

Android, introduction

Clément Raievsky
Avec la courtoisie de
Sébastien Jean



Département Informatique

2024

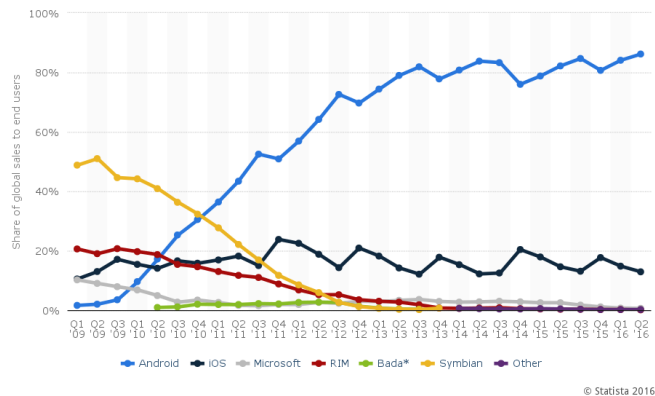
Android en bref

- ▶ Android est un **système d'exploitation**
 - ▶ Développé à partir de 2003 par *Android Inc.*, racheté par Google en 2005, apparu officiellement en version *beta* fin 2007
 - ▶ **open source** (basé sur **Linux**)
- ▶ Android équipe
 - ▶ des **smartphones et tablettes**, des netbooks
 - ▶ des **objets connectés** : TVs, montres, lunettes (*Google Glasses*)
 - ▶ des **voitures**



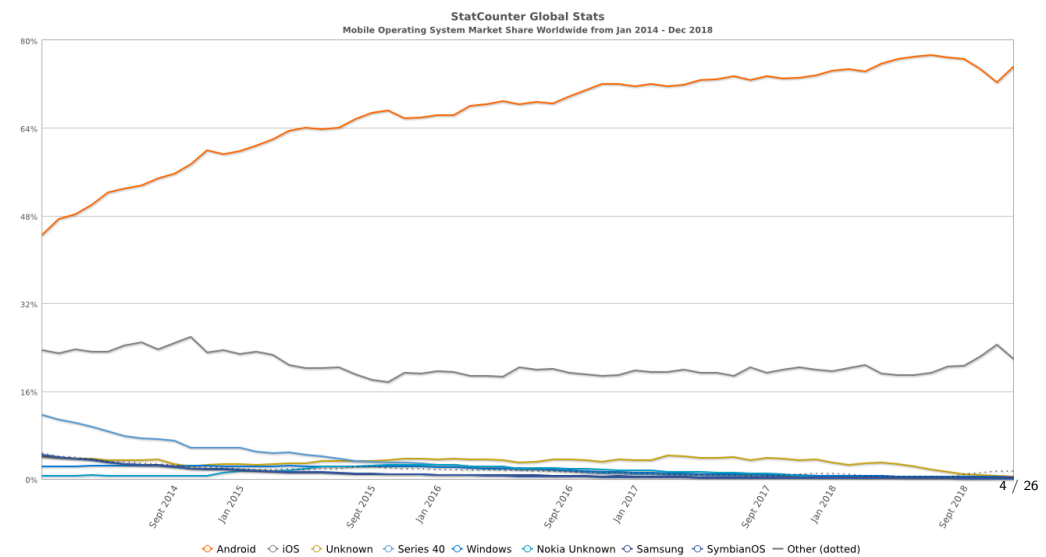
Android en bref

- ▶ Plus de 2 milliard d'appareils Android en activité en 2018
- ▶ Large domination du marché des systèmes d'exploitation mobiles 2009 - 2016



Android en bref

2014 - 2018



Android en bref

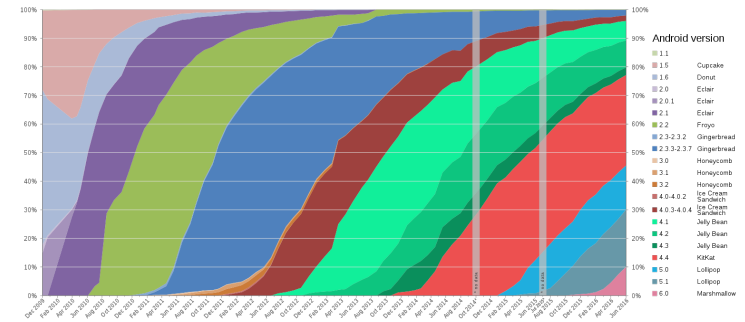
Une offre d'applications très riche

- ▶ **~2.5 million d'applications** début 2019
 - ▶ moins de 5% sont payantes
 - ▶ ~12% "Low Quality"
 - ▶ source : <http://www.appbrain.com/stats>
- ▶ En comparaison :
 - ▶ ~2 millions d'applications sur l'*AppStore* d'Apple
 - ▶ ~0.7 million d'applications sur le *Windows Store* de Microsoft

