

APPLICATIONS MOBILES



OBJECTIFS

- Introduire le développement des applications mobiles
- Cas d' étude : Android
 - Découvrir la programmation sous Android
- **Connaissances préalables recommandées** : Avoir de bonnes notions sur *Java*, *JavaScript* et *XML*.

PLAN

- Introduction
- La plateforme Android

INTRODUCTION

- Appareils mobiles
 - De plus en plus petits et légers
 - De plus en plus intelligents
 - De plus en plus de fonctionnalités
 - Défi : tout intégrer au sein d' une même plateforme
- → Plusieurs appareils et OS existent !



INTRODUCTION

○ Systèmes d' exploitation Mobile

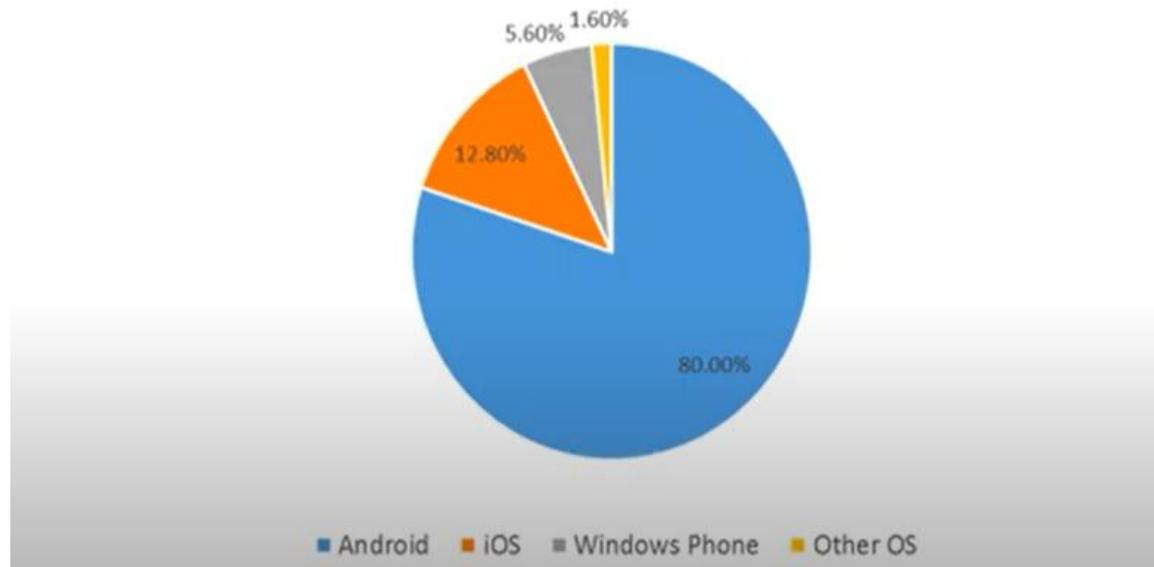


symbian



OS	Télé/Société	Langage	Remarques
iOS	iPhone	C / Objective C ou HTML	Plutôt populaire
Android	Google	Java	Open source, gratuit
Windows Mobile	Windows Phone	C# / C++	
Blackberry OS	Blackberry/RIM	Java	
Symbian	Nokia	C++ / J2ME / Python / Ruby	Bientôt abandonné

Les Smartphones vendus selon leur OS (2018)



INTRODUCTION

- Prendre en considération, pendant le développement d' applications mobiles:
 - Ecran très petit
 - Pas de clavier/Souris (du moins, un clavier virtuel)
 - Les appareils fonctionnent sur batterie (attention à la consommation!)
- Dans le cadre de ce cours :
 - Nous nous intéressons aux applications mobiles sous





Il y a deux méthodes pour développer une application mobile:

- Développement mobile natif
- Développement mobile cross plateformes

Développement mobile natif

- **Une application native** est une application développée spécifiquement pour un système d'exploitation. De ce fait, une application mobile développée pour fonctionner sur iOS ne fonctionnera pas sur Android et vice-versa.

