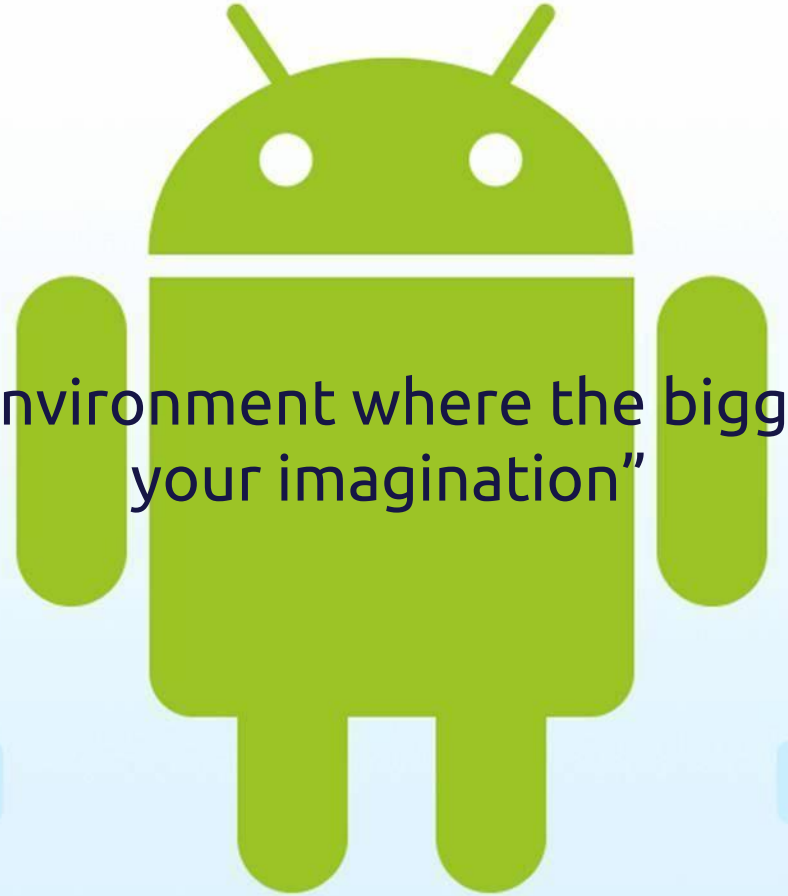
Android

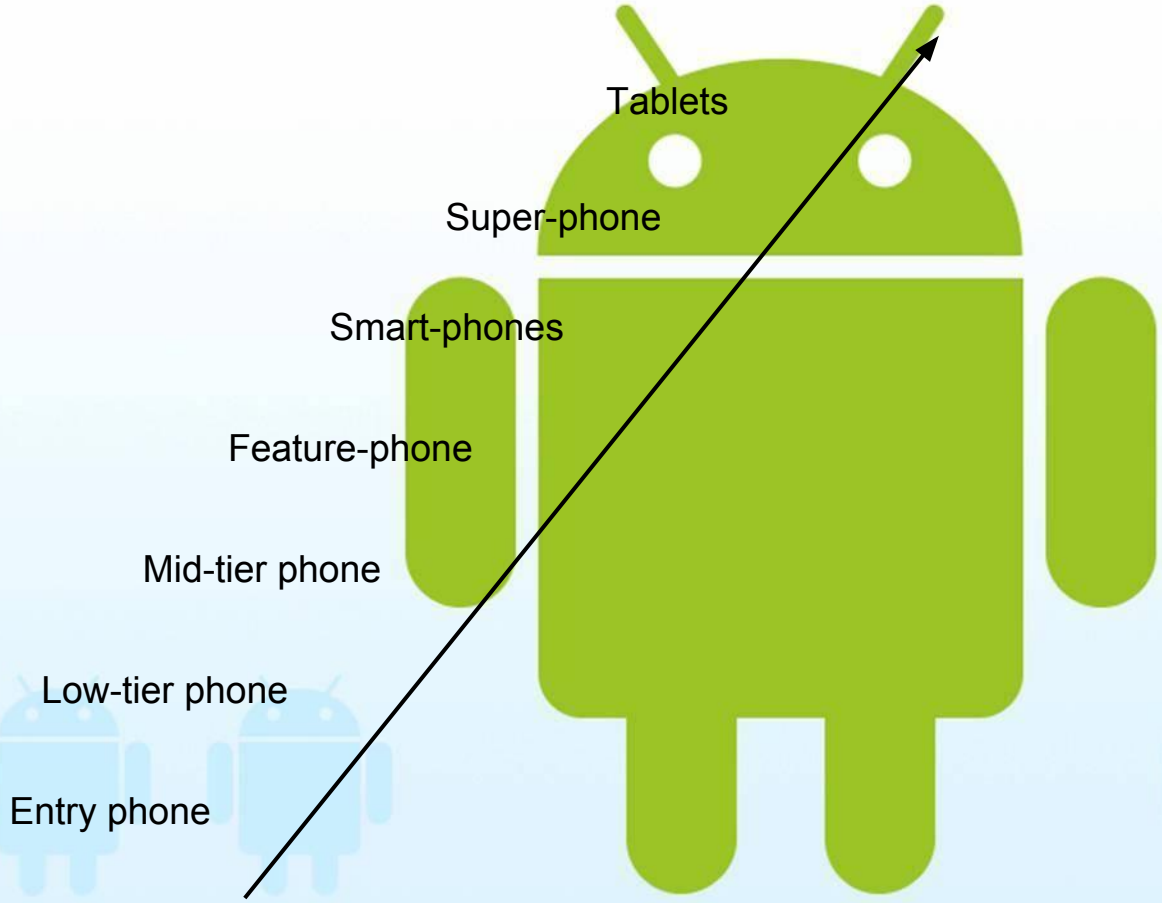


“Android is an environment where the biggest limitation is your imagination”

Android

- L'économie de la téléphonie mobile
- Le projet Android
- Overview
  - Framework
  - IHM
  - Géolocalisation
  - Réseau
  - Téléphonie
  - Stockage de données
  - Bluetooth
  - Capteurs
  - Camera
  - NFC
  - Outils de développement
  - Structure logicielle
  - Cross plateforme
- Overview (suite)
  - Market
  - Apprentissage

L'économie de la téléphonie mobile



L'économie de la téléphonie mobile

1 milliard de terminaux vendus en 2015

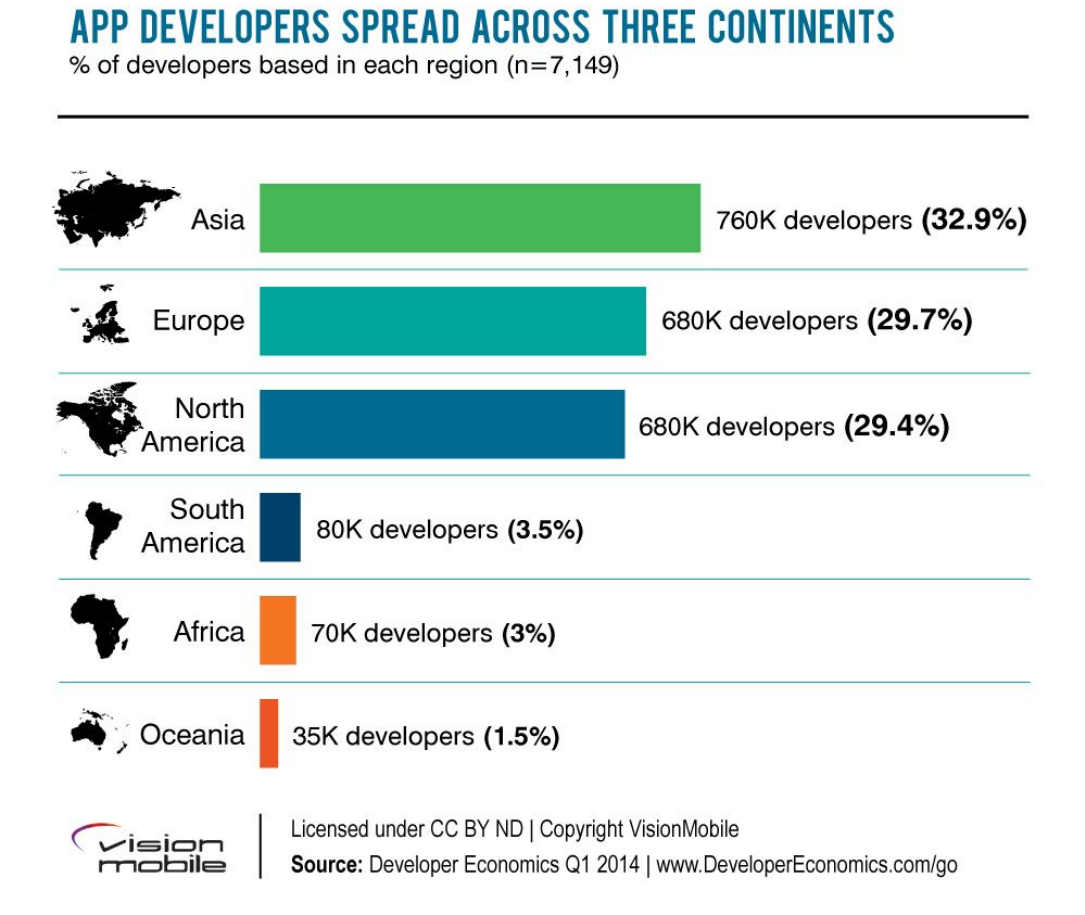
Company	4Q15 Units	4Q15 Market Share (%)	4Q14 Units	4Q14 Market Share (%)
Samsung	83,437.7	20.7	73,031.5	19.9
Apple	71,525.9	17.7	74,831.7	20.4
Huawei	32,116.5	8.0	21,038.1	5.7
Lenovo*	20,014.7	5.0	24,299.9	6.6
Xiaomi	18,216.6	4.5	18,581.6	5.1
Others	177,798.0	44.1	155,551.6	42.3
Total	403,109.4	100.0	367,334.4	100.0

