

# NFA025 : Mise en œuvre de la programmation de smart-phones et tablettes tactiles

---

*Jean-Ferdy Susini*  
*Maître de Conférences - CNAM*  
*Département Informatique*

le cnam

Paris, 14 févr. 2013



# Le système Android

---

Sources : <http://www.android.com>, [wikipedia...](#)

le cnam

Paris, 14 févr. 2013



# Android

3

- Système d'exploitation Open Source (licence Apache) ciblé **principalement** sur la téléphonie mobile et les tablettes tactiles
- Initialement développé par la société éponyme rachetée par Google en 2005
- premier SDK publié en nov. 2007 et création de l'OHA (Open Handset Alliance)

<http://developer.android.com/index.html>

# Les outils de développement Android

4

- l'Android SDK (Software Development Kit) amène des outils :
  - Android SDK Manager (commande android)
  - les API (voir à <http://developer.android.com/reference/packages.html>)
  - AVD (Android Virtual Device) gère les émulateurs
  - adb (Android Debug Bridge)
  - DDMS (Dalvik Debug Monitor Service)
  - D'autres outils : aapt (Android Asset Packaging Tool), sqlite, zipalign, dx, llvm-c, aidl...

# Les versions

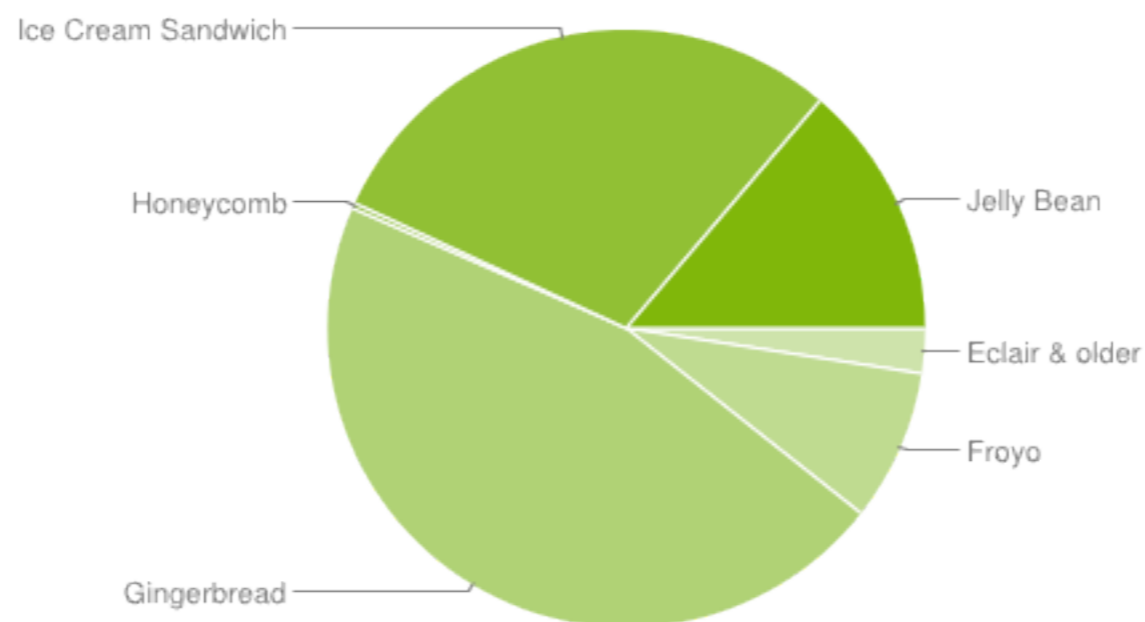
5

1.0		1 nov. 2007	version développeurs, distribuée avec le SDK avant la sortie du premier téléphone Android
1.1	Petit Four	1 oct. 2008	version incluse dans le premier téléphone, le HTC G1/Dream
1.5	Cupcake	1 avr. 2009	nouvelles fonctionnalités et mises à jour de l'interface graphique
1.6	Donut	1 sept. 2009	nouvelles fonctionnalités et mises à jour de l'interface graphique
2.0	Eclair	1 oct. 2009	nouvelles fonctionnalités et mises à jour de l'interface graphique
2.1	Eclair	1 janv. 2010	correction du trop grand nombre de bugs de la 2.0
2.2	Froyo	1 mai 2010	vitesse améliorée (JIT), nouvelles fonctionnalités et mises à jour de l'interface graphique
