



# NormandyJUG

## Présentation d'Android



Avec l'aimable autorisation de l'auteur original de cette présentation.

**Florent Garin**

[florent.garin@jugtoulouse.org](mailto:florent.garin@jugtoulouse.org)

<http://www.docdoku.com>



# Présentation des intervenants

**Nicolas Giard**

<http://www.noocodecommit.com>

Ingénieur NTIC chez Team Partners

Co-fondateur du NormandyJUG

<http://www.normandyjug.org>

Spécialiste [Apache Wicket](#)





# Sommaire

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





# Sommaire

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





# La concurrence : iPhone

- *Celui derrière qui tout le monde court*
- *A fixé les standards*
  - *Tactile*
  - *AppStore*
- *Matériel et OS faits par Apple*
- *Objective-C*
- *Plateforme fermée*
- *Peut s'imposer dans les entreprises ?*





# La concurrence : Palm

- *Un précurseur*
- *Périclité depuis plusieurs années*
- *Renouveau avec le Pré*
- *Adieu PalmOS, bonjour WebOS*
- *Clavier*
- *Multi-touch*
- *Framework Mojo = HTML5+API JavaScript*





# La concurrence : Nokia

- *Solide réputation*
- *Leader avec 37% de PDM mais en diminution*
- *Gamme très (trop ?) large*
- *A loupé la vague tactile*
- *OS Symbian*
- *Symbian Foundation*





# La concurrence : BlackBerry

- *Connoté « Enterprise »*
- *Aussi gros que Nokia*
- *« Inventeur » du push mail*
- *Se met au multimédia*
- *RIM fait le hard et le soft*







# La concurrence : Windows Mobile

- *Windows CE date de plus de 10 ans*
- *Peu d'innovation*
- *Ne fait pas rêver*
- *Au départ simple déclinaison de Windows sur le mobile*
- *Intégration facile aux produits Microsoft*
- *Multi-constructeurs*





# La concurrence : JavaME

- *Massivement déployé*
- *GUI limité*
- *Différences entre les téléphones*
- *Dédié aujourd'hui aux appareils entrée de gamme*





# La concurrence : JavaFX

- *S'appuie sur JavaME*
- *Mais pourrait un jour lâcher cet héritage*
- *Peine à décoller*
- *Une promesse = développement unique et déploiements multiples : bureau, web, mobile*





