Android une Introduction

jean-michel Douin, douin au cnam point fr
version : 26 Septembre 2012

Notes de cours

Sommaire

- **Un peu d'historique**
- **Andoid OS comme *middleware***
  - **Applications et évènements gérés par le middleware**
  - **Une approche déclarative des IHM en XML**
  - **Une configuration en XML**
  - **Linux et Java**
    - **sans la JVM mais avec une DVM**
  - **…**
- **Principes de base**
- **Ce n'est qu'une introduction …**

Bibliographie utilisée

http://developer.android.com/resources/index.html

Le livre de Mark Murphy chez Pearson

Le livre écrit par Florent Garin

Android

Le cours de Victor Matos

http://grail.cba.csuohio.edu/~matos/notes/cis-493/Android-Syllabus.pdf

Plusieurs livres

Android A Programmers Guide - McGraw Hill

Professional Android Application Development - Wrox

Android : les objectifs

- **http://www.android.com**

- **http://www.OpenHandsetAlliance.com**

  "… **Open Handset Alliance**™, a group of 47 technology and mobile companies have come together to accelerate innovation in mobile and offer consumers a richer, less expensive, and better mobile experience.

  • Together we have developed Android™, the first complete, open, and free mobile platform.

  • We are committed to commercially deploy handsets and services using the Android Platform. "

Qu'est-ce que Android ?

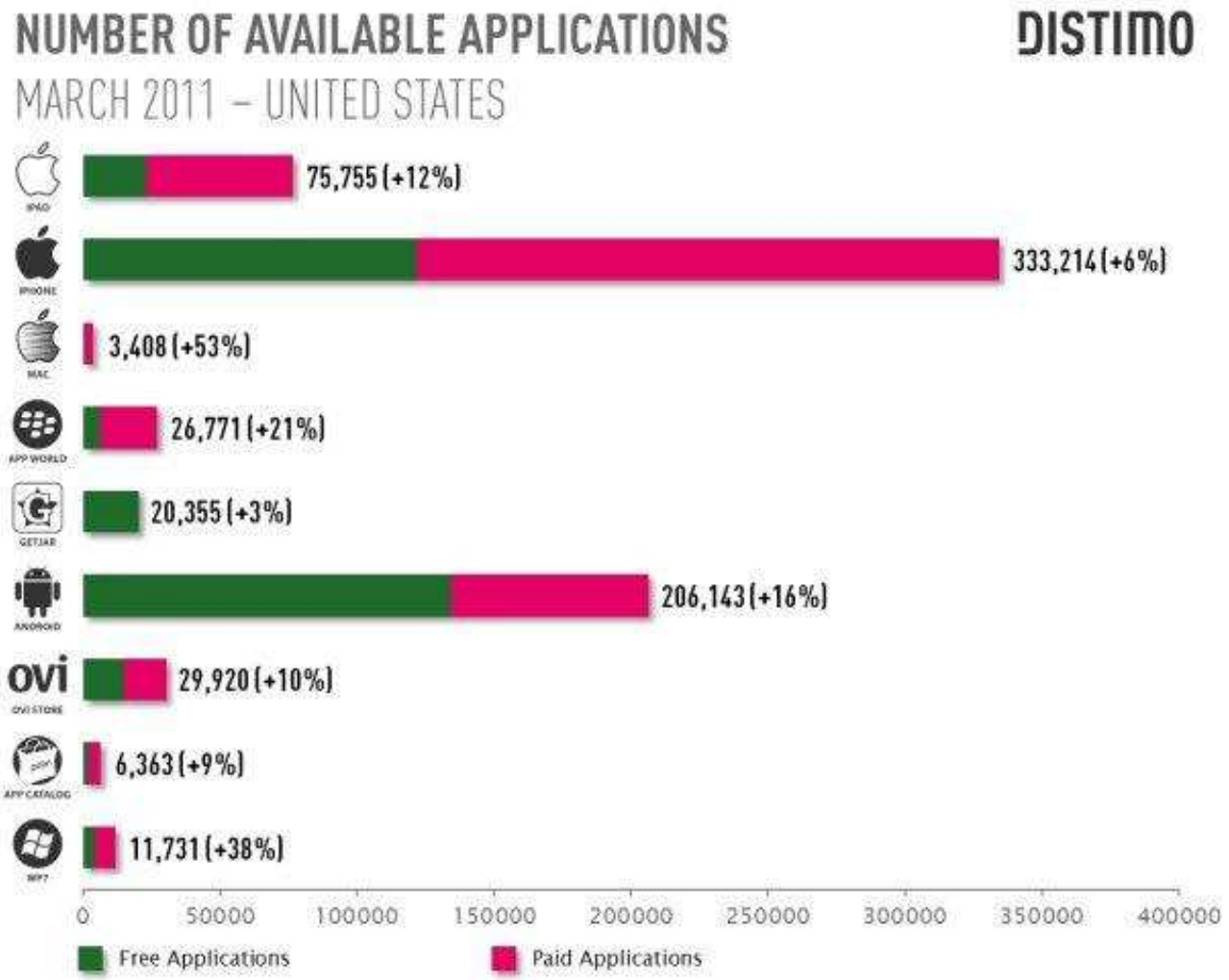
- Une plate forme ouverte, un ensemble de librairies, de composants logiciels pour les systèmes embarqués et mobiles

  - **Un système d'exploitation**
    - **Linux**
  - **Un intergiciel (middleware)**
  - **Nombreuses librairies**
    - **IHM,**
    - **Téléphonie,**
    - **Multimédia,**
    - **Capteurs,**
    - **Internet, cartographie**
    - **…**

Pourquoi Android ?

- **Indépendant d'une architecture matérielle**
- **Dédié aux systèmes embarqués et pas seulement**
- **Ambitions de Google/Apple**
  - **Marketing**
  - **Les applications**
    - **Nombreuses et gratuites sur AndroidMarket**
    - **Nombreuses et payantes sur AppStore**

Applications gratuites … Mars 2011



- Source : http://blog.vintive.com/nombre-dapplications-par-smartphone-les-chiffres/
-

Les autres

- Apple
- Microsoft
- Nokia
- Palm
- Research in Motion (BlackBerry)
- Symbian

- Quid de JavaFX et javaME ? ….
  - javaFX/Flash prometteur mais :
    - Lancé en 2009, qui l'utilise ?
      - http://java.sun.com/javafx/1/tutorials/core/
  - javaME obsolète ?

    - Smartphone ? Pour les pays riches et émergents ?

Projections selon Gartner
http://www.gartner.com/it/page.jsp?id=1622614

Table 1
Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units)

OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	**296,647**	**467,701**	**630,476**	**1,104,898**

Source: Gartner (April 2011)

Principes de base

- **Un aperçu en quelques diapositives**
  - **Architecture logicielle**
  - **Le simulateur, les API**
  - **Une démonstration**

Android les grandes lignes

- **Composants Android**
- **Outils de Développement**
- **Architecture Logicielle**

- **Développement**
  - **en java avec quelques directives et configurations en syntaxe XML**

- **Deux exemples**
  - **Démonstration**
    - **Deux exemples en quelques lignes de java**

Composants Android

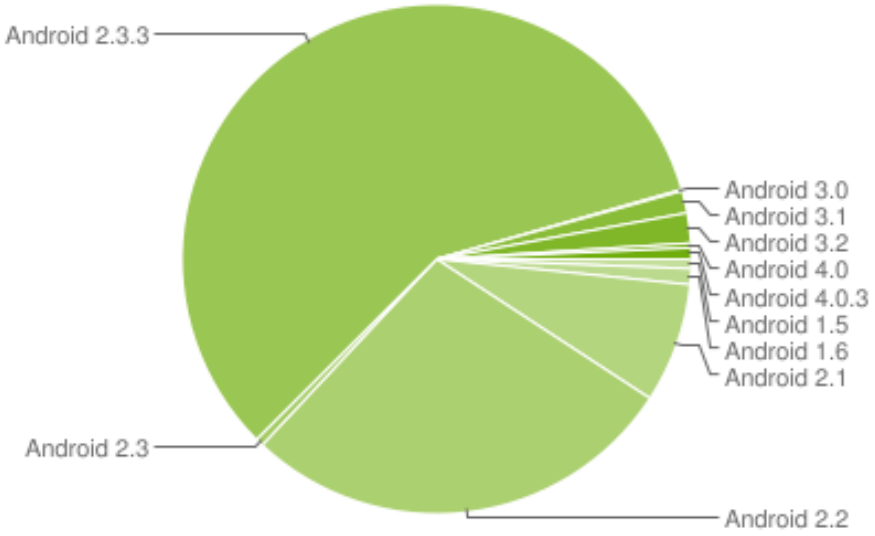
- **Framework de déploiement d'applications**
- **Dalvik comme machine virtuelle** (à registres != JVM de Sun à pile)
- **Navigateur intégré, WebKit** (webkit utilisé par safari, Google Chrome…)
- **SQLite**
- **Multimédia support PNG, JPG, GIF, MPEG4, MP3, H.263**

