

# Android

Présentation d'Android

ubuntu<sup>o</sup>party

20 novembre 2010

Florent Garin

<http://www.docdoku.com>

<http://twitter.com/docdoku>



# Florent Garin

- Co-fondateur de DocDoku  
Société innovante IT  
<http://www.docdoku.com>
- Co-fondateur du JUG de Toulouse  
Ouvert à tous, les technologies Java  
<http://www.jugtoulouse.org>
- Open Source  
<http://code.google.com/p/docdoku>

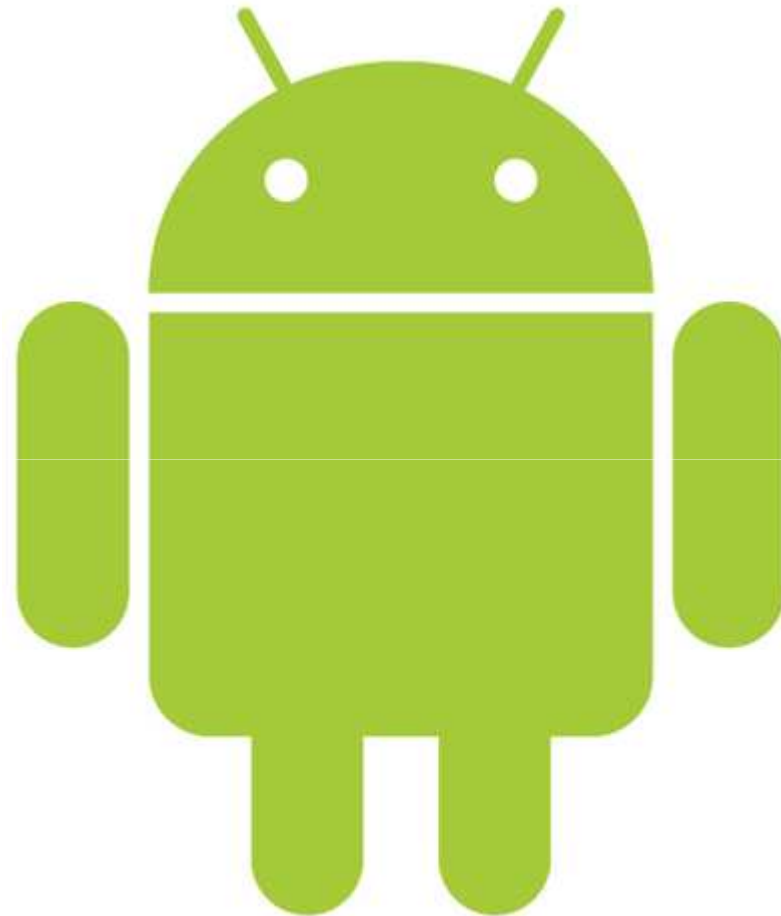


# Agenda

---

- **Qu'est-ce qu'Android ?**
- **Développer, Compiler, Débugger**
- **Le framework et l'API**
- **L'écosystème**
- **Conclusion**

# Qu'est-ce qu'Android ?



# Historique

- *Un SmartPhone est la fusion d'un téléphone portable et d'un PDA*
- *Quelques dates :*
  - *1997, le WAP*
  - *2002, le BlackBerry et son fameux push mail*
  - *2007, l'iPhone pose les bases :*
    - *Un vrai navigateur*
    - *Une ergonomie tactile et repensée pour le mobile*
    - *Un App Store*
  - *Fin 2009, sortie d'Android 2.0*
  - *Début 2010 plus d'une dizaine de téléphones*

# Qu'est-ce qu'Android ? : Une approche ouverte

- *Open Source (licence Apache)*
- *L'Open Handset Alliance*
  - *HTC*
  - *Motorola*
  - *Samsung*
  - *NEC*
  - *...Mais aussi*
  - *DoCoMo*
  - *Google*
  - *eBay*
  - *nVidia*

# Qu'est-ce qu'Android ? : Une pile complète

- *Un OS basé sur un noyau linux 2.6*
  - *Gère le matériel (Wifi, Bluetooth, la mémoire, les processus...)*
- *Des bibliothèques C/C++ (moteur HTML/Javascript Webkit, SQLite...)*
- *Environnement d'exécution managé, VM Dalvik*
- *Un framework de développement Java*
  - *Optimisé pour Dalvik*
  - *Spécialisé mobile*
  - *Structurant*
  - *Propose un modèle de coopération inter-applications*
- *Un ensemble d'applications de base (gestion des contacts, lecteur multimédia, client de messagerie...)*



