Laboratorio di programmazione di sistemi mobili e tablet

Prefazione

Marco Ronchetti
Università degli Studi di Trento

Intro to the course

2 teachers (Marco Ronchetti - Giuseppe Riccardi)

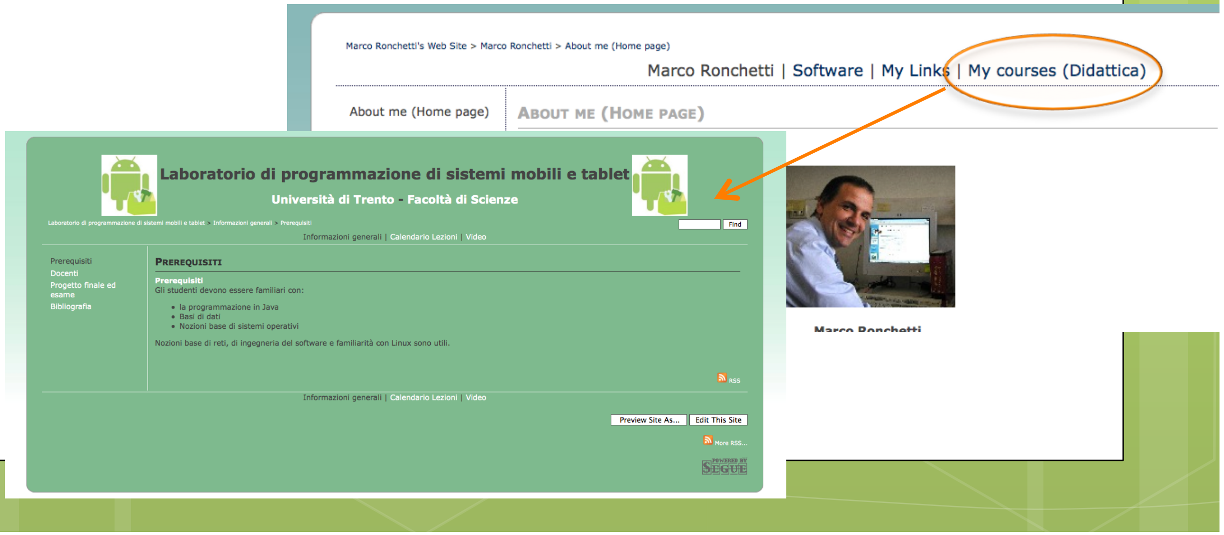
2 teaching assistants (Carlo Menapace – Arindam Ghosh)

Some guest teachers

Final project

web site: google for "marco ronchetti", go to "My courses"

Videos are available


Marco Ronchetti's Web Site > Marco Ronchetti > About me (Home page)
Marco Ronchetti | Software | My Links | My courses (Didattica)
About me (Home page)
About me (Home page)
Marco Ronchetti
Laboratorio di programmazione di sistemi mobili e tablet
Università di Trento - Facoltà di Scienze
Laboratorio di programmazione di sistemi mobili e tablet > Informazioni generali > Prerequisiti
Find
Informazioni generali | Calendario Lezioni | Video
Prerequisiti
Docenti
Progetto finale ed esame
Bibliografia
RSS
Informazioni generali | Calendario Lezioni | Video
Preview Site As...
Edit This Site
More RSS...
Powered by Segue

Prerequisiti

Prerequisiti

Gli studenti devono essere familiari con:

- la programmazione in Java
- Basi di dati
- Nozioni base di sistemi operativi

Nozioni base di reti, di ingegneria del software e familiarità con Linux sono utili.

Why is mobile programming different?

- Screen: from small phones to large TV sets
- OS version (multiple APK)
- Scarce resources (memory, disk)
- Unreliable and mutable connectivity (GSM, WiFi)
- Data transfer: costly, slow, high latency
- Battery
- Priorities (what if a phone call comes in?)
- User interaction (no kbd, gestures…)
- Devices (accelerometer, GPS, camera, audio, mic)
- Speech APIs
- Inter-app communication
- Security threats
- Development model (cross compilation)
- Distribution model (store)

Design philosophy

Applications should be:

- – Fast
  - In spite of the constraints: < 200 MB RAM, slow processor
- – Responsive
  - Apps must respond to user actions within 5 seconds
- – Secure
  - Apps declare permissions in manifest
- – Seamless
  - Usability is key, persist data, suspend services
  - The OS may kill processes in background as needed

History and context

Marco Ronchetti
Università degli Studi di Trento

Moore’s law

The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.

The period often quoted as "18 months" is due to David House, an Intel executive, who predicted that period for a doubling in chip performance (being a combination of the effect of more transistors and them being faster).

