APPLICATIONS MOBILES

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** : chaines 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 ressource et prête pour le déploiement

Android Applications Design

APPLICATION **DESIGN**:

- **GUI** Definition
- **Events** Management
- Application **Data** Management
- **Background** Operations
- **User** Notifications

ANDROID APPLICATION

- Les applications sont programmées en Java
- L’ ensemble (code Java, données, fichiers de ressources et fichier XML) est compilé par l’ Android SDK et regroupé dans un paquetage Android : .apk
- chaque application tourne sur son propre processus Linux
- Application : ensemble de composants, un fichier Manisfest.xml et Ressources
  - Composants :
    - Activités
    - Services
    - Content providers
    - Broadcast receivers

Les composants d'une App Android

Activities

1. **Offre** Une Interface utilisateur
2. Généralement représente Un seul écran
3. Peut contenir un/plusieurs Vues
4. Etend la classe Activity

Services

1. **Pas** d'Interface utilisateur
2. S'excute en **Arriere Plan**
3. Etend la classe **Service**

Application= ensemble de composants Android

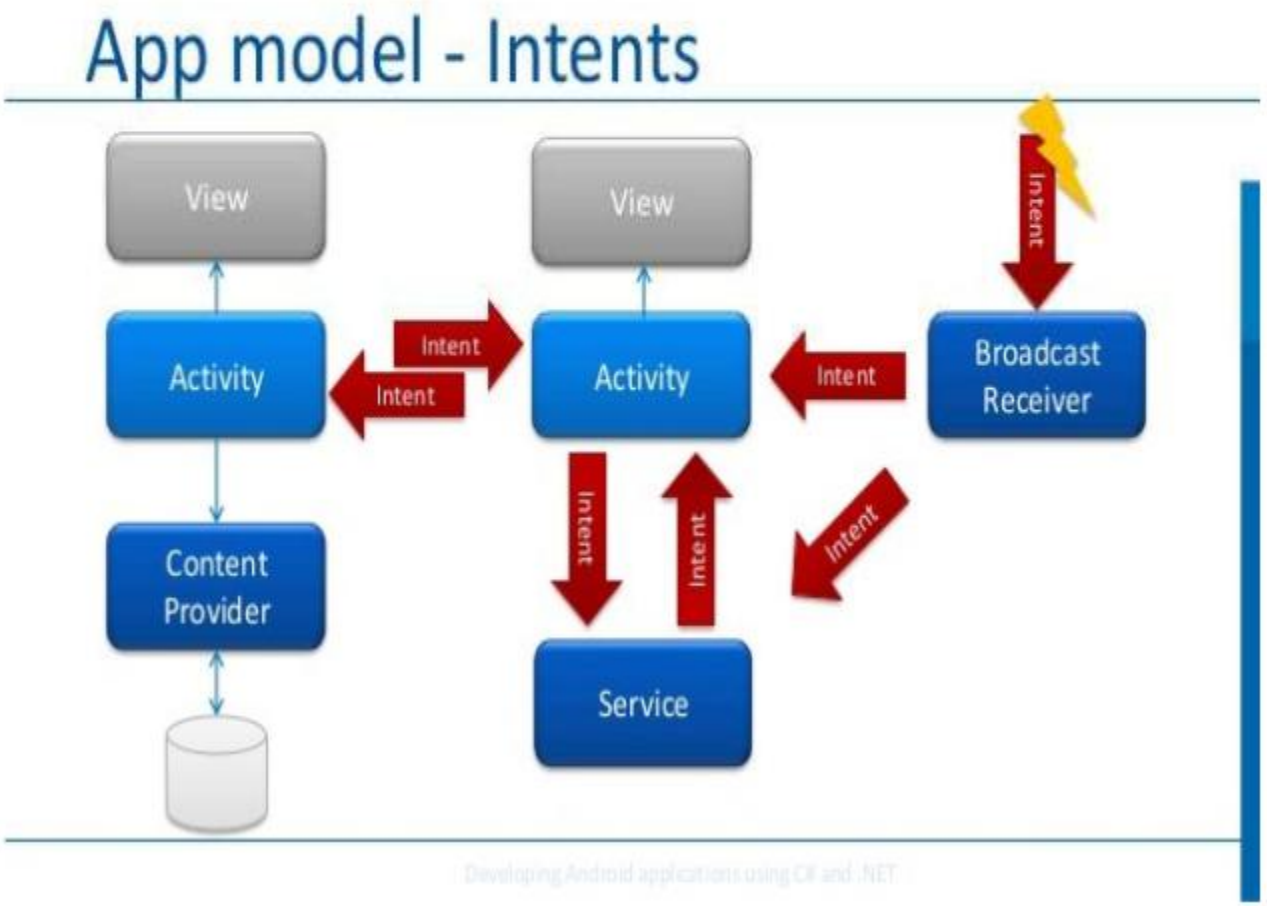
Intent/Broadcast Receiver

1. **Reçoit et répond aux broadcast** Intents
2. Pas d'IU mais peut démarrer une Activity
3. Etend la classe BroadcastReceiver

Content Provider

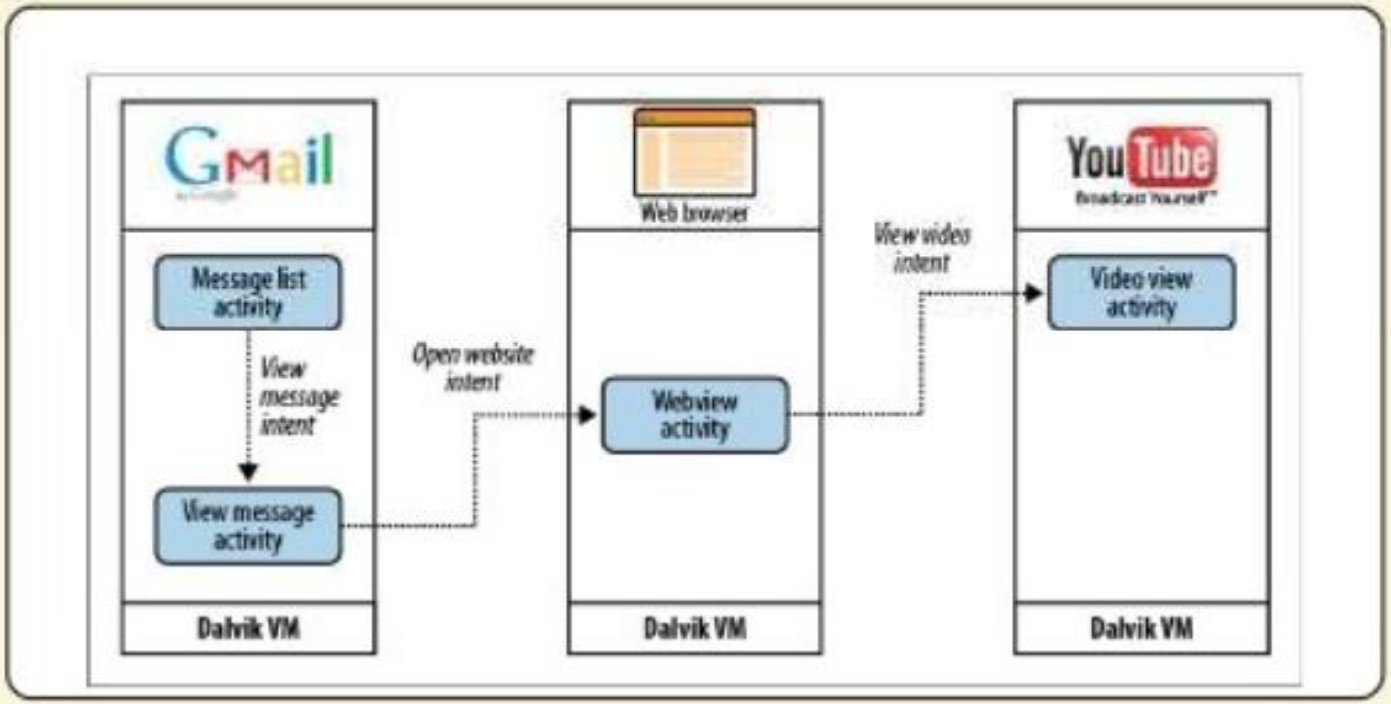
1. Données **d'une application** accessibles à d'autres Apps
2. Données sauvegardées dans la base de données SQLite
3. Extend la classe ContentProvider

