

Développement d' Applications Mobiles

CHAPITRE II: ANDROID OS

I. Système Android

- ❖ Android est un système d'exploitation (OS) mobile, open source et basé sur Linux.
 - Le site source.android.com offre le code source d'Android.
- ❖ Ce système d'exploitation a été développé par Google et ***Open Handset Alliance (OHA)***.
- ❖ Plus de 30 sociétés ont participé à l'OHA, dont Qualcomm, Broadcom, HTC, Intel, Samsung, Motorola, Sprint, Texas Instruments et les opérateurs sans fil japonais comme KDDI.

I.1. SDK Android

- ❖ Pour le développement d'applications, Google dispose un **(SDK Software Development Kit)** aux développeurs.
- ❖ Le SDK Android est composé de:
 - Des bibliothèques API (Application Programming Interface).
 - Emulateur qui permet de créer des dispositifs virtuels pour tester les application.
 - En plus que la documentation et des exemples de code disponible sur le site : <https://developer.android.com/>

1.2. Historique

Date & version	Nom & API	Nouveautés
Septembre (2008) 1.0	- (API 1)	Première version du système
Février (2009) 1.1	- (API 2)	Sauvegarder des pièces jointes MMS
Avril (2009) 1.5	Cupcake (API 3)	<ul style="list-style-type: none">- On-screen keyboard- Enregistrement vidéo
Septembre (2009) 1.6	Donuts (API 4)	<ul style="list-style-type: none">- Quick search box- Amélioration Android Market!

1.2. Historique

Date & version	Nom & API	Nouveautés
Octobre (2009) 2.0-2.1	Eclair API (5-7)	- Amélioration Google Maps.
Juin 2010 (2.2)	Froyo API (8)	Hotspot wifi
Décembre 2010 (2.3)	Gingerbread API (9-10)	- NFC - APIs pour le développement des jeux
Février (2011) 3.0-3.2	HoneyComb API (11-13)	- Designer pour tablette

1.2. Historique

Date & version	Nom & API	Nouveautés
Octobre (2011) 4.0	Ice cream Sandwic API (14-15)	<ul style="list-style-type: none">- Plateforme unifier- Control de l'utilisation des données.
Juin 2012 (4.1-4.3)	Jelly bean API (16-18)	<ul style="list-style-type: none">- Google Now- Google Chrome
Octobre 2013 (4.4)	KitKat API (19-20)	<ul style="list-style-type: none">- OK google
Novembre (2014) 5.0	Lollipop API (21-22)	<ul style="list-style-type: none">- ART au lieu de Dalvik- Cryptage des données

1.2. Historique

Date & version	Nom & API	Nouveautés
Octobre (2011) 4.0	Ice cream Sandwic API (14-15)	<ul style="list-style-type: none">- Plateforme unifier- Control de l'utilisation des données.
Juin 2012 (4.1-4.3)	Jelly bean API (16-18)	<ul style="list-style-type: none">- Google Now- Google Chrome
Octobre 2013 (4.4)	KitKat API (19-20)	<ul style="list-style-type: none">- OK google
Novembre (2014) 5.0	Lollipop API (21-22)	<ul style="list-style-type: none">- ART au lieu de Dalvik- Cryptage des données