- **Dépendant du matériel**
  - **GSM**
  - **Bluetooth, EDGE, 3G, WiFi**
  - **Caméra, GPS, boussole et accéléromètre**
  - **Température,**
  - **…**

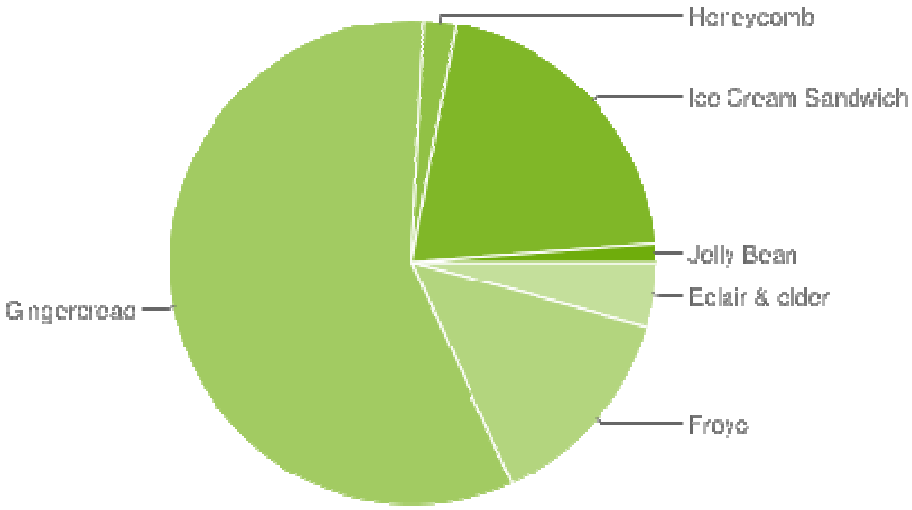
Outils de développement

- **SDK Android**
  - –En ligne de commandes
  - –Plug-in sous eclipse
  - –Émulateur
  - –Débogueur
  - –Traces fines d'exécution
  - –Tests unitaires
  - –Outils de mise au point
    - •Mesure de mémoire et performance

Quelle version ?



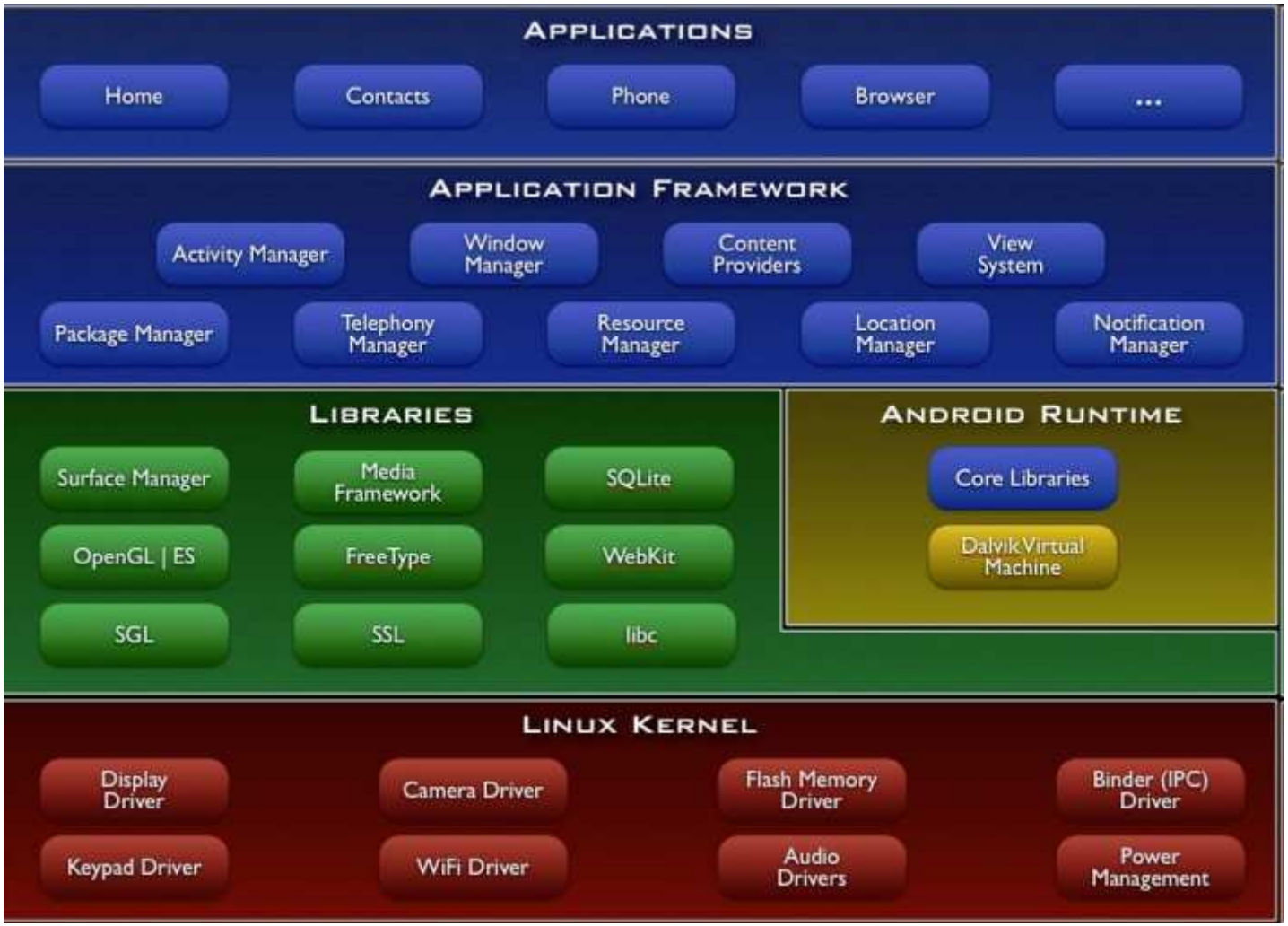
Mai 2012



Septembre 2012

- http://developer.android.com/resources/dashboard/platform-versions.html

Composants Android



http://developer.android.com/guide/basics/what-is-android.html

Dévelppement d'une application

1. **Obtention des .class**

2. **Génération de l'apk, Android Package file**

Développement 1/2 : Obtention des .class

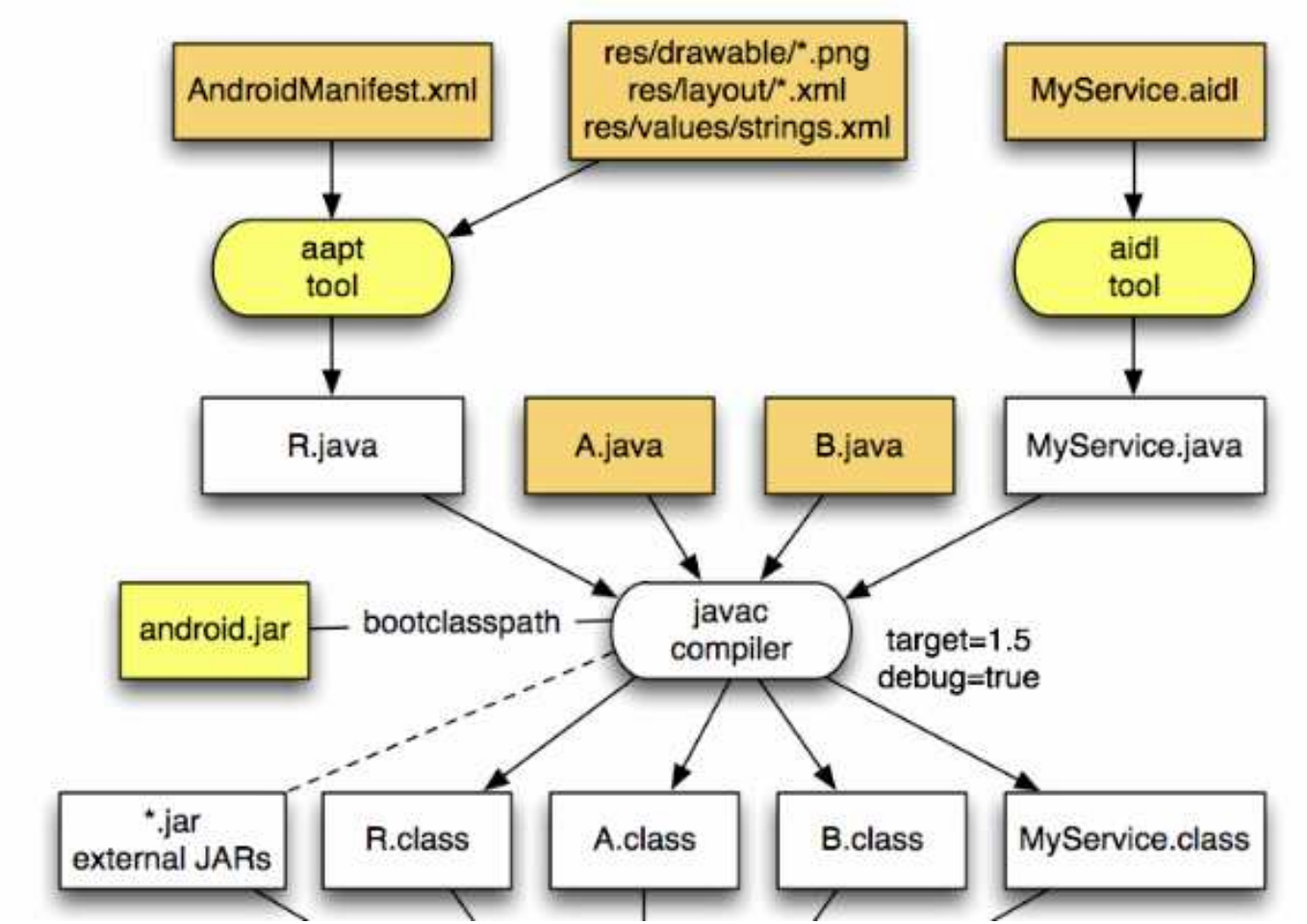
- **Fichier de configuration XML**
  - **Un source Java est généré, le fichier de ressources R.java**
    - **Configuration de l'application, IHM, String, …**
      - **Approche déclarative de l'IHM**
- **java**
  - **Paquetage java.lang,**
    - **Attention ce ne sont pas les librairies du JavaSE *(android.jar n'est pas rt.jar)***
- Compilateur javac de Sun/Oracle
  - javac –bootclasspath android.jar android/introduction/*.java

