo paradigma UI
"Metro" / "Modern" UI



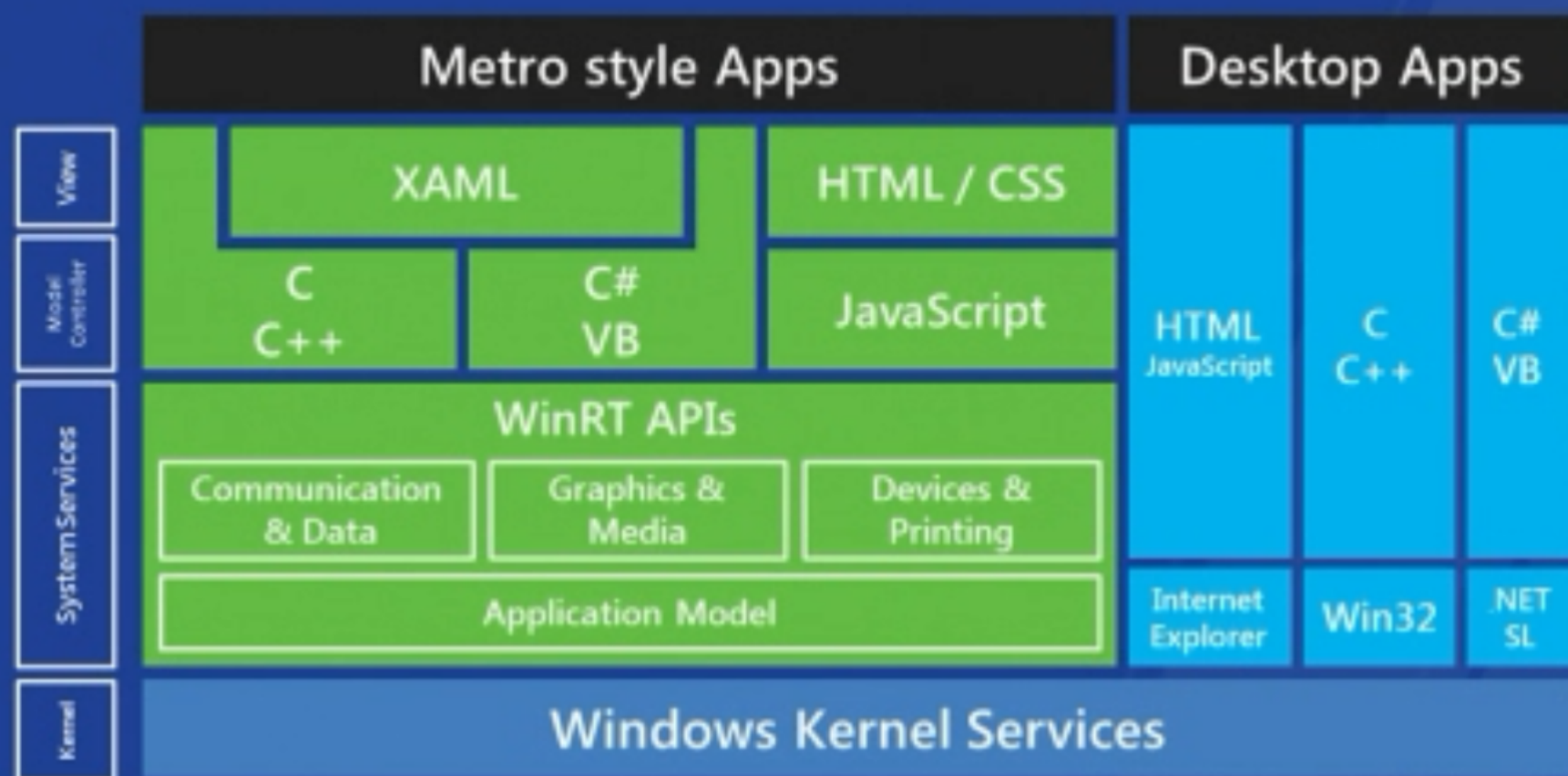
2.2.1 Compatibilità

Windows 8 mantiene la retrocompatibilità pressochè totale con Windows 7:

- Classiche applicazioni Win32/COM
- Applicazioni .NET
- DirectX, XAudio, Media Foundation, etc.

Attenzione: tutto questo è però **assente** su Windows RT (e.g. su "*Surface RT*").

Windows 8 Platform and Tools



2.2.1 Modern UI

- Layout grafico pulito ed aperto
- Gerarchia informativa
- Sfruttare i bordi (*context* e *charms* bar)
- Live-tile
- Fast & fluid

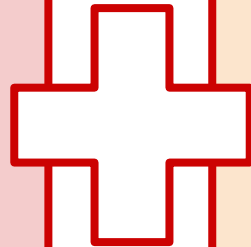


2.2.1 Modalità

- Applicazione singola
 - Modalità *snap*
 - Charms bar
 - Search
 - Share
 - Devices
 - Settings
-

2.2.1 Sviluppo e SDK

Visual Studio 2012



Windows SDK 8.0

oppure

Visual Studio Express 2012
for Windows 8

Free!

Metro style application APIs

User Interface

HTML5/CSS

XAML

DirectX

Controls

Data Binding

SVG

Tiles

Input

Accessibility

Printing

Devices

Geolocation

Portable

Sensors

NFC

Communications & Data

Contracts

Local & Cloud Storage

Web

Notifications

Streams

Background
Transfer

XML

Networking

SMS

Media

Playback

Capture

PlayTo

Visual
Effects

Fundamentals

Application Services

Threading/Timers

Memory Management

Authentication

Cryptography

Globalization

2.2.1 Pattern di programmazione

- **MVVM + Data binding**

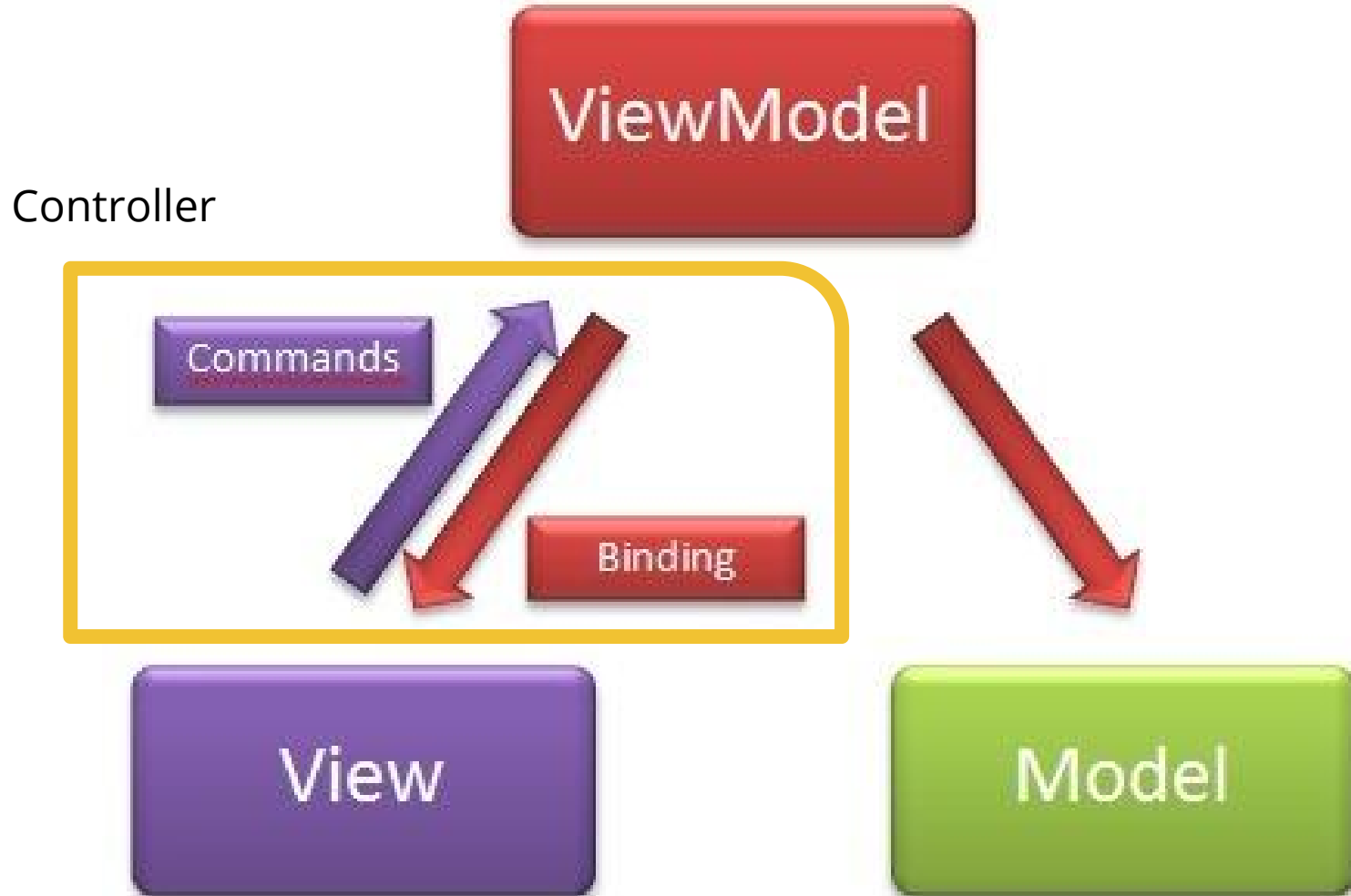
Martin Fowler "*Presentation Model*".

Model-View-ViewModel e binding automatico dei dati all'interfaccia grafica. Il *controller* è "automatico".

- **Async**

Tutto è sempre asincrono. Non esistono API sincrone in WinRT. Supporto tramite nuove feature di C# 5.

2.2.1 Model View ViewModel



2.2.1 ViewModel

```
public class ItemViewModel {  
  
    public string Title  
        { get; private set; }  
  
    public string Description  
        { get; private set; }  
  
}
```

Classe "POCO", ma può essere di più...

2.2.1 View

```
<ApplicationPage
  <StackPanel
    x:Name="LayoutRoot"
    Background="Transparent">
      <TextBlock
        Style="{StaticResource LargeStyle}">
          Titolo</TextBlock>
      <Button>Push me!</Button>
    </StackPanel>
  </ApplicationPage>
```

2.2.1 Data binding

```
<TextBlock  
    Text="{Binding Path=Title}" />
```

```
<TextBlock  
    Text="{Binding Path=Description}" />
```

La view si aggiorna automaticamente.

2.2.1 Data binding (2)

- Path
- Mode: OneTime, OneWay, TwoWay
- Converter: ValueConverter

Source → *Target*

- Boolean to image
 - Enum to visibility
 - Etc...
-

2.2.2

Windows Phone

2.2.2 Da Windows Mobile...

Ottobre 2010: Windows Phone 7

2004 "Photon", 2008 da Windows Mobile a WP.
Modern UI. Live Tiles. Hubs.
Office, Skydrive, Sharepoint, etc. integration.

Maggio 2011: Windows Phone 7.5 *Mango*

Multitasking. Twitter hub.

Ottobre 2012: Windows Phone 8 (e 7.8)

Nuovo kernel (CE → NT).