Quoted from wikipedia

Moore’s law

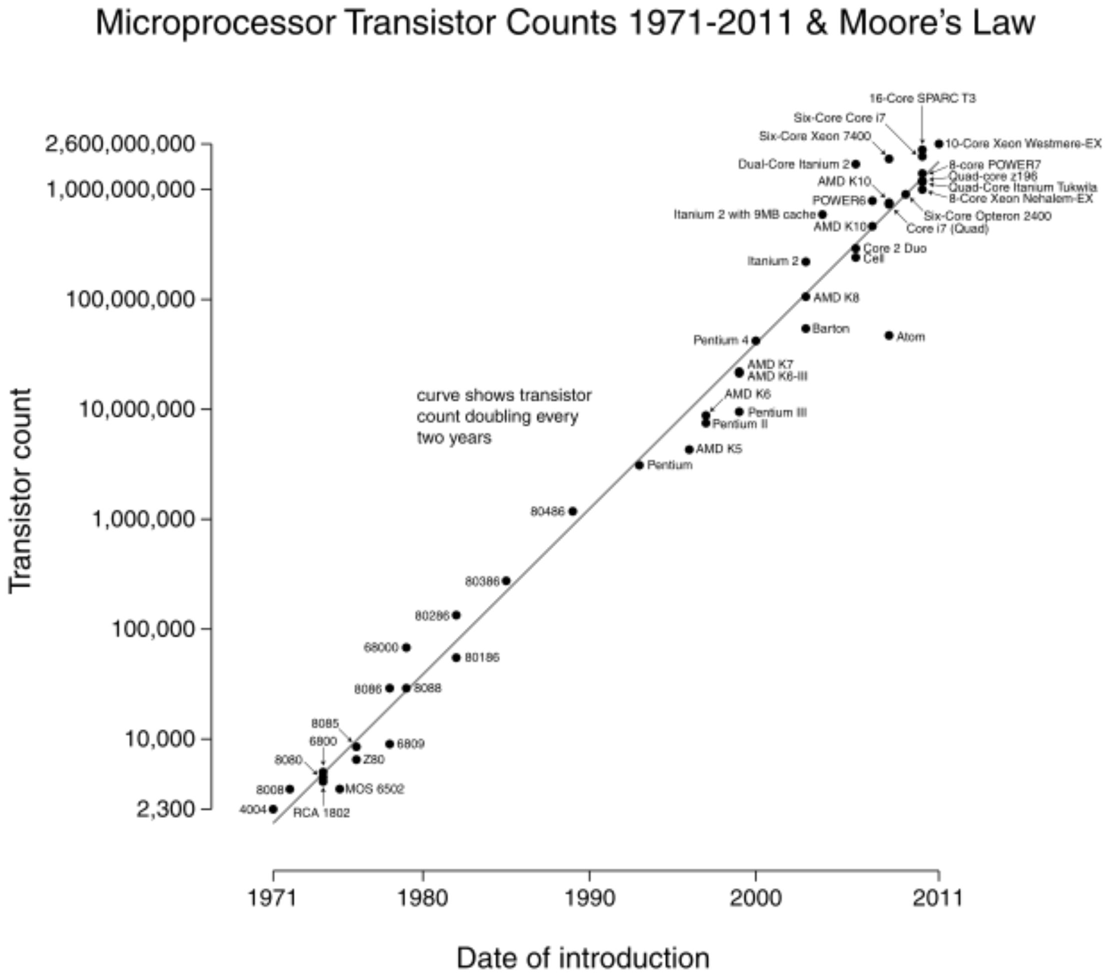


Image from wikimedia

NAND scaling timeline

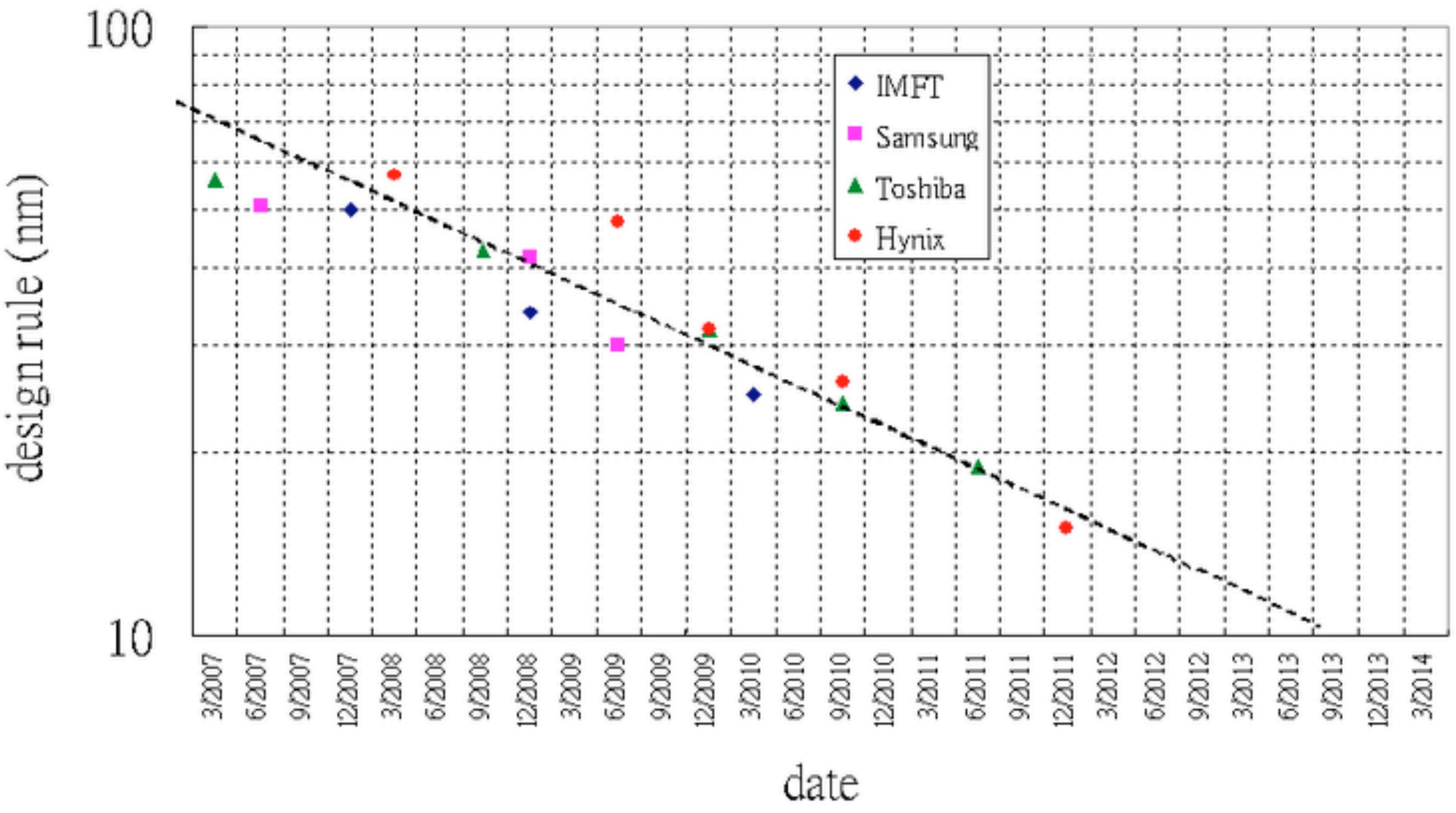


Image from wikimedia

Hard disk capacity

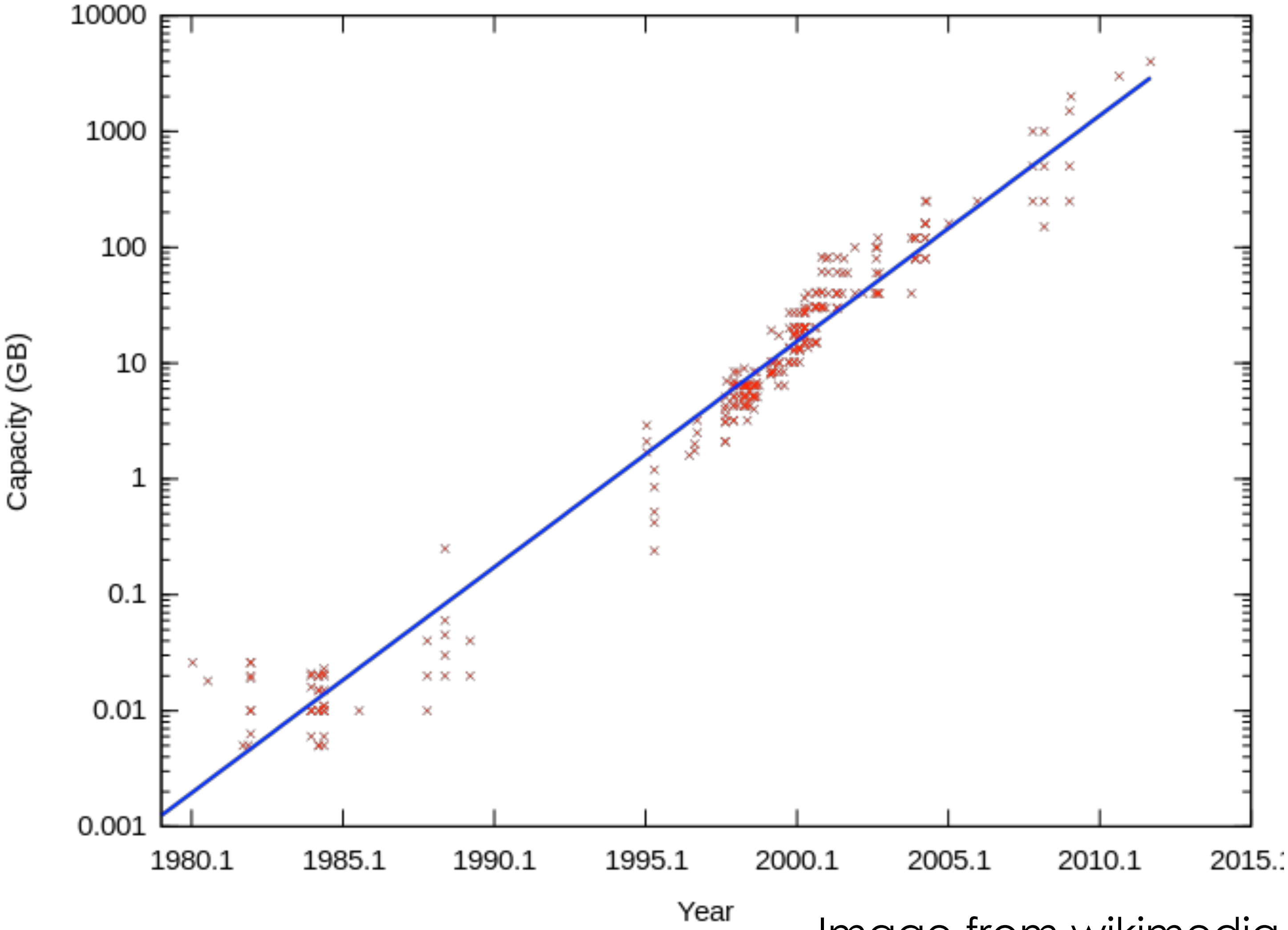
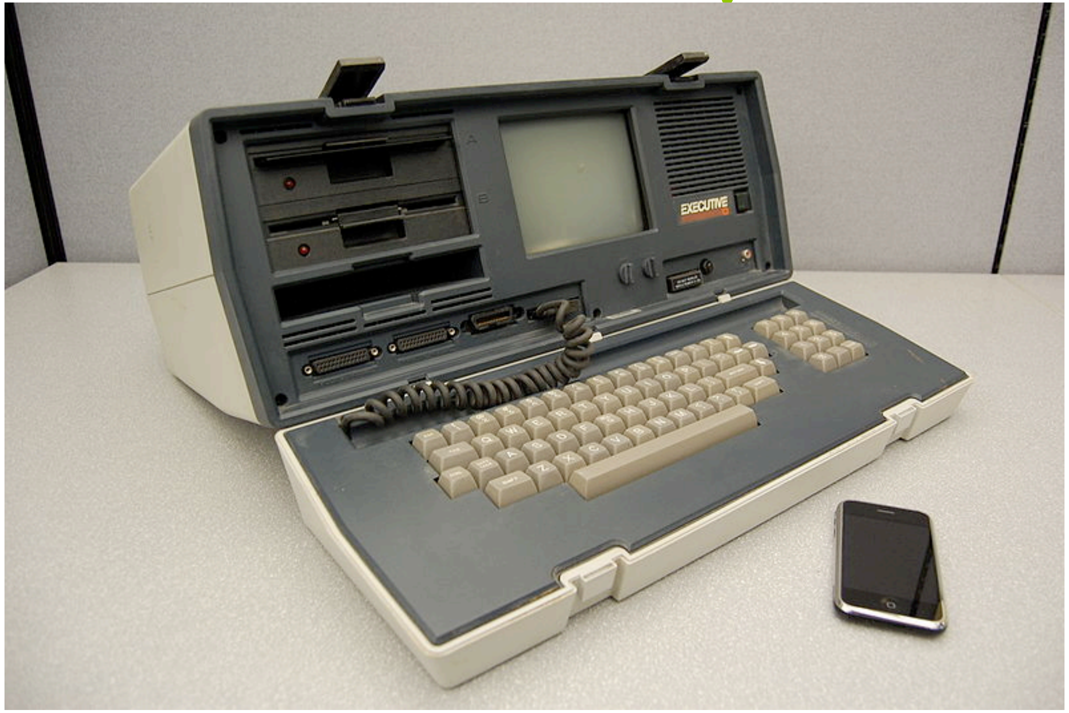
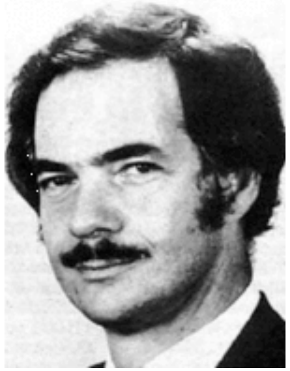


