


Chapitre V:

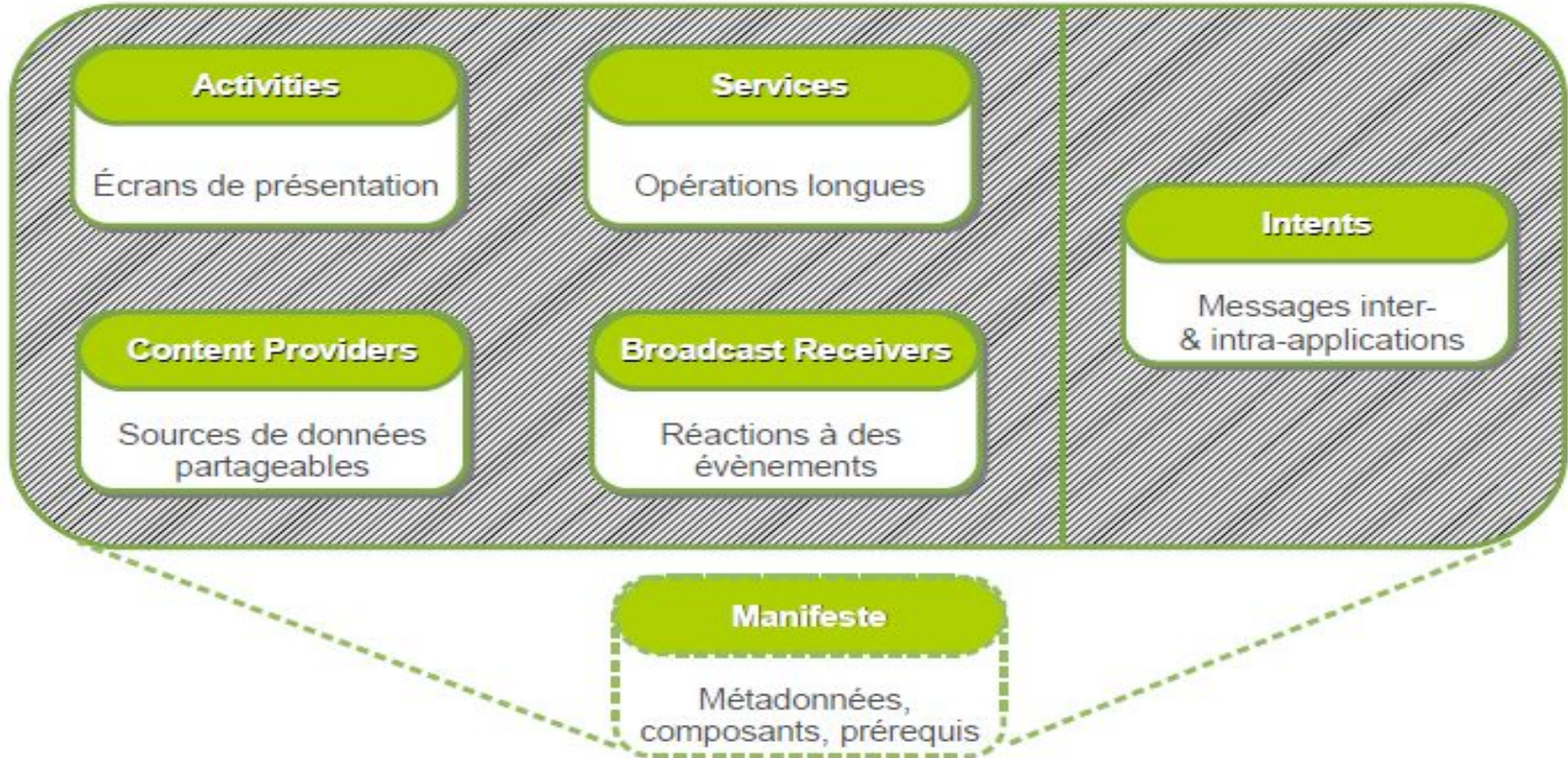
Les composants d'une application mobile



Introduction

- Une application Android est composée de plusieurs composants fondamentaux qui interagissent pour offrir une expérience complète à l'utilisateur.
- Ces composants sont gérés par le **système Android** et communiquent via un mécanisme appelé **Intent**.
- Les **composants d'application** (tels que les activités, services, etc.) sont les **éléments de base** d'une application Android.
- Ces composants sont **faiblement couplés**, c'est-à-dire qu'ils fonctionnent de manière indépendante, mais sont **déclarés et connectés via le fichier `AndroidManifest.xml`**, qui définit leur existence et la manière dont ils interagissent entre eux.