# Sommaire

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





# Qu'est-ce qu'Android ?





# Une approche ouverte

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





# Une pile complète

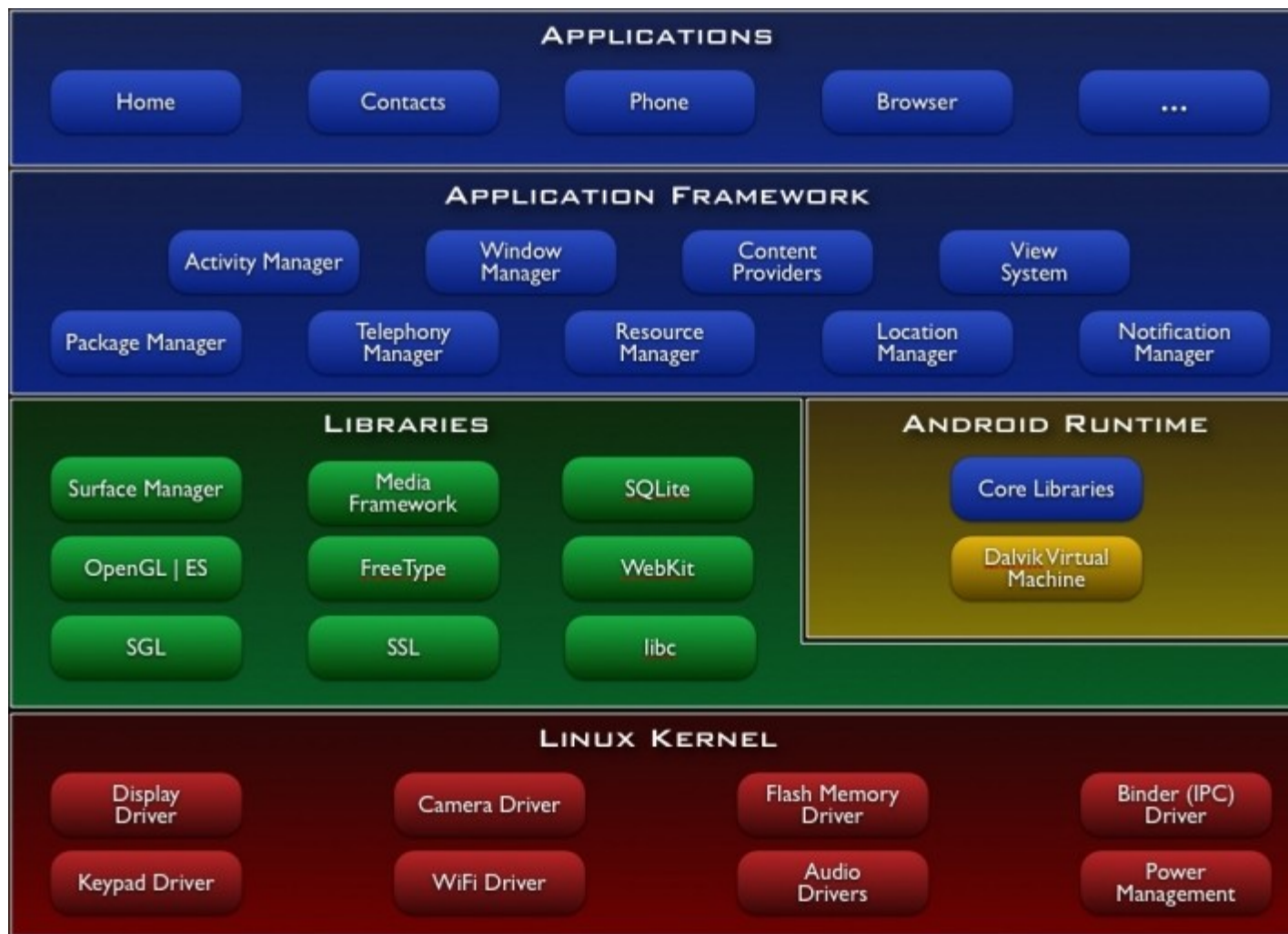
- *Un système d'exploitation (noyau Linux)*
- *Des librairies C/C++*
- *Une machine virtuelle Dalvik*
- *Un framework et une API Java*
- *Des applications par défaut*