2.3	Gingerbread	1 déc. 2010	dernière version dédiée uniquement aux smartphones. Cette version est parfois utilisée sur de petites tablettes.
3.0	Honeycomb	1 févr. 2011	réservé aux tablettes tactiles et aux téléviseurs connectés <sup>12</sup> , cette mise à jour comprend de nombreux changements dans l'interface
4.0	Ice Cream Sandwich	1 oct. 2011	cette nouvelle version, fortement inspirée d'Honeycomb, unifiée pour smartphones, tablettes et Google TV apporte de nombreux changements
4.1	Jelly Bean	1 juil. 2012	ajoute un système de notification améliorée, Google Now un système concurrent de Siri et le Project Butter qui augmente la fluidité d'Android;
4.2	Jelly Bean	1 nov. 2012	nouvelle interface de l'appareil photo et l'introduction de Photosphère permettant une prise des photos à 360° type Street View, d'un système multi-compte uniquement sur tablette, de Type Gesture permettant d'écrire avec le clavier rien qu'en glissant le doigt et d'améliorations de Google Now...

# Quel API choisir ?

6

4.2.x	Jelly Bean	13 nov. 2012	17	1,40 %
4.1.x	Jelly Bean	9 juil. 2012	16	12,20 %
4.0.x	Ice Cream Sandwich	19 oct. 2011	14-15	29 %
3.x.x	Honeycomb	22 févr. 2011	11-13	1,30 %
2.3.x	Gingerbread	6 déc. 2010	9-10	45,60 %
2.2.x	Froyo	20 mai 2010	8	8,10 %
2.0 ; 2.1	Eclair	26 oct. 2009	5-7	2,20 %
1.6	Donut	15 sept. 2009	4	0,20 %



<http://developer.android.com/about/dashboards/index.html>

# Structure du système Android

7

*Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).*



*This document is shared by Jean-Ferdy Susini according to terms described in the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#).*



# Structure du système Android

7

Noyau modifié écrit essentiellement en C et en assembleur :

- ajout d'une couche d'abstraction des périphériques
- ajout d'un mécanisme d'IPC spécifique : les Binders



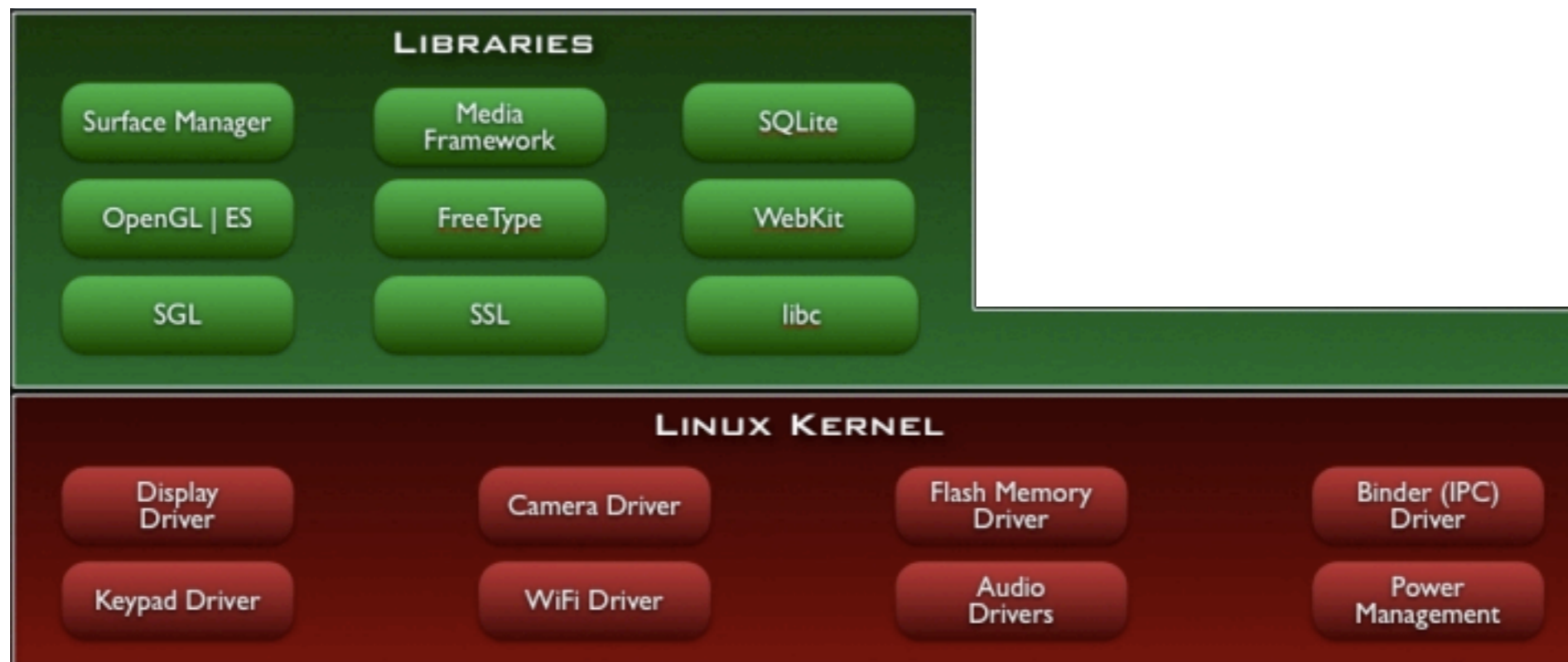
*Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).*



