



IUT de Paris - Rives de Seine
Université Paris Cité

R4.Real.11 – Développement pour applications mobiles

– Cours 1 –

Programmation mobile sous Android

Pr Chaouche A.-C.

ac.chaouche@gmail.com

Prérequis

- Système d'exploitation
- Programmation



Objectifs du cours

- Connaître les systèmes d'exploitation mobiles
- Assimiler les couches de l'architecture d'Android
- Comprendre le cycle de développement d'une application
- Assimiler la structure d'un projet Android

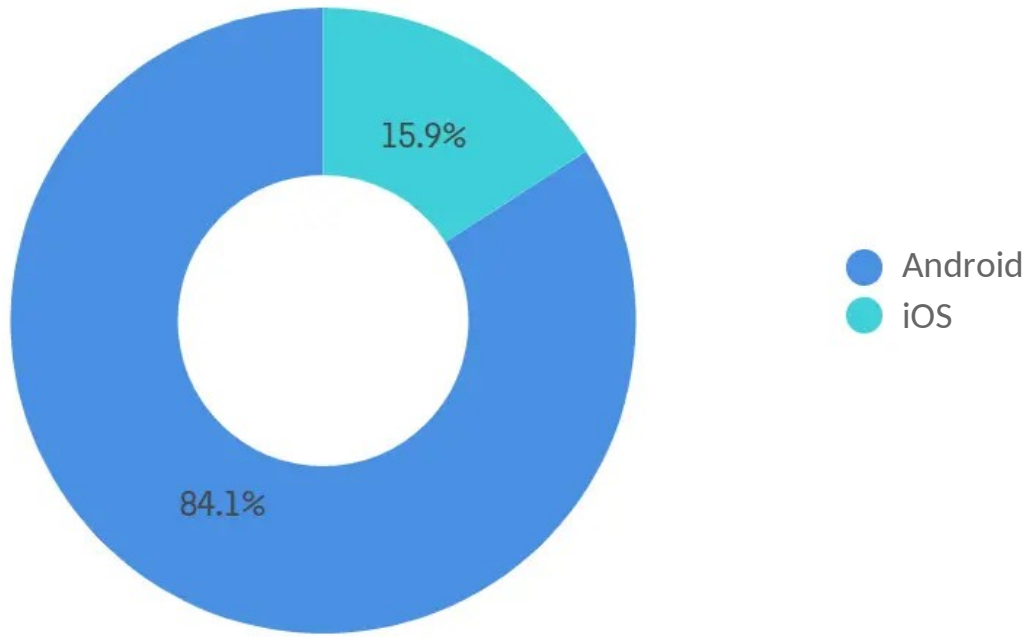
Développement Mobile : Natif vs. Web

	Application native	Application Web
Portabilité	Développement spécifique à chaque plateforme	Navigateur Web
Développement	Nécessité du SDK Maîtrise d'un langage	Langage Web (HTML, JS, CSS, PHP, ...)
Mise à jour	Re-téléchargement à travers le App Store	Fichiers au niveau du serveur Web
Disponibilité	Online et offline	Online (connexion internet nécessaire)
Fonctionnalités	Toutes les fonctionnalités du mobile (GPS, voix, notifications, capteurs, ...)	Limitées aux possibilités du navigateur

OS mobiles

	Plateforme/OS	Langage	IDE	SDK	
	Symbian OS 9.5	Code natif C++ Java	Eclipse	Carbide.c++ Java ME	Abandonné en 2014
	BlackBerry 10 (RIM)	Java C/C++	Eclipse	BlackBerry 10 Native SDK	Abandonné en 2022
	Windows Phone 10	C++ C# VB .Net	Visual studio	Windows Phone SDK .NET Framework 4.6 SDK	Abandonné en 2019
	iOS 26	Swift objectif-C	Xcode	IPhone SDK	Sortie en sept 2025
	Android 16	Java Kotlin Code natif C++	Android studio	Android SDK + JDK	Sortie en juin 2025

Part de marché des OS mobiles



Source : Mordor Intelligence (2024)



Android OS – OHA

Startup créée en 2003 et rachetée **par Google en 2005**

Open Alliance Handset (OHA) en 2007 :

- **Regroupement de + de 50 entreprises** : fabricants de matériel, opérateurs mobiles, développeurs d'applications, ...
- **Objectif** : développer des normes ouvertes pour des applications de téléphonie mobile
- **1^{er} mobile sous Android, en 2008 :**
T-Mobile G1

Operator	Handset Makers	Software Companies	Commercialization Companies	Semiconductor Companies
        	         	           	     	            

Matériel supportant Android

- Tablette et Smartphone
- Netbook et appareil photo
- Télévision et Frigo
- Tableau de bord (auto)
- Android wear
- Google glass



Android OS – Points forts



CONSTRUCTEURS

- Système linux
- Projet open source
- Licence Apache (coût nul)