http://www.gartner.com/newsroom/id/2623415

L'économie de la téléphonie mobile

Les chiffres de la téléphonie

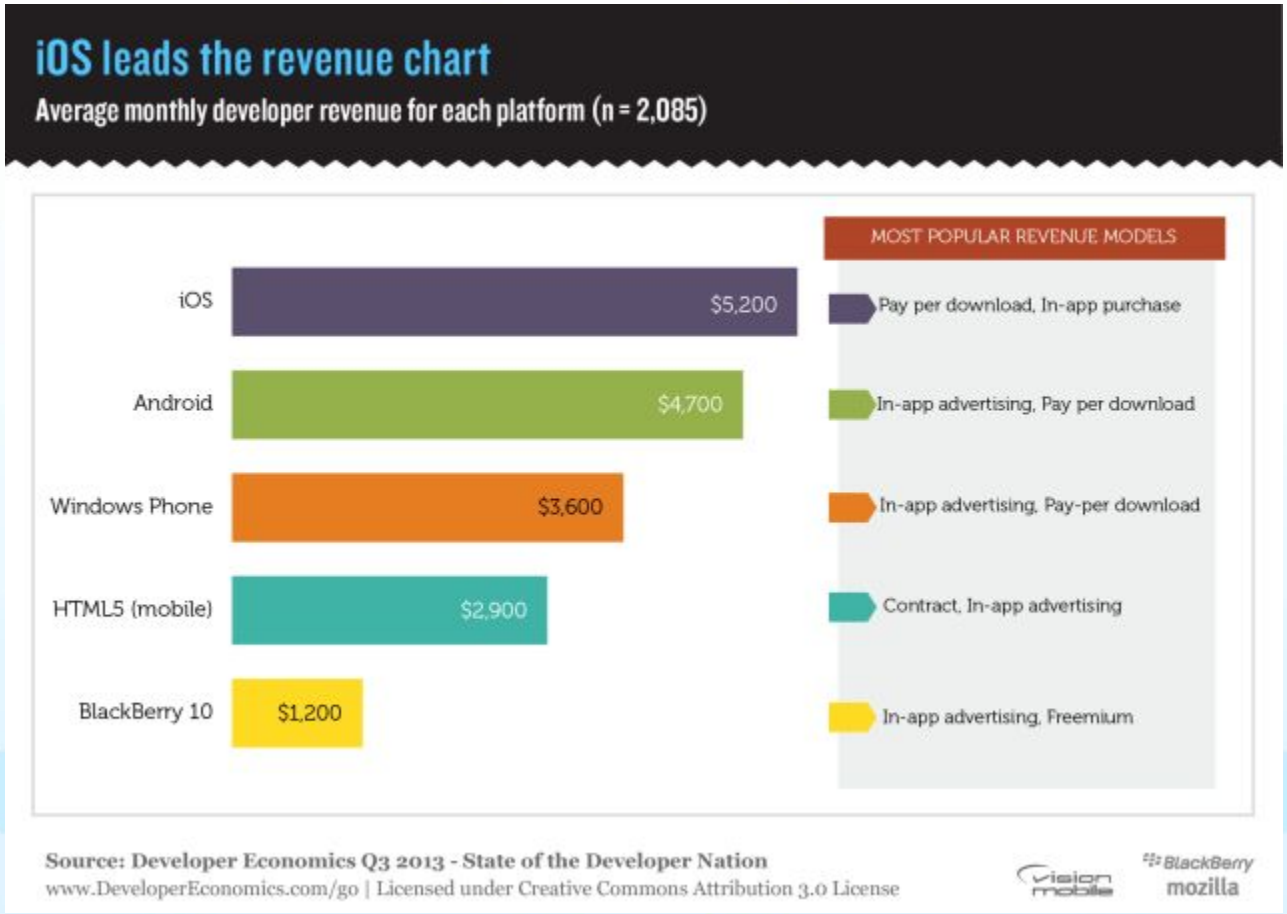
The global app economy was worth $68 billion in 2013 and is projected to grow to $143 billion in 2016. Out of a total global mobile developer population of 2.3 million individuals in 2013, Asia has the most app developer citizens at 760,000 individuals. For more forecasts on developer population, platforms, revenues and revenue models see our App Economy Forecasts 2013-2015 report.



http://www.developereconomics.com/report/q1-2014-68-billion-app-economy/

L'économie de la téléphonie mobile

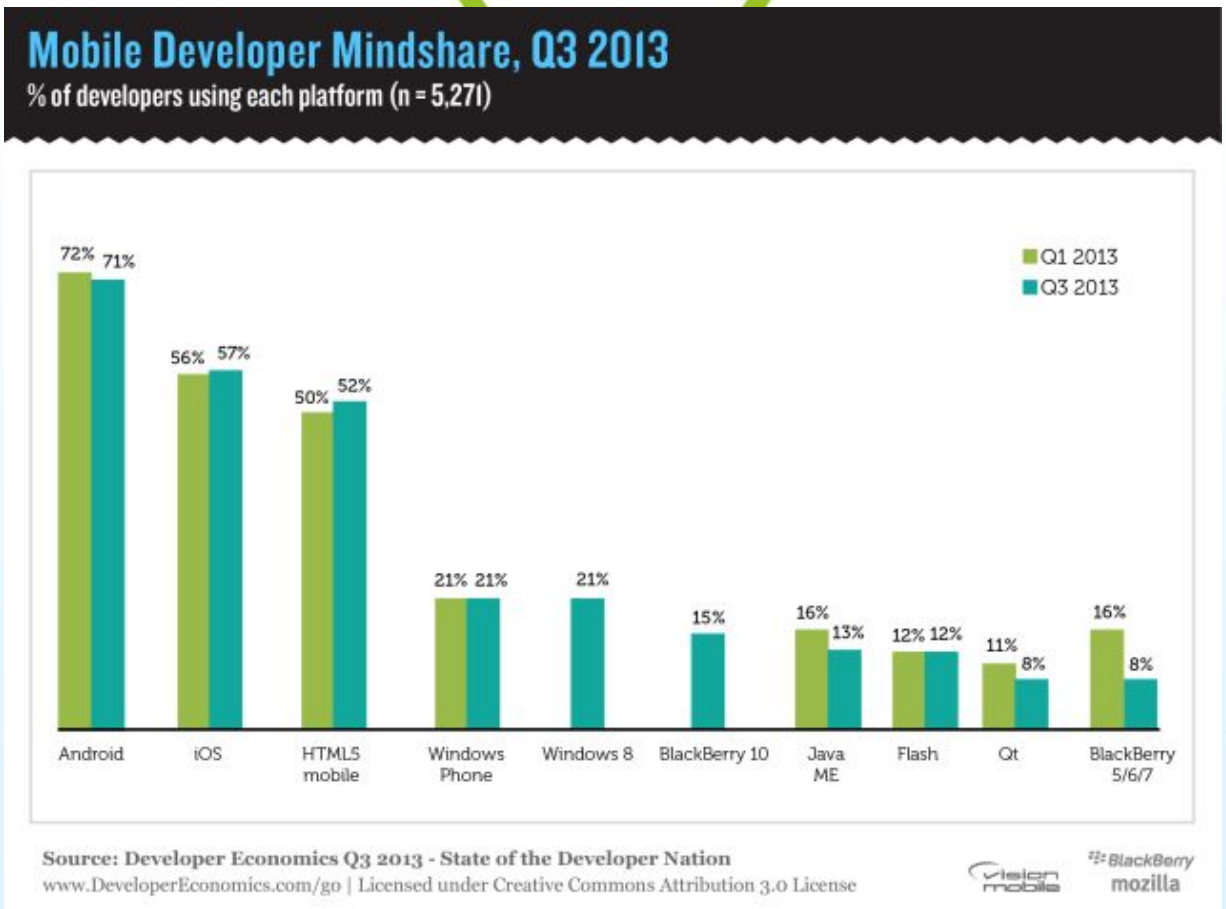
Les chiffres de la téléphonie



http://www.developereconomics.com/report/q1-2014-68-billion-app-economy/

L'économie de la téléphonie mobile

Les développeurs d'applications



http://www.developereconomics.com/platforms/

Quelques dates

- Juillet 2005 : Google achète Android Inc.
- 5 Nov 2007 : Open HandSet Alliance Google, HTC, Intel, Motorola ...
- 12 Nov 2007 : Première version Android OHA

THE OPEN HANDSET ALLIANCE

- Google et 33 autres sociétés forme l'Open Handset Alliance
- Cette alliance part d'un objectif commun : favoriser l'innovation sur les appareils mobiles
- Fournir aux développeurs un nouveau degré d'ouverture
- Accélérer les développements

THE OPEN HANDSET ALLIANCE

- Fabricants HTC, LG, Motorola, Samsung,
- Opérateurs mobiles (China Mobile Communications, KDDI, DoCoMo, Sprint / Nextel, T-Mobile, Telecom Italia, Telefonica),
- Semi-conducteurs (Audience, Broadcom, Intel , Marvell, NVidia Qualcomm, SiRF, Synaptics),
- Logiciels (Ascender, eBay, Esmertec, Google, LivingImage, LiveWire, Nuance, Packet Video, SkyPop, SONiVOX),
- Les sociétés de commercialisation (Aplix, Noser, TAT, Wind River).