Windows Phone

2.2.2 Due universi

WP 7.0-7.8

- Kernel Win CE
- .NET Compact 3.5
- Silverlight 3+
- XNA

WP 8.0

- Kernel Win NT
 - .NET Core 4.5
 - WinRT
 - DirectX
-

2.2.2 Compatibilità

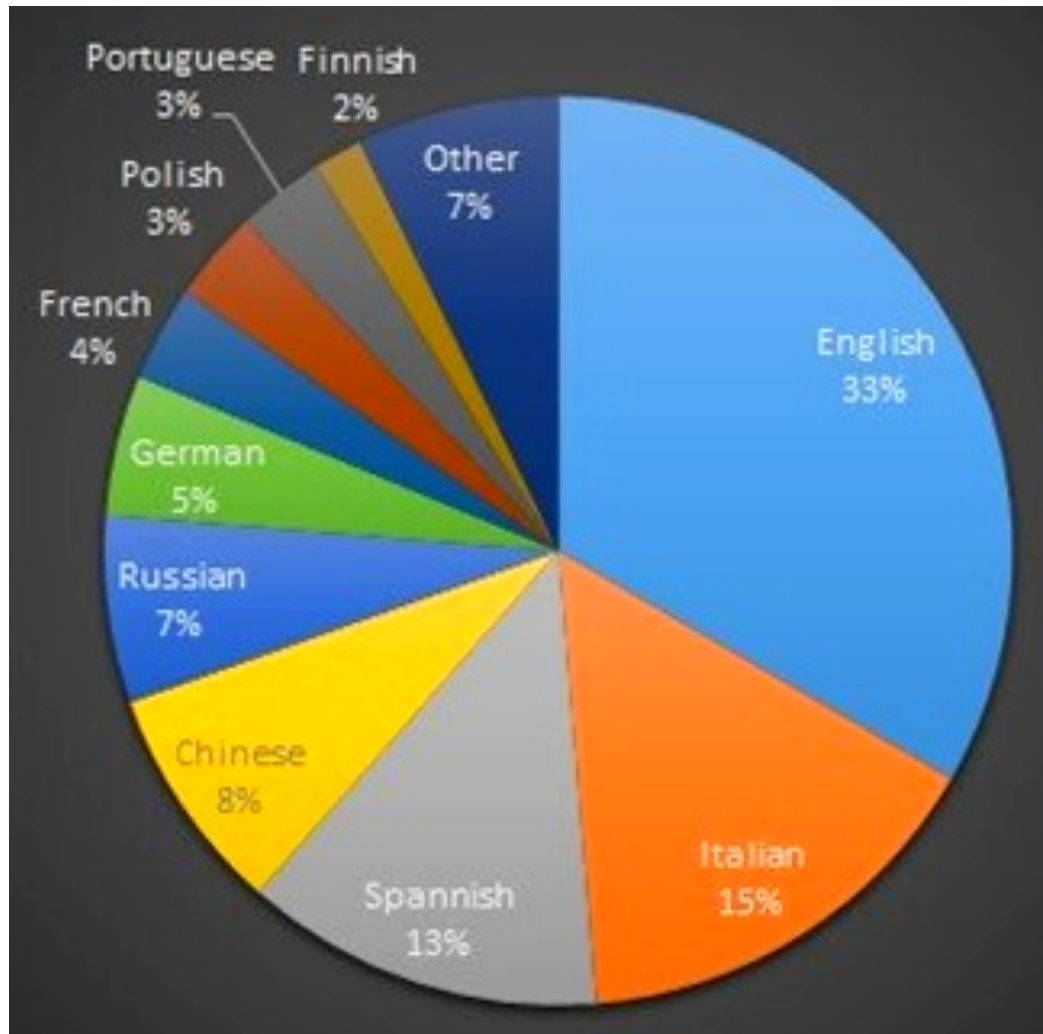
- Windows Phone 8 è retrocompatibile con applicazioni WP7.
- WP7.8 **non** è compatibile con nuove applicazioni WP8!

C'è intersezione tra i due ambienti (.NET), per cui è possibile (facile?) sviluppare applicazioni per WP7.8 e WP8.

2.2.2 Share

- Windows Mobile, *bye bye*: → 0%.
 - Nel **2013**: 3-5%.
Terza piattaforma più popolare,
compete con Blackberry.
-

2.2.2 Mercato italiano



Stando ad un report
Kandar del 2013,

Windows Phone 8
+ Nokia Lumia

hanno ottenuto uno share
complessivo del

14-15%

2.2.2 Modern UI in WP



2.2.2 Modello applicazioni

Ispirato al modello di iOS:

Mai più di 1 applicazione in esecuzione.

Forti limiti ai servizi in background.

Forti limiti all'accesso del HW.

Esperienza coesa, fluida ed affidabile.

Conservazione della batteria.

Live tiles con background agents / push.

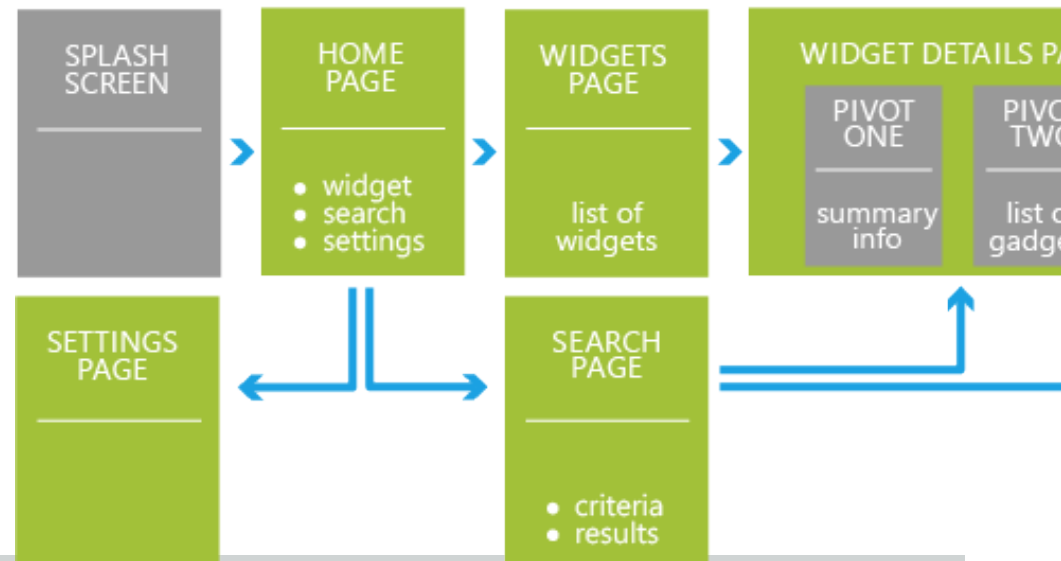
2.2.2 Modello a pagine

Applicazioni composte da elementi *Page*.

Navigazione *stack-based*, ogni pagina ha URI.

Deep-linking con URI interni (stack vuoto).

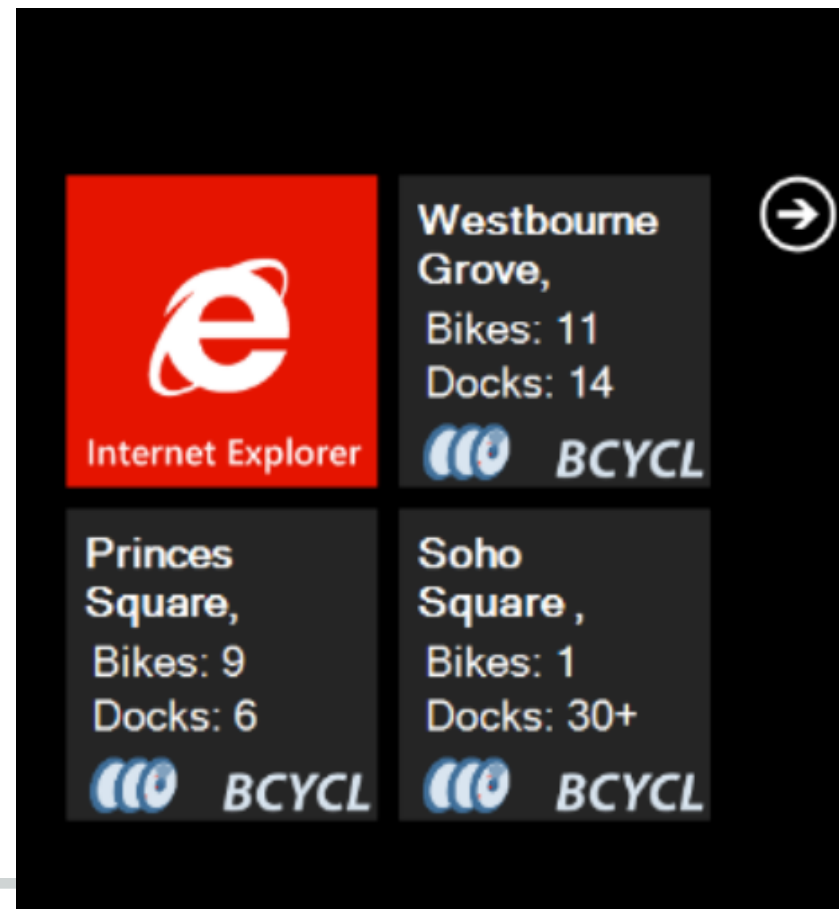
Parametri (*query string*) utilizzabili per la navigazione.
`page.xaml?id=123`



2.2.2 A cosa serve il *deep-linking*?

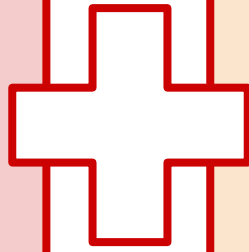
- Secondary live-tiles: collegate ad una pagina specifica.
- Toast notifications: aprono la pagina d'interesse.

Non hanno stack di navigazione, il tasto "back" porta fuori dalla applicazione (*integrated feel*).



2.2.2 Sviluppo e SDK

Visual Studio 2012



WP SDK 8.0

oppure

Visual Studio Express 2012
for Windows Phone

Free!

