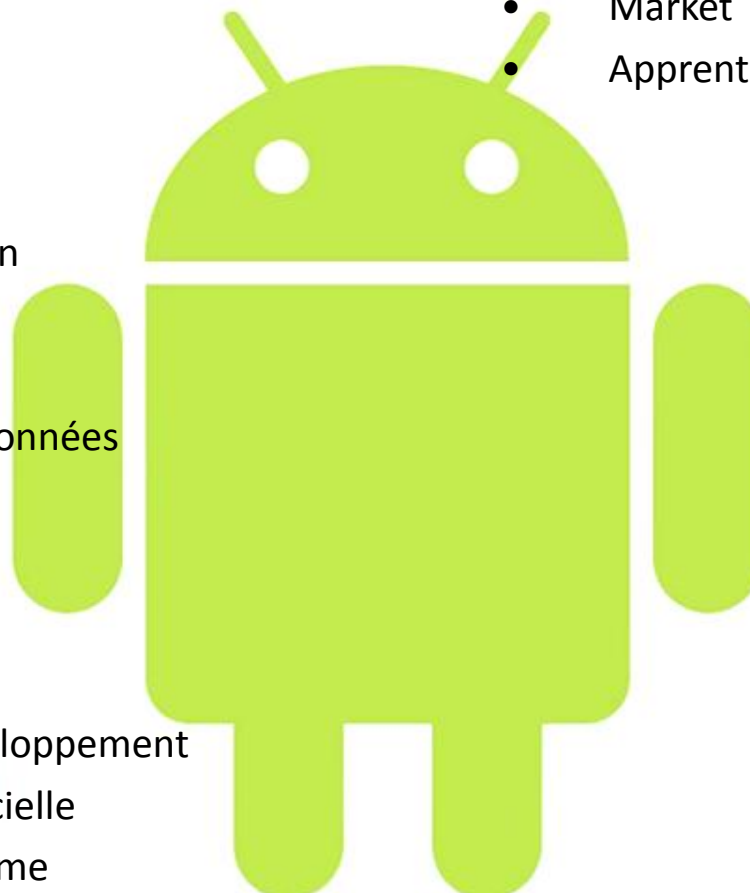


Android



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

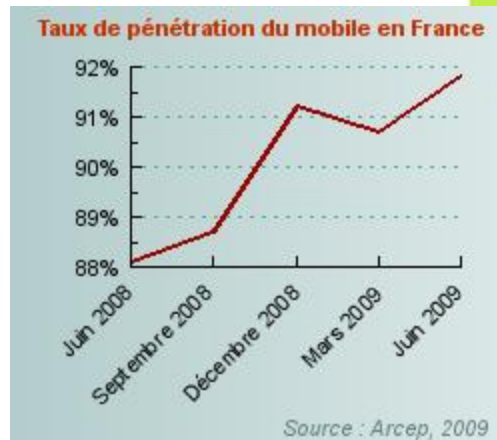
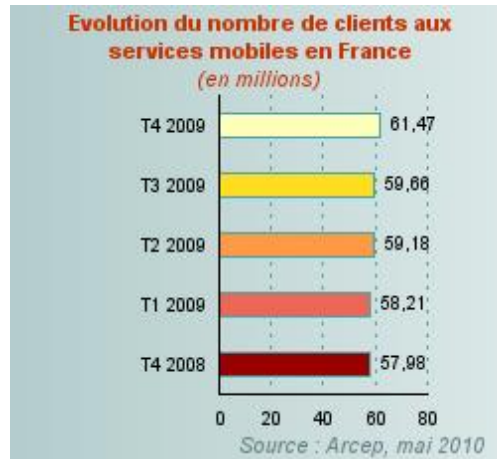
- L'économie de la téléphonie mobile
- Le projet Android
- Overview
 - Framework
 - IHM
 - Geolocalisation
 - Réseau
 - Téléphonie
 - Stockage de données
 - Bluetooth
 - Capteurs
 - Camera
 - NFC
 - Outils de développement
 - Structure logicielle
 - Cross plateforme
- Overview (suite)
 - Market
 - Apprentissage



L'économie de la téléphonie mobile



61,5 millions d'abonnements mobile en France au dernier trimestre 2009



- En France : Les revenus des services mobiles atteignent 4,9 milliards d'euros au dernier trimestre 2009 (nombre de cartes SIM en service)

- Parmi ces cartes SIM, 2,1 millions ne permettent que l'accès à Internet et 1,5 million sont utilisées pour des objets communicants

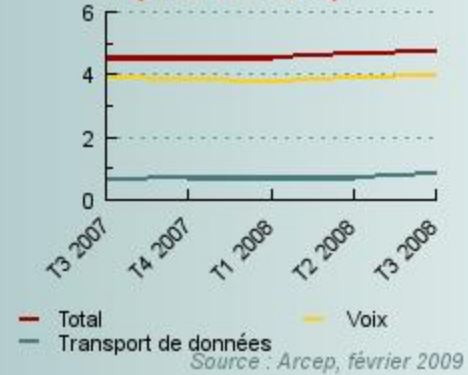
- **Le taux de pénétration du mobile proche de 92 % en France**
Il y avait fin juin 2009, 56,6 millions de clients mobiles en France pour 62,1 millions d'habitants, selon les données de l'Arcep. Soit un taux de pénétration de 91,8 %.

L'économie de la téléphonie mobile



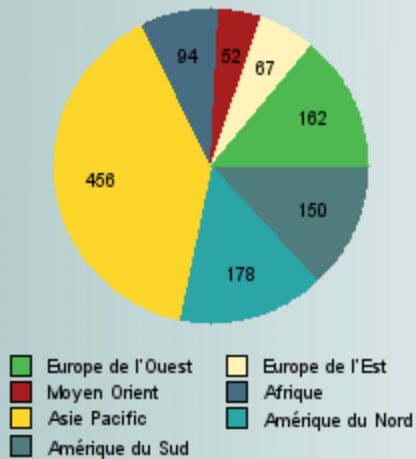
Les chiffres de la téléphonie

Revenus des services mobiles en France (milliards d'euros)



- En France : **4,7 milliards d'euros** - des revenus des services mobile sur le marché de détail au troisième trimestre 2008

Nombre de mobiles vendus dans le monde en 2009 (en millions d'unités)

