



CHAPITRE 3

LA CLASSE ACTIVITY

• Mr. MEGHAZI

2016-2017

Cours pour les Master II - GL

Contenu



- La Classe « Activity »
- La pile de retour des tâches (Task BackStack)
- Le cycle de vie d'une Activité
- Démarrage des activités
- ...

Une Activité ...

- Fournie une interface visuelle d'interaction avec l'utilisateur
- Chaque activité s'en charge et se focalise sur une chose que l'user peut faire, comme:
 - ▣ Consulter un mail
 - ▣ Afficher une interface d'authentification ...
- En respectant cette convention, on va finir par créer des applications en chainant ensemble plusieurs activités, laissant l'user naviguer à travers

Naviguer à travers les activités

- Android permet la navigation par l'intermédiaire de plusieurs moyens:
 - ▣ Tâches
 - ▣ Le « Task BackStack »
 - ▣ L'interruption et la reprise des activités

Les tâches

- Pour Android une tâche est seulement un ensemble d'activités en relation
- Ces activités n'ont pas à être, forcément, partie de la même Application
- La majorité des tâches initient leur exécution à partir du « Home Screen »

La pile des tâches (1)



- Quand une activité est lancée, elle est placée en haut de la pile « BackStack »
- Quand une activité est détruite elle est sortie (déplier) du BackStack

La pile des tâches (2)

- Comment cette pile fonctionne ?



Le Cycle de vie d'une Activité

- Les activités sont créées, suspendues, reprises et détruites, si nécessaire , quand une Application s'exécute.
- ▣ Certaines de ces actions dépendent du comportement de l'user
- ▣ Certaines d'autres dépendent d'Android
 - **Exemple:** Android peut « Tuer » des activités quand il aura besoin de leurs ressources

Les états du cycle de vie (Activity)

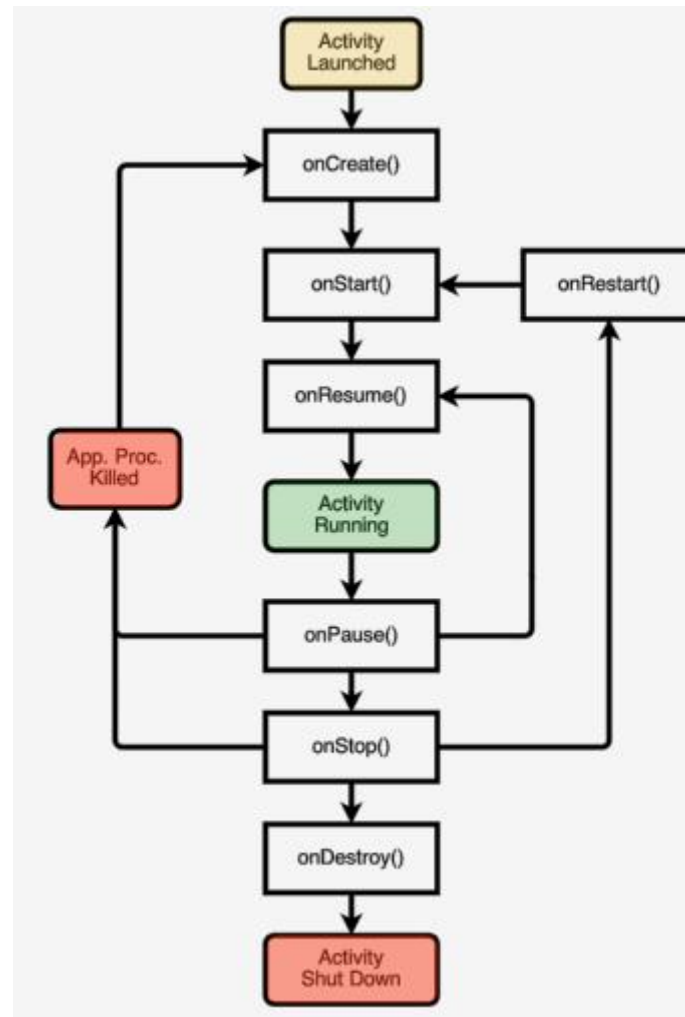


- **Resumed / Running** - visible, permet l'interaction avec l'user
- **Paused** - Visible, l'user ne peut pas interagir, peut être terminer
- **Stopped** - non Visible, peut être terminer

