
Introduction au développement Android

UE élective TAF

Thierry Duval, Mayte Segarra
Département Informatique

Objectifs de l'enseignement

- Expliquer le rôle et l'intérêt des éléments de l'écosystème Android utilisés dans le développement d'une application
- Suivre la chaîne de développement d'une application Android
- Développer une application interactive simple en appliquant les principes de base de conception au niveau de l'IHM et de la gestion de données persistantes
- Rechercher de manière autonome des ressources pertinentes pour les futurs développements Android

Qu'est-ce qu'Android ? Android是什么



- Un système d'exploitation (comme Linux, Windows ou Mac OS) Android是一种操作系统，类似于Linux、Windows或Mac OS
- Dédié aux dispositifs « *wearables* », comme iOS ou Windows mobile 主要应用于智能手机、平板电脑、手表和汽车等设备
 - ✓ L'expérience utilisateur d'un smartphone, tablette, montre, voiture, nécessite des interactions et un fonctionnement particuliers...
- Parts du marché des mobiles (source : <https://fr.statista.com/infographie/8204/parts-de-marche-des-systemes-exploitation-pour-smartphones/>)
 - ✓ 84 % Android, 16 % iOS 市场份额占比：84%的Android，16%的iOS
- Basé sur Linux et ouvert [licence Apache 2.0] (source : <http://source.android.com/source>) Android基于Linux并采用开源许可证

Historique

- 2003, Android est une PME américaine
- 2005, rachat par Google
- 01/2007, iPhone
- 11/2007, *Open Handset Alliance* (35 entreprises à l'époque, plus de 80 aujourd'hui) pour développer un OS « *open source* »
- 2008, 1ère version Android (1.5) puis rapidement 1.6
- Actuellement, version 14 « Android 14 » (fin 2023)

Philosophie

- « *Open source* » (sources : <http://source.android.com/source>)
- Presque gratuit : 25 USD pour publier autant d'applications que vous voulez sur Google Play
- Facile à développer, à publier et à vendre (mais pour pas cher...)
- Flexible (tablettes, montres, voitures, avec ou sans clavier, processeurs différents...)
 - ✓ Problèmes de compatibilité d'applications
 - ✓ Contraintes des systèmes embarqués (RAM, priorité des tâches ...)

Architecture de la plateforme Android



Dalvik si Android < 5.0

Environnement de développement

➤ JDK : et oui, c'est du Java ...

- ✓ JDK vs JRE
- ✓ Vérifier l'installation Java sur une machine ('java -version')
- ✓ Si pas installé, RdV à
<http://www.oracle.com/technetwork/java/javase/downloads>

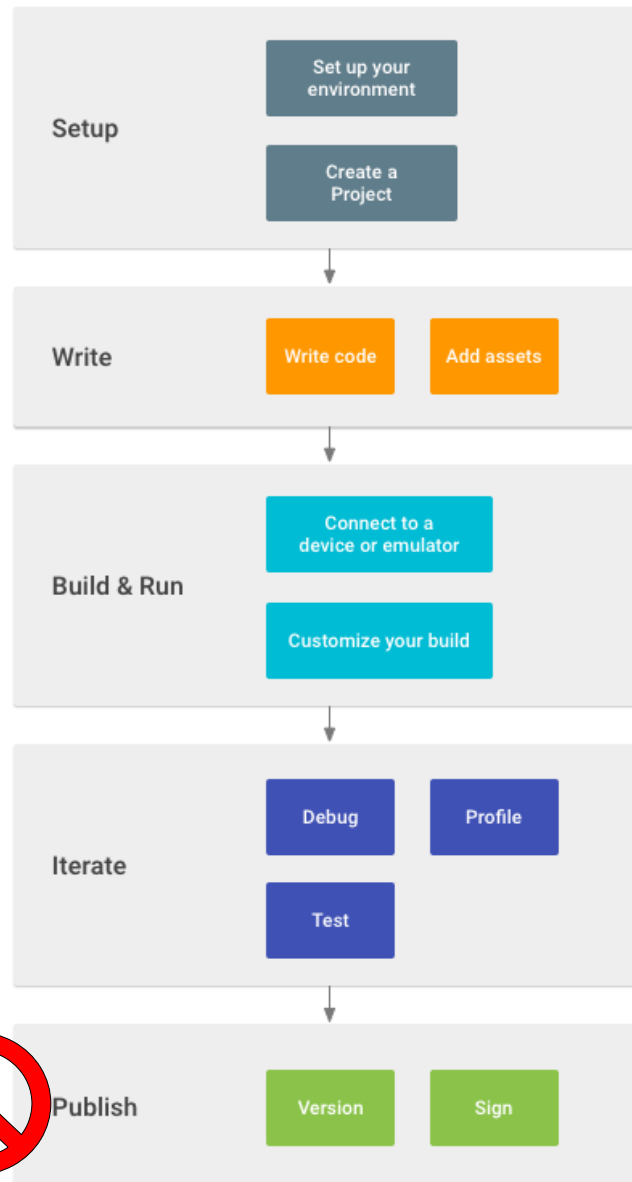
➤ Android Studio (depuis 2014)