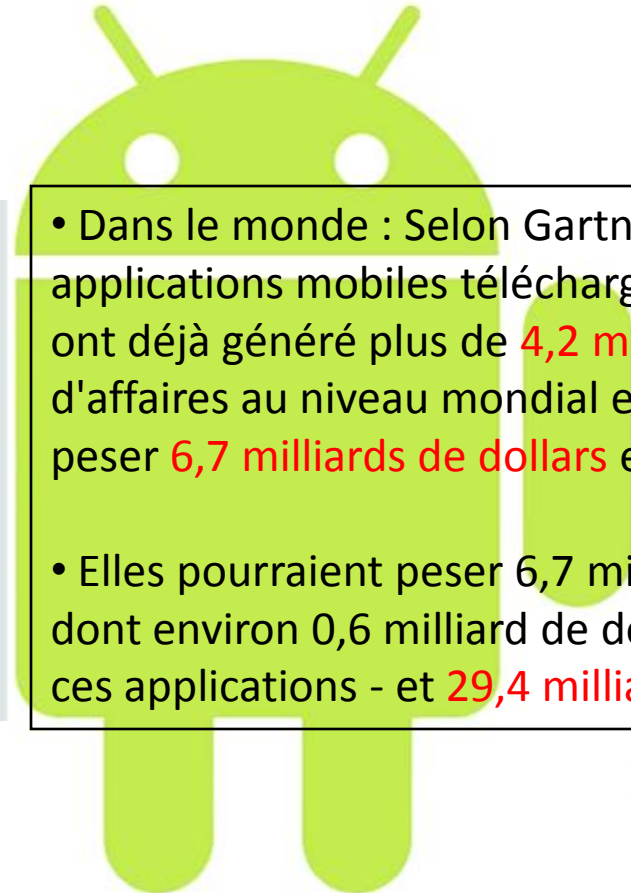


Source : GfK, février 2010

- Dans le monde : **1,16 milliard** de téléphones mobiles ont été vendus dans le monde en 2009

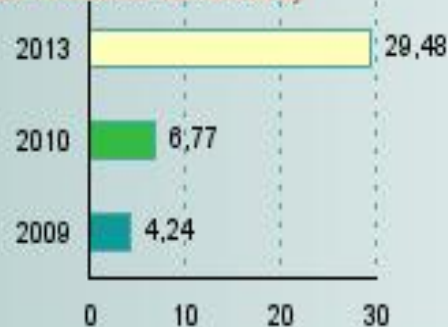


6,7 milliards de revenus pour les applications mobiles en 2010



Prévisions de chiffre d'affaires mondial pour les applications mobiles

(en milliards de dollars)



Source : Gartner, décembre 2010

- Dans le monde : Selon Gartner, seulement 18 % des applications mobiles téléchargées sont payantes. Elles ont déjà généré plus de **4,2 milliards de dollars** de chiffre d'affaires au niveau mondial en 2009. Elles pourraient peser **6,7 milliards de dollars** en 2010
- Elles pourraient peser 6,7 milliards de dollars en 2010 - dont environ 0,6 milliard de dollars via la publicité sur ces applications - et **29,4 milliards en 2013**.

L'économie de la téléphonie mobile



Prévision Gartner



OS	2009	2010	2011	2014
Symbian (Nokia)	80,876.3	107,662.4	141,278.6	
Market Share (%)	46.9	40.1	34.2	30.2
Android (Google)	6,798.4	47,462.1	91,937.7	259,306.4
Market Share (%)	3.9	17.7	22.2	29.6
Research In Motion (BlackBerry)	34,346.8	46,922.9	62,198.2	102,579.5
Market Share (%)	19.9	17.5	15.0	11.7
iOS (Apple)	24,889.8	41,461.8	70,740.0	130,393.0
Market Share (%)	14.4	15.4	17.1	14.9
Windows Phone	15,031.1	12,686.5	21,308.8	34,490.2
Market Share (%)	8.7	4.7	5.2	3.9
Other Operating Systems	10,431.9	12,588.1	26,017.3	84,452.9
Market Share (%)	6.1	4.7	6.3	9.6
Total Market	172,374.3	268,783.7	413,480.5	875,573.8



Quelques dates

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



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 ont un noyau open source (licence GPL), mais conserve une couche (cadre d'application, framework multimédia, applications propriétaires)



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



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), et
- Les sociétés de commercialisation (Aplix, Noser, TAT, Wind River).



Clés du succès

- ANDROID est gratuit : Apache V2, <http://www.apache.org/licenses/LICENSE-2.0.txt>
- . L'utilisation de la licence Apache est essentiel, 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 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 succès

- 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)
- Convergence des applications grand public
- Google Market



Clés du succès

- Le succès provient de la portabilité du code : votre application est toujours la même indépendamment du matériel
- Dalvik VM a été optimisé pour les systèmes à faible empreinte mémoire
- Dalvik n'exécute pas directement le Byte code (dex)
- Juste In Time exécution : hybrid entre le binaire et l'interprété

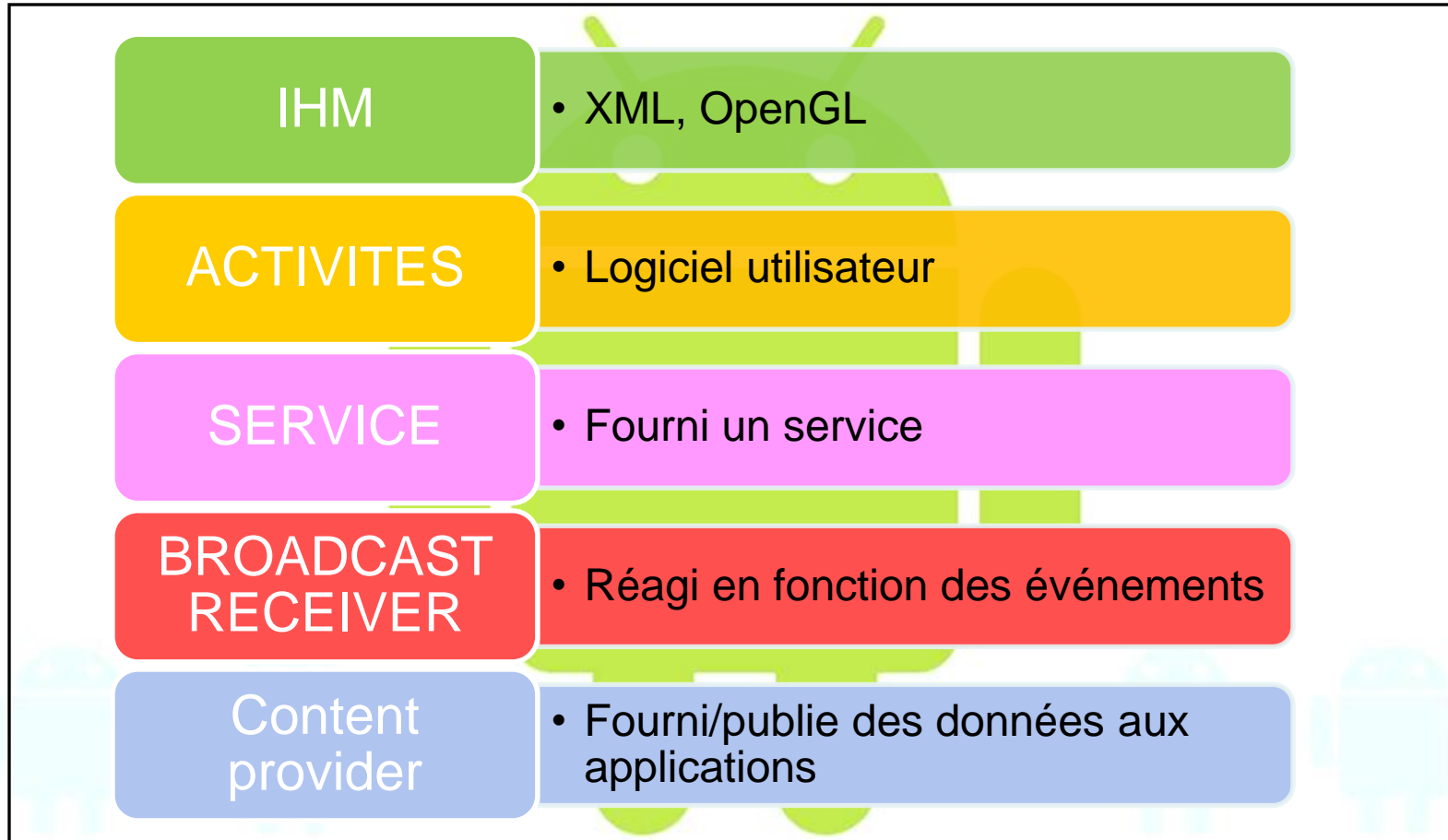


La VM ANDROID s'appel DALVIK après qu'un ingénieur de Google y soit parti en vacances