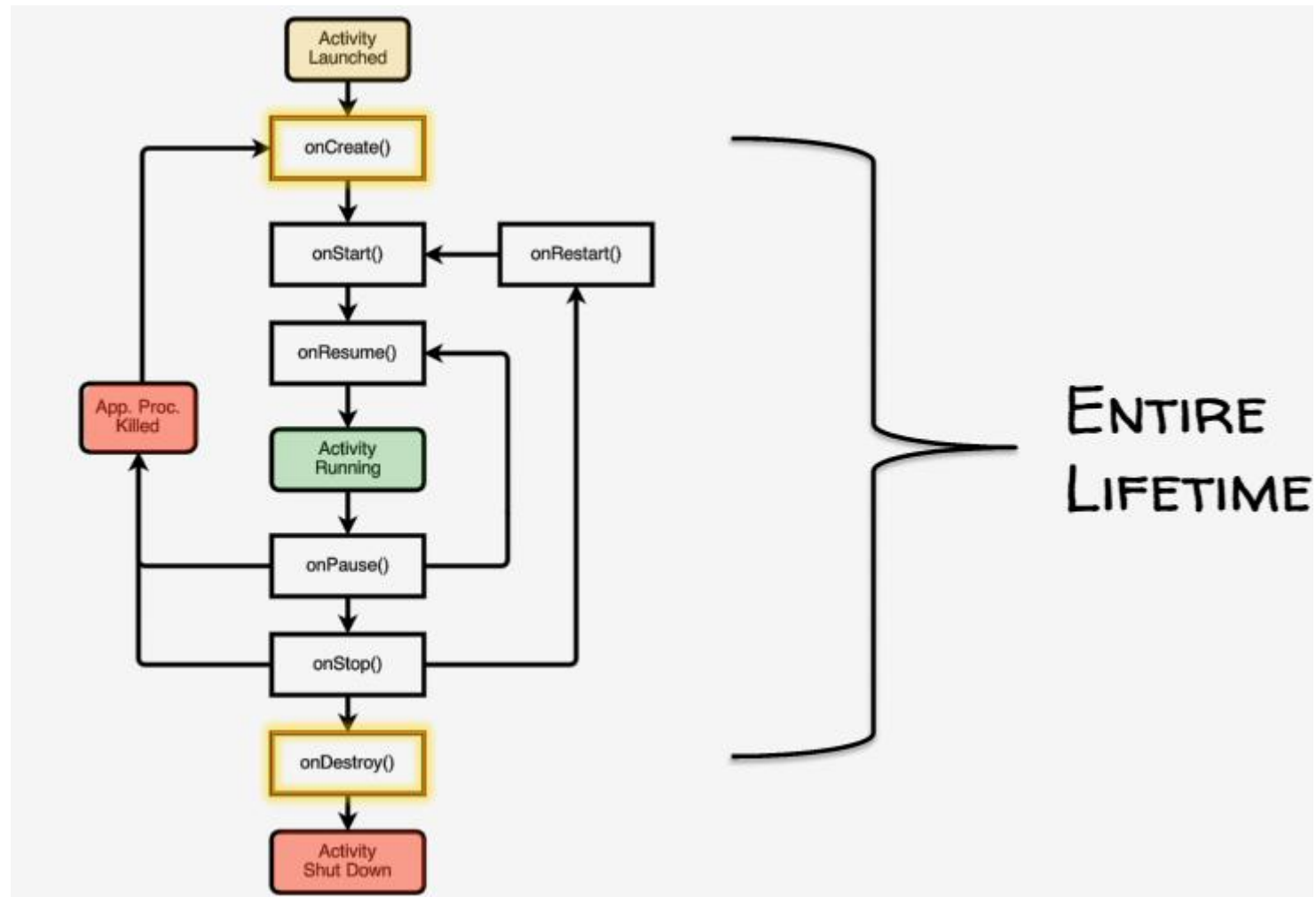
Les méthodes du cycle de vie

- Android annonce les changements dans le cycle de vie d'une activité en invoquant des méthodes spécifiques au activités:
 - ▣ `protected void onCreate(Bundle savedInstanceState)`
 - ▣ `protected void onStart()`
 - ▣ `protected void onResume()`
 - ▣ `protected void onPause()`
 - ▣ `protected void onRestart()`
 - ▣ `protected void onStop()`
 - ▣ `protected void onDestroy()`