- ✓ IDE privilégié par Google pour le développement d'applications
- ✓ Besoins : (source : <http://developer.android.com/studio>)
 - × 8 Go RAM, 8 Go disque
 - × Windows 8/10, MAC OS ≥ 10.14 , Linux glibc ≥ 2.31
 - × Résolution écran $\geq 1280 \times 800$

Processus de développement Android Android应用程序开发流程

source :

<http://developer.android.com/studio/workflow.html>



Installer et configurer l'IDE
pour le développement
安装配置IDE

Écrire l'application : code et
ressources pour les dispositifs visés
编写应用程序代码和资源

Générer le binaire et l'exécuter sur un
dispositif ou l'émulateur
生成二进制文件

Déboguer et optimiser le code
调试优化代码

Mettre à disposition des utilisateurs
sur Google Play
在Google Play上发布应用程序



Présentation rapide d'une première application

Présentation rapide d'une première application

Phase de « *setup* »

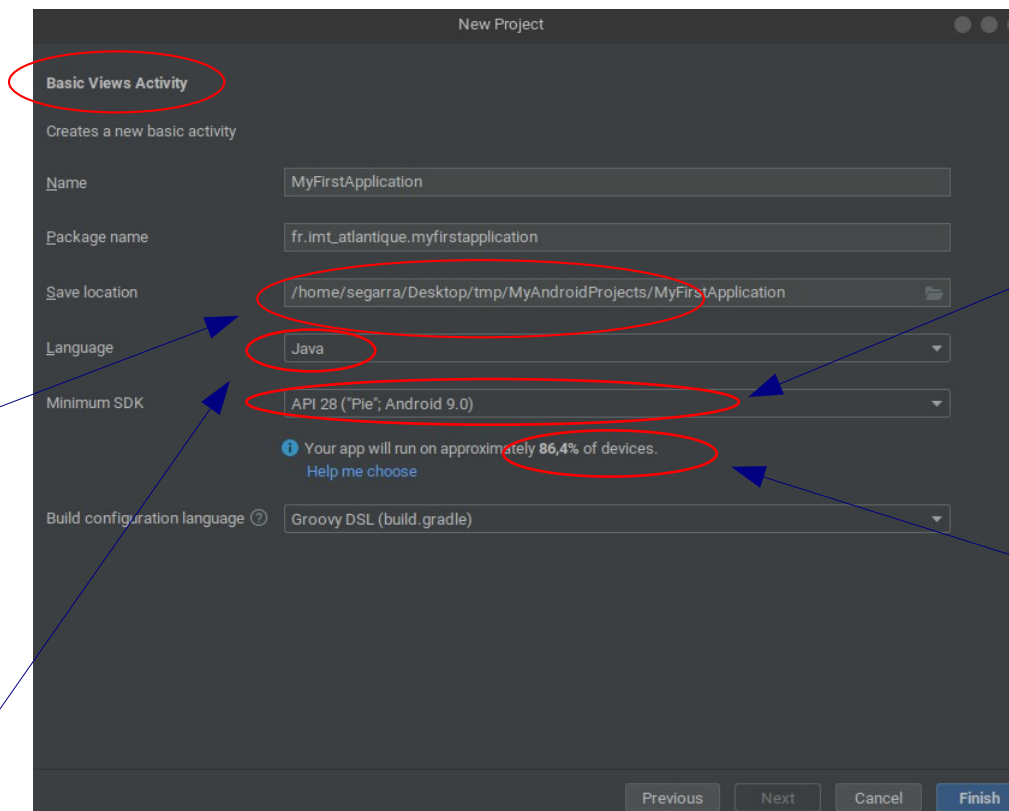
Écrire l'application

Exécuter l'application

[« Derrière les rideaux »]

