

---

# Écosystème Android

---

 [Macha DA COSTA](#)

 [@MachaDaCosta](#)



# ABC d'Android

---

- A. Plateforme Android
- B. Environnement de Développement
- C. Principes de Programmation

# A. Plateforme Android

---

 Bienvenue dans le monde d'Android

② Pourquoi choisir Android ?

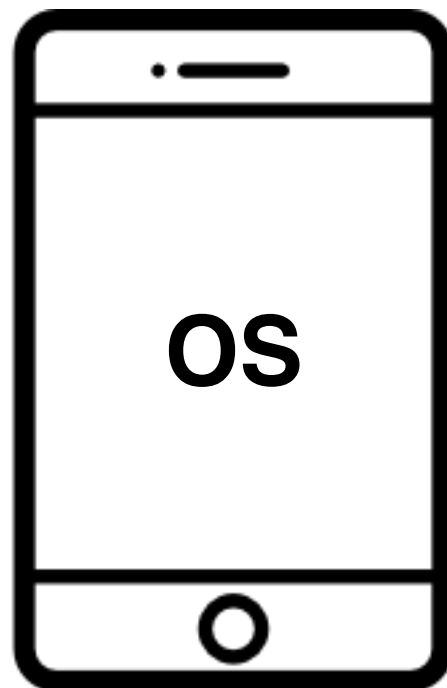
 Historique

 Chemin d'une Application

# A. Plateforme Android

---

 Bienvenue dans le monde d'Android



# A. Plateforme Android

---

 Bienvenue dans le monde d'Android

Open Handset Alliance

Google

NOKIA



Wiko



htc

SONY

QUALCOMM®



# A. Plateforme Android

---

 Bienvenue dans le monde d'Android

Couche logicielle

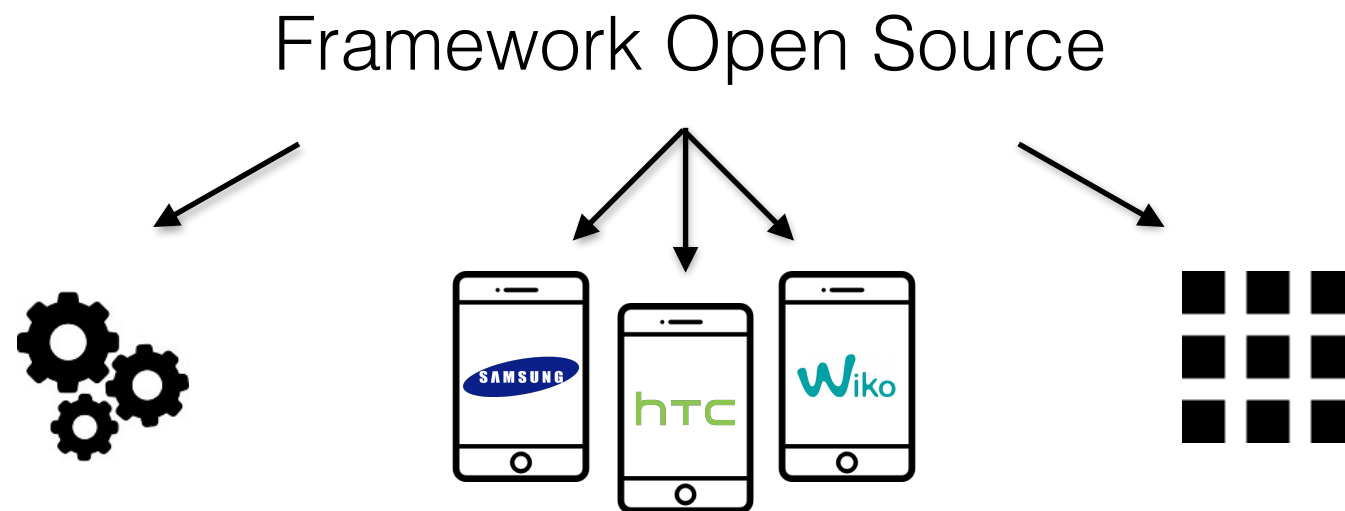
—  
Couche matérielle



# A. Plateforme Android

---

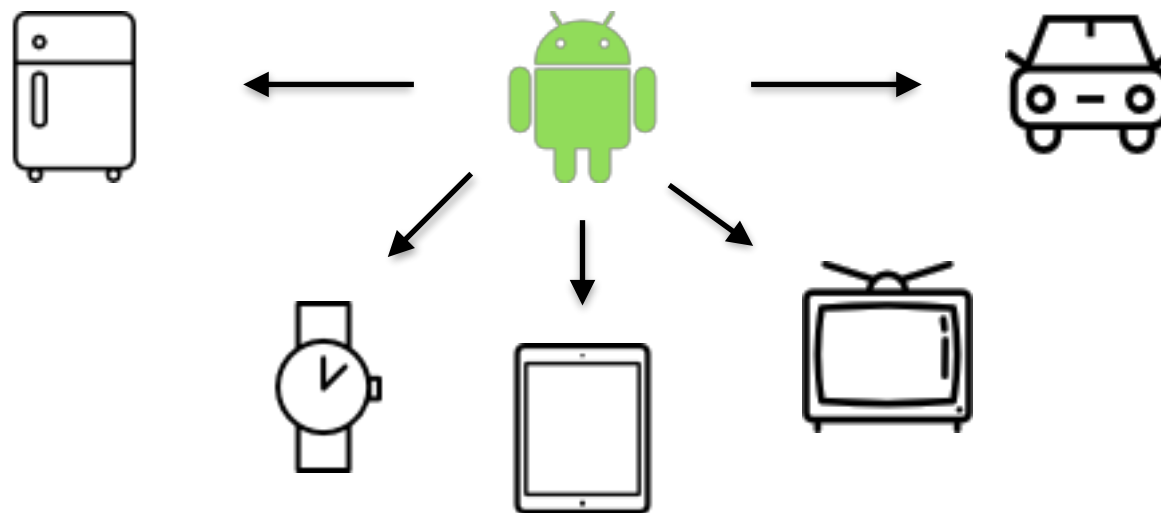
 Bienvenue dans le monde d'Android



# A. Plateforme Android

---

 Bienvenue dans le monde d'Android





A.