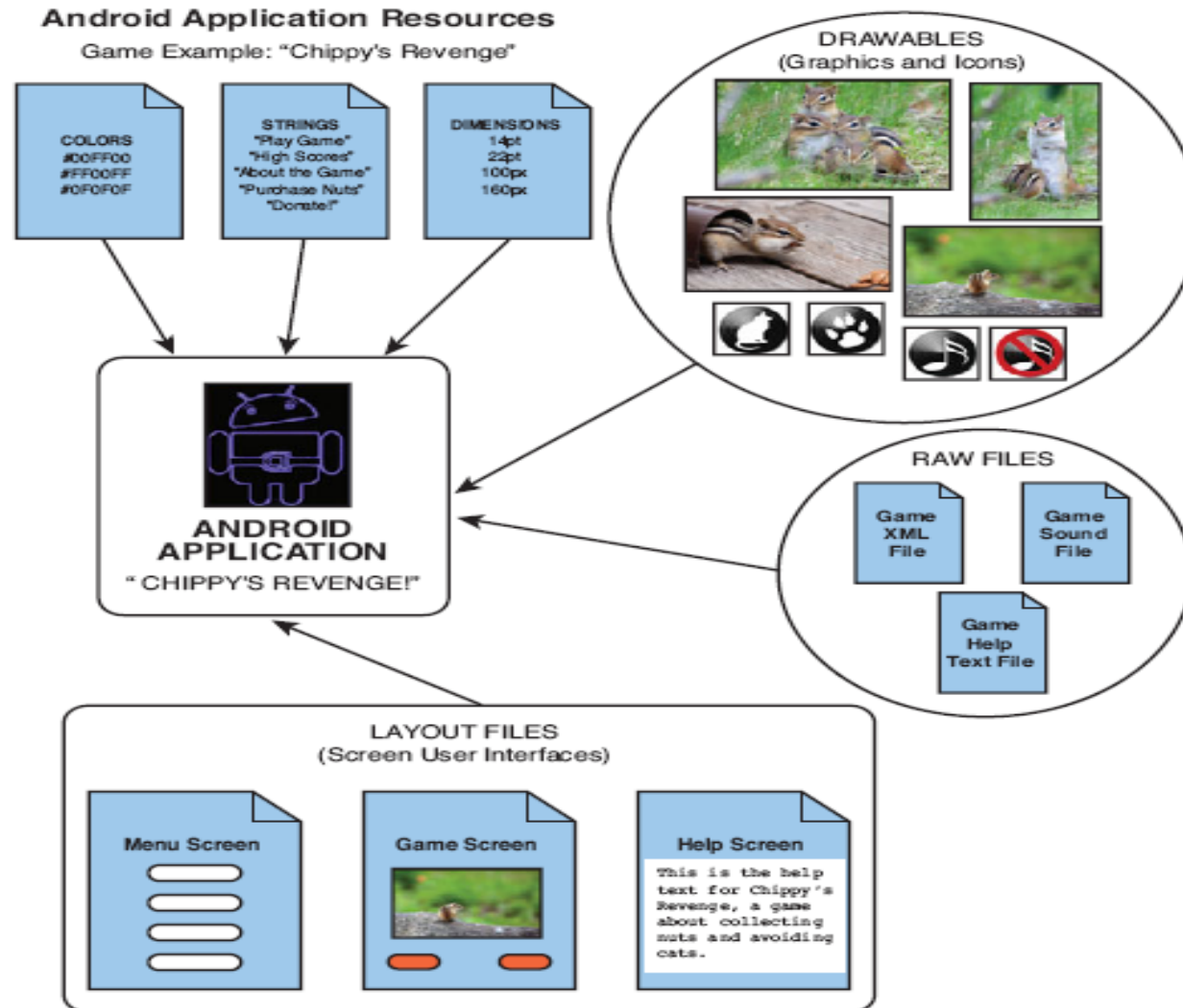
Overview : Framework



FRAMEWORK ANDROID : composants logiciels



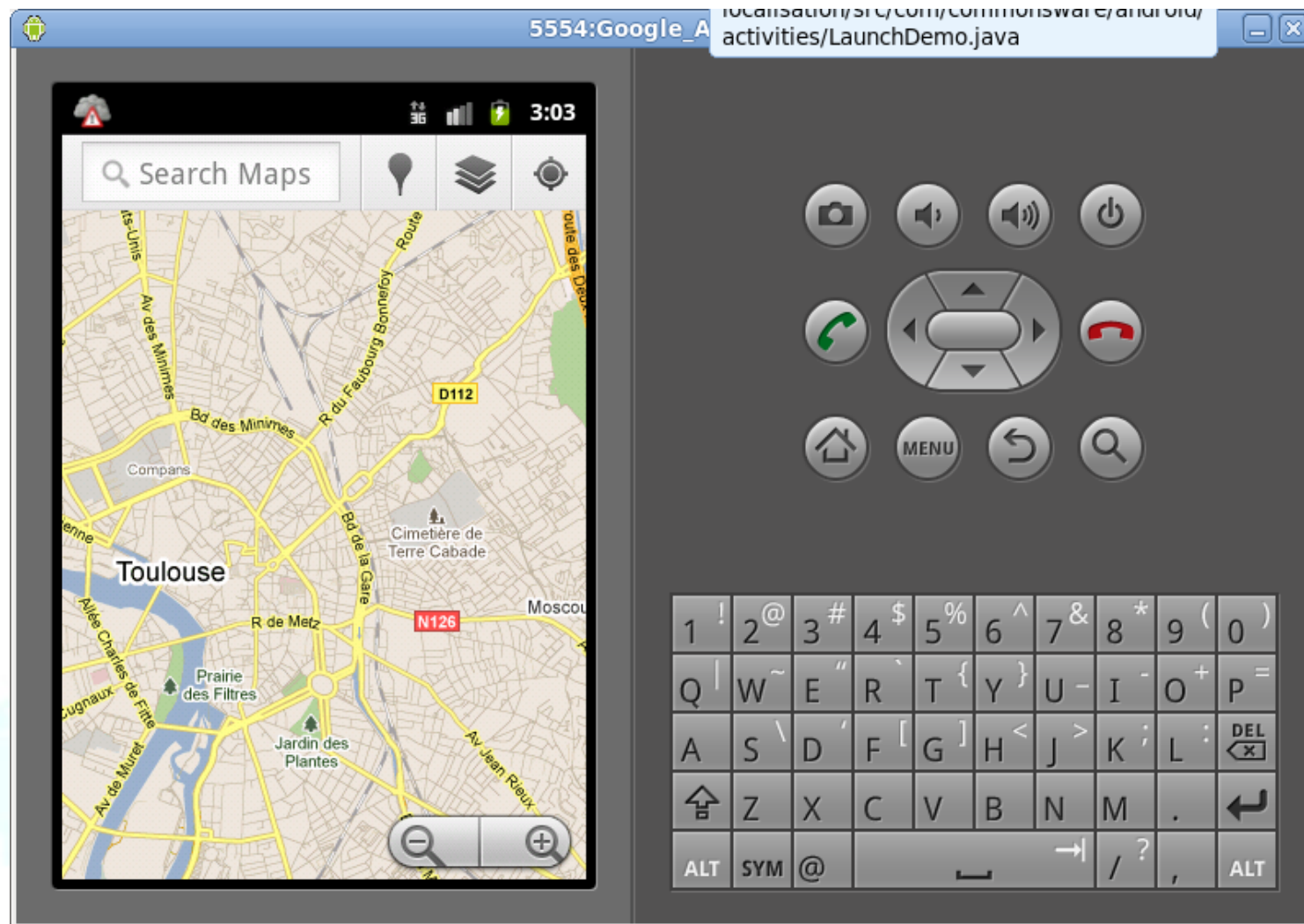
Overview: IHM Principes de base



Overview : Géolocalisation



Overview : Géolocalisation



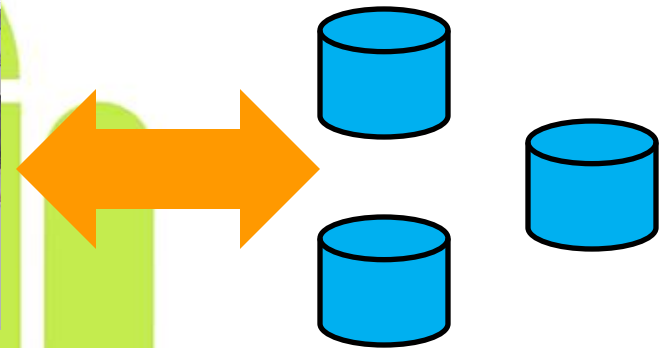
Overview : Réseau



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

In this recipe, the HTTP GET is used to retrieve XML or JSON data (see <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).

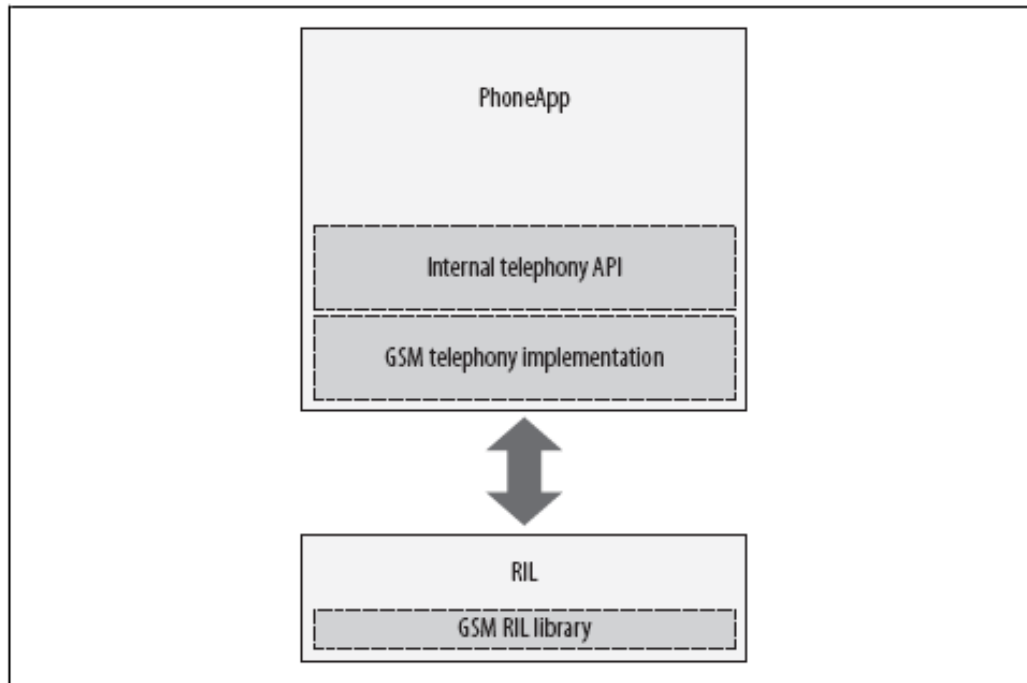