Windows Phone 8 Applications

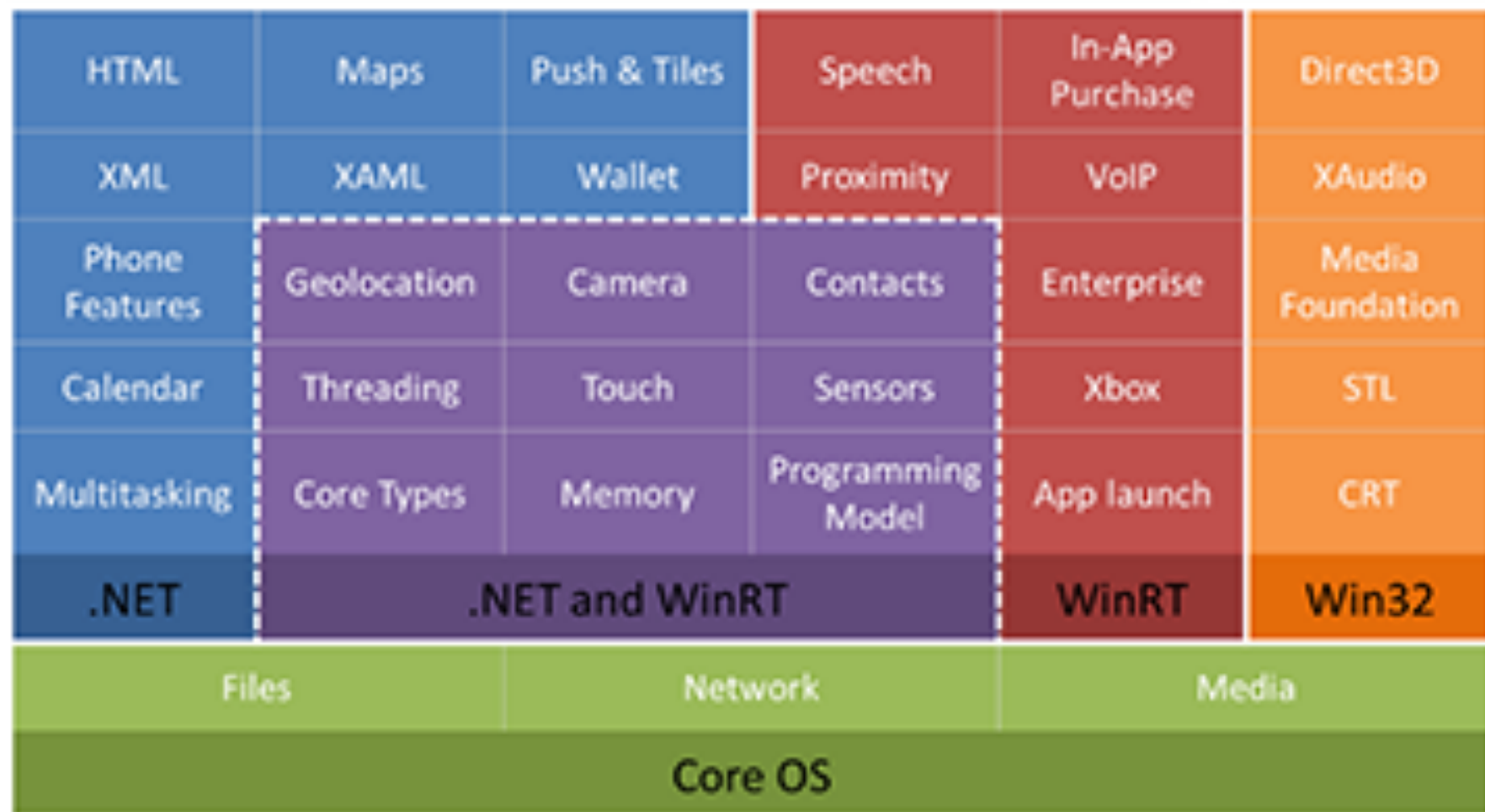
XAML

XAML + Native

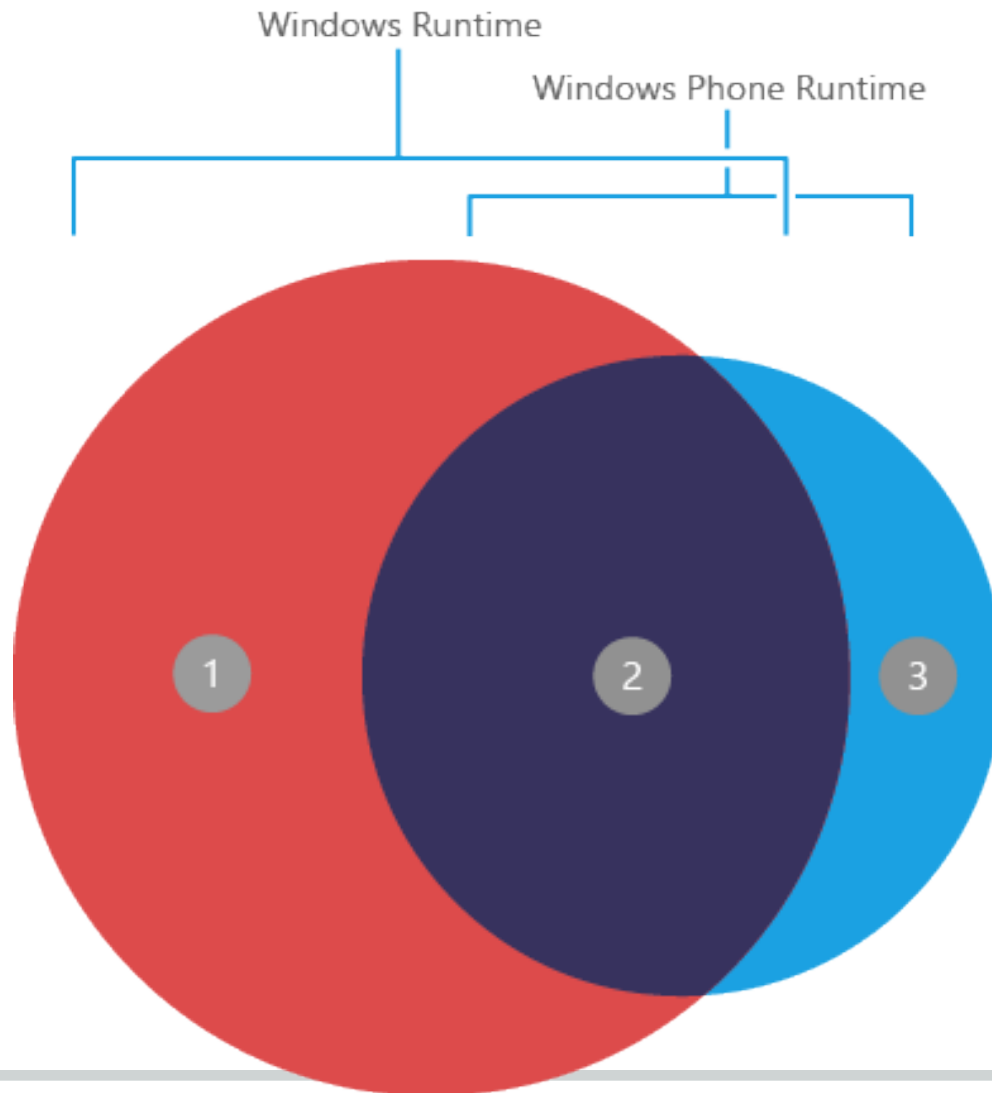
Native

C# and VB

C++



2.2.2 Windows Phone 8 + WinRT



2.3

BlackBerry 10

2.3 Cos'è

BlackBerry è un marchio appartenente alla società BlackBerry Limited (prima Research In Motion - RIM)



BlackBerry 10 è il nuovo moderno sistema operativo lanciato sul mercato nell'ultimo anno

2.3 Evoluzione

Il primo dispositivo BB fu un cercapersone!
Il BB 850.

Dispositivi pensati per utenti di profilo
Professional/Enterprise



Caratteristiche:

Tastiera fisica

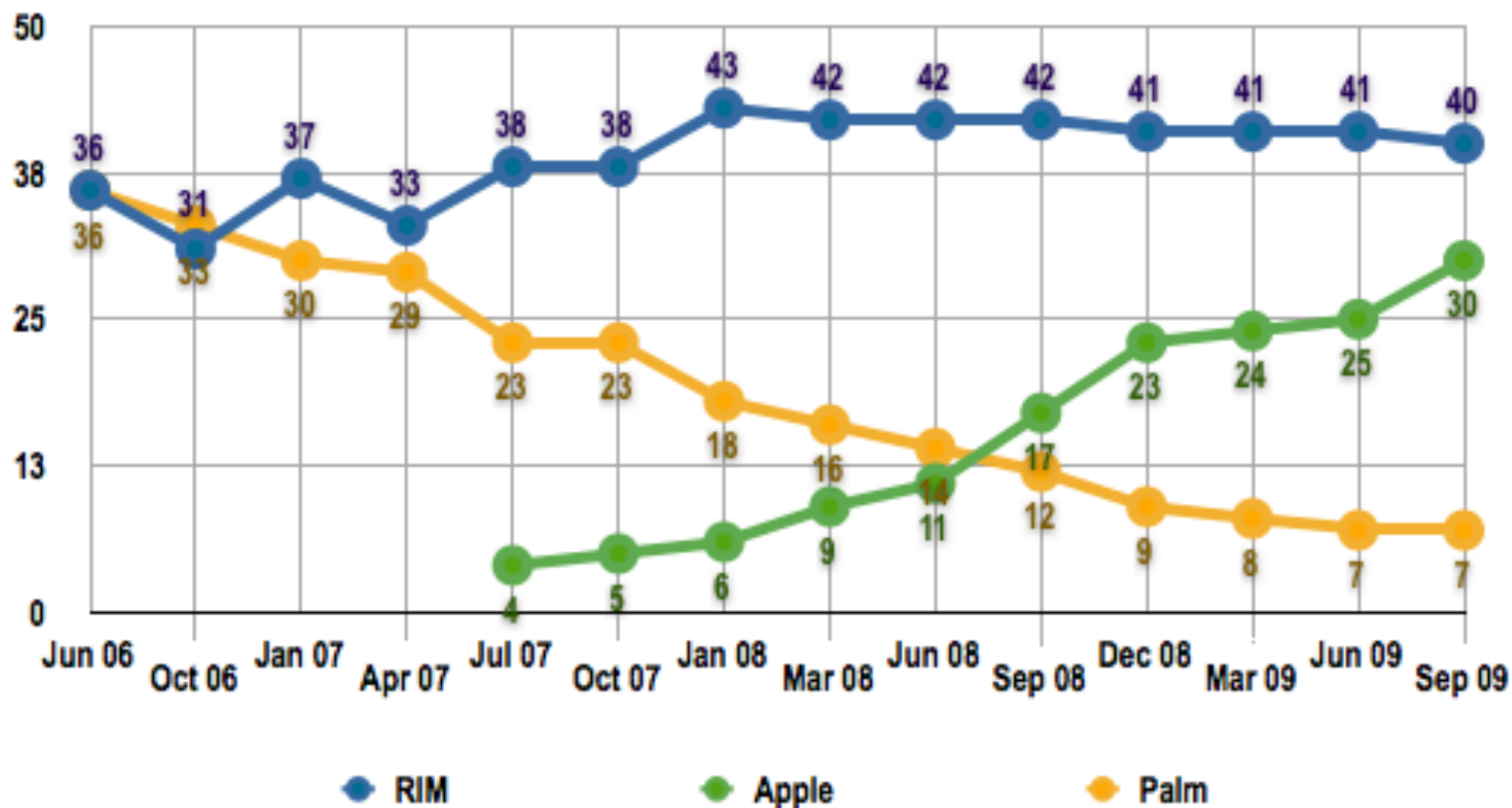
Integrazione fra i diversi sistemi di
comunicazione (sms, email) **BBHub**

Notifiche push ante litteram

Dispositivi di fascia media/alta

2.3 Evoluzione

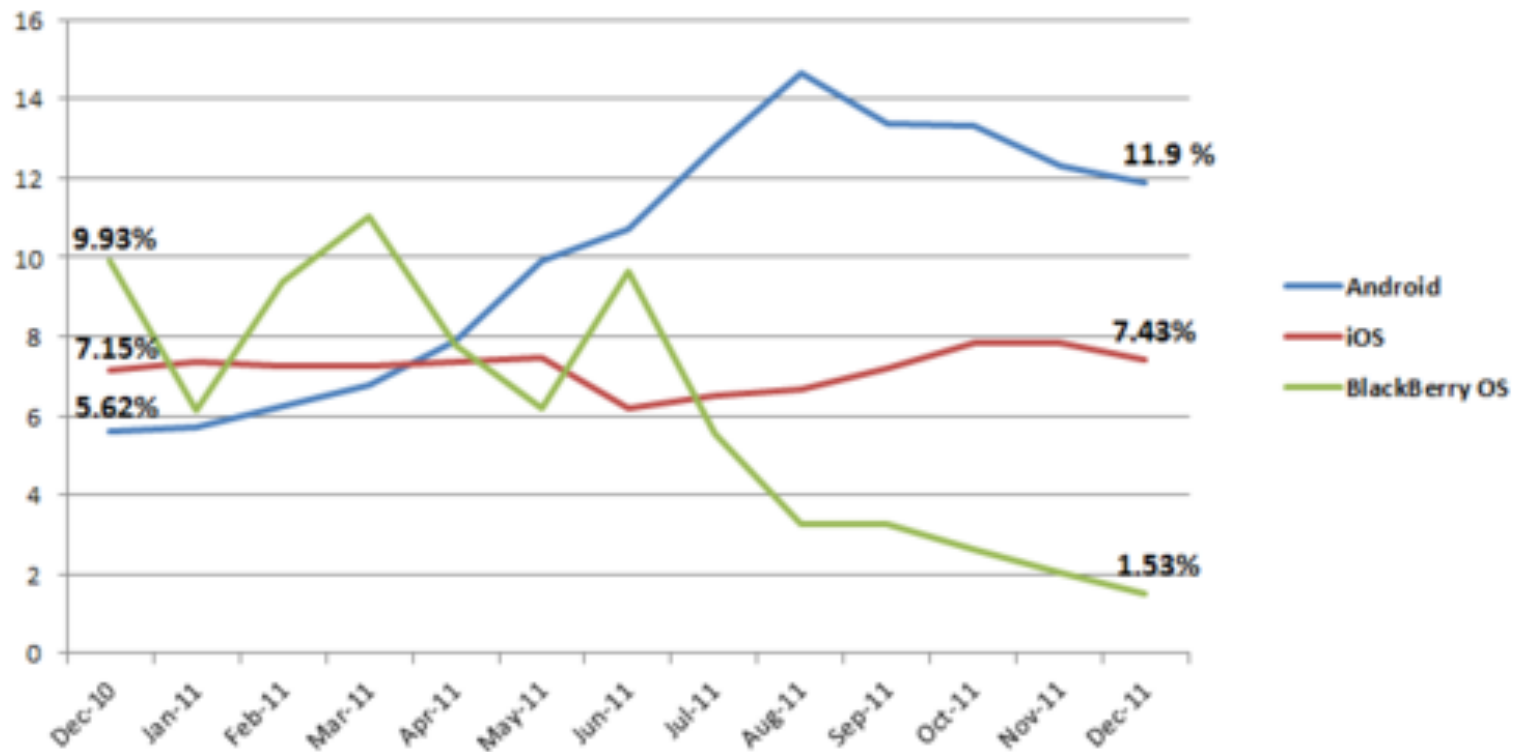
Leadership salda fino alla fine del 2009