# Plateforme Android



# A. Plateforme Android

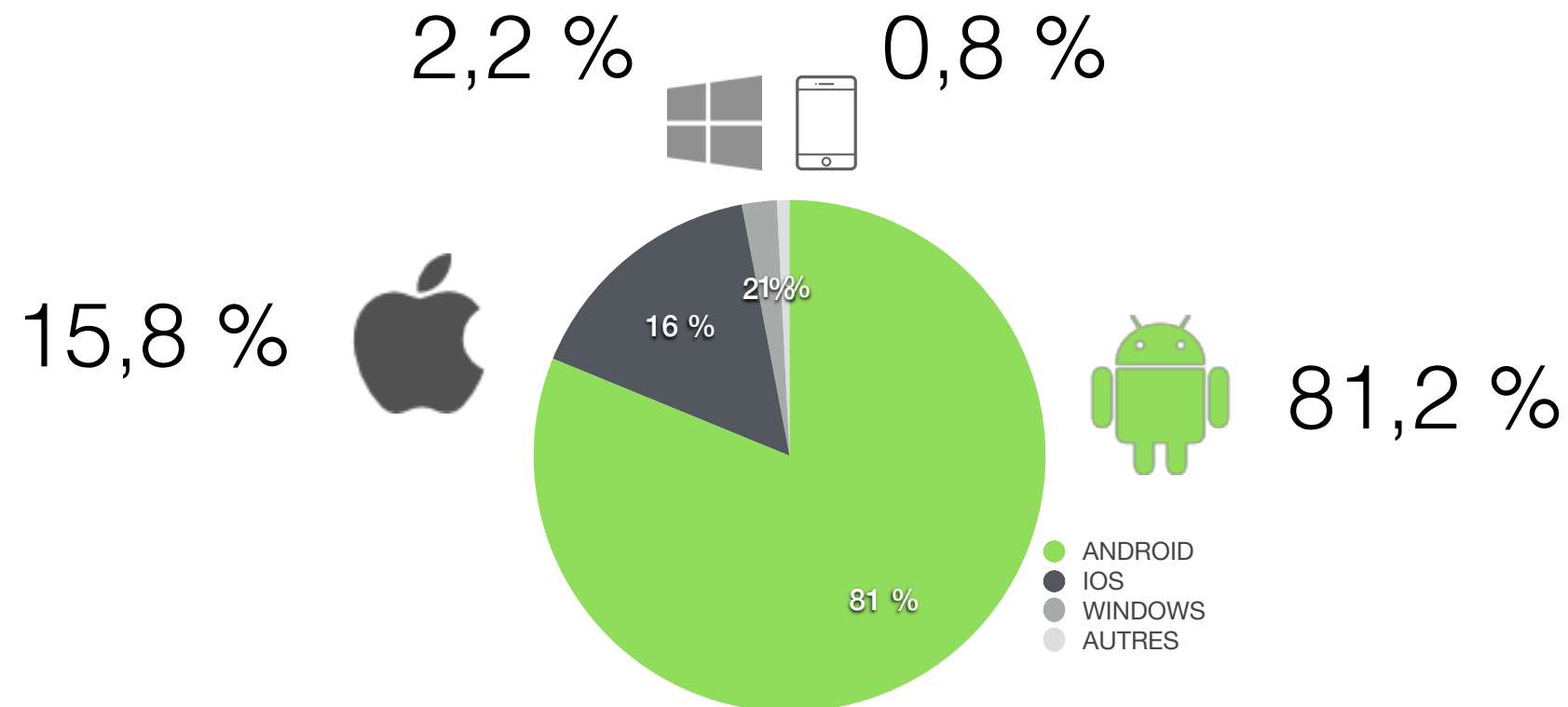
---

② Pourquoi choisir Android ?



# A. Plateforme Android

② Pourquoi choisir Android ?



# A. Plateforme Android

## ② Pourquoi choisir Android ?



Region	2015* Shipment Volumes	2015* Market Share	2015* YoY Growth	2019* Shipment Volumes	2019* Market Share	2019* YoY Growth	5 year CAGR
Android	1,161.1	81.2%	9.5%	1,538.1	82.6%	4.8%	7.7%
iOS	226.0	15.8%	17.3%	263.4	14.1%	3.1%	6.5%
Windows Phone	31.3	2.2%	-10.2%	43.6	2.3%	11.4%	4.5%
Others	11.3	0.8%	-16.8%	17.1	0.9%	6.5%	4.7%
<b>TOTAL</b>	<b>1,429.8</b>	<b>100.0%</b>	<b>9.8%</b>	<b>1,862.3</b>	<b>100.0%</b>	<b>4.7%</b>	<b>7.4%</b>

Source: IDC Worldwide Quarterly Mobile Phone Tracker, December 2, 2015.

\* Forecast data Volumes en millions

Évolution des OS Mobiles dans le Monde, Déc. 2015 - Source : IDC

# A. Plateforme Android

---

② Pourquoi choisir Android ?



38.9%



30.9%



30.2%

# A. Plateforme Android

---

② Pourquoi choisir Android ?



+ Portabilité

+ Compétences Web

+ Vitesse de développement



+ Ergonomie OS

+ Rendu optimal

+ Accès matériels

# A. Plateforme Android

---

## Historique

- Octobre 2003** : Startup Android à Palo Alto
- Août 2005** : Rachat d'Android par Google
- Novembre 2007** : Annonce officielle OHA
- Septembre 2008** : Premier téléphone Android
- Avril 2009** : Première pâtisserie Android Cupcake

