Chapitre II :
La plateforme Android

Plateforme Android

- La plateforme Android est l'environnement logiciel pour les applications Android, comprenant le système d'exploitation qui fournit services de base et API pour accéder aux fonctionnalités de l'appareil, ainsi que les bibliothèques et frameworks nécessaires à la création d'applications.
  - Plateforme pour les appareils mobiles
  - Gratuite.
  - Open source
  - Flexible

Qu'est-ce que “API” ?

- **Application programming interface**
  - En programmation informatique, une interface de programmation d'application (API) est **un ensemble** de définitions de **sous-programmes**, de **protocoles** et **d'outils** pour la construction de logiciels d'application.
  - En termes généraux, il s'agit d'**un ensemble de méthodes** de communication clairement définies entre différents composants logiciels.
  - Une bonne API **facilite** le développement d'un programme informatique en fournissant tous les blocs de construction, qui sont ensuite assemblés par le programmeur.

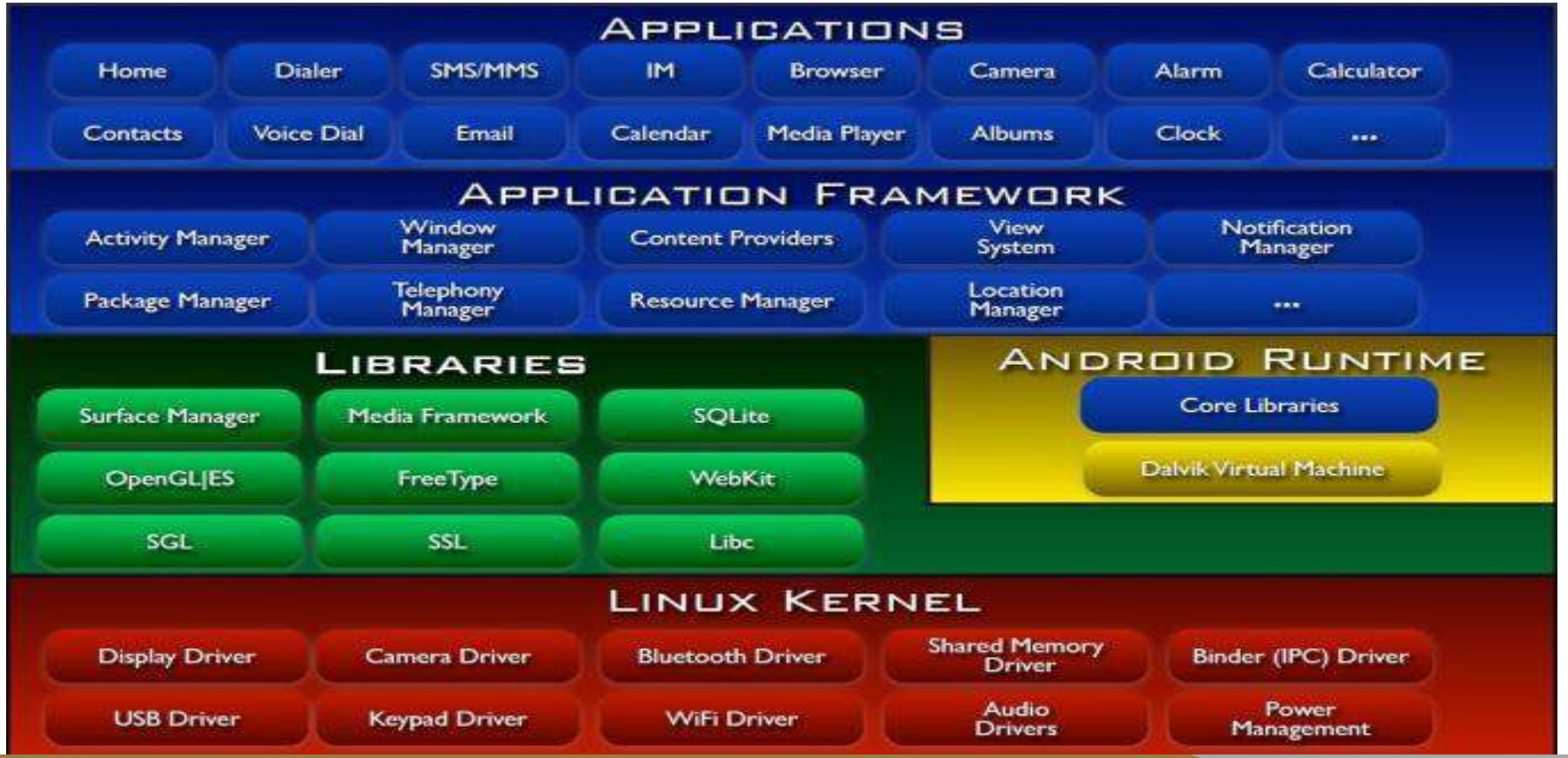
Architecture du système d'exploitation Android