Développement 2/2 : Obtention de l'application

- **De tous les .class en .dex**
  - De la JVM à la machine Dalvik
  - D'une machine à pile en machine à registres
- **Génération de l'application .apk**
  - Une archive signée
  - Téléchargement : émulateur ou mobile

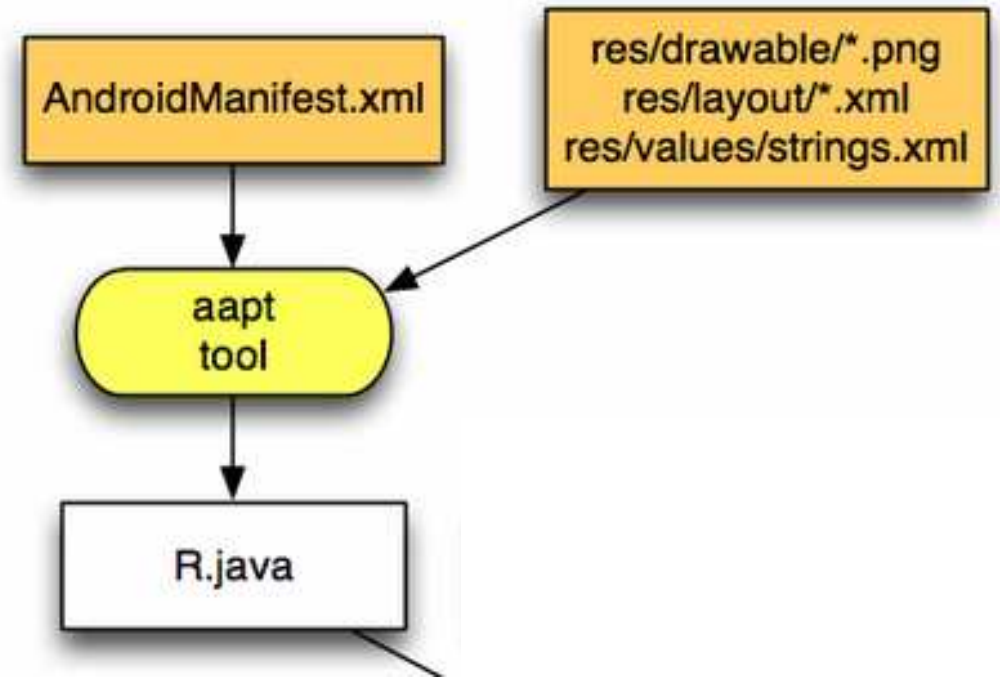
En détail, développement 1/2

Compilation : obtention des .class



Source : http://stuffthathappens.com/blog/wp-content/uploads/2008/11/android_flow.png

Aspect déclaratif et configuration

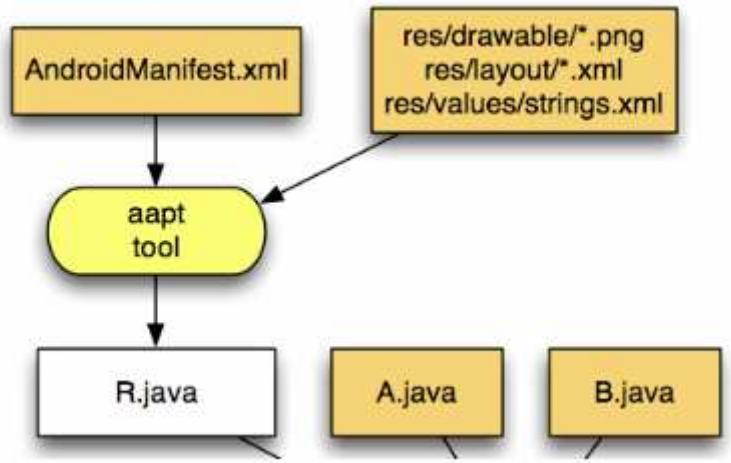


- **Syntaxe XML**
  - **R.java généré par l'outil aapt ou eclipse**

Développement 1/2
Fichier de configuration, AndroitManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="test.biblio"
      android:versionCode="1"
      android:versionName="1.0">

      <uses-permission android:name="android.permission.INTERNET" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Demo"
                  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>
```

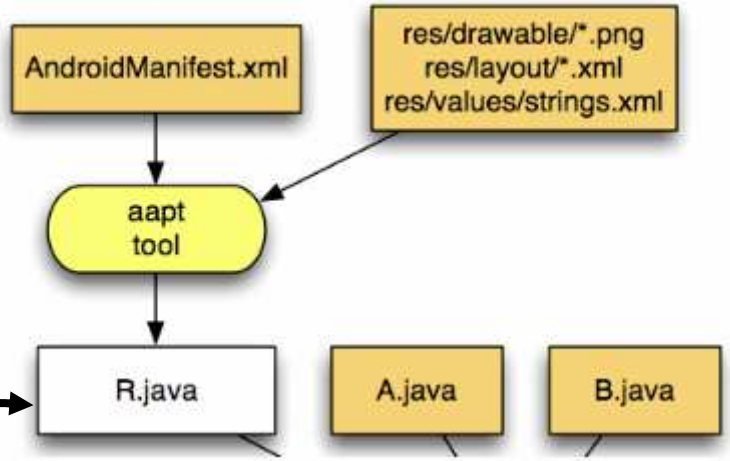
Le fichier de Ressources XML associé à l'IHM




```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  …
</LinearLayout>
```

- **Répertoire res/**
  - **Projet eclipse**
- **res/layout/main.xml**

Le fichier R.java

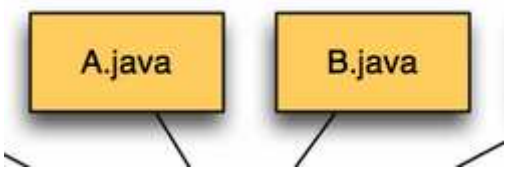



```
package test.biblio;
public final class R {
    …
    public static final class layout {
        public static final int main=0x7f030000;
    }}
```

- **/test/biblio/R.java** **AUTO-GENERATED FILE. DO NOT MODIFY.**

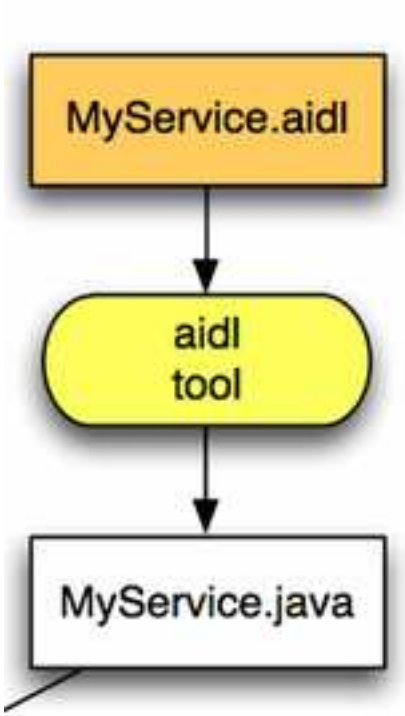
Un premier source java, juste pour la syntaxe…