To search any topic, the topic just needs to be appended to the query. For example, to search information on the National Basketball Association (NBA), the following query returns JSON data:

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



Android Telephony Internals



Layers of telephony packages

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.



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>



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/>



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.



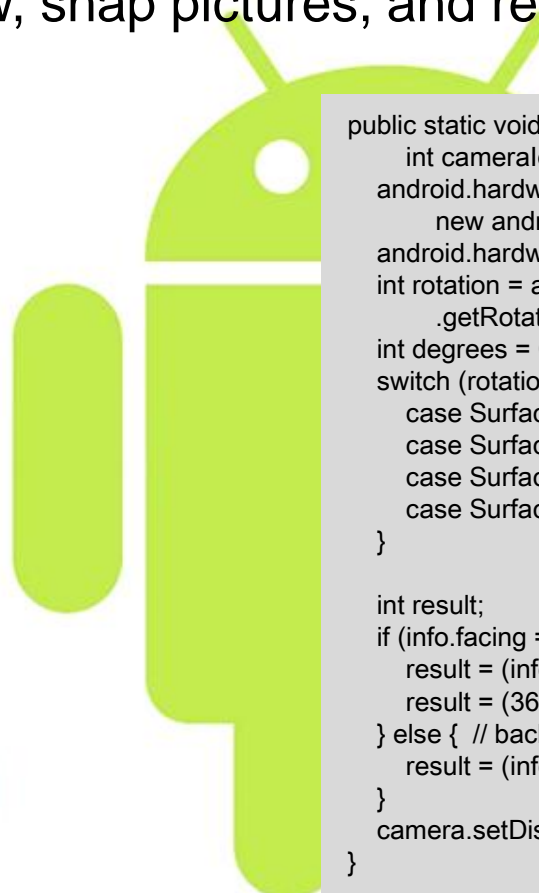
Flash memory is a non-volatile computer storage chip that can be electrically erased and reprogrammed.