L'architecture du système d'exploitation Android est conçue pour offrir un haut niveau d'abstraction par rapport au matériel sous-jacent et pour prendre en charge une large gamme d'appareils avec différentes configurations.

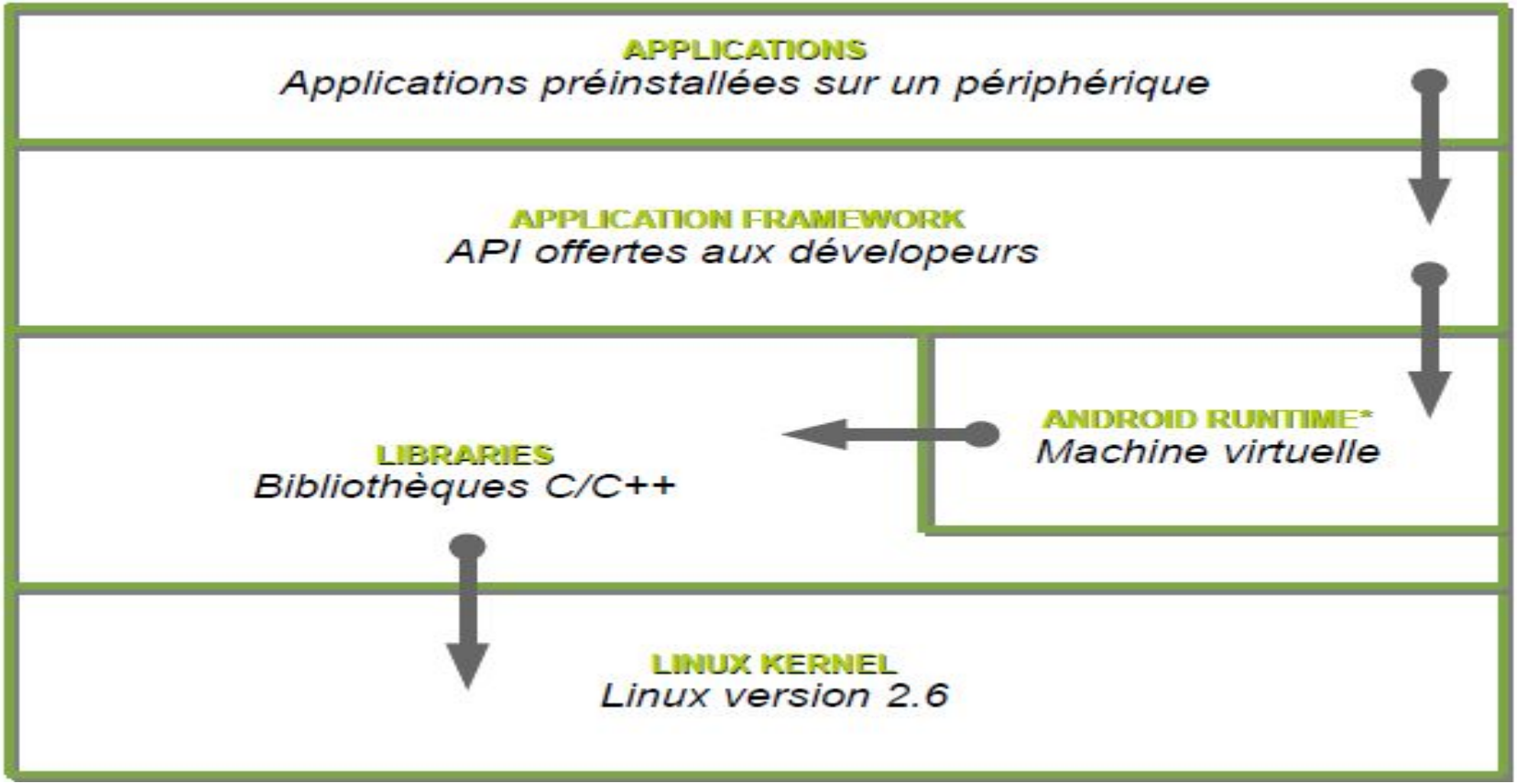
l'architecture d'Android se compose de plusieurs composants clés, notamment :