Constat : FRAGMENTATION LOGICIEL

- Chaque marque a un environnement d'application différents
- Assemblage de dizaines de morceaux de logiciel tiers pour créer une plate-forme de téléphone
- Java était censée changer cette situation, avec J2ME et les recommandations sans fil Java : CDC, CLDC, MIDP, JTWI, MSA, etc
- Les smartphones sont construits avec J2ME et des extensions fournisseurs qui limitent la portabilité des applications
- Linux à ce jour a un noyau open source (licence GPL), mais conserve une couche (cadre d'application, framework multimédia, applications propriétaires)

Le projet Android

Clés du succés

- ANDROID est gratuit : Apache V2, http://www.apache.org/licenses/LICENSE-2.0.txt

- L'utilisation de la licence Apache est essentielle, car il permet aux fabricants de combinés de prendre le code Android, le modifier selon leurs besoins, le communiquer ou pas à la communauté open source

- Le personnage Android lui-même est sous licence "creative commons by (3.0)« , usage libre

- “If Google didn’t act, we face a draconian future. One man, one company, one device would control our future, If you believe in openness and choice, welcome to Android.”

Clés du succés

- Google s’appuie sur sa notoriété
- Les constructeurs abandonnent les OS couteux pour ANDROID
- Les téléphones mobiles d'aujourd'hui sont très puissants = ordinateurs de poche
- Processeur basse consommation (ARM multi-coeurs)
- Convergence des applications grand public
- Google Market

Le projet Android

Clés du succés

- Le succés provient de la portabilité du code : votre application est toujours la même indépendement du matériel

- La VM a été optimisée pour les systèmes à faible emprunte mémoire et compilation du code

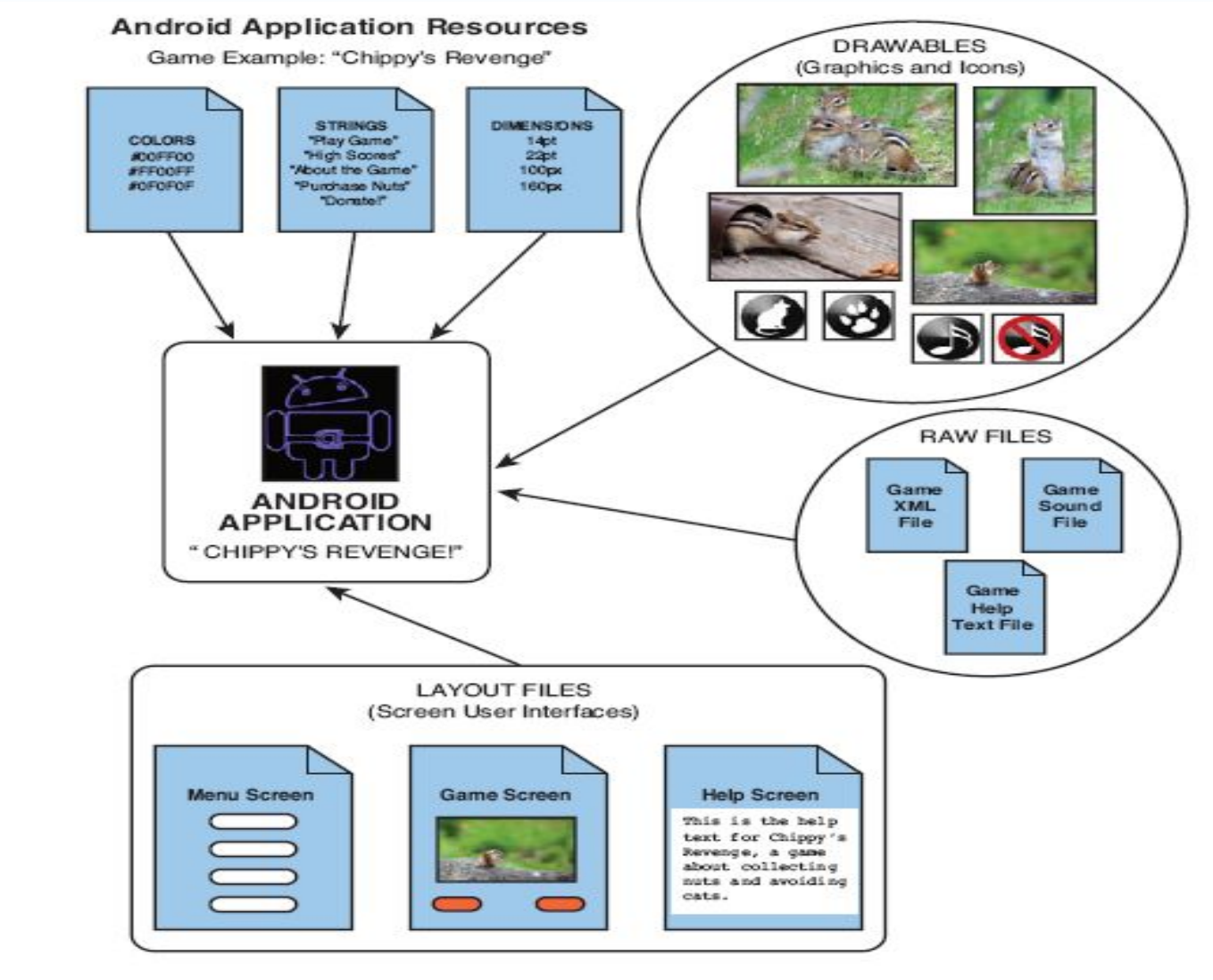
L'ancienne VM ANDROID s'appellait DALVIK après qu'un ingénieur de Google y soit parti en vacances, puis maintenant ART (Android Runtime).

Overview : Framework

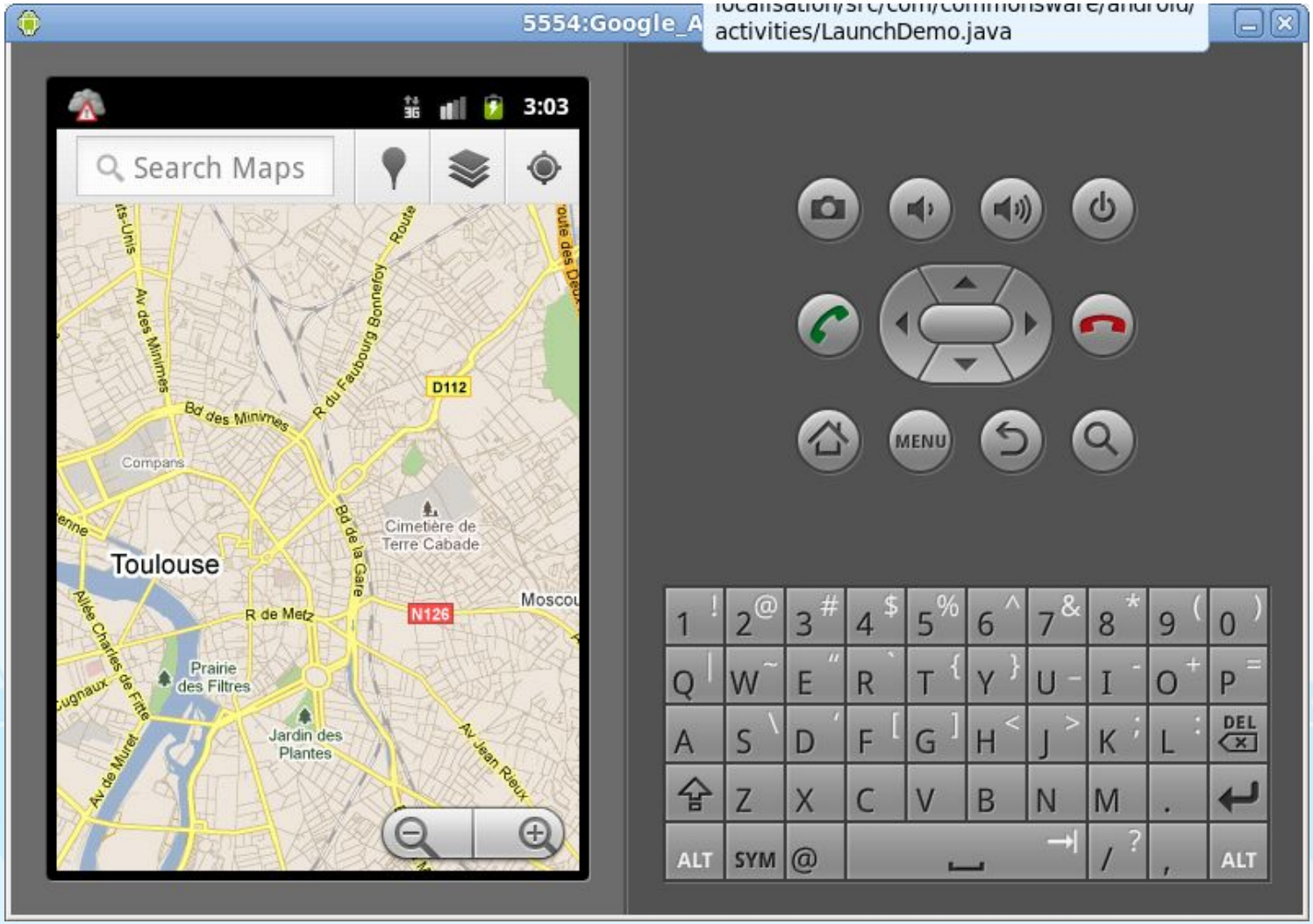
FRAMEWORK ANDROID : orienté composants logiciels

