

Laurent Provot <a href="mailto:laurent.provot@uca.fr">laurent.provot@uca.fr</a>

Septembre 2017

# Introduction



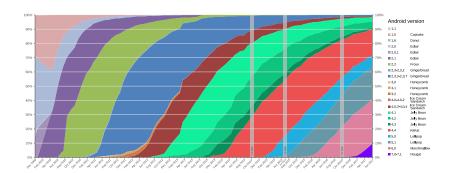
### L'environnement

### Spécificités du développement sur plateformes mobiles :

- Ressources limitées
- IHM variées
  - Écrans tactiles
  - GPS
  - Accéléromètres
  - Caméras
- Diversité des appareils
- Ultra-connectés



## La plateforme Android



### ■ Évolution très rapide

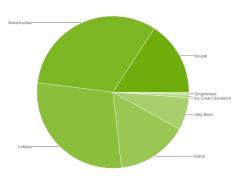
<a href="https://www.bidouille.org/misc/androidcharts">https://www.bidouille.org/misc/androidcharts</a>



## La plateforme Android

### Fragmentation des versions au 11 septembre 2017

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.4%
4.2.x		17	3.5%
4.3		18	1.0%
4.4	KitKat	19	15.1%
5.0	Lollipop	21	7.1%
5.1		22	21.7%
6.0	Marshmallow	23	32.2%
7.0	Nougat	24	14.2%
7.1		25	1.6%



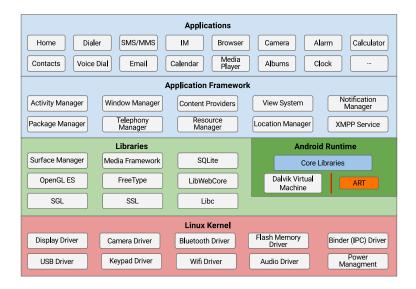
<a href="https://developer.android.com/about/dashboards/index.html">https://developer.android.com/about/dashboards/index.html</a>



# Architecture d'Android



## Architecture logicielle d'Android





# Organisation du système de fichiers

- Boot image : Noyau Linux et RAMDisk
- System image : OS Android et Applications
- Data image : Données utilisateur persistantes
- Recovery image : Utilisé pour recompiler / mettre à jour le système
- Radio image : Gère la partie « connectivité » de l'appareil



## Architecture d'une application Android



▼ 🛅 res layout ▶ in layout-land ▶ menu ▶ mipmap-hdpi ▶ impmap-mdpi ▶ impmap-xhdpi ▶ impmap-xxhdpi ▶ □ values ▶ □ values-sw600dp ▶ □ values-w820dp AndroidManifest.xml aitignore. I app.iml build.gradle proguard-rules.pro

## Les ressources d'une application Android

Dossier res : stockage des fichiers ressources

- anim : fichiers d'animations
- color : fichiers de description de listes de couleurs d'état
- drawable : images de résolutions différentes pour les écrans
- layout : fichiers XML décrivant la structure de écrans de l'application
- menu : définition XML de menus
- raw : ressources définies dans des fichiers bruts (mp3, avi, ...)
- values : stockage clé/valeur de données simples (string, dimensions, couleurs, ...)
- xml : configuration d'objets avec des paramètres stockés en XMI



### Qualificatifs de ressources

- 1 Country code du système (mcc310-mnc004)
- Code langage / region (fr-rFR)
- 3 Direction de lecture (*Idrtl*)
- 4 Tailles (xlarge)
- 5 Orientation de l'écran (land)
- 6 Densité de pixel (xxhdpi)
- **7** Version d'API (v11)

<a href="http://developer.android.com/guide/topics/resources/providing-resources.html">http://developer.android.com/guide/topics/resources/providing-resources.html</a>



rchitecture d'Android

Fichier AndroidManifest.xml: fichier de description de l'application, présent obligatoirement à la racine de l'application.

- Nom du package (Android) de l'application
- Descriptions des entités avec leurs capacités et comportements, ainsi que la manière dont elles sont lancées (launchMode)

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

</manifest>



■ Donne le point d'entrée de l'application (activité lancée en première)

```
<application
  android:icon="@drawable/ic launcher"
  android:label="@string/app_name"
  android:theme="@style/AppTheme" >
  <activity android:name="fr.iut.pm.od.CrimeListActivity" >
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
  <activity
    android:name="fr.iut.pm.od.CrimeActivity"
    android:label="@string/app_name" >
    <meta-data android:name="android.support.PARENT ACTIVITY"</pre>
      android:value=".CrimeListActivity"/>
  </activity>
</application>
```



- Identifie les périphériques nécessaires pour faire fonctionner l'application
  - Puce Caméra, NFC, GPS, ...

```
<uses-feature android:name="android.hardware.camera" />
```

- Identifie les permissions
  - INTERNET
  - ACCESS\_FINE\_LOCATION
  - CAMERA
  - READ\_CONTACTS
  - WRITE\_EXTERNAL\_STORAGE
  - SEND SMS
  - ..

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

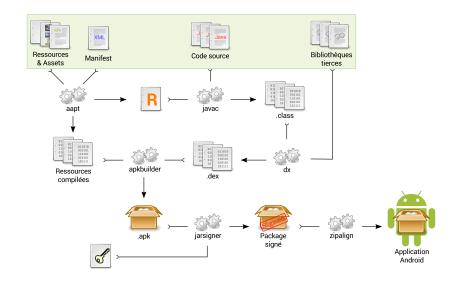


- L'attribut <uses-sdk>
- Donne les niveaux d'API de l'application (pas les versions)
  - minSdkVersion: niveau minimum d'API nécessaire pour le fonctionnement de l'application. S'il n'est pas renseigné, Android considère que l'application est compatible avec toutes les versions
  - targetSdkVersion : version où le fonctionnement normal de l'application a été testé
  - maxSdkVersion : attribut inutile depuis la compatibilité ascendante des API Android.