```
package test.biblio;
import android.app.Activity;
import android.os.Bundle;

public class Demo extends Activity {
    ....
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

    setContentView(R.layout.main); // association IHM <-> Activity
        ....
}
}
```




```
package test.biblio;
public final class R {
    …
    public static final class layout {
        public static final int main=0x7f030000;
    }}
```

Services, même principe en tâche de fond



- **Une application sans IHM,**
  - **un couple<processus,DVM> peut lui être dédié**

Développement 2/2

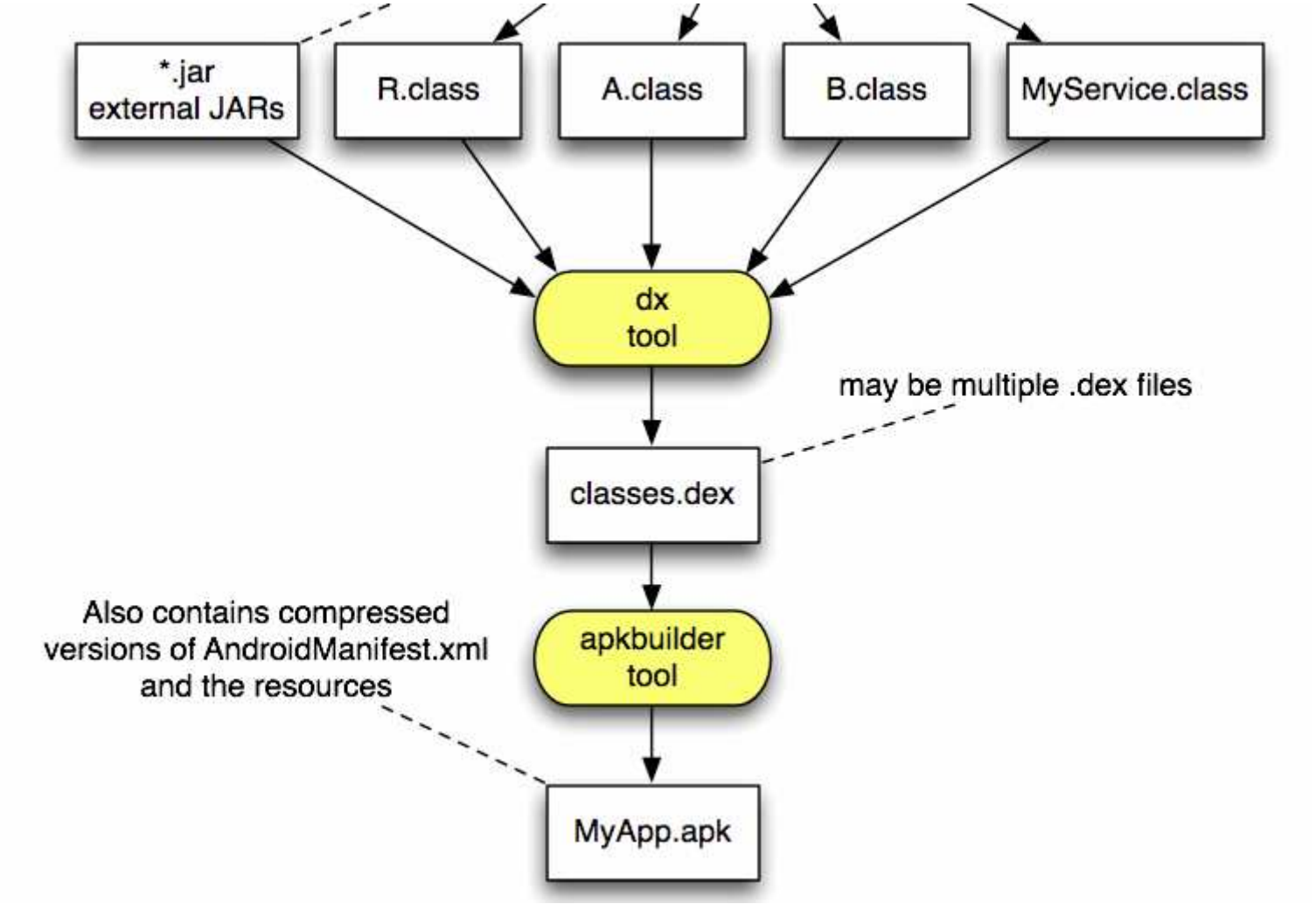
- **Génération de l'apk**

- **Assemblage des différents .class**

Conversion des .class en .dex

du « bytecode » en Dalvik Virtual Machine Code

Développement 2/2 Génération de l'application

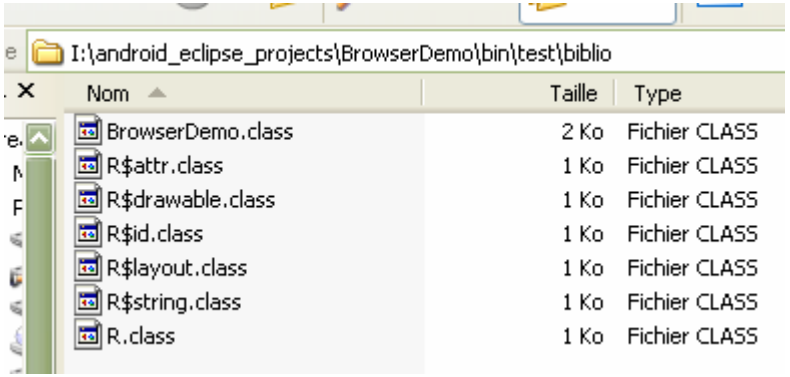


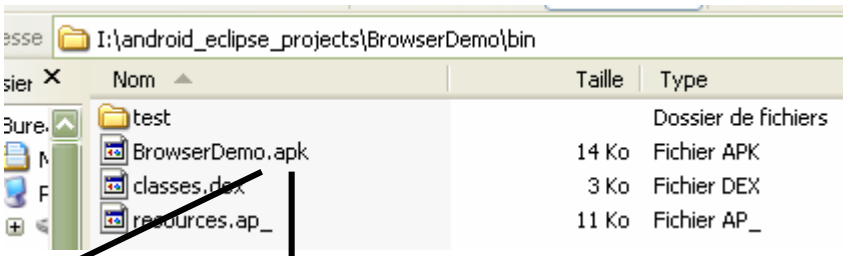
Source : http://stuffthathappens.com/blog/wp-content/uploads/2008/11/android_flow.png

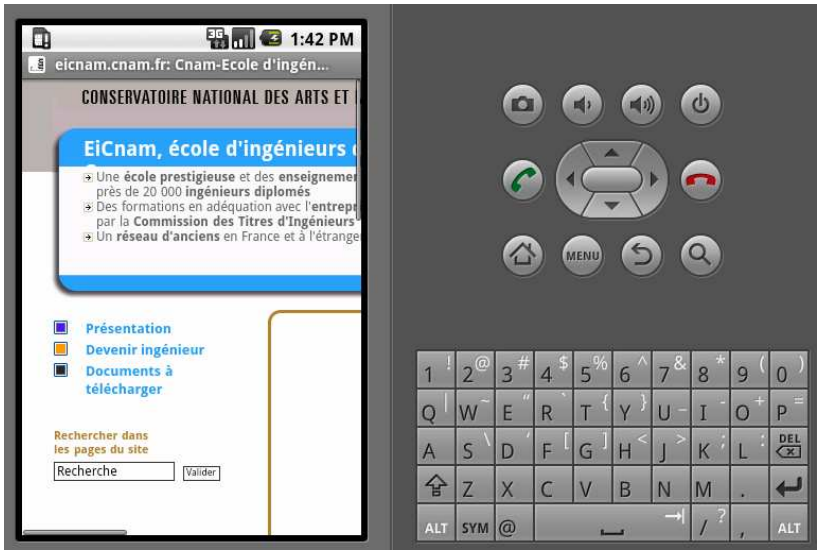
Développement 2/2, suite

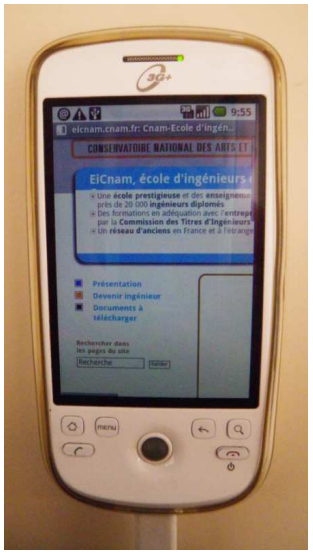
- **Du .class en .dex**
  - **Assemblage de tous les .class vers un .dex**
  - **Une machine par application, un processus Linux**
  - **Les applications communiquent via l'intergiciel**
    - **Une application peut être composée de plusieurs activités**
    - **Les activités communiquent via des variables globales, de la mémoire persistante,…**
- **Génération de l'application .apk**
  - **Assemblage, édition des liens**
  - **Une archive signée**
  - **Téléchargement : émulateur ou mobile**

Exécution émulateur et/ou mobile .apk



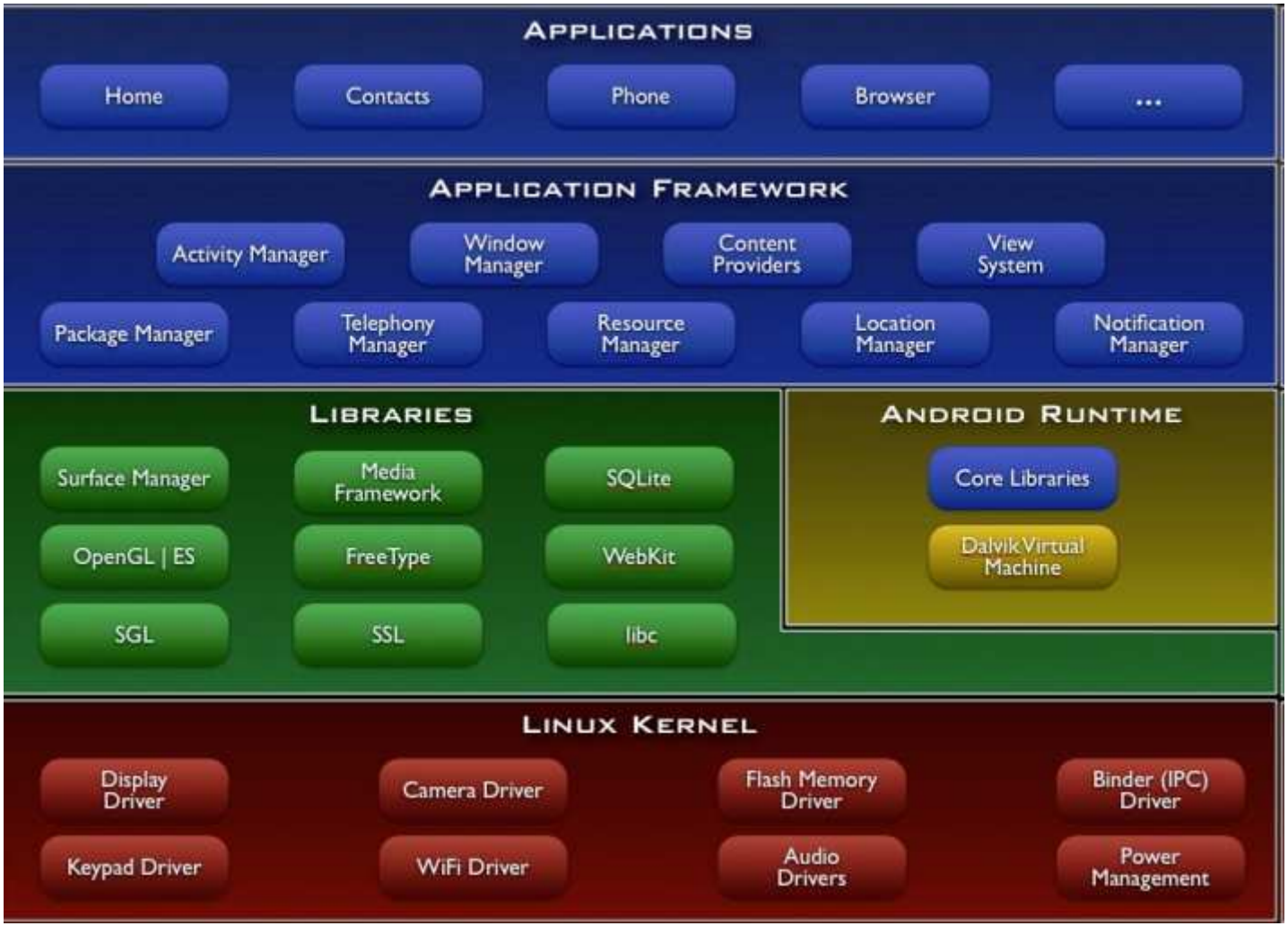






Démonstration

Composants Android



http://developer.android.com/guide/basics/what-is-android.html

Android OS

- **Un ensemble d'API**
  - http://developer.android.com/guide/basics/what-is-android.html

Middleware Android OS, un extrait

- **View** System listes, boutons,… navigateur (WebView)
- **Resource Manager,** accès aux String, aux descriptifs de l'ihm
  - R.java …
- **Activity Manager** gestion du cycle de vie d'une application
  - Une application android peut être composée de plusieurs activités