DÉVELOPPEURS

- Langage Java
- Kit de développement (SDK) gratuit
- Android Market



UTILISATEURS

- Système fonctionnel, intuitif et évolutif
- Multi-tâches

Versions d'Android OS

Noms de code : Desserts



Cupcake
Android 1.5



Donut
Android 1.6



Eclair
Android 2.0/2.1



Froyo
Android 2.2.x



Gingerbread
Android 2.3.x



Honeycomb
Android 3.x



Ice Cream Sandwich
Android 4.0.x



Jelly Bean
Android 4.1.x



KitKat
Android 4.4.x



Lollipop
Android 5.0



Marshmallow
android 6.0



Nougat
android 7.0



Oreo
android 8.0



Pie
android 9.0



android 10



Red Velvet Cake
android 11



Snow cone
android 12



Tiramitsu
android 13



Upside Down Cake
android 14



Vanilla Ice Cream
android 15



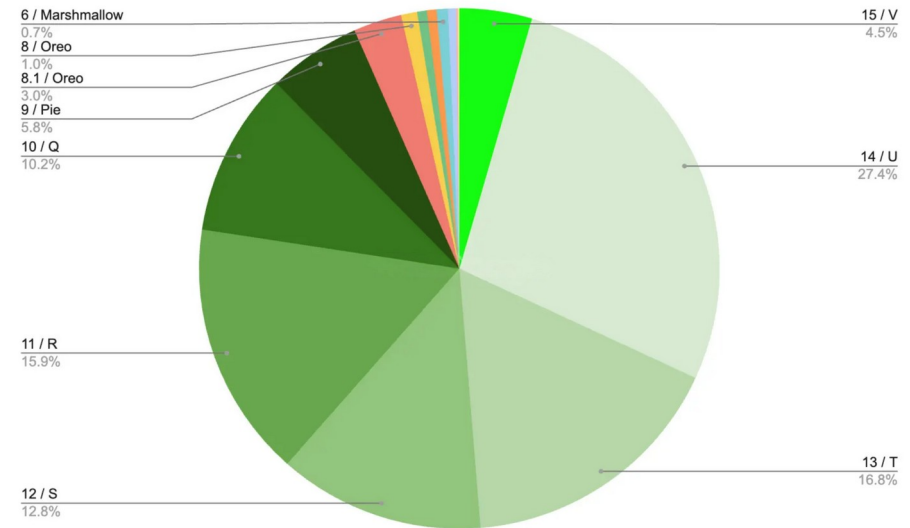
Baklava
Android 16

(Sortie en juin 2025)

Version actuelle :