# A. Plateforme Android






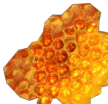







## Historique





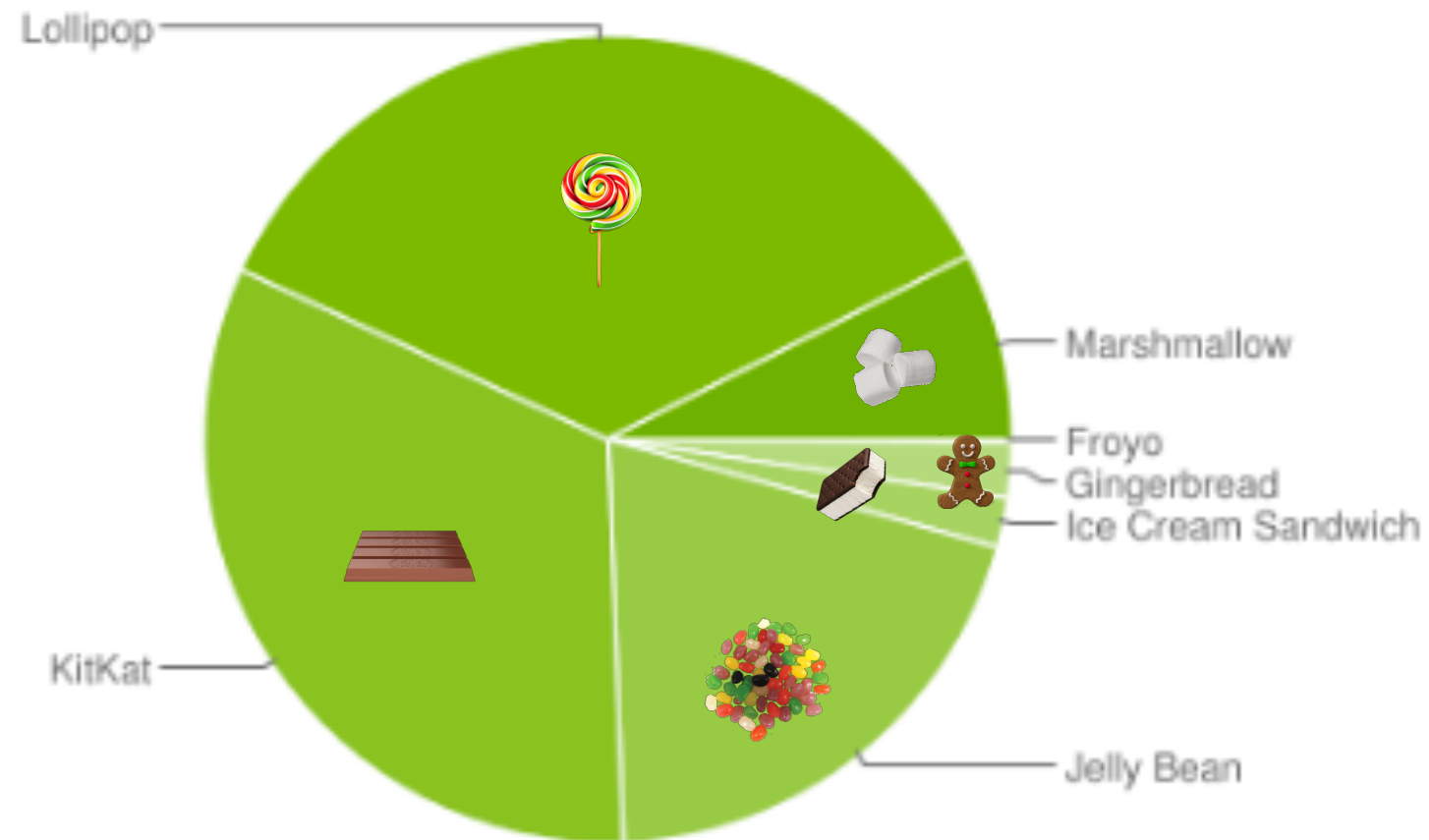
# A. Plateforme Android

## Historique

			Version	API	Répartition
	<b>Avril 2009</b>	: Cupcake	1.5	3	0,0 %
	<b>Septembre 2009</b>	: Donut	1.6	4	0,0 %
	<b>Octobre 2009</b>	: Eclair	2.1	7	0,0 %
	<b>Mai 2010</b>	: Froyo	2.2	8	0,0 %
	<b>Décembre 2010</b>	: Gingerbread	2.3	10	0,8 %
	<b>Février 2011</b>	: Honeycomb	3.X	13	0,0 %
	<b>Octobre 2011</b>	: Ice Cream Sandwich	4.0	15	0,8 %
	<b>Juin 2012</b>	: Jelly Bean	4.1	18	6,8 %
	<b>Octobre 2013</b>	: KitKat	4.4	20	18,1 %
	<b>Octobre 2014</b>	: Lollipop	5.1	21	30,8 %
	<b>Octobre 2015</b>	: Marshmallow	6.0	23	31,2 %
	<b>Octobre 2016</b>	: Nougat	7.0	24	8,9 %
	<b>Septembre 2017</b>	: Oréo	8.0	26	

# A. Plateforme Android

## 📅 Historique



Répartition des versions, Mai 2016 - *developer.android*

# A. Plateforme Android

---

 Chemin d'une Application

 Phase de Conception

 Phase de Développement

 Phase de Lancement

 Phase de Maintenance

# A. Plateforme Android

---

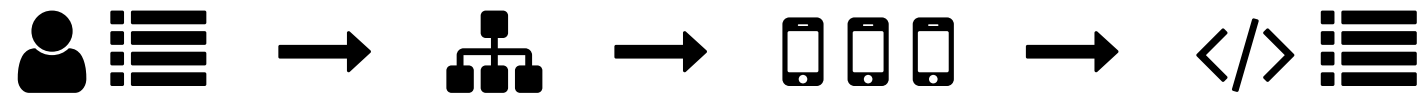
↻ Chemin d'une Application



# A. Plateforme Android

---

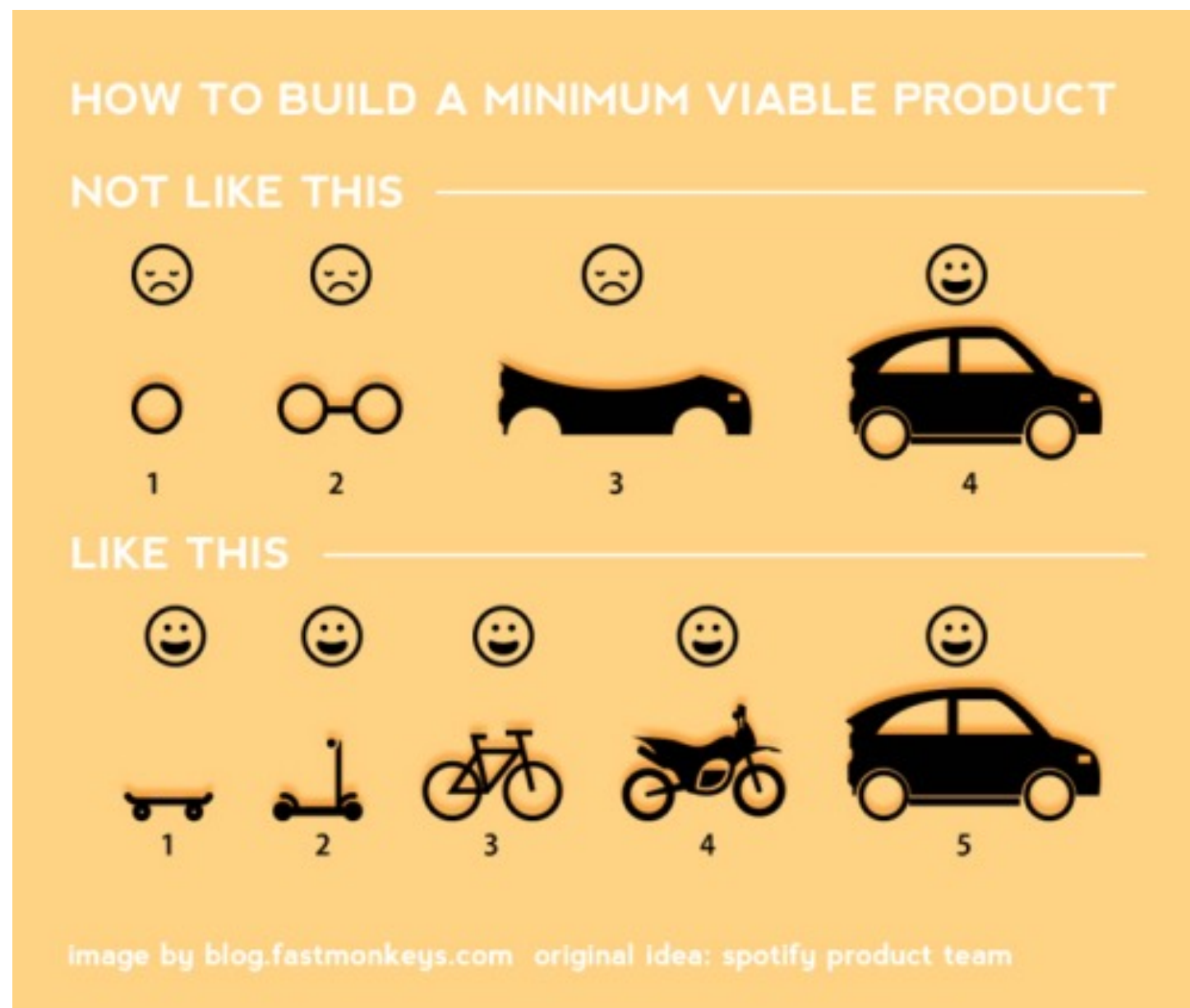
## ↻ Chemin d'une Application



# A. Plateforme Android

---

## 🔄 Chemin d'une Application



# A. Plateforme Android

---

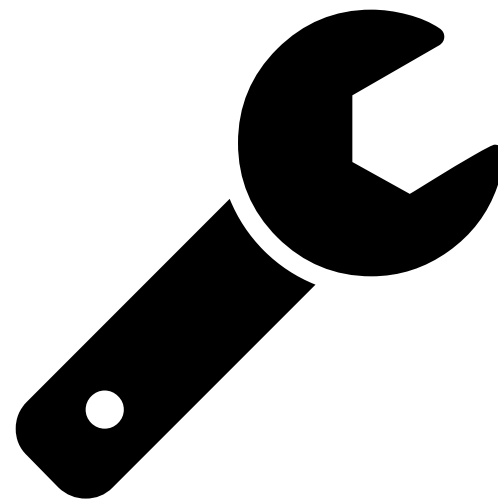
↻ Chemin d'une Application



# A. Plateforme Android

---

↻ Chemin d'une Application





# B. Environnement de Développement

---

</> Langage Kotlin ou C/C++

 Android Studio

 SDK Android

 Déploiement

# B. Environnement de Développement

---

</> Langage Kotlin



# B. Environnement de Développement

---

</> Langage Kotlin ou C/C++



JDK :