# Une pile complète







# Sommaire

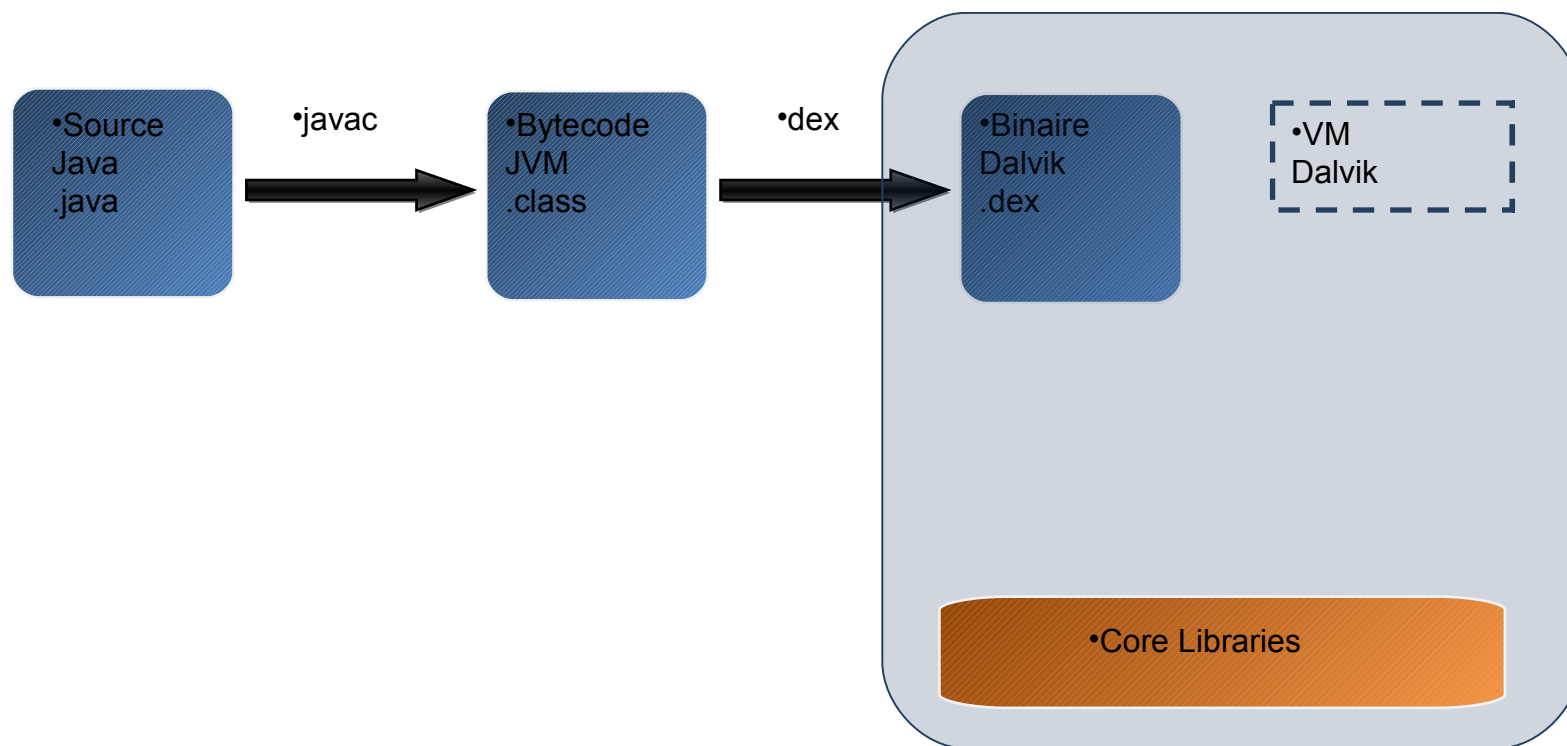
- **La concurrence**
- **Qu'est-ce qu'Android ?**
- **Développer, Compiler, Débugger**
- **Le framework et l'API**
- **Conclusion**





# Développer, Compiler, Débugger

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





# Développer, Compiler, Débugger

- *NDK Android (Native Development Kit)*
  - *Evolution récente*
  - *Basé sur JNI (Java Native Interface)*
  - *Juste un pont avec les libs natives et la VM*





# 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 shows icons for running, debugging, and other IDE functions. The main workspace is divided into several panels:

- Devices:** A list of system components and their IDs, including `com.android.phone` (624), `android.process.acore` (627), `com.android.alarmclock` (658), `android.process.media` (660), `com.android.mms` (695), `org.florentgarin.android.configuration` (750), and `com.android.inoutmethod.latin` (765).
- Emulator Control:** A panel for controlling the emulator, including telephony status (Voice: home, Speed: Full, Data: home, Latency: None) and telephony actions (Incoming number:).
- LogCat:** A panel showing system logs, including messages from `KeyCharacterMap`, `dalvikvm`, and `ActivityManager`.
- Heap:** A panel displaying memory usage statistics, including a table of heap objects and a bar chart showing memory usage over time.