Image from wikimedia

The mobile computer





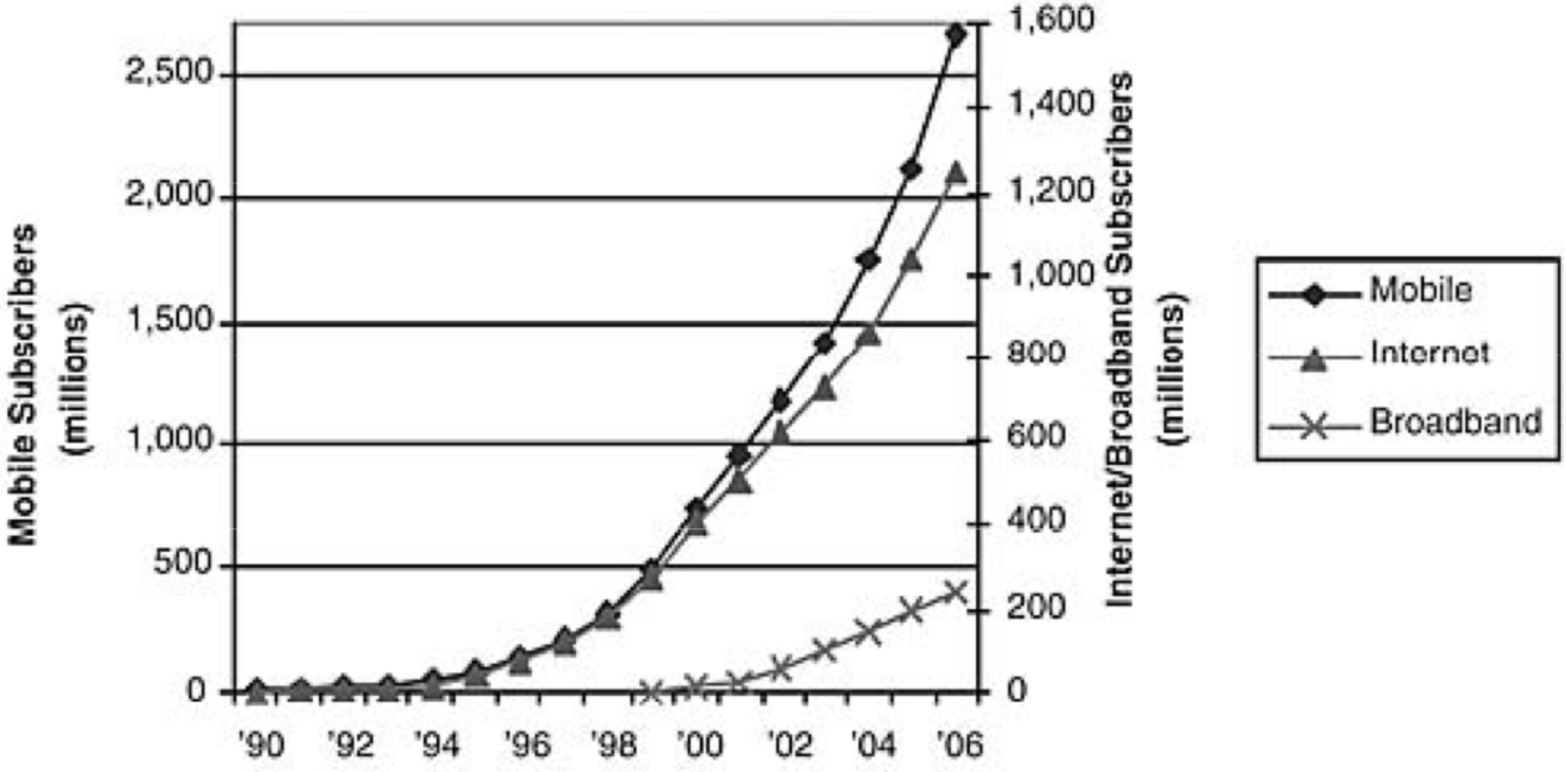
Adam Osborn

An Osborne Executive (early 80s) an an iPhone.

- 13,050 g / 135g = 100 times heavier [1]
- 4MHz / 412 Mhz = 100 times slower
- $2500 / $200-300 = 10 times more expensive
- (52cm x 23cm x 33cm)/(115mm x 61mm x 11.6mm)
- = 485 times as large (volume)

Images from wikimedia

Growth of mobile device diffusion



Smartphones

IBM Simon: concept product (1992), sold 1993

- mobile phone
- calendar,
- address book
- world clock
- calculator,
- note pad
- e-mail client
- send and receive faxes
- games.

touchscreen

Smartphones etc.

PDA (Personal digital assistant)

Psion, 1986 - Apple Newton, 1992 – Palm 1992

Portable Media Player

SaeHan/Eiger MPMan, 1997 - Apple iPod 2001

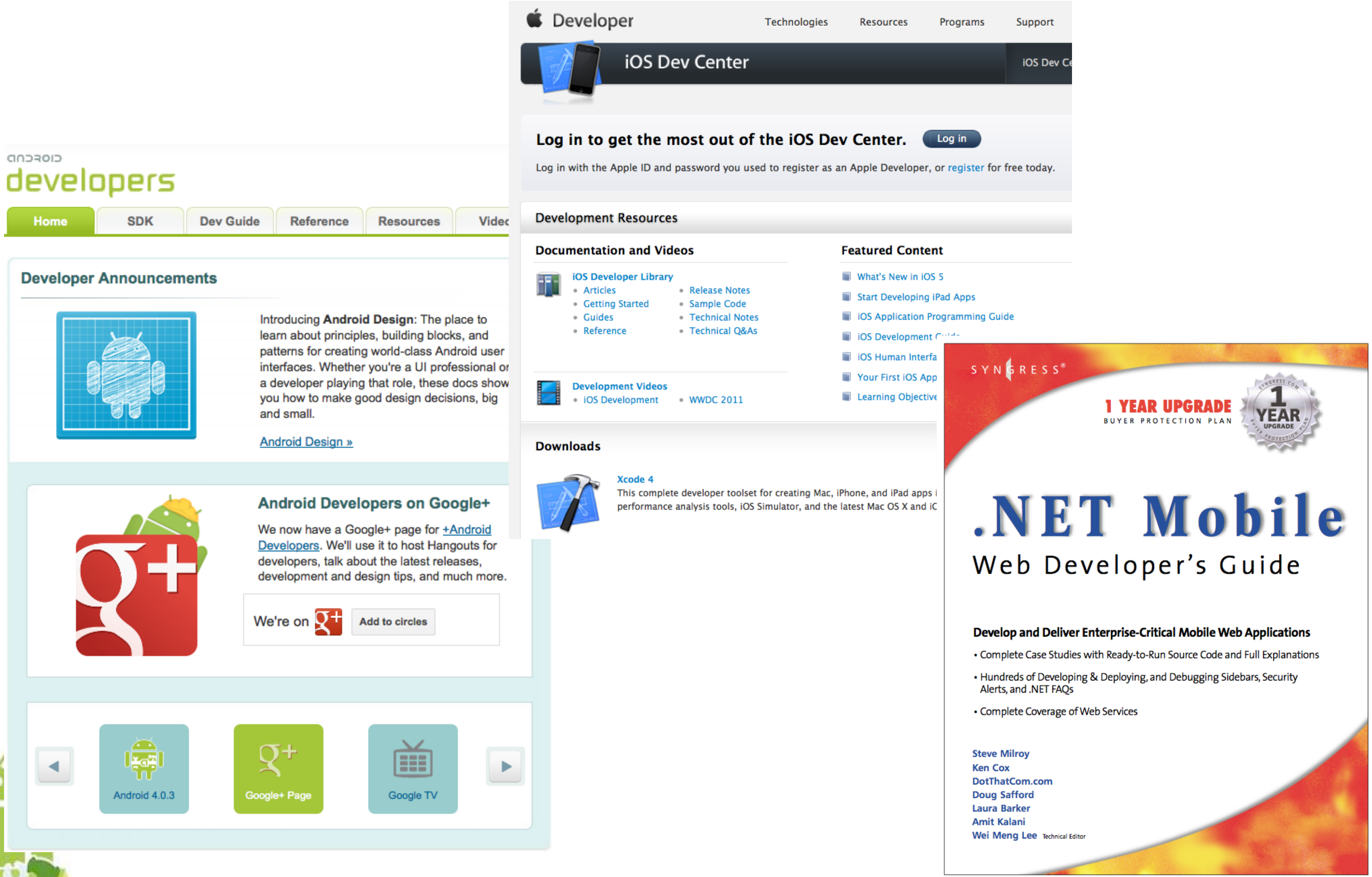
Smartphones

Nokia Communicator, 2000

Symbian OS 2000 => 2011

Blackberry, 2002 =>

Mobile: many worlds…



android developers

Developer Announcements

Introducing **Android Design**: The place to learn about principles, building blocks, and patterns for creating world-class Android user interfaces. Whether you're a UI professional or a developer playing that role, these docs show you how to make good design decisions, big and small.

Android Design »

Android Developers on Google+

We now have a Google+ page for +Android Developers. We'll use it to host Hangouts for developers, talk about the latest releases, development and design tips, and much more.

Log in to get the most out of the iOS Dev Center.

Log in with the Apple ID and password you used to register as an Apple Developer, or register for free today.

Development Resources

Documentation and Videos

iOS Developer Library

- Articles
- Getting Started
- Guides
- Reference
- Release Notes
- Sample Code
- Technical Notes
- Technical Q&As

Development Videos

- iOS Development
- WWDC 2011

Featured Content

- What's New in iOS 5
- Start Developing iPad Apps
- iOS Application Programming Guide
- iOS Development G
- iOS Human Interfa
- Your First iOS App
- Learning Objective

Downloads

Xcode 4

This complete developer toolset for creating Mac, iPhone, and iPad apps i performance analysis tools, iOS Simulator, and the latest Mac OS X and iC

.NET Mobile

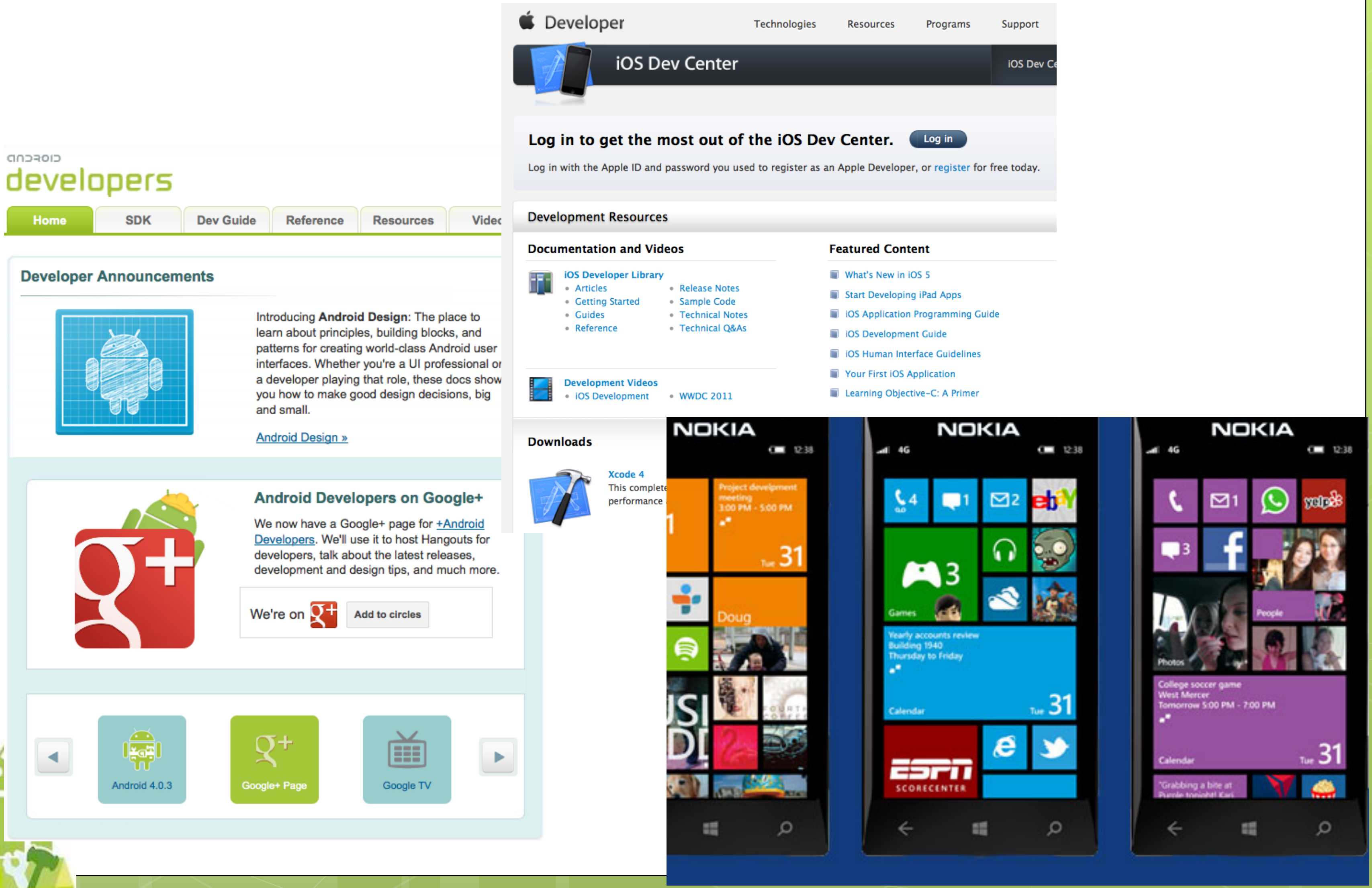
Web Developer's Guide

Develop and Deliver Enterprise-Critical Mobile Web Applications

- Complete Case Studies with Ready-to-Run Source Code and Full Explanations
- Hundreds of Developing & Deploying, and Debugging Sidebars, Security Alerts, and .NET FAQs
- Complete Coverage of Web Services