2.3 Evoluzione

...fino alla debacle

Mobile OS Market Share in Asia



2.3 Evoluzione

HARDWARE > PHONES

Apple iPhone closing in on BlackBerry market share

Pre reversing Palm's fortunes?

◆ VINTI

BLACKBERRY: quota di mercato azzerata, l'azienda lascia il Giappone

Non è ancora ufficiale, ma secondo il quotidiano Nikkei Business Daily, **BlackBerry** (fino a poco

NOTIZIE



RIM Blackberry scivola all'1.6% del mercato USA

Blackberry 10 arriverà il 30 gennaio, ma continua a perdere market share. Apple in grande spolvero, ma soprattutto negli USA. In Europa domina Android. La classifica di Kantar WorldPanel

Il 27 novembre 2012 di Redazione  1

2.3 Evoluzione

BlackBerry 10: Re-designed, Re-engineered, and Re-invented

BlackBerry 10 Platform Launches on Two New Smartphones



MARKETWIRE

Press Release: BlackBerry – Wed, Jan 30, 2013 11:22 AM EST

2.3 BlackBerry 10

BlackBerry® Hub

BlackBerry® Flow

BlackBerry® Messenger

BlackBerry® Balance

Time Shift

BlackBerry 10 browser

BlackBerry® Remember

BlackBerry® Safeguard

The BlackBerry® World™

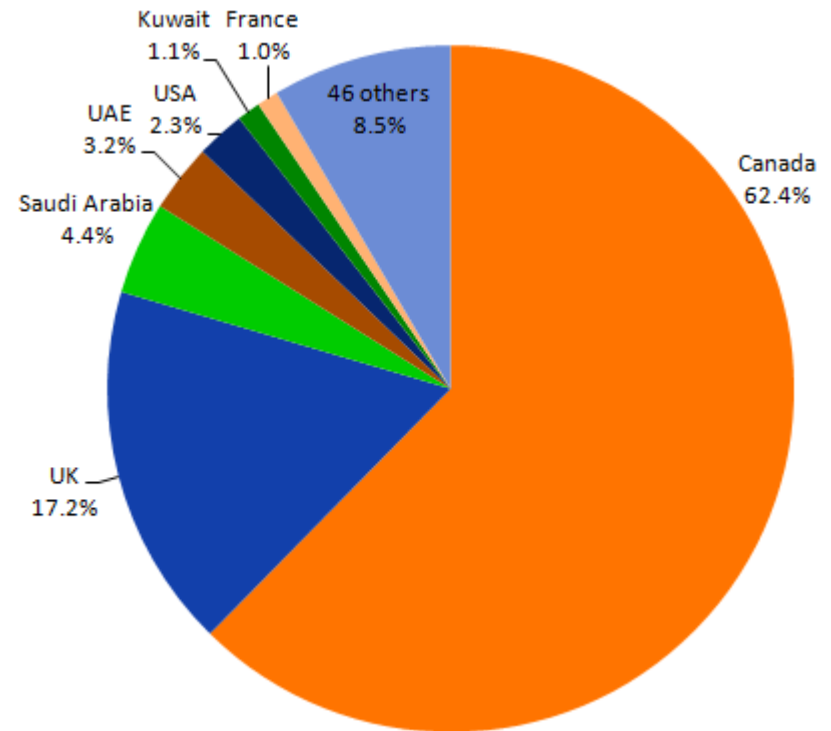


2.3 BlackBerry World

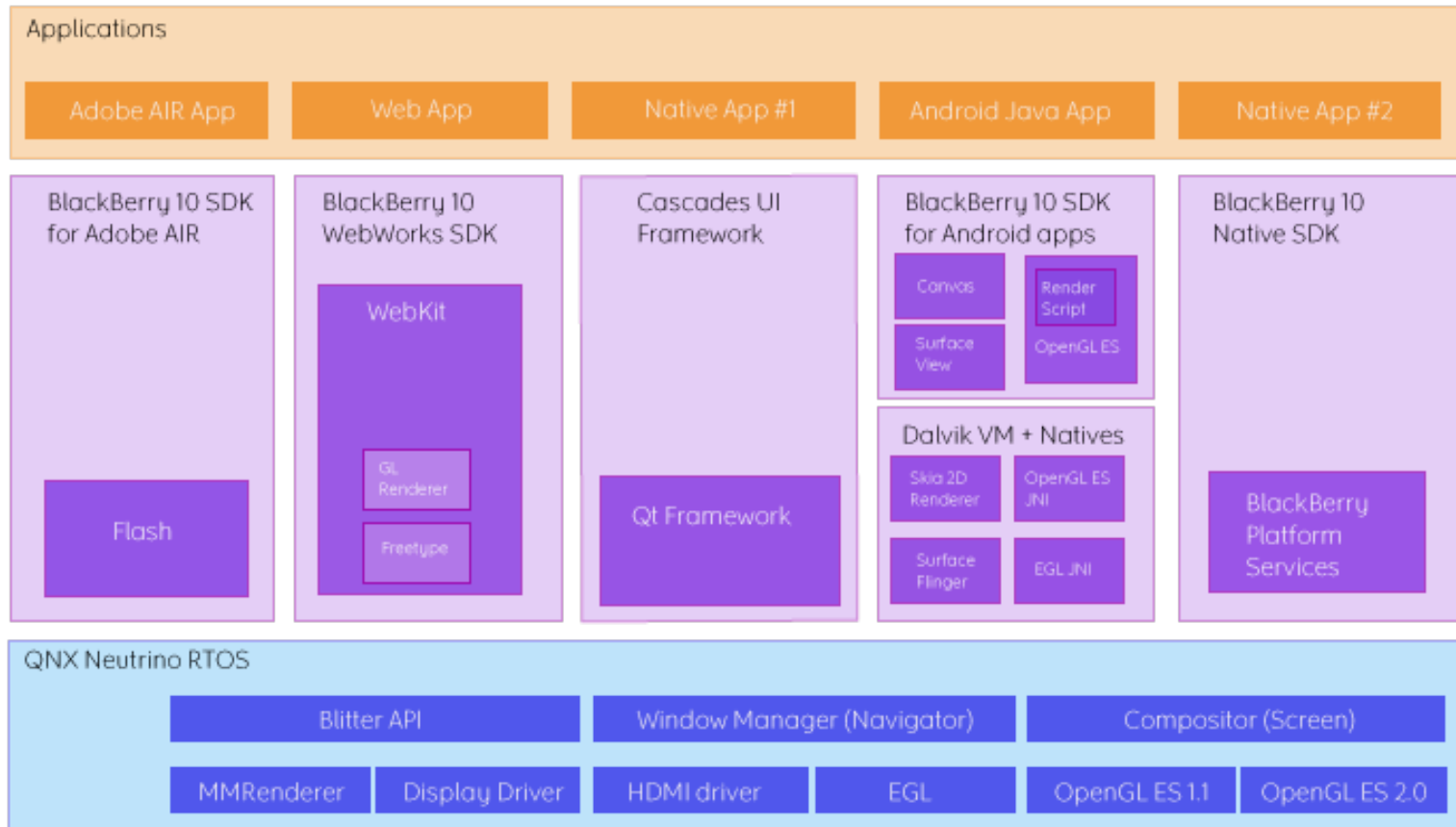
Rilasciato il primo aprile del 2009
105,000 app a settembre 2012
Disponibile in 113 paesi
Circa 6 milioni di download/day
Circa 4 miliardi di download

Nel 2012 il 13% degli sviluppatori
BB ha guadagnato più di
100.000\$

Programma 10K BlackBerry Commitment



2.3 BlackBerry 10 Software Stack



2.3 BlackBerry 10 development

Want to
create
an app?

With **BlackBerry**
there are always
options

The **BlackBerry** Experience

*Create something that fits like a glove.
Cascades will give you the BlackBerry
Experience out of the box.*

Bring what **you've got**

*We support lots of different platforms
and frameworks so that you will feel
right at home.*

OR JUMP
STRAIGHT TO IT

Cascades



Core Native

Adobe AIR

Android

Java

Platform
Partners

2.3 BlackBerry 10 Development



Cascades UI framework

C++ Application Framework

Qt + QML

QNX Momentics IDE o Cascades Eclipse plugin



JavaScript / HTML5 / CSS

WebWorks SDK



AdobeAIR

ActionScript 3.0

Adobe Flash Builder 4.7 o FDT5 IDE

2.3 BlackBerry 10 Development



Repacking di App Android

Plugin Eclipse + Online Packager + CLI Tool



Java... Java **non è supportato** in BlackBerry 10

2.3 BlackBerry Distribuzione

Publicazione gratuita

Processo di revisione dell'applicazione

Submissione al BlackBerry World in 1/2
settimane

Get it at



BlackBerry
World™

2.3 Pro / Contro



Possibile programmare su diverse piattaforme e diversi linguaggi

Da Android il TTM è vicino allo zero (dovrebbe)

Iscrizione gratuita

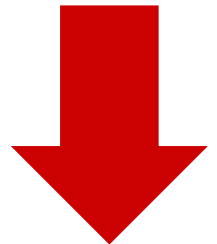
Dispositivi performanti

Ottime revenue dal market

Piattaforma che stenta a stare a galla

Target ristretto

Per una resa grafica nativa è necessario programmare in C/C++



2.3 References

<http://blog.trafficdito.com/post/16166710161/deadberrya-eulogy-for-blackberry-2012-prediction>

<http://www.forbes.com/sites/ewanspence/2013/01/30/blackberry-must-ignore-market-share>

http://www.theregister.co.uk/2009/10/28/us_consumer_smartphone_biz/

<http://finance.yahoo.com/news/blackberry-10-designed-engineered-invented-162241400.html>

<http://news.ebscer.com/2013/03/blackberry-10-users-by-country/>

<http://crackberry.com/blackberry-app-world-stats-devcon-europe>

2.4

Bada

2.4 Bada - cos'è

- Piattaforma per sistemi mobile sviluppata da Samsung
- Kernel ibrido
 - Linux
 - FreeBSD
 - RTOS



2.4 Bada - evoluzione (1)

- Primo annuncio di Samsung (novembre 2009)
- Primo smartphone Bada 1.0: S8500 Wave
 - MWC (febbraio 2010) - 1 giugno 2010
 - 1 milione in 4 settimane



2.4 Bada - evoluzione (2)

- Bada 2.0 (1 settembre 2011)
 - Samsung Wave 3 e modelli inferiori

 - 25 febbraio 2013: Samsung annuncia la fine dello sviluppo di Bada in favore di Tizen
-

2.4 Bada - diffusione

- Secondo Gartner fino al terzo trimestre 2012 gli smartphone venduti con s.o Bada sono superiori a quelli WP
- Battuta d'arresto nel quarto trimestre 2012

trimestre	market share (% smartphone)	dispositivi venduti
Q3 2012	3%	5.2M
Q2 2012	2.7%	4.1 M
Q1 2012	2.6%	3.8 M
Q4 2011	2.3%	4 M
Q3 2012	2%	2.6 M

2.4 Bada - dispositivi

- Samsung Wave S8500
 - Samsung Wave II S8530
 - Samsung Wave III S8600
 - Samsung Wave Magical
 - Samsung Wave Young
 - Samsung Wave 525 (Lite)
 - Samsung Wave 723
 - Samsung Wave 533 (Pro)
 - Samsung Wave Young
 - Samsung Wave M
 - Samsung Wave 578
 - Samsung Wave 575
-

2.4 Bada - SW stack (1)

Device
Applications

Service
Applications

Web and Flash
Applications

bada

Framework

C++ and Web frameworks that consist of an application framework and functions exported by underlying layers.

