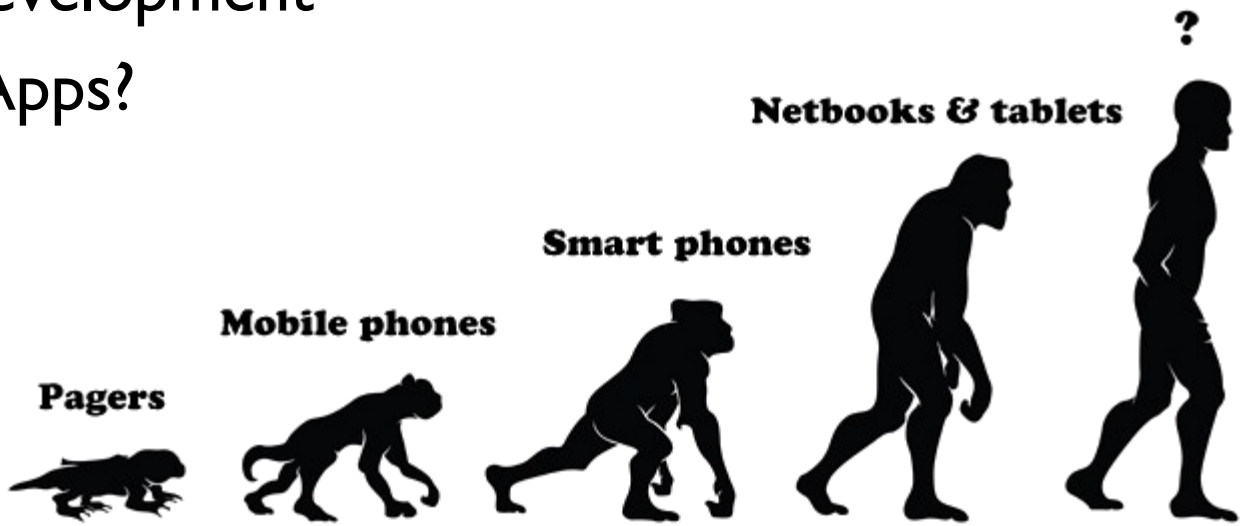
# ANDROID

## Programming basics

# Overview

- Mobile Hardware History
  - Android evolution
- Android smartphone overview
  - Hardware components at high level
  - Operative system
- Android App development
- Why Android Apps?



# History of Mobile Hardware

- PDA – Personal Digital Assistants were precursors
  - Personal data management and productivity (contacts, address book, etc)
  - Online synchronization
  - Limited dialup Internet connectivity
  - Bluetooth
  - Cell data services - eventually 3G
  - WiFi

- Mobile Phones (early ’90)
- PDAs and Phones merge - Palm Pilot, etc,
- Smartphones replaced PDA - more capabilities, browser, apps

# Mobile Phones

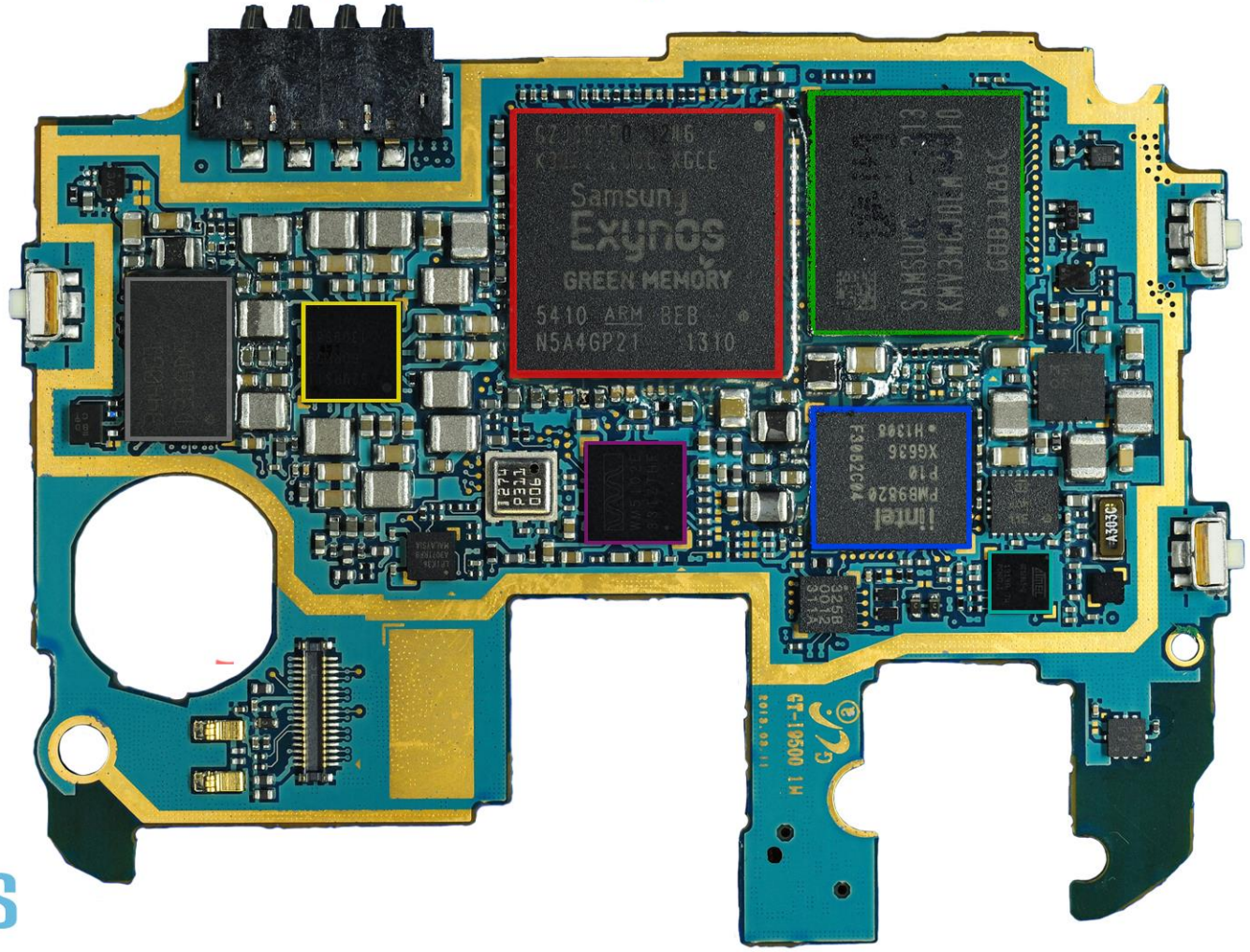
- 1st mobile phone – Motorola Brick DynaTAC 8000x – 1983
- Bag phones – car phones – early 90s
- Camera phones – late '90s
- Addition of data services


# Mobile Development Evolves

- WAP (Wireless Application Protocol) standard/browsers. Wireless Markup Language
  - Considered clunky and limited, but it was cross platform
- Proprietary formats emerged to better take advantage of hardware capabilities:
  - Palm OS (became Garnet OS)
  - RIM Blackberry OS
  - Java Micro Edition
  - Symbian OS (Sony Ericsson, Motorola, Samsung)
  - Windows Phone (Nokia)
  - iPhone iOS
- Major players now:
  - iOS
  - Android
  - Windows Phone 7

# Samsung Galaxy S4 COMMUNICATIONS BOARD, FRONT

- **Samsung Exynos 5410**
  Exynos Octa Eight-Core processor
- **Samsung K3QF2F200C-XGCE**
  2GB LP DDR3 (K4E4E324EB die) mobile DRAM
- **Intel PMB9820**
  X-GOLD 636 Baseband Processor
- **Samsung KMV3W000LM-B310**
  Multichip Memory - 64 MB Mobile DDR SDRAM, 16 GB MLC NAND Flash, Controller
- **Samsung S2MPS11**
  Power Management IC
- **Wolfson Micro WM5102E**
  Audio Hub CODEC with Voice Processor DSP
- **Atmel UC128L5-U**
  32 bit Microcontroller with 128KB Flash (custom package)
- **Broadcom BCM4335**
  Wi-Fi 802.11 a/b/g/n/ac, dual-band, DLNA, Wi-Fi Direct, Wi-Fi hotspot all-in-one IC