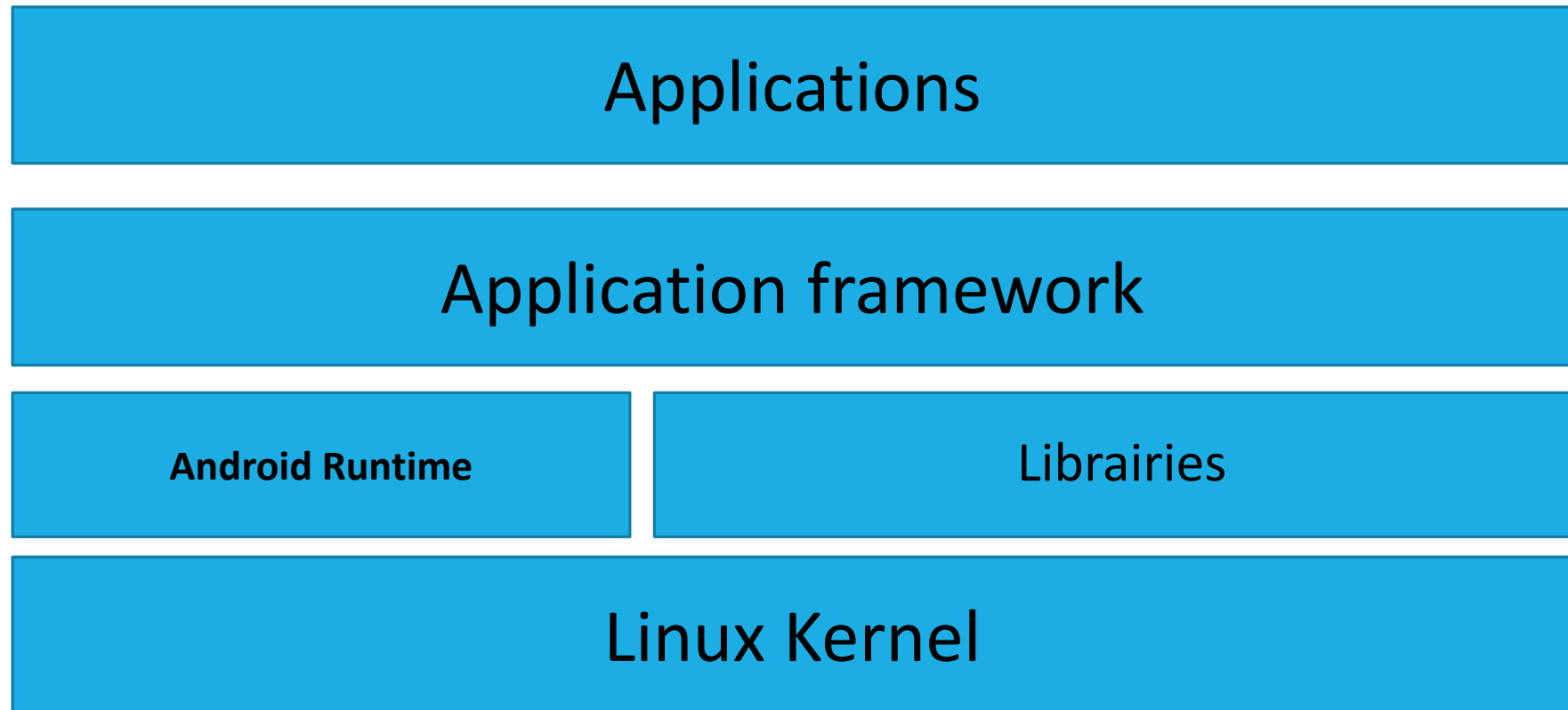
1.2. Historique

Date & version	Nom & API	Nouveautés
Octobre (2015) 6.0	Marshmallow (23)	- fingerprint biometric unlocking, USB Type-C, android Pay
Aout (2016) 7.0-7.1	Nougat API (24-25)	- Affichage simultané de plusieurs applications - Basculement entre applications amélioré.
21 août 2017.	Oreo API(26-27)	- sécurité
6 août 2018	Pie API (28)	-Simplicité, IA, Wellbeing Digital