# Structure du système Android

7

Ensemble de bibliothèques écrites en C/C++, offrant les principaux services du système d'exploitation, ces bibliothèques sont : soit des adaptations (parfois conséquentes) issus du système GNU/Linux soit des créations spécifiques

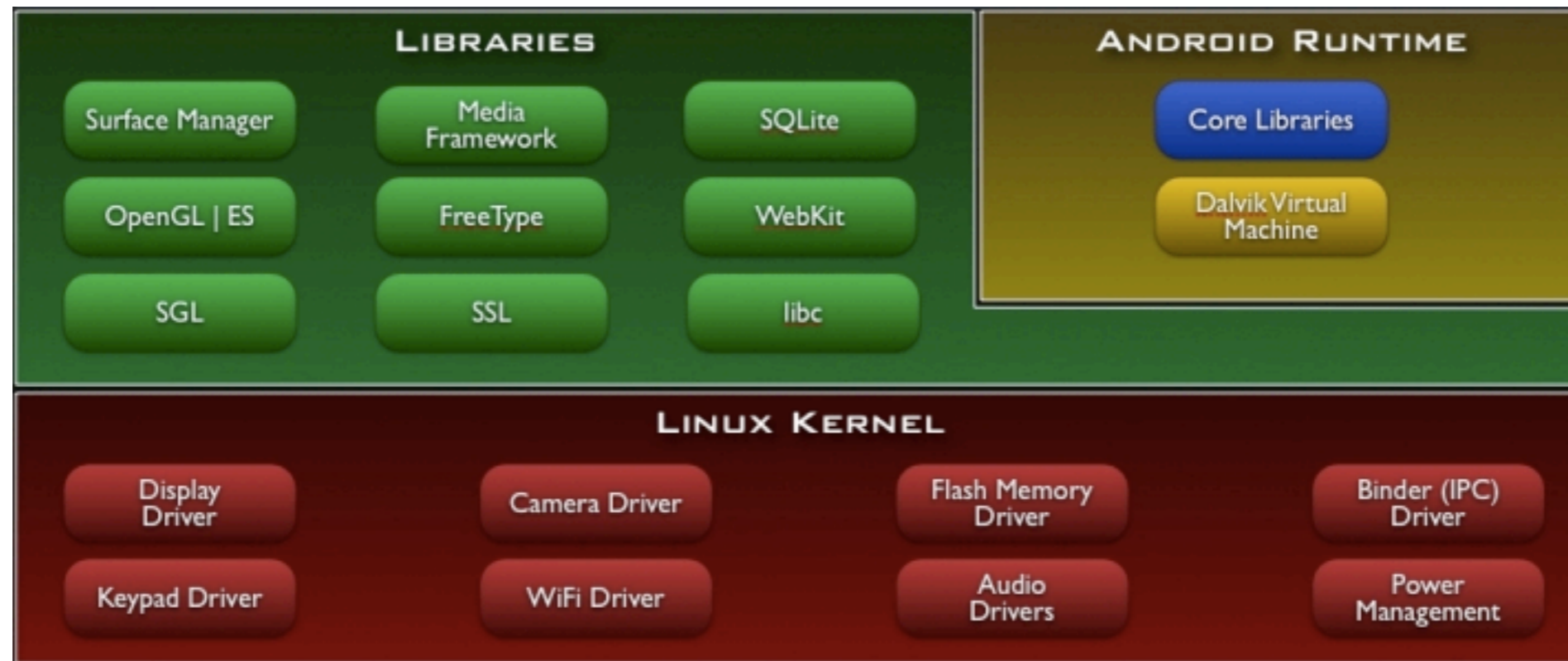


*Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).*

# Structure du système Android

7