Steve Milroy
Ken Cox
DotThatCom.com
Doug Safford
Laura Barker
Amit Kalani
Wei Meng Lee Technical Editor

Mobile: many worlds…



android developers

Home | SDK | Dev Guide | Reference | Resources | Videc

Developer Announcements

Introducing **Android Design**: The place to learn about principles, building blocks, and patterns for creating world-class Android user interfaces. Whether you're a UI professional or a developer playing that role, these docs show you how to make good design decisions, big and small.

Android Design »

Android Developers on Google+

We now have a Google+ page for +Android Developers. We'll use it to host Hangouts for developers, talk about the latest releases, development and design tips, and much more.

We're on g+ | Add to circles

Developer | Technologies | Resources | Programs | Support

iOS Dev Center | iOS Dev Ce

Log in to get the most out of the iOS Dev Center. Log in

Log in with the Apple ID and password you used to register as an Apple Developer, or register for free today.

Development Resources

Documentation and Videos

iOS Developer Library
- Articles
- Getting Started
- Guides
- Reference
- Release Notes
- Sample Code
- Technical Notes
- Technical Q&As

Development Videos
- iOS Development
- WWDC 2011

Featured Content

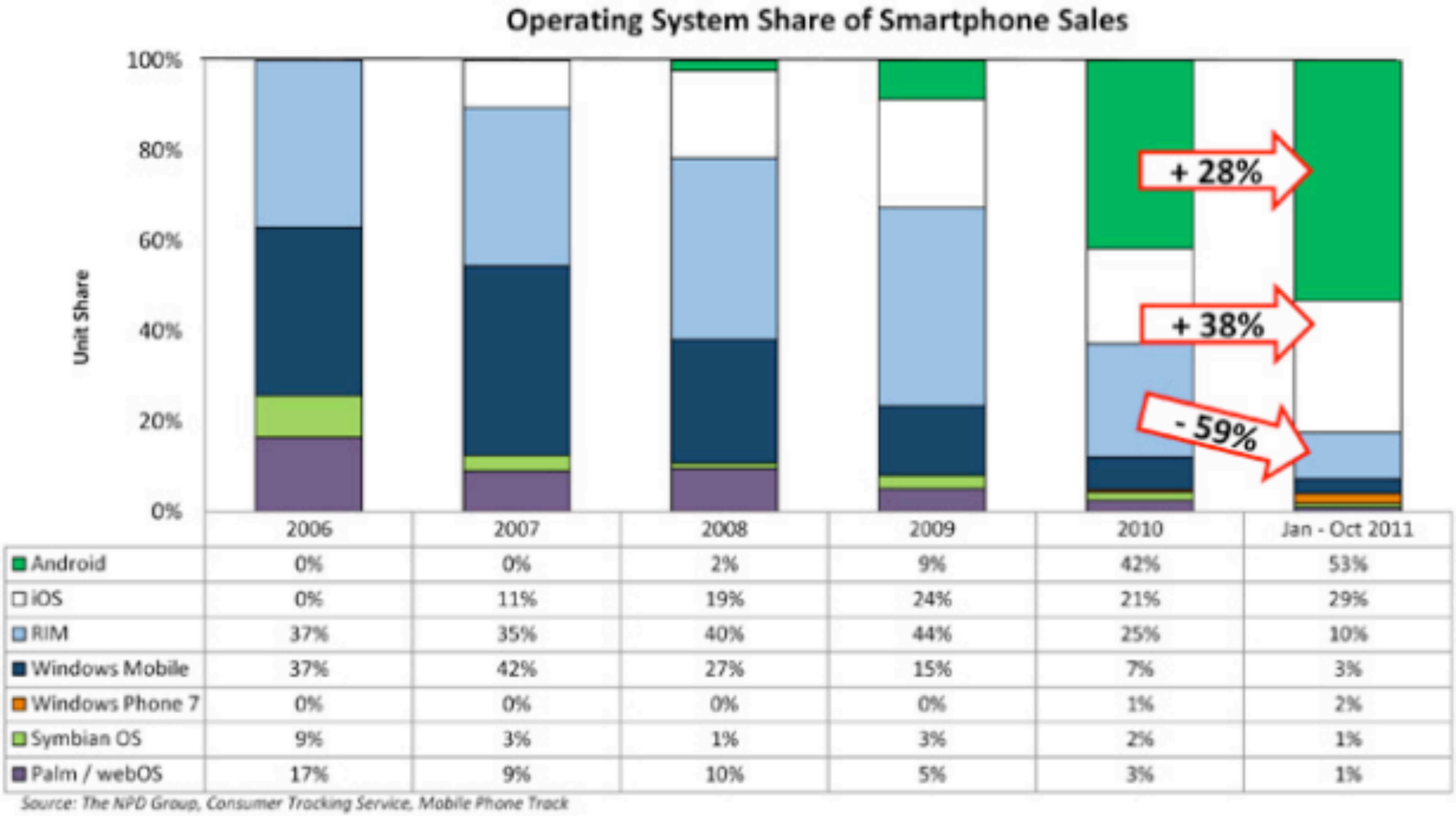
- What's New in iOS 5
- Start Developing iPad Apps
- iOS Application Programming Guide
- iOS Development Guide
- iOS Human Interface Guidelines
- Your First iOS Application
- Learning Objective-C: A Primer

Downloads

Xcode 4
This complete
performance

NOKIA
NOKIA
NOKIA
Project development meeting 3:00 PM - 5:00 PM
Tue 31
Doug
Games
Yearly accounts review Building 1940 Thursday to Friday
Calendar
Tue 31
ESPN SCORECENTER
People
Photos
College soccer game West Mercer Tomorrow 5:00 PM - 7:00 PM
Calendar
Tue 31

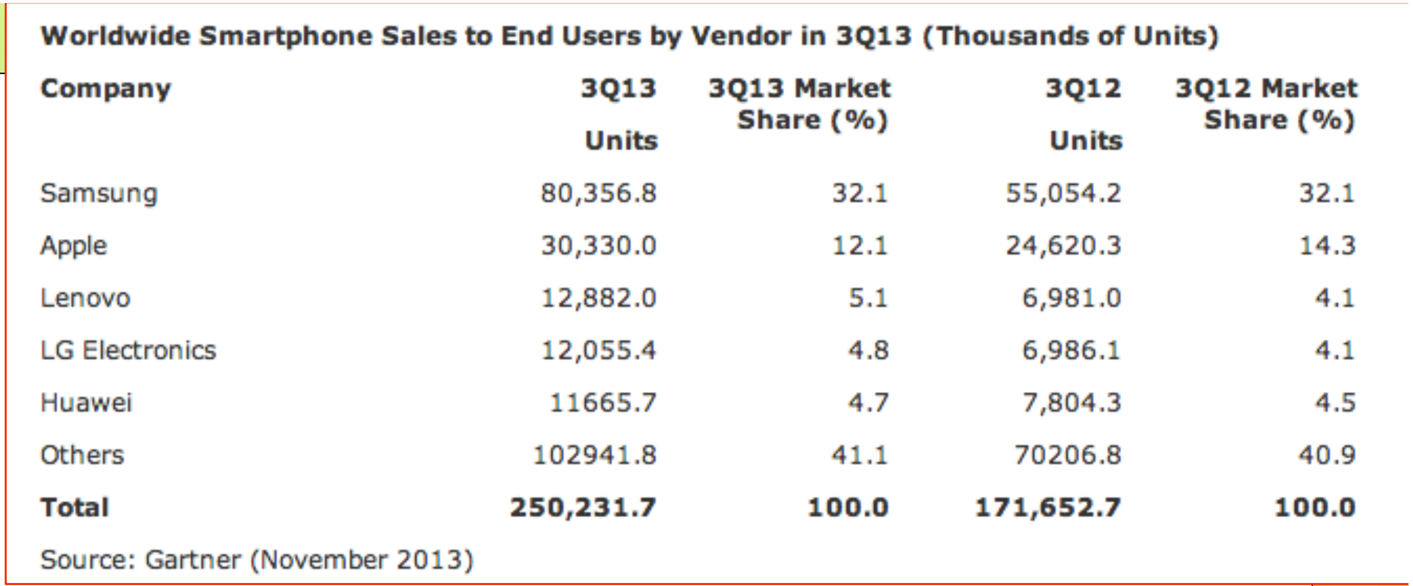
What the market says:



	2006	2007	2008	2009	2010	Jan - Oct 2011	2013
🟩 Android	0%	0%	2%	9%	42%	53%	82%
□ iOS	0%	11%	19%	24%	21%	29%	12%
🟦 RIM	37%	35%	40%	44%	25%	10%	1,8%
■ Windows Mobile	37%	42%	27%	15%	7%	3%	3,6%
🟧 Windows Phone 7	0%	0%	0%	0%	1%	2%	
🟩 Symbian OS	9%	3%	1%	3%	2%	1%	0,2%
🟪 Palm / webOS	17%	9%	10%	5%	3%	1%	

Source: The NPD Group, Consumer Tracking Service, Mobile Phone Track

13/11/13



Worldwide Smartphone Sales to End Users by Vendor in 3Q13 (Thousands of Units)