Ceci est dû au fait qu'il existe différents langages et technologies pour chaque système d'exploitation.

Introduction

Développement mobile natif

Entreprise	Appareils	Système d'exploitation	Plateforme de téléchargement	Langage de programmation	Environnement de développement
Apple	Iphone/Ipad	iOS	App Store	Objective C/ Swift	Xcode
Google	Les téléphones portables et tablettes des marques: Samsung, Oppo, LG, Huawaei,...etc	Android	Google play/ Amazon	Java/kotlin	Android Studio
Microsof	Windows phone	Windows phone (Windows 10 mobile)	Windows store	C#, C, C++	Visual Studio

Développement mobile cross plateformes

Une **application mobile Cross plateforme** est une application développée une seule fois - **en un seul code** - tout en étant compatible avec les principaux systèmes d'exploitation des smartphones: iOS, Android, Windows Phone. Nous citons trois plateformes:

- Plateforme de développement Mobile Hybride: IONIC et CORDOVA
- Plateforme de développement Mobile Cross Platform REACT NATIVE
- Plateforme de développement Mobile Cross Platform FLUTTER

Développement mobile natif: les avantages

Les avantages de développer les applications en natif :

 ***Meilleure rapidité***, fiabilité et dotée d'une meilleure réactivité ainsi qu'une résolution supérieure ce qui assure une meilleure expérience utilisateur.

 ***Un accès plus facile aux*** fonctionnalités natives avancées du téléphone comme la caméra ,le GPS, le répertoire etc....

Développement mobile cross plateformes

Un des principaux avantages d'une application cross plateformes:



Plus facile et plus rapide à développer qu'une application native.



La maintenance de l'application sera également plus facile puisqu'il n'y a qu'une seule version à revoir pour plusieurs plateformes.



Cependant, toutes ces facilités ont un prix : les performances de l'application sont moins bonnes et moins stables puisque le système est moins adapté à chaque plateforme.



Présentation d'Android



ANDROID

- Né en 2004 (startup)
 - Racheté par Google en 2005
 - Publié en 2007, version 1.5
 - ... de nombreuses versions depuis, on est à la 8.0.
-
- Android = Système d'exploitation pour smartphones, PDA et tablettes
 - Open source
 - Gratuit
 - Basé sur un noyau Linux 2.6
 - Conçu pour intégrer les applications Google : Gmail, Google Maps, Google Agenda, YouTube et géolocalisation

ANDROIDVERSIONS

<https://socialcompare.com/fr/comparison/android-versions-comparison>

- Les différentes versions de Android ont des noms de dessert ou sucreries (dans un ordre alphabétique)

Nom	Version	Date
Android	1.0	09/2008
Petit Four	1.1	02/2009
Cupcake	1.5	04/2009
Donut	1.6	09/2009
Gingerbread	2.3	12/2010
Honeycomb	3.0	02/2011
Ice Cream Sandwich	4.0.1	10/2011
Jelly Bean	4.1	07/2012
KitKat	4.4	10/2013
Lollipop	5.0	10/2014
Marshmallow	6.0	05/2015
Nougat	7.0	09/2016

Les avantages d'Android

-  **Open source:** Le contrat de licence pour Android respecte les principes de l'open source.
-  **Gratuit (ou presque):** Android est gratuit, autant pour vous que pour les constructeurs. En revanche, pour poster vos applications sur le Play Store, il vous en coûtera la modique somme de 25\$.
-  **Facile à vendre:** Le Play Store (anciennement Android Market) est une plateforme très visitée pour le partage des applications.
-  **Flexible:** Le système est extrêmement portable, il s'adapte avec différents dispositifs (smartphones, tablettes, TV, four, machines à laver, ...).



Les avantages d'Android

-  **Complémentaire:** combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.
-  **Facile à développer:** Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles à y accéder.