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);
}
```



A [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)



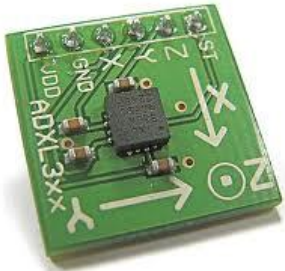
ANDROID

- 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



SensorManager lets you access the device's sensors



Accelerometer



Temperature

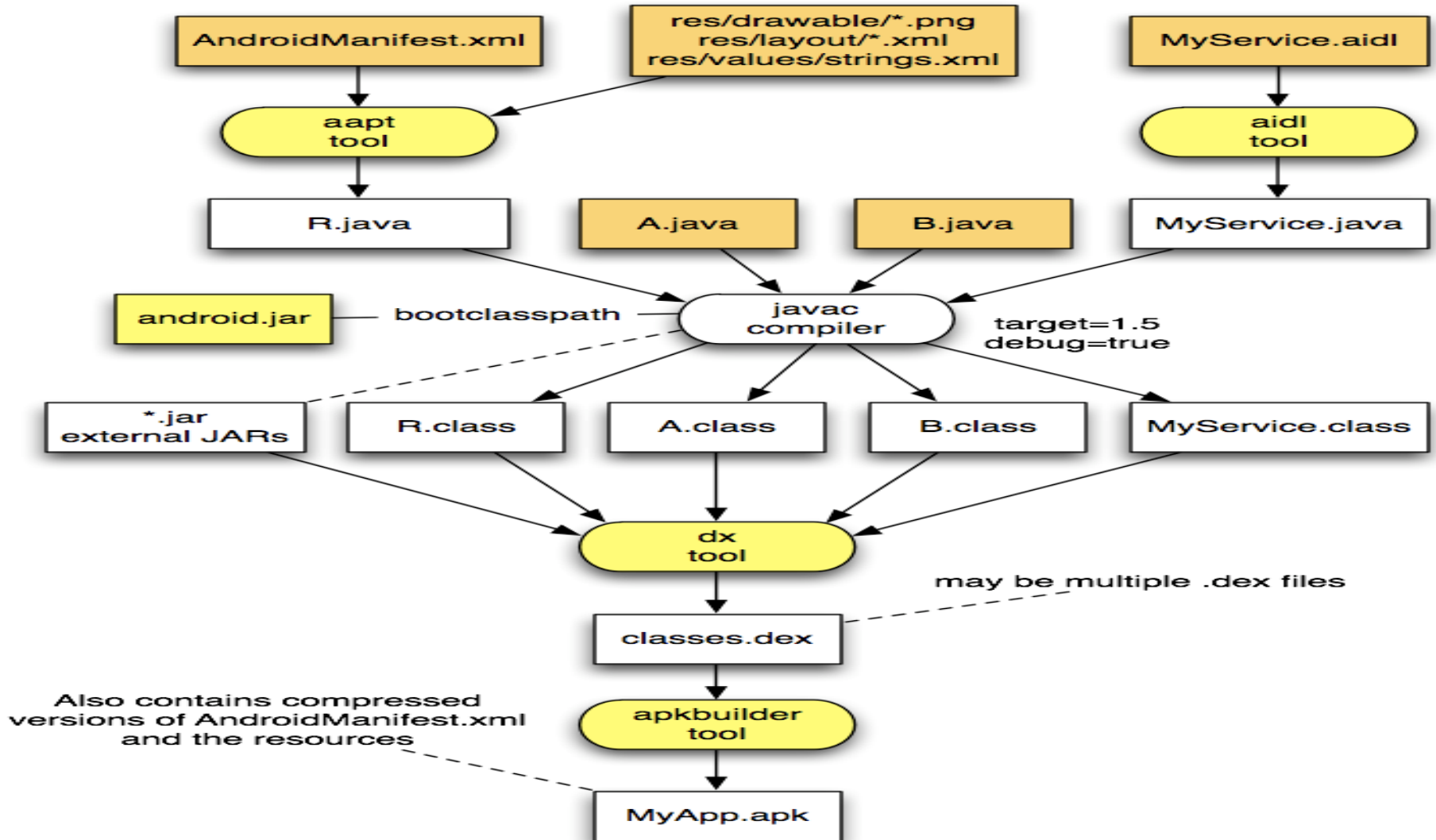
int	TYPE_ACCELEROMETER	A constant describing an accelerometer sensor type.
int	TYPE_ALL	A constant describing all sensor types.
int	TYPE_GRAVITY	A constant describing a gravity sensor type.
int	TYPE_GYROSCOPE	A constant describing a gyroscope sensor type
int	TYPE_LIGHT	A constant describing an light 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	<i>This constant is deprecated. use SensorManager.getOrientation() instead.</i>
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



[getQuaternionFromVector](#)(float[] Q, float[] rv)

Helper function to convert a rotation vector to a normalized quaternion.

Overview : outils de développement





SDK : Linux / Win32

- **Android Development Tools** : Création et débogage des applications
- **Android Emulator** : Emulateur de téléphone
- **Android Virtual Devices** : Instance d'un téléphone virtuelle
- **Dalvik 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 réécrire le byte code java en byte code android