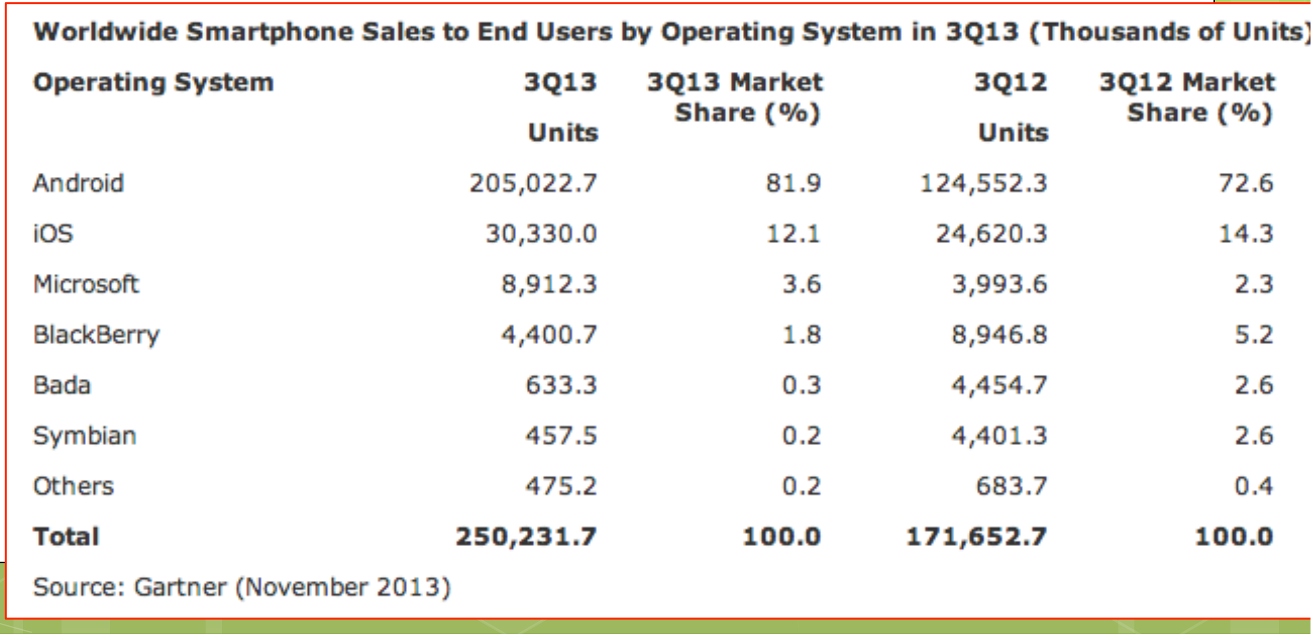
Company	**3Q13** **Units**	**3Q13 Market Share (%)**	**3Q12** **Units**	**3Q12 Market Share (%)**
Samsung	80,356.8	32.1	55,054.2	32.1
Apple	30,330.0	12.1	24,620.3	14.3
Lenovo	12,882.0	5.1	6,981.0	4.1
LG Electronics	12,055.4	4.8	6,986.1	4.1
Huawei	11665.7	4.7	7,804.3	4.5
Others	102941.8	41.1	70206.8	40.9
Total	**250,231.7**	**100.0**	**171,652.7**	**100.0**

Source: Gartner (November 2013)

Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013

- Western Europe Grew for the First Time this Year

- Lenovo Became the No. 3



Worldwide Smartphone Sales to End Users by Operating System in 3Q13 (Thousands of Units)

Operating System	**3Q13** **Units**	**3Q13 Market Share (%)**	**3Q12** **Units**	**3Q12 Market Share (%)**
Android	205,022.7	81.9	124,552.3	72.6
iOS	30,330.0	12.1	24,620.3	14.3
Microsoft	8,912.3	3.6	3,993.6	2.3
BlackBerry	4,400.7	1.8	8,946.8	5.2
Bada	633.3	0.3	4,454.7	2.6
Symbian	457.5	0.2	4,401.3	2.6
Others	475.2	0.2	683.7	0.4
Total	**250,231.7**	**100.0**	**171,652.7**	**100.0**

Source: Gartner (November 2013)

Apple vs. Google

- Open Handset Alliance

– Commitment to openness, shared vision, and concrete plans

- Compare with Mac/PC battles

– Similar (many PC manufacturers, one Apple)

– Different (Microsoft sells Windows, Google gives away Android)

Open Handset Alliance

Consortium founded by 34 companies (Google, Telecom Italia…) in 2007

Today it has 84 members

Includes:

- Mobile operators
- Software companies
- Commercialization companies
- Semiconductor companies
- Handset Manifacturers

Android or iOS ?

iOS:

- Develop in Objective-C
- Develop (only) on Macs (with emulator)

iPhone Open Development

- Jailbreak your iPhone or Touch
- Develop on any computer
    - Apps will not work with App Store
    - Device may not work with upgrades
- Need device! (iPhone or Touch)

Android:

- Develop in Java
- Develop on any platform (with emulator)

Introduction to Android

Laboratorio di programmazione di sistemi mobili e tablet

Marco Ronchetti, Università di Trento

What Android is not

- A Java ME implementation
- Part of the Linux Phone Standard Forum
- “only” an application layer
- A mobile phone

What Android is

From: http://source.android.com/

Android is:

an open-source software stack for mobile devices,

and a corresponding open-source project led by Google.

“We created Android in response to our own experiences launching mobile apps. We wanted to make sure that there was no central point of failure, so that no industry player can restrict or control the innovations of any other.”

“That's why we created Android, and made its source code open.” (under Apache Software Licence, 2.0)

The Android system tries to avoid incorporating GPL components

Why Android is not LGPL

LGPL requires either:

- shipping of source to the application;
- a written offer for source;
- linking the LGPL-ed library dynamically and allowing users to manually upgrade or replace the library.

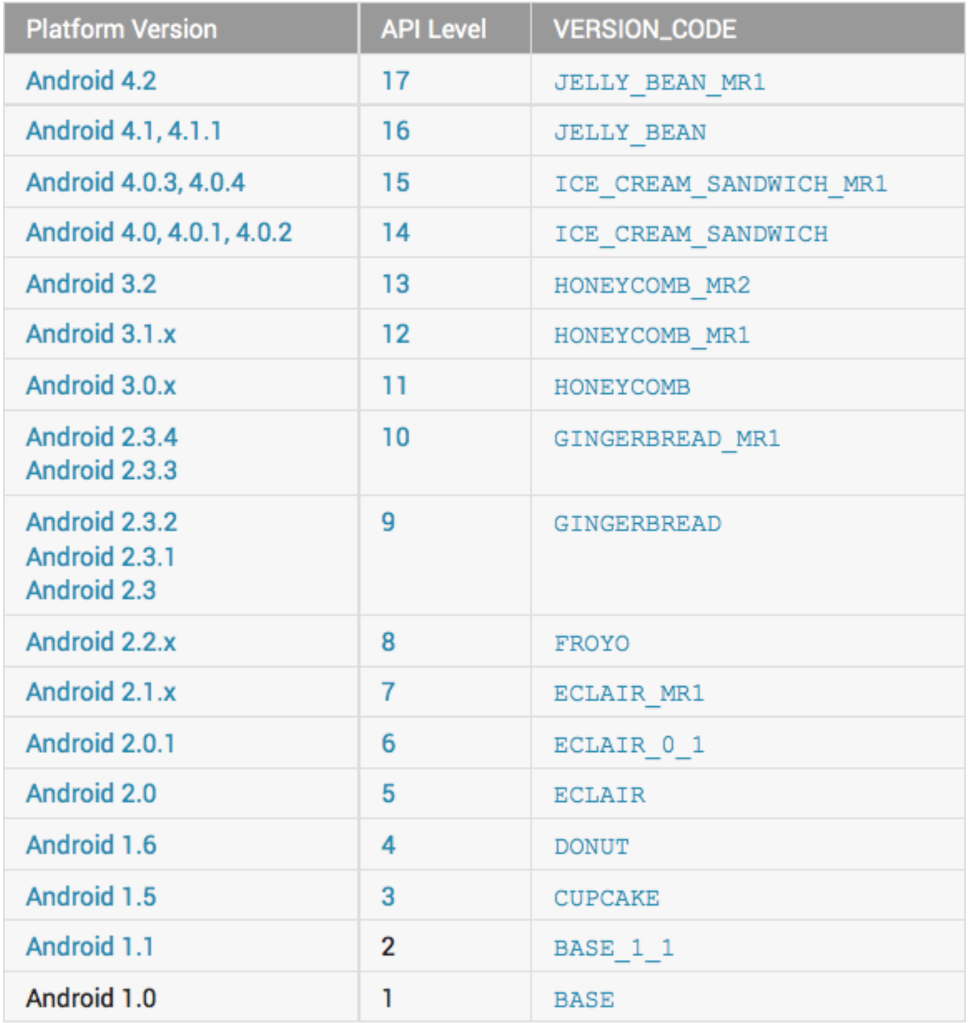
Since Android software is typically shipped in the form of a static system image, complying with these requirements ends up restricting OEMs' designs. (For instance, it's difficult for a user to replace a library on read-only flash storage.)