Une **API**, ou « **interface de programmation** » en français, est un ensemble de règles à suivre pour pouvoir dialoguer avec d'autres applications. Par exemple, je peux demander à Google Maps de m'afficher la carte de Mohammedia !

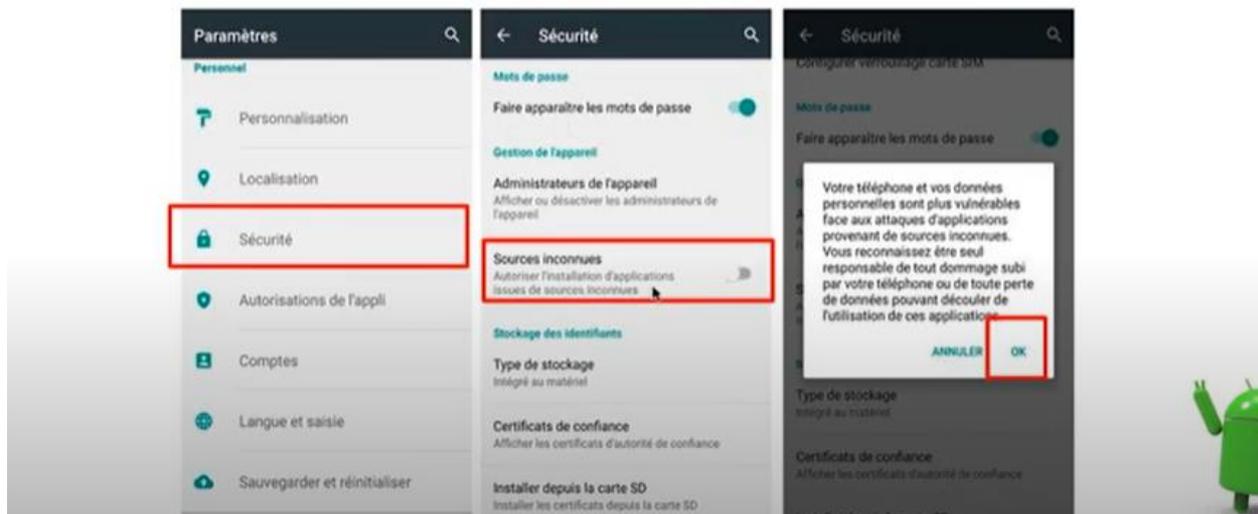
Télécharger une application

Pour télécharger une application Android, vous pouvez utiliser l'une des méthodes suivante:

- 🎯 **Google Play store** (ancien Android Market): Système standard de téléchargement/distribution d'applications. Des centaines de milliers d'applications disponibles (57% gratuites).
- 🎯 **APK** (Android PacKage): un paquet contenant les fichiers d'une application. C'est grâce à lui qu'il est possible d'installer une application sur le téléphone.

Comment installer un .apk sur téléphone ?

Etape 1: Autoriser l'installation d'applications issues de sources inconnues.