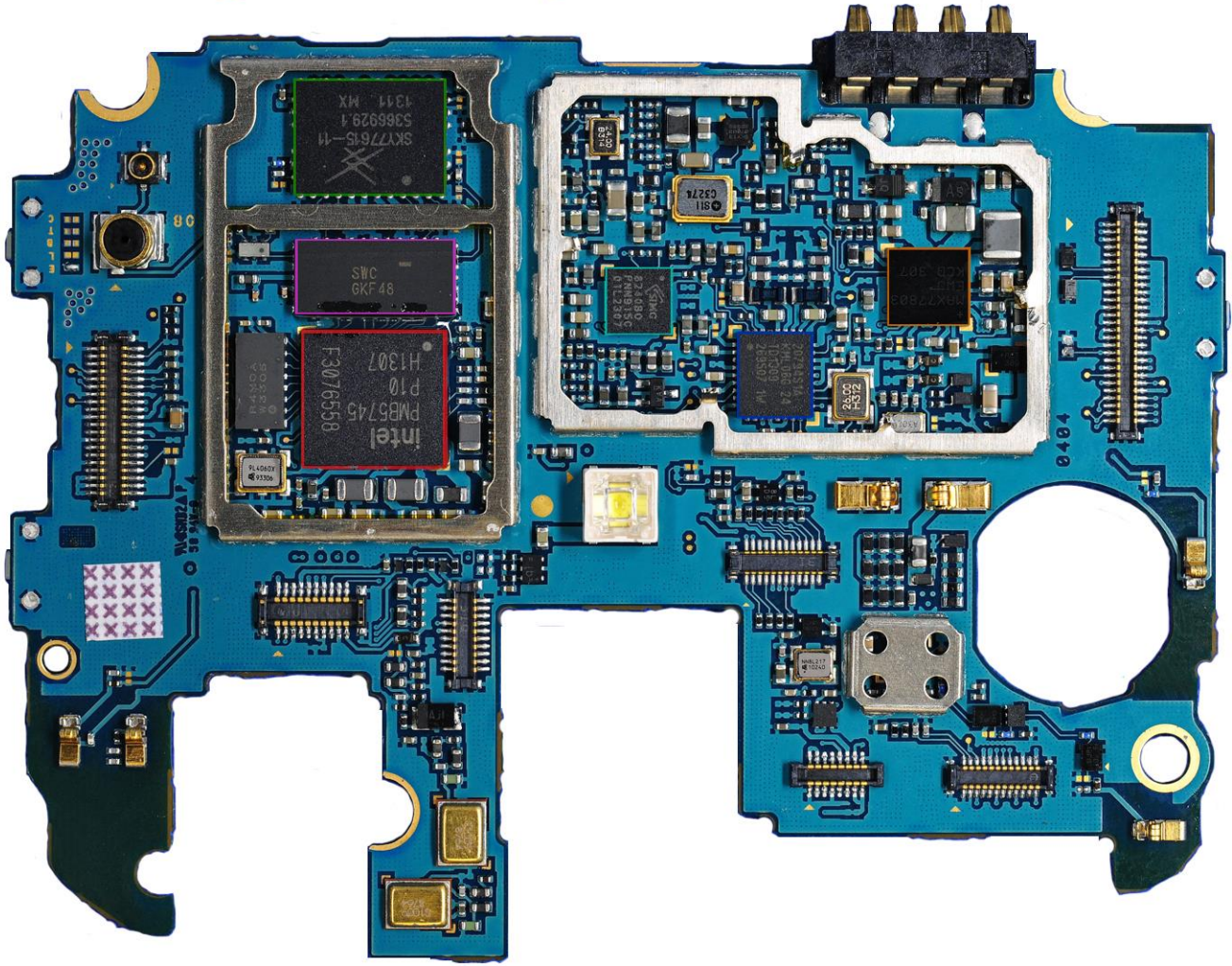



TECHINSIGHTS

# Samsung Galaxy S4

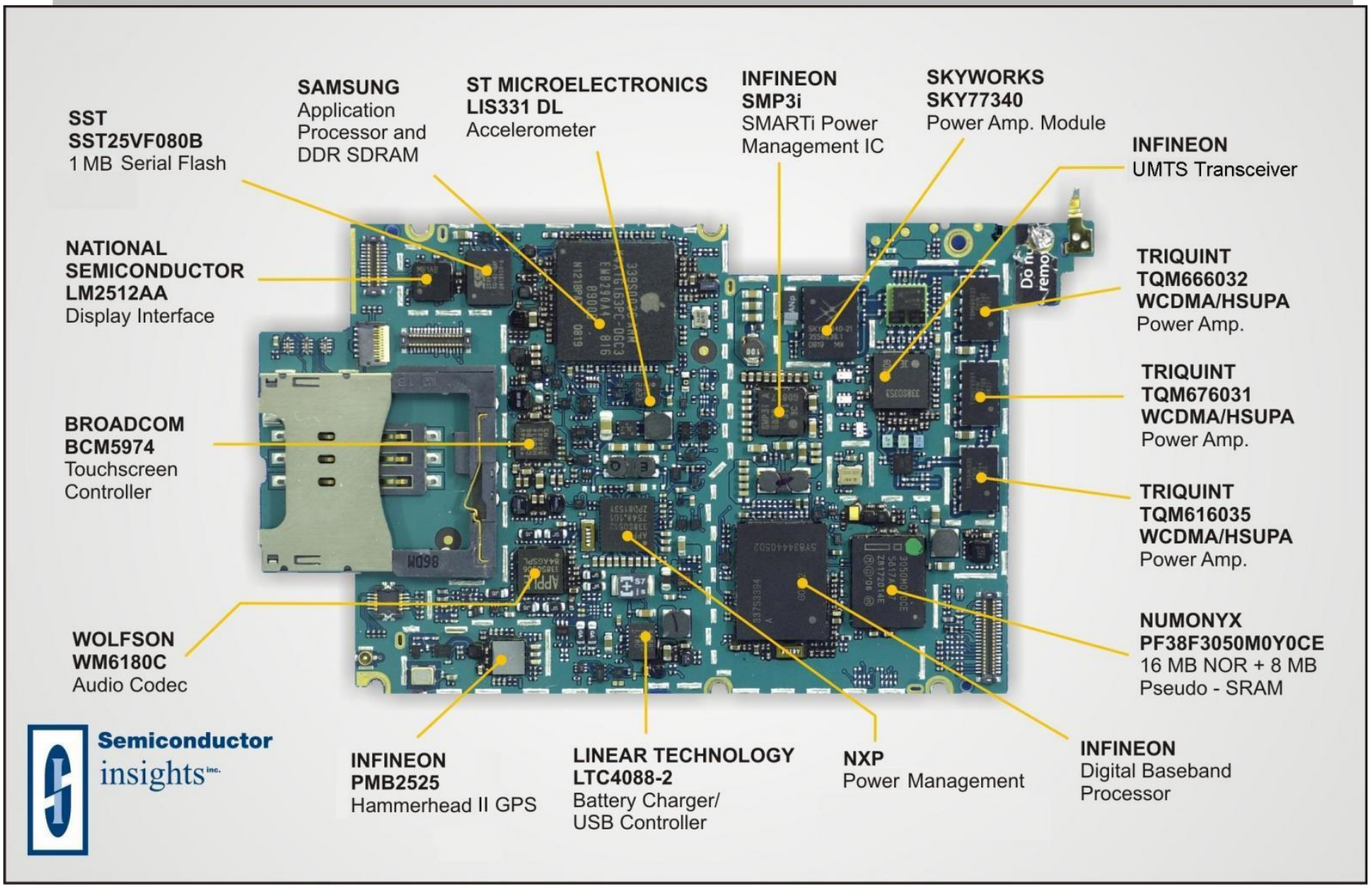
COMMUNICATIONS BOARD, BACK

- **Intel PMB5745**
  Intel SMARTi UE3 RF Transceiver
- **Broadcom BCM20794**
  NFC controller IC
- **SkyWorks SKY77615-11**
  Multimode Multiband Power Amplifier Module (WCDMA-HSDPA-HSUPA)
- **Murata SWC GKF48**
  Antenna Switch Module
- **Maxim MAX77803**
  Power Management
- **SIMG (Silicon Image) SiI8240**
  MHL 2.0 transmitter with HDMI input




TECHINSIGHTS

# SMARTPHONE: INTERNAL STRUCTURE (simplified)




Semiconductor insights inc.

# Main HW Difference with iOS

- Different device size (more HW manufactors)
- Expandable memory
- USB connection
- External HW easier to product


# Android evolution

# Android project starts …

- 2003: The project starts (OS for mobiles)
- 2005: Google purchased the initial developer of the OS, Android Inc.
  - Start Dalvik VM development


- 2007: Open Handset Alliance (OHA) consortium announced (34 founding members)
  - Mobile handset makers (i.e. HTC), software developers (Google), some mobile carriers (i.e. Telecom) and chip makers (i.e. Qualcomm)
  - SDK development

# … continues …

android
open source project

- 2008: T-Mobile G1 announced
  - SDK 1.0 released
  - Google sponsors first Android Developer Challenge
  - Android Open Source Project (Apache license)
  - Android Dev Phone 1 released
- 2009: New SDK release
  - Cupcake (SDK 1.5)
    - Softkeyboard with autocomplete feature
    - Auto-rotation option
  - Donut (SDK 1.6)
    - New camera features
    - Search features improved (Quick/Voice)
  - Éclair (SDK 2.0/2.0.1/2.1)
    - New camera features
    - Multiple accounts

# … until …

- 2010: Nexus One released to the public
  - Froyo (SDK 2.2)
    - Expandable memory
    - USB tethering
  - Gingerbread (SDK 2.3)
    - UI update
    - NFC
- 2011: New SDK release
  - Honeycomb (SDK 3.0/3.1/3.2) for tablets only
    - New UI tablet oriented
    - Multi-core processor supporting
  - Ice Cream Sandwich (SDK 4.0/4.0.1/4.0.2/4.0.3)
    - WIFI direct
    - Changes to the UI
    - Face unlock
- 2012:
  - Ice Cream Sandwich (SDK 4.0.4)
    - Stability improvement
  - Jelly Bean (SDK 4.1)
    - Google Now

# … last days

- 2013:
  - Kit Kat (SDK 4.4)
    - NFC capabilities through Host Card Emulation
    - Wireless printing support
    - Storage access framework
    - New framework for UI transition

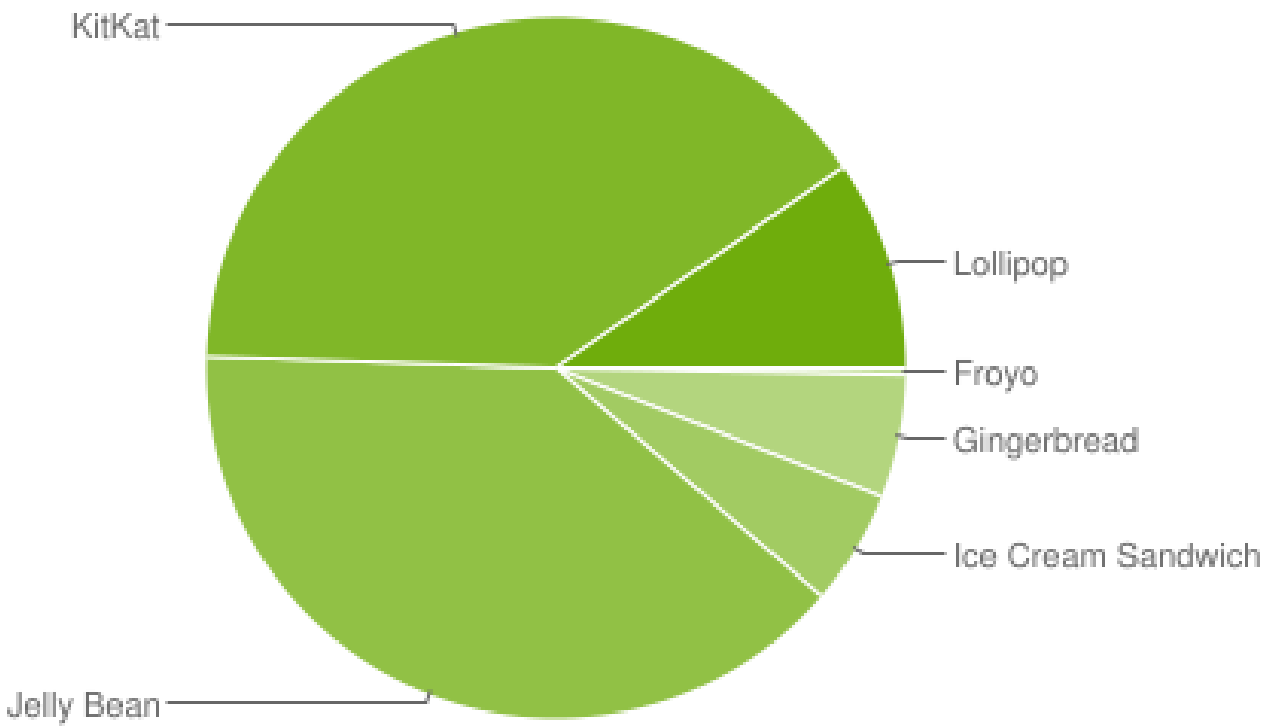
- 2014:
  - Lollipop (SDK 5)
    - Android RunTime (ART) with ahead-on-time (AOT) compilation
    - 64-bit CPU

# Last release: Marshmallow [2015]

- 2016:
  - Marshmallow(SDK 6.0)
    - Introduction of Doze mode, which reduces CPU speed while the screen is off in order to save battery lifeWireless printing support
    - Post-install/run-time permission requests
    - App permissions now granted individually at run-time, not all-or-nothing at install time.