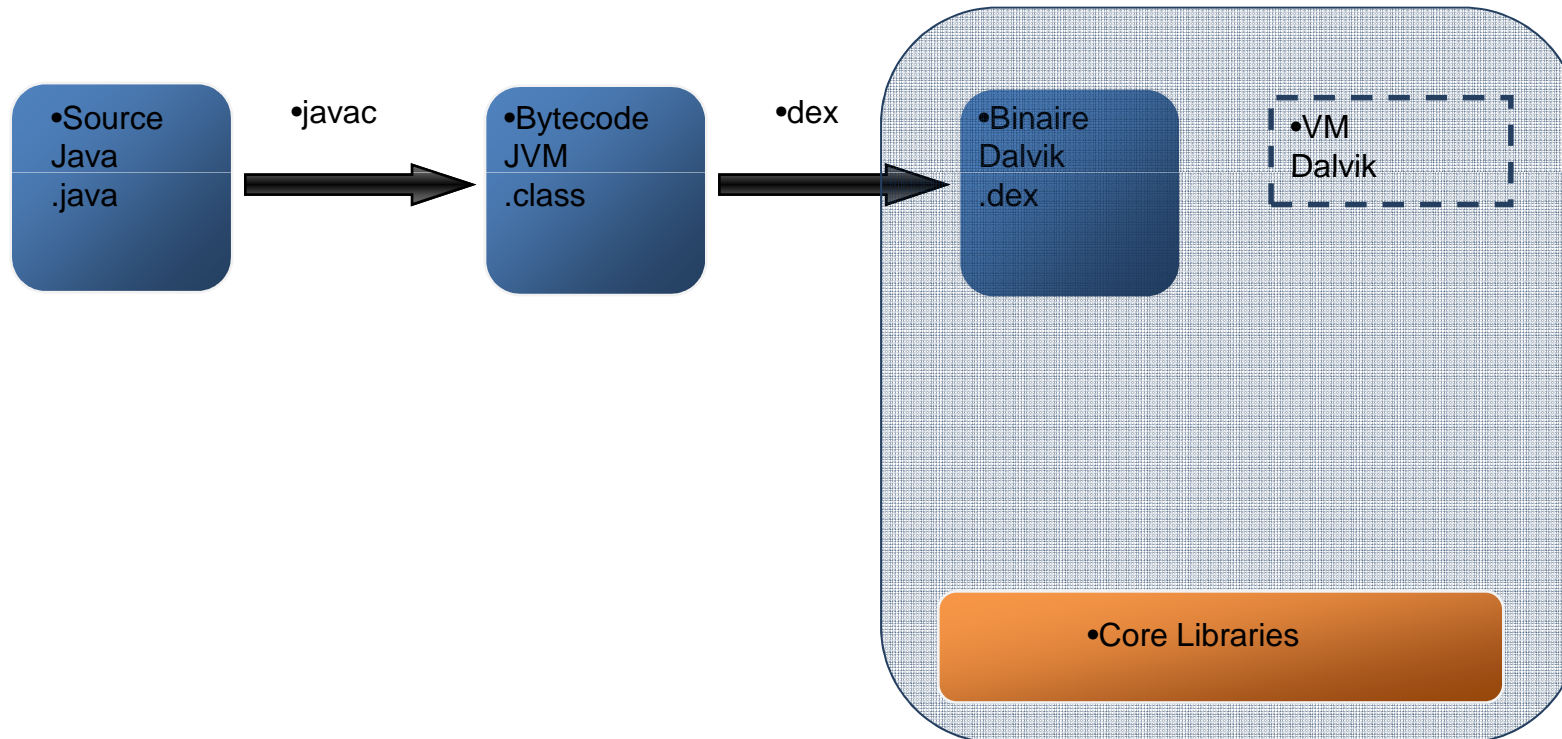
# Qu'est-ce qu'Android ? : Une pile complète