ID	Heap Size	Allocated	Free	% Used	# Objects
1	2,754 MB	2,041 MB	729,500 KB	74,13%	44 291

Type	Count	Total Size	Smallest	Largest	Median
free	1 099	719,625 KB	16 B	238,742 KB	24 B
data object	32 082	939,633 KB	16 B	528 B	32 B
class object	2 089	343,430 KB	168 B	176 B	168 B
1-byte array (byte[], boolean[])	1 197	127,609 KB	24 B	8,023 KB	40 B
2-byte array (short[], char[])	7 042	504,336 KB	24 B	28,023 KB	40 B
4-byte array (object[], int[], float[])	1 871	174,000 KB	24 B	16,023 KB	40 B
8-byte array (long[], double[])	10	1,117 KB	40 B	248 B	96 B





# Sommaire

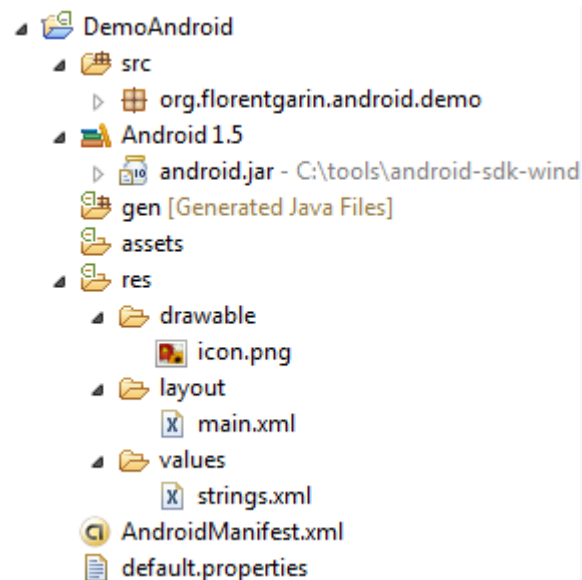
- **La concurrence**
- **Qu'est-ce qu'Android ?**
- **Développer, Compiler, Débugger**
- **Le framework et l'API**
- **Conclusion**





# Le framework et l'API

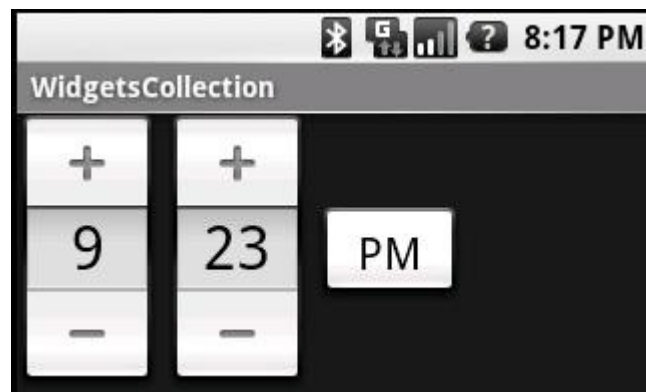
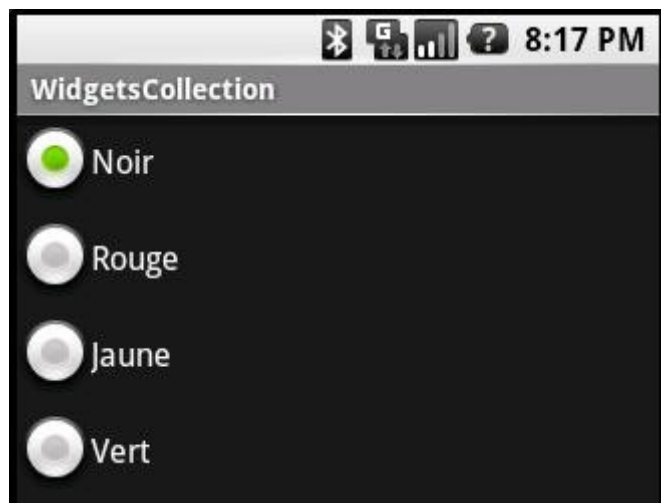
- *Android est un framework*
  - *Définition XML UI*
  - *Localisation/i18n*
  - *Fichier Manifest*
  - *Gestion des ressources*
  - *Persistance des données*
  - *Navigation entre écrans*
  - *Traitements longs asynchrones*
- *Environnement Managé*