- + Plus courant
- + Outils Android



NDK :

- + Plus de performances
- + Bibliothèques C/C++

# B. Environnement de Développement

---



Pré-requis

+ JDK (Oracle)

+ Programmation Orientée Objet / Fonctionnelle

# B. Environnement de Développement

---



Logiciel basé sur IntelliJ

Site Officiel Android  
+ [developer.android.com](https://developer.android.com)

Installation  
+ Android Studio  
+ SDK Android  
+ Plugin Kotlin.

ou bien Android Studio 3.0 Canary

# B. Environnement de Développement

---

## Android Studio

### Utilisation

- + Créer un nouveau projet
- + Ouvrir un projet
- + Récupérer un projet depuis un gestionnaire de version (GIT, SVN)
- + Importer un projet Android (Eclipse)
- + Importer un exemple Android
- + Configurer SDK, Préférences, Plugins, Import, Export
- + Lire la Documentation



# B. Environnement de Développement

---

## Android Studio



### Principaux outils dans un projet

- + Gestion des émulateurs /Moniteur d'appareils Android (ADV)
- + Gestion du SDK Android
- + Exécution du projet

# B. Environnement de Développement

---

## Android Studio

Gradle (Maven, Ant)

- + Configurer processus de compilation
- + Créer plusieurs exécutables
- + Gérer les dépendances
- + Configuration du projet
- + Numéro de version
- + Nom de la version





# B. Environnement de Développement

---

 SDK Android

 Outils indispensables pour créer Application Android

 Pour chaque version Android

# B. Environnement de Développement

---



- + **aapt** (Android Asset Packaging Tool)
- + **adb** (Android Debug Bridge)
- + **ddms** (Dalvik Debug Monitor Service)
- + Système émulateur
- + Documentation
- + Projets Exemples

# B. Environnement de Développement

---



Configuration via Android Studio

 Mettre à jour les versions

 Télécharger les dernières versions

# B. Environnement de Développement

---

## SDK Android

Configuration via Android Studio (Gestionnaire)

### + Section Tools

+ SDK Tools, SDK Platform-tools, Android SDK Build Tools



### + Section Android X.X (5.0.1)

+ Doc, Platform, Sample, Google APIs, Source

### + Section Extras

+ Android Support Library, Google Repository, Google Play Services, ( Google USB<sup>40</sup> Driver Package )

# B. Environnement de Développement

---

## SDK Android

- + *add-ons*
- + build-tools
- + *docs*
- + extras
- + Emulator
- + platform-tools
- + platforms
- + *samples*
- + *sources*
- + skins
- + tools



# B. Environnement de Développement

---

## ☐ Déploiement

Simuler différents appareils



- + Type d'appareils (téléphone, tablette)
- + Type de téléphone (Nexus, Galaxy, etc.)
- + Taille d'écrans (résolutions)
- + Version d'Android (API 21, API 28)

# B. Environnement de Développement

---

## ☐ Déploiement

Alternatives

- + Genymotion

  - + machine virtuelle

(fluidité, rapidité, performances)



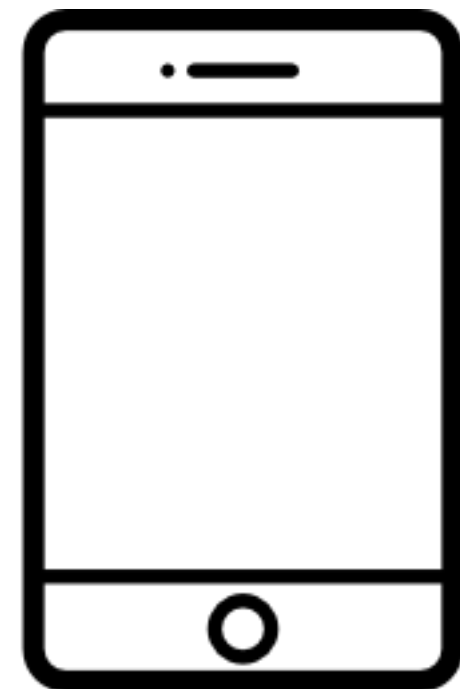
# B. Environnement de Développement

---

## ☐ Déploiement

Alternatives

+ Appareils physiques





# C. Principes de Programmation

---

 Architecture Android

 Composantes Android

 Cycle de vie d'une Activity






 Manifeste

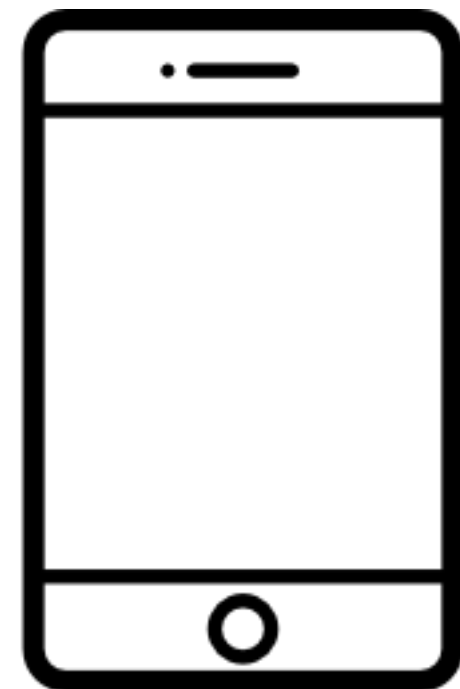
# C. Principes de Programmation

---

## Architecture Android

### Systeme contraint

-  Mémoire disponible
-  Stockage disponible
-  Consommation de la batterie
-  Débit réseau
-  Diversité des appareils