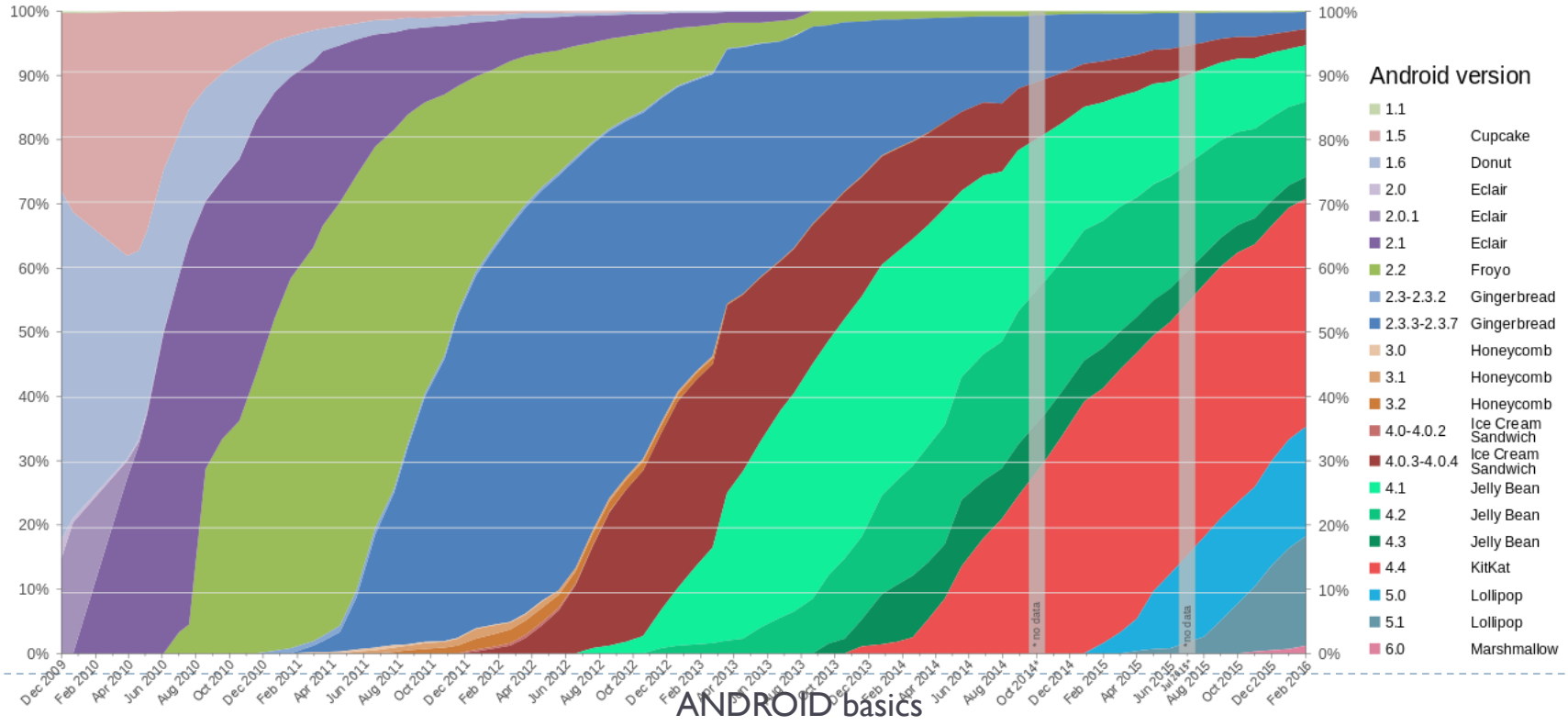
# Last Year Releases Distribution

- Froyo: 0.3%
- Gingerbread: 5.7%
- IceCreamS: 5.3%
- JellyBean: 39.2%
- KitKat: 39.8%
- Lollipop: 9.7%



# Current Releases Distribution

- Froyo: 0.1%
- Gingerbread: 2.2%
- IceCreamS: 2.0%
- JellyBean: 20.9%
- KitKat: 32.5%
- Lollipop: 35.6%
- Marshmallow: 7.5%

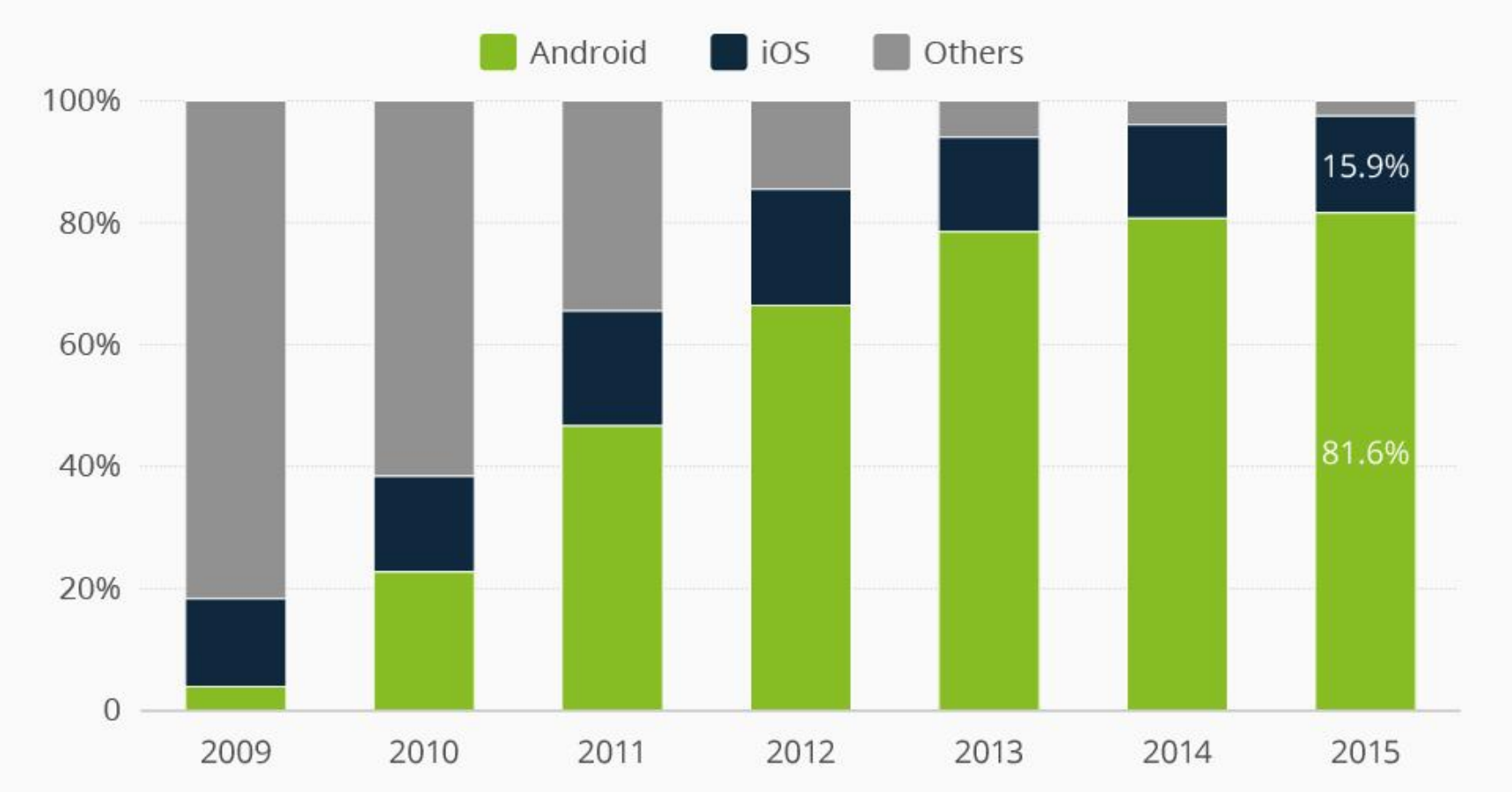


# Android is Growing Fast

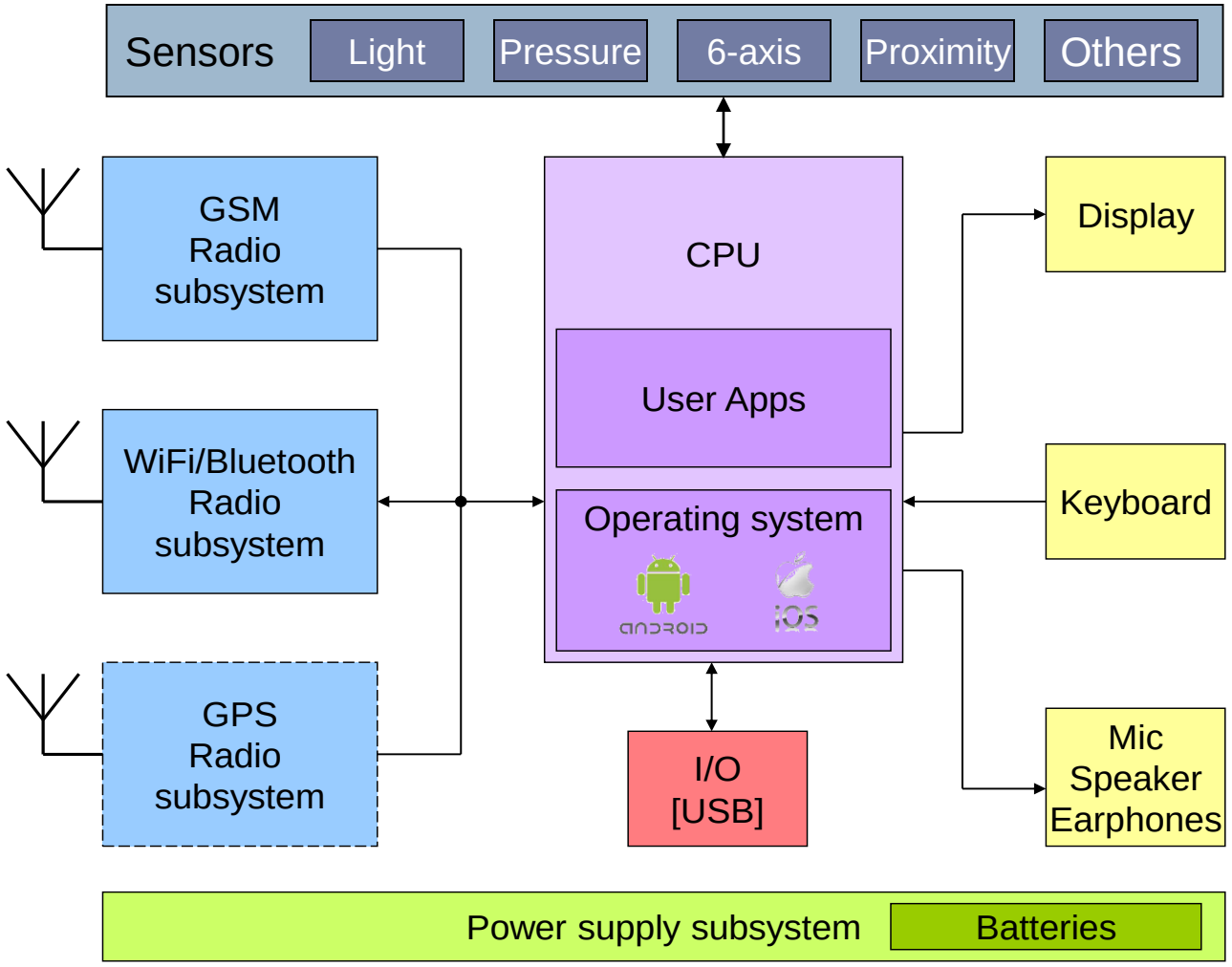
- It's the largest installed base of any mobile platform and growing fast (1.4 billion users)


  - Every day more than 1.5 million new Android devices are activated worldwide
  - 53.3% of the total smartphone' market in 2016
  - 70% of the overall tablet market
  - Not only smartphone: Smart tv, radio (more than 4000 devices)