5 / 26

Versions d'Android

- ▶ Version courante : 12



- ▶ source : Wikimedia

6 / 26

Coexistence de versions d'Android

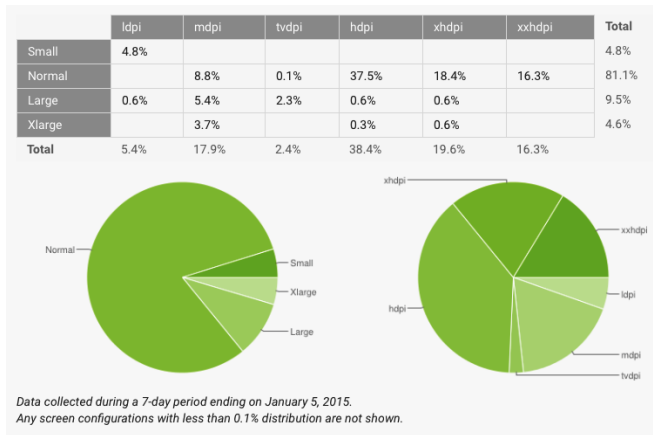
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

Coexistence d'équipements de taille et résolution variable

- ▶ **Tailles** d'écrans
 - ▶ **Small** → entre 2 et 3.5 pouces
 - ▶ **Normal** → entre 3 et 4.5 pouces
 - ▶ **Large** → entre 4 et 7 pouces
 - ▶ **XLarge** → supérieur à 7 pouces
- ▶ **Résolutions** d'écrans
 - ▶ **ldpi** → 120 dpi
 - ▶ **mdpi** → 160 dpi
 - ▶ **hdpi** → 240 dpi
 - ▶ **xhdpi** → 320 dpi
 - ▶ **xxhdpi** → 480 dpi
 - ▶ **xxxhdpi** → 640 dpi

8 / 26

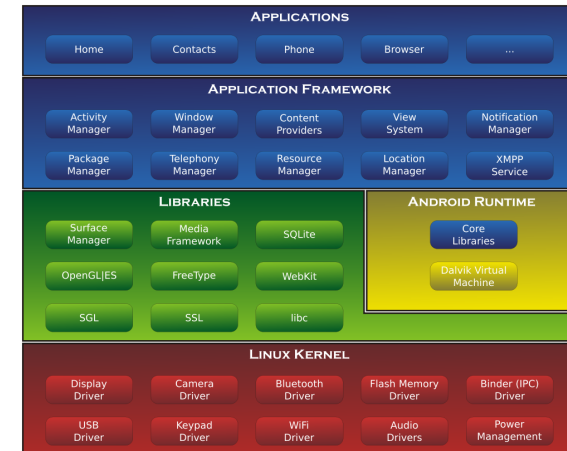
Equipements compatibles Android



► [Source](http://developer.android.com/about/dashboards) : <http://developer.android.com/about/dashboards>

9 / 26

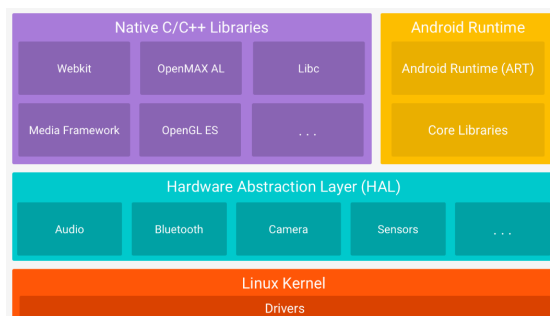
Architecture générale



► [Source](#) : Wikimedia

10 / 26

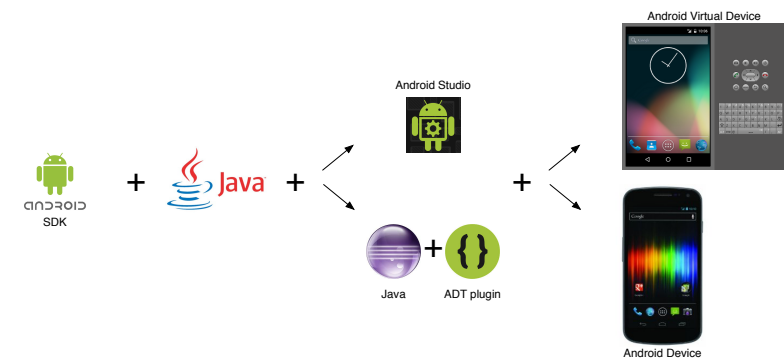
Architecture générale



Source : developer.android.com/guide/platform

11 / 26

Outils de développement



- **Android Studio** est depuis décembre 2014 l'**IDE officiel**
- **IntelliJ IDEA** est un sur-ensemble d'Android Studio