# Développer, Compiler, Débugger

- *SDK Android + JDK 1.5 ou 1.6*
- *javac -bootclasspath android.jar*



# La machine virtuelle Dalvik

- *Conçue pour fonctionner sur un mobile*
  - *CPU peu véloce*
  - *RAM réduite (64Mo)*
  - *Pas d'espace de Swap*
  - *Economise l'usage de la batterie*
  - *register-based et non stack-based*
  - *Plusieurs VM peuvent s'exécuter en parallèle*
- *Mais des limitations :*
  - *Pas de compilateur JIT*
  - *GC mark & sweep et non générationnel*

# Développer, Compiler, Débugger

- *Un plugin Eclipse*
- *Un émulateur*
- *Ant pour les autres IDE*
- *Debugger graphique*

The screenshot displays the Eclipse IDE interface for Android development. The top toolbar includes icons for Devices, Threads, Heap, and File Explorer. The main workspace is divided into several panels:

- Devices:** A list of virtual devices with columns for Name, PID, and other details.
- Emulator Control:** Controls for telephony status, including Voice, Data, Speed, and Latency settings.
- Heap Memory Monitor:** A window showing heap usage statistics. It includes a table with columns for ID, Heap Size, Allocated, Free, % Used, and # Objects. Below this is a detailed table of heap objects with columns for Type, Count, Total Size, Smallest, Largest, and Median. A 'Cause GC' button is also present.
- LogCat:** A window displaying system logs with columns for Time, pid, tag, and Message.

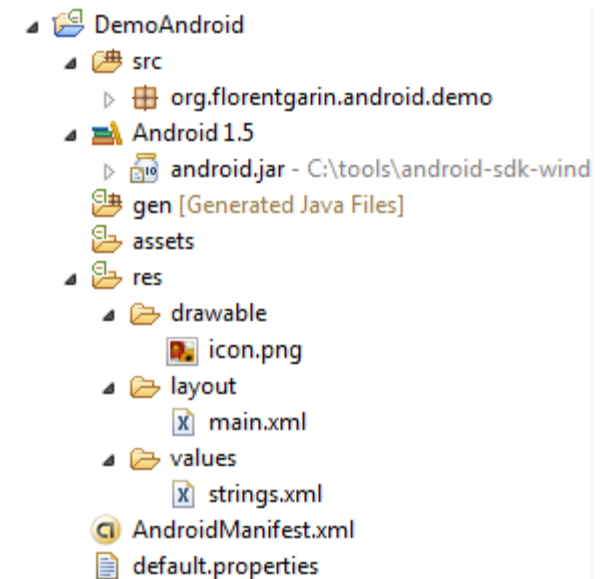
The LogCat window shows the following log entries:

Time	pid	tag	Message
07-08 17:06:19.338	W 750	KeyCharacterMap	Using default keymap: /system/usr/keychar...
07-08 17:06:44.791	I 750	dalvikvm	dvmDdmHandleHpsgChunk(when 1, what 0, heap...
07-08 17:06:59.992	I 579	dalvikvm	dvmDdmHandleHpsgChunk(when 1, what 0, heap...
07-08 17:07:03.479	I 624	dalvikvm	dvmDdmHandleHpsgChunk(when 1, what 0, heap...
07-08 17:07:05.060	I 627	dalvikvm	dvmDdmHandleHpsgChunk(when 1, what 0, heap...
07-08 17:07:15.658	I 579	ActivityManager	Starting activity: Intent { action=android...



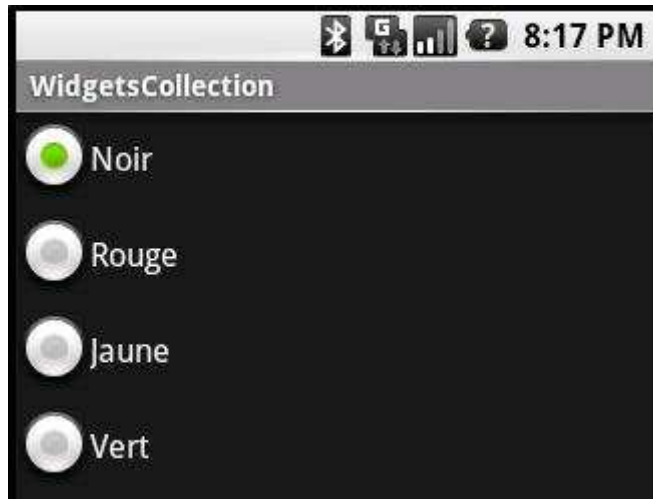
# Le framework et l'API

- *Android est un framework*
  - *Définition XML UI*
  - *Localisation/i18n*
  - *Fichier Manifest*
  - *Gestion des ressources*
  - *Persistance des données*
  - *Navigation entre écrans*
  - *Traitements longs asynchrones*
- *Environnement Managé*