Service

Service-oriented functions that are provided by application engines and server-assisted components.

Device

Core functions of a device platform that are provided by system, security, graphics, multimedia, and communication components.

Kernel

OS kernel.

2.4 Bada - SW stack (2)

- Kernel: RTOS o Linux in base al dispositivo
 - Device: primitive offerte dal sistema operativo attivo
 - grafica
 - telefonia
 - multimedia
 - Service: funzioni rese disponibili dall'application engine
 - Contatti e messaggistica
 - RESTful WS per funzionalità server-assisted
-

2.4 Bada - SW stack (3)

- Framework:
 - C++ Framework
 - gestione ciclo di vita applicazioni
 - classe e interfacce per l'utilizzo dei componenti sottostanti
 - Web framework
 - HTML-CSS-JS
 - API specifiche per il controllo dell'UI e degli eventi
-

2.4 Bada - SDK

- Binari e librerie
 - Bada IDE
 - basato su Eclipse (C++/JS)
 - Javascript debugger
 - UI builder WYSIWYG
 - UI sequencer
 - Emulatore
 - Iniettore di eventi
 - Analizzatore di prestazioni
-

2.4 Bada - pro e contro

- Pro

- Mancanza di sviluppatori
- Forte spinta di Samsung prima dello stop

- Contro

- Market share basso e destinato al declino
 - Poca documentazione
 - API non competitive con i concorrenti
-

2.5

Tizen

2.5 Tizen - cos'è

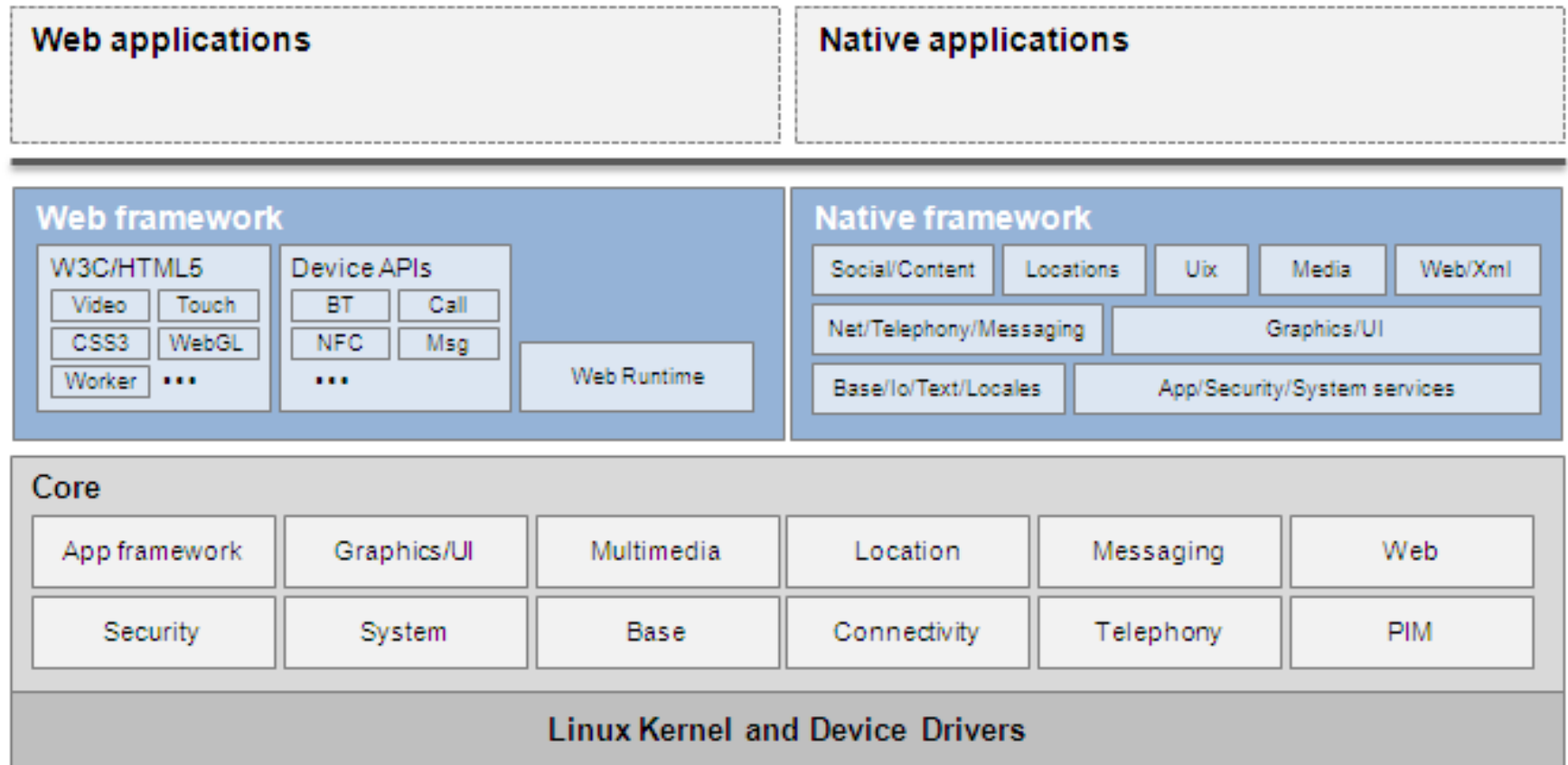
- Sistema operativo open-source per
 - smartphone/tablet/IVI device/Smart TV
- Chi c'è dietro
 - Linux Fondazione, Intel, **Samsung**
 - Automotive Grade Linux
 - Jaguar, Land Rover, Nissan, Toyota
 - Fujitsu, Nec, nVidia, Texas Instruments

TIZEN[™] 

2.5 Tizen - evoluzione

- Dapprima *conosciuto* come LiMo (settembre 2011)
 - Sistema operativo open-source per
 - smartphone/tablet/**IVI** device/Smart TV
 - Tizen 1.0 (*Larkspur*) (30 aprile 2012)
 - Automotive Grade Linux Workgroup (16 settembre 2012)
 - piattaforma di riferimento
 - Tizen 2.0 (*Magnolia*) (18 febbraio 2013)
 - Primi dispositivi Samsung tra Q3 e Q4 del 2013?
-

2.5 Tizen - SW stack



2.5 Tizen - SDK

- Binari e documentazione
 - Tizen IDE
 - Connection explore view
 - Log view
 - CSS preview view
 - JS log console view
 - Emulatore
 - Generatore di certificati
-

2.5 Tizen - Linguaggi supportati (1)

- Web Applications

- HTML5 al top (492/500)
- Temi grafici standard per coerenza visiva
- API aggiuntive specifiche per l'accesso all'hw del device:

- Bluetooth,
- NFC,
- Notifiche,
- ...



2.5 Tizen - Linguaggi supportati (2)

- Native Applications
 - C++
 - GUI builder e XML per l'UI
 - Documentata e realmente utilizzabile solo da versione 2.0
-

2.5 Tizen - pro e contro

- Pro

- Semplicità di sviluppo
 - JavaScript
- Flessibilità e prestazioni
 - C++
- Consorzio di aziende che spinge per adozione in settori non esplorati

- Contro

- Alto rischio di fallimento
 - Poca documentazione
 - Market share nullo
 - Strategia possibile solo a lungo termine
-

2.6

Web applications

2.6 Web applications - cosa sono

- Pagine web evolute che sono andate oltre all'utilizzo basico dell'HTTP e dell'HTML
 - Il browser diventa una sorta di sand-box su cui far girare il nostro codice
-

2.6 Web applications - esempi

- Una delle applicazioni stand-alone più utilizzate nel mondo è il client di posta elettronica
- Google con GMail ha interpretato al meglio il passaggio al modello web app
- Ma oltre a questo?



2.6 Web applications - esempi



LUCIDCHART
THE CANVAS FOR THE WORLD'S IDEAS



Microsoft®

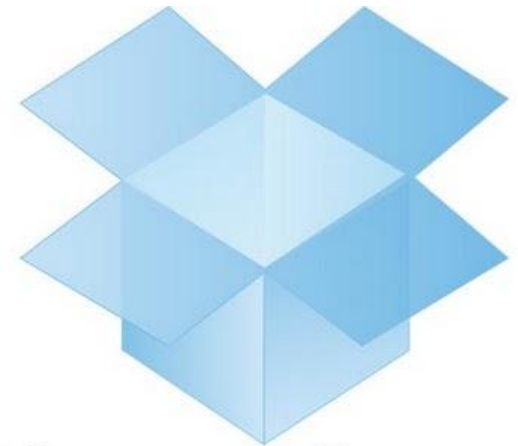
Office 365

 **Grooveshark**

Immagine fornita da marymatley
e su www.istockphoto.it



Google Drive



Dropbox

 **force.com**

2.6 Web applications - panoramica

- In principio era un client che chiedeva ad un server una pagina statica con testo ed eventualmente link ad altre pagine
 - Poi ci si rese conto che rendere le pagine dinamiche fosse cosa buona e giusta
 - Programmazione server-side tramite CGI, Perl, J2EE, ASP.NET, PHP, Ruby, Python, ...
 - Anche l'aspetto divenne importante per rendere più gradevoli le pagine
 - fogli di stile CSS
-

2.6 Web applications - panoramica

- Con uno sforzo in più le pagine vennero dotate di logica anche nel client
 - nascita di JavaScript e VBscript
 - A questo punto le pagine erano effettivamente complete
 - Ma un particolare limitava la user experience
 - lo scambio client-server richiedeva un refresh della pagina per instaurare la connessione HTTP
-

2.6 Web applications - panoramica

- Necessità di chiamate asincrone
 - nascita di XMLHttpRequest
 - Tramite delle API la pagina web può comunicare tramite browser con il server
 - *AJAX (Asynchronous JavaScript and XML)*
 - la richiesta avviene in background senza bloccare la pagina
 - il risultato della richiesta viene inviato al JS
 - JS interpreta il risultato e può decidere di aggiornare la pagina dinamicamente
-

2.6 Web applications - HTML5 (1)

- Questo è ciò che c'era fino a pochi anni fa
- Intuizione delle potenzialità del web
 - necessità di un nuovo standard, che va sotto il cappello di HTML5
- Massimo disaccoppiamento tra struttura, stile e logica



2.6 Web applications - HTML5 (2)

- Caratteristiche
 - *nuovi tag semantici (struttura/motori di ricerca)*
 - *nuovi attributi nei form*
 - **animazioni**
 - **maggiori proprietà per gli stili**
 - **trasformazioni 2D/3D**
 - **font personalizzati**
 - **storage API**
 - **database SQL locale**
 - **audio e video**
 - **grafica 2D/3D**
-

2.6 Web applications - HTML5 (3)

- Roadmap

	2012	2013	2014	2015	2016
HTML 5.0	RC	CF	R		
HTML 5.1	1D		LC	RC	R
HTML 5.2				1D	

RC = Release candidate
CR = Call for review
R = Recommendation
1D = 1st working draft
LC = Last Call

2.6 Web applications - pro e contro

- Pro

- Semplicità di sviluppo
- Portabilità estrema
- Ubiquità del servizio

- Contro

- Necessità di framework complessi per progetti di grandi dimensioni
 - Compatibilità difficile in vecchi browser (tradeoff)
 - (Visibilità?)
-

3.