- **Content Providers** Accès aux données d'autres applications, partage, persistance de données
  - **Notification Manager** autorise des alertes dans la barre de statut
  - TelephonyManager
  - ….
  - http://developer.android.com/guide/basics/what-is-android.html

Librairies

C/C++ …

- SGL comme moteur pour le 2D
- FreeType comme fontes de caractères

Dalvik VM, au lieu de JVM

- **Machines à registres**

- **Chaque application à sa propre DVM**
  - **Communication inter-applications assurée par le middleware**
  - **Multi thread assuré par Linux**
  - **Accès aux capteurs par le noyau Linux**

Introduction aux Applications Android

- **Une présentation, vocabulaire**

- **Mots-clés**
  - **Applications,**
  - **Communication, évènements, intentions,**
  - **Services en tâche de fond,**
  - **Persistance.**

- **Une Application est composée d'une ou de plusieurs** *Activity*
  - *Une activity*
  - **Surcharge de certaines méthodes,**
    - **Du déjà vu : Applet, MIDlet, Servlet,…**
  - **Le cycle de vie est imposé par le framework**
    - **Déjà vu : pour une Applette init() puis start() …**

Avertissement

- **C’est une introduction**
  - **Le vocabulaire**
  - **Les grandes lignes**
  - **Quelques analogies seront faites avec du java « traditionnel »**

- **Les Essentiels**
  - **Activity**
  - **BroadcastReceiver**
  - **Service**
  - **ContentProvider**

Introduction … Classes

- **Activity**
  - Une interface utilisateur
    - Démarre d'autres activités, émet des évènements(intentions, intent)
  - Une configuration de type XML, permissions, librairies,
- **BroadcastReceiver**
  - Bus de messages
  - Émission et réception d'intentions
- **Service**
  - Pas d'interface, un service à rendre, en tache de fond
  - Intention de servir
- **ContentProvider**
  - Données rendues persistantes (pour d'autres applications)
  - Un fichier, base SQLite

Deux exemples, deux Activity

1. **Installation d'un navigateur en 2 lignes (WebView)**

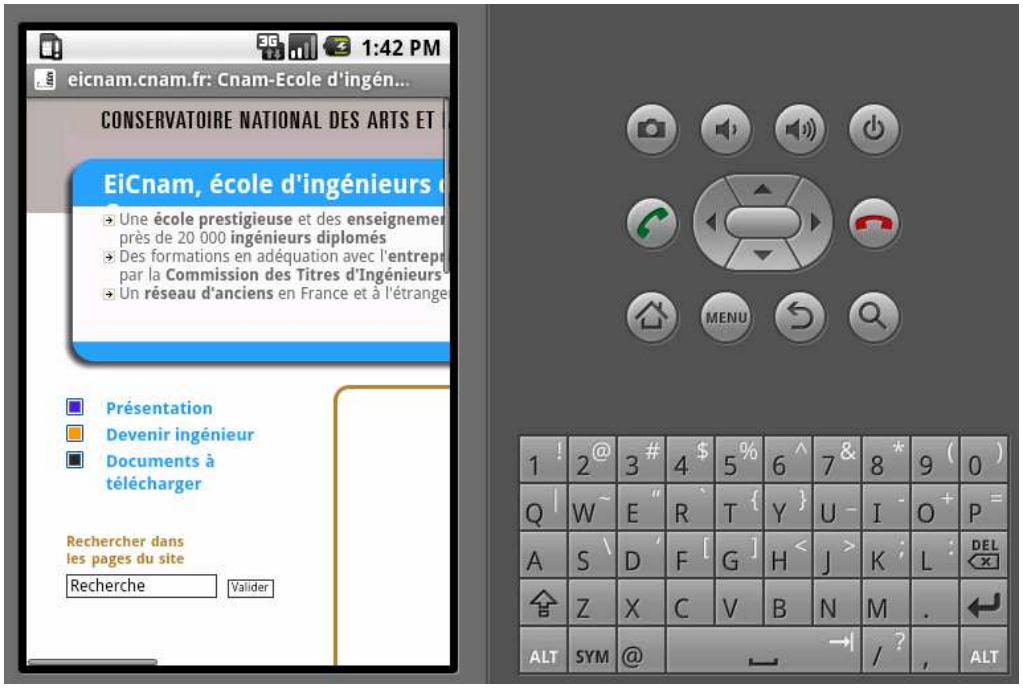
2. **Une toute petite IHM**
   - **Un écran constituée dun bouton, d'un écouteur,**
   - **A chaque clic, l'heure est affichée !**

   – **À télécharger ici**
     - **http://jfod.cnam.fr/seja/android/exemples/**

Activity Usage du WebKit, 2 lignes