# Le framework et l'API : IHM

- *De nombreux widgets (View)*
- *Et quelques layouts (ViewGroup)*



# Le framework et l'API : IHM

## ● Définition par du code

```
19     public void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         LinearLayout layout = new LinearLayout(this);
22         layout.setOrientation(LinearLayout.VERTICAL);
23         TextView firstNameLabel = new TextView(this);
24         firstNameLabel.setText("Prénom");
25         EditText firstNameText = new EditText(this);
26
27         TextView lastNameLabel = new TextView(this);
28         lastNameLabel.setText("Nom");
29         EditText lastNameText = new EditText(this);
30
31         layout.addView(firstNameLabel);
32         layout.addView(firstNameText);
33
34         layout.addView(lastNameLabel);
35         layout.addView(lastNameText);
36         setContentView(layout);
37     }
38 }
```

# Le framework et l'API : IHM

- *Définition déclarative*

```
1<?xml version="1.0" encoding="utf-8"?>
2<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3    android:orientation="vertical"
4    android:layout_width="fill_parent"
5    android:layout_height="fill_parent"
6    >
7<TextView
8    android:layout_width="fill_parent"
9    android:layout_height="wrap_content"
10   android:text="@string/hello"
11   />
12</LinearLayout>
13
```



## Le framework et l'API : Activity

- *Point d'entrée de l'application*
- *Représente un écran (ContentView)*
- *Navigation par Intent*
- *Cycle de vie géré par le système*



# Le framework et l'API : Activity

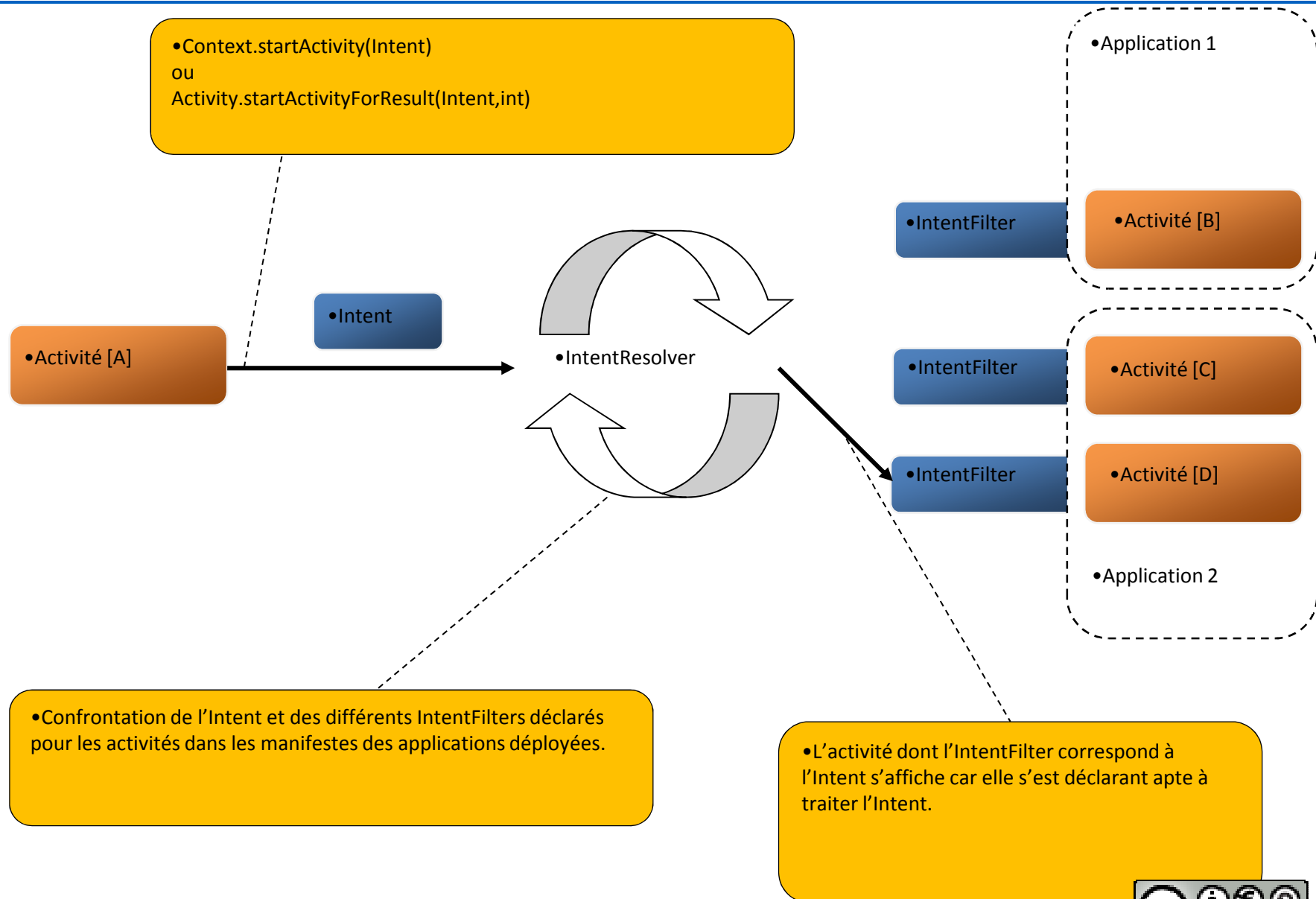
```
1 package org.florentgarin.android.demo;
2
3
4 import android.app.Activity;
5
6
7
8
9 public class DemoActivity extends Activity {
10
11     /*
12      * L'interface graphique est définie déclarativement
13      * par le fichier "res/layout/main.xml".
14      *
15      */
16     /** Called when the activity is first created. */
17     @Override
18     public void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.main);
21
22         final WebView web = (WebView) findViewById(R.id.web);
23         web.getSettings().setJavaScriptEnabled(true);
24         web.loadUrl("http://www.google.com/");
25     }
26 }
```