Les composants d'une App Android



Exemples

Intents



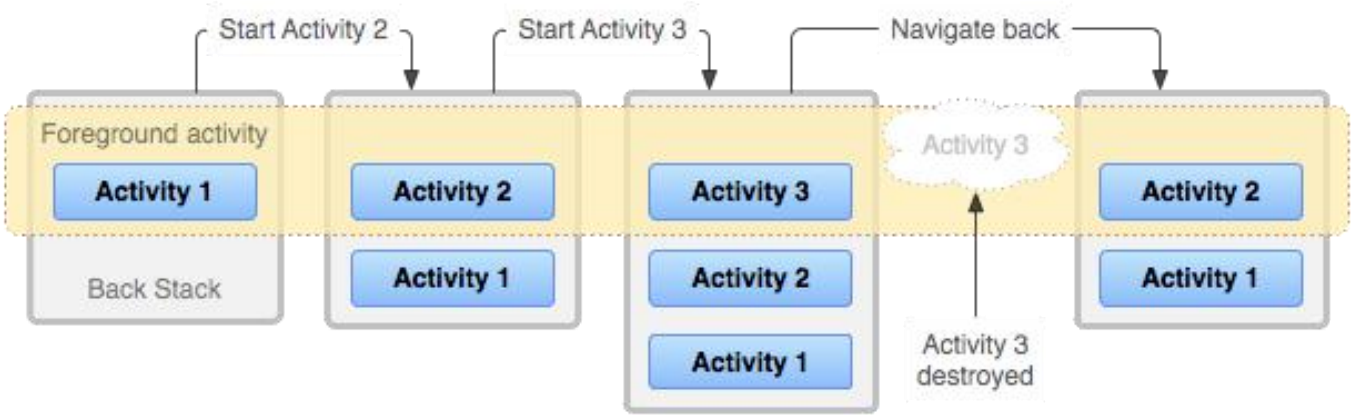
Used to start activities, start/stop services, or send broadcasts

ANDROID APPLICATION ACTIVITÉS

- Sous classe de **android.app.Activity**
- Une activité : écran que voit l'utilisateur + traitements réactifs
- Exemple : email application
  - Quand l'utilisateur ouvre l'application et voit la liste des email c'est une activité
  - Quand il clique sur un email et l'ouvre c'est une autre activité
  - Quand il essaie d'écrire un nouvel email, c'est une autre activité aussi
- Pile d'activités
- Gérer le cycle de vie à l'aide des méthodes onStart(), onPause()…etc.