En ligne de commande dans le SDK :

tools/ : ensemble de scripts et commande binaire pour développer sous ANDROID, les principaux :

android — Creates Android project files and to manage AVDs.

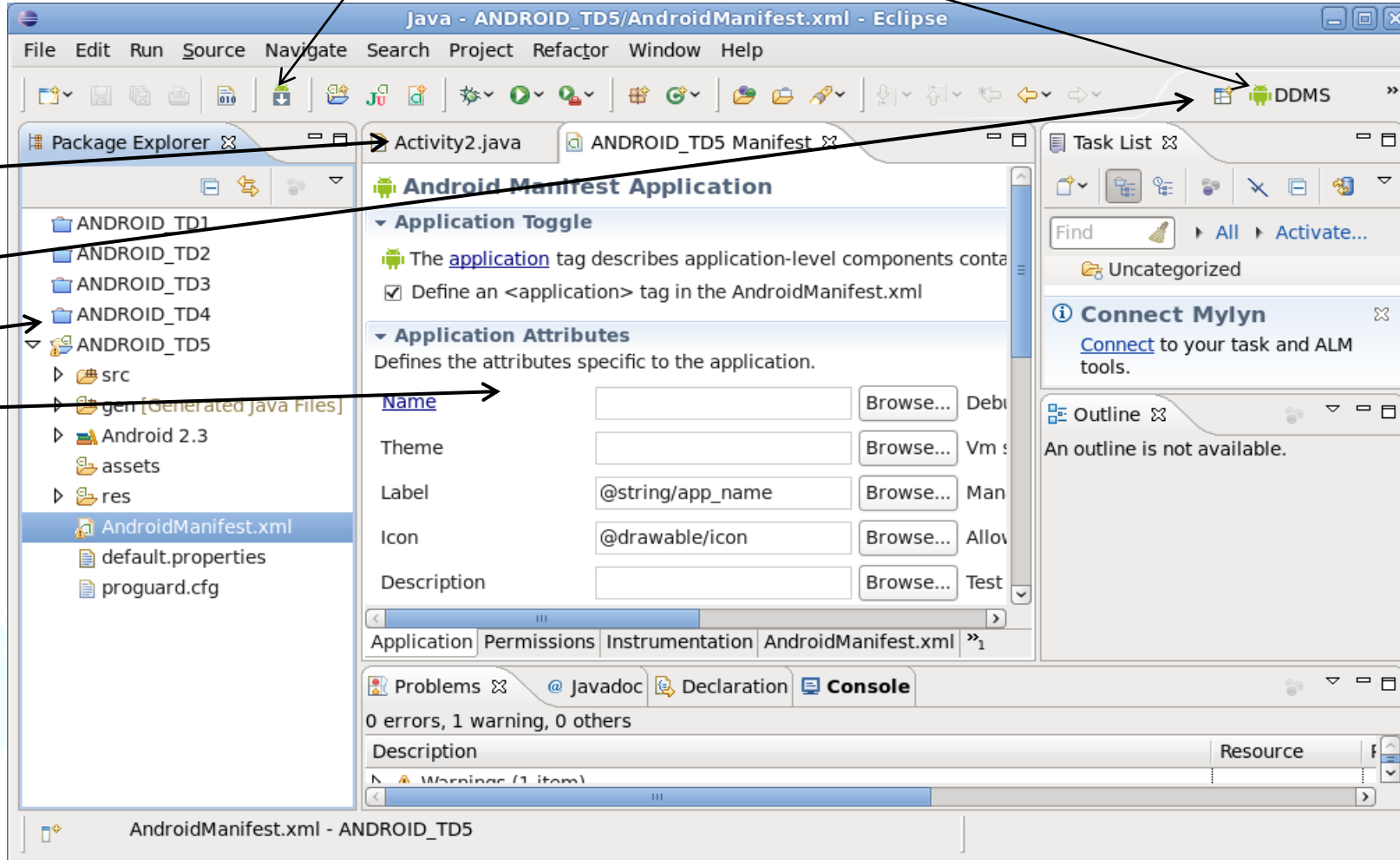
aapt (Android Asset Packaging Tool)—Packages Android project files into apk files for installation on the emulator and handset.

ddms (Dalvik Debug Monitor Service)—Has a user interface of its own, which resembles the Eclipse DDMS perspective.

adb (Android Debug Bridge)—Has a command-line interface for interacting with the emulator and the device.



IDE : Eclipse + Plugin Eclipse



view
perspective
project
code

Overview : outils de développement



DEBUG : Perspective DDMS (Dalvik Debug Monitor Service)

DDMS - ANDROID_TD5/src/example/android/Activity2.java - Eclipse

File Edit Run Source Refactor Navigate Search Project Window Help

Devices

Name	Process ID	Private Bytes
android.pro	312	8674
com.svox.p	324	8676
com.androi	333	8678 / 8700
example.a	353	8680

Emulator Control

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

LogCat

Filter:

Threads Heap Allocation Tracker File Explorer

Name	Size	Date	Time	Perm
data		2011-01-08	21:55	drwx
app		2011-01-10	20:46	drwx
example.android-2.apk	20264	2011-01-10	20:46	-rw-r
example.helloandroid-2	13261	2011-01-08	21:04	-rw-r