# Le framework et l'API : Intent

- Une « *intention* »
- Constituée de :
  - *Component (Résolution explicite)*
  - *Action, String (ex : Intent.ACTION\_WEB\_SEARCH)*
  - *Data/Type (URI, ex : « tel:0561234567 »)*
  - *Category (ex : Intent.CATEGORY\_LAUNCHER)*

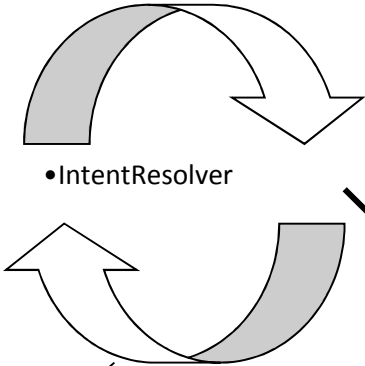
# Le framework et l'API : Intent



•Context.startActivity(Intent)  
ou  
Activity.startActivityForResult(Intent,int)

•Activité [A]

•Intent



•IntentFilter

•Application 1  
•Activité [B]

•IntentFilter

•Activité [C]

•IntentFilter

•Activité [D]

•Application 2

•Confrontation de l'Intent et des différents IntentFilters déclarés pour les activités dans les manifestes des applications déployées.

•L'activité dont l'IntentFilter correspond à l'Intent s'affiche car elle s'est déclarant apte à traiter l'Intent.



# Le framework et l'API : Intent

- *En cas d'indécision*



# Le framework et l'API : Service

- *Pour les tâches de fonds*
- *Indépendant des activités*
- *Définit en AIDL (Android Interface Definition Language)*
- *Similaire à Corba*
- *Parcelable = Serializable*

```
1 package org.florentgarin.android.service;
2
3 import org.florentgarin.android.service.Invoice;
4
5 interface ISampleService {
6
7     String echo(in String message);
8     int max(int a, int b);
9     void duplicate(int index, inout long[] data);
10    Invoice getInvoice(int number);
11 }
```