1. **Linux Kernel**
2. **Native Libraries**
3. **Android Runtime**
4. **Application Framework**
5. **Applications**

Architecture logicielle d'Android

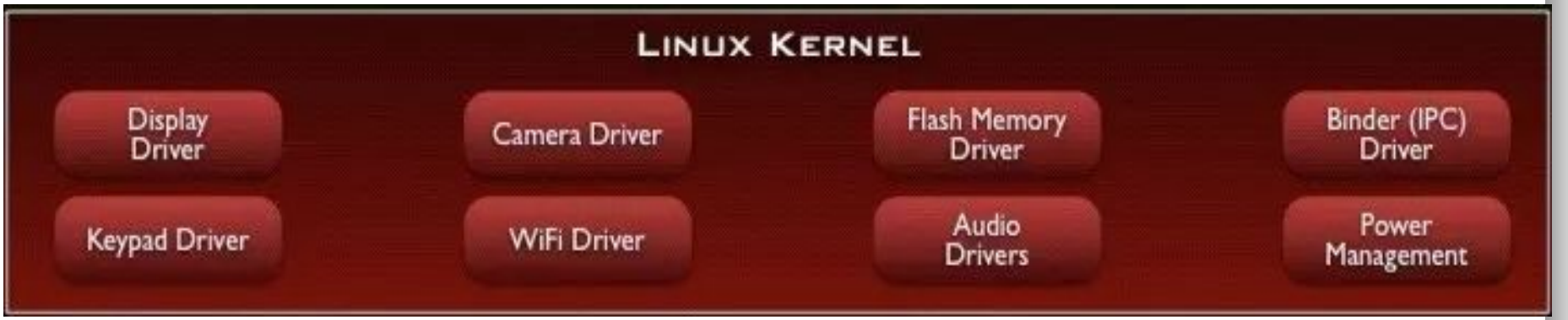


Architecture logicielle d'Android



Linux kernel

Le noyau Linux sert d'interface avec le matériel sous-jacent, faisant ainsi le pont entre le matériel et le logiciel. Il gère les processus, la mémoire, les droits des utilisateurs et l'accès au matériel. Ce niveau permet également d'accéder aux périphériques tels que la caméra, l'audio, le Bluetooth, etc. Il assure également la sécurité et la gestion de la mémoire.



Les librairies

- Android utilise des bibliothèques natives en C/C++. Elles fournissent **un accès direct aux ressources du système**. C'est en quelque sorte **la base** de l'ensemble du système au niveau supérieur.

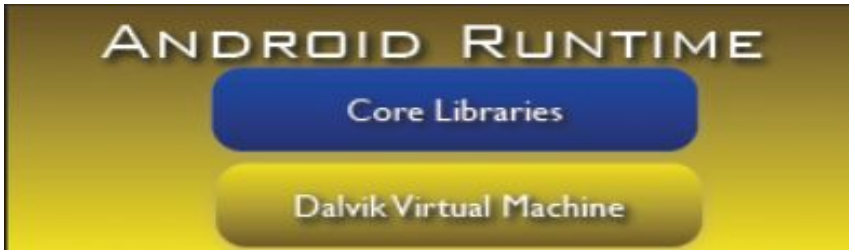
Les librairies

- **SQLite :** base de données relationnelle
- **FreeType :** moteur de rendu des polices de caractère
- **WebKit :** moteur de rendu HTML
- **Media Framework :** cette bibliothèque gère tout ce qui se rapporte aux données multimédia.
- **SGL :** gère le graphisme en 2D.
- **Surface Manager :** ensemble de fonctions permettant la gestion du dispositif d'affichage. Il permet de s'assurer que les pixels s'affichent bien à l'écran.
- **OpenSSL :** algorithme de cryptage des données
- **OpenGL/ES :** gère le graphisme en 3D