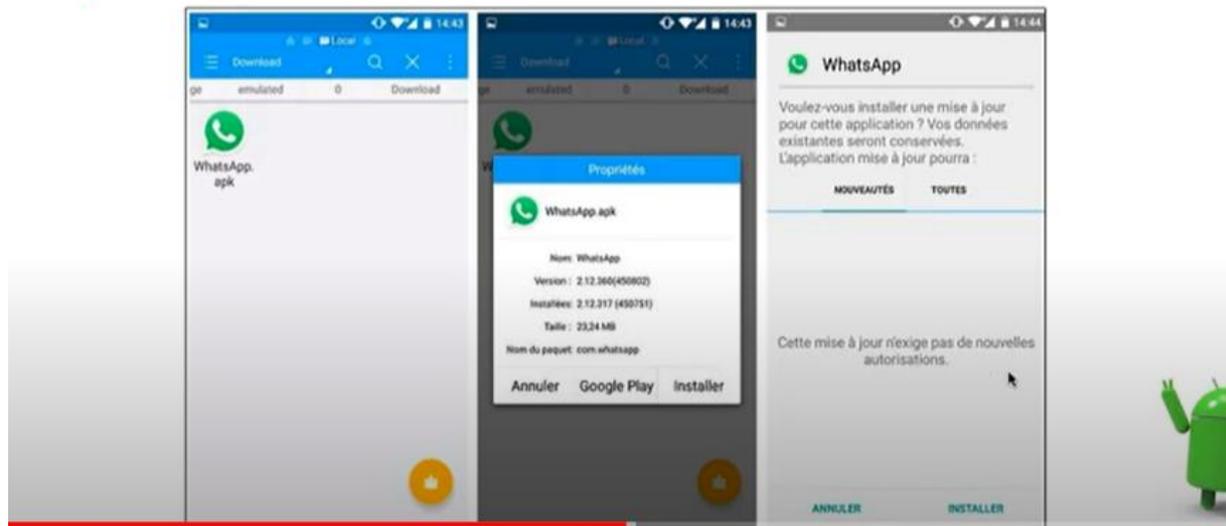
Comment installer un .apk sur téléphone ?

Etape 2: Télécharger l'APK

1. Téléchargez l'APK en question (<https://www.apkmirror.com/>).
2. Allez dans vos téléchargements et ouvrez-la.
3. Cliquez sur l'application, puis confirmez en cliquant sur *Installer*.

Comment installer un .apk sur téléphone ?

Etape 3: Installer l'APK



ANDROID ARCHITECTURE

- Architecture en « pile logicielle » constituée de 4 couches





Couche « Application » : regroupe un ensemble d' applications par défaut comme le navigateur web, l' agenda, le gestionnaire de contacts, ...etc.

ANDROID ARCHITECTURE

- Couche « Application Framework » : Cette couche permet au programmeur de construire de nouvelles applications, a travers des API, cette couche permet aux applications Android d'interagir avec le système Android. Elle permet :
 - L'accessibilité aux données des autres applications à travers des contentProviders
 - Activity Manager = gestion de cycle de vie d'une application.
 - Content Providers = le partage de données entres les applications.
 - Resource Manager = gestion des fichiers non codés (images, video, ..
 - Notifications Manager = affichage d'alerte dans la barre de titre.
 - Views System = IHM
 - ... etc.

ANDROID ARCHITECTURE

- Couche « Librairies & Runtime » : Android possède de nombreuses bibliothèques natives. Elles fournissent un accès direct aux ressources du système. C'est en quelque sorte la « base » de l'ensemble du système au niveau supérieur
- Contient des bibliothèques pour de nombreuses fonctions communes telles que :
 - les rendus graphiques (Open GL),
 - Le stockage de données (SQLite),
 - La navigation web (WebKit)
 - ...etc.
- contient aussi la machine virtuelle Dalvik pour l'exécution des programmes.