# Le framework et l'API : BroadcastReceiver

- *Réagit aux Intent*
- *Deux modes d'enregistrement :*
  - *Statiquement dans le manifest*
  - *Dynamiquement dans le code*
- *Deux modes d'émission :*
  - *Classique (sendBroadcast)*
  - *Ordonnée (sendOrderedBroadcast)*

# Le framework et l'API : BroadcastReceiver

## ● Méthode *onReceive* à implémenter

```
10 public class SMSBroadcastReceiver extends BroadcastReceiver {
11
12     @Override
13     public void onReceive(Context context, Intent intent) {
14         Bundle extras = intent.getExtras();
15         SmsMessage[] sms = null;
16         if (extras != null){
17             Object[] pdus = (Object[]) extras.get("pdus");
18             sms = new SmsMessage[pdus.length];
19             for (int i=0; i<pdus.length; i++){
20                 sms[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
21             }
22
23             String message = "SMS reçu de : ";
24             message += sms[0].getOriginatingAddress();
25             message += "\n";
26             message += sms[0].getMessageBody();
27
28             Toast.makeText(context, message, Toast.LENGTH_LONG).show();
29         }
30
31     }
32 }
```

## Le framework et l'API : ContentProvider

- *Expose les données de l'application*
- *Accédé indirectement au travers du ContentResolver*
- *Basé sur le paradigme REST*
- *Classe abstraite*
  - *query(Uri, String[], String, String[], String)*
  - *insert(Uri, ContentValues)*
  - *update(Uri, ContentValues, String, String[])*
  - *delete(Uri, String, String[])*
  - *getType(Uri)*
- *Se prête bien à un stockage physique SQLite*

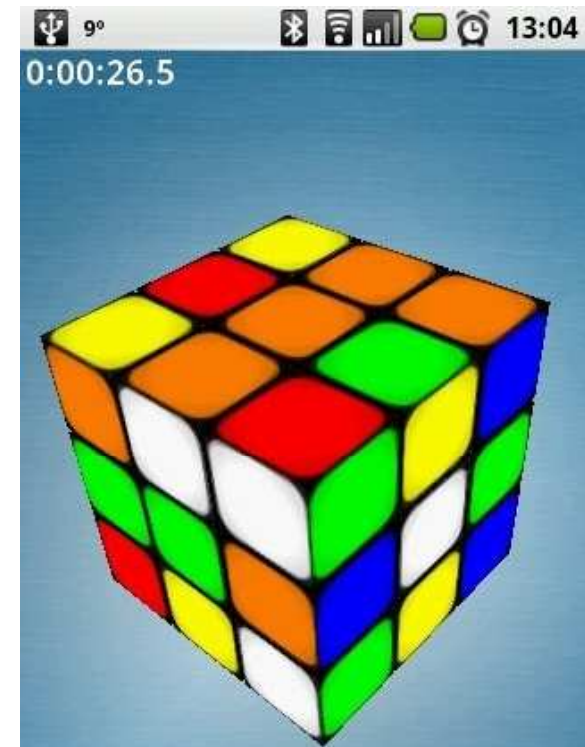


## Le framework et l'API : Le réseau

- *Apache HttpClient 4.0 au cœur*
- *JSON (JavaScript Object Notation)*
- *POX (Plain Old XML)*
- *SOAP mis de côté (KSOAP2)*
- *Pas d'offre pour le device to device communication, alternative Smack XMPP*

# Le framework et l'API : et aussi

- *3D avec OpenGL ES*
  - *Binding Java JSR 239*  
*Le même que JavaME*
- *2D langage XML pour les animations*
- *MediaPlayer*
- *MediaRecorder*
- *Le vibreur*
- *Géolocalisation (Google Map)*
- *AppWidgets*
- *Styles & Thèmes*
- *Mouse Gesture*
- *Synthèse et reconnaissance vocale*
- ...