```
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;

public class BrowserDemo extends Activity {
    private WebView browser;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        browser=(WebView)findViewById(R.id.webWiew);
        browser.loadUrl("http://eicnam.cnam.fr/");
    }
}
```

2 lignes



OnCreate est déclenché par le framework Android

```
public class BrowserDemo extends Activity {
    private WebView browser;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

// installation de l'IHM
        setContentView(R.layout.main);
// accès au composant graphique
        browser=(WebView)findViewById(R.id.webView);

        browser.loadUrl("http://eicnam.cnam.fr/");
    }
}
```

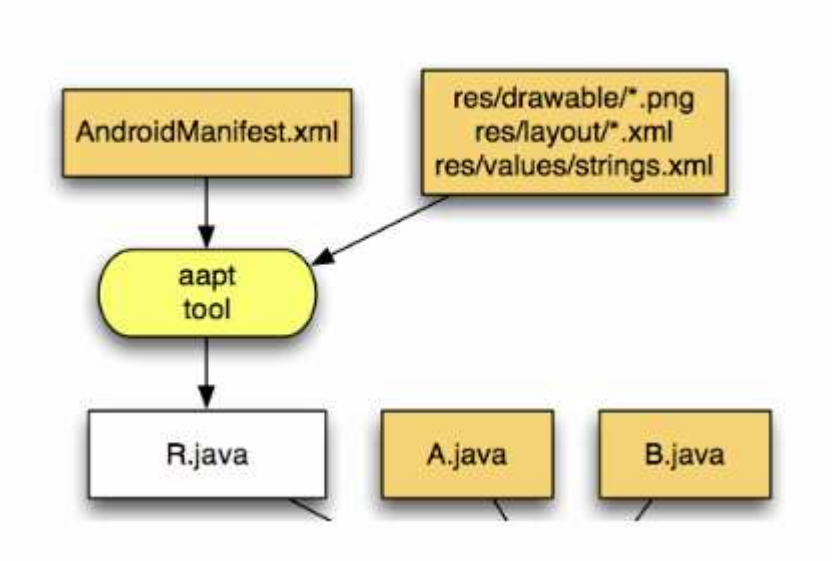
R.layout.*main*, R.id.*webView* ?

Une configuration XML

- **R.id.*webView*** ***?*** *R.layout.main ?*

  – ***En Entrée***
    - ***Fichiers de configuration XML***

  – ***En Sortie***
    - ***Source Java, R.java***



Une IHM, deuxième exemple

Une IHM

- **Un bouton, un écouteur, un clic et l'heure est affichée !**

- **En approche *traditionnelle***
  - **Tout est codé en Java IHM comprise**

– **En approche *déclarative***
  - **Usage d'XML pour la configuration, de java pour l'utilisation**

Activity Un Click et l'heure est actualisée

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import java.util.Date;


public class Now extends Activity implements View.OnClickListener {
  private Button btn;

  @Override
  public void onCreate(Bundle bundle) {
    super.onCreate(bundle);
    btn = new Button(this);       // <- un bouton
    btn.setOnClickListener(this);// <- un écouteur auprès de cette vue

    setContentView(btn);          // <- le bouton occupe l'écran
  }

  public void onClick(View view) {  // <- à chaque click
    btn.setText(new Date().toString());
  }
}
```

Notes : Vue apparentée swing
Ici un MVC à lui tout seul …

Activity Un Click et l'heure est actualisée

- **Approche déclarative, attribut onClick**

- **Démonstration**

Activity, méthodes à redéfinir

- **MonActivity extends** `android.app.`**`Activity`**`;`

```
@Override
  protected void onCreate(Bundle savedInstanceState){
```

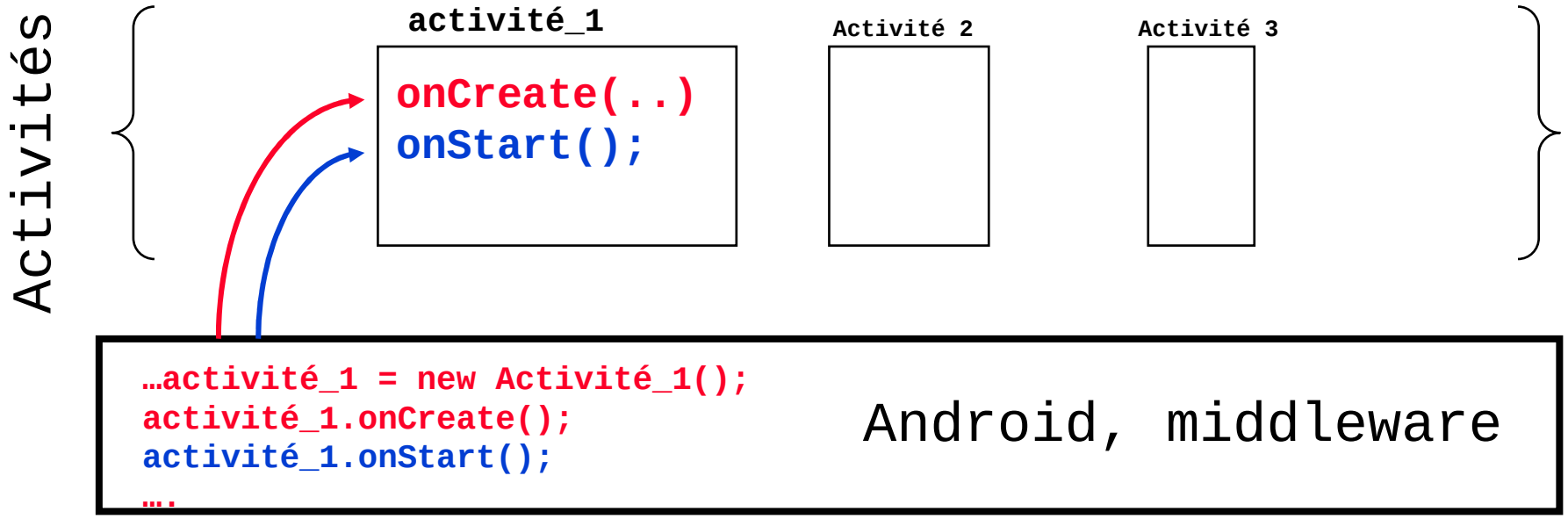
public class android.app.Activity

package android.app;

```
public class Activity extends ApplicationContext {

        protected void onCreate(Bundle savedInstanceState){

        protected void onStart();

        protected void onRestart();

        protected void onResume();

        protected void onPause();

        protected void onStop();

        protected void onDestroy();

        … etc …
}
```

induit un cycle de vie imposé par le « framework »

Inversion de Contrôle… Rappel




```
public class Activity extends ApplicationContext {
      protected void onCreate(Bundle savedInstanceState);
      protected void onStart();
       …..}
```

1

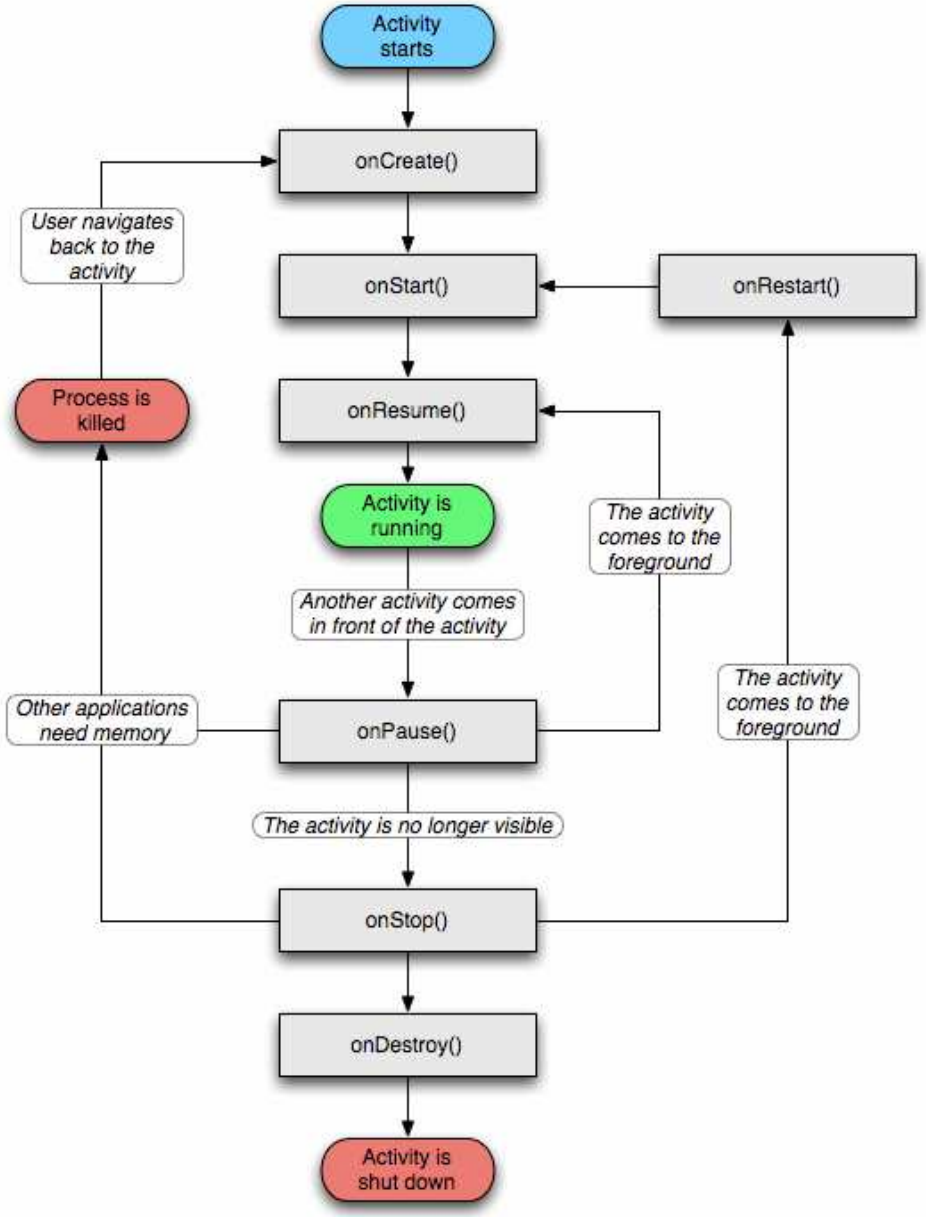
http://developer.android.com/reference/android/app/Activity.html