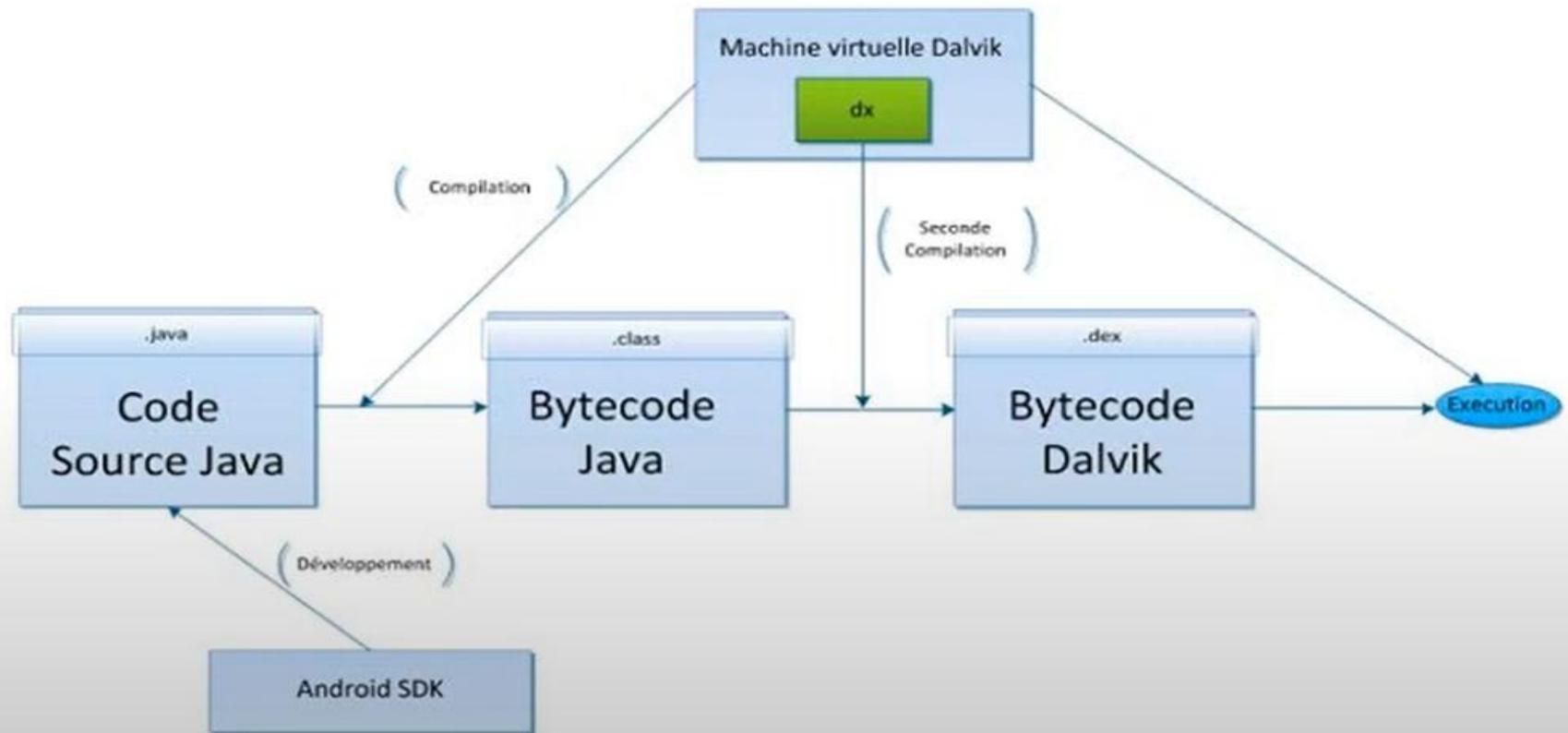


Couche « Noyau Linux » : Android s'appuie sur un noyau Linux 2.6 qui agit également comme une couche d'abstraction entre le matériel et le reste de la pile logicielle pour accéder aux périphériques: Camera, Audio, Bluetooth, etc. Il apporte aussi les services de sécurité, le gestionnaire de la mémoire, etc.

Niveau adjacent : Android Runtime (l' environnement d' exécution):

Pour qu' un programme fonctionne, il doit d' abord passer par une étape de compilation, qui consiste à traduire votre code Java en bytecode. Dans le cas d' Android, ce bytecode sera ensuite lupal un logiciel qui s' appelle la machine virtuelle Dalvik.