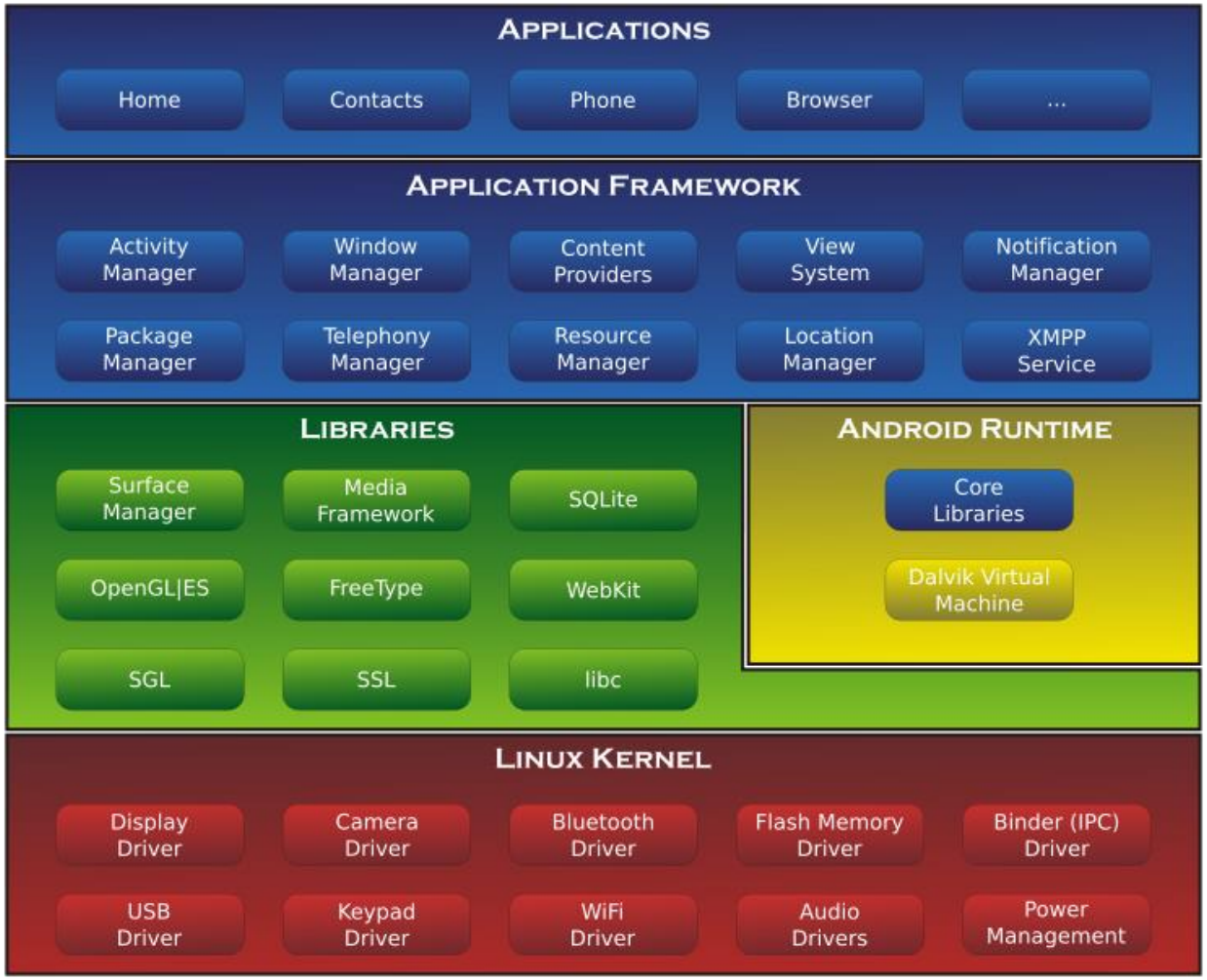
# Android is Growing Fast



# Smartphone high level block diagram



# Android Architecture



# Linux Kernel

- Android Linux Kernel has differentiated from Linux Kernel
  - From 2.6 ver to 3.8
- Basic SO services
  - Abstraction between hardware and software
  - Security
  - Memory management
  - Process management

# Libraries

- Run in system background
- Use C/C++ language
- Four types of libraries
  - Bionic Libc, system C libraries
  - Function libraries, supporting multimedia, web browser, SQlite
  - Native servers
  - Hardware Abstraction Libraries

# Core Libraries

- Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language
  - Core APIs for Java language provide a powerful, yet simple and familiar development platform

# Dalvik Virtual Machine

- Android custom implementation virtual machine
  - Provides application portability and runtime consistency
  - Runs optimized file format (.dex) and Dalvik bytecode
  - Java .class / .jar files converted to .dex at build time
- Designed for embedded environment
  - Supports multiple virtual machine processes per device
    - Each process an instance of the DVM
  - Highly CPU-optimized bytecode interpreter
  - Efficiently Using runtime memory
- The Dalvik VM relies on the Linux kernel for underlying functionality (threading and low-level memory management)

# Application Framework

- Simplify the reuse of components
  - Applications can publish their capabilities and any other application may then make use of those capabilities
- Applications is a set of services and systems,  include
  - Views system, content providers, resources manager and so on
- Frameworks
  - Activity Manager
  - Notification Manager
  - Resource Manager
  - Content Providers
  - Views

# Applications

- Contain a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others


- All yours Apps will belong to this layer
- All applications are written in Java programming language

# Software Development Kit (SDK)

- Software Development Kit (SDK) enables developers to create applications for the Android platform
- Sample projects source code
- Custom virtual machine
- Development tools:
  - Dalvik Debug Monitor Service (DDMS)
  - Android Debug Bridge (ADB)
  - Android Emulator
- SDK download link: http://developer.android.com/sdk/index.html

# Emulator

- Virtual mobile device on PC
- Allows to develop and test apps on PC without a physical device (simulate interrupt)


Android SDK Emulator

# Traditional VS App programming

- Only one App at a time ("multitasking")
- Only one window → Simplified UI
- Limited system access ("sandboxing")
- Limited resources and memory.
- Instant App opening and closing: application should start and quit instantaneously.
- App has her own lifecycle….
- Code must apply to many kind of devices

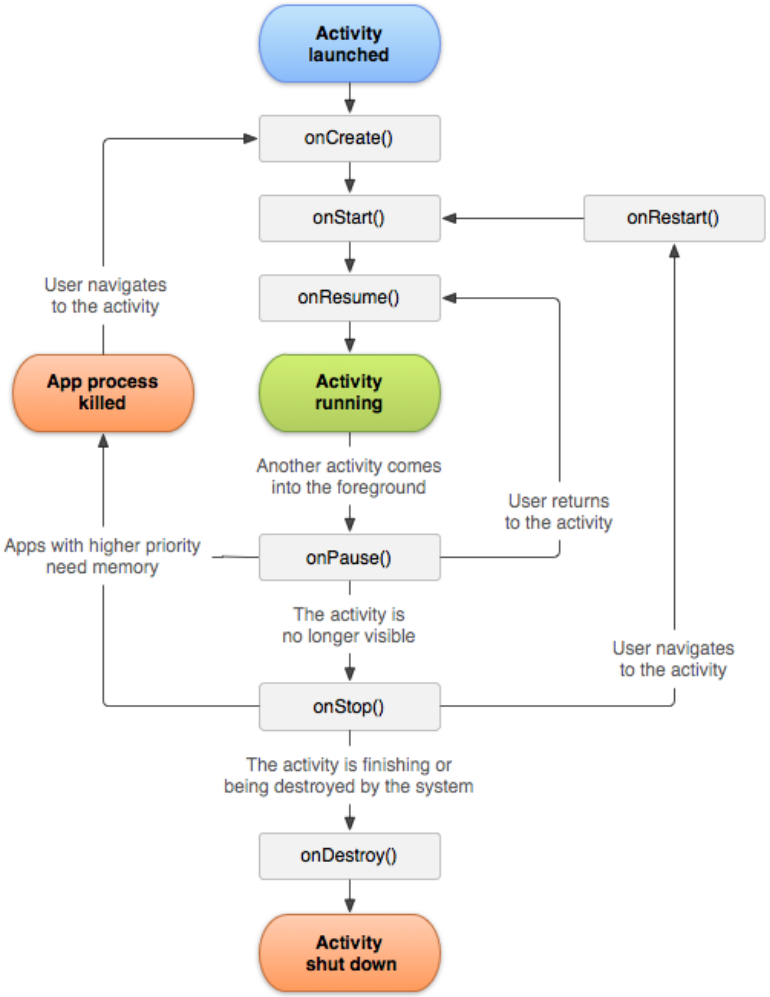
# Android App Basic Components

- Activities
  - Single screen of application, only "on screen" activity is in running state
  - Single App many Activities which can exchange data
  - Activities have a event-driven life-cycle
  - Activities is composed by graphic components
    - UI is built using a hierarchy of View and ViewGroup objects
      - View are usually UI widgets (e.g. textfield, button)
      - ViewGroup are invisible view containers that define how the child views are laid out (grid or list)
    - Android provides an XML vocabulary that corresponds to View and ViewGroup so you can define your UI in XML using a hierarchy of UI elements

# Widgets

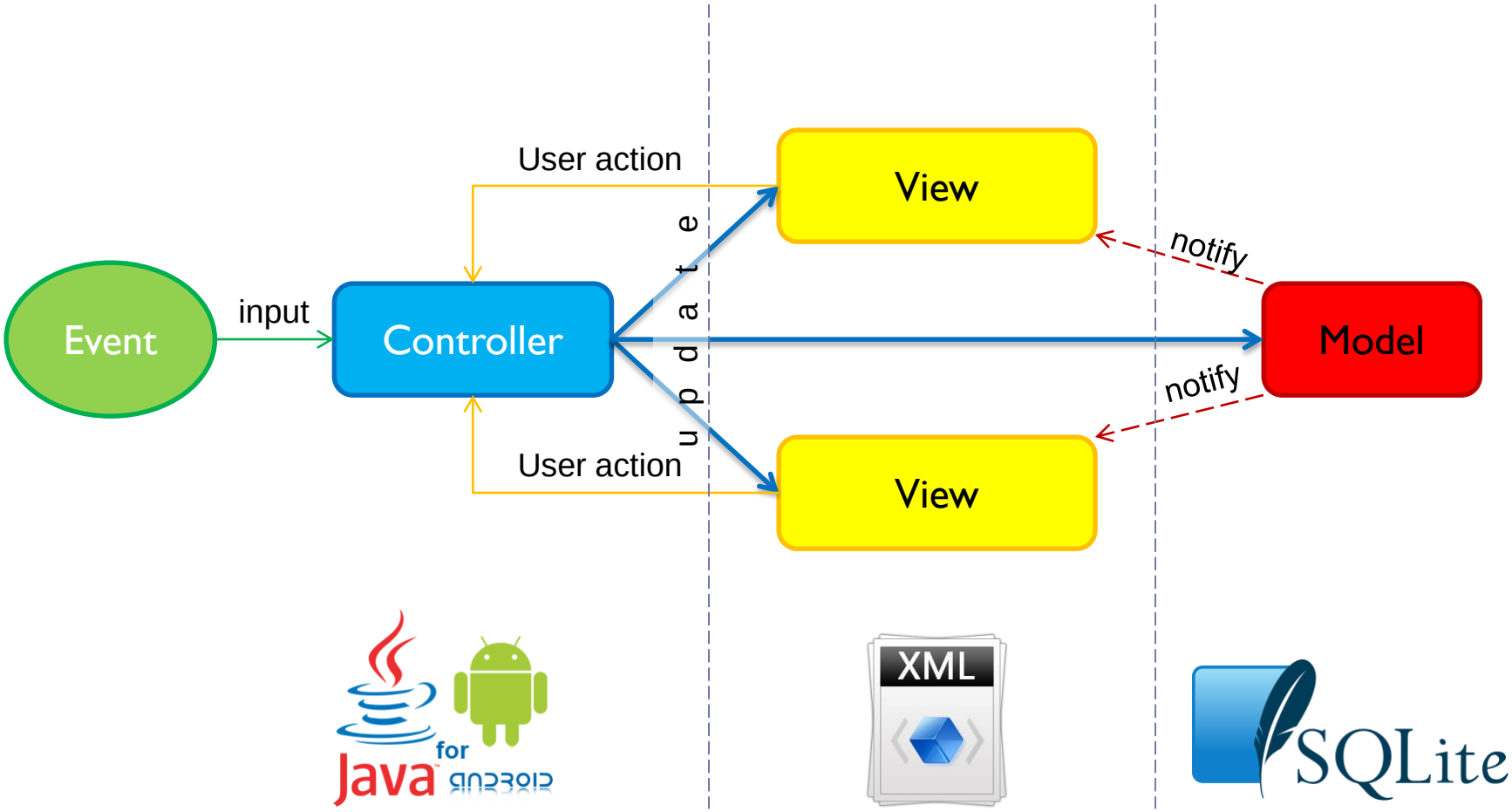

- Text View
- Button
- Toggle Button
- Check Box
- Radio Button
- Checked Text View
- Progressing Bar
- Seek Bar
- Quick Contact Badge
- Radio Group
- Rate Bar