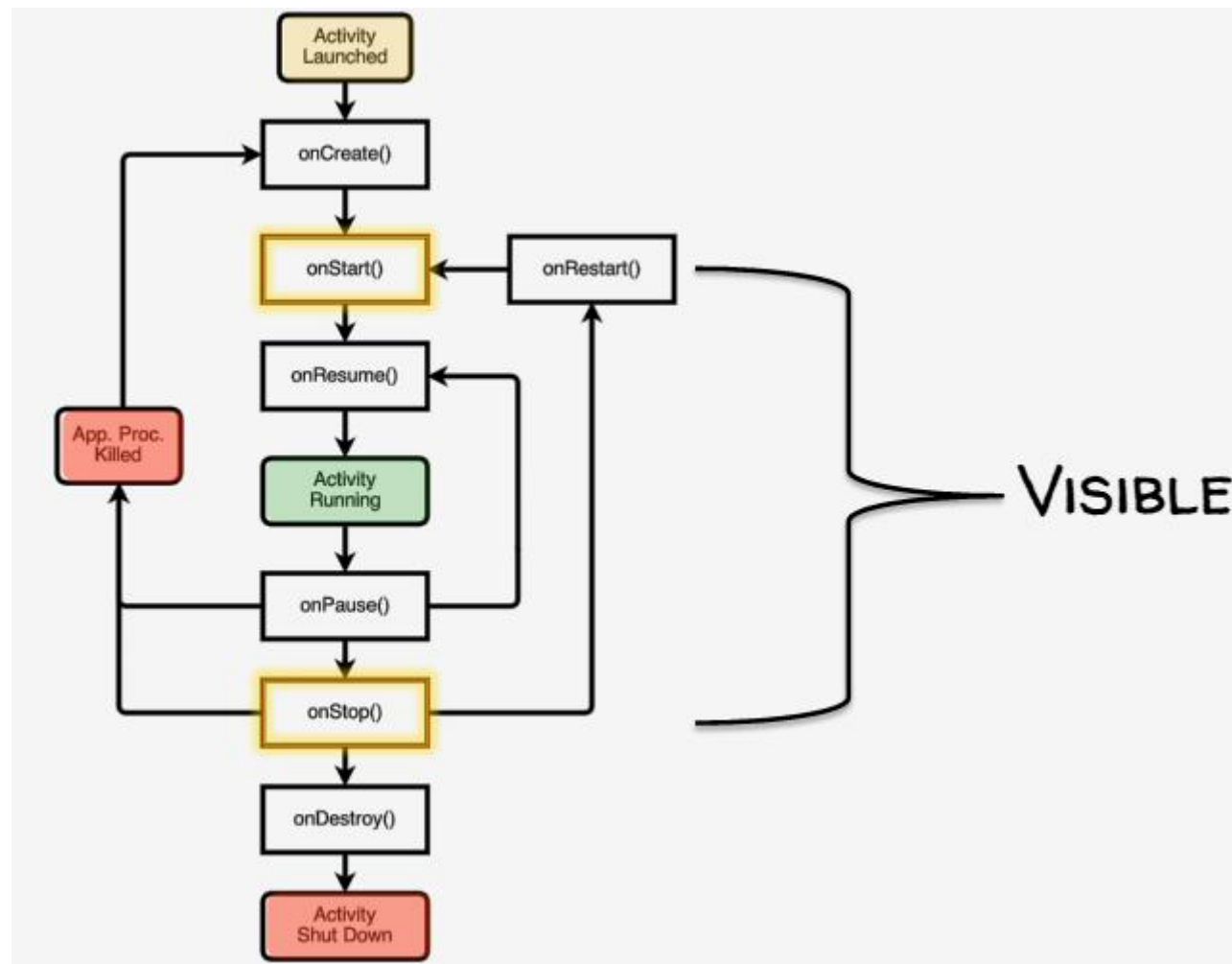
Le cycle de vie d'une activité (1)