- **SQLite :** base de données relationnelle
- **FreeType :** moteur de rendu des polices de caractère
- **WebKit :** moteur de rendu HTML
- **Media Framework :** cette bibliothèque gère tout ce qui se rapporte aux données multimédia.
- **SGL :** gère le graphisme en 2D.
- **Surface Manager :** ensemble de fonctions permettant la gestion du dispositif d'affichage. Il permet de s'assurer que les pixels s'affichent bien à l'écran.
- **OpenSSL :** algorithme de cryptage des données
- **OpenGL/ES :** gère le graphisme en 3D

Android Runtime (l'environnement d’exécution)

- Cette section présente la Dalvik Virtual Machine, une machine virtuelle Java spécialement conçue pour Android.
- Dalvik VM: fournit une machine virtuelle alternative, adaptée aux limitations des appareils mobiles, permet l'exécution simultanée de plusieurs applications.
- Elle utilise des fonctionnalités Linux comme la gestion de la mémoire et le multithreading, typiques de Java, et assure que chaque application Android s'exécute dans son propre processus avec une instance dédiée de la VM.
- Apartir de la version Android5.0 (Lollipop), Dalvik a été remplacé par ART (AndroidRunTime)



Application Framework

- Le cadre d'application est un ensemble d'API de haut niveau que les développeurs utilisent pour créer leurs applications. Il offre une gamme complète de services, incluant la gestion des activités, des services, des fournisseurs de contenu, et des notifications. Le cadre inclut aussi des outils pour la gestion de l'interface utilisateur, comme le système d'affichage, qui permet de créer et de gérer des interfaces utilisateur de manière flexible et efficace.

Application Framework

- **Couche d'application :** le composant de plus haut niveau de l'architecture Android est l'ensemble des applications qui fonctionnent sur l'appareil. Ces applications sont développées à l'aide du Cadre d'Application et des bibliothèques de Base, et elles offrent une large gamme de fonctionnalités, incluant des jeux, des outils de productivité et des réseaux sociaux.

Android studio

Android Studio est un environnement de développement intégré (IDE) conçu spécifiquement pour le développement d'applications Android. Il a été développé par Google et est basé sur la plateforme IntelliJ IDEA. Parmi les principales fonctionnalités d'Android Studio, on trouve :

- **Éditeur de code** : l'éditeur de code offre la complétion automatique, la mise en surbrillance du code et la vérification des erreurs pour faciliter la programmation.
- **Éditeur de mise en page :** l'éditeur de mise en page offre une interface visuelle pour concevoir l'interface utilisateur des applications.
- **Émulateur :** Android Studio inclut un émulateur permettant aux développeurs de tester leurs applications sur un appareil virtuel.
- **Outils de débogage :** Android Studio propose des outils de débogage pour identifier et corriger les erreurs dans le code.

Android studio

- **Système de build Gradle :** Android Studio utilise Gradle pour la construction, le test et le déploiement des applications.
- **Plugins :** Android Studio prend en charge une large gamme de plugins pour étendre les fonctionnalités de l'IDE.
- **Environnement de développement intégré :** Android Studio fournit un environnement intégré pour écrire, tester et déployer des applications Android.

Android SDK

- **SDK** signifie **Kit de développement logiciel**. Il s'agit d'un ensemble d'outils de **développement logiciel** regroupés dans un seul paquet installable.
- Les SDK sont utilisés pour **développer** des applications pour une plateforme spécifique, comme Android ou iOS.
- Un SDK Android inclut des **bibliothèques, des outils, des exemples de code et des documentations nécessaires** au développement d'applications Android.
- Le SDK Android inclut les composants suivants :
  - **Émulateur Android :** un appareil virtuel qui fonctionne sur votre ordinateur pour tester votre application.
  - **Outils SDK Android :** un ensemble d'outils incluant ADB (Android Debug Bridge), qui permet de communiquer avec un émulateur ou un appareil, et AAPT (Android Asset Packaging Tool), qui emballe les ressources dans un APK.

Android SDK

- **Outils Plateforme Android :** un ensemble d'outils offrant des fonctionnalités supplémentaires pour le débogage et le chargement de votre application.
- **Bibliothèque de support Android :** un ensemble de bibliothèques de support pour intégrer la compatibilité dans votre application.
- **APIs Google :** un ensemble d'APIs permettant d'accéder aux services Google depuis votre application.
- **Code exemple et documentation :** Des exemples de code et de la documentation pour vous aider à démarrer le développement d'applications Android.