# 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 }
```





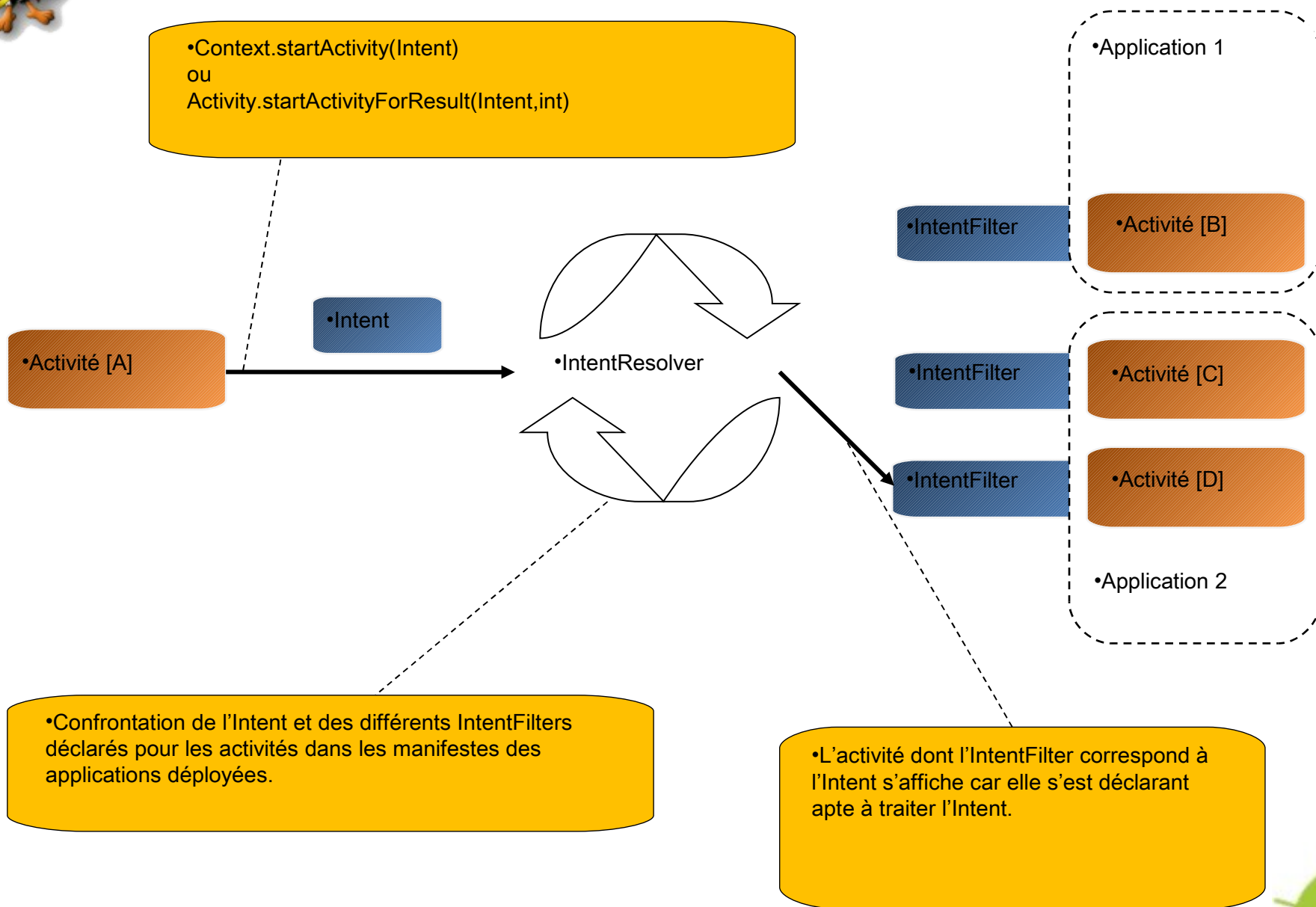
# Le framework et l'API : Intent

- 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





# 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*