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17"/>



## Processus de construction d'une application





# Les applications Android



## Les entités principales

### Les composants applicatifs :

- L'activité (Activity) : Une fonctionnalité qu'un utilisateur peut faire (modifier une note, numéroter un appel)
- Le service (Service) : Processus en arrière plan (jouer une musique, mettre à jour l'icône de température)
- Le fournisseur de contenu (Content Provider) : Gère, sauvegarde et récupère, des données (récupération des données d'un contact)
- Le gaget (App Widget): vue miniature d'application pouvant être intégrée à une autre application (gaget météo de l'écran principal)

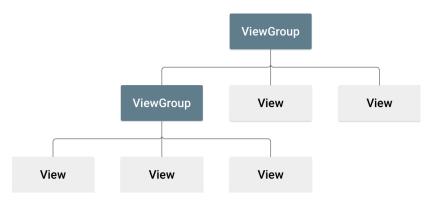
## Les entités principales

#### Les composants d'interaction :

- L'intention (Intent) : Communications asynchrone entre les entités du système (démarrer une application)
- Le récepteur (Broadcast Receiver) : Reçoit et traite des messages (déclencher une alarme à la réception d'un message)
- Les feedbacks (Notification, Toast) : un message affiché dans la barre de notification ou une info-bulle temporaire (arrivée d'un mail, traitement photo terminé)

## La partie visible des applications

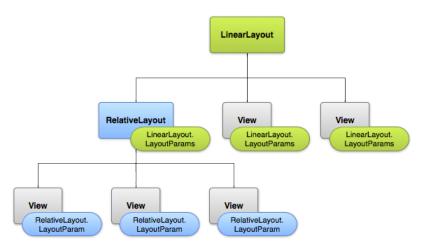
■ Gestion par design pattern Composite





## L'agencement des vues

 Organisations par Layouts (comme en JavaFX, mais pas les mêmes)



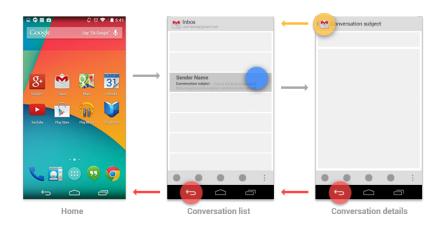


# Navigation sous Android

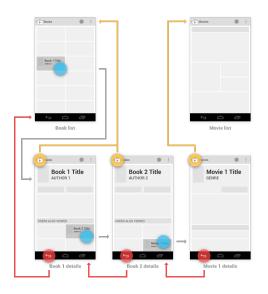
- 2 patterns classiques : Back Navigation et Up Navigation
- Back Navigation
  - $\blacksquare$  avec la touche  $\bigcirc$  (ou  $\bigcirc$  pour les ancienne versions)
  - navigation chronologique (en fait à travers les backstacks)
- 2 Up Navigation
  - avec l'App Bar
  - navigation hierarchique entre les activités

<a href="https://developer.android.com/design/patterns/navigation.html">https://developer.android.com/design/patterns/navigation.html</a>

## Navigation sous Android



## Navigation sous Android





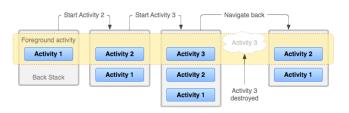
Les applications Android

# Gestion des applications

- Android N'est PAS mono-processus
- Chaque application est lancée dans sa propre machine virtuelle ART (ou Dalvik)
  - Mémoire propre
  - Espace disque réservé pour une application
  - UID propre
- Une application n'accède pas directement à l'espace disque d'une autre

# Fonctionnement des applications Android

- Le système Android gère des tâches (tasks)
- Chaque tâche possède sa propre pile d'activités (back stack)
  - Gérée par défaut comme une structure LIFO
- Appui sur Back : dépile l'activité au sommet de la pile et la détruit
- Appui Home : retourne à l'écran d'accueil et met la tâche en attente en arrière plan





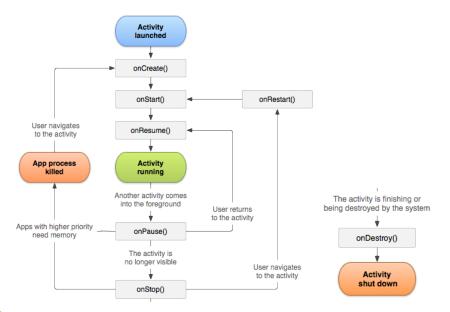
# Tâches et navigation





Les applications Android

## Cycle de vie d'une activité

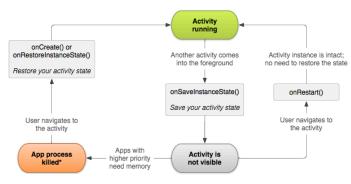




27 / 29

## Cycle de vie d'une activité

- Si besoin de ressources, une tâche mise en attente peut être détruite
- Attention à bien gérer les données transitoires de l'application qui doivent être sauvegardées !
- onSaveInstanceState(Bundle outState) et onRestoreInstanceState(Bundle savedInstanceState)





# Cycle de vie d'une activité

- Implémentation par défaut des View : sauvegarde automatique si elles possèdent un ID (android:id)
- Pour données transitoires, si données persistantes alors onPause()