IHM	• XML, OpenGL
ACTIVITES	• Logiciel utilisateur
SERVICE	• Fourni un service
BROADCAST RECEIVER	• Réagi en fonction des événements
Content provider	• Fourni/publie des données aux applications

Overview: IHM Principes de base



Overwiew : Géolocalisation

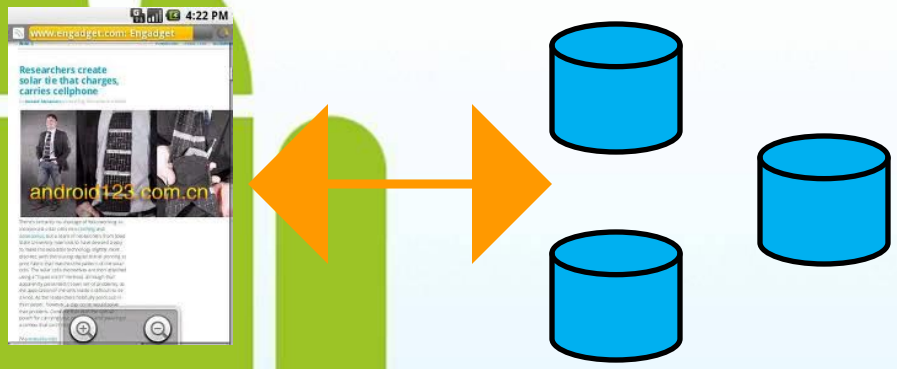


Overview : Réseau

HTTP / WEBKIT communication (java.net / android.net)

HTTP GET permet de récupérer des données au format XML ou JSON (http://www.json.org/ for an overview)

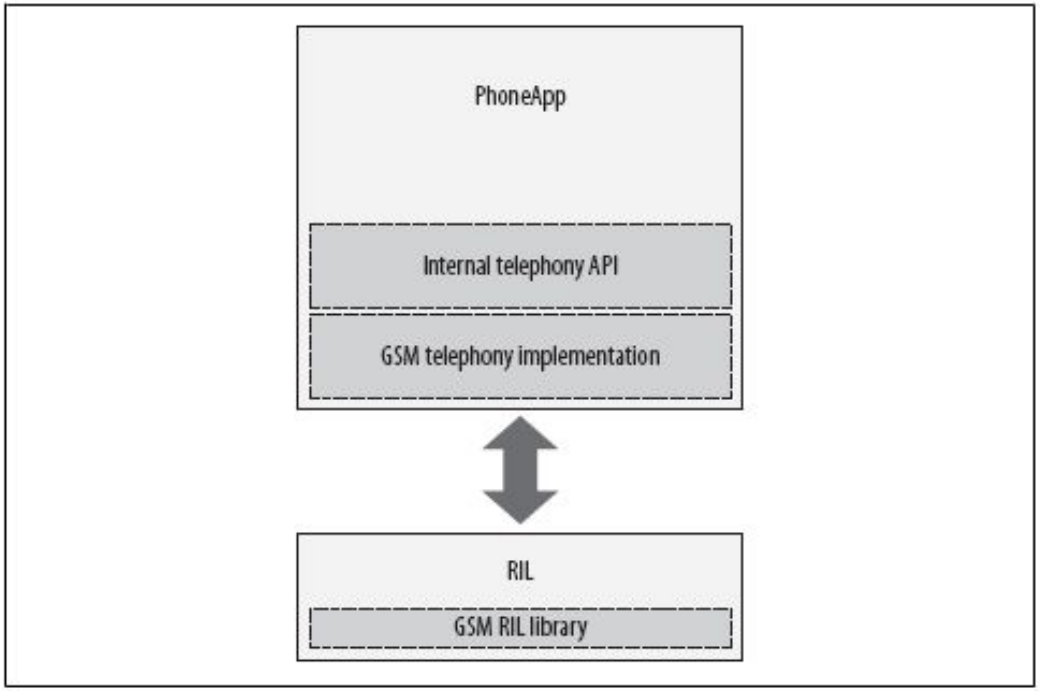
Android includes three XML parsers: the traditional W3C DOM parser (org.w3c.dom), a SAX parser (org.xml.sax), and the XML pull parser. It also has a JSON parser (org.json).



To search any topic, the topic just needs to be appended to the query. For example, to search information on the National Basketball Association (NBA), the following query returns JSON data:

http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=NBA

Overview : Téléphonie (voix, data)



Les couches de téléphonie

This package is layered over an implementation of telephony internals for a particular telephony technology, such as GSM or CDMA. That layer, in turn, communicates with a Radio Interface Layer (RIL) that is implemented as a daemon in Android.

Overview : Bluetooth

Bluetooth from the **IEEE standard 802.15.1** is an open, wireless protocol for exchanging data between devices over short distances.A common example is from a phone to a headset, but other applications can include proximity tracking.To communicate between devices using Bluetooth, four steps need to be performed:

1. Turn on Bluetooth for the device.
2. Find paired or available devices in a valid range.
3. Connect to devices.
4. Transfer data between devices.

http://developer.android.com/guide/topics/wireless/bluetooth.html

Overview : Les données

Persisting data to a database

- One nice convenience that the Android platform provides is the fact that a relational database is built in.

- Android uses SQLite (open-source, stand-alone SQL database)

- SQLite doesn't have all of the features of larger client/server database products, but it does cover just about anything you might need for local data storage. SQL usage in general : CREATE, INSERT, UPDATE, DELETE, and SELECT

- Any databases you create will be accessible by name to any class in the application, but not outside the application.

http://www.sqlite.org/

Overview : Les données

External storage via an SD card

One of the advantages the Android platform provides over some other similar device competitors is that it offers access to an available Secure Digital (SD) flash memory card.

Flash memory is a non-volatile computer storage chip that can be electrically erased and reprogrammed.
Flash memory is non-volatile, meaning no power is needed to maintain the information stored in the chip

Overview : Camera

The Camera class is used : to set image capture settings, start/stop preview, snap pictures, and retrieve frames for encoding for video

```
public static void setCameraDisplayOrientation(Activity activity,
        int cameraId, android.hardware.Camera camera) {
    android.hardware.Camera.CameraInfo info =
            new android.hardware.Camera.CameraInfo();
    android.hardware.Camera.getCameraInfo(cameraId, info);
    int rotation = activity.getWindowManager().getDefaultDisplay()
            .getRotation();
    int degrees = 0;
    switch (rotation) {
        case Surface.ROTATION_0: degrees = 0; break;
        case Surface.ROTATION_90: degrees = 90; break;
        case Surface.ROTATION_180: degrees = 180; break;
        case Surface.ROTATION_270: degrees = 270; break;
    }

    int result;
    if (info.facing == Camera.CameraInfo.CAMERA_FACING_FRONT) {
        result = (info.orientation + degrees) % 360;
        result = (360 - result) % 360;  // compensate the mirror
    } else {  // back-facing
        result = (info.orientation - degrees + 360) % 360;
    }
    camera.setDisplayOrientation(result);
}
```

Overview : NFC

A smartphoneA smartphone or tabletA smartphone or tablet with an NFC chip could make a credit cardA smartphone or tablet with an NFC chip could make a credit card payment or serve as keycardA smartphone or tablet with an NFC chip could make a credit card payment or serve as keycard or ID card. NFC devices can read NFC tags on a museum or retail display to get more information or an audio or video presentation. (Sony & Philips)

ANDROID

- Provides access to Near Field Communication (NFC) functionality
- Allowing applications to read NDEF message in NFC tags
- A "tag" may actually be another device that appears as a tag