app-private		2011-01-08	21:01	drwx
backup		2011-01-10	20:45	drwx
dalvik-cache		2011-01-10	20:46	drwx
data		2011-01-08	21:47	drwx
dontpanic		2011-01-08	21:01	drwx
local		2011-01-08	21:01	drwx
lost+found		2011-01-08	21:01	drwx

The DDMS tool can be launched separately from Eclipse

Internal Thread

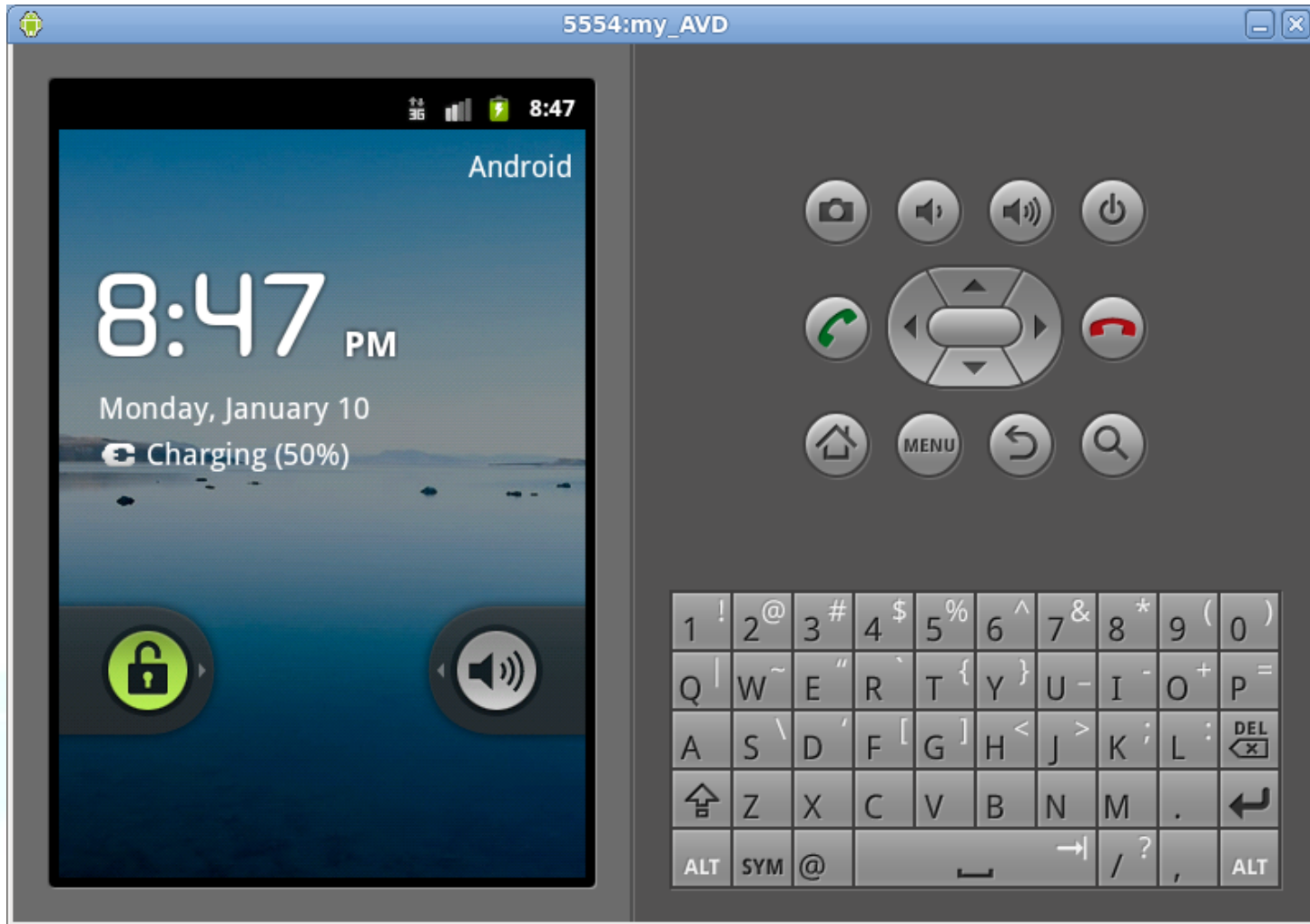
Simulating Incoming SMS Messages to the Emulator

Simulate an Incoming Call to an Emulator

Logging



Emulateur (téléphonie, sms, GPS)



Overview Structure logicielle



APPLICATIONS



APPLICATION FRAMEWORK



JNI

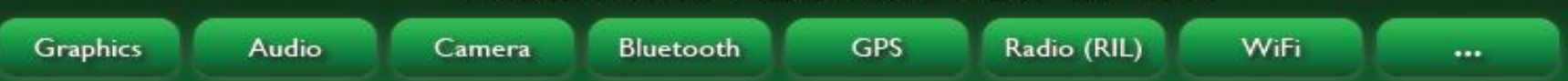
LIBRARIES



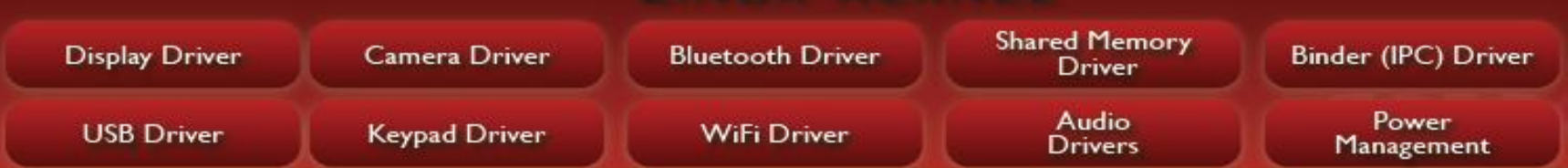
ANDROID RUNTIME



HARDWARE ABSTRACTION LAYER



LINUX KERNEL



Overview structure logicielle : Application

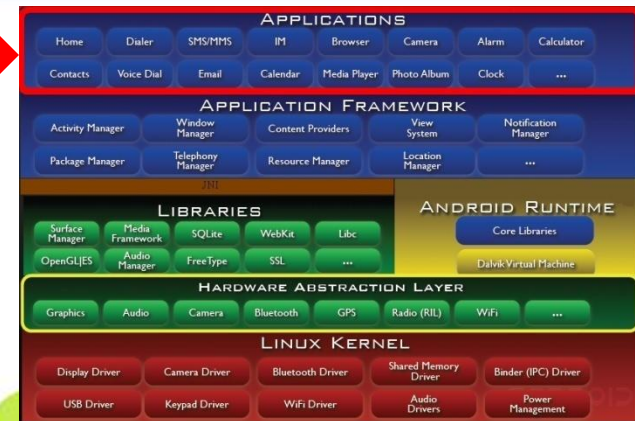


Composé d'un ensemble **d'application** de base :

- 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



Gere 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

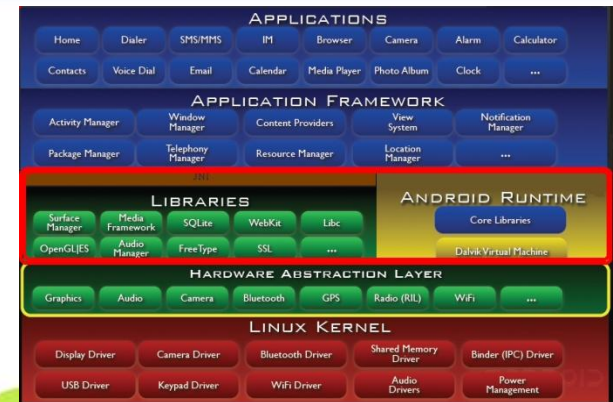
Ses capacités sont fournies aux Framework a travers la JNI

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 drivers kernel
- 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)**



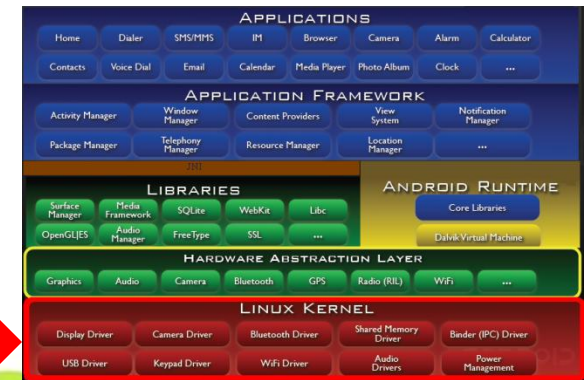
Overview structure logicielle : Linux Kernel



- 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
```