Le Runtime Android écrit en C/C++ et Java, apporte le support Java (non officiel) à travers la Dalvik VM et les librairies implantant les APIs Android (Core Libraries)

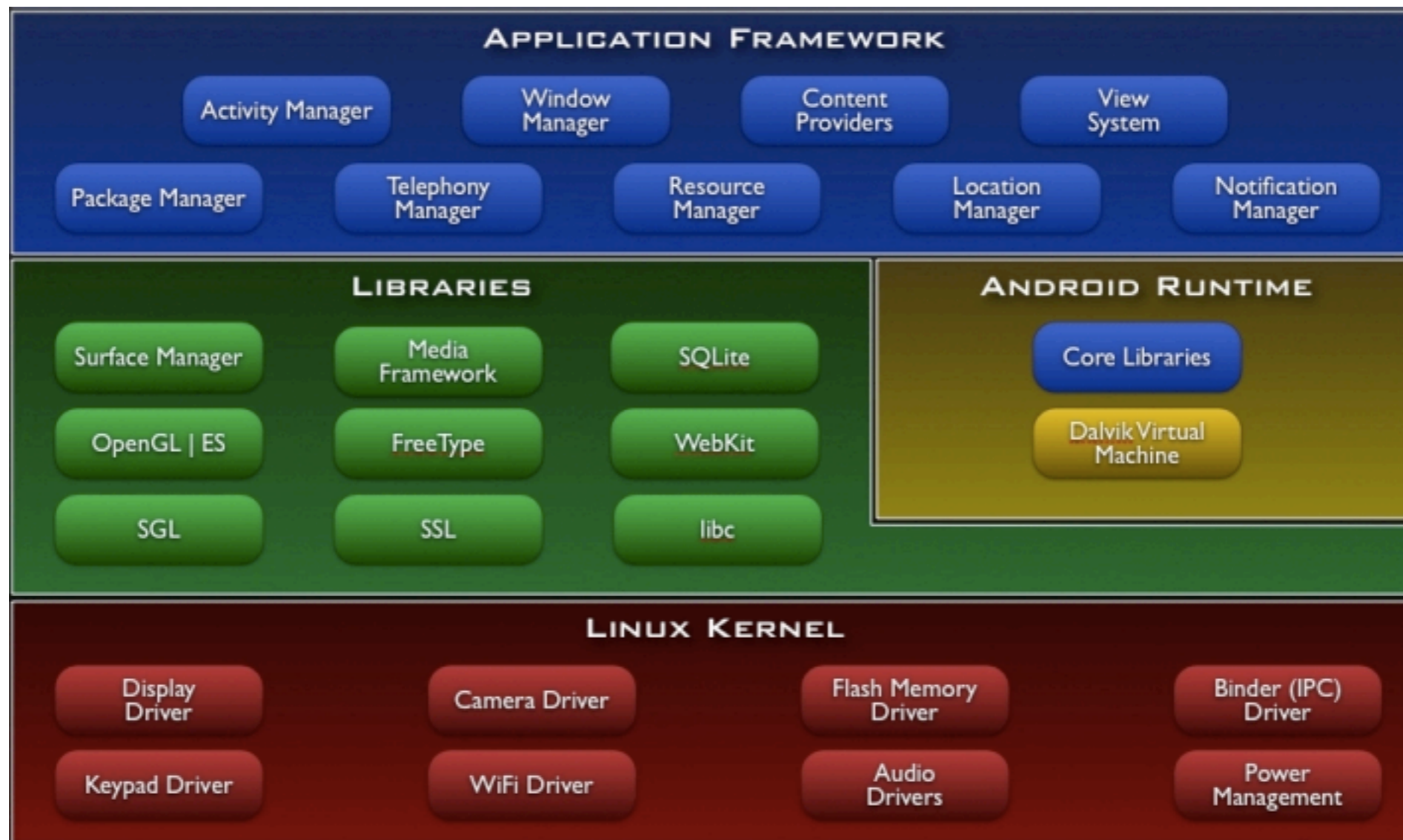


*Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).*

# Structure du système Android

7

Les composants applicatifs implémentant les services de haut niveau, écrit en Java (parfois aussi en utilisant le NDK)