# Activity Life Cycle




| Callback | When |
|---|---|
| onCreate() | App creation |
| onStart() | Activity visible |
| onResume() | after onStart |
| onRestart() | after onStop |
| onPause() | another Activity called |
| onStop() | Activity invisible |
| onDestroy() | Before Activity destruction |

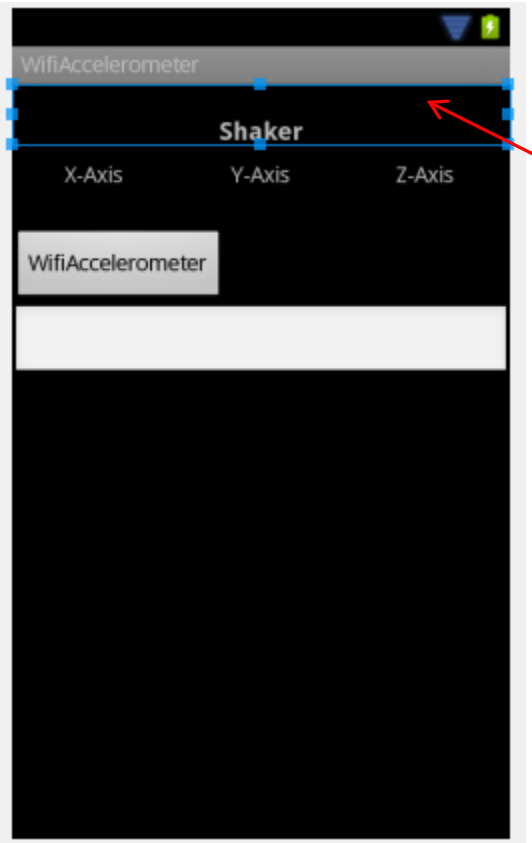
# Programming Pattern: MVC



# More Basic Components

- Intents
  - Asynchronous message that allow Activity to request functionality from other components (e.g. Activity)
- Services
  - Like Activities but run in background
  - No user interactions
- Content providers
  - Standard interface for sharing data among Applications
- Broadcast receivers
  - Receive notification from Android system

# Layout before

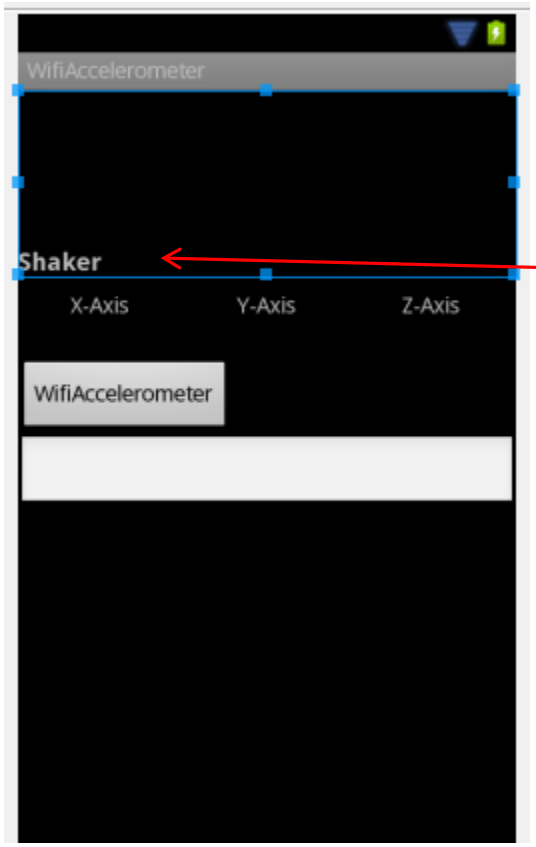



WifiAccelerometerAct | R.java | main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
    android:paddingTop="20dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:textStyle="bold"
    android:gravity="center"
    android:text="Shaker"/>
    <TableLayout
    android:paddingTop="10dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*">
    <TableRow>
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:text="X-Axis"
    android:gravity="center"/>
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

Graphical Layout | main.xml

# Layout after (var.1)

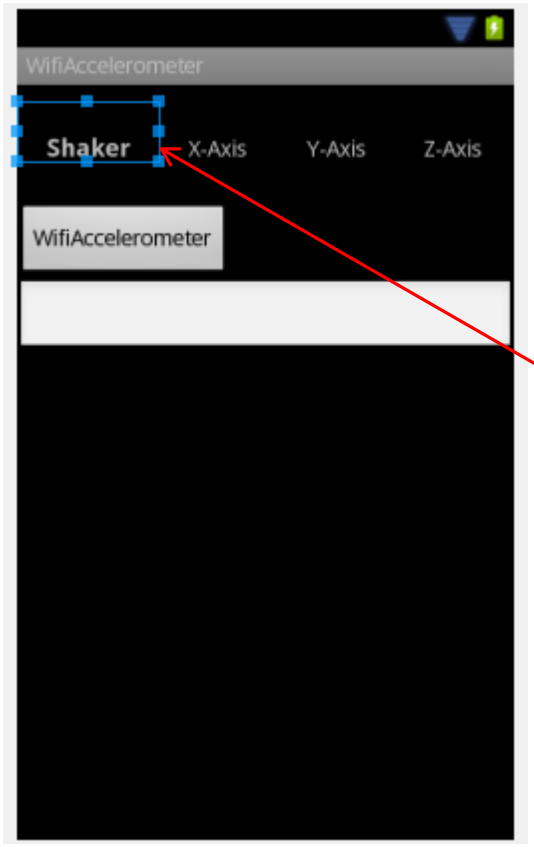



WifiAccelerometerAct | R.java | *main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
    android:paddingTop="100dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:textStyle="bold"
    android:gravity="default"
    android:text="Shaker"/>
    <TableLayout
    android:paddingTop="10dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*">
    <TableRow>
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:text="X-Axis"
    android:gravity="center"/>
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