Activity : les états,
http://developer.android.com/reference/android/app/Activity.html

C'est le framework
qui contrôle tout

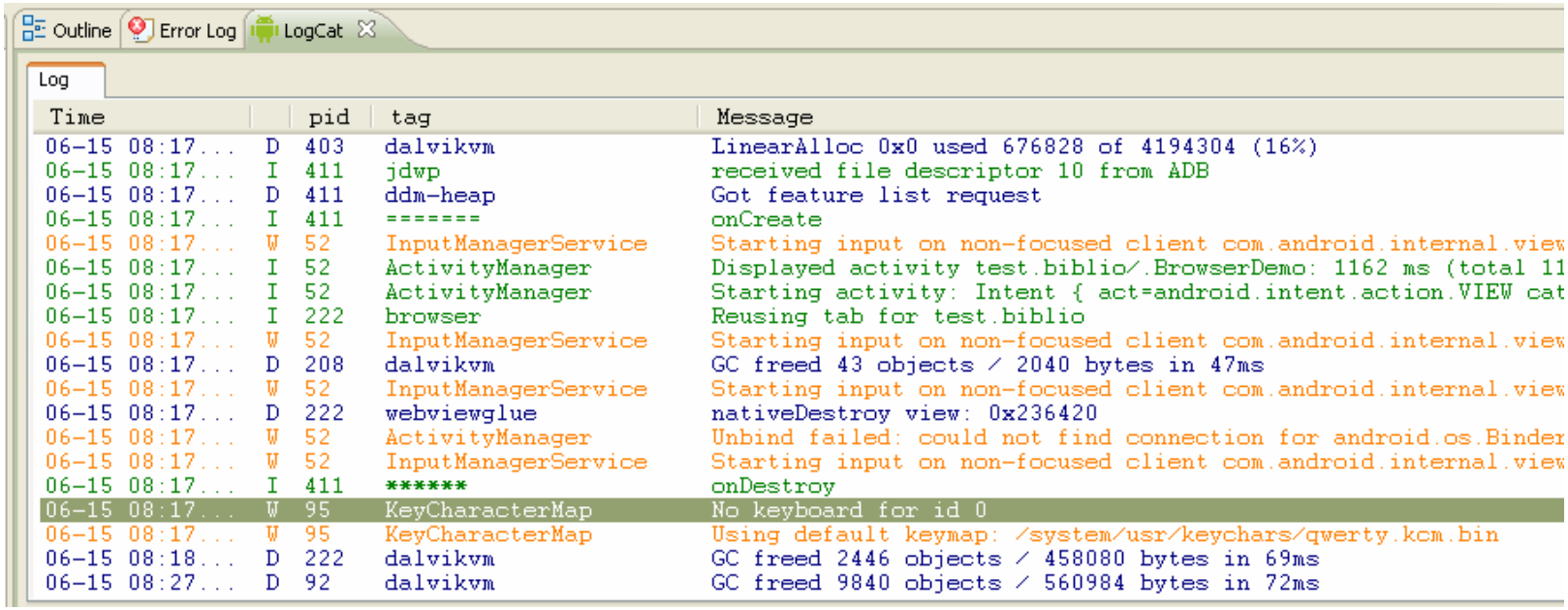
En tache de fond :
Empiler(l'activité);

onResume
activité au 1er plan = Dépiler()

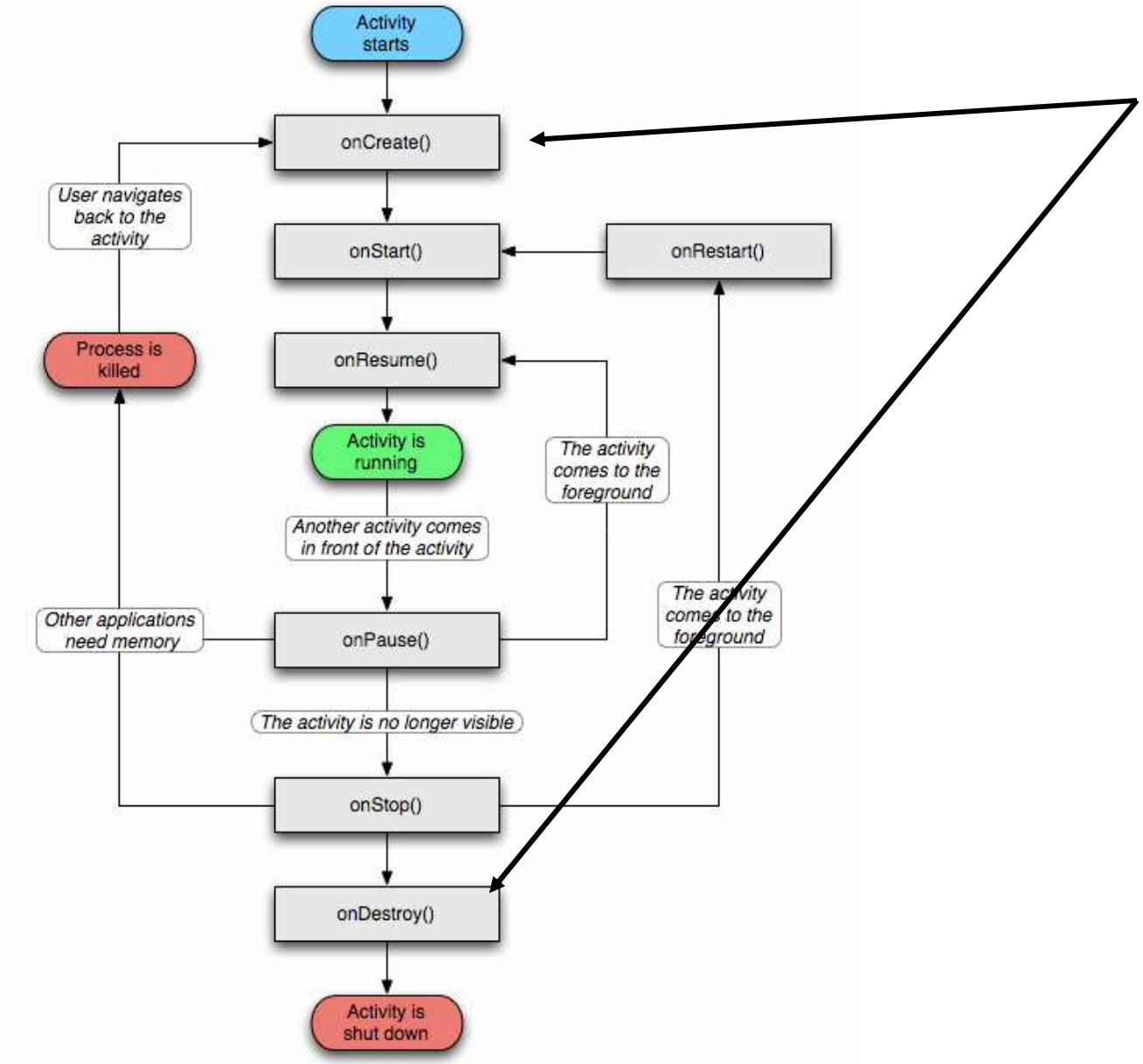


Démonstration, Activity dans tous ses états

```
public class BrowserDemo extends Activity {
  private WebView browser;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.i("=======","onCreate");
        // cf. page précédente
    }
    public void onDestroy(){
      super.onDestroy();
      Log.i("******", "onDestroy");
    }
}
```