Dalvik virtual Machine (DVM)

- Consider the loop inside this simple method

```
public static long sumArray(int[ ] arr)
{
    long sum = 0;
    for (int i : arr)
    {
        sum += i;
    }
    return sum;
}
```

Java Byte Code

```
000b: iload 05
000d: iload 04
000f: if_icmpge 0024
0012: aload_3
0013: iload 05
0015: iaload
0016: istore 06
0018: iload_1
0019: iload 06
001b: i2l
001c: ladd
001d: lstore_1
001e: iinc 05,
```



30% less instructions

DVM Byte Code

```
0007: if-ge v0, v2, 0010
0009: aget v1, v8, v0
000b: int-to-long v5, v1
000c: add-long/2addr v3, v5
000d: add-int/lit8 v0, v0, #int 1
000f: goto 0007
```

ANDROID

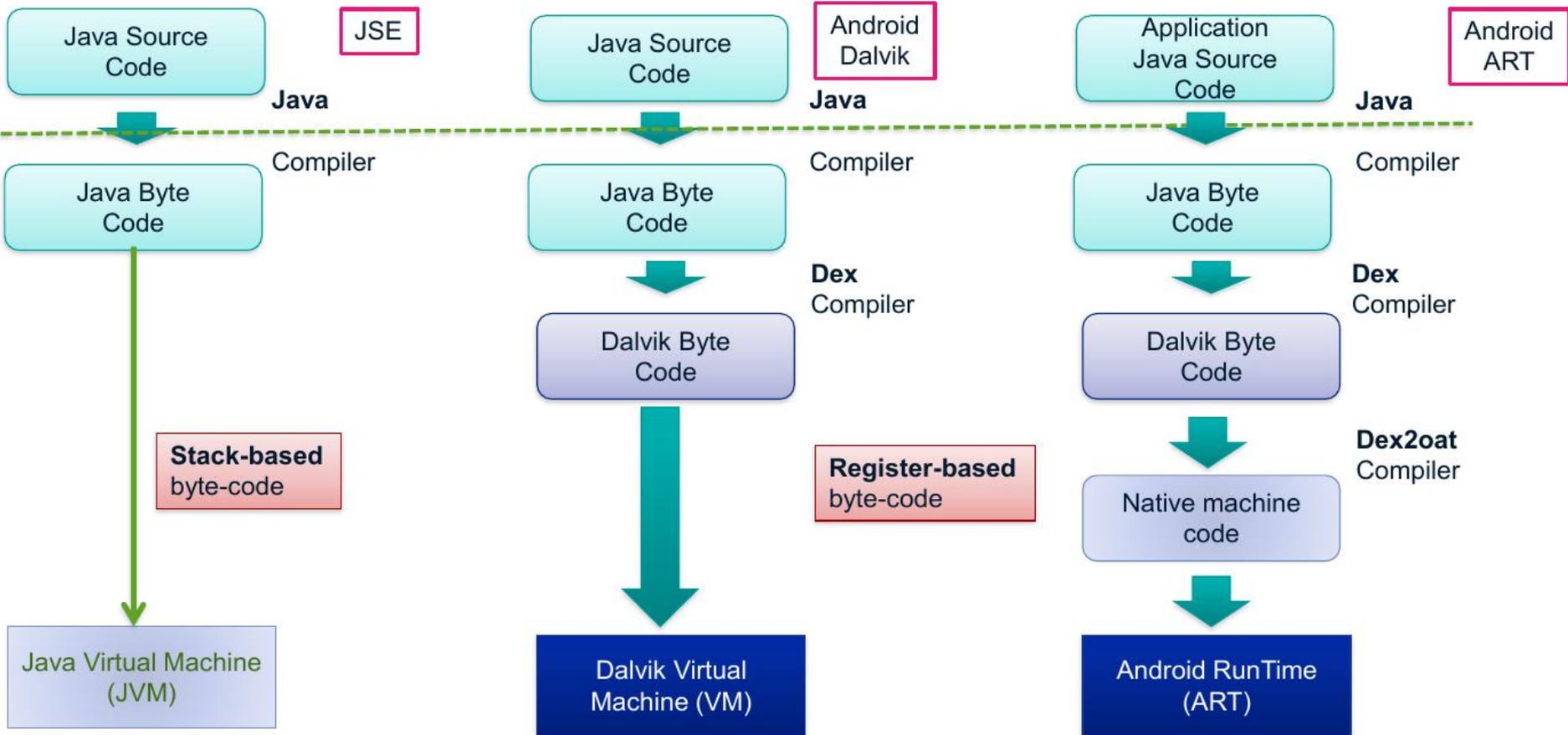
MACHINE VIRTUELLE DALVIK

- Dalvik VM est une machine virtuelle open source destinée aux téléphones mobiles et tablettes.