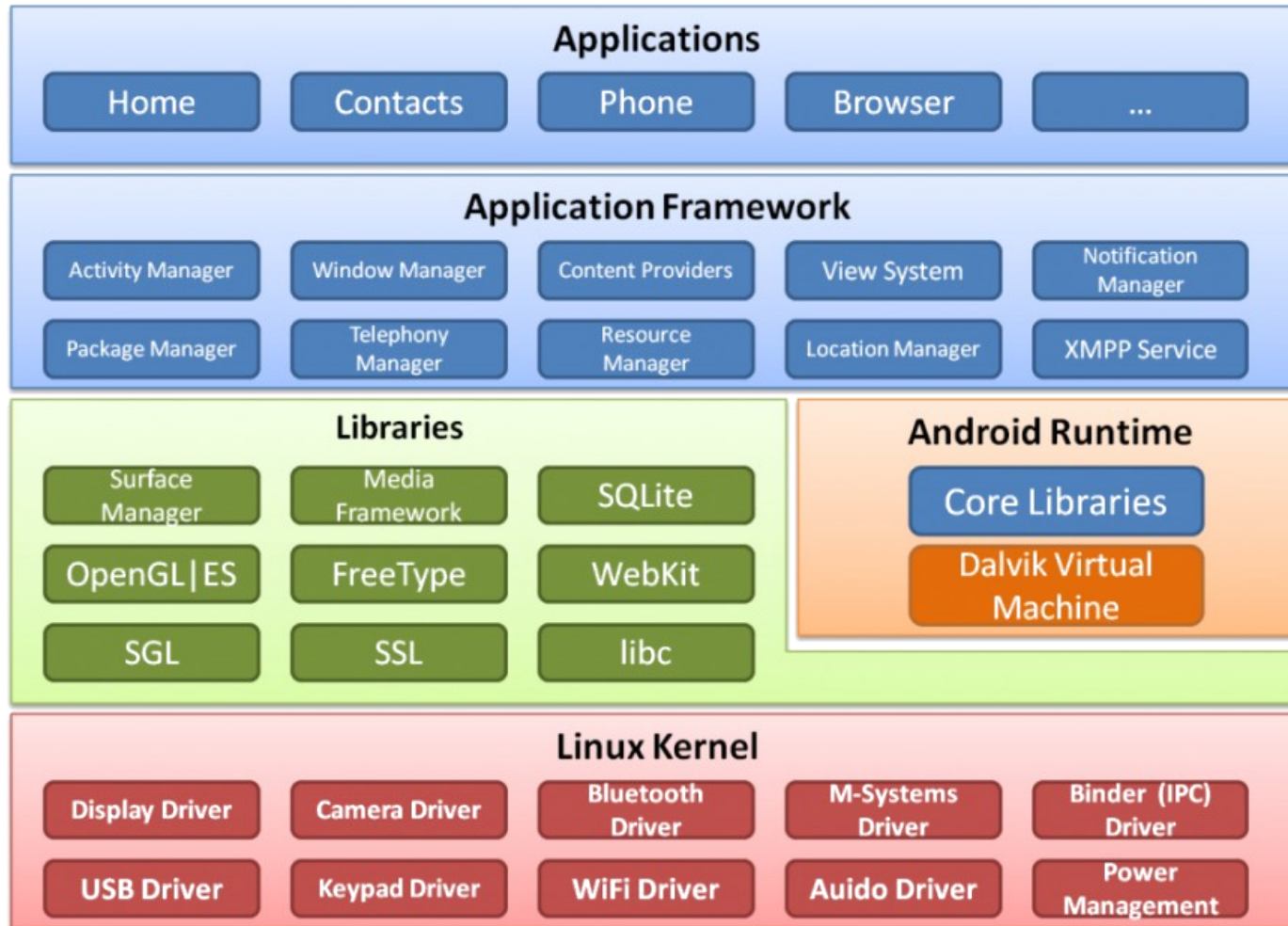
Versions d'Android OS – Distribution

Version	Nom	API	Distribution
16	Baklava	36	--
15	Vanilla Ice Cream	35	4.5 %
14	Upside Down Cake	34	27.4%
13	Tiramisu	33	16.8%
12L	Snow Cone	32	5.2 %
12		31	7.6%
11	Red Velvet Cake	30	15.9%
10	Quince Tart	29	10.2%
9.0	Pie	28	5.8%
8.1	Oreo	27	3.0%
8.0		26	1.0%
7.1	Nougat	25	0.6 %
7.0		24	0.6 %
6.0	Marshmallow	23	0.7 %
5.1	Lollipop	22	0.5 %
5.0		21	0.1 %
4.4	KitKat	19	0.1 %



Source : Google Play, Avril 2025

Architecture d'Android (1/2)



Architecture d'Android (2/2)

Noyau Linux Kernel

- fournit le pilotes matériels, la gestion de processus, la mémoire, le réseau et la gestion de l'alimentation

Bibliothèques

- Libc, SQLite, WebKit, OpenGL, ...

Moteur d'exécution Android

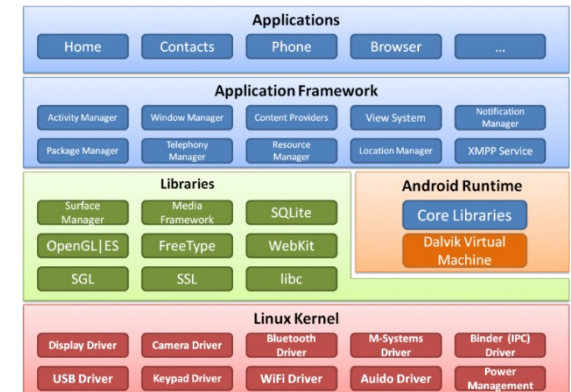
- offre les bases de la plateforme applicative
- utilise la machine virtuelle - Java Dalvik

Plateforme applicative

- fournit les classes utiles et des abstractions matérielles

Couche applicative

- IHM (interface graphiques) décrites en XML
- modèle de données et contrôleurs applicatifs écrits en java



Machine virtuelle « Dalvik »

La VM Dalvik offre l'avantage de toute machine virtuelle

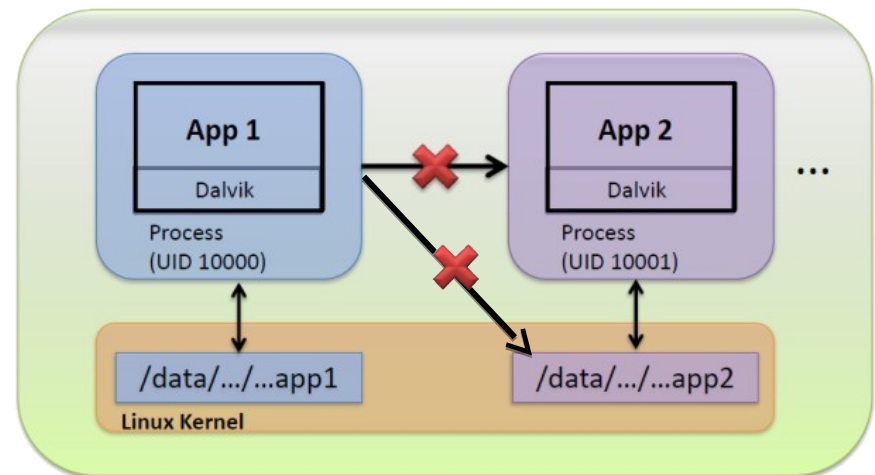
- Couche d'abstraction entre le développeur d'applications et des implémentations matérielles particulières

La VM Dalvik n'est pas une VM (Virtual Machine) Java

- Tient compte des contraintes de CPU et mémoire
- Exécute des fichiers .dex (Dalvik Executable) optimisés