12 / 26

Android Vs Java

Applications Android = applications Java mais

- ▶ Machine virtuelle (JVM) spécifique
 - ▶ *Dalvik* jusqu'à Android 5.0
 - ▶ *Android Runtime (ART)* depuis 2014
- ▶ Librairie C spécifique : Bionic
- ▶ Fort *Sandboxing*

Contraintes fortes sur :

- ▶ les **ressources** (CPU, mémoire)
- ▶ la nécessité d'être réactif et d'économiser l'**énergie**
- ▶ la **connectivité intermittente ou limitée**

Framework évènementiel

Centré sur l'interaction utilisateur

13 / 26

Notion d'application

▶ **Application Android** = **unité logicielle**, sous forme d'archive **.apk**

▶ **Code métier**

▶ **Activités** ou **services**

▶ **Ressources**

▶ **directives de placement graphique** des différents écrans (*layout*)

▶ icônes, images, texte internationalisé, ...

▶ **Descripteur** (manifest)

▶ **contraintes de compatibilité**

▶ liens activités/layouts, ...

14 / 26

Activité Vs Service

▶ **Activité**

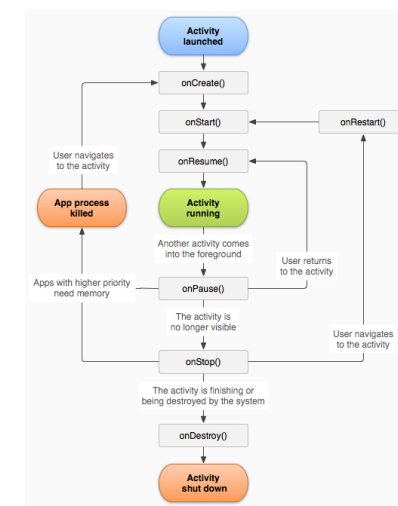
- ▶ **Tâche associée à l'interface utilisateur** (classe **Activity**)
- ▶ S'exécutant dans le **thread UI**
- ▶ Ecrite dans un **style évènementiel**
 - ▶ redéfinition des méthodes de *callback* liées au cycle de vie imposé

▶ **Service**

- ▶ **Tâche dissociée de l'interface utilisateur** (classe **Service**)
- ▶ S'exécutant en **background**
- ▶ Pouvant communiquer avec les activités et services

15 / 26

Cycle de vie des activités



16 / 26

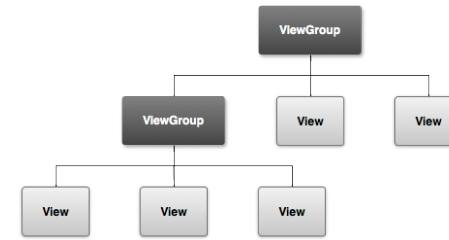
Hello World - Démo

À remarquer :

- ▶ Arborescence du projet
 - ▶ Code métier/de l'activité
 - ▶ Ressources, layouts notamment
 - ▶ Manifeste de l'application
- ▶ À quoi ressemble le manifest
- ▶ Code de l'activité principale
- ▶ Designer, vue xml et vue graphique
- ▶ Vue Android et vue filesystem du projet
- ▶ Spécialisation des ressources
 - ▶ textes
 - ▶ layouts

17 / 26

Layout : vues et conteneurs simples



View → objet visible

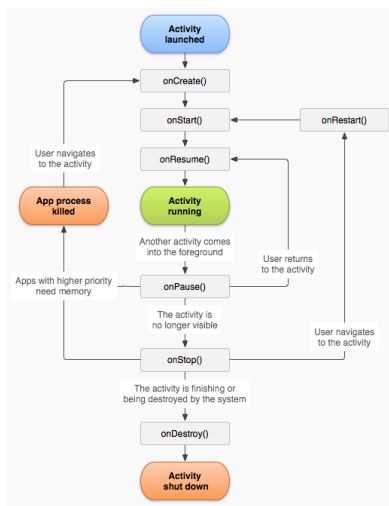
- ▶ **Widget** : bouton, champ texte, image, barre de progression, ...
- ▶ Générant ou non des **événements capturables**

ViewGroup → conteneur (groupe de vues)

- ▶ **Structure hiérarchique**
- ▶ **Stratégie de placement**
- ▶ **Par contraintes**

18 / 26

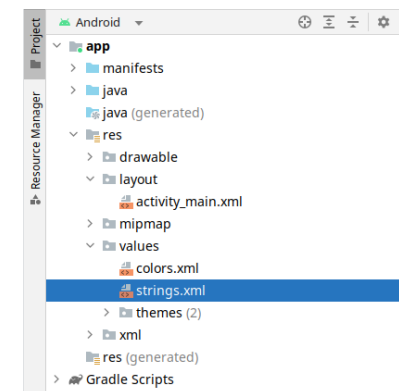
Rappel : Cycle de vie des activités



19 / 26

Ressources

Regroupées dans le répertoire **res** du projet



20 / 26

Accessibles dans les fichiers xml avec la syntaxe @

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_label"
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