- Conçue pour exécuter du code Java pour des systèmes ayant des contraintes de place mémoire et rapidité d'exécution
- Permet l'exécution simultanée de plusieurs applications sur un appareil de faible capacité.
- Exécute du code .dex (Dalvik executable) = des .class adaptés à l'environnement Android
- Incorporée dans l'OS Android

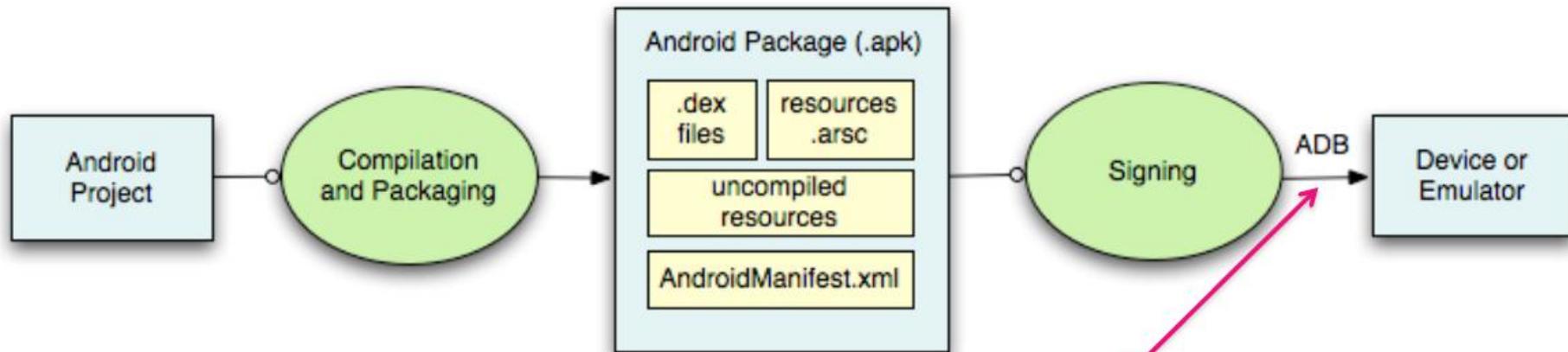
ANDROID RUNTIME (ART)

- À partir de la version 5.0 (Lollipop), ART (Android RunTime) remplace la machine virtuelle Dalvik
- ART utilise la compilation par avance (Ahead of Time compilation)
 - Pendant le déploiement, le code de l'application est traduit en code machine
 - Code compilé, jusqu' à 30 % plus grand
 - Permet une exécution plus rapide
 - Compilation 1 seule fois → économie de batterie