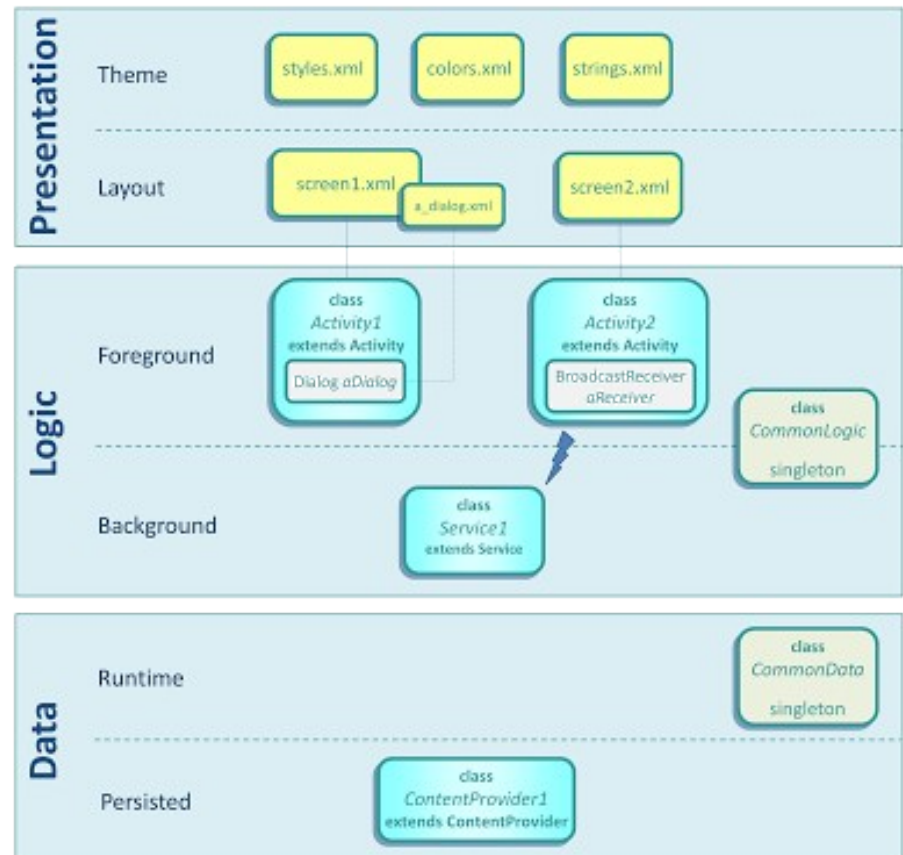
La VM crée une instance Dalvik pour chaque processus (lourd)

- Les applications sont totalement indépendantes ("sandbox")
 - Evite un plantage généralisé !
- Espaces protégés (mémoire, disque)
 - Augmente la sécurité

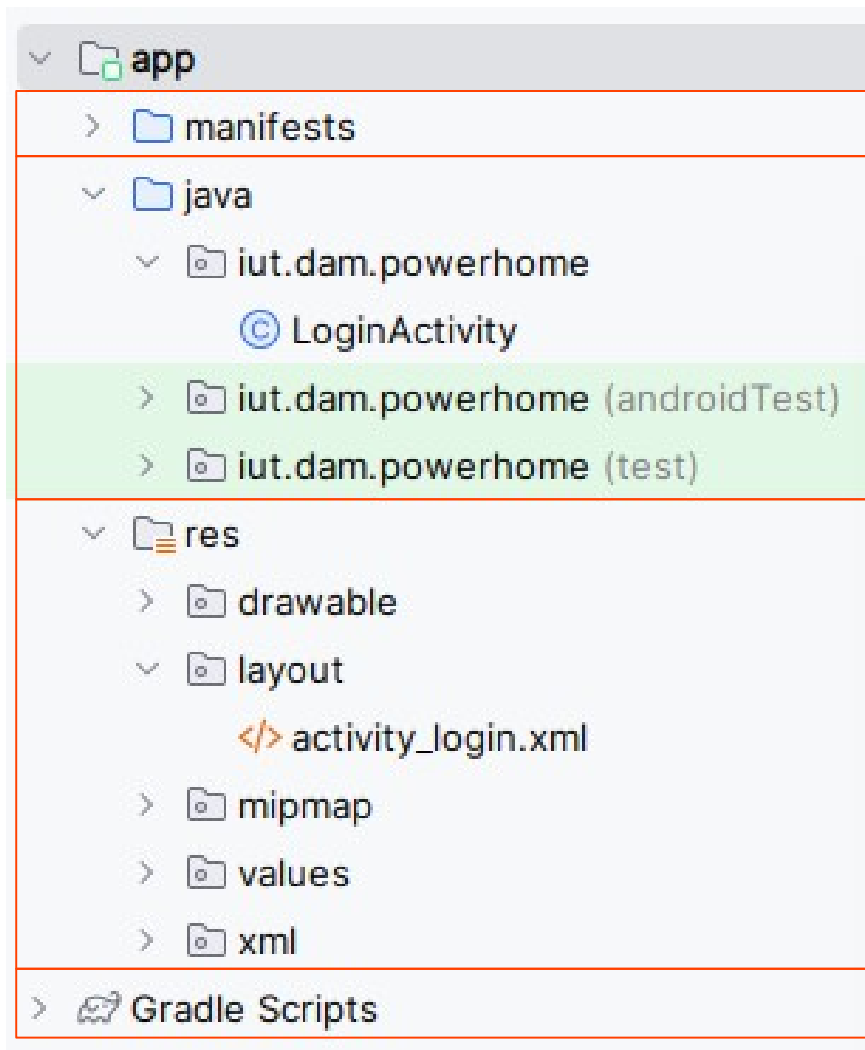


Structure d'un projet (1/2)

- Un projet Android possède une structure semblable à **MVC**, en séparant la vue du contrôle
- Cette structure permet d'organiser le projet en 3 couches :
 - **Présentation**
 - **Logique**
 - **Données**



Structure d'un projet (2/2)



Manifest

Fichiers java

Ressources organisées
(texte, ...)