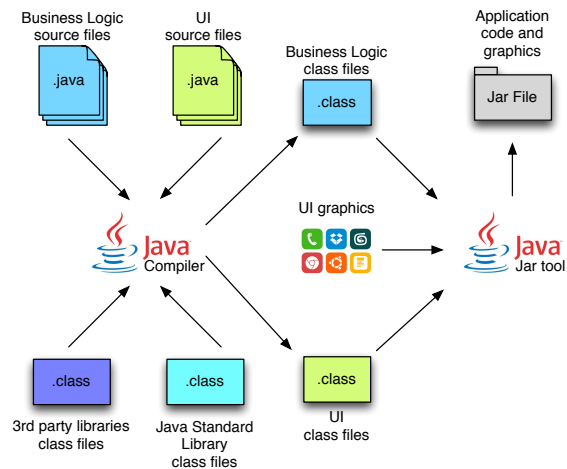
Accessibles en Java via la classe R

Générée avant la compilation

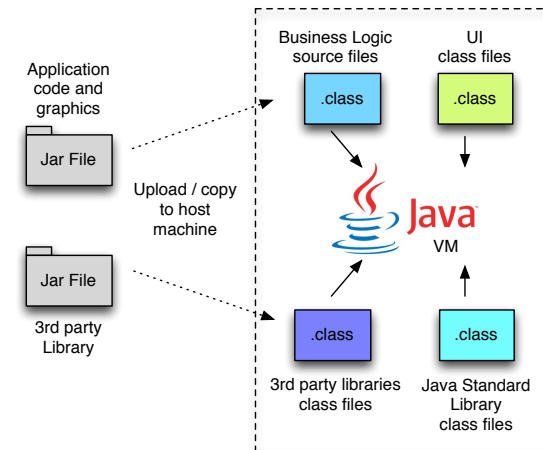
Par exemple dans l'activité de départ : R.layout.activity_main

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

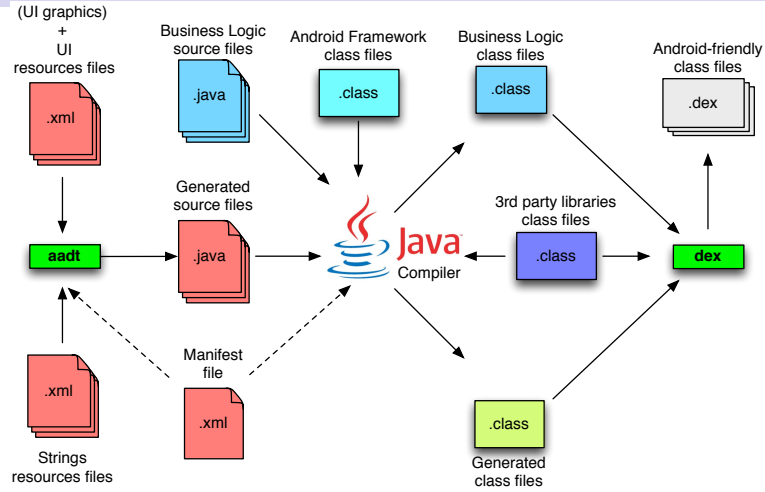
Rappels : compilation/packaging d'une application Java



Rappels : déploiement / exécution d'une application Java

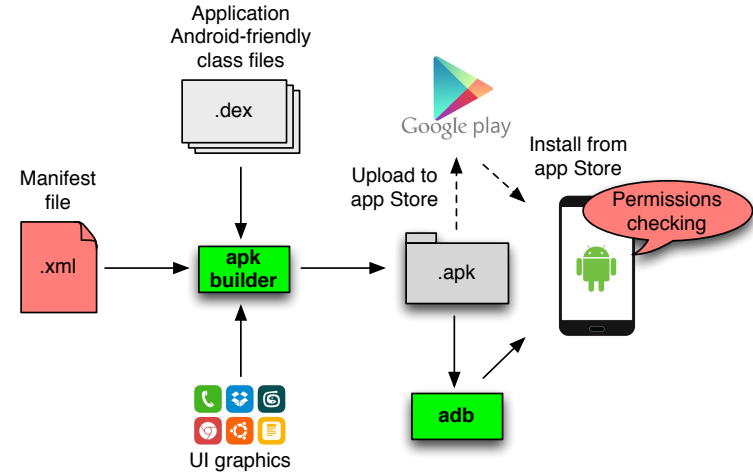


Android : compilation (simplifiée)



25 / 26

Android : déploiement / exécution (simplifiés)



26 / 26