# Développement natif

---

- *Le développement natif est possible*
- *NDK Android (Native Development Kit)*
  - *Evolution récente*
  - *Basé sur JNI (Java Native Interface)*
  - *Juste un pont avec les libs natives et la VM*
  - *Nécessite le SDK Android*

# Développement natif

- *Le NDK Android prend de plus en plus d'importance*
  - *Nouvelles commandes : ndk-build et ndk-gdb pour compiler et debugger*
  - *Lors du déploiement, Android s'assure qu'une version compilée de la lib native existe pour le CPU*
  - *Autorise la manipulation des objets Bitmap depuis le code natif (à partir d'Android 2.2) avec JNI*
  - *Aujourd'hui seuls les processeurs ARMv5TE et ARMv7-A sont supportés*
  - *x86 est prévu*

# Contenu du NDK

- *Les bibliothèques classiques suivantes sont disponibles*
  - *libc (C library)*
  - *libm (math library)*
  - *libz (Zlib compression)*
  - *OpenGL ES 1.1 and OpenGL ES 2.0*
- *Mais aussi quelques bibliothèques spécifiques à Android*
  - *liblog (Android logging)*
  - *libjnigraphics (Pixel buffer access)*
- *Et enfin pour interagir avec tout code Java*
  - *JNI interface headers*
- *Recourir à d'autres bibliothèques est périlleux !*



## Côté développeur

- *Un environnement de développement Android est nécessaire*
- *Un environnement GNU est indispensable*
  - *Natif sur Linux et Mac*
  - *Cygwin sur Windows*
- *Décompresser le fichier de distribution du NDK*
- *Un répertoire « jni » à la racine du projet accueillera les sources natives*

```
<manifest>
...
<uses-sdk android:minSdkVersion="3" />
<uses-feature android:glEsVersion="0x00020000" />
...
</manifest>
```

- *Déclarer les pré-requis exigés pour le téléphone*



# Les apports du NDK

- *Les objectifs du NDK Android*
  - *Favoriser les portages rapides*
  - *Il ne s'agit pas d'un SDK natif !*
  - *Les gains de performance ne sont pas systématiques*
  - *Pour une nouvelle application, il faut préférer un développement Java Android classique*

## Et l'écosystème ?





# L'écosystème

- *Android est Open Source mais est-il vraiment ouvert ?*
  - *Des voix s'élèvent pour protester contre la mainmise de Google sur Android*
  - *L'Open Handset Alliance ne serait qu'une façade*
- *La réalité est différente selon le point de vue des acteurs*
  - *Google*
  - *OEM*
  - *Les développeurs d'applications*
  - *Les utilisateurs*

# L'écosystème: Google

- *Google façonne la plateforme idéale à la génération de revenus tirés de leurs services*
  - *Google (AdWords, AdMob, AdSense)*
  - *Google Apps*
  - *Google Checkout*
  - *...*
- *Pour cela, il lui faut distribuer massivement son système*
  - *Non facturé aux fabricants*
  - *Licence ASF permissive*



# L'écosystème: Google

- *Google s'appuie sur plusieurs leviers pour asseoir son contrôle*
  - *Branches privées pour quelques partenaires*
  - *Autorité exclusive sur le codebase*
  - *La marque Android*
  - *Peu ou pas de visibilité sur la Roadmap*
  - *Certaines parties d'Android ne sont pas ouvertes*
  - *Agrément exigé pour les OEM pour l'accès à l'Android Market*
  - *Evolution rapide, encourageant les OEM à ne pas s'écarter de Google*

# L'écosystème: Les OEM

- *Les OEM*
  - *L'essentiel de leur énergie passe dans la customisation*
  - *Difficile pour certains de prendre part à la direction du projet*
- *Toutefois, les OEM peuvent se faire entendre*
  - *Android 3.0 supporterait mieux les tablettes*
- *Et les économies en R&D sont bien réelles*

# L'écosystème: Développeurs et utilisateurs

- *Peut être les grands gagnants*
  - *Un SDK complet et gratuit*
  - *Un framework modulaire et ouvert*
  - *Des smartphones à prix réduit*
  - *Une diffusion d'applications simple et peu censurée*

# Conclusion

- *Une copie parfaite ? Non !*
  - *Documentation parfois incomplète voir fausse*
  - *Evolutions de dernières minutes*
  - *Un contrepoids à Google serait souhaitable*
  - *Néanmoins un progrès énorme a été accompli*
- *Futur*
  - *W3C Geolocation API Specification*
  - *WebGL*
  - *HTML 5 Canvas*
  - *Le navigateur plateforme universelle ?*

## Conclusion : Pour aller plus loin ;=)

- *Android développer des applications mobiles pour les google phones*



- *RDV sur <http://www.docdoku.com/identite/contact/>*