Composants applicatifs



Activité : définition

Une activité est le composant fondamental d'une application Android qui représente une **interface utilisateur**.

Elle permet à l'utilisateur d'interagir avec l'application via une **interface spécifique**.

Une application comportant plusieurs écrans possédera plus d'activités.

Chaque activité est dédiée à un objectif précis, tel que :

- Afficher des données (ex. : liste de contacts)
- Modifier des informations (ex. : profil utilisateur)
- Exécuter une action (ex. : lire une vidéo)

Méthodes clés : `onCreate()`, `onStart()`, `onResume()`, `onPause()`

Service

- Un service s'exécute en arrière-plan pour effectuer des opérations de longue durée ou pour exécuter des tâches pour des processeurs distants.
- Sert à effectuer des opérations ou des calculs en dehors de l'interaction utilisateur
- Contrairement à une activité, un service ne propose pas d'interface utilisateur ; il exécute uniquement en arrière-plan avec l'entrée de l'utilisateur .
- **Un service \approx une activité sans GUI**
- **Exemples :**
 - Lecture de musique en tâche de fond
 - Téléchargement de fichier
 - Synchronisation automatique

Service

Deux types de services :

Service Local :

- 1) S'exécute **dans le même processus** que l'application.
- 2) Idéal pour les tâches internes qui ne nécessitent pas de communication avec d'autres applications.
- 3) Exemple : gestion des notifications, téléchargement en arrière-plan.

Service

Deux types de services :

Service distant (Inter-Process Communication – IPC) :

- 1) S'exécute dans **un processus indépendant** de celui de l'application.
- 2) Utilisé pour interagir avec d'autres applications ou pour fournir un service à d'autres processus.
- 3) **Exemple** : Un service de musique qui peut être contrôlé par plusieurs applications.

Tous les services héritent de la classe **Android.app.Service**.

Récepteur de diffusion (Broadcast receivers)

- Gérer la communication entre le système d'exploitation Android et les applications.
- Permettre la communication entre les composants des applications Android à travers l'envoi des messages via l'objet : **Intent**
 - **Ex :** communication entre deux activités.
- Les récepteurs de diffusion répondent simplement aux messages de diffusion provenant d'autres applications ou du système.
- Ne nécessite pas une interface graphique
- Un broadcast receiver est une classe qui étend `Android. Content. BroadcastReceiver`
- Un récepteur s'abonner/désabonner via le fichier manifest ou par programme.

Récepteur de diffusion (Broadcast receivers)

Les **récepteurs de diffusion** permettent à une application de **réagir à des messages (ou événements)** envoyés par le système Android ou par d'autres applications.

- **System-defined** : Notifications émises par le système, comme :
 - Batterie faible (**ACTION_BATTERY_LOW**)
 - Réception d'un SMS (**SMS_RECEIVED**)
 - Changement de connectivité (Wi-Fi activé/désactivé)
 - Appareil démarré (**BOOT_COMPLETED**)
- **User-defined** : Notifications spécifiques aux besoins de l'utilisateur, comme :
 - Notification lorsqu'un **solde bancaire est négatif**
 - Alerte après une **commande validée**

Fournisseur de contenu (Content provider)

- Les Content Providers facilitent l'accès aux données stockées dans une application, comme des bases de données SQLite ou des fichiers, en les rendant accessibles à d'autres applications.
- Ils constituent **le seul moyen sécurisé** de partager des données entre différentes applications Android.
- Lorsqu'une application souhaite accéder à des données, elle utilise la classe **ContentResolver**, qui agit comme un intermédiaire pour gérer les opérations de lecture, d'écriture et de mise à jour des données.
- Un content provider est une classe qui étend **Android. Content.ContentProvider**
- **Exemples d'usage :**
 - Accès aux contacts, photos, messages

Intent

- Faciliter la communication entre les différentes applications et composants, comme les Activités ou Services.
- **Invoquer** d'autres activités ou services.
- **Partager des fonctionnalités** pour réutiliser du code entre applications.
- **Exemple**
 - Utiliser l'**appareil photo** pour prendre une photo.
 - Ouvrir une page Web avec le **navigateur** via une URL.
 - Envoyer un e-mail avec l'application **Email**.