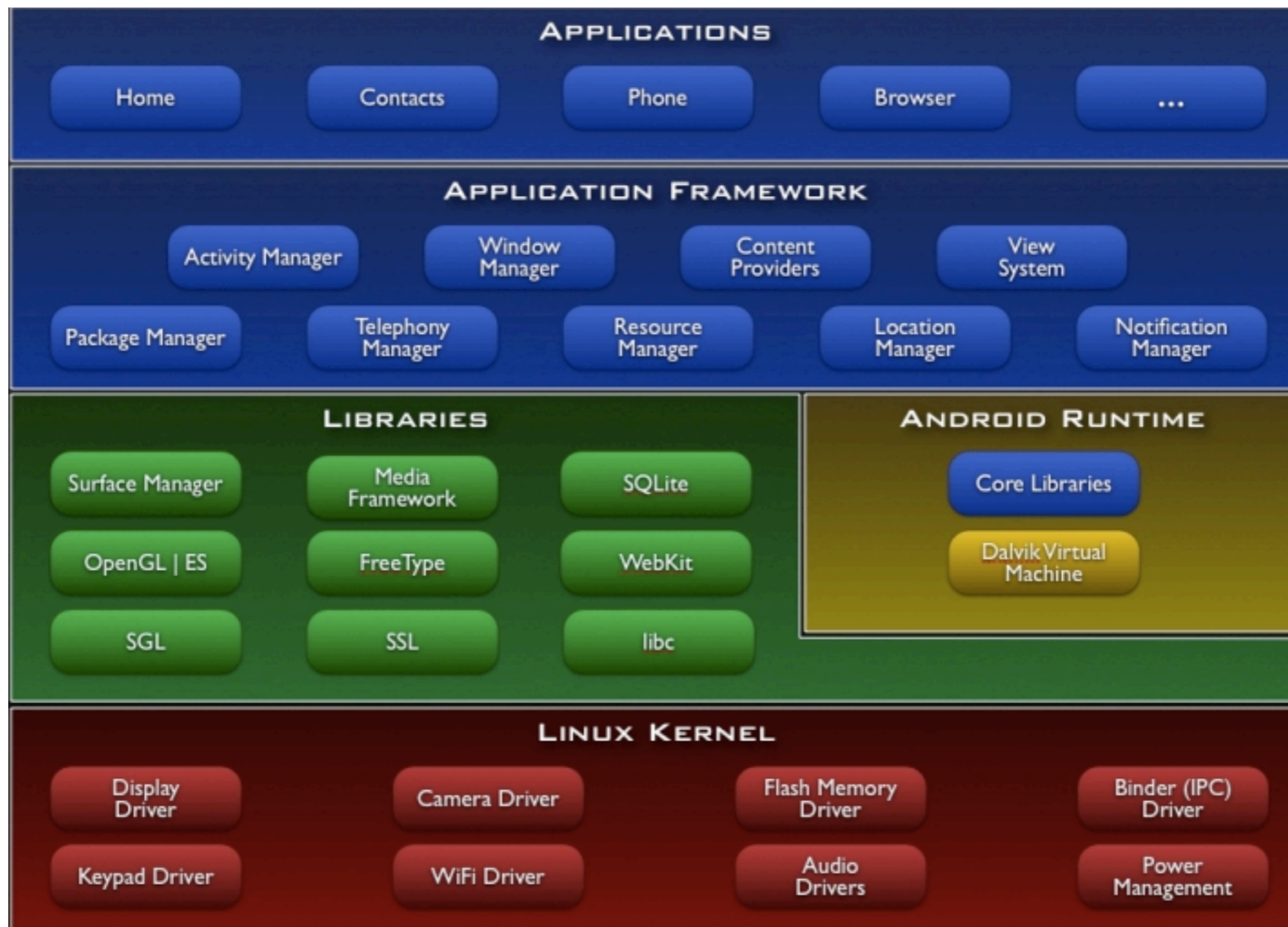


Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).