Phase de « *setup* »

- Installation Android Studio (attention, il faut avoir une connexion filaire) → Déjà fait normalement ...
- Création d'une application « *MyFirstApplication* »



Attention à ne pas donner des caractères spéciaux, des espaces, des accents

SDK minimum

Dispositifs couverts

Java ici

Android SDK

➤ SDK (« *Software Development Kit* »)

用于为特定目标开发（编程，编译，执行，测试，调试和打包）的工具

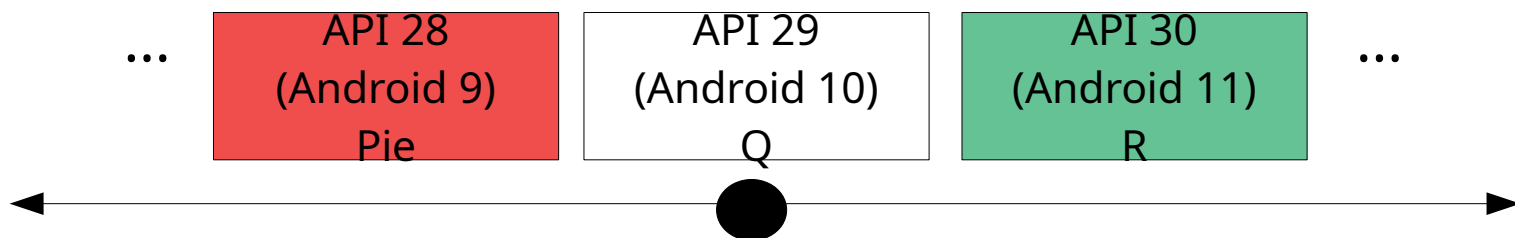
- ✓ Outils pour développer (programmer, compiler, exécuter, tester, déboguer et « packager ») pour une cible particulière