Android RunTime (Dalvik) JVM

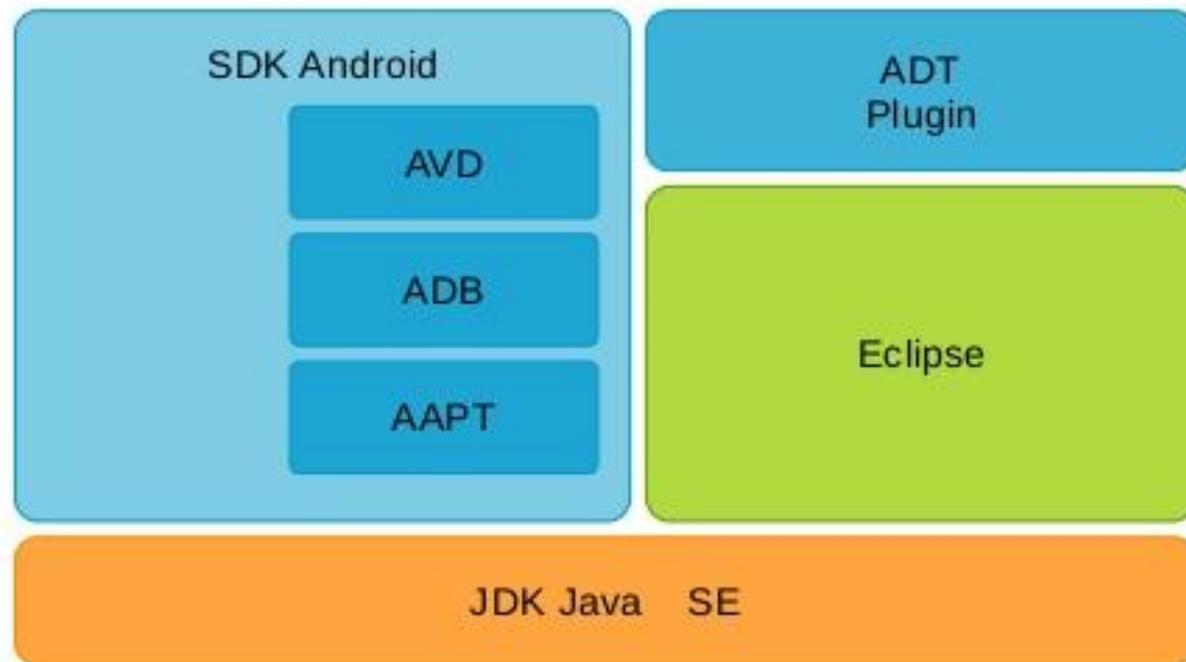


Android RunTime (Dalvik) JVM



- ✧ **ADB** is a client server program that connects clients on developer machine to devices/emulators to facilitate development.
- ✧ An IDE like **Android Studio** handles the entire **development process**

ANDROID PILE DES OUTILS DE DÉVELOPPEMENT



ANDROID SDK

- SDK : Software Development Kit
- Kit de développement logiciel de Android
- Contient les outils nécessaires pour créer, compiler et déployer les applications Android :
 - Exemples
 - Documentation
 - API de programmation du système
 - Emulateur pour tester les applications

ANDROID

SDK (AVD, ADB, AAPT)

- AVD : Android Virtual Device
 - Appareil virtuel que le développeur peut utiliser dans l'émulateur pour tester ses application
- ADB : Android Debug Bridge
 - Permet de connecter un appareil Android virtuel ou réel, dans le but de gérer le périphérique ou de déboguer votre application
- AAPT : Android Assets Packaging Tool
 - Sert à regrouper toutes les classes et ressources d'un projet dans un fichier .apk

ANDROID

ADT

- ADT : Android Developer Tools
- Un ensemble de composants (plug-ins), qui étendent l'IDE (Eclipse) avec des capacités de développement Android

ANDROID

PROJET À BASE D' ADT

- Un projet basé sur le plug-in ADT contient :
 - **src/** : les sources Java du projet
 - **libs/** : bibliothèques tierces
 - **res/** :
 - **res/drawable** : ressources images
 - **res/layout** : description des IHM en XML
 - **res/values** : chaînes de caractères et dimensions
 - **gen/** : les ressources auto-générées par ADT
 - **assets/** : ressources brutes
 - **bin/** :
 - **bin/classes** : les classes compilées en .class
 - **Bin/classes.dex** : exécutable pour la JVM Dalvik
 - **Bin/myapp.apk** : application empaquetée avec ses ressources et prête pour le déploiement

RSOURCES

- Applications Mobile; cours de R. meghatria et M. Khaled ; UDBKM 2018
- Internet