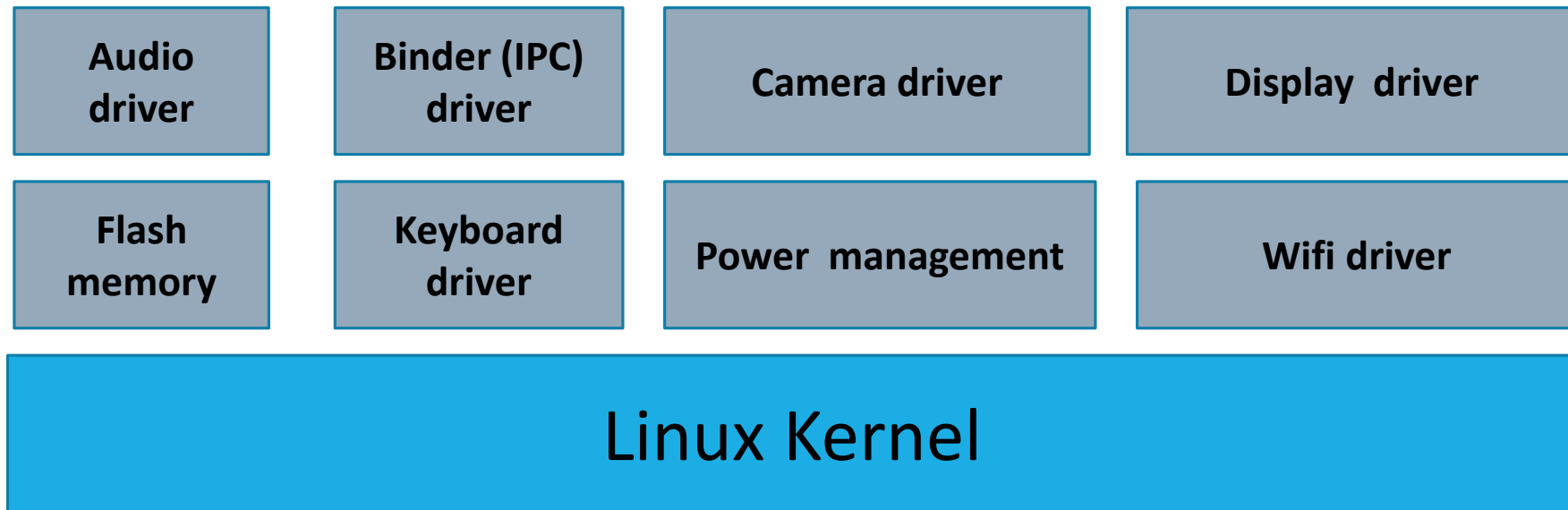
1.2. Historique

- ❖ Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs, à améliorer l'aspect graphique ou encore à ajouter de nouvelles fonctionnalités.
- ❖ La dernière version de la plateforme est maintenant Android 10.
 - Meilleure confidentialité: Les Apps nécessitent une permission pour accéder à votre position et à vos fichiers.
 - Support pour les téléphones pliables
 - Mode sombre
 - Navigation par geste

I.3. Architecture du système Android



I.3.1 Noyau



I.3.1 Noyau

❖ La plate-forme Android est composée de différentes couches:

- 1. Le noyau linux:** Le noyau Linux, comme n'importe quel OS, est responsable de la gestion du matériel à l'aide de drivers, la mémoire, les processus ou encore les couches réseaux basses.