➤ Android SDK

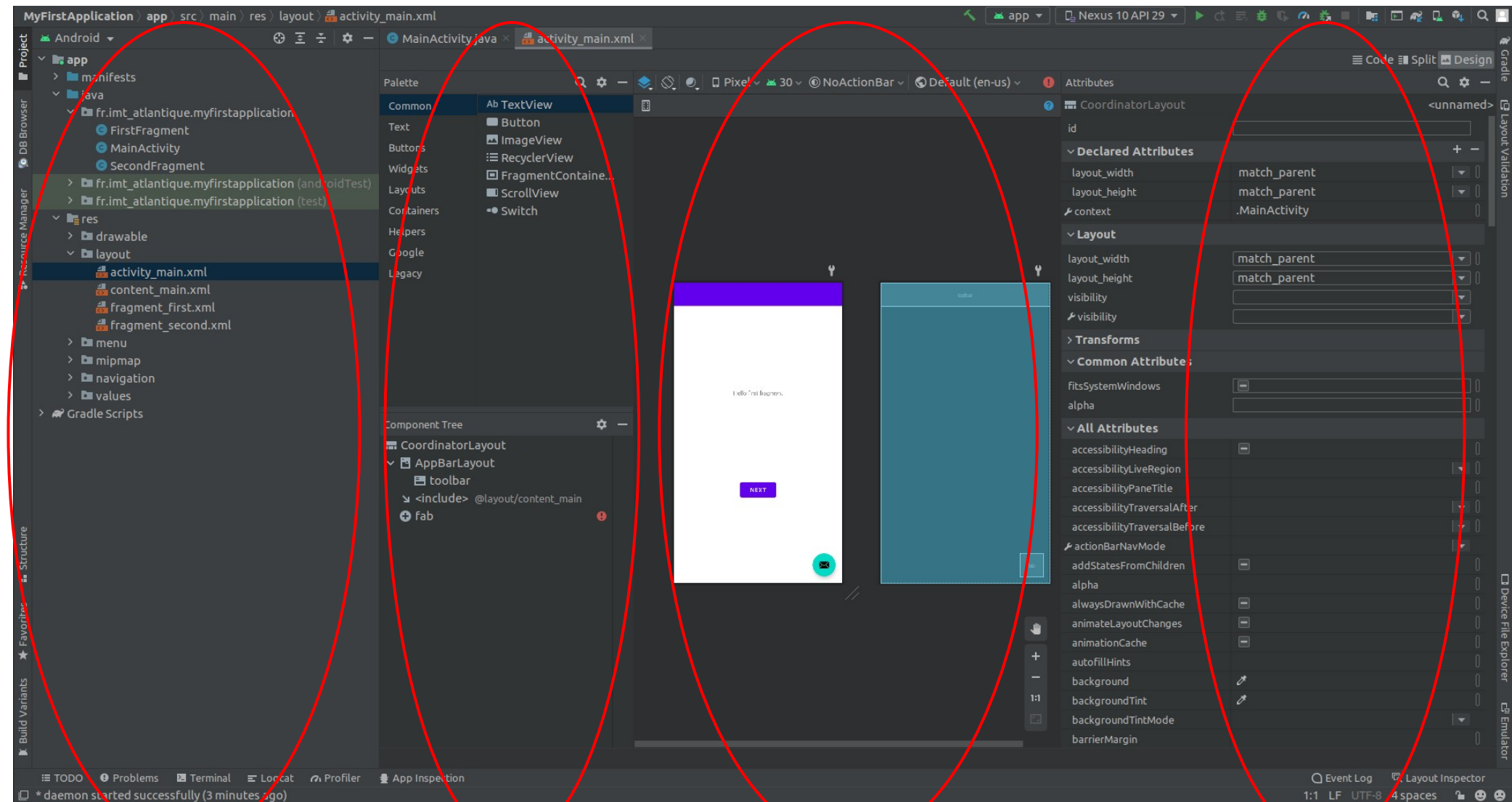
用于开发目标为 Android 设备的应用程序的工具

- ✓ Outils pour développer des applications ayant comme cible un dispositif Android “API 级别”和 Android 版本“配对”
- ✓ « *API Level* » et version de Android vont « de pair »
- ✓ « *Forward compatibility* » mais pas de « *backward compatibility* » “向前兼容”

➤ SDK minimum (« *Minimum API level* »)



Après ajout d'une « Activity »



1

2

3

4

Si on récapitule, une application Android ...

- Un projet dans Android Studio
- Android Studio génère un squelette pour l'application
avec Android Studio 生成应用程序的框架包括
 - ✓ Un fichier AndroidManifest.xml avec des informations sur l'application 一个包含应用程序信息的 AndroidManifest.xml 文件
 - ✓ Un répertoire 'java' avec le code source de l'application 一个包含应用程序源代码的 'java' 目录
 - ✓ Un répertoire 'res' avec les ressources utilisées pour l'IHM 一个包含用于用户界面的资源的 'res' 目录
 - ✓ Les *scripts* nécessaires à la construction du « *binaire* »
- [Peut avoir un ensemble de « *application modules* », un par type de dispositifs cible (téléphone/tablette, voiture, montre ...)]

Écrire l'application

- Modifier le squelette généré par Android Studio pour
 - ✓ Ajouter le code Java et les ressources nécessaires à l'IHM
 - ✓ Modifier le fichier AndroidManifest.xml
 - ✓ Modifier les *scripts* de construction de l'application si nécessaire

Android 应用程序：被设计为一系列屏幕，有具有关联行为的图形组件

- Application Android : conçue comme une suite d'écrans ayant des composants graphiques qui ont un comportement associé

对于应用程序的每个功能，需要：

- ✓ Pour chaque fonction de l'application il faut :
 - × Définir l'(les)écran(s) : composants graphiques et leur agencement (« *layout* », gabarit en français) 定义屏幕：图形组件及其布局
 - × Réaliser le comportement que l'on souhaite (la fonction) 实现所需的行为
 - × Relier le comportement aux composants graphiques 将行为与图形组件关联起来