NDEF : NFC Data Exchange Format

Overview : Sensors

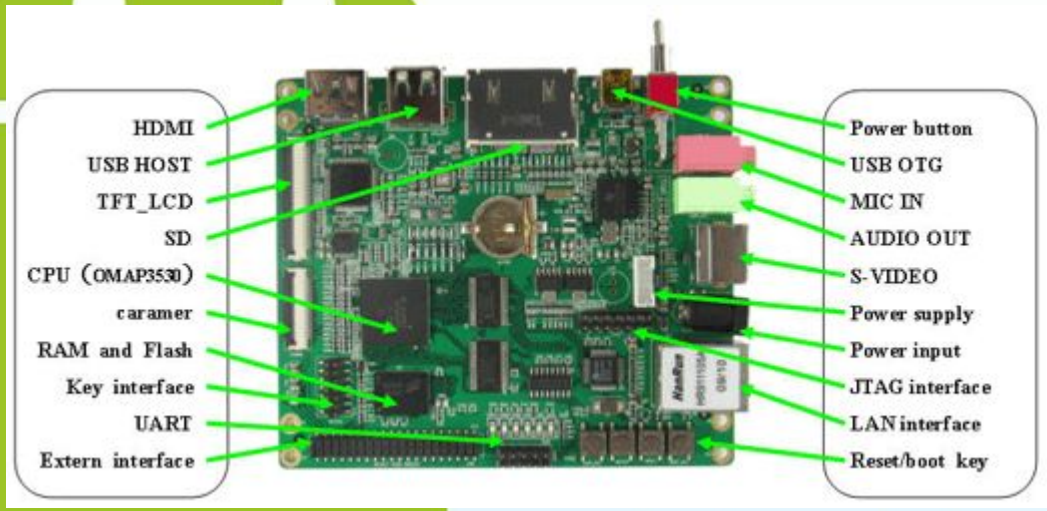
SensorManager lets you access the device's sensors

Accelerometer

Temperature

int	TYPE_ACCELEROMETER	A constant describing an accelerometer sensor type.
int	TYPE_ALL	A constant describing all sensor types.
int	TYPE_GRAVITY	A constant describing a gravity sensor type.
int	TYPE_GYROSCOPE	A constant describing a gyroscope sensor type
int	TYPE_LIGHT	A constant describing an light sensor type.
int	TYPE_LINEAR_ACCELERATION	A constant describing a linear acceleration sensor type.
int	TYPE_MAGNETIC_FIELD	A constant describing a magnetic field sensor type.
int	TYPE_ORIENTATION	*This constant is deprecated. use SensorManager.getOrientation() instead.*
int	TYPE_PRESSURE	A constant describing a pressure sensor type
int	TYPE_PROXIMITY	A constant describing an proximity sensor type.
int	TYPE_ROTATION_VECTOR	A constant describing a rotation vector sensor type.
int	TYPE_TEMPERATURE	A constant describing a temperature sensor type

Overview : Carte de développement

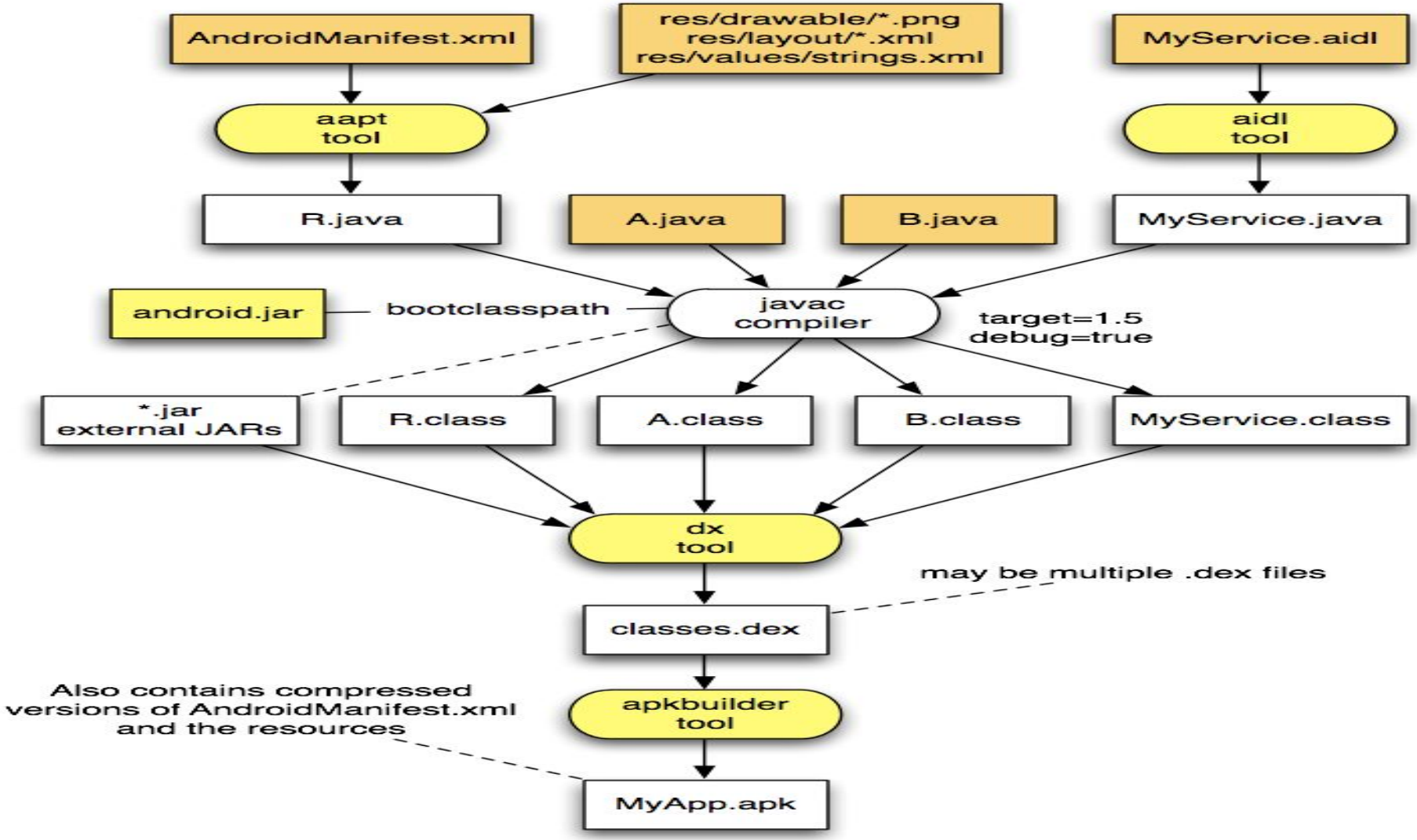


http://elinux.org/Devkit8000#Devkit8000_Board_Overview

SDK : Linux / Win32
Software Development toolKit

- **Android Development Tools** : Création et débogage des applications
- **Android Emulator** : Emulateur de téléphone
- **Android Virtual Devices** : Instance d'un telephone virtuelle
- **Dalvík Debug Monitor Service** : gestion des processus
- **Android Debug Bridge** : Installation des applications
- **Android Asset Packaging Tool** : générateur de fichier d'installation
- **Mksdcard** : Création des SdCard virtuelles
- **Dx** : Permet de réecrire le byte code java en byte code android

Overview : outils de développement



Android Studio

Android Studio is a the Android development environment based on IntelliJ IDEA. Similar to Eclipse with the ADT Plugin, Android Studio provides integrated Android developer tools for development and debugging.