# Structure du système Android

7

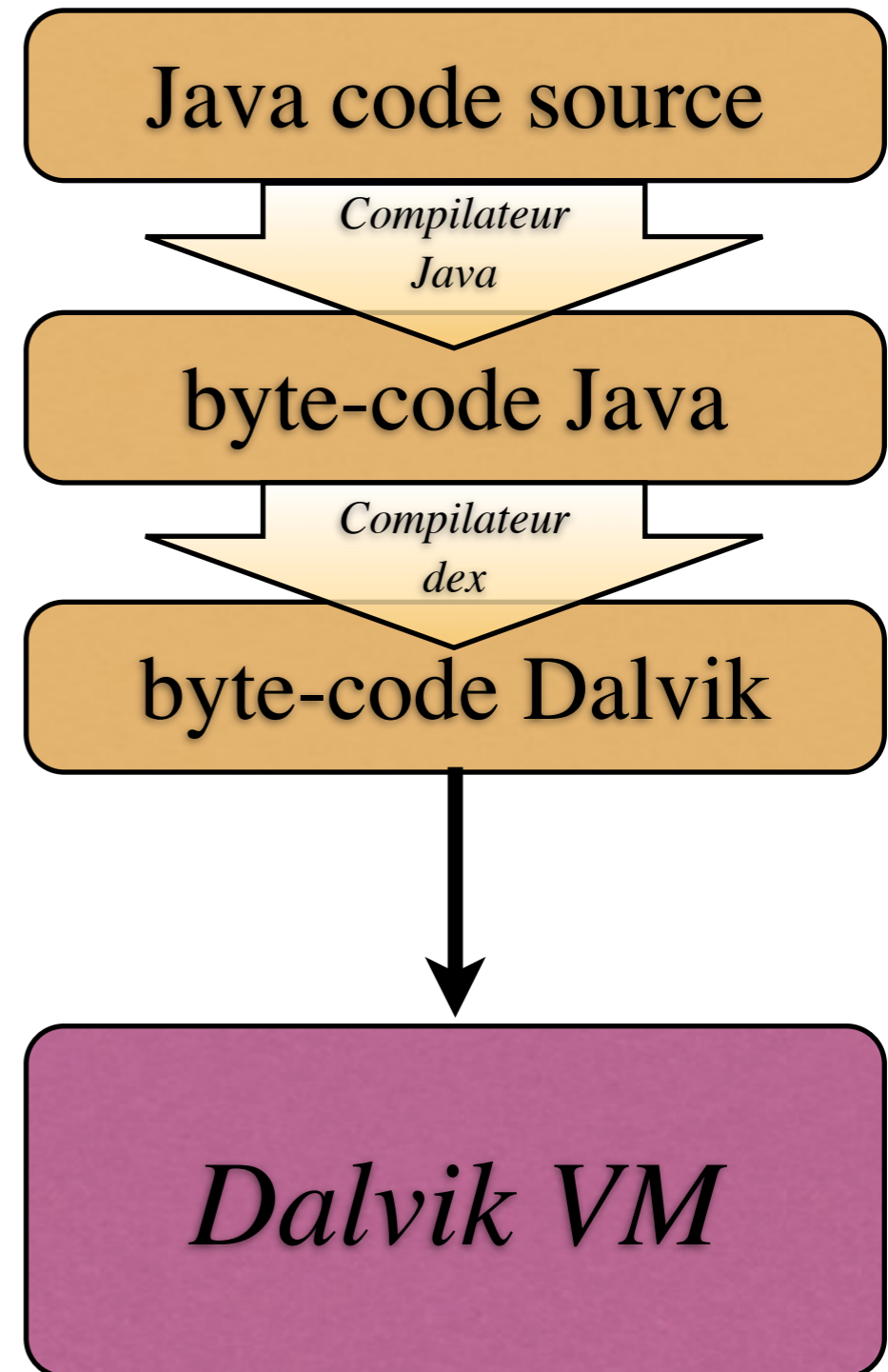


Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).

# Dalvik VM

8

- Il ne s'agit pas d'une JVM, elle exécute son propre byte-code
- Les fichiers .class font place à un fichier .dex
- architecture à registres par opposition à l'architecture à pile de la JVM
- Une VM par processus et donc par application
- Les VMs sont clonées et non créées (processus zygote)



# Applications Android

9

- Une application Android se concrétise sous la forme d'un fichier archive dont le nom possède l'extension `.apk`. Cette archive contient :
  - un fichier manifest décrivant le contenu de l'archive, la nature de l'application et ses points d'entrée
  - le code source contenu dans fichier dex (éventuellement pré-lié pour améliorer les performance : `.odex`)
  - des fichiers de ressources contenant des données manipulées par l'application : image, son, description d'interface graphique, fichiers de localisation, ...