Exécuter l'application dans l'émulateur

➤ Émulateur Android : Android 模拟器

- ✓ Simule un dispositif, le matériel et le logiciel (notamment la version d'Android à utiliser) 模拟设备, 硬件和软件
- ✓ Permet d'exécuter et tester les applications sans dispositif physique → outil privilégié pour s'affranchir de la fragmentation 允许在没有物理设备的情况下运行和测试应用程序
不能模拟设备的所有功能
- ✓ Ne simule pas toutes les possibilités d'un dispositif (voir <https://developer.android.com/studio/run/advanced-emulator-usage?hl=fr>)

➤ Outils dans l'écosystème Android :

- ✓ « *Android Virtual Device* » (AVD)
- ✓ « *Device Manager* » gère tous les dispositifs, physiques et virtuels (Tools → Device Manager dans Android Studio)

Exécuter l'application sur un dispositif physique

- Activer option « *USB debugging* » (ou « *wireless debugging* » à partir d'Android 11) sur le téléphone
- Outils dans l'écosystème Android
 - ✓ « *Android Debug Bridge* » (ADB) pour la communication entre le PC et le téléphone ou un AVD

在手机上启用“USB 调试”选项（或从 Android 11 开始使用“无线调试”）

Android 生态系统中的工具

ADB 用于 PC 和手机或 AVD 之间的通信

Présentation rapide pour une première application

Phase de « *setup* »

Écrire l'application

Exécuter l'application

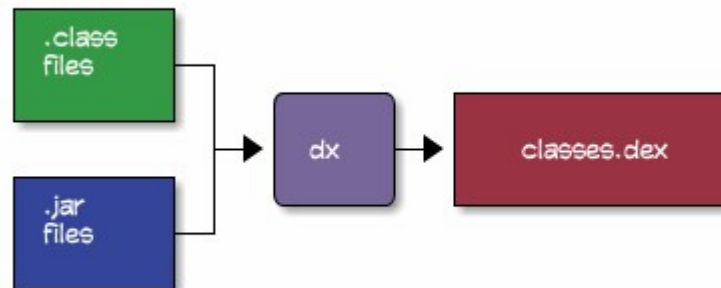
[« Derrière les
rideaux »]

Des sources à l'APK 从源代码到 APK

- Installer une application sur un Android : installer un fichier APK (« *Android Package* ») 在 Android 上安装应用程序
- Création d'un APK 创建 APK:
 - 从 .java 文件到 .class 文件的工具 → javac
 - 从 .class 文件到 .dex 文件的工具 → dx
 - ✓ Fichiers .java en .class → outil 'javac'

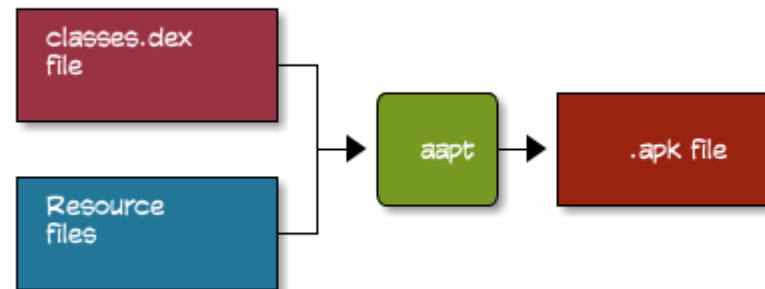


- ✓ Fichiers .class en dex → outil 'dx'

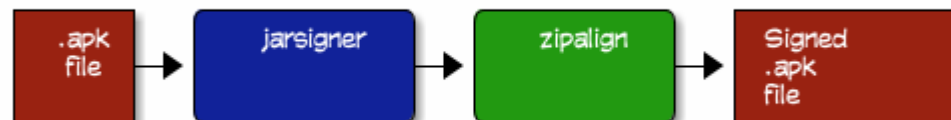


Des sources à l'APK (2)