# C. Principes de Programmation

---

## Architecture Android

 → Contraintes de développement

 → Création de nouveaux objets

 → Utilisation des ressources

 → Utilisation de la batterie

 → Débit réseau

 → Différences d'affichage

→ Fragmentation des versions Android

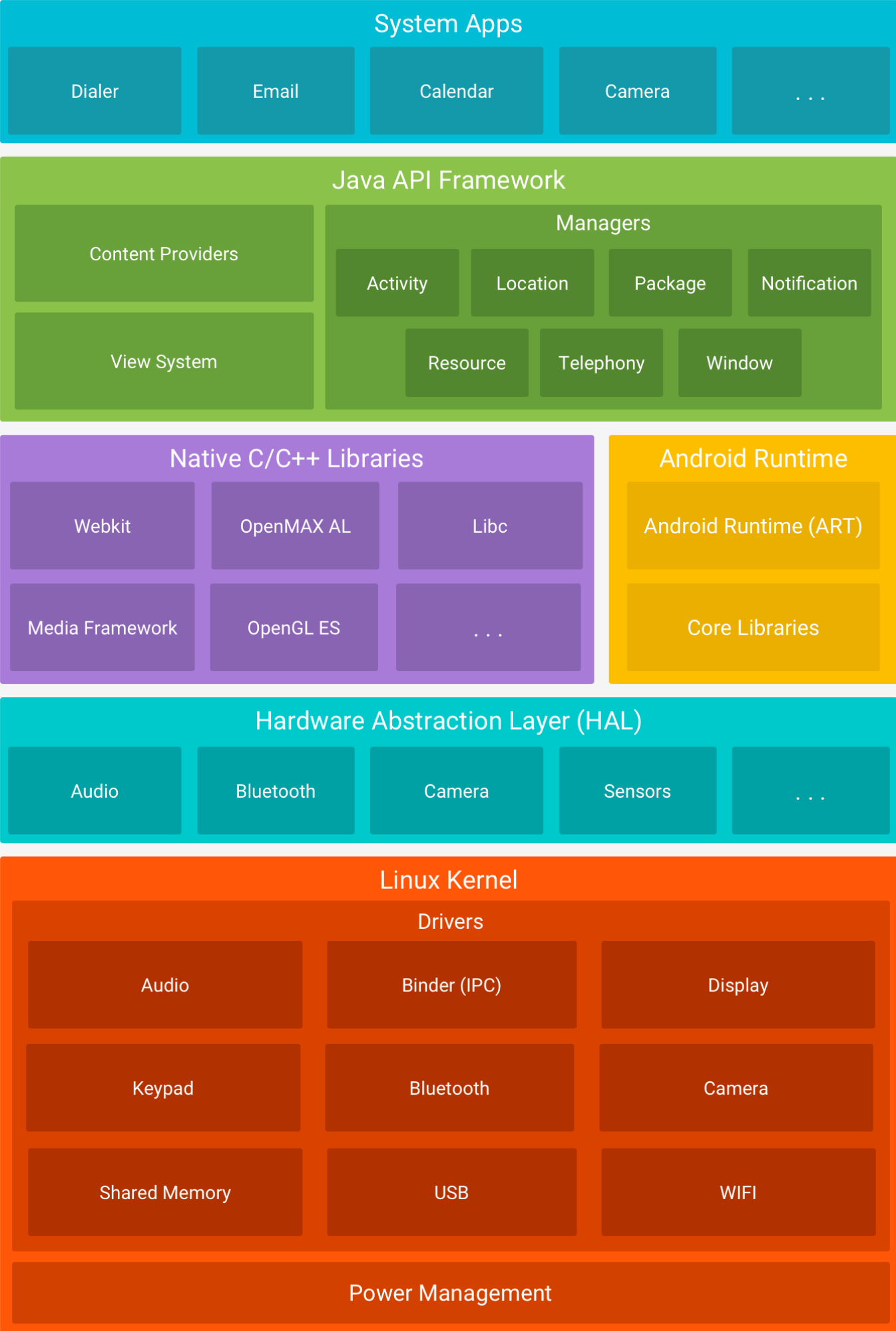
# C. Principes de Programmation

---

## Architecture Android

5 parties distinctes

- + Application
- + Framework Android
- + Bibliothèques
- + Android RunTime
- + Linux Kernel



# C. Principes de Programmation

---

## Architecture Android

### ART

- + Remplace DALVIK (Just In Time)
  - + performances, espace de stockage
- + AOT Compilation
- + Amélioration du Garbage Collector
- + Amélioration du débogage d'application

# C. Principes de Programmation

---

## Architecture Android

APK = Exécutable de l'application

Génération d'APK via **aapt**

+ signé ou non

# C. Principes de Programmation

---

## Architecture Android

APK :

 Code

 Ressources

 Assets

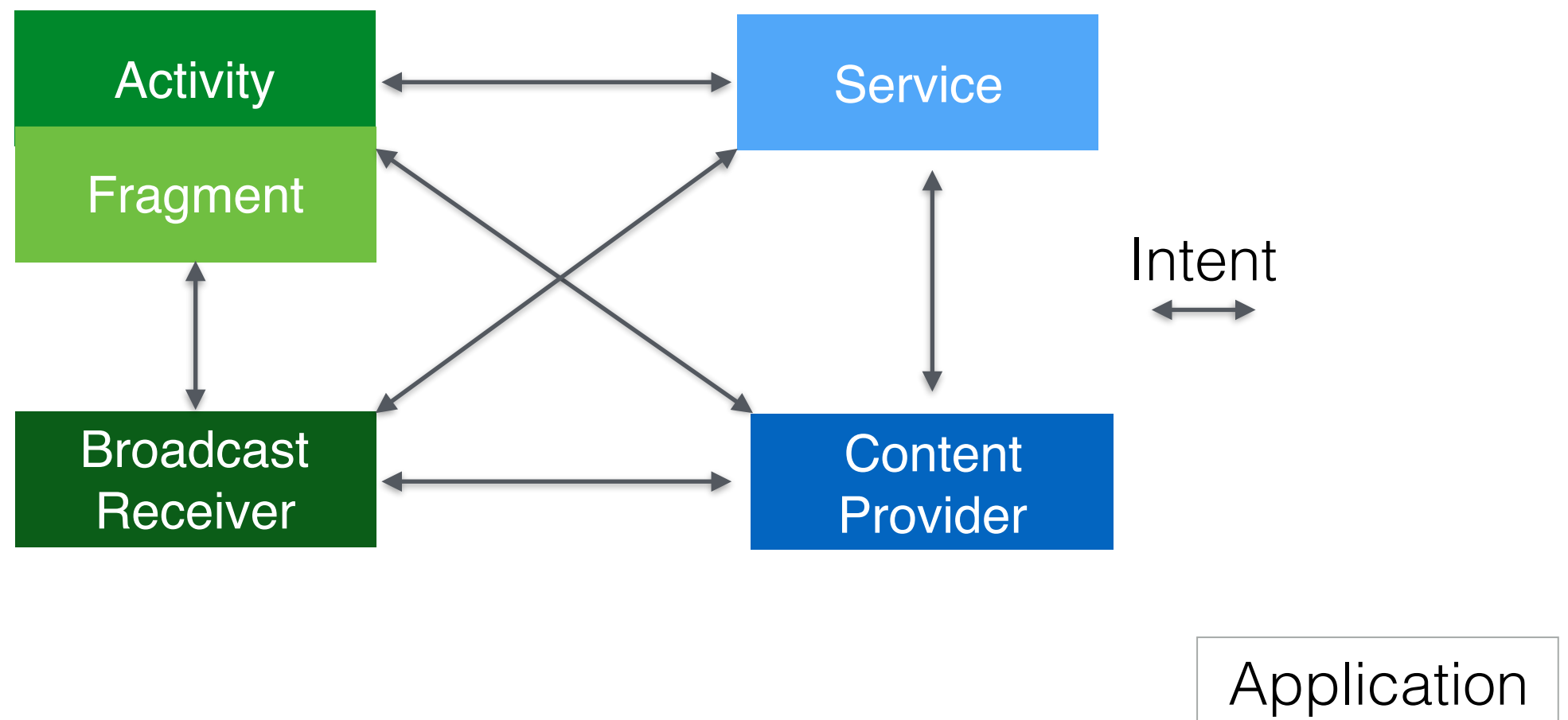
 Certificats

 Manifeste



# C. Principes de Programmation

## Composantes Android



# C. Principes de Programmation


---



## Composantes Android

### Activity

*Activité*, composante principale

- + Application composée de plusieurs Activity
- + Souvent le point d'entrée d'une application
- + Représente un écran
- + Associée à une vue XML
- + Déclarée dans le fichier Manifeste 