For more details: http://source.android.com/source/licenses.html

History

- Oct 2003 Android, Inc. founded in Palo Alto
- 2005 Google buys Android, Inc..
- 2007 Open Handset Alliance is announced. Android is officially open-sourced.
- 2008 Android SDK 1.0 is released. The G1 phone, manufactured by HTC, is sold by T-Mobile USA.
- 2009 sees a proliferation of Android-based devices (20+ devices run Android).
- 2010 Android is 2nd only to RIM as best-selling smart phone platform. 60+ devices run Android

Platform versions



	Platform Version	API Level	VERSION_CODE
	Android 4.2	17	JELLY_BEAN_MR1
	Android 4.1, 4.1.1	16	JELLY_BEAN
Nov.2011	Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
	Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
	Android 3.2	13	HONEYCOMB_MR2
Feb 2011	Android 3.1.x	12	HONEYCOMB_MR1
	Android 3.0.x	11	HONEYCOMB
	Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Dic 2010	Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Mag 2010	Android 2.2.x	8	FROYO
	Android 2.1.x	7	ECLAIR_MR1
Ott. 2009	Android 2.0.1	6	ECLAIR_0_1
	Android 2.0	5	ECLAIR
Apr 2009	Android 1.6	4	DONUT
	Android 1.5	3	CUPCAKE
	Android 1.1	2	BASE_1_1
Sept. 2008	Android 1.0	1	BASE

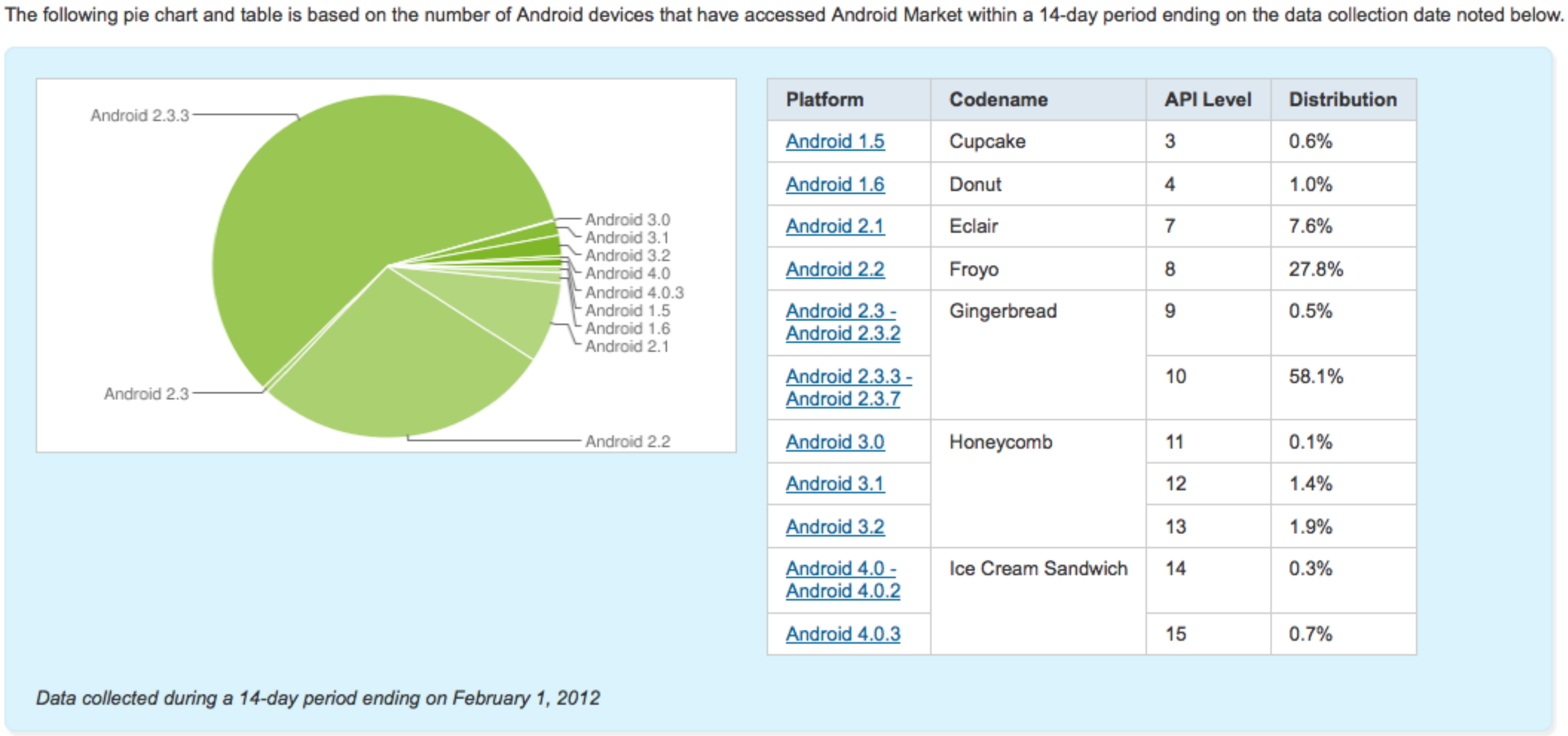
http://developer.android.com/guide/appendix/api-levels.html

Functionalities by version

- 1.5 Integrazione con servizi Google
- 1.6 Sintesi vocale, ricerca vocale, gestures
- 2.0 Miglior supporto videocamera, multitouch
- 2.2 Migliori prestazioni. Open GL ES2.0, Javascript e Flash. Tethering. Installazione apps su SD
- 2.3 Video chat in GoogleTalk. UI migliorata, Download manager
- 3.0 Ottimizzata per tablet. Aggiunta la barra di sistema e Action Bar. Possibilità di criptare tutti i dati personali.
- 3.1 Supporto per le periferiche USB
- 4.0 UI completamente riprogettata. Prestazioni migliorate. Dettatura real time. Face Unlock. Fotocamera migliorata. "Contatti" con integrazione con i social network

Android version distribution Feb 2012

http://developer.android.com/resources/dashboard/platform-versions.html



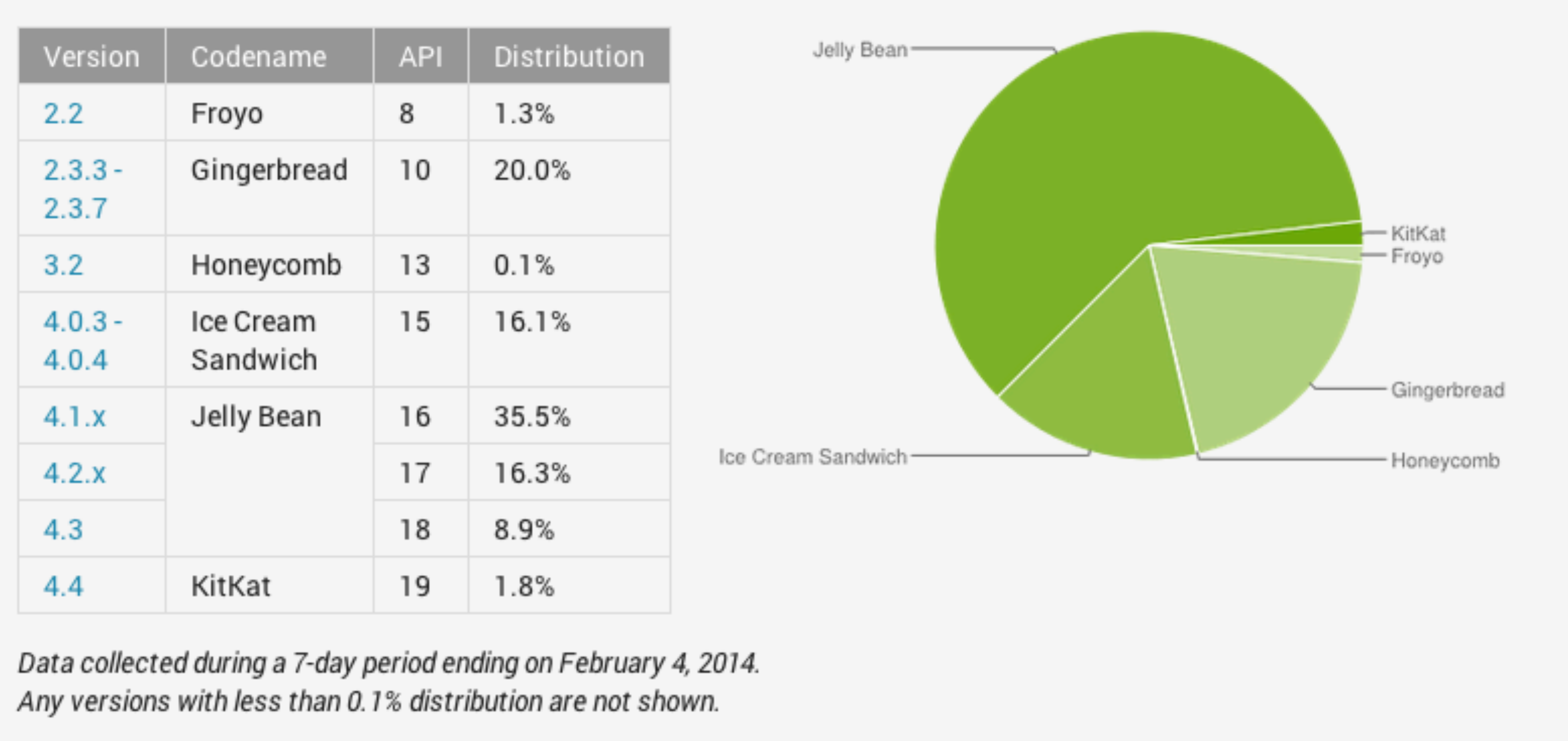
The following pie chart and table is based on the number of Android devices that have accessed Android Market within a 14-day period ending on the data collection date noted below.

Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	0.6%
Android 1.6	Donut	4	1.0%
Android 2.1	Eclair	7	7.6%
Android 2.2	Froyo	8	27.8%
Android 2.3 - Android 2.3.2	Gingerbread	9	0.5%
Android 2.3.3 - Android 2.3.7		10	58.1%
Android 3.0	Honeycomb	11	0.1%
Android 3.1		12	1.4%
Android 3.2		13	1.9%
Android 4.0 - Android 4.0.2	Ice Cream Sandwich	14	0.3%
Android 4.0.3		15	0.7%

Data collected during a 14-day period ending on February 1, 2012

Android version distribution – Feb 2014

http://developer.android.com/resources/dashboard/platform-versions.html



Version	Codename	API	Distribution
2.2	Froyo	8	1.3%
2.3.3 - 2.3.7	Gingerbread	10	20.0%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	16.1%
4.1.x	Jelly Bean	16	35.5%
4.2.x		17	16.3%
4.3		18	8.9%
4.4	KitKat	19	1.8%

Data collected during a 7-day period ending on February 4, 2014.
Any versions with less than 0.1% distribution are not shown.

Android and Linux

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model.

The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

Development and governance

- At any given moment, there is a current latest release of the Android platform
- Device builders and Contributors work with the current latest release, fixing bugs, launching new devices, experimenting with new features, and so on.
- In parallel, Google works internally on the next version of the Android platform and framework, working according to the product's needs and goals. We develop the next version of Android by working with a device partner on a flagship device whose specifications are chosen to push Android in the direction we believe it should go.
- When the "n+1"th version is ready, it will be published to the public source tree, and become the new latest release.

Three components

The Android Compatibility Program defines the technical details of Android platform and provides tools used by OEMs to ensure that developers' apps run on a variety of devices.

The Android SDK provides built-in tools that Developers use to clearly state the device features their apps require.