ANDROID APPLICATION ACTIVITÉS

- Pile d’activités

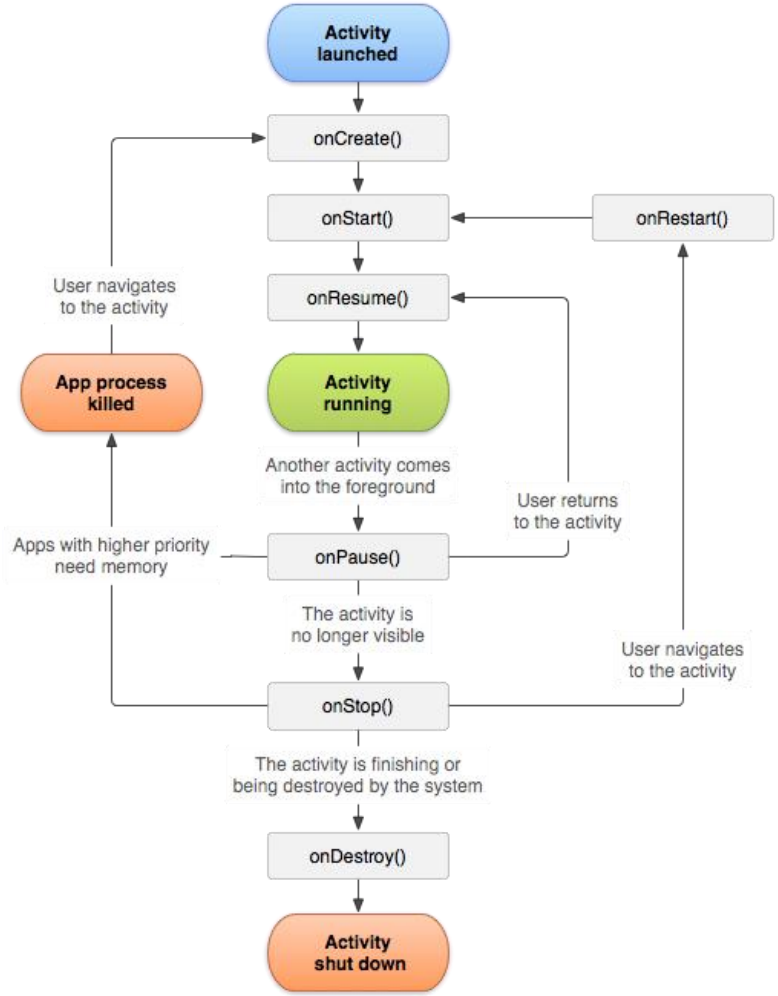


ANDROID APPLICATION
ACTIVITÉS CYCLE DE VIE

- Hébergée sur système embarqué
- → cycle de vie semblable à celui d'une application Java ME
  - Active → suspendue
  - Suspendue → active
  - Suspendue → détruite

```
public class Main extends Activity {
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.acceuil); }
  protected void onDestroy() {
    super.onDestroy(); }
  protected void onPause() {
    super.onPause(); }
  protected void onResume() {
    super.onResume(); }
  protected void onStart() {
    super.onStart(); }
  protected void onStop() {
    super.onStop(); } }
```

ANDROID APPLICATION
CYCLE DE VIE D' UNE ACTIVITÉ



ANDROID APPLICATION
SAUVEGARDE D' INTERFACES D' ACTIVITÉS

- L' objet Bundle permet de restaurer les interfaces d' une activité déchargée de la mémoire
- Bundle passé en paramètre à la méthode onCreate()
- Exemple :
  - lorsque l' on appuie sur la touche Home
  - Android peut décharger les éléments graphiques (gagner des ressources)
  - Appui long sur Home : l' app peut avoir perdu les valeurs saisies sur zone de texte …

ANDROID APPLICATION SERVICES

- Sous classe de **android.app.Service**
- Un service : tâche de fond invisible
- Ne fournit pas d'interface graphique
- Un service tourne indépendamment de l'activité qui l'a crée
- Ex : lecture de musique, téléchargement en cours…etc.

ANDROID APPLICATION CONTENT PROVIDER

- Sous classe de **android.content.ContentProvider**
- Content provider ou fournisseur de contenu
- Utiliser pour stoker et récupérer les données et les rendre accessibles par toutes les applications
- Partage de données entre applications
- On peut créer un fournisseur de contenus pour des données qu'on veut partager
- Exemple : la liste des contacts, la galerie photos

ANDROID APPLICATION BROADCAST RECEIVERS

- Sous classe de **android.content.BroadcastReceiver**
- Écoute d'événements globaux ou spécifiques
- Un récepteur d'information est un composant à l'écoute d'information
- Il indique le type d'information qui l'intéresse et pour laquelle il se mettra en écoute
- Ex : batterie faible, réseau WI-FI connecté, appel entrant… etc.
- Sert de point d'entrée pour les autres composants
- L'application réceptrice d'informations (c'est à dire possédant un récepteur d'informations) n'a pas besoin d'être lancée.
- Un récepteur n'est pas une IHM mais peut en lancer une (éventuellement petite : une barre de notification)
- Les informations sont des **Intent**

ANDROID APPLICATION INTENTS

- Sous classe de **android.content.Intent**
- Outil de communication entre les activités, services et récepteurs d'information.
- Permet à une activité de formuler publiquement une demande,
- Un événement (intent) est une "intention" à faire quelque chose contenant des informations destinées à un composant Android
- Ex : prendre une photo, récupérer un numéro, …etc.
  ou lancer une autre activité déterminée.