V1.0.1 http://developer.android.com/sdk/installing/studio.html

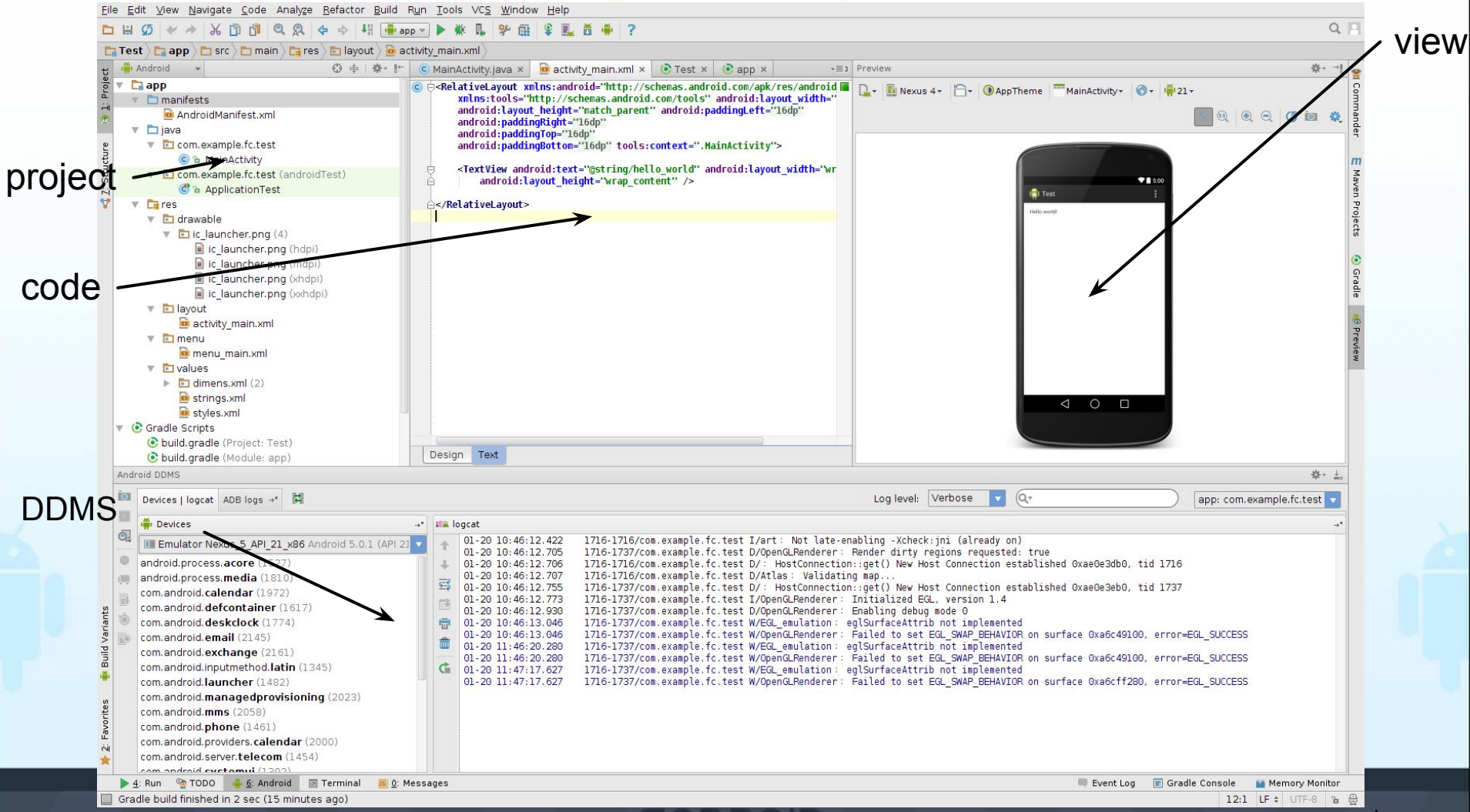
Linux

- GNOME or KDE desktop
- GNU C Library (glibc) 2.11 or later
- 2 GB RAM minimum, 4 GB RAM recommended
- 400 MB hard disk space
- At least 1 GB for Android SDK, emulator system images, and caches
- 1280 x 800 minimum screen resolution
- Oracle® Java Development Kit (JDK) 7

Tested on Ubuntu® 12.04, Precise Pangolin (64-bit distribution capable of running 32-bit applications).

Overview : outils de développement

IDE : Android Studio



*

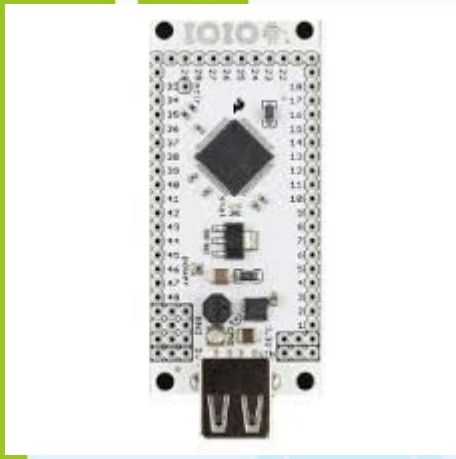
NDK : **Native Development Tools**

- Performance-critical portions of your apps in native code (rev. 5)
- Introduced to support the development of games and similar applications that make extensive use of native code
- Includes a new toolchain (based on GCC 4.4.3)
- Provides a default C++ STL implementation (based on STLport)
- OpenGL ES textures
- The latest release of the NDK supports these ARM instruction sets:
  - ARMv5TE (including Thumb-1 instructions)
  - ARMv7-A (including Thumb-2 and VFPv3-D16 instructions, with optional support for NEON/VFPv3-D32 instructions)

- **When to Develop in Native Code ?**
  - The NDK will not benefit most applications !
  - Increases application complexity
  - Native code does not result in an automatic performance increase

ADK : **Accessory Development Kit**

- Développement d'accessoires électroniques compatibles Android
- Compatible carte Arduino ADK
- Compatible carte IOIO
- ....



GDK : Glass Development Kit

- Add-on au Android SDK
- Mirror API

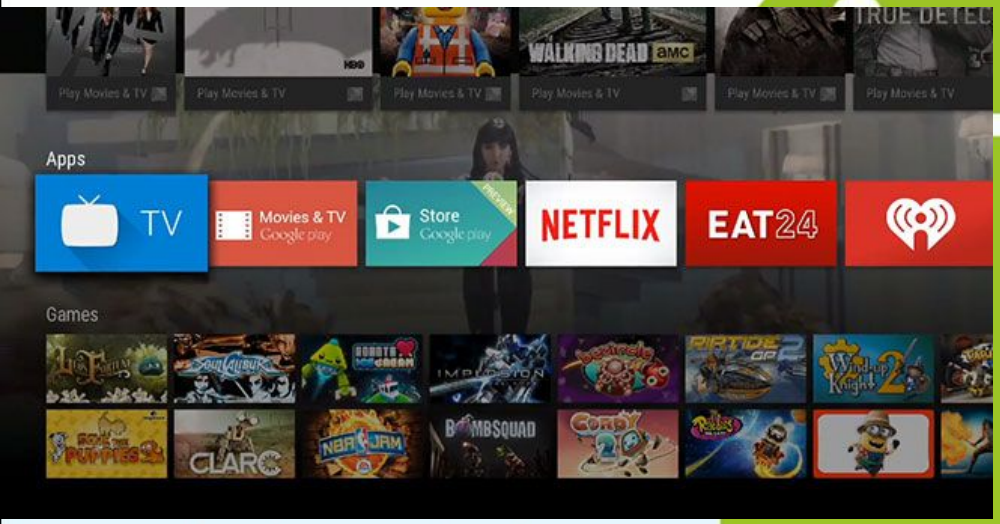
WEAR: montres, objets connectés ...

- Android dans une montre ou autre objet compatible

AUTO

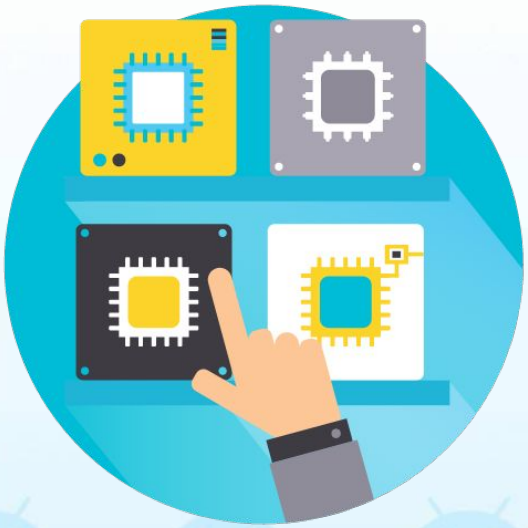
- Système de navigation ...

TV:



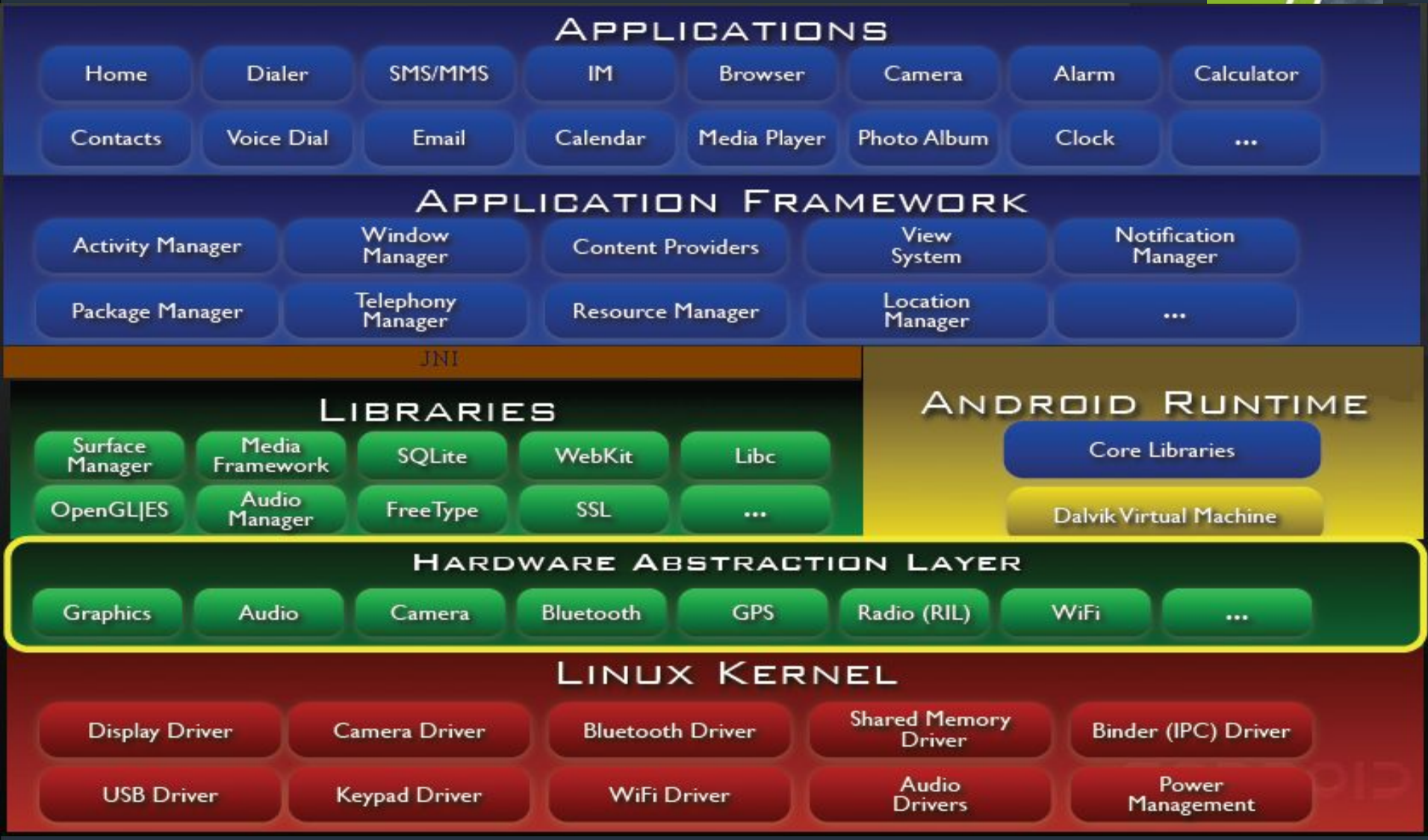
- Protocole entre TV et smart système
- API ...