← Héritage

```
{
```

```
    /** Called when the activity is first created. */
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState)
```

← Surcharge

```
{
```

```
    super.onCreate(savedInstanceState);
```

← Référence à la classe « mère »

```
    setContentView(R.layout.main);
```

```
}
```

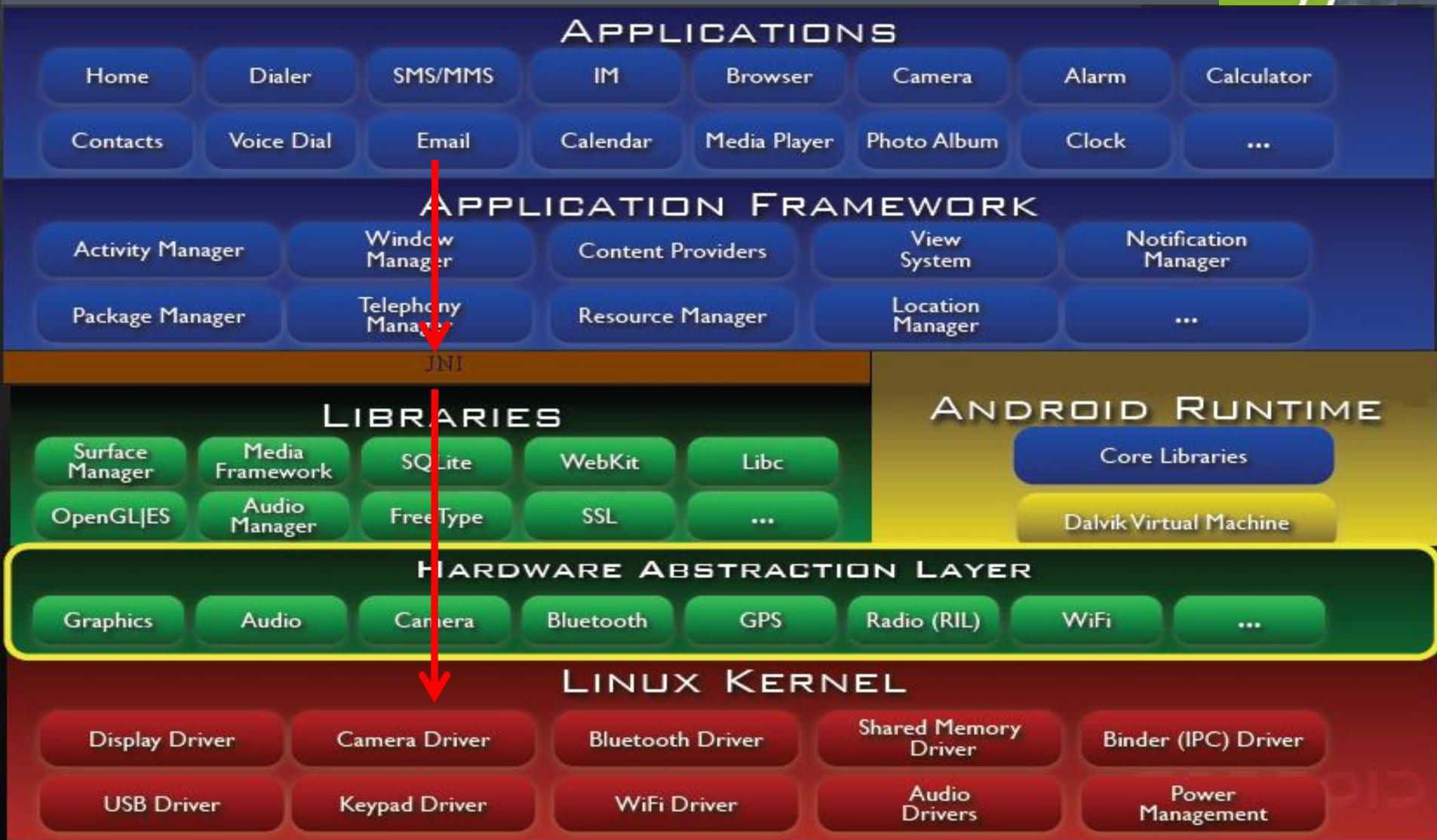
```
}
```



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

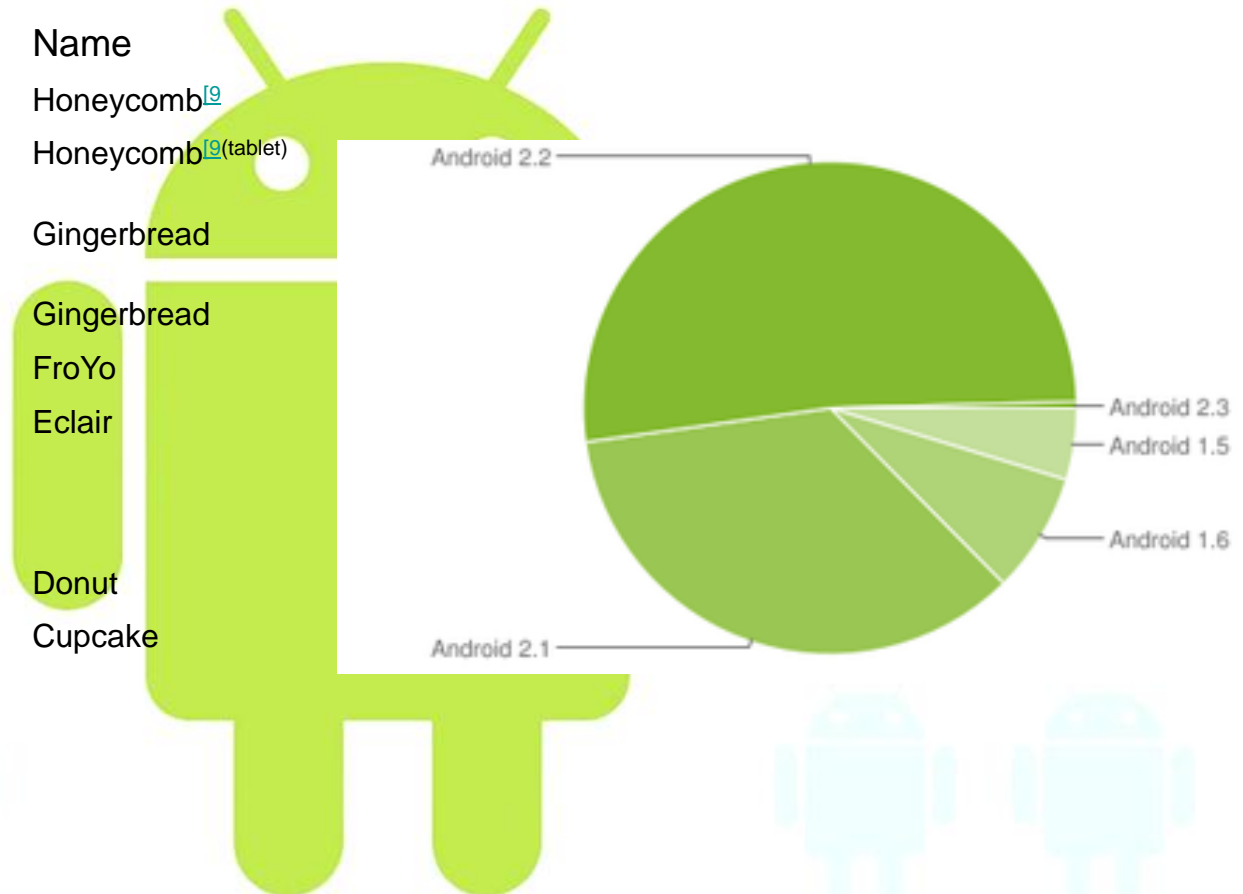
Overview structure logicielle : NDK



Overview structure logicielle : version



Platform Version	API Level	Name
Android 3.1 -	12	Honeycomb ¹⁹
Android 3.0	11	Honeycomb ¹⁹ (tablet)
Android 2.3.4	10	Gingerbread
Android 2.3.3		
Android 2.3	9	Gingerbread
Android 2.2	8	FroYo
Android 2.1	7	Eclair
Android 2.0.1	6	
Android 2.0	5	
Android 1.6	4	Donut
Android 1.5	3	Cupcake
Android 1.1	2	
Android 1.0	1	





Cross-Platform Native Frameworks

- **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://appinventor.googlelabs.com/about/> WYSIWYG mobile development



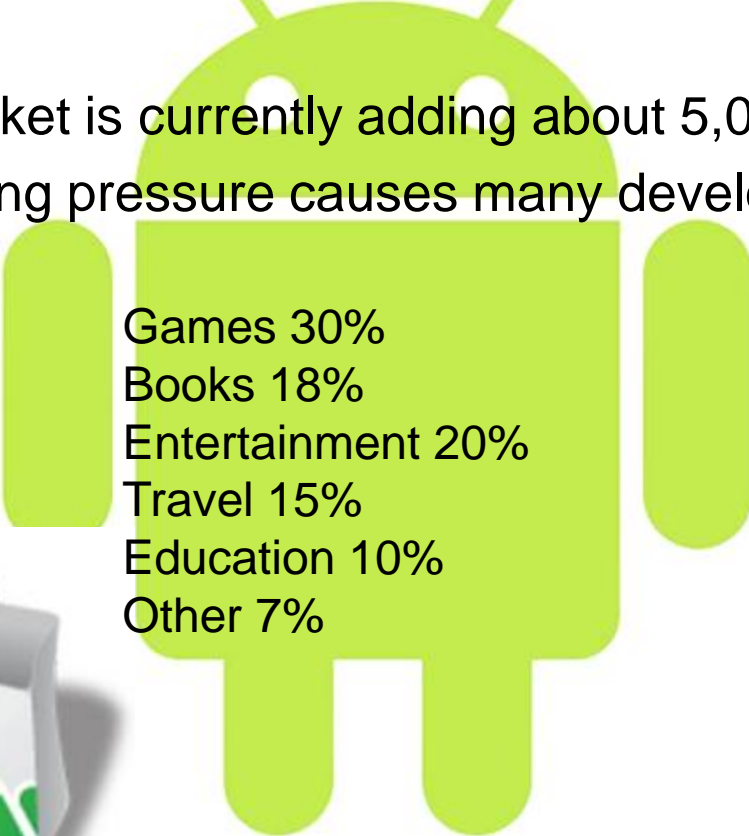
Android SDK Or App developed using web standards ?





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

- JDK : 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
- Modification du noyau complexe

Formation de base 5 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/>