Sviluppo multiplatforma

3.1

Approcci generali

3.2

PhoneGap

3.2 PhoneGap - cos'è

- Framework di sviluppo di applicazioni mobili (strano, in questo corso?) cross-platform (ahhh)
- Basato su HTML5 - CSS3 - JS
- API per accesso all'hardware del dispositivo



Phone**Gap**

3.2 PhoneGap - storia

- Originariamente sviluppato da Nitobi
 - People's Choice Award
@ O'Really Media's 2009 Web 2.0 Conference
 - Approvazione da parte di Apple del modello di sviluppo per il suo AppStore
 - Nitobi acquisita da Adobe (ottobre 2011)
 - Donazione e nascita di Apache Cordova per il codice di base
 - PhoneGap distribuzione di proprietà di Adobe
 - PhoneGapBuild: creazione in cloud degli eseguibili
-

3.2 Diffusione nel mercato

- Nitobi lancia PhoneGap 1.0 il 29 luglio 2011
 - Un successo: oltre 600000 downloads
 - 44000 downloads mensili
 - Adobe lancia PhoneGap 2.0 il 20 luglio 2012
 - 100000 downloads mensili
 - PhoneGap days
 - US: 20 luglio 2012, Portland
 - EU: 14 settembre 2012, Amsterdam
 - Adobe lancia PhoneGap 2.5 il 28 febbraio 2013
-

3.2 Dispositivi supportati

- iOS
- Android
- BlackBerry 5/6
- WebOS
- Windows Phone 7/8
- Windows 8
- Symbian
- Bada/Tizen

Alcune limitazioni nelle API dovute alle carenze hardware dei diversi dispositivi

3.2 Esperienza d'uso

- Personalizzazione
 - totale
 - l'utilizzo di HTML5-CSS3 ci permette di creare un UI personalizzata senza troppi sforzi e con la certezza che sia la stessa in tutte le piattaforme
-

3.2 Esperienza d'uso

- Coerenza grafica

"Non è tutto oro quello che luccica"

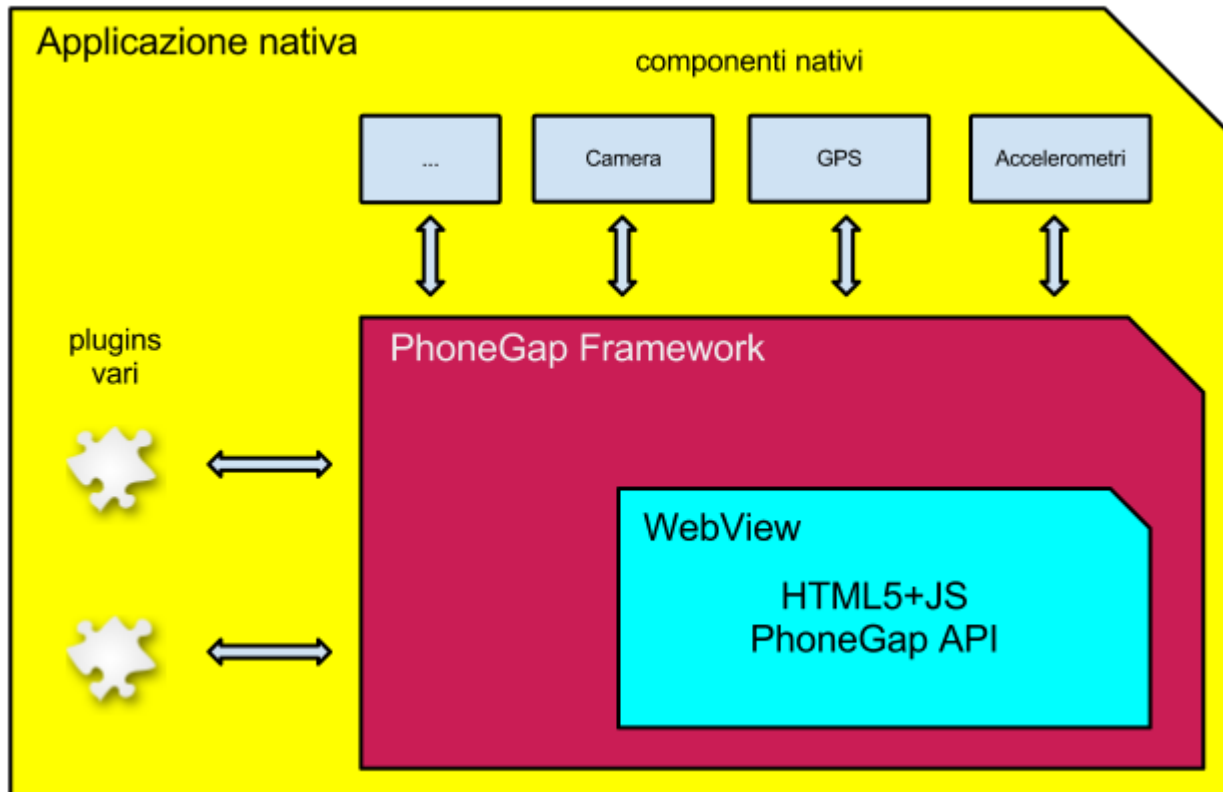
- Look and feel dell'applicazione non può essere coerente con tutte le piattaforme
 - alla stessa maniera l'integrazione con il sistema
 - Sensazione generale di avere un'applicazione non al top
-

3.2 Esperienza d'uso

- Flessibilità
 - Disponibili plugin per utilizzare feature specifiche di alcune piattaforme (e.g. Android service)
 - a questo livello si perde però il grande vantaggio dello sviluppo web unificato
 - Prestazioni non al massimo a causa di JS
-

3.2 Sviluppo: SW stack

- Essenzialmente un wrapper di WebApp



3.2 Sviluppo: SW stack

- Binding API native - JS

- Accelerometer
 - Camera
 - Capture
 - Compass
 - Connection
 - Contacts
 - Device
 - Events
 - File
 - Geolocation
 - Globalization
 - InAppBrowser
 - Media
 - Notification
 - Splashscreen
 - Storage
-

3.2 Sviluppo: SDK/IDE

- Necessità di utilizzo degli IDE appropriati per ogni piattaforma
 - Tools per la creazione automatica di progetti base per ogni piattaforma
 - Vincoli sugli strumenti HW
 - PhonegapBuild di Adobe
 - sfrutta la potenza del cloud per aggirare tali limiti
 - modello commerciale "a servizio" ibrido: gratuito per applicazioni open-source, a partire da 9.99\$ per applicazioni private
-

3.2 Sviluppo: linguaggi

- Contenuti -> HTML5
 - Stili -> CSS3
 - Logica -> jS

 - Possibilità di utilizzare API platform-specific per usi particolari
 - perdita di portabilità
-

3.2 Sviluppo: marketplaces

- Applicazioni PhoneGap pacchettizzate come app native
 - Firma dei pacchetti tramite tool
 - Pacchetto pronto per l'upload sui market
 - anche Apple che prevede maggiori restrizioni per l'accesso al market ha approvato il metodo PhoneGap
-

3.2 Pro e contro

- Pro

- Portabilità totale
- Semplicità di sviluppo
- Curva di apprendimento
- UI completamente personalizzabile

- Contro

- Scarsa integrazione
 - Reattività
 - Supporto caratteristiche avanzate
 - UI inconsistente
-

3.3

Xamarin (portable .NET)

3.3 Microsoft .NET

Software framework nato nel 2002:

- Runtime d'esecuzione (CLI/CLR)
Gestione memoria, sicurezza, typing.
- Librerie di base (Base Class Library)

Versione: **4.5**

Sfruttato su molti stack software, ma principalmente nel mondo MS.

(Windows, WP, IIS, SQL Server, ASP.NET, VS...)



3.3 Mono

CLI e C# sono standard ECMA.
Parte dei framework basati su .NET sono stati rilasciati sotto licenze open-source.

Mono: implementazione open, per Windows, Mac OS X, Linux...



Sviluppato da Novell → **Xamarin.**

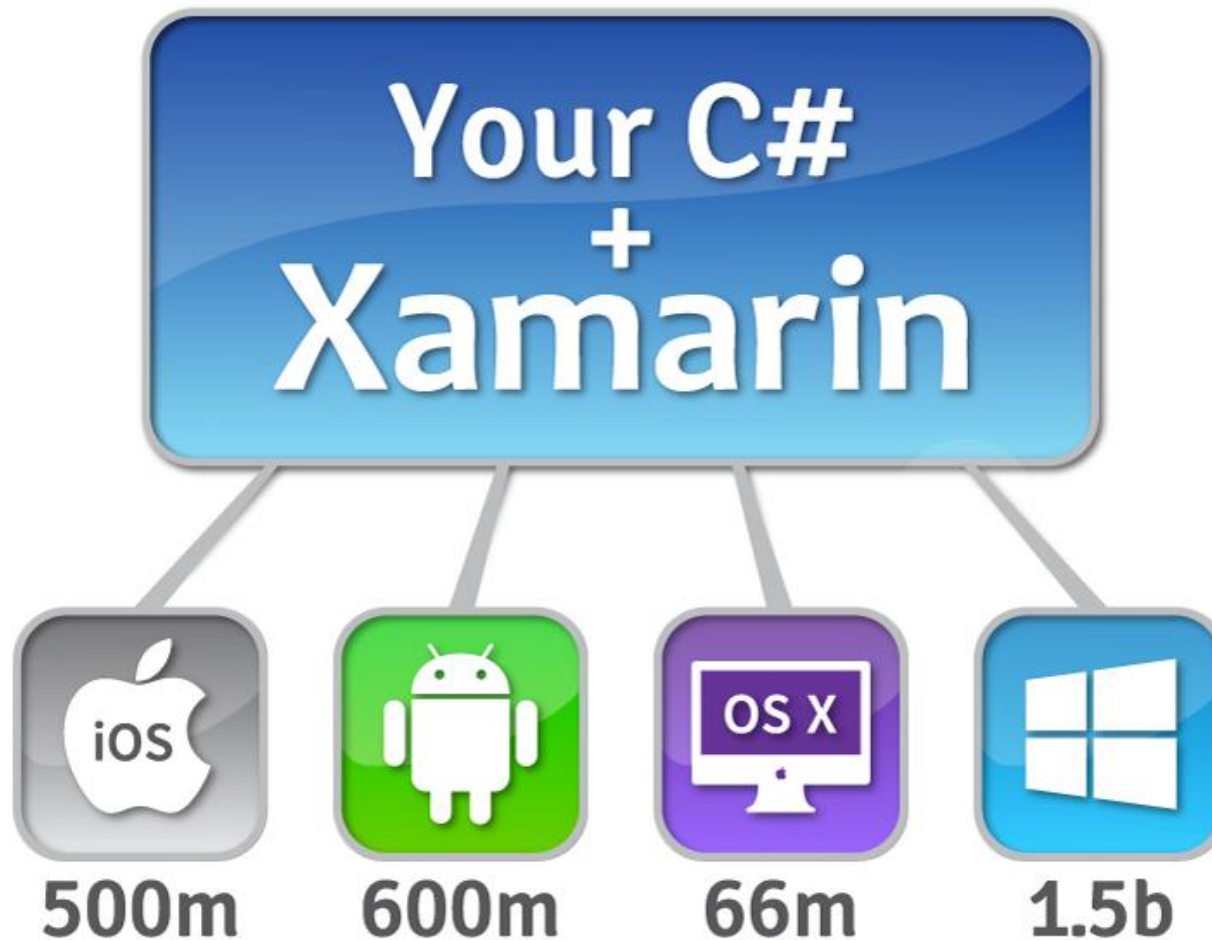
3.3 Xamarin

Continua lo sviluppo di Mono, MonoDevelop e un set di framework per lo sviluppo multiplatforma (licenza commerciale).