Gradle

Processus de développement (1/2)

Compilation du code source

- Écrit en Java et compilé en Dalvik bytecode (.dex)

Archivage (Packaging)

- Archive .apk

Signature

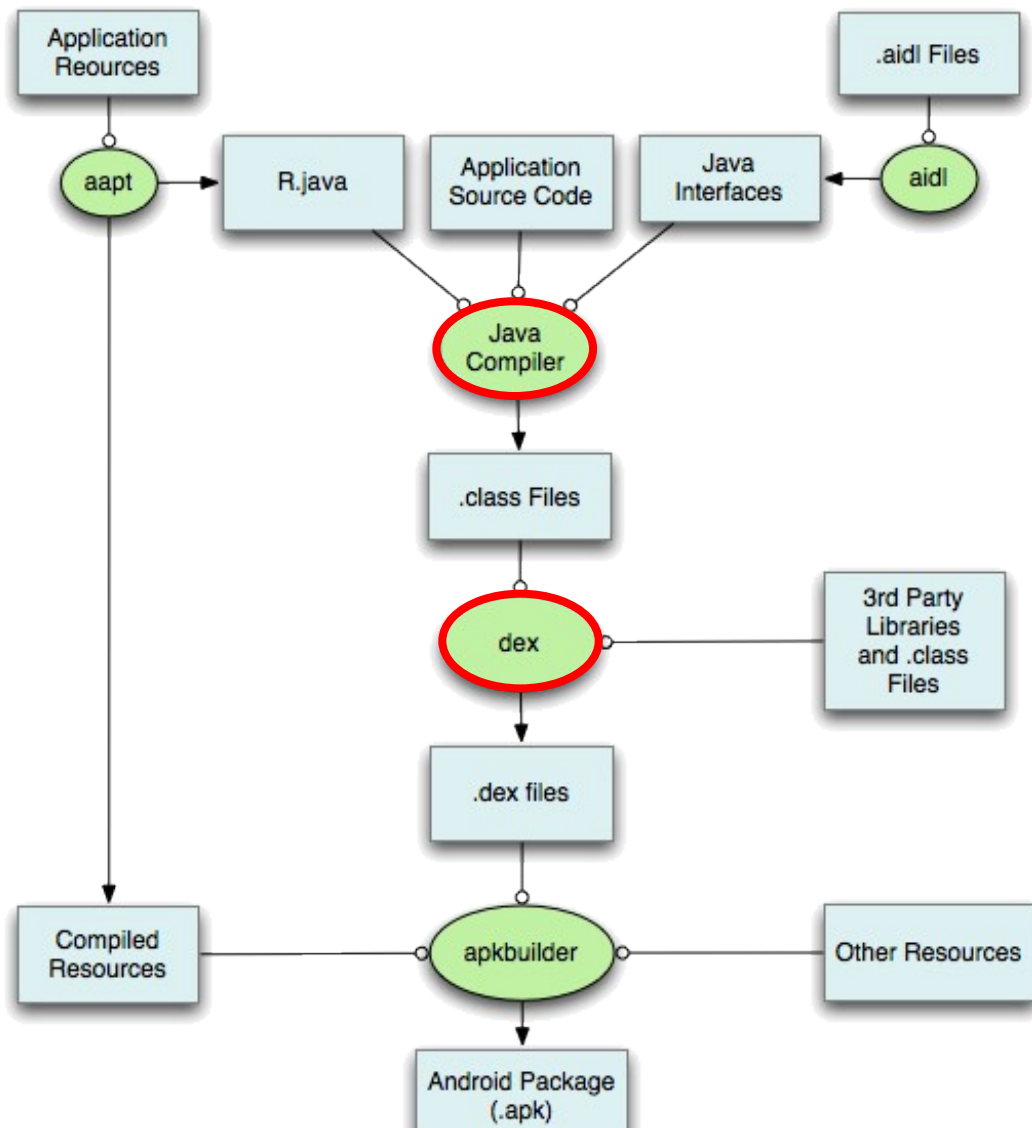
- Les applications doivent être signées pour être déployées (pas nécessaire en mode développement)