Anatomie de l'application Android

AndroidManifest.xml : décrit les Caractéristiques fondamentales de l'application et définit chacun de ses composants.

Java : contient les fichiers source .java pour le projet. Par défaut, il inclut un fichier source MainActivity.java

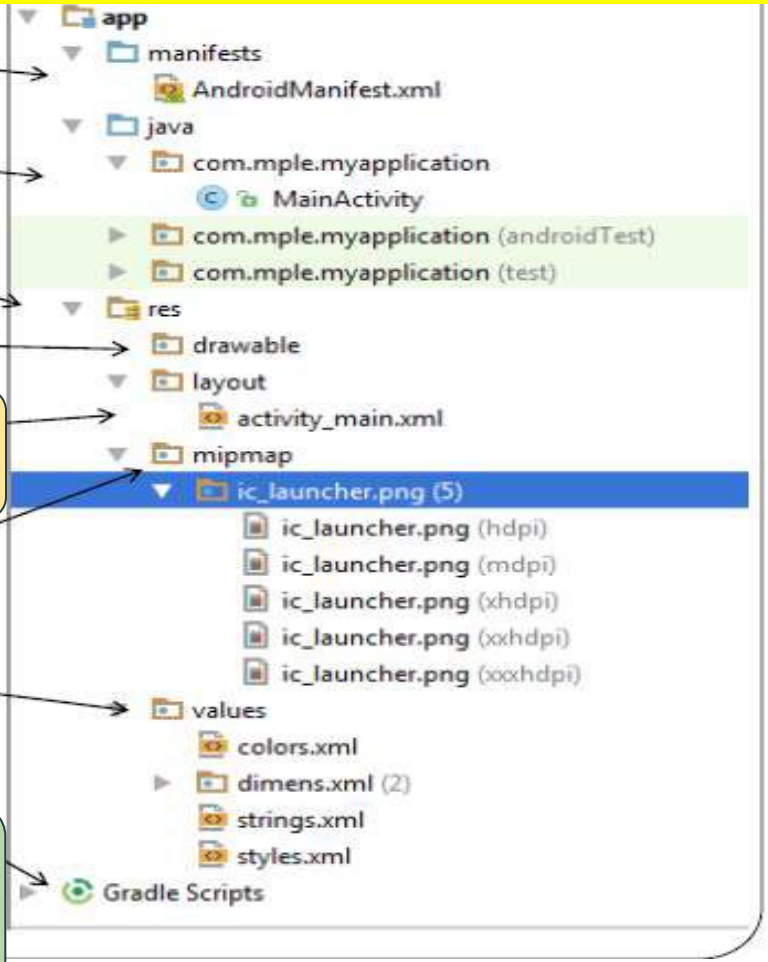
res/drawable : fichiers XML et images, icônes utilisés dans l'interface. répertoire pour les objets conçus pour les écrans haute densité.

Res/layout : un répertoire pour les fichiers qui définissent l'interface utilisateur de l'application.

mipmap : images, icônes utilisés dans l'interface

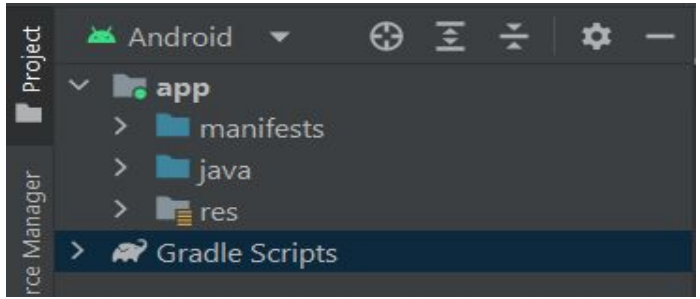
Values : un répertoire pour d'autres fichiers XML divers contenant une collection de ressources, telles que des définitions de chaînes et de couleurs.

radle scripts : l'outil de compilation du projet. Un fichier généré tomatiquement qui contient compileSdkVersion, buildToolsVersion, plicationId, minSdkVersion, targetSdkVersion, versionCode et rsionName.