Xamarin

- Gratuito: progetti minori (limite bytesize),
 - **Indie** (299\$): privati,
 - **Business** (999\$): con supporto.
-

3.3 .NET everywhere



Ma **non** è
*"write once,
run everywhere"*

Non si ferma al
minimo comune
denominatore:
permette di sfruttare
ogni singola
piattaforma.

3.3 Come funziona

Bytecode CIL viene post-processato...



...e ne esce un'applicazione iOS/Android/WP.

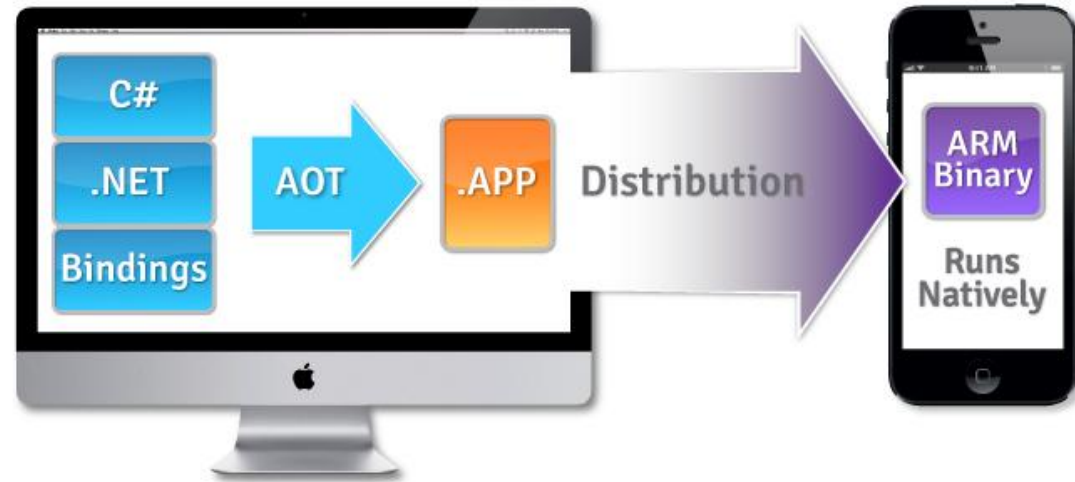
Pari ad un'app nativa.

Ma con l'aggiunta del runtime Mono.

3.3 Come funziona (2)

iOS: full
binary AOT
compilation

iOS



Android:
AOT
conversion
(JIT runtime)

Android



3.3 Vantaggi

- Portabilità di vasta quantità del codice
Ma richiede un'architettura software che lo prevede e sia incentrata sulla portabilità.
 - Utilizzo di linguaggi avanzati (C#, F#, etc.)
 - Utilizzo delle capacità di base di .NET.
 - Ma anche sfruttamento delle capacità native delle piattaforme.
-

3.3 Beautiful code: LINQ

Query dichiarative.

```
from p in Table<Person> ()
  where p.ID == id
  let orders = from o in Table<Order>()
  where o.person = id
  select {
    Id = id,
    Orders = orders.ToList()
  };
```

3.3 Beautiful code: easy XML

Parsing di XML (o JSON, o altro...) con facilità.

```
var doc = XmlDocument.Load(url);
foreach(var item in doc.Root.Elements())
{
    var text = item.Value;
}
```

3.3 Beautiful code: lambda style

Eventi e delegati sono concetti di prim'ordine nel runtime. Nessuna necessità per classi anonime e altri costrutti verbosi (Android).

```
button.TouchUpInside += (s, o) => {  
    message.Text = "Hello!";  
};
```

3.3 Beautiful code: parallelismo

Supporto al *multi-threading* nel linguaggio:

```
from item in items.AsParallel()  
    let result = QualcosaDiLento(item)  
    select result;
```

3.3 Beautiful code: async

Chiamate asincrone supportate direttamente dal linguaggio e dal runtime:

```
public async bool Do(){  
    var doc = await CaricaDocumento(url);  
    var dom = await Parse(doc);  
    var result = await Controlla(dom);  
    return result;  
}
```

3.3 Svantaggi

- **Runtime aggiuntivo:** +2.5 MB e lieve impatto sui tempi di avvio.
 - A volte l'**astrazione** non è perfetta: si intravedono oggetti NSObject o jobject.
 - Nessuna **generazione dinamica** di codice: iOS non lo permette. Per cui, no *Reflection.Emit*, no generazione di classi a runtime, no *Dynamic...*
-

3.3 Architettura software

Libreria comune di "*shared code*".

Basato su versione *portable* di .NET, senza feature specifiche.

Codice diverso per ogni piattaforma.

Si appoggia sul codice in comune, ma offre UI

nativa e (opzionalmente) feature aggiuntive/specifiche.



3.3 Portable .NET

Definizione labile:
quale sottoinsieme di
funzionalità si usa?

Xamarin utilizza ha
librerie comuni molto
piccole, con ampia
compatibilità.

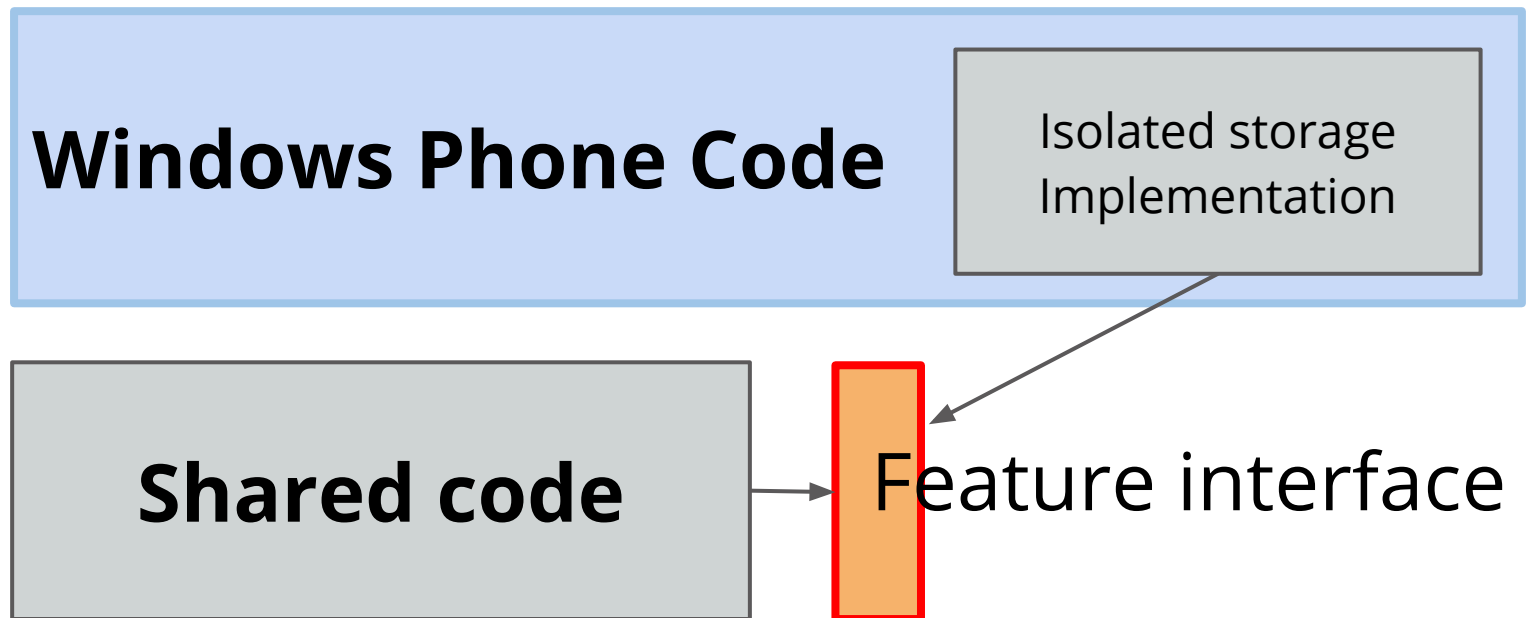


3.3 Cosa non è portable?

- Interfaccia grafica.
 - Capacità hardware.
 - Funzioni HTTP avanzate.
HttpWebRequest base, NSURLConnection su iOS...
 - Feature multimediali.
 - Accesso al Filesystem!
Ogni piattaforma è diversa.
-

3.3 Platform injection

Le feature non portabili vanno *iniettate* nel codice condiviso.



4.

Licenze e application store

4.1 Costi

- **Google Play:** una tantum 25\$.
 - **iOS App Store:** 99\$/anno.
 - **Windows Phone Store:** 99\$/anno.
 - **Windows Store:** 49\$/anno (per privati).
-

4.2 Revenue sharing

In generale tutti seguono la formula:

30% ad Apple/Microsoft/Google...

70% allo sviluppatore

Su ***tutti*** i ricavi: anche su transazioni *in-app*.

4.3 Open-source e licenza

Applicazioni scritte con Apple SDK possono essere rilasciate *solo* su App Store.

GPL e altre licenze sono illegali su App Store e Windows Phone Store.

Android molto più liberale.

5.

Android

5. Android - cos'è

- Sistema operativo open-source inizialmente pensato per dispositivi mobili e poi evolutosi
- Attualmente leader del mercato
 - e lo sarà per molto tempo, secondo le stime



5. Android - storia

- Android Inc. fondata nel 2003 da Andy Rubin con l'obiettivo di creare

"smarter mobile devices that are more aware of its owner's location and preferences"

- Inizialmente sviluppato in segreto e già con problemi di budget nel primo anno di vita
- Passa poco tempo e Google acquisisce la società (17 agosto 2005)

- Rubin comincia a lavorare con kernel Linux



5. Android - storia (2)

- L'OHA (*Open Handset Alliance*) si svelò al mondo il 5 novembre 2007
 - vi facevano parte diversi operatori di telefonia, di dispositivi e di SoC
 - ufficialmente il consorzio sviluppa standard aperti da utilizzare nel mondo mobile
- Quello stesso giorno viene annunciato il progetto Android
 - Framework per dispositivi mobili
 - Linux kernel 2.6



5. Android - storia (3)

- L'anno successivo viene presentato il primo prodotto, l'HTC Dream (22 ottobre 2008)
 - Android 1.0 "Apple Pie"
 - Android Market
 - Notifiche
 - Google Apps
 - Media Player
 - Camera
 - WiFi
 - Bluetooth



5. Android - storia (4)

- Da quel momento lo sviluppo è prorompente
- Prime versioni con una vera diffusione sul mercato
 - 1.5 "*Cupcake*" (13 aprile 2009)
 - 1.6 "*Donut*" (16 settembre 2009)
- Ogni 6 mesi circa una nuova versione (fin'ora)
 - attualmente 4.2 "*Jelly Bean*"



5. Android - storia (5)



5. Android - storia (6)

- Dal 2010 Google ha lanciato una sua serie di dispositivi Nexus sviluppati da partner e con una esperienza d'uso Android pura