Graphical Layout | main.xml

# Layout after (var.2)

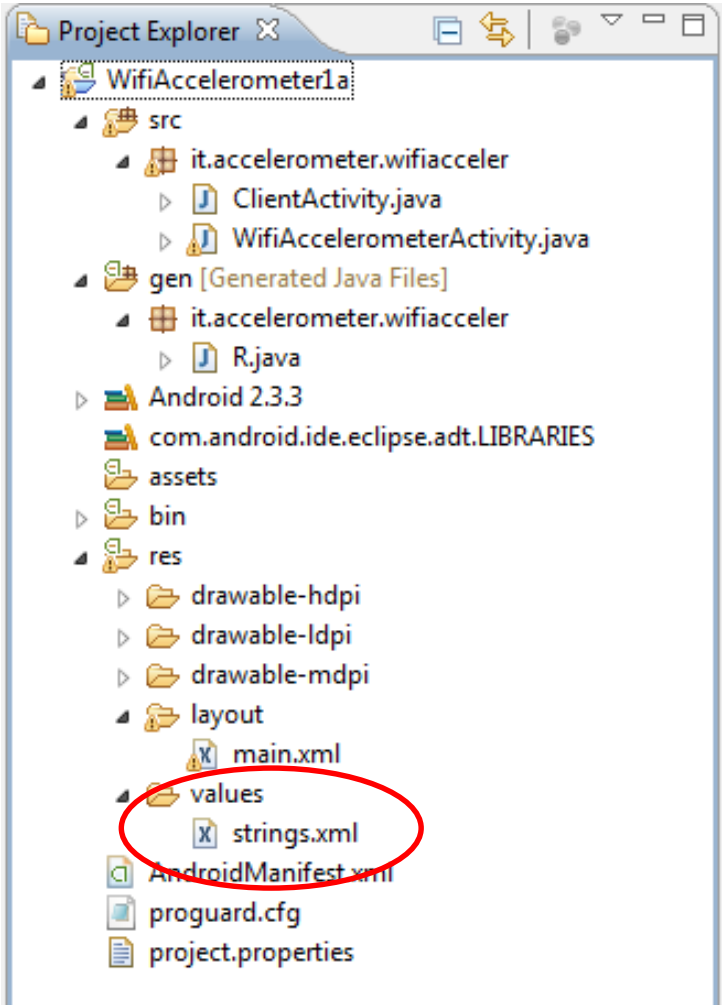



WifiAccelerometerAct | R.java | main.xml

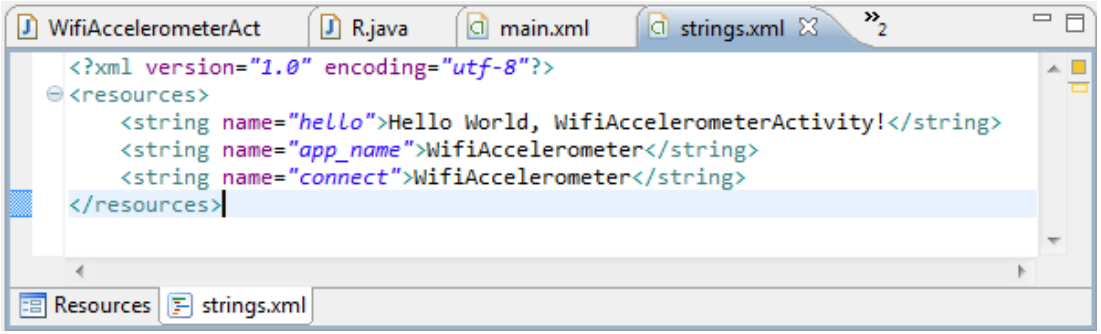
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TableLayout
    android:paddingTop="10dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*">
    <TableRow>
    <TextView
    android:paddingTop="20dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:textStyle="bold"
    android:gravity="center"
    android:text="Shaker"/>
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="14sp"
    android:text="X-Axis"
    android:gravity="center"/>
    <TextView
    android:layout_width="wrap_content"
```

Graphical Layout | main.xml

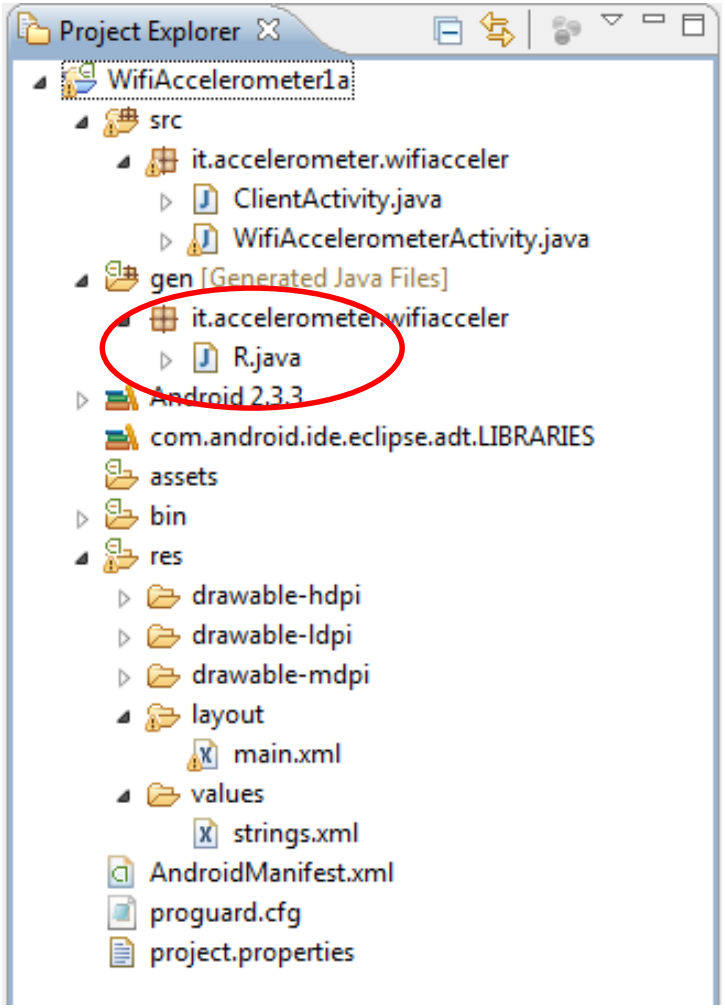
# Project Files: Values




- Folder containing strings constant value

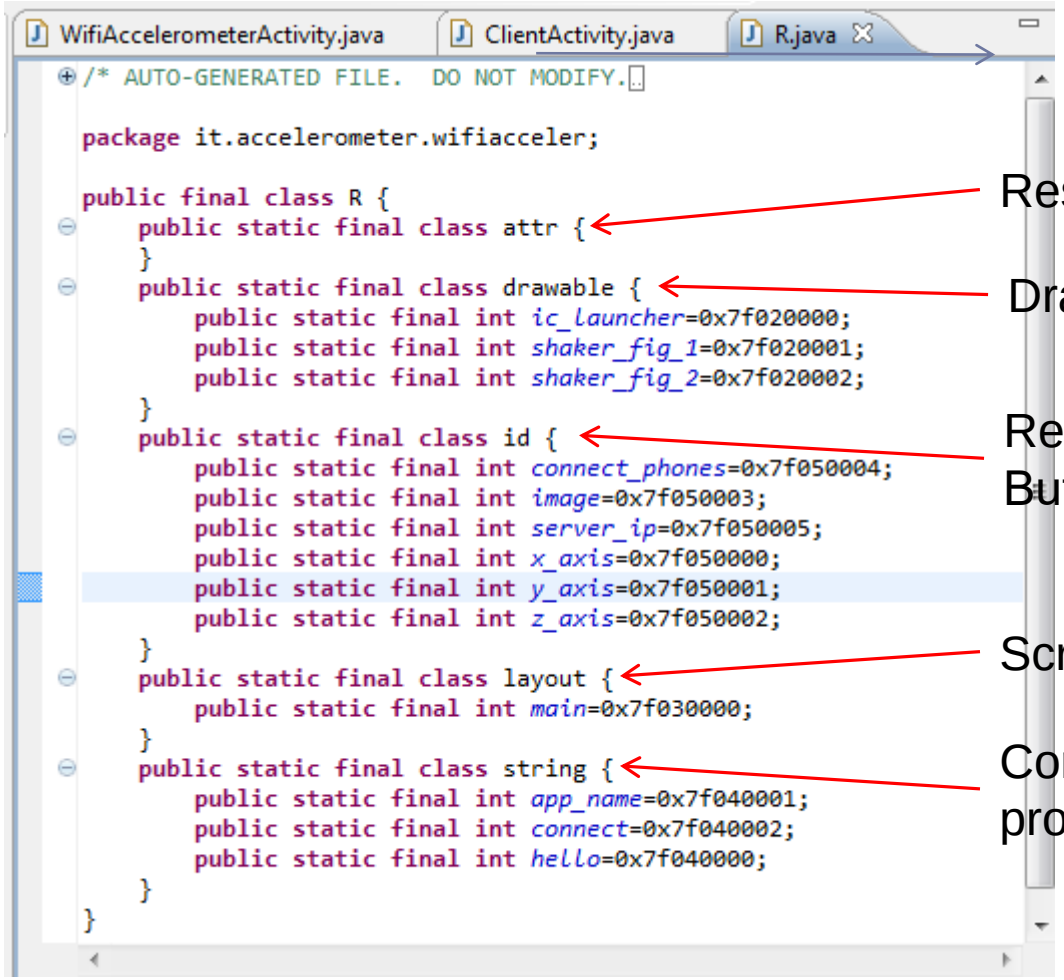


# Project Files: R.java




- R.java file, an index of all resources defined in the file

# R.java




Resource such as animation

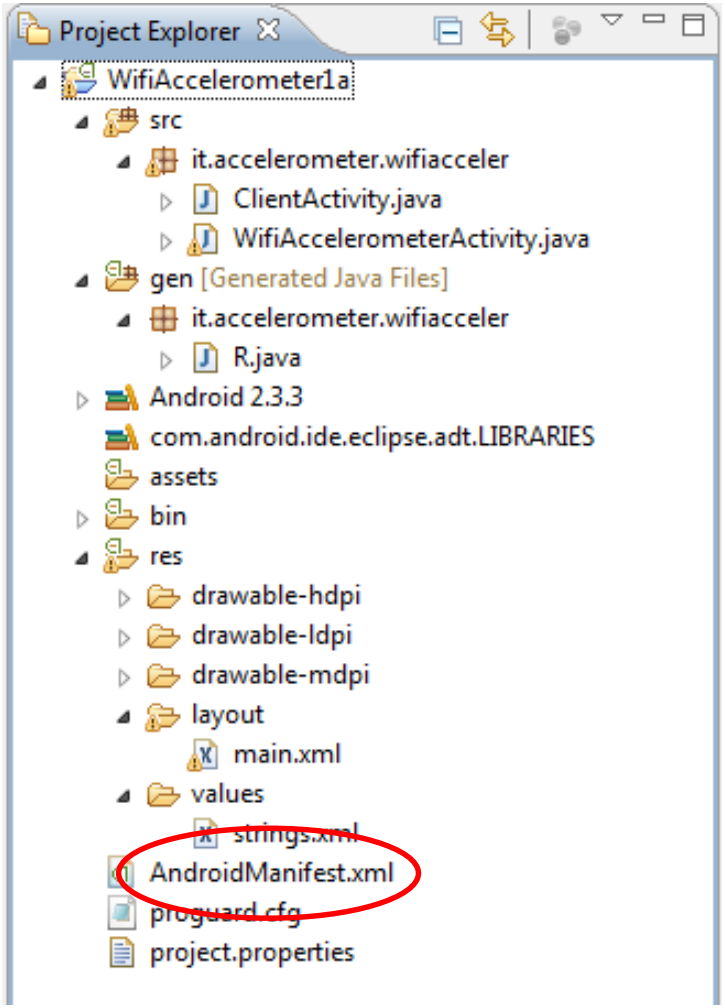
Drawable resource such as images

Resource such as Text View, Button… active stuff

Screen view of the Activity

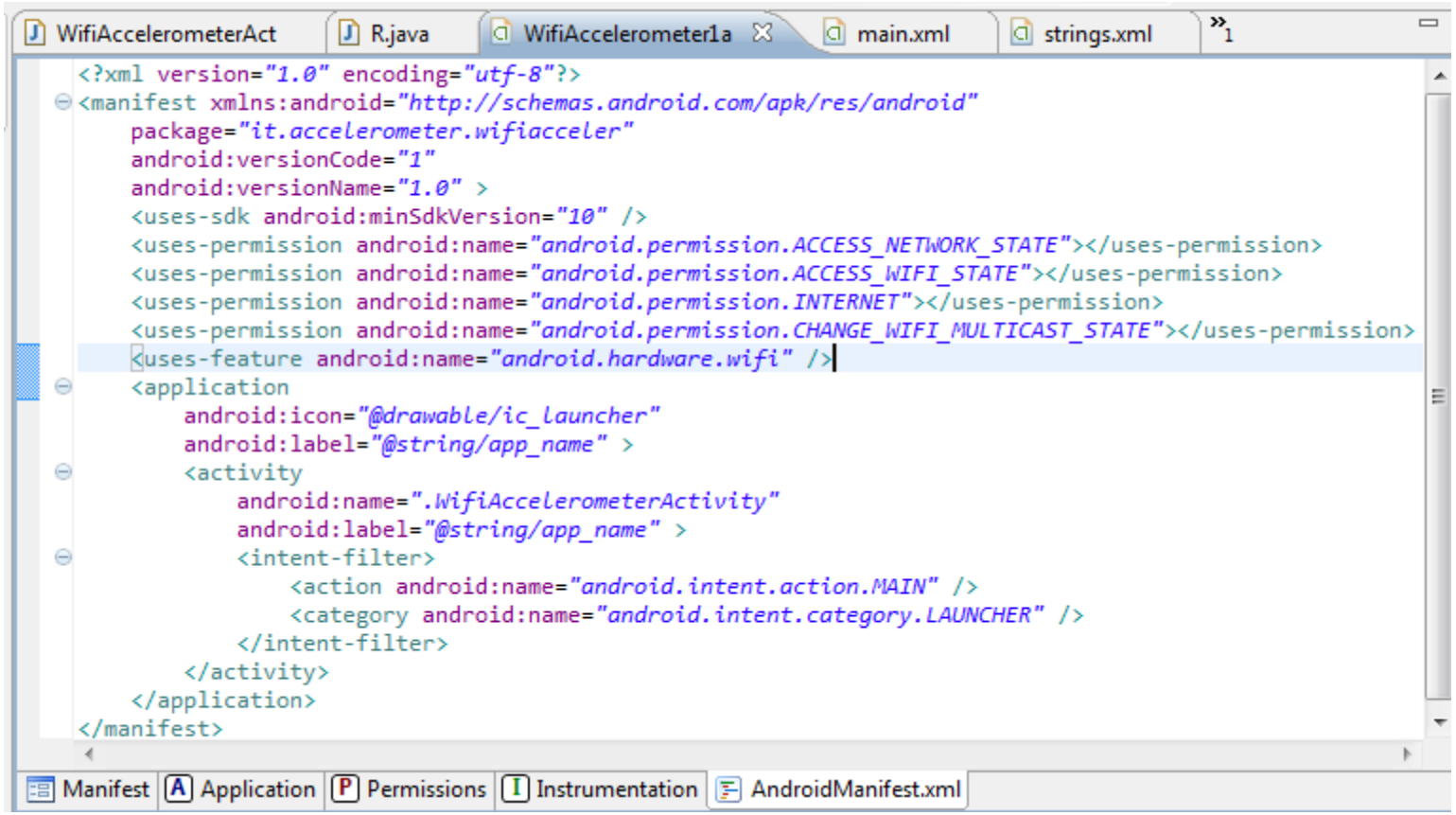
Constant String of the project

# Project Files: Manifest



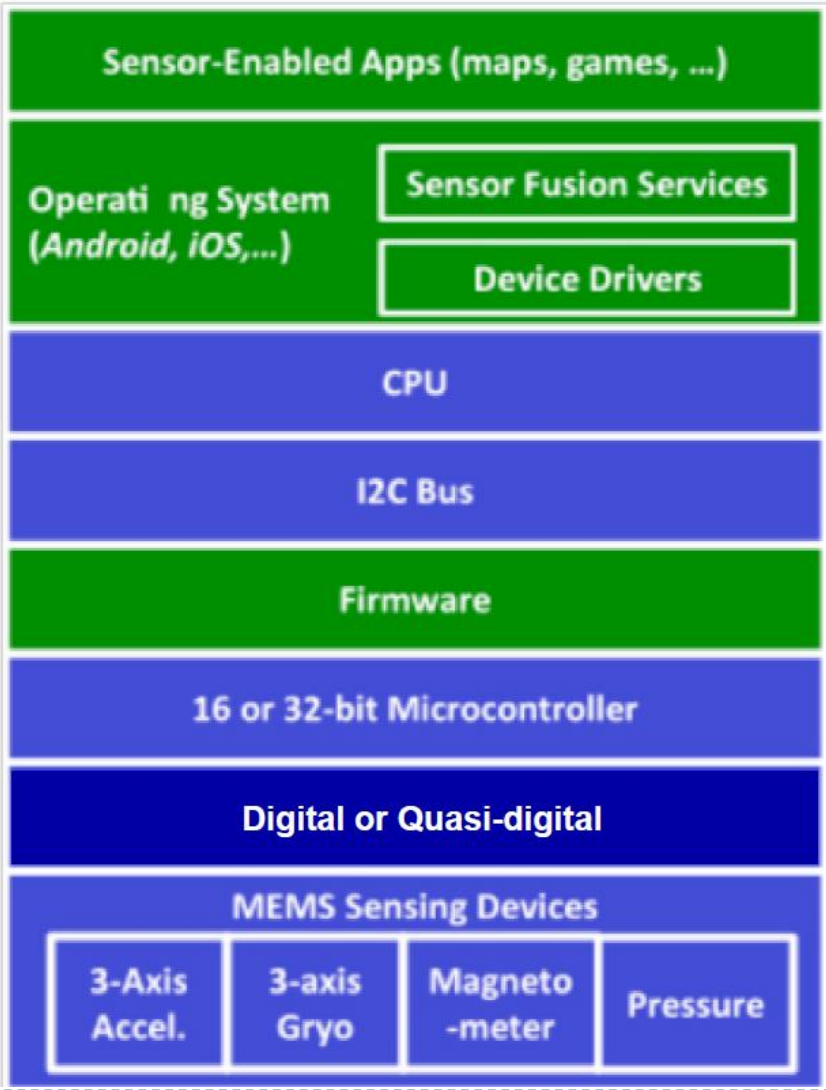

- Describe the essential information about the application to the Android system

# Manifest