En résumé : émulateur + LogCat , traces bien utiles



Démonstration, Activity dans tous ses états

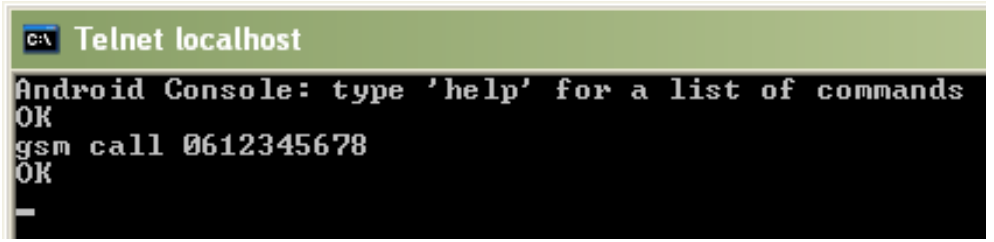
```
public class BrowserDemo extends Activity {

  public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);Log.i("=======","onCreate");}

   public void onStart(){super.onStart();Log.i("=======","onStart");}

   public void onResume(){
     super.onResume();
     Log.i("=======","onResume");
   }

   public void onPause(){
     super.onPause();
     Log.i("=======","onPause");
   }

   public void onStop(){
     super.onStop();
     Log.i("******","onStop");
   }

    public void onDestroy(){
      super.onDestroy();
      Log.i("******", "onDestroy");
    }
}
```

OnPause -> onResume

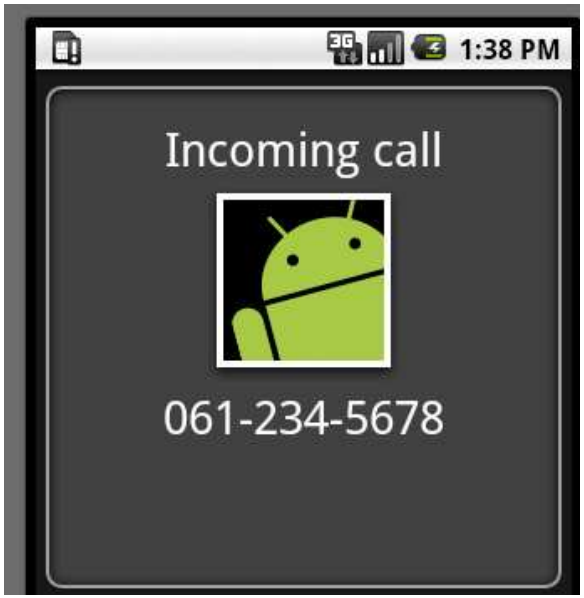
1)

OnPause

2-1) telnet localhost 5554



2-2)

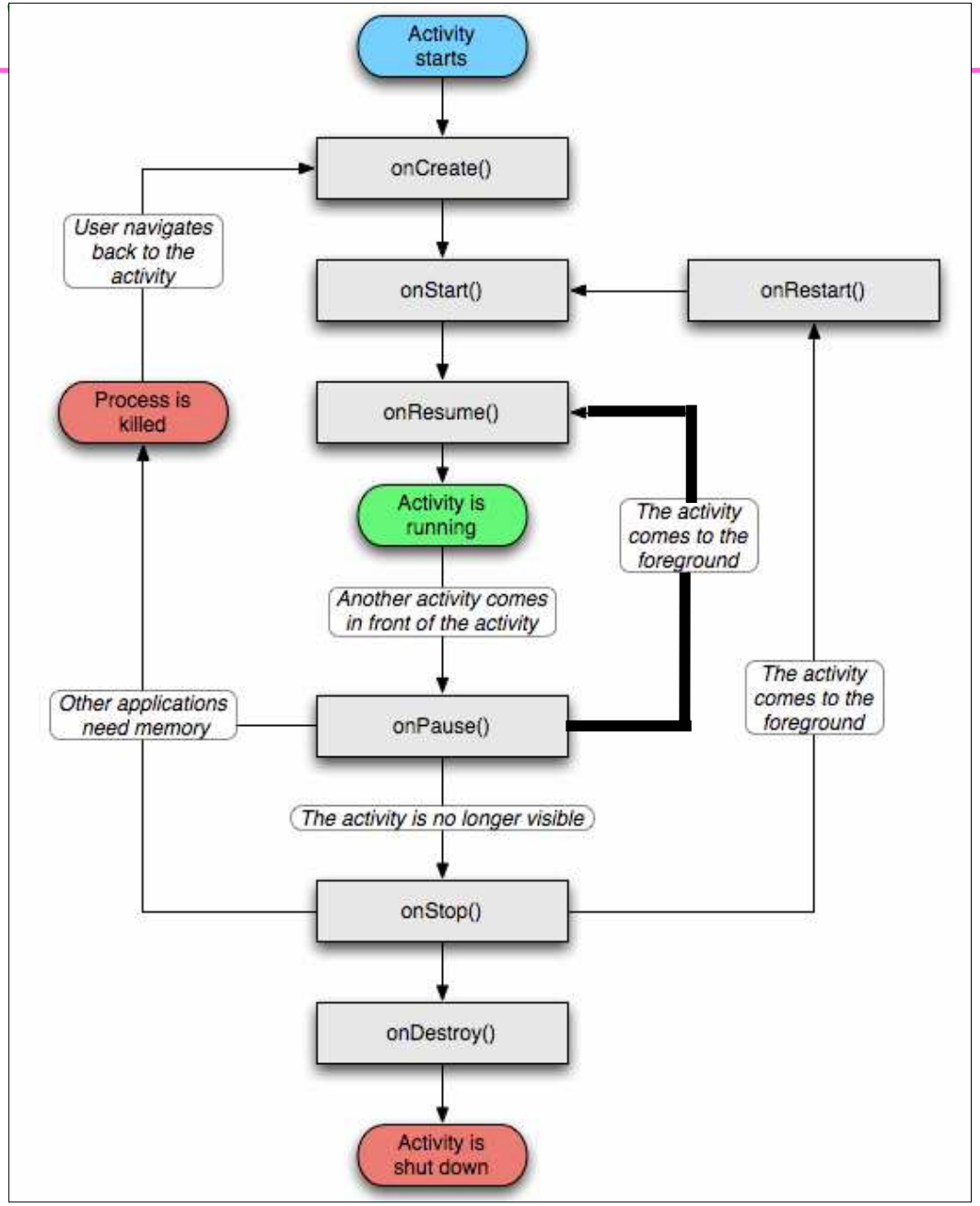


2)

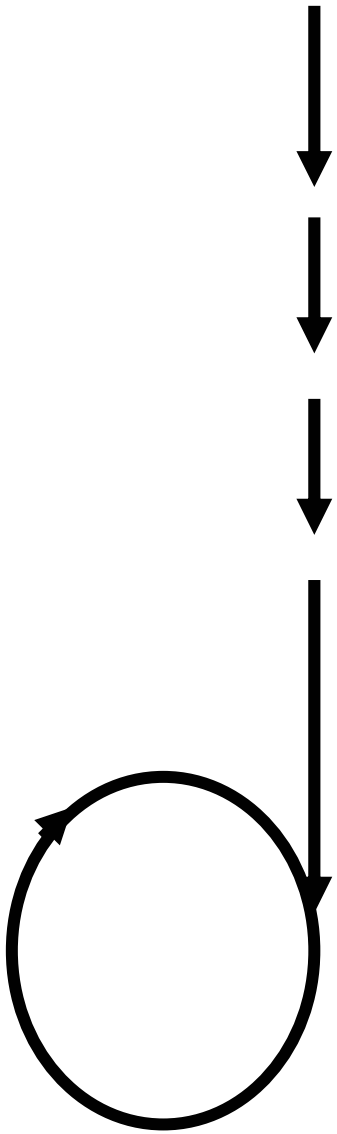
OnResume

3)

En résumé :


Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
Other applications need memory
onPause()
The activity comes to the foreground
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Le fil d'exécution, une activity A



1. **Démarrage d'une A, un processus, une DVM**
   - **Création de l'instance par Android, un thread (main) lui est associé**
2. **Appel de onCreate()**

- **Appel de onStart()**
- **Appel de onResume()**
- **A est dans une boucle d'attente des évènements**
  - **Évènements de l'utilisateur**
  - **Intention du système**
  - **Un appel téléphonique**

Application, Activity …

- un processus linux contient une application,
- Une application, peut contenir une ou plusieurs activités,
- Une activité se trouve dans un certain état, cf. cycle de vie
- Les threads locaux au processus, sont indépendants d'une activité
- Une activité peut être dans un processus Linux