Exécution

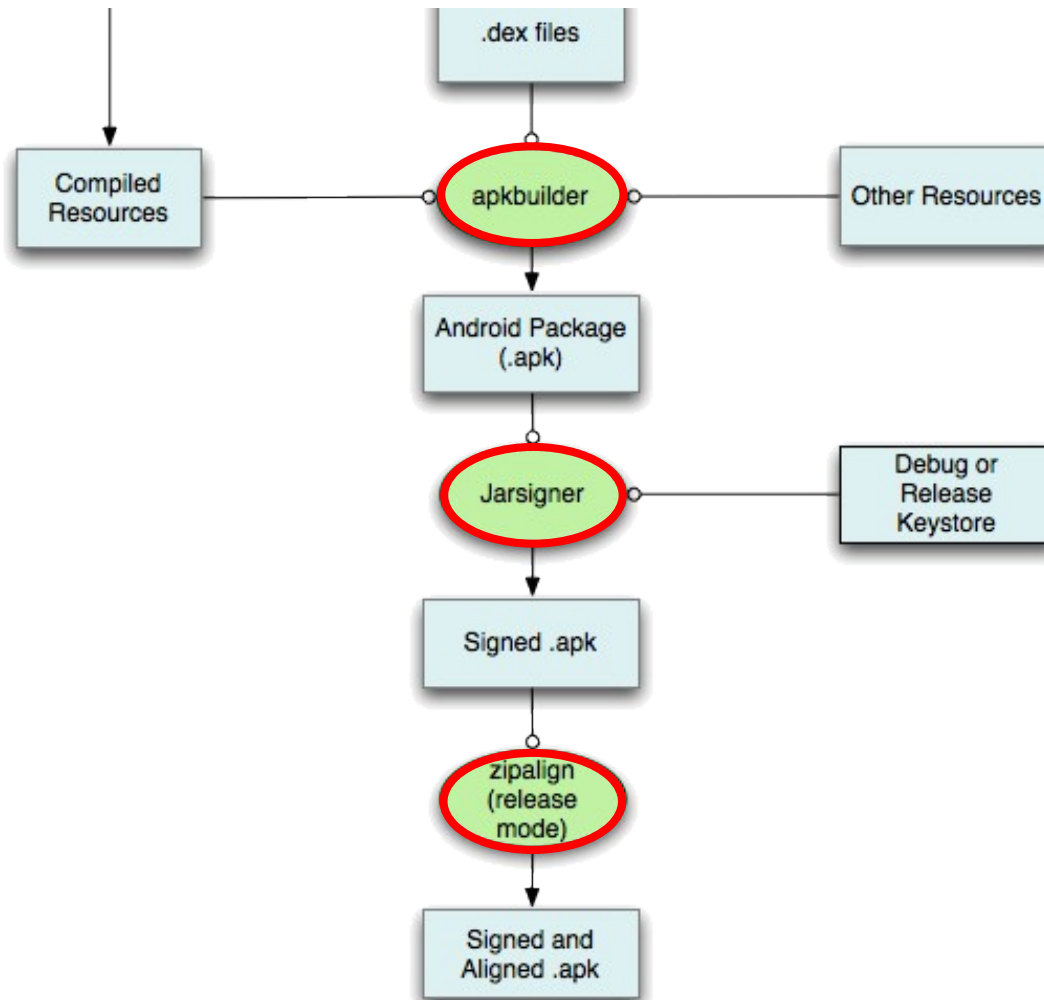
- Sur un support mobile physique (smartphone par exemple)
- Sur un émulateur



Processus de développement (2/2)



Processus de développement (2/2)



Types d'applications sous Android

Applications de premier plan

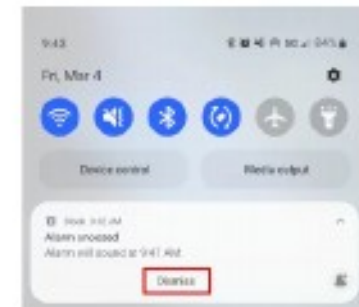
- Utilisable quand elle est visible et mis en pause lorsqu'elle ne l'est pas
- **Exemples** : Jeux, Appli.

Applications d'arrière-plan

- Interaction limitée en dehors de sa configuration
- **Exemples** : Alarme, Filtrage d'appels

Widgets

- Composant visuel interactif que les utilisateurs peuvent ajouter à leur écran d'accueil
- **Exemples** : Horloge, prévisions météo



Exécution sur l'émulateur intégré

AVD (Android Virtual Device)

- AVD Manager
- Lent au démarrage et à l'usage
- Gourmant en ressources
- Emulation vs. Simulation

Fonctionnalités non-disponibles

- Appels téléphoniques réels
- Capteurs en général
- Connexions USB
- Évolution de la charge de la batterie
- ...



Exécution sur un appareil mobile (1/4)

Fonctionnalités de base

- Capteurs et composants intégrés (Vibreux, batterie, bluetooth, ...)