# C. Principes de Programmation

---

## Composantes Android

### Fragment

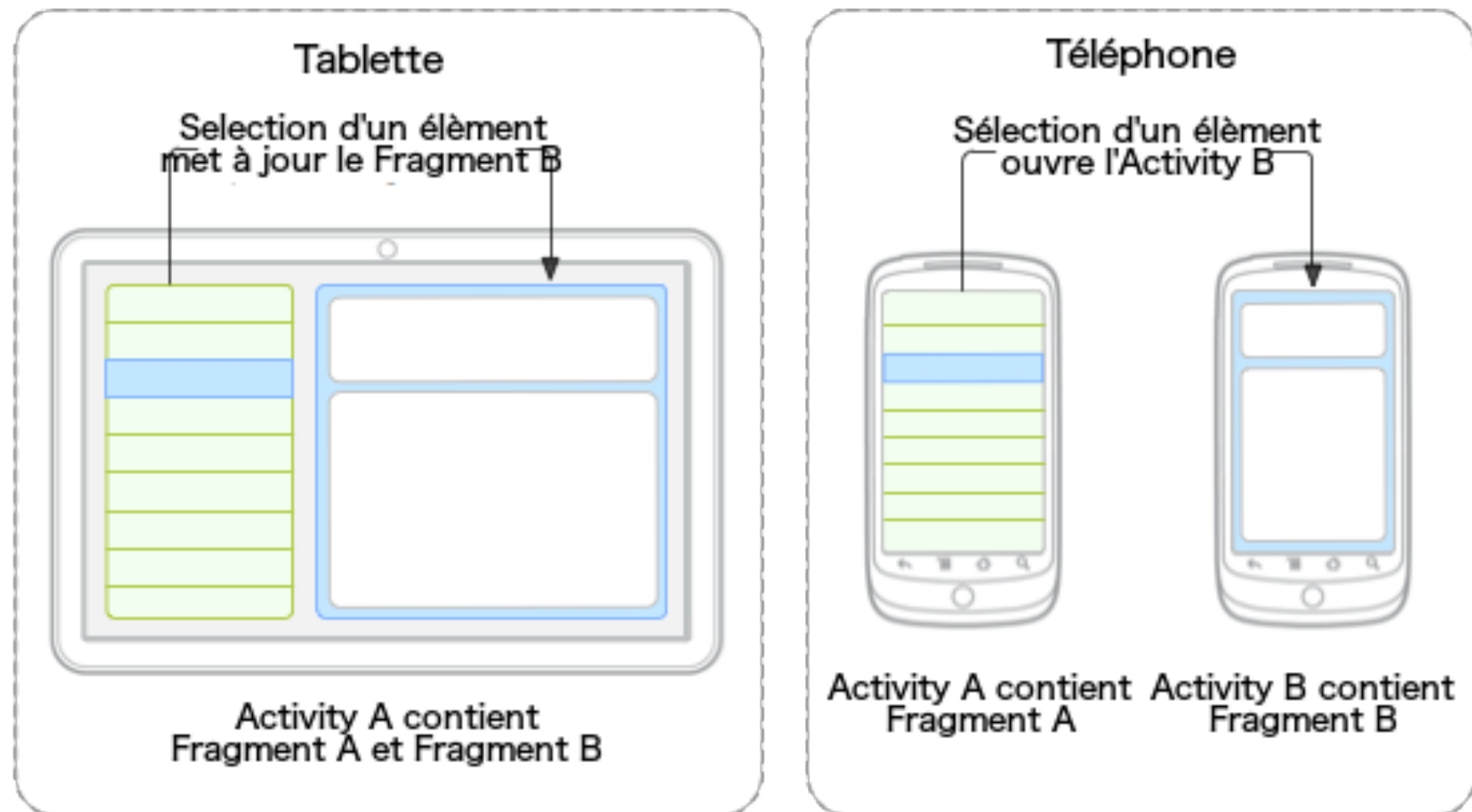
*Fragment*, Sorte de 'Sous Activity'

- + Apparue dans Android 3.0 
- + Optimisation de l'interface pour tablette 
- + Représente une portion d'écran
- + Associé à une vue XML
- + Contenue dans une Activity

# C. Principes de Programmation

## Composantes Android

### Fragment








# C. Principes de Programmation

---

## Composantes Android

### Service

#### *Service*

- + Processus sans interface
- + Exécution d'un traitement en tâche de fond
- + Utilisé pour traitement lourd 
- + Appel distant 
- + Téléchargement 
- + Calcul intensif 
- + Lecture audio 

# C. Principes de Programmation

---

## Composantes Android

### Broadcast Receiver

#### *Récepteur d'évènements*

+ Réagit à un évènement système

✉ Réception SMS

🔄 Démarrage du téléphone

🔒 Verrouillage de l'écran

+ Traitement de tâches légères

📱 Lancer une app

🔄 Mettre à jour des données

# C. Principes de Programmation

---

## Composantes Android

### Content Provider

*Fournisseur de contenu*

+ Partage les données stockées

 base SQLite

 fichiers, etc.

+ Exemple

 Contact

 Agenda

 Média

# C. Principes de Programmation

---

## Composantes Android



La classe *Application*

↻ Instanciée tout au long de l'exécution de l'app

✓ Maintient l'état global



Surcharge de la classe

- + Sauvegarde de variables globales
- + Gestion des ressources
- + Déclaration de constantes globales