1.3.2 Android Runtime

Core libraries

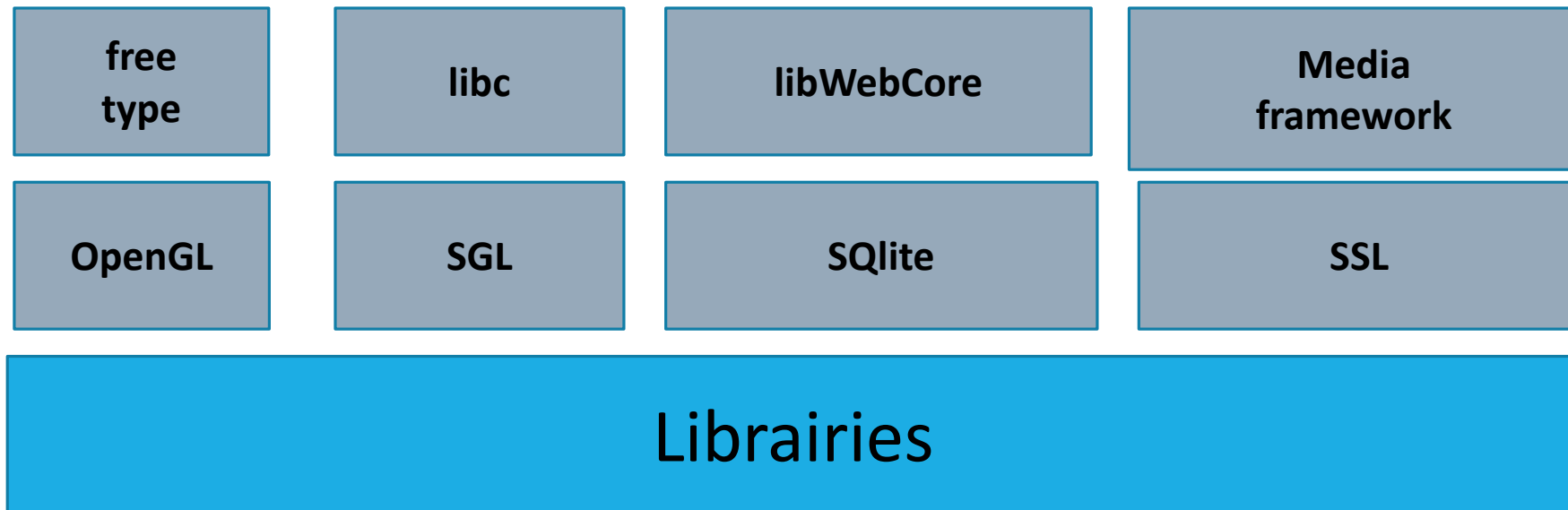
Dalvik Virtual machine

Android runtime

1.3.2 Android Runtime

- ❖ L'environnement d'exécution Android Runtime : Toute application est exécutée dans son propre processus, dans sa propre **Dalvik Virtual Machine**.
- ❖ Le noyau Linux sur lequel la Dalvik virtual machine s'appuie pour gérer le multithreading, la mémoire, la gestion des processus, etc.
- ❖ A partir de Lollipop (5.0), Android dispose d'ART comme JVM pour Android. Elle compile l'application au moment de *l'installation* (Ahead-of-time compilation: compilation à l'installation).

1.3.3 Bibliothèques



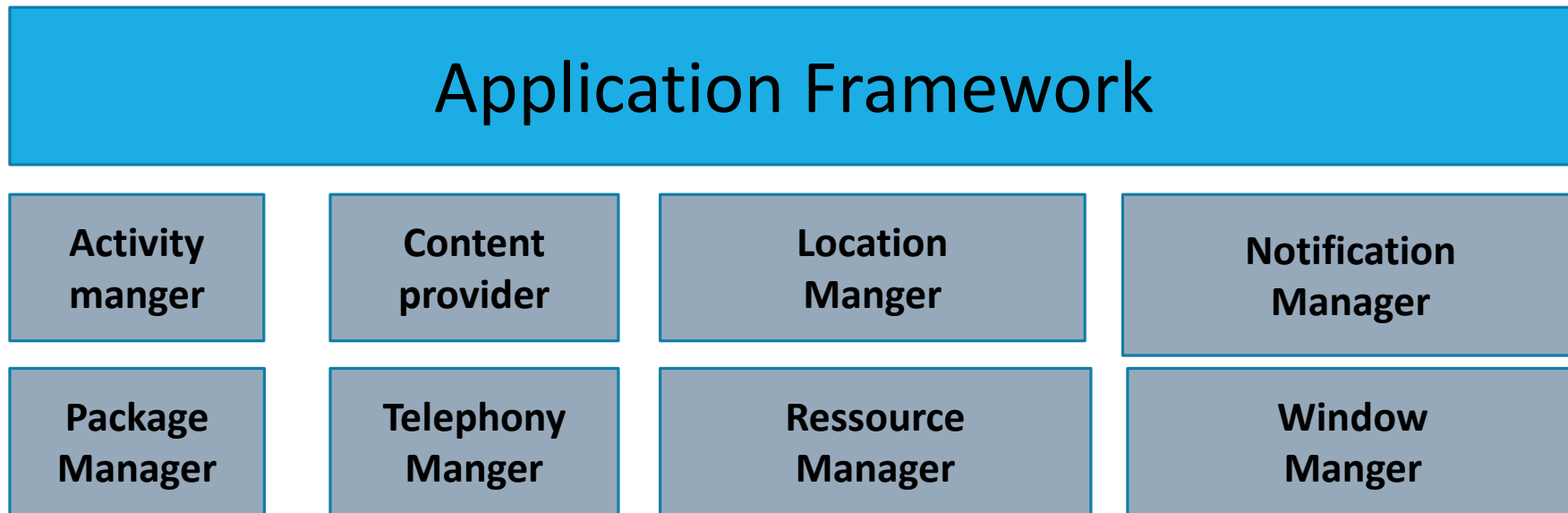
1.3.3 Bibliothèque

- ❖ Les bibliothèques (des bibliothèques graphiques, multimédias, SQLite pour les bases de données, WebKit pour la navigation web,...)
 - *Libc*: it is c standard lib.
 - *SSL*: Secure Socket Layer for security
 - *SGL*: 2D picture engine where SGL is “Scalable Graphics Library”
 - *OpenGL/ES*: 3D image engine *Media Framework*: essential part of Android multi-media