Projet Android

Voici quelques-uns des répertoires et fichiers importants inclus dans un projet Android :



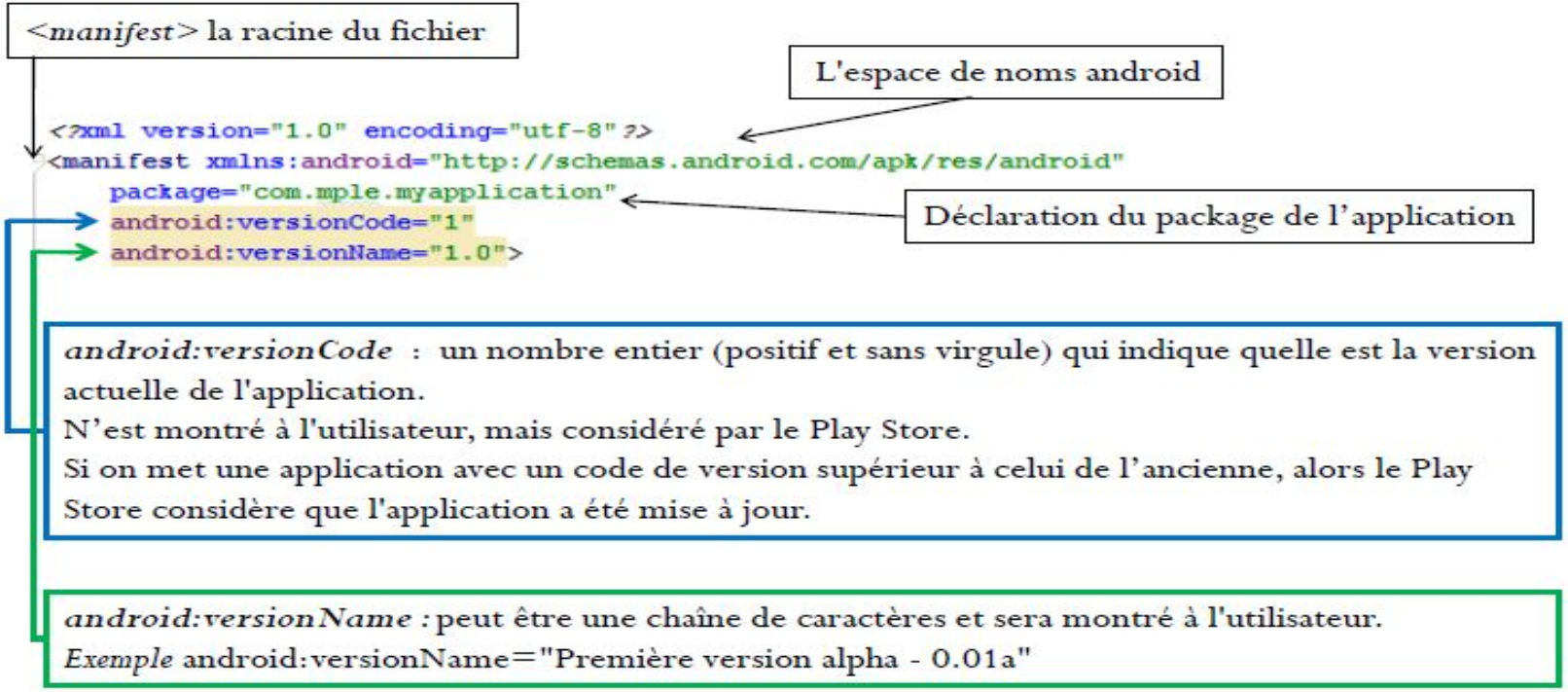
1. **Manifest :** le fichier AndroidManifest.xml est le fichier de configuration principal d'une application Android. Il contient des informations sur l'application, telles que le nom de l'application, le nom du paquet, les permissions requises et les activités qui composent l'application.
2. **Java :** ce répertoire contient les fichiers de code source Java de l'application. Le fichier de l'activité principale et toutes les autres classes Java de l'application s'y trouvent également.
3. **Res :** le répertoire res contient toutes les ressources non codées utilisées par l'application, telles que les images, les fichiers de mise en page, les chaînes de caractères et les styles.
4. **Gradle scripts :** les fichiers build.gradle contiennent les paramètres du projet et de l'application pour le système de construction Gradle. Ils servent à configurer le processus de construction de l'application, y compris les dépendances et les options de compilation.