- Installer une application sur un Android
- Création d'un APK
 - ✓ Fichiers .java en .class → outil 'javac'
 - ✓ Fichiers .class en .dex → outil 'dx'
 - ✓ Fichiers .dex + ressources en .apk → outil 'aapt'

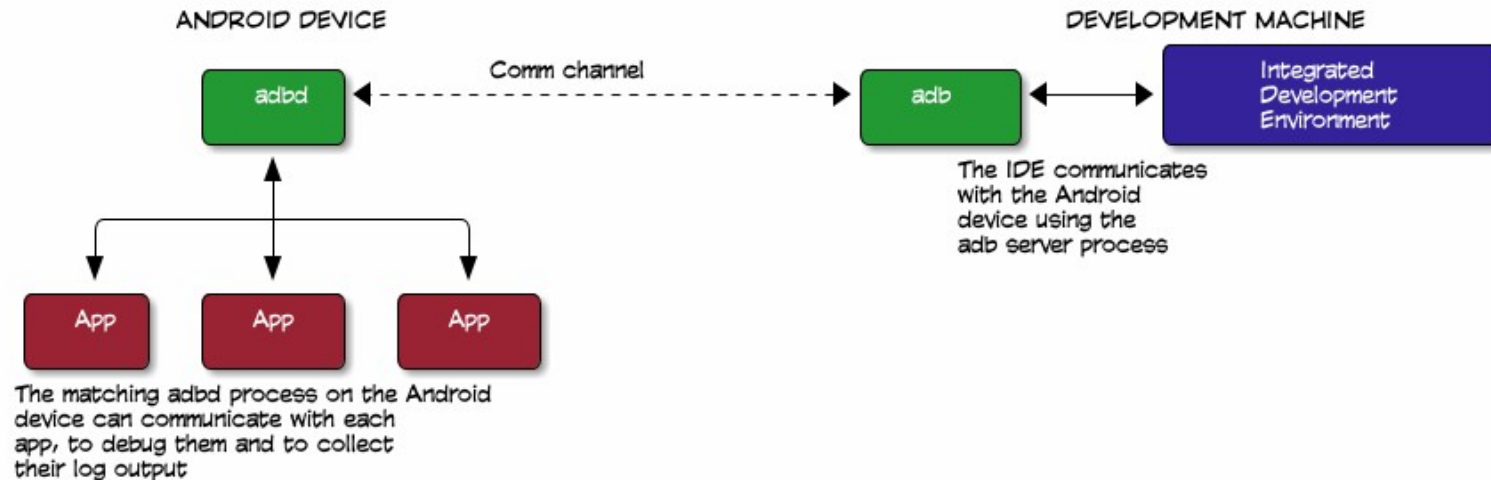


- ✓ [Signer le fichier .apk]



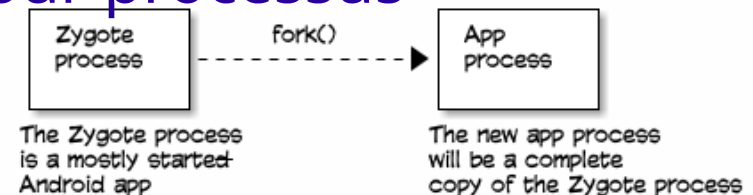
De l'APK au binaire

➤ Déploiement de l'APK



➤ Exécution de l'application

✓ Création de l'espace mémoire pour processus



✓ Fichiers .dex en OAT → outil 'dex2oat'

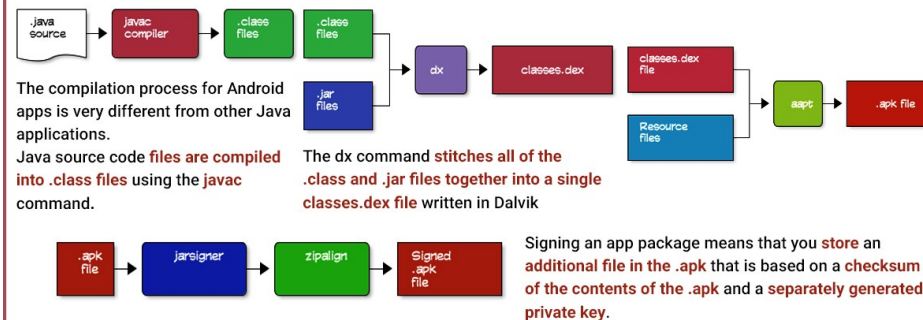


De l'APK au binaire (2)