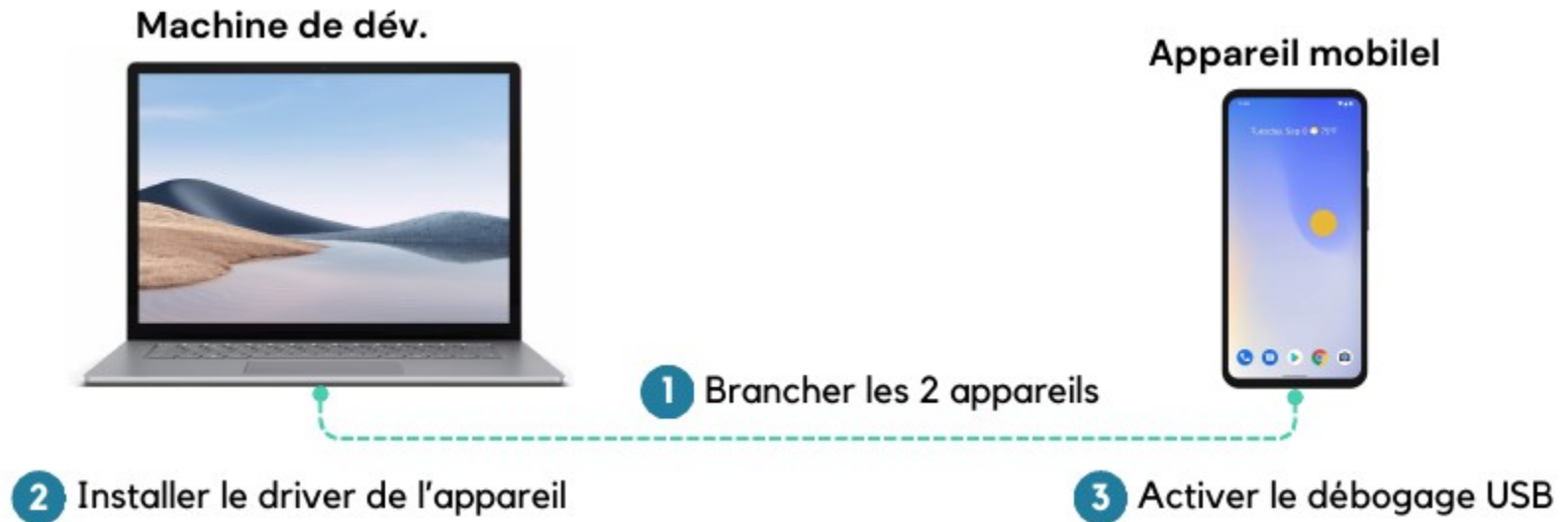
Caractéristiques d'un appareil mobile

- Un processeur à puissance limitée
- Une RAM et des capacités de stockage limitées
- Un écran avec une résolution fixe
- Des coûts élevés de transfert de données
- Une batterie à autonomie limitée



Exécution sur un appareil mobile (2/4)

Configuration de connexion

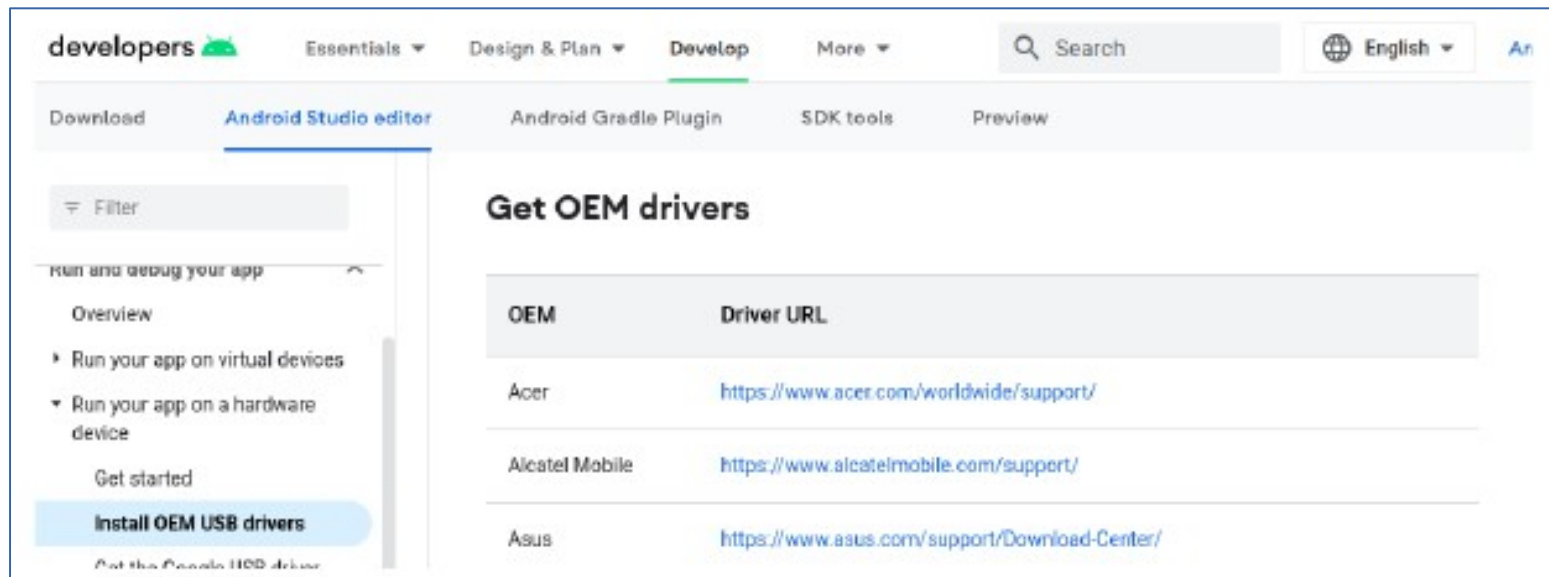


Exécution sur un appareil mobile (3/4)