1.3.3. Bibliothèque

- *SQLite*: Embedded database
- *Web Kit*: Kernel of web browser
- *Free Type*: Bitmap and Vector
- *Surface Manager*: Manage different windows for different applications

1.3.4. Application Framework

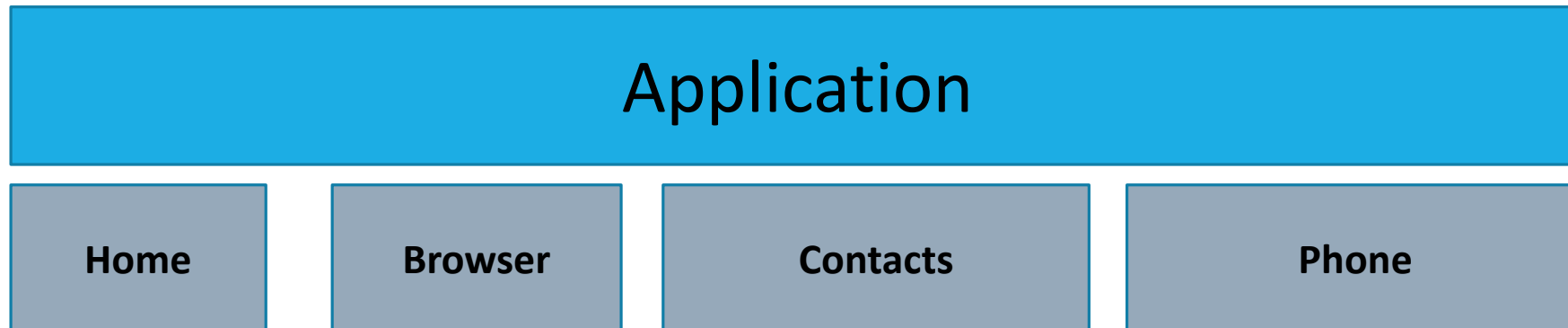


1.3.4. Application Framework

- ❖ Cette couche permet au programmeur de construire de nouvelles applications.
- ❖ Cette couche fournit la gestion :
 - des Activities pour l'enchaînement des écrans,
 - des Views (IHM),
 - des ressources qui sont les fichiers et les images (Resource Manager),
 - des ContentProviders (fournisseurs de contenus) qui permettent l'accessibilité aux données des autres applications (les partages de données).
 - des notifications (affichage d'alerte dans la barre de titre)

1.3.5. La couche Application

❖ Navigateur, gestion des contacts, application de téléphonie, Bureau,...