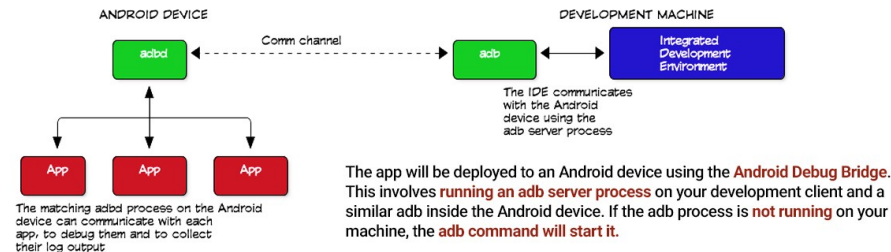
- Déploiement de l'APK
- Exécution de l'application
 - ✓ Modèle d'exécution basé sur Linux
 - × Un processus par application → indépendance des applications

Des sources à l'exécution

How APK is Build?

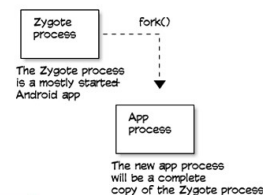


How APK is Deployed?



How APK is Run?

A process called **Zygote** is used to launch the app. Zygote is an incomplete version of an Android process—its **memory space** contains all the **core libraries** that are needed by any app, but it **doesn't yet include any of the code** that's specific to a particular app.



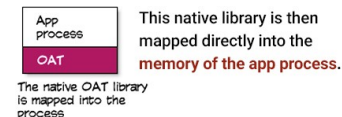
Zygote creates a **copy of itself** using the fork system call. Android is a Linux system, and the fork call can duplicate a process like Zygote very quickly. This is the reason Zygote process is used: **it's a lot faster to duplicate an half-started process like Zygote**, than it is to load a new process from the main system files. Zygote means your app launches faster.



The new app process now **needs to load the code** that's specific to your app. Android will **convert the Dalvik byte-codes in classes.dex into native machine code**.

Java code is now converted into a piece of native compiled code. Technically, the classes.dex will be converted into an **ELF shared object**.

Android calls this library format **OAT** and the tool that converts the classes.dex file is called **dex2oat**.



Organisation pédagogique

Qu'est cette UE et que n'est elle pas ?

➤ Ce n'est pas un enseignement

- ✓ où des particularités de certains outils sont expérimentées
- ✓ où des « *features* » avancées d'Android sont abordées
- ✓ où aller chercher la dernière technologie à la mode (ou qui vient de sortir)

➤ C'est un enseignement

- ✓ où les bases Java objet sont considérées comme acquises
- ✓ où des outils de base pour le développement Android sont expérimentés

Objectifs de l'enseignement

- Expliquer le rôle et l'intérêt des différents éléments de l'écosystème Android utilisés dans le développement d'une application
- Suivre la chaîne de développement d'une application Android
- Développer une application interactive simple en appliquant les principes de base de conception au niveau de l'IHM et de la gestion de données persistantes
- Rechercher de manière autonome des ressources pertinentes pour les futurs développements Android

Organisation de l'enseignement

- Conçu comme un ensemble de thématiques dans le développement Android
 - ✓ IHM, Persistance, Cartes, Capteurs
- Activités de type TP (travail en binôme) systématiquement
 - ✓ Mais des points « cours » peuvent être faits pendant les séances pour introduire les nouveaux concepts
- Évaluation
 - ✓ Oral sur le travail des binômes pour la majorité des thématiques
 - ✓ Marquées en rouge sur le site Web de l'UE

Espaces de travail

- **Site Web** : formations.imt-atlantique.fr/introductionAndroid
 - ✓ Supports séances
 - ✓ Programmation de l'enseignement
 - ✓ Critères d'évaluation
- **Moodle** : moodle.imt-atlantique.fr/course/view.php?id=794
 - ✓ Création des binômes de travail
 - ✓ Dépôt du code de certaines étapes de développement