Fichier Manifest

- Le fichier **AndroidManifest.xml** est un composant crucial d'un projet Android qui fournit des informations **essentielles** sur l'application au système Android.
- Il contient des détails sur le **nom de paquet** de l'application, **sa version**, ses composants tels que **les activités**, **les services**, **les récepteurs de diffusion** et **les fournisseurs de contenu**, **les permissions requises** par l'application et d'autres métadonnées.
- Ce fichier manifeste est **nécessaire** pour chaque application Android et agit comme un plan indiquant comment l'application **interagira** avec le système d'exploitation Android. Il doit se trouver à **la racine** du dossier source du projet et est automatiquement créé lors de la création d'un nouveau projet Android dans Android Studio.
- En résumé, le fichier manifeste est **un élément essentiel** de toute application Android, détaillant sa **structure** et définissant son **interaction** avec le système d'exploitation.

Fichier Manifest

Le fichier AndroidManifest.xml



Fichier Manifest

La version d'Android (API 15) ou supérieure pour pouvoir utiliser cette application (cette application ne sera proposée sur Google Play uniquement si un utilisateur utilise cette version d'Android ou une version supérieure).

```
<uses-sdk android:minSdkVersion="15"
          android:targetSdkVersion="23" />
```

La version à partir de laquelle on pourra exploiter à fond l'application

Fichier Manifest

Il décrit les attributs et les composants de l'application.
Par défaut, l'application n'a qu'un composant, l'activité principale.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="My Application"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

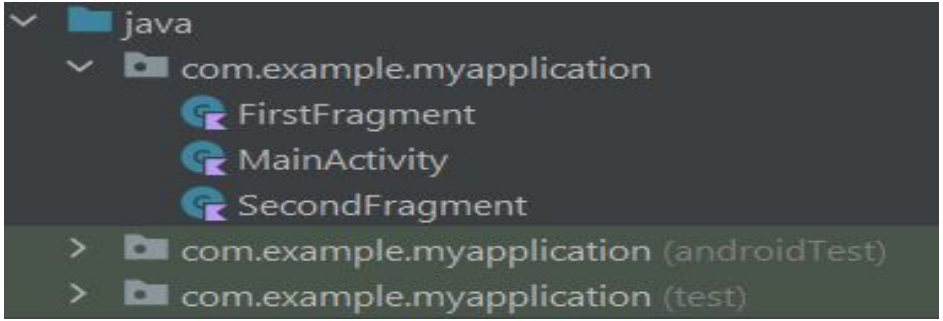
Définition de l'icône, du nom et du thème de l'application

Déclaration des activités

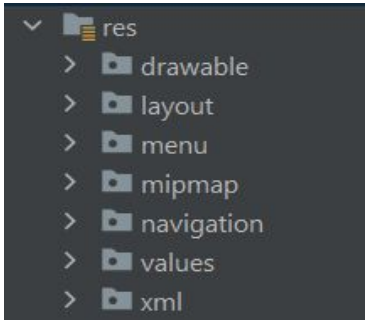
Indique que l'activité est le point d'entrée de l'application et doit s'afficher dans le lanceur d'applications

Dossier Java

- Le dossier **"java"** dans un projet Android contient tout le code source Java de l'application. Cela inclut toutes les classes et le code utilisé pour construire les fonctionnalités de l'application. C'est ici que les développeurs écrivent la majorité du code de l'application, y compris la définition de l'interface utilisateur, la gestion des données et du stockage, le traitement des événements d'entrée et la mise en œuvre de la logique de l'application.
- Le code rédigé dans le dossier "java" est compilé en bytecode qui s'exécute sur l'appareil Android, permettant ainsi à l'application de fonctionner et d'offrir ses fonctionnalités aux utilisateurs.



Dossier Res



Le dossier res dans un projet Android contient toutes les ressources utilisées par l'application, les types courants de ressources que vous
trouverez dans le dossier res :
telles que les fichiers de mise en page, les images, les chaînes de caractères et plus encore. Voici que

- **layout** : contient les fichiers XML qui définissent l'interface utilisateur de l'application.
- **drawable** : contient les images et autres ressources utilisées dans l'application.
- **values** : Contient les fichiers XML qui définissent diverses valeurs utilisées dans l'application, comme les chaînes de caractères, les couleurs et les dimensions.
- **anim** : contient les fichiers XML qui définissent les animations utilisées dans l'application.
- **menu** : contient les fichiers XML qui définissent les menus utilisés dans l'application.

Ces exemples illustrent quelques-unes des ressources que vous pourriez trouver dans le dossier res. Le contenu spécifique de ce dossier peut varier selon les besoins de votre application.