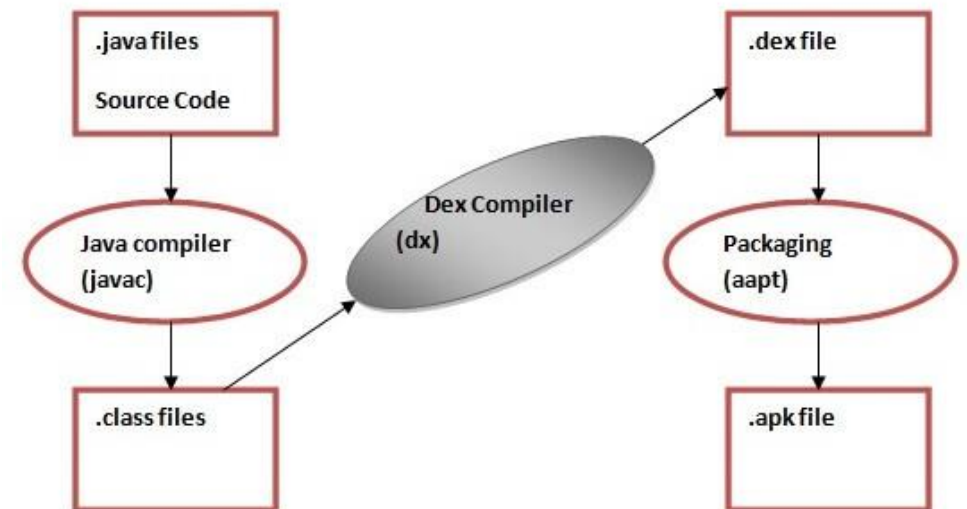
II. Application compilation

❖ Dans Android, les classes Java seront converties en bytecode DEX et ce format de bytecode DEX est traduit en code machine à l'aide des runtimes ART ou Dalvik.

➤ Le JIT ou Just In Time ne compile qu'une partie de votre code et par la suite utilise très moins d'espace physique sur votre appareil.

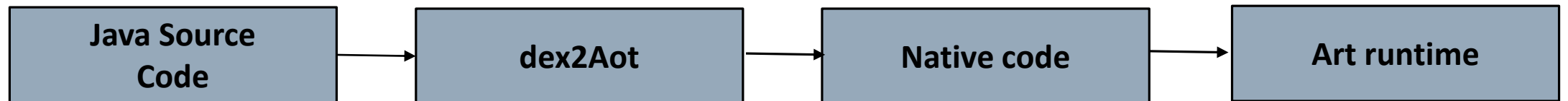
➤ **Dalvik est un compilateur Just In Time (JIT).**

aapt: Android Asset **Packaging** Tool



II.1 Dalvik vs ART

- ❖ ART utilise l'approche AOT (Ahead Of Time) et compile tout le code pendant l'installation, tandis-que le Dalvik utilise l'approche JIT (Just In Time) et la compilation sera faite dynamiquement pendant l'exécution.



III. Éléments d'une application android

- ❖ Une application Android peut être composée des éléments suivants:
 - activités (**Activity**): Il s'agit d'une partie de l'application présentant une interface à l'utilisateur;
 - Des services (**Service**): il s'agit d'une activité tâche de fond sans interface associée;
 - Des fournisseurs de contenus (**android.content.ContentProvider**) permettent le partage d'informations au sein ou entre applications;
 - Des récepteurs d'*Intents* (**android.content.BroadcastReceiver**): permet de déclarer être capable de répondre à des Intents;

III. Éléments d'une application android

❖ Un intent est une sous classe de **android.content.Intent**,
Un événement (intent) est une "intention" à faire quelque chose à un composant Android. Les activités, services et récepteurs d'informations utilisent les Intents pour communiquer entre eux.

III.1 Le fichier manifest

- ❖ Toute application doit contenir le fichier XML `AndroidManifest.xml`,
 - Ce fichier déclare, les différents composants de l'application
- ❖ Ce fichier contient aussi :
 - Les permissions nécessaires à l'application (accès internet).
 - Le numéro minimum de la version API utilisée,
 - des logiciels et matériels utilisés par l'application (caméra)
 - bibliothèques supplémentaires à l'API de base (Google maps, library)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="andro.jf"
android:versionCode="1"
android:versionName="1.0">
  <application android:icon="@drawable/icon"
    android:label="@string/app_name">
    <activity android:name=".Main"
      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>...</service>
    <receiver>...</receiver>
    <provider>...</provider>
  </application>
</manifest>
```

III.2. Les ressources

- ❖ Les ressources sont des éléments constituant l'application autre que le code source (fichiers image, vidéo, du texte...)
- ❖ Les ressources d'une application Android sont utilisées dans le code au travers de la **classe statique R**.
- ❖ Les ressources sont spécifiables dans res :
 - Les menus.
 - Les images (R.drawable).
 - Dimensions (R.dimen).
 - Des couleurs (R.color).

III.2. Les ressources

- ❖ Les chaînes constantes de l'application sont situées dans `res/values/strings.xml`.

➤ Exemple:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">TP1</string>
<string name="hello">Hello word!</string>
</resources>
```

- ❖ La récupération de la chaîne se fait via le code :

```
Resources res = getResources();
String str = res.getString(R.string.hello);
```

Android