2. Installer le driver de l'appareil

Lien : <https://developer.android.com/tools/extras/oem-usb.html#Drivers>

- Choisir le constructeur de l'appareil
- Suivre les instructions pour installer le driver approprié

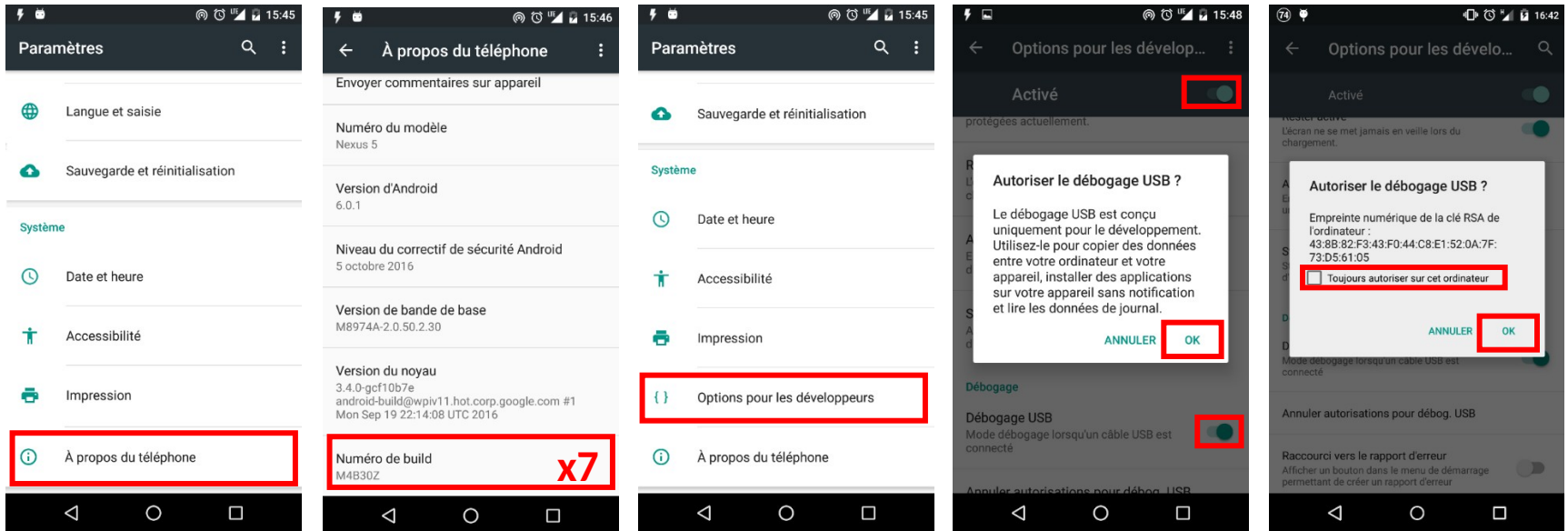


The screenshot shows the 'developers' website with the 'Develop' tab selected. The 'Android Studio editor' link is highlighted in the navigation bar. On the left sidebar, under 'Run and debug your app', the 'Install OEM USB drivers' option is highlighted. The main content area is titled 'Get OEM drivers' and contains a table with OEMs and their driver URLs.

OEM	Driver URL
Acer	https://www.acer.com/worldwide/support/
Alcatel Mobile	https://www.alcatelmobile.com/support/
Asus	https://www.asus.com/support/Download-Center/

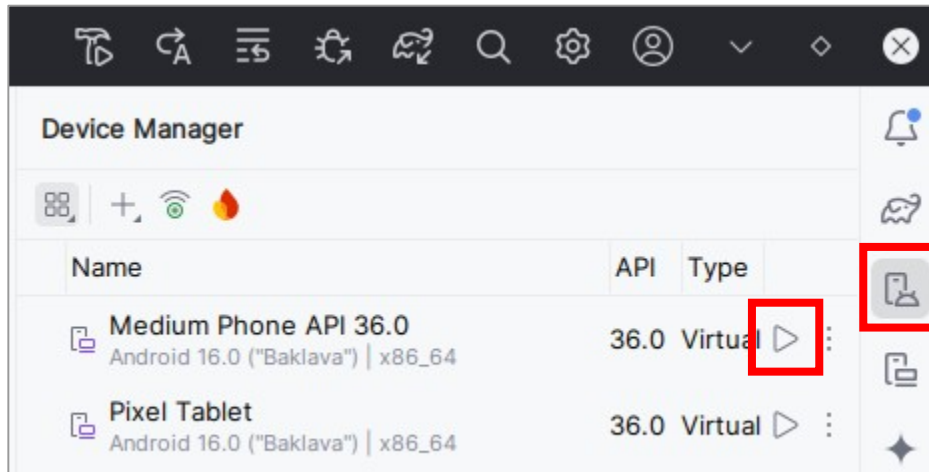
Exécution sur un appareil mobile (4/4)

3. Activer le débogage USB



Exécution de l'application

1. Lancer l'emulateur (ou brancher un appareil physique)



2. Sélectionner l'appareil et lancer l'exécution