The Android Market shows apps only to those devices that can properly run them.

Building compatible devices

A three-step process:

Obtain the Android software source code.

Comply with Android Compatibility Definition Document (CDD). The CDD enumerates the software and hardware requirements of a compatible Android device.

Pass the Compatibility Test Suite (CTS). You can use the CTS (included in the Android source code) as an ongoing aid to compatibility during the development process.

"For a variety of legal and business reasons, we aren't able to automatically license Android Market to all compatible devices."

The main building blocks

Device Hardware: Android runs on a wide range of hardware configurations including smart phones, tablets, and set-top-boxes. Android is processor-agnostic, but it does take advantage of some hardware-specific security capabilities (e.g. on ARM).

Android Operating System: The core operating system is built on top of the Linux kernel. All device resources are accessed through the operating system.

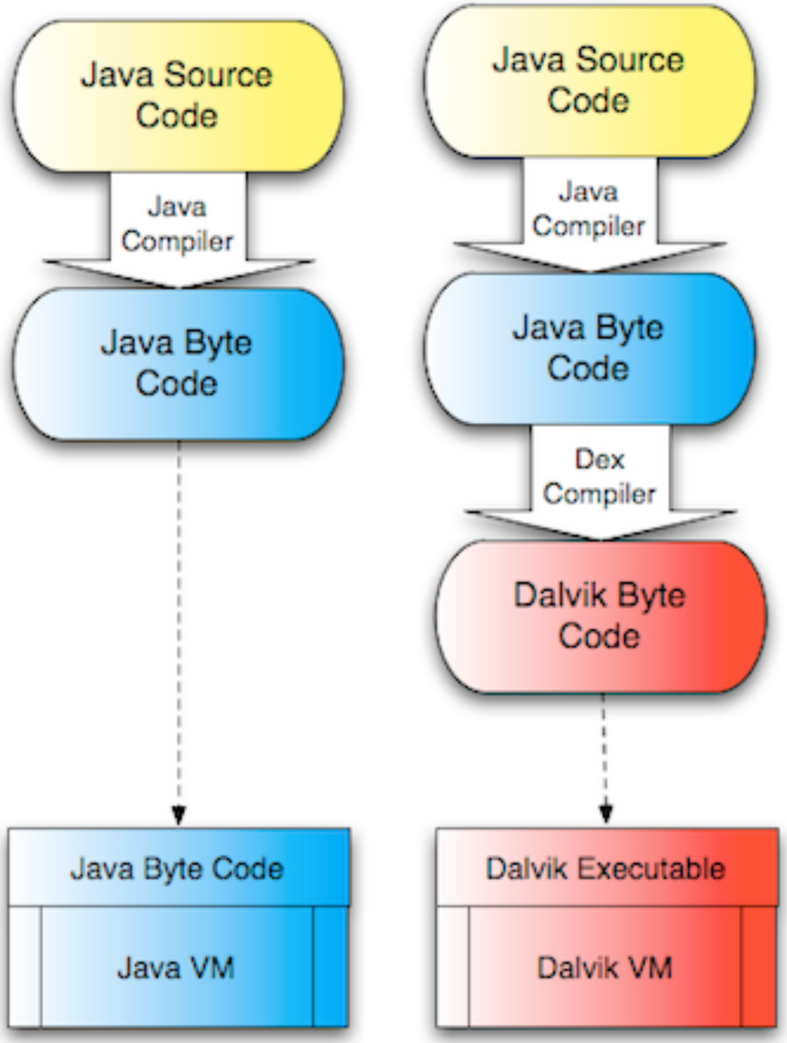
Android Application Runtime: Android applications are most often written in Java and run in the Dalvik V.M.

However, many applications, including core Android services and applications are native applications or include native libraries.

Both Dalvik and native applications run within the same security environment, in the Application Sandbox.

Applications get a dedicated part of the filesystem in which they can write private data, including databases and raw files.

Java vs. Dalvik



Dalvik is the managed runtime used by applications and some system services on Android. Dalvik was originally created specifically for the Android project.

Specification of the bytecode format, .dex (dalvik executable) and Dalvik VM Instruction Formats are available at

http://source.android.com/tech/dalvik/index.html

Dalvik

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine.

Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Android is non standard Java

Standard Java distributions:

1. Java Standard Edition: used for development on basic desktop-type applications.
2. Java Enterprise Edition (aka J2EE or JavaEE): used for development of enterprise applications.
3. Java Micro Edition (aka J2ME or JavaME): Java for mobile applications.

Android’s Java set of libraries is closest to Java Standard Edition. The major difference is that Java libraries for user interface (AWT and Swing) have been taken out and replaced with Android-specific user interface libraries. Android also adds quite a few new features to standard Java while supporting most of Java’s standard features.

What Android supports

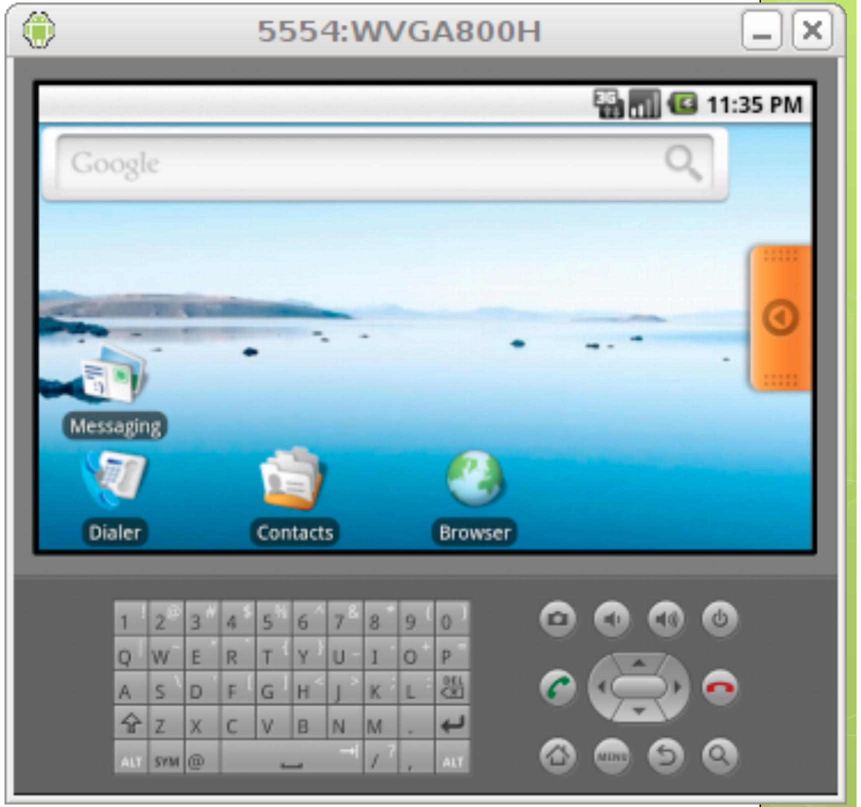
- User Interface
  - IO widgets (buttons, textboxes, lists)
  - Images
  - 2D/3D drawing
- Database
- Integrated browser
- Media support (audio, video, images; camera)
- Application framework lifecycle
- Connectivity (bluetooth, wi-fi, EDGE, 3G)
- Sensors (GPS/Geo-location, accelerometer, compass)
- GSM Telephony (call – sms)
- Google Maps
- Multiple processes
  - Managed by Android Dalvik VM
  - Background Services
  - Interprocess communications (e.g. Intents)
- Rich development environment including a device emulator, debugging tools, memory and performance profiling, Eclipse plug in

The emulator: can simulate…

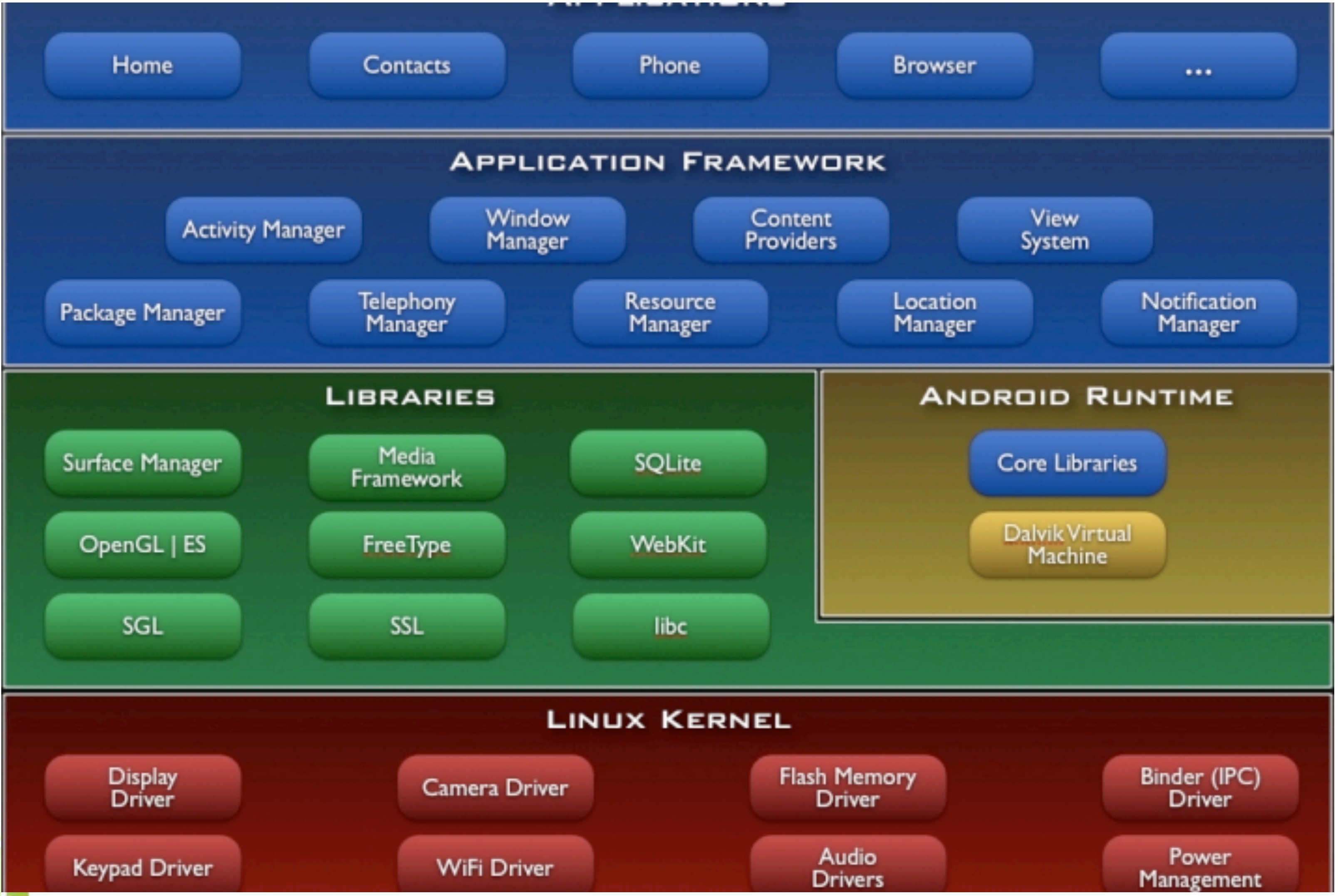
- Battery charge level/AC charging state
- GPS signal Connectivity:
- phone calls
- SMS
- Network speed/delay
- SD card

No support for:

- USB connections
- Bluetooth
- Device-attached headphones
- SD insert/eject
- Camera/video input
- Determining connected state



The Android stack



Home
Contacts
Phone
Browser
...