# Applications Android

10

- Une application installé dans un terminal Android, dispose d'un identifiant unique (user id), qui isolera l'application des autres sauf si on accorde des permissions explicitement (sandboxing utilisateur)
- Une application s'exécute dans un processus unique, lancé dès qu'un des points d'entrée de l'application est sollicité
- Chaque application possède sa propre Dalvik VM
- Des exceptions peuvent-être consenties, quand plusieurs applications possèdent le même certificat (même signature => possibilité même uid ou même DVM)



# Le fichier Manifest Android

11

- Fichier xml regroupant des méta-données sur l'application
- Déclaration des composants et des Intent-Filters associés, des permissions, ...

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fr.cnam.nfa025"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <application android:label="@string/app_name" android:icon="@drawable/app_icon">
        <activity android:name="TP1"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name="fr.cnam.nfa025.Service"/>
    </application>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
</manifest>
```

# Applications Android

12

- Propose un modèle à composants, offrant plusieurs points d'entrées dans l'application. On distingue 4 composants :
  - Les activités (Activity)
  - Les services (Service)
  - Les fournisseurs de contenu (Content Provider)
  - Les traitements d'événements diffusés (Broadcast Receiver)

# Les Activités

13

- Composant central de l'interface utilisateur d'une application
- Le modèle de comportement est celui d'une page Web
- Une activité ne devrait pas avoir d'état propre sauvegardé en mémoire (stateless)
- Seul composant gérant l'interface graphique utilisateur
- Chaque nouvel écran présenté à l'utilisateur est porté par une activité différente (sauf onglets)
- Le système gère la navigation (lancement, touche back, ...), dialogue avec des tâche gérant l'état ou les données

# Les Activités

14

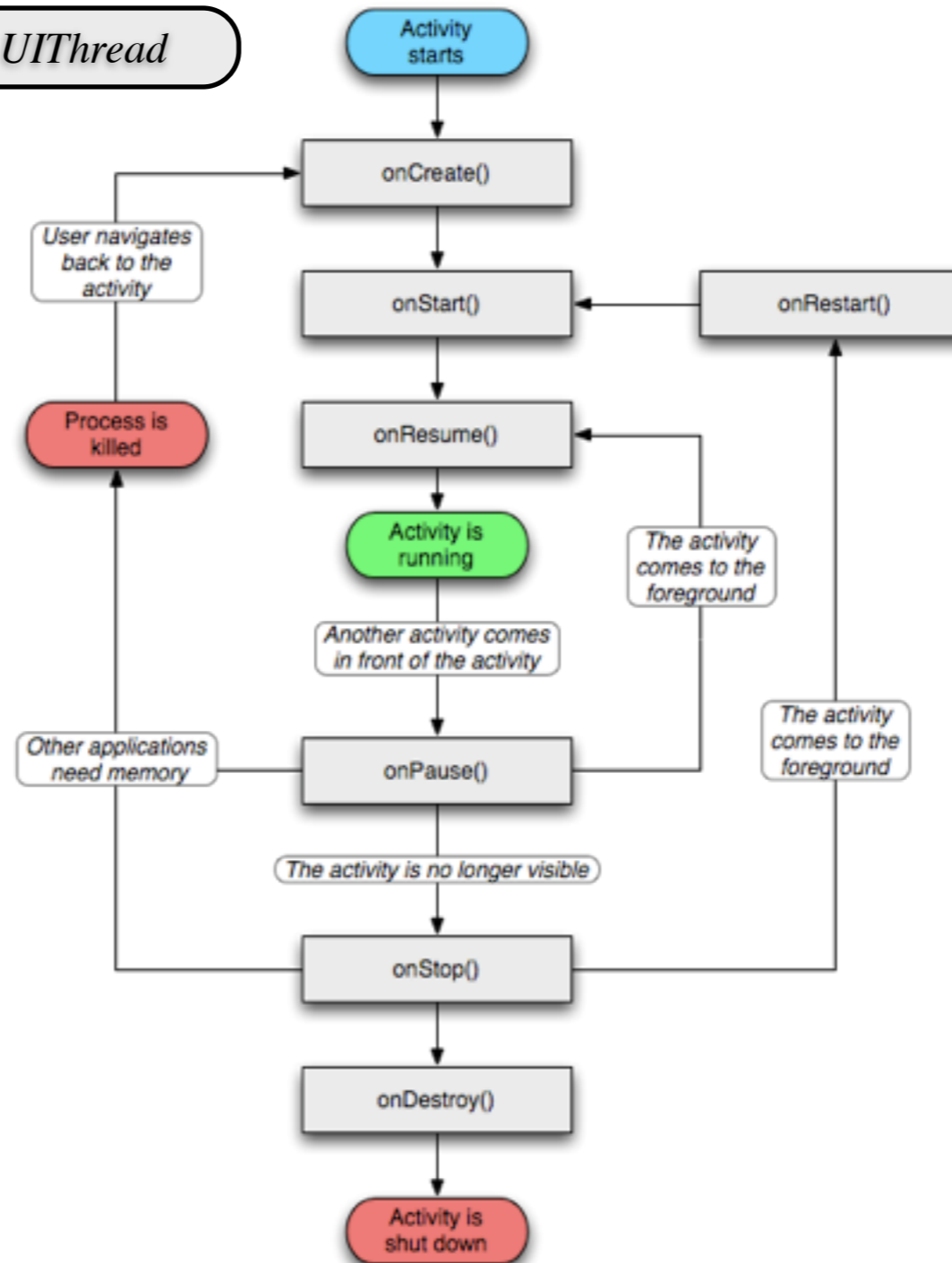
- Concrètement il s'agit d'une sous classe de la classe :  
**android.app.Activity**
- Instanciée par le runtime Android, l'utilisateur ne crée pas les instance d'une Activité
- Le cycle de vie de l'activité est géré par le runtime du système Android
- Comportement général de l'activité est obtenu par redéfinition d'un certain nombre de méthodes

# Cycle de vie d'une activité

15

`StartActivity(Intent)`

UIThread



Portions of this page are modifications based on work created and [shared by the Android Open Source Project](#) and used according to terms described in the [Creative Commons 2.5 Attribution License](#).

# Impact sur la définition des IHM

16

- Modèle de la thread unique de gestion de l'affichage :  
UIThread => toutes les opérations attachées à l'activité doivent-être brèves sous peine de bloquer la thread principale
- Un timer supervise une activité => arrêt forcé en cas de non réponse de l'activité à l'issu de la période de test
- Toute tâche complexe doit-être délégué à une thread spécifique ou un autre composant applicatif non graphique afin de préserver la réactivité du système



# Exemple (HelloWorld)

17