IoT (Brillo):

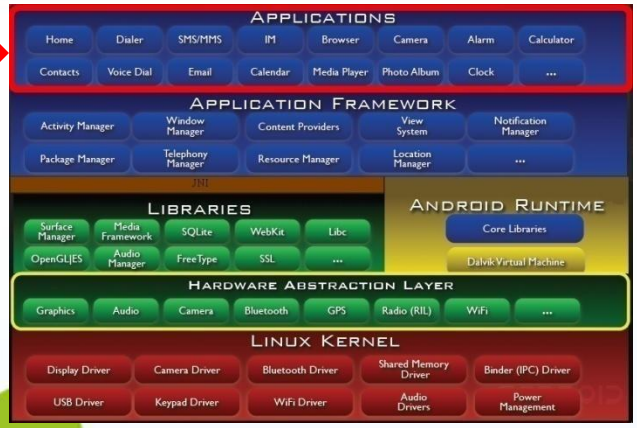


- Internet of Things

Overview Structure logicielle



Overview structure logicielle : Application



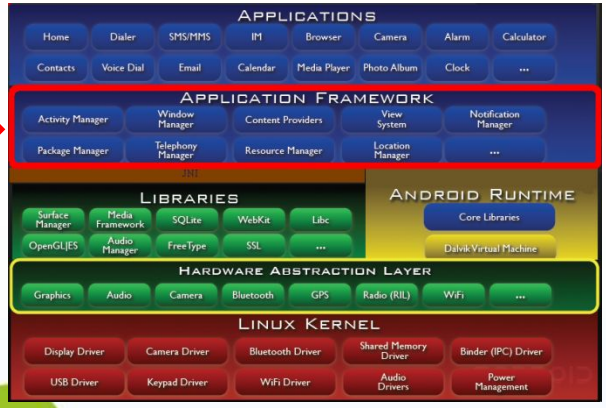
Composé d'un ensemble **d'application** de base :

- Ecrites en langage JAVA
- Fonctionne avec son propre processus linux
- Fonctionne avec sa propre machine virtuelle

Exemple:

- Camera
- Media Player
- Maps

Overview structure logicielle : Framework



Gère l'accès à toutes les couches inférieures

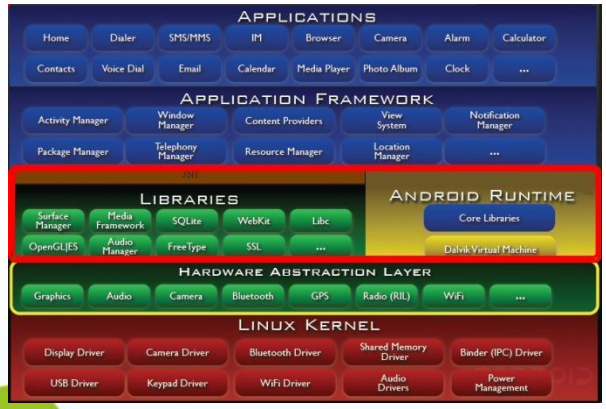
Composé de **services** logiciels et matériels :

- Ces services qui n'ont aucune interaction avec les utilisateurs
- Fournissent des API pour développer des applications : Framework
- Conçu pour simplifier la réutilisation des composants

  Chaque Framework publie ses capacités aux autres

  Pour être accessible nécessité d' avoir une permission
- Permet d'accéder aux services matériels a travers la JNI : Java Native Interface

Overview structure logicielle : Middleware



Il est constitué de deux entités distinctes :

1 : **Les Bibliothèques** : Fourni des Bibliothèques C/C++ pour l'accès à la couche noyau

Ses capacités sont fournies aux Framework a travers la JNI

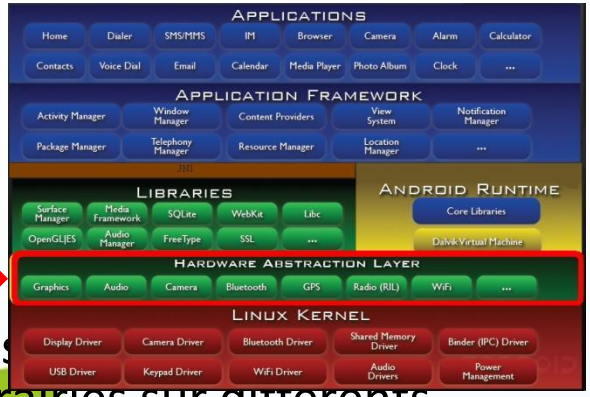
Exemple: : Media Framework , Audio Manager

2 **: L'Android Runtime** :

Constitué de la machine virtuelle et du core libraries

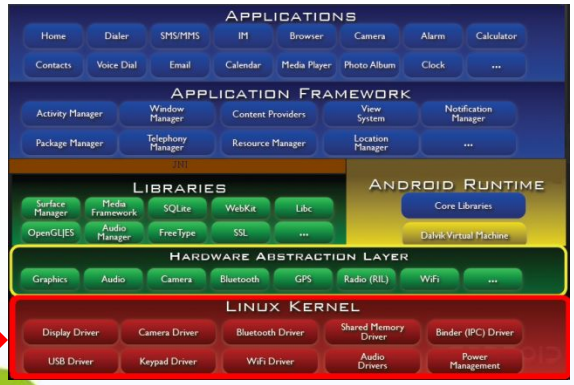
- Machine virtuelle : Dalvik Virtual Machine (DVM) transforme le bytecode Java en Dalvik bytecode
- Core Library :
  - Fourni le langage Java disponible pour les applications,
  - Reprend en partie l'API de la JSE 1.6

Overview structure logicielle : Hardware Abstraction Layer



- Se situe entre la couche MiddleWare et Kernel Linux
- Sépare la plateforme logique des interfaces matériels
- Elle fourni les interfaces que doivent implémenter les
- Son but principal est de faciliter la portabilité des librairies sur différents matériels car :
  - Pas tous les drivers ont des interfaces standardisés
  - Certains drivers sont sous licence
  - Android a des besoins spécifiques pour les drivers
  - Google propose : **Compatibility Definition Document (CDD)**

Overview structure logicielle : Linux Kernel



- Basé sur une version simplifiée du noyau **Linux 2.6 adapté aux mobiles**
  - Ressources mémoires limitées
  - Capacité CPU limitée

- Il possède une version customisée du glibc pour la compilation du code C : la bionic libc