APPLICATION FRAMEWORK

Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager

LIBRARIES

Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc

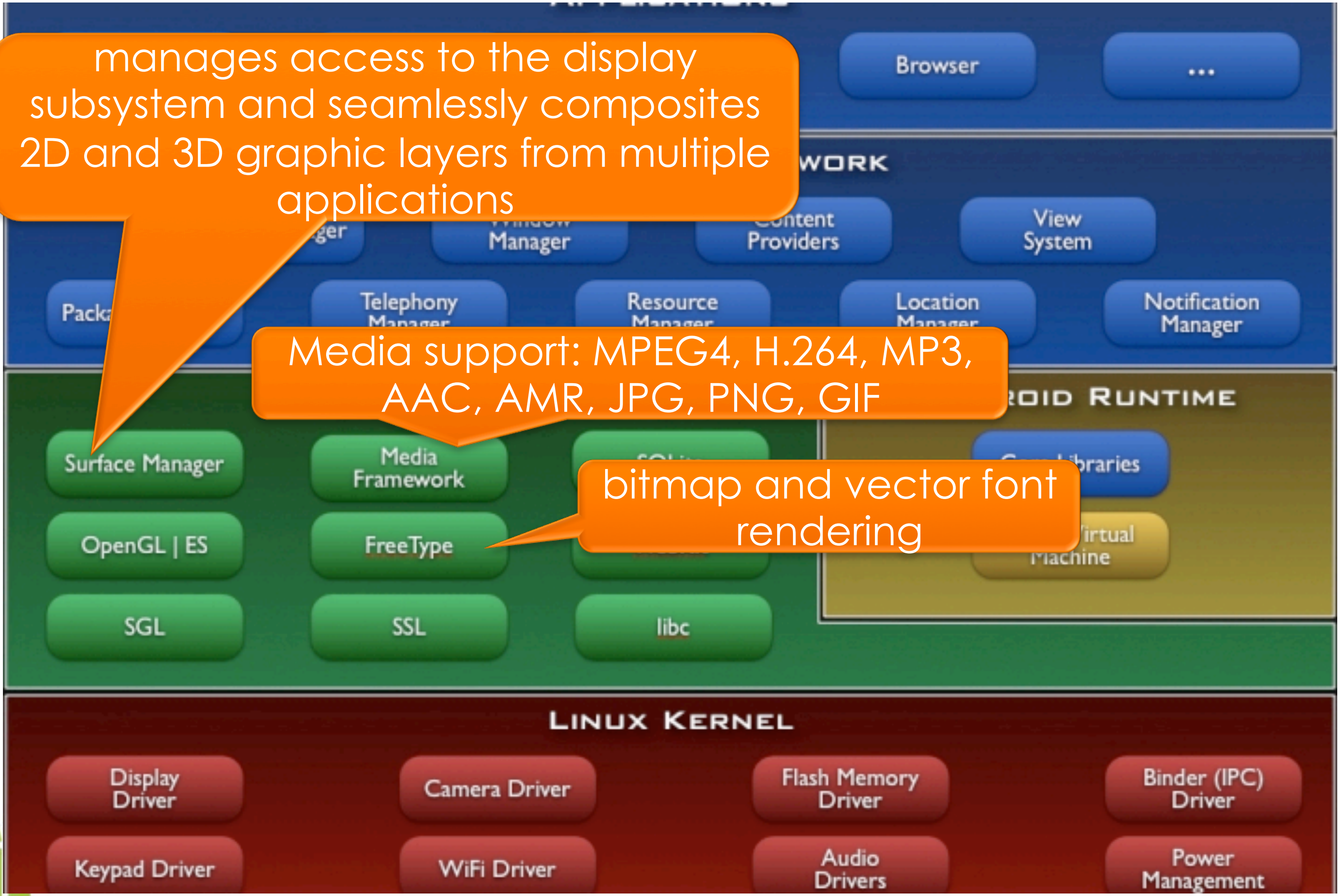
ANDROID RUNTIME

Core Libraries
Dalvik Virtual Machine

LINUX KERNEL

Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

The Android stack

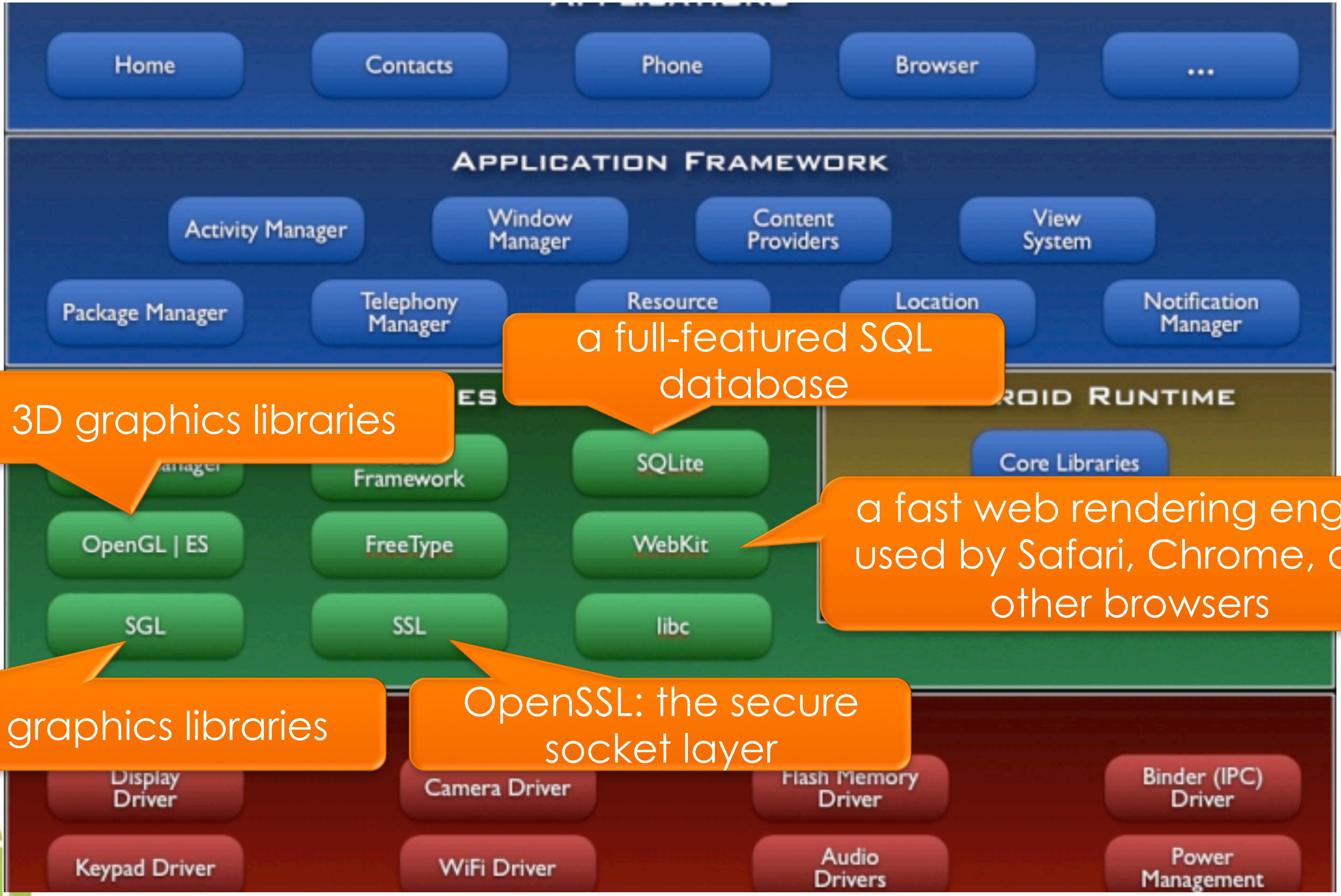


manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

Media support: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF

bitmap and vector font rendering

The Android stack



a full-featured SQL database

3D graphics libraries

a fast web rendering engine used by Safari, Chrome, and other browsers

2D graphics libraries

OpenSSL: the secure socket layer

Android NDK

The Android NDK is a toolset that lets you embed components that make use of native code in your Android applications.

Android applications run in the Dalvik virtual machine. The NDK allows you to implement parts of your applications using native-code languages such as C and C++. This can provide benefits to certain classes of applications, in the form of reuse of existing code and in some cases increased speed.

When to develop in NDK

“Using native code does not result in an automatic performance increase, but always increases application complexity.”

“In general, you should only use native code if it is essential to your application, not just because you prefer to program in C/C++.”

Android applications

Pre-Installed Applications:

phone, email, calendar, web browser, and contacts. These function both as user applications and to provide key device capabilities that can be accessed by other applications. Pre-installed applications may be part of the open source Android platform, or they may be developed by an OEM for a specific device.

User-Installed Applications:

Android provides an open development environment supporting any third-party application. The Android Market offers users hundreds of thousands of applications.

Cloud-based services

Android Market: a collection of services that allow users to discover, install, and purchase applications from their Android device or the web.

The Market also provides community review, application license verification, and other security services.

Android Update Service: delivers new capabilities and security updates to Android devices, including updates through the web or over the air (OTA).

Application Services: Frameworks that allow Android applications to use cloud capabilities such as

- (backing up) application data and settings
- cloud-to-device messaging (C2DM) for push messaging.