```
1package org.florentgarin.android.service;
2
3import org.florentgarin.android.service.Invoice;
4
5interface 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

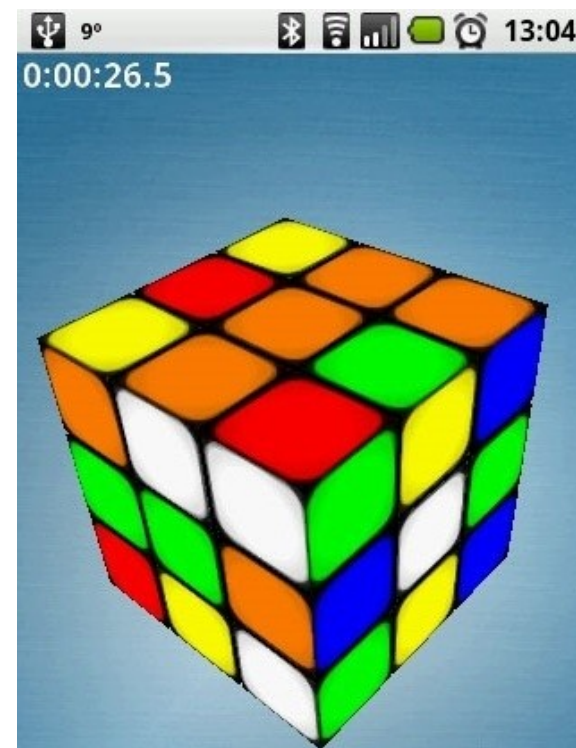
- *Apache HttpClient 4.0 au cœur*
- *JSON (JavaScript Object Notation)*
- *POX (Plain Old XML)*
- *SOAP mis de côté (KSOAP2)*
- *Pas d'offre pour le device to device communication, alternative Smack XMPP*





# 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*
- ...





# Sommaire

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





# Conclusion

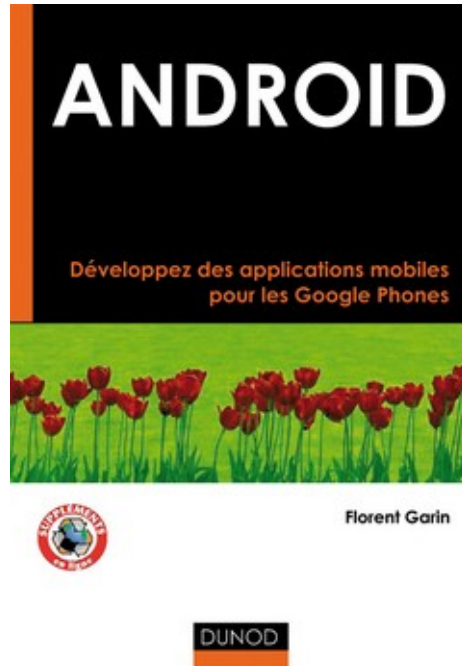
- *Une copie parfaite ? Non !*
  - *Documentation parfois incomplète voir fausse*
  - *Evolutions de dernières minutes*
  - *Manque la communication mobile à mobile*
  - *Néanmoins un progrès énorme a été fait*
- *Futur*
  - *W3C Geolocation API Specification*
  - *WebGL*
  - *HTML 5 Canvas*
  - *Le navigateur plateforme universelle ?*





# Conclusion : Pour aller plus loin ;=)

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







# Conclusion : et aussi...

- *Informations générales*
  - <http://android-france.fr>
  - <http://www.frandroid.com>
  - *Et de nombreux blogs et sites partout dans le monde*
- *Pour les développeurs*
  - <http://developer.android.com>
  - <http://android-developers.blogspot.com>
  - *Android France (catégorie Développement)*
  - *A vous de chercher / trouver*