- HTC One
- Samsung Nexus S
- Samsung Galaxy Nexus
- Asus Nexus 7



- LG Nexus 4
- Samsung Nexus 10

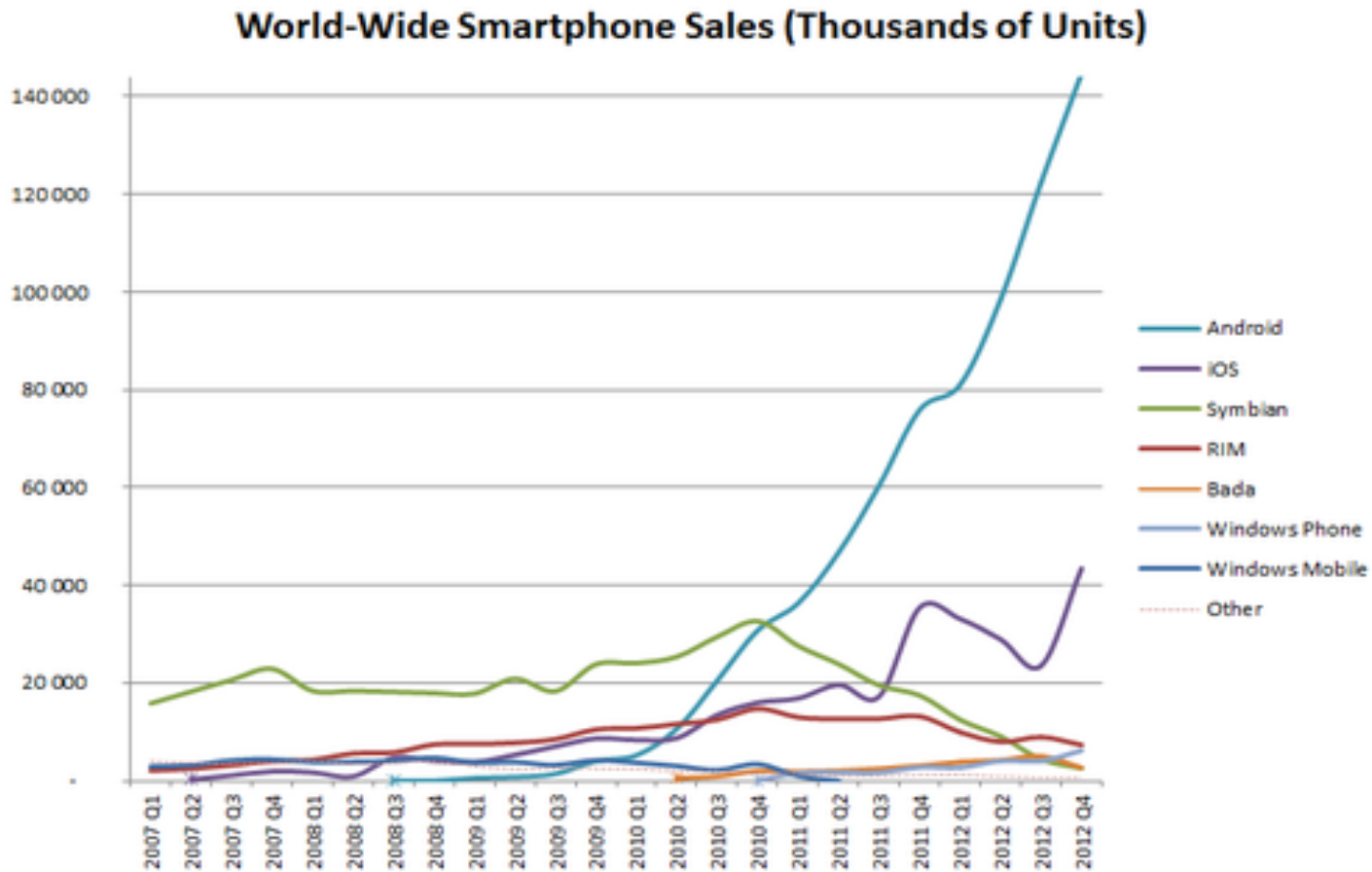


5. Android - diffusione

- **Esclusi Apple** (per ovvi motivi) e **Nokia** (a stretto contatto con Microsoft), tutti i principali produttori di smartphone si affidano ad Android, dalla fascia bassa ai modelli di punta
 - Ma nel mercato il riscontro sarà stato effettivamente così positivo?
-

5. Android - diffusione

- A quanto pare sì



5. Android - diffusione

- Il settore tablet è rimasto appannaggio Apple fino al Q2 2012 (circa)
 - secondo le previsioni della società IDC nel 2013 i tablet Android saranno il 48.8% del mercato contro il 46% di tablet Apple



5. Android - prospettive di crescita

- Rimanendo nel settore tablet, sempre secondo IDC, le previsioni al 2017 vedono decrementi sia per iOS che per Android
 - a vantaggio di tablet Windows

Sistema operativo	Quote di mercato 2013	Quote di mercato 2017
Android	48.8%	46.0%
iOS	46.0%	43.5%
Windows	2.8%	7.4%
Windows RT	1.9%	2.7%
Altri	0.6%	0.4%
Totale	100.0%	100.0%

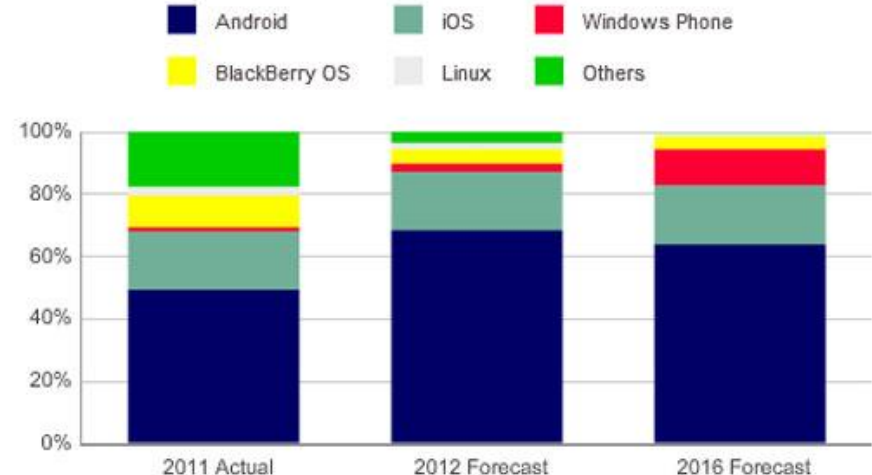
5. Android - prospettive di crescita

- Passando al settore smartphone, sempre secondo IDC, le previsioni al 2016 vedono lo stesso scenario

- Android: 68.3% -> 63.8%
- iOS: 18.8% -> 19.1%
- BB: 4.7% -> 4.1%
- WP: 2.6% -> 11.4%
- Linux: 2.0% -> 1.5%
- Altri: 3.6% -> 0.1%



Worldwide Smartphone Market
Forecast by OS, Unit Shipments, 3Q 2012



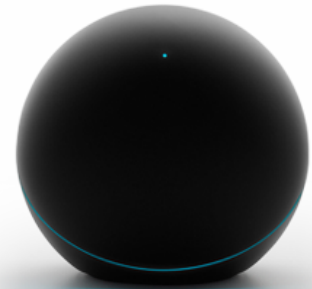
5. Android - non solo smartphone

- La flessibilità di Android concede l'utilizzo in dispositivi molto diversi da tablet/smartphone
 - questo è un vantaggio sia per i produttori, sia per gli sviluppatori che hanno a disposizione settori sempre freschi



5. Android - Player evoluti

- Samsung Galaxy Player
 - simile ad uno smartphone ma con risalto delle qualità sonore
- Google Nexus Q
 - dispositivo per lo streaming e la riproduzione di flussi multimediali
 - 299\$
 - Top 10 "fails" del 2012 (CNN)



5. Android - Smartwatch

- Motorola Motoactv
 - uno di primi smartwatch con funzionalità specifiche per il fitness e lo svago
 - il display ridotto hanno limitato molto l'usabilità
 - pulsanti rapidi ai lati non sono riusciti a migliorare la situazione
- Apple ha in cantiere un suo smartwatch



5. Android - Fotocamere

- Samsung Galaxy Camera
 - CCD 16 megapixel migliore rispetto agli smartphone
 - Zoom ottimo 21x
 - Connettività completa e GPS
 - Software specifico per l'organizzazione di foto con relativa condivisione
 - Potrebbe essere il primo di un settore forse nuovo



5. Android - TV set

- Google TV

- distribuzione personalizzata su base Honeycomb
- diversi produttori
 - Sony
 - Logitech
 - ...



- Dongle HDMI

- produttori cinesi sfruttano la flessibilità di Android per creare prodotti a basso costo ma dalle alte prestazioni
 - < 100\$
 - usabilità?



5. Android - Netbook?

- Fin'ora i Chromebook hanno utilizzato il s.o. ChromeOS ma una statua al Googleplex ed alcune indiscrezioni lasciano presagire una possibile fusione con Android
 - "il padre di Android, Andy Rubin, lascia il comando per seguire altri compiti in Google" (13 marzo 2013)
 - al suo posto subentra Sundai Pichai, già responsabile Chrome e ChromeOS



5. Android - finisce qui?

- Questi ultimi device sono tentativi molto timidi di portare Android oltre il classico settore degli smartphone
 - Anche il tentativo di Google con il Nexus Q lo è stato
 - Ma forse perché in Google si era già al lavoro su un progetto che potrebbe rivelarsi sconvolgente
 - è stato annunciato e presentato al Google I/O 2012
 - i primi kit di sviluppo verranno consegnati quest'anno agli sviluppatori che sono riusciti a prenotare i pochi posti disponibili al prezzo di 1500\$
-

5.1 Google Glass

Stiamo parlando dei Google Glass, dispositivo indossabile con obiettivo la realtà aumentata

Per noi sono molto interessanti perché pare proprio che il sistema operativo sia su base Android

GLASS



5.1 Google Glass - HCI

Il sistema di controllo principale mostrato nei primi video era tramite linguaggio naturale

Durante la conferenza *South by Southwest (SXSW)* ancora in corso sono state presentate interazioni più avanzate

- tramite il movimento dell'occhio si potrà per esempio accendere lo schermo
 - tramite una leggera inclinazione della testa navigare tra le schermate
 - si attendono maggiori dettagli al Google I/O 2013
-

5.1 Google Glass - "Ok Glass"

Alcune funzionalità estrapolate dai vari video

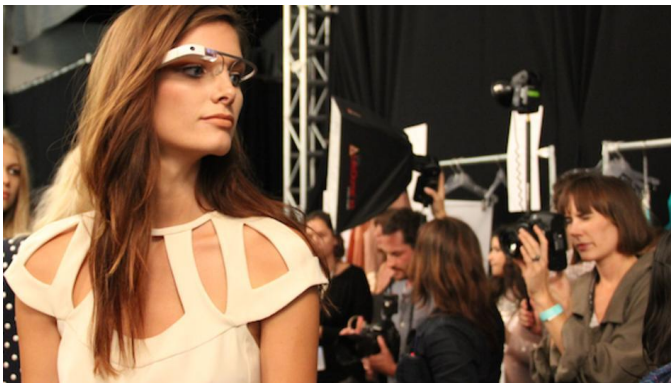
- foto e video
- Google Now
- Google+ Hangout
- ricerca
- traduzione
- navigazione
- messaggi
- meteo
- dettagli voli aerei



5.1 Google Glass - Apps

Alcune applicazioni mostrate al SXSW

- Gmail
- Evernote
- Skitch
- New York Times
- Path



5.1 Google Glass - Conclusione

Forse un video vale più di mille parole



End.