```
./AndroidManifest.xml  
./ant.properties  
./build.xml  
./gen/fr/cnam/nfa025/BuildConfig.java  
./gen/fr/cnam/nfa025/R.java  
./res/layout/main.xml  
./res/values/strings.xml  
./src/fr/cnam/nfa025/HelloWorld.java
```



# Exemple (HelloWorld)

18

```
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="Hello World, HelloWorld"
11    />
12 </LinearLayout>
13
14
```

# Exemple (HelloWorld)

19

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/
  res/android"
3     package="fr.cnam.nfa025"
4     android:versionCode="1"
5     android:versionName="1.0">
6     <application android:label="@string/app_name" >
7         <activity android:name="HelloWorld"
8             android:label="@string/app_name">
9             <intent-filter>
10                <action
11                    android:name="android.intent.action.MAIN" />
12                <category
13                    android:name="android.intent.category.LAUNCHER" />
14            </intent-filter>
15        </activity>
16    </application>
17 </manifest>
```

# Exemple (HelloWorld)

20

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">HelloWorld</string>
4 </resources>
5
```

# Exemple (HelloWorld)

21

```
1  /* AUTO-GENERATED FILE.  DO NOT MODIFY.
2  *
3  * This class was automatically generated by the
4  * aapt tool from the resource data it found.  It
5  * should not be modified by hand.
6  */
7
8  package fr.cnam.nfa025;
9
10 public final class R {
11     public static final class attr {
12     }
13     public static final class layout {
14         public static final int main=0x7f020000;
15     }
16     public static final class string {
17         public static final int app_name=0x7f030000;
18     }
19 }
20
```