```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.accelerometer.wifiacceler"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"></uses-permission>
    <uses-feature android:name="android.hardware.wifi" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".WifiAccelerometerActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

# Sensor layers in smartphone



# Sensor Categories

- HW based sensors
  - Physical component built into the device
  - The data is directly acquired measuring specific environmental properties
- SW based sensors
  - Emulate a HW based sensor behavior
  - The data are derived from one or more HW based sensors

# Sensor typologies

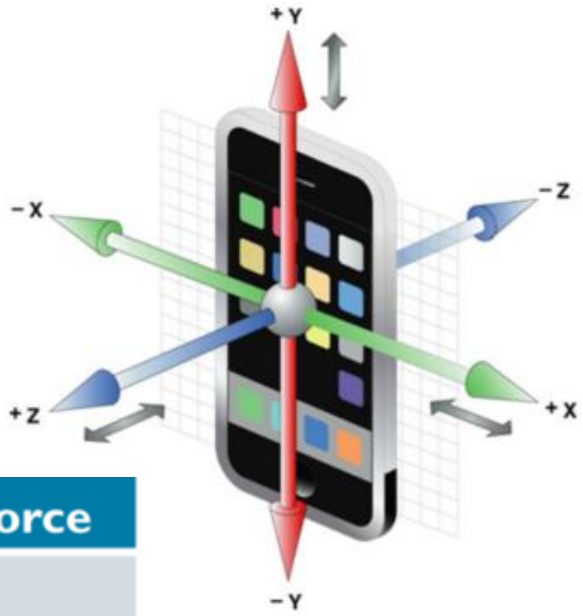
- Motion sensors
  - e.g. acceleration, rotation
- Position sensors
  - e.g. GPS, proximity
- Enviromental sensors
  - e.g. light, temperature, sound

# Sensor List

| Sensor | Function Type | Software-based or Hardware-based |
|---|---|---|
| Accelerometer | Motion Sensor | Hardware-based |
| Gyroscope | Motion Sensor | Hardware-based |
| Gravity | Motion Sensor | Software-based |
| Rotation Vector | Motion Sensor | Software-based |
| Magnetic Field | Position Sensor | Hardware-based |
| Proximity | Position Sensor | Hardware-based |
| GPS | Position Sensor | Hardware-based |
| Orientation | Position Sensor | Software-based |
| Light | Environmental Sensor | Hardware-based |
| Thermometer | Environmental Sensor | Hardware-based |
| Barometer | Environmental Sensor | Hardware-based |
| Humidity | Environmental Sensor | Hardware-based |

# Accelerometer

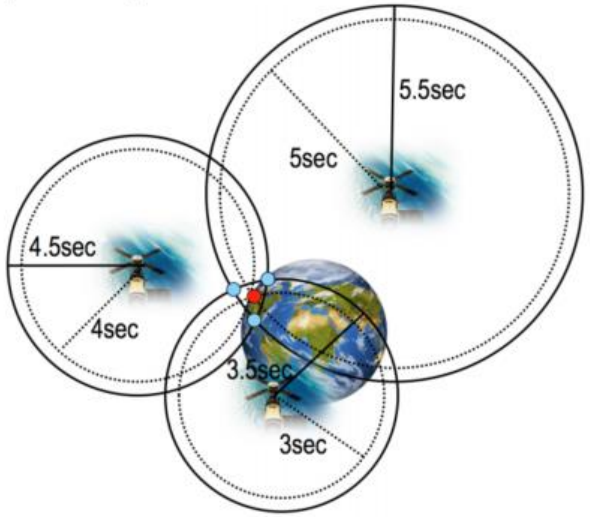
- Typically usages
  - screen orientation
  - inclination for game input
  - vibrations measurements




| Example | G Force |
|---|---|
| Standing on earth at sea level | 1g |
| Bugatti Veyron from 0 to 100 km/h (2.4s) | 1.55g |
| Space Shuttle, maximum during launch and reentry | 3g |
| Formula 1 car, peak lateral in turns | 5-6g |
| Death or serious injury | 50g |
| Shock capability of mechanical Omega watches | 5000g |

# GPS

- Typically usages:
  - Location
  - Lateration/Triangulation of cell towers or wifi networks (with database of known locations for towers and networks)
  - Location of associated cell tower or wifi network
  - Need connect to 3 satellites for 2D positioning, 4 satellites for 3D positioning
  - More visible satellites increase precision
  - Typical precision 20-50m
  - Maximum precision: 10m



# Gyroscope

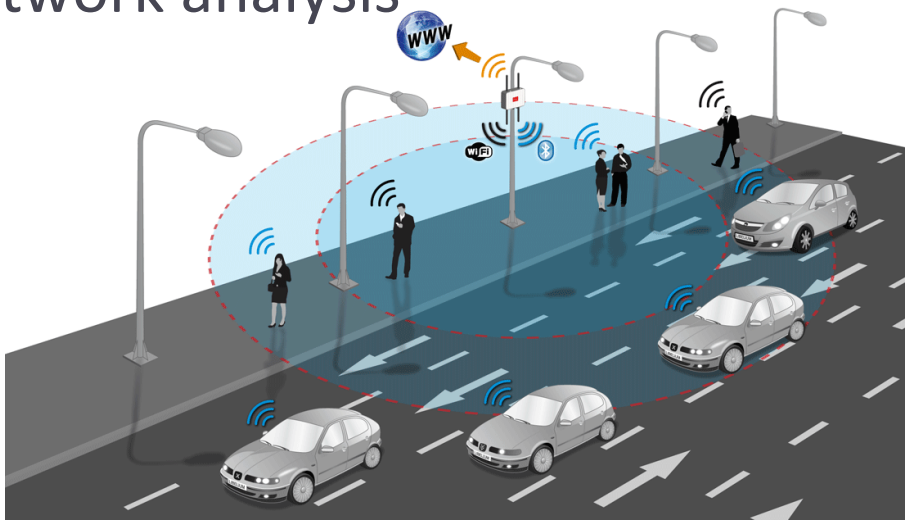
- Usages:
  - Measurements of rate of rotation (angular speed)
- 3 values related to the axes
  - Pitch value (rotation around X axis)
  - Roll value (rotation around Y axis)
  - Yaw value (rotation around Z axis)

# Why Android Apps?

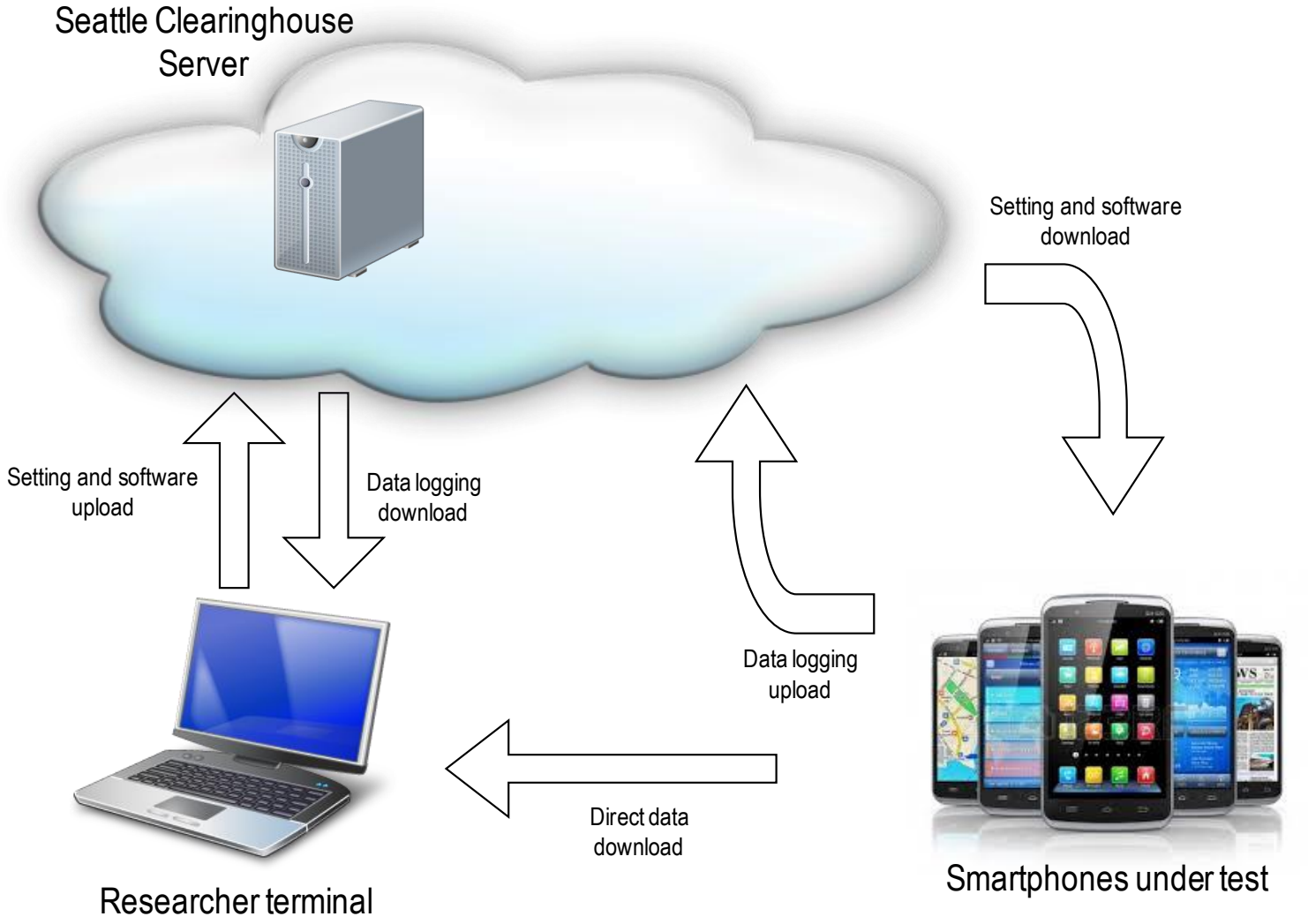
- Cloud computing – sharing computational resources
  - Seattle Project