A déclarer dans le fichier Manifeste



# C. Principes de Programmation

---

## ↻ Cycle de vie d'une Activity

Android gère les ressources disponibles

→ Fermeture d'application

◎ Libérer ressources

# C. Principes de Programmation

---

## Cycle de vie d'une Activity

Choix de l'application à fermer

☒ État du processus

 Date

# C. Principes de Programmation

---

## ↻ Cycle de vie d'une Activity

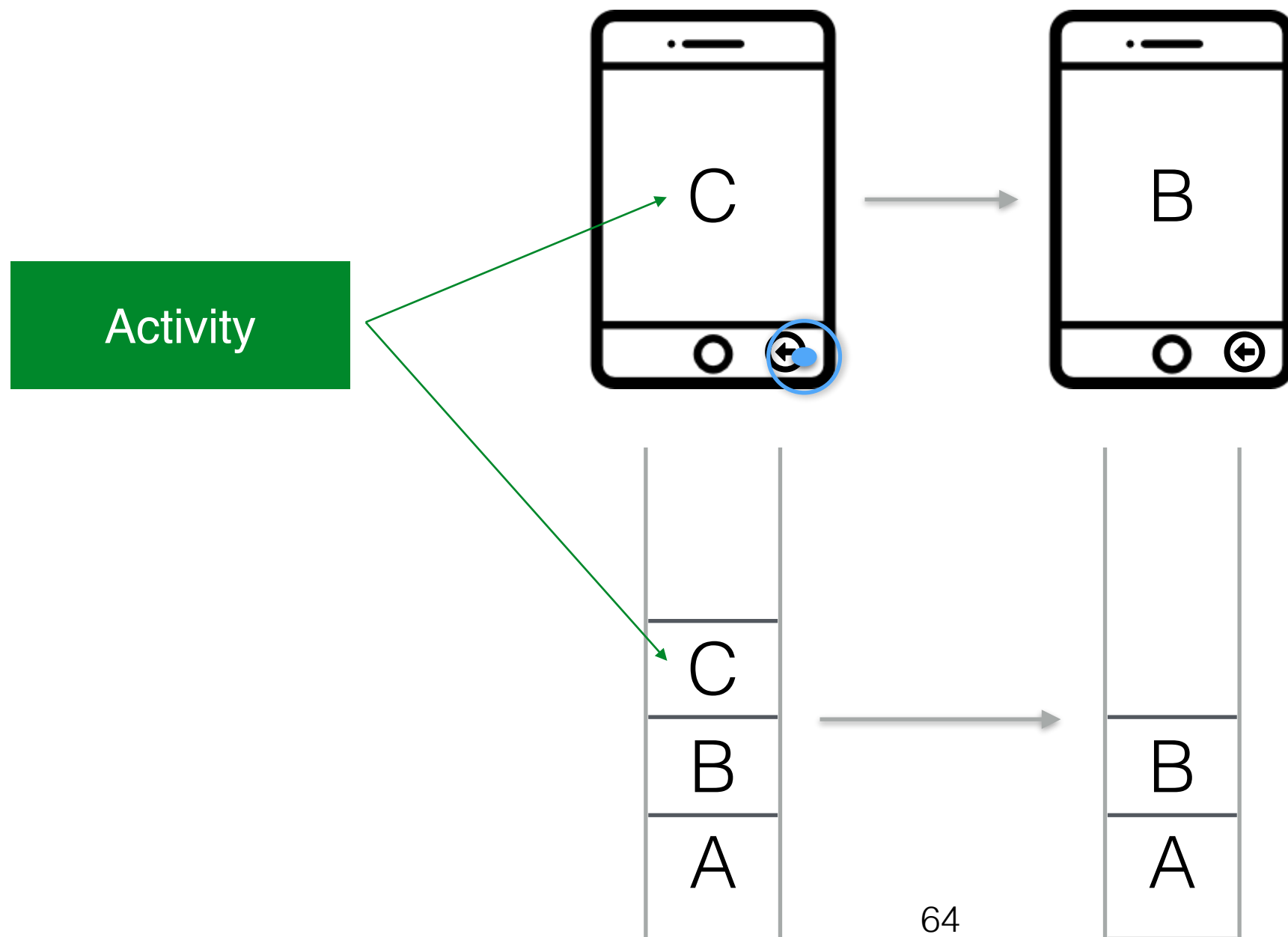
Back Stack



# C. Principes de Programmation

---

## ↻ Cycle de vie d'une Activity



# C. Principes de Programmation

---

## Cycle de vie d'une Activity

### États

 En création

 En cours d'exécution

 En pause

 Stoppée

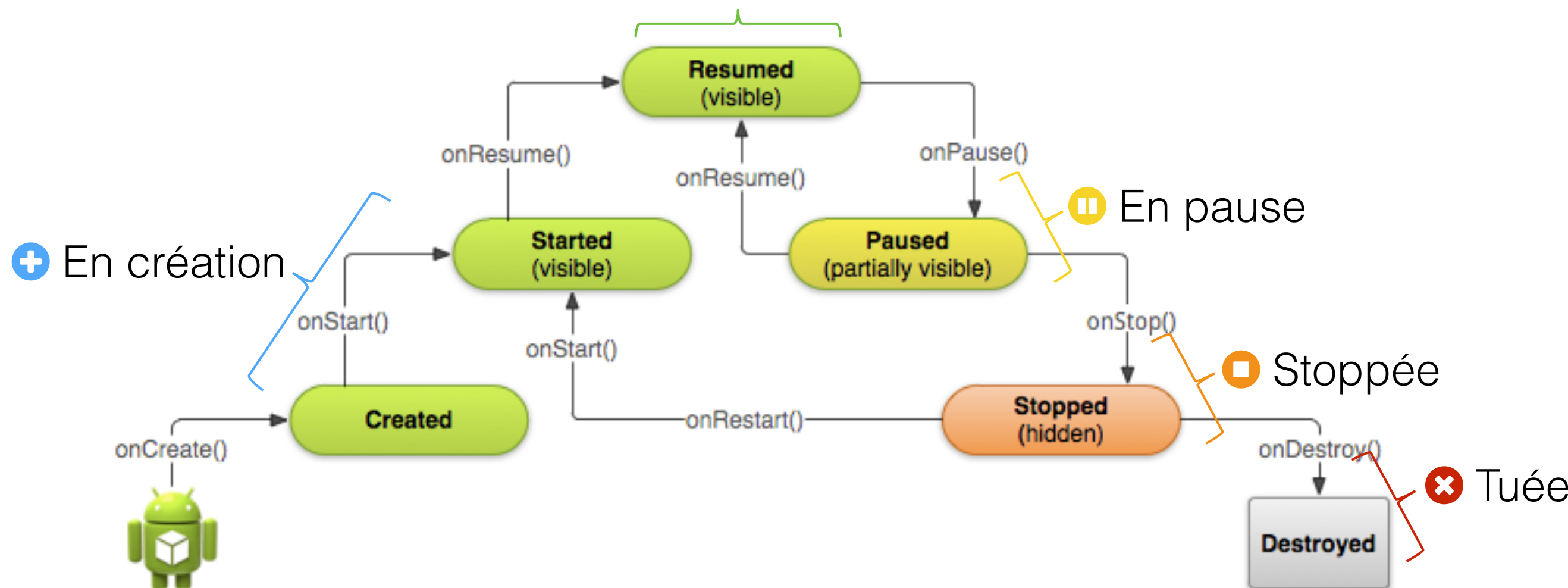
 Tuée

Activity

# C. Principes de Programmation

## 🔄 Cycle de vie d'une Activity

▶ En cours d'exécution



# C. Principes de Programmation

---

## ↻ Cycle de vie d'une Activity

`onCreate()` : l'Activity est créée

➔ `onStart()`

`onStart()` : l'Activity est visible à l'utilisateur

➔ `onResume()` | `onStop()`

`onResume()` : l'Activity est prête à interagir avec l'utilisateur

➔ `onPause()`

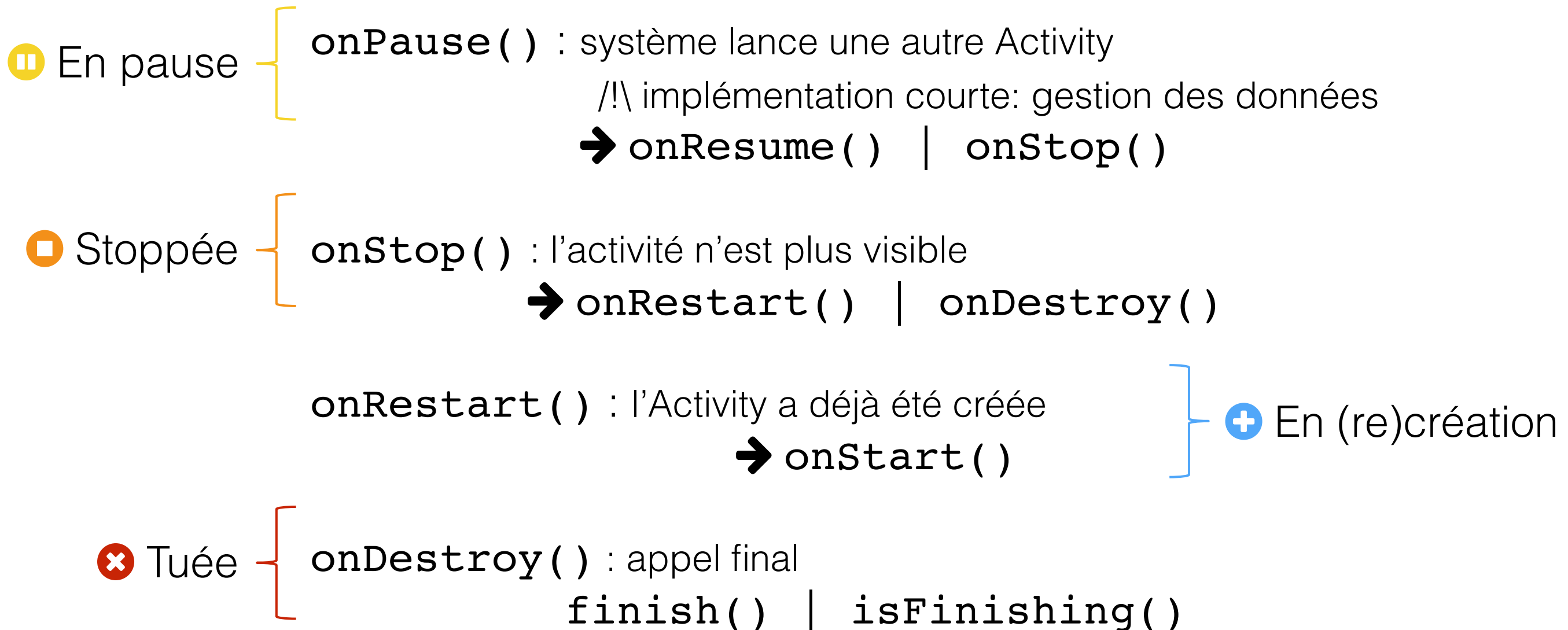
⊕ En création

▶ En cours d'exécution

# C. Principes de Programmation

---

## ↻ Cycle de vie d'une Activity





# C. Principes de Programmation

---

## Manifeste

### **AndroidManifest.xml**

 Composantes  
+ Activities  
    Configuration  
    Lanceur d'application  
    Affichage Portrait|Paysage

+ Services

 Permissions




 Web Service

# C. Principes de Programmation

---

## Manifeste

La version 1.9.0 peut accéder aux éléments suivants :

-  Données de localisation
  - position précise (GPS et réseau)
  
-  Photos/multimédia/fichiers
  - Modifier ou supprimer le contenu de la mémoire de stockage USB
  - Lire le contenu de la mémoire de stockage USB
  
-  Espace de stockage

 Permissions d'une application

# C. Principes de Programmation

## Manifeste

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.chillcoding.hello">

    <uses-permission android:name="android.permission.INTERNET" />


    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">

        <activity
            android:name="com.chillcoding.hello"

        .
        .
        .

    </manifest>
```

← Déclaration xmlns  
← Identifiant


←  Permission

← Application

← Icône

← Nom

← Thème

←  Composante



# Premier Projet

---

+ Création

📁 Arborescence

📄 Fichiers clés

📱 Déploiement



# Premier Projet

---

## + Création

### **Informations du projet :**

+Name

+Company Domain

+Package Location

+Appareils et Version

+Choix de l'Activité principale

+Conventions de Nommage des fichiers utilisés

+Activity, layout xml, menu, titre



# Premier Projet

---



## Arborescence

+app