FRAMEWORK ANDROID : composants logiciels

Développement par héritage :

```
package test.android;

import android.app.Activity;

public class test extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Héritage

Surcharge

Référence à la classe « mère »

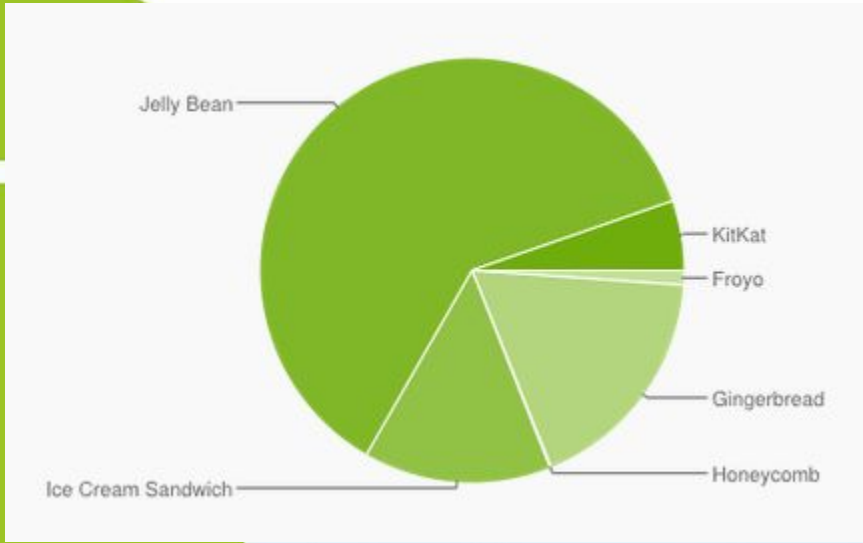
Overview structure logicielle : version

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

Pour connaître les différences des versions : http://developer.android.com/about/index.html

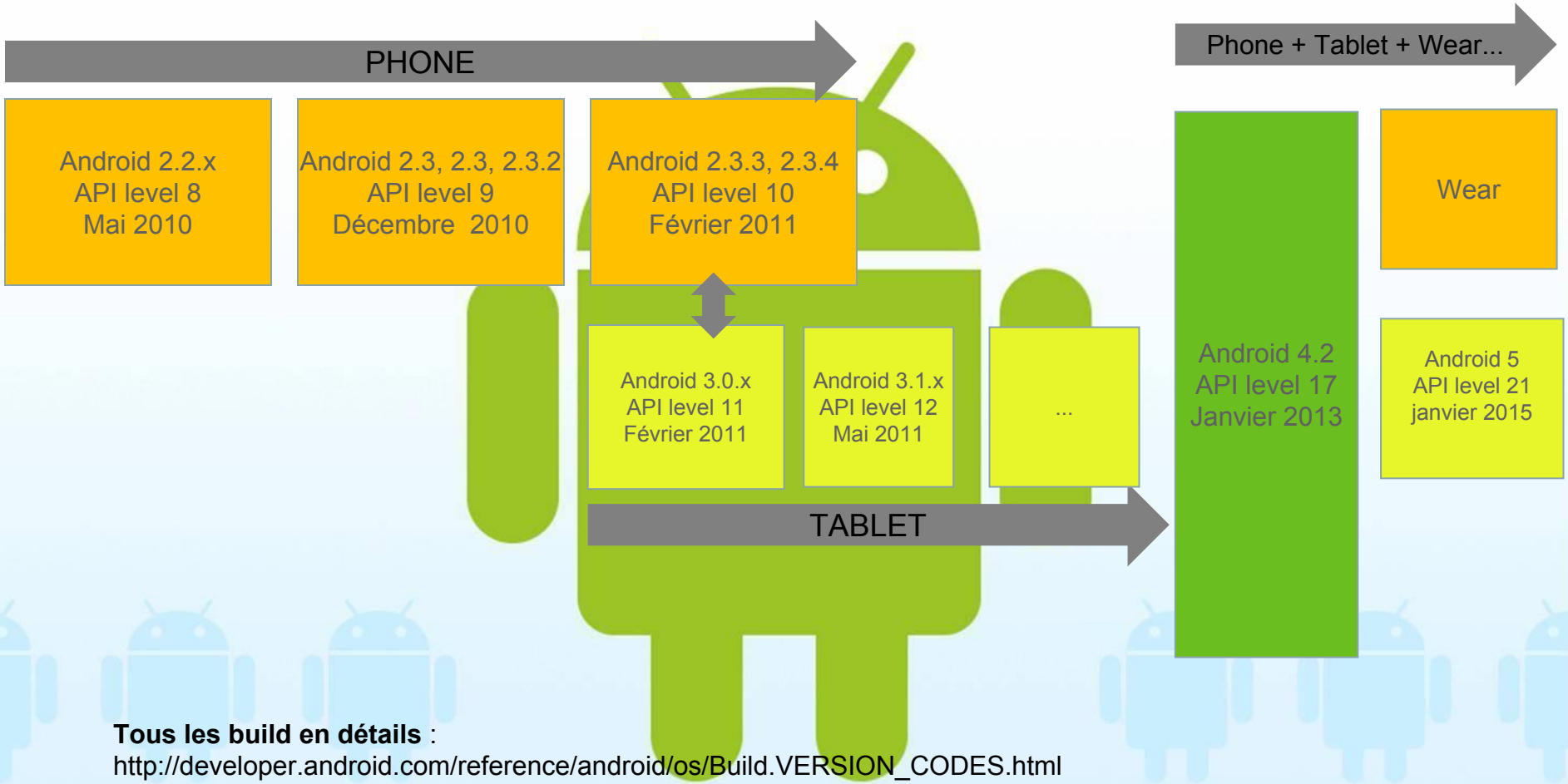
Overview structure logicielle : version

Version	Codename	API	Distribution
2.2	Froyo	8	1.1%
2.3.3 - 2.3.7	Gingerbread	10	17.8%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	14.3%
4.1.x	Jelly Bean	16	34.4%
4.2.x		17	18.1%
4.3		18	8.9%
4.4	KitKat	19	5.3%



http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels

Overview structure logicielle : version



Tous les build en détails :
http://developer.android.com/reference/android/os/Build.VERSION_CODES.html
http://developer.android.com/guide/topics/manifest/uses-sdk-element.html

Cross-Platform Native Frameworks

- Xamarin : http://xamarin.com/ Native User Interfaces, Native API Access, Native Performance
- TITANIUM : http://www.appcelerator.com/ Titanium is a commercially supported, open source platform for developing native cross-platform applications using web technologies : iPhone and Android
- Rhodes **:** http://rhomobile.com/ Rhodes is available for most major smartphones including the iPhone, Research in Motion (BlackBerry), Android, Windows Mobile, and Symbian. Using HTML, CSS, JavaScript and Ruby programming languages
- PhoneGap (http://phonegap.com/) is an open source framework for building native mobile applications using HTML, CSS, and Javascript for iPhone, Android, BlackBerry,
- CORONA : http://www.anscamobile.com/ apps, games, and eBooks with Corona SDK
- APP INVENTOR : http://mitmobilelearning.org/ WYSIWYG mobile development
- Adobe AIR http://www.adobe.com/products/air.html
- WINDEV : http://www.windev.com/windevmobile/android.html

Overview : Développement

Android SDK

ou

App developed web standards ?

Selling your app to millions of customers

- The Android Market is currently adding about 5,000 apps per month
- The intense pricing pressure causes many developers to start off at a low price

Games 30%
Books 18%
Entertainment 20%
Travel 15%
Education 10%
Other 7%

LEARNING ANDROID FRAMEWORK

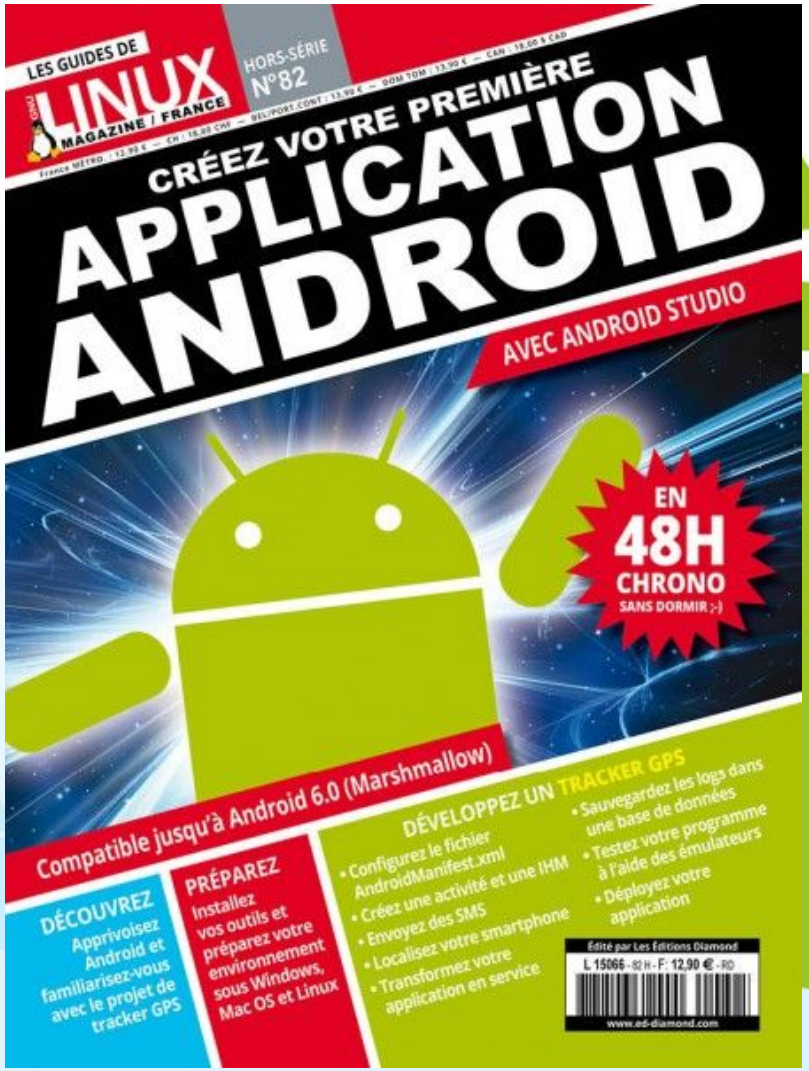
- SDK : Apprentissage linéaire
  - Pré-requis : Connaissance du langage objet, Java std, Linux Std
  - Connaissance de l'API Android
- NDK : Apprentissage plus « long »
  - JNI + C/C++
  - Maintenance plus difficile
- ADK : Pré-requis C + JDK, connaissance de ANDROID
- GDK : Pré-requis SDK
- Modification du noyau complexe

Formation de base 3/4 jours

links

http://developer.android.com/
http://developer.android.com/guide/index.html
http://developer.android.com/guide/webapps/index.html
http://source.android.com/
http://code.google.com/
http://developer.android.com/tools/samples/index.html
== > **http://developer.android.com/training**

links



https://github.com/frameproject/gpstracker