**Seattle**
Open peer-to-peer computing

- Data from smartphones sensors useful for multi-disciplines analysis e.g.
  - GPS and accelerometers for behavior analysis
  - WI-FI elaboration for network analysis
- Sensibility Testbed (cloud sensing) is yet in development

# Sensibility Testbed – Work in progress



# Sensibility Testbed – noise issue

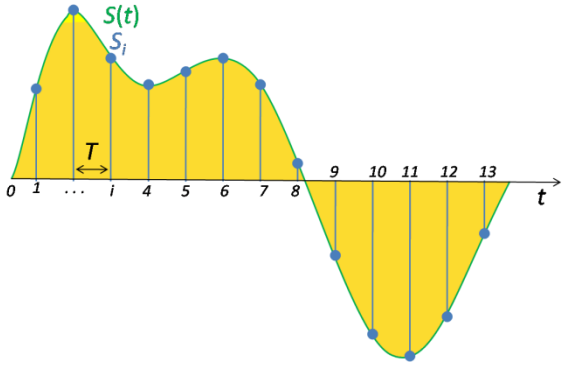
- Residential zone close to night life locals or concert place (well-known 'Movida' in Milan)
- Industrial activities (daily and overnight)

# Sensibility Testbed – advantages

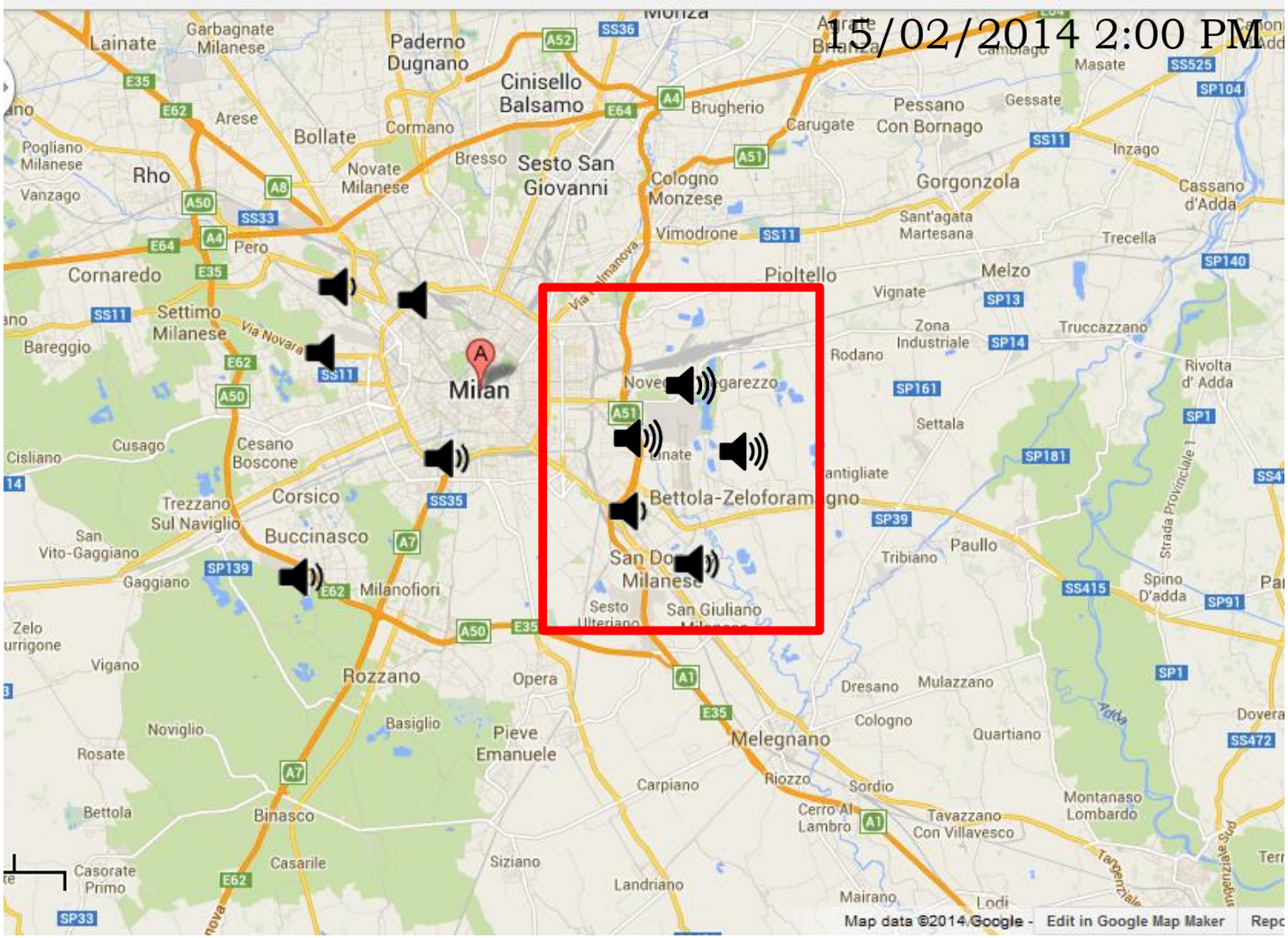
- No need the direct present of officer

- Can prevent citizens complaint

- Provide multiple measurements distributed on all the area at different time periods



# Sensibility Testbed – advantages

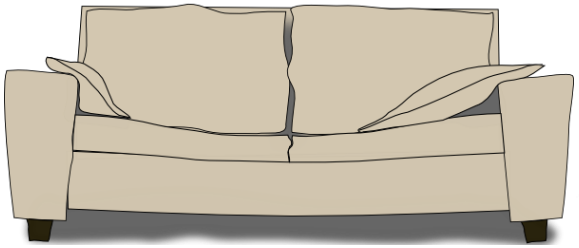


# Athlet monitoring

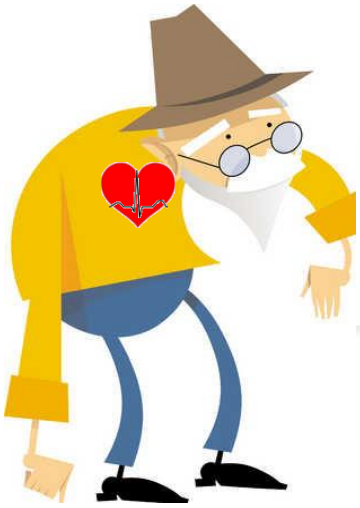

***Geo and time tagging***

# Patient monitoring

***Home continuous monitoring***

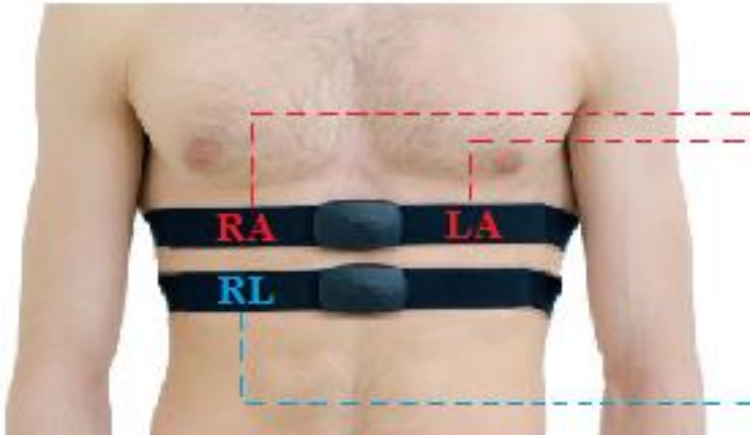


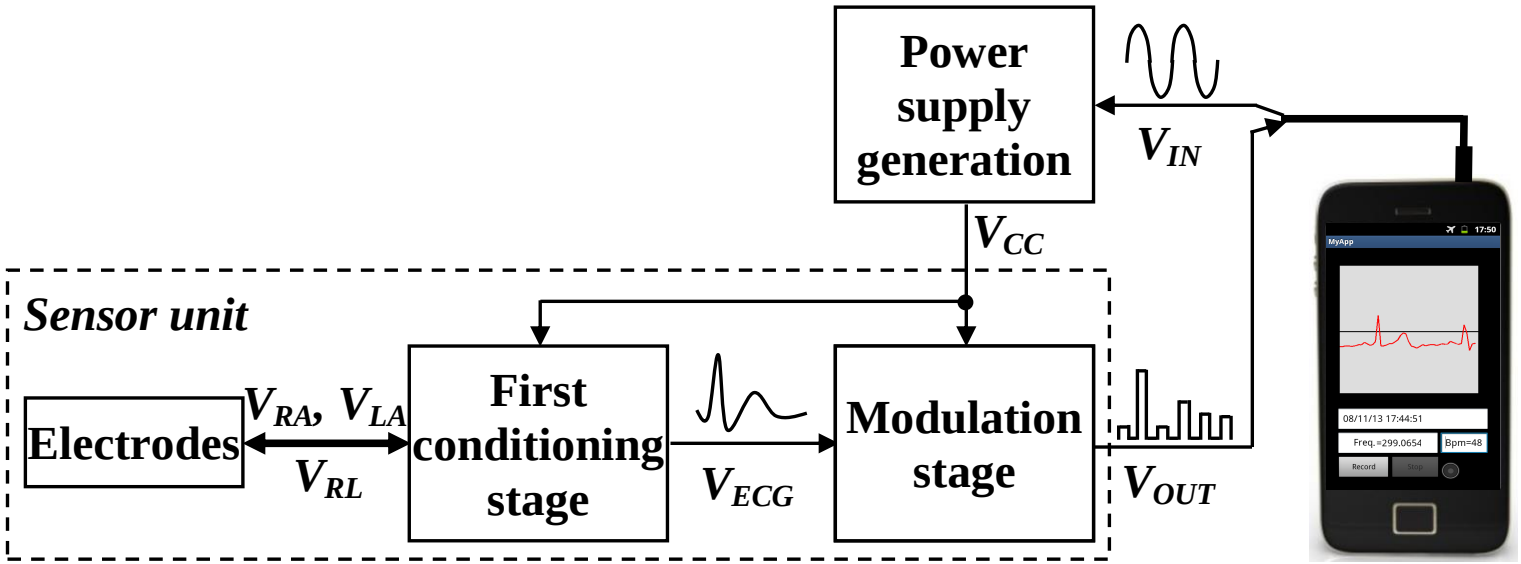
***Remote continuous monitoring***



***Prompt assistance***

# System


RA
LA
RL
Signal ($V_{RA}$, $V_{RL}$)
Reference ($V_{RL}$)


Power supply generation
$V_{IN}$
$V_{CC}$
Sensor unit
Electrodes
$V_{RA}$, $V_{LA}$
$V_{RL}$
First conditioning stage
$V_{ECG}$
Modulation stage
$V_{OUT}$
MyApp
08/11/13 17:44:51
Freq.=299.0654
Bpm=48
Record
Stop

# Final Application

**ECG (smartphone) and PPG (tablet) acquisition.**
**ECG through input jack audio – PPG through Bluetooth**