+manifests

+java

+res

+drawable

+layout

+values

+menu

+Gradle Scripts



# Premier Projet

---

## Fichiers clés

- + Manifeste
- + Ressources
  - +drawable
  - +layout
  - +values
- + Fichier Générer
  - +gen/R.java
- + Fichier source
- + Fichier de configurations



# Premier Projet

---

## Exercices

1. Quel fichier modifier pour changer le texte affiché à l'écran ? Modifiez pour afficher « Hello Kotlin ! ».
2. Quel fichier modifier pour ajouter des éléments graphiques à l'écran ? Ajoutez du texte.
3. Dans le fichier des ressources relatif aux images, res/drawable, ajoutez une image de votre choix.
4. Ajoutez l'image à côté du texte.
5. Ajoutez un bouton qui change le texte initial en un autre texte.





# Premier Projet

---



## Annexe

Commentaires XML : `<!-- -->` JAVA / KOTLIN : `//`

Téléchargement :

SDK : Version Android, dernière + avant dernière (pour le modèle de projet)

Émulateur : Google API x86 Atom System Image (API 27)

Instant Run -> Télécharger complètement pour la version de son tel.

Internationalisation et strings.xml

LogCat

findViewById ou extensions Kotlin

Click : en XML avec `onClick` en JAVA/KOTLIN avec `setOnClickListener` ou `implements`



# Premier Projet

---

## Convention de nommage

NomDeClasse : CamelCase

nomDeVariable : camelCase

CONSTANTE : UPPER\_CASE

fichier\_ressource : snake\_case

idElementGraphique

id\_string

# Conclusion

---

 Plateforme Android

 Environnement de Développement

 Principes de Programmation

 Premier Projet

« Tout seul on va plus vite,  
ensemble on va plus loin. »  
- proverbe africain

# Sources

---

- [Open Handset Alliance](#)
- [We are social : Digital, social and mobile in 2015](#)
- [Kantar WorldPanel : Part de marché des OS Mobile](#)
- [Developer Android: Dashboard, Platform versions](#)
- [Sébastien Mosser : Appliquer des principes d'agilité](#)
- [Developer Android: Android NDK](#)
- [Developer Android: Get Last Android Studio](#)
- [Udacity: Android Development for Beginners \(Video\)](#)
- [Developer Android: Application Fundamentals \(API Guide\)](#)
- [Developer Android: Activities](#)
- [Edition Eni : Android 5 Les fondamentaux du développement d'applications Java \(Livre\)](#)
- [developer.android.com: Activity in Android](#)
- [developer.android.com: Managing Life Cycle](#)
- <https://developer.android.com/guide/platform/>

# Sources

---

- Quizz ABC d'Android

# Premier projet

- AK 1 : Installer un environnement de développement Android
- Introduction à Kotlin
- AK 3 : Utiliser extensions kotlin

# Quizz ABC d'Android

---



# QUIZZ

---

Adb ? Où ? Utilité ?

*AppCompatActivity* ?

*Fragment* ?

Cycle de vie ?

R ? Pb. ? Comment le générer ?

*res/* ?

Défaut/Avantage d'Android ?

Cycle de Vie d'une *Activity* ?