ANDROID APPLICATION
INTENTS EXPLICITES

- Dans l'intent explicite :
  - Le composant ciblé est explicitement cité ou précisé ou donné
- Exemple :
  - Activation d'une certaine activité à partir d'une autre

ANDROID APPLICATION
INTENTS IMPLICITES

- Dans l'intent implicite :
  - le composant à activer n'est pas précis car :
    - L'intent implicite annonce sous forme de message général le travail demandé ou l'action visée
    - N'importe quel application sur le système qui est capable de réaliser la tâche visée peut se proposer à prendre en charge cette action
    - L'utilisateur dans ce cas aura un choix à faire entre ces applications qu'ont répondues
- Exemple :
  - lecture d'une video ou …

ANDROID APPLICATION
FICHIER MANIFEST (XML)

- Toutes les applications doivent comporter un fichier nommé exactement AndroidManisfest.xml dans le répertoire principale
- Fournit au système Andoid des informations concernant l'application
- Décrit les composants utilisés dans l'application
- Décrit aussi les permissions nécessaires pour le fonctionnement de l'application (accès internet, accès en lecture/écriture aux données partagées)
- Donne le numéro minimum de la version API qui peut être utilisée
- Parmi les composants, seuls les récepteurs d'évènements (BroadcastReceiver) ne sont pas forcément dans le manifeste. Les autres (Activity, Service, ContentProvider) doivent l'être sinon ils ne seront jamais lancés quel que soit le code écrit !

ANDROID APPLICATION
FICHIER MANIFEST (XML)

```xml
<?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>
```

ANDROID APPLICATION RESSOURCES

- Utilisées dans le code au travers la classe statique R
- R est régénérée automatiquement par ADT, à chaque modification du projet
- Toutes les ressources sont accessibles dès que:
  - Elle sont déclarées dans AndroidManifest.xml
  - OU que le fichier associé est placé dans le répertoire adéquat
- Méthode d'accès:

```
android.R.type_ressource.nom_ressource
```

Une méthode spécifique pour les objets graphiques permet de les récupérer à partir de leur id, ce qui permet d'agir sur ces instances même si elles ont été créées via leur définition XML:

```
TextView texte = (TextView)findViewById(R.id.le_texte);
texte.setText("Here we go !");
```

ANDROID APPLICATION RESSOURCES

Ressource	Répertoire	Descritpion
Dessins (drawables)	/res/drawables	Images (fichiers PNG ou JPEG par exemple) ou fichiers XML qui décrivent un dessin (Drawable).
Valeurs simples	/res/values	Permet de définir des chaînes de caractères, des couleurs, des dimensions, des styles et des tableaux statiques de chaînes ou d'entiers via des fichiers XML. Chaque type est stocké dans un fichier séparé
Mises en page (layouts)	/res/layout	Fichiers XML avec les mises en page utilisées pour définir l'interface utilisateur des activités et fragments. Styles et Thèmes /res/values Fichiers qui définisse

ANDROID APPLICATION RESSOURCES (EXEMPLE)

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Test</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

    <string-array name="operationsystems">
        <item>Ubuntu</item>
        <item>Android</item>
        <item>Microsoft Windows</item>
    </string-array>

    <color name="red">#ffff0000</color>

    <dimen name="mymargin">10dp</dimen>

</resources>
```

- Exemple : values.xml dans /res/values est un fichier XML qui définit quelques constantes de chaîne, un tableau de chaînes, une couleur et une dimension.

ANDROID APPLICATION RESSOURCES (CHAINES CONSTANTES)

- Chaines constantes : nom de l'application, labels des boutons, nom de menu…etc.
- Situées dans : res/values/strings.xml
- L'externalisation des chaines permettra l'internationalisation de l'application

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

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

```
MyProject/
  res/
     values/
          strings.xml
     values-es/
          strings.xml
     values-fr/
          strings.xml
```

ANDROID APPLICATION
AUTRES RESSOURCES

- Les tableaux
- Les menus
- Les images (R,drawable)
- Les Couleurs (R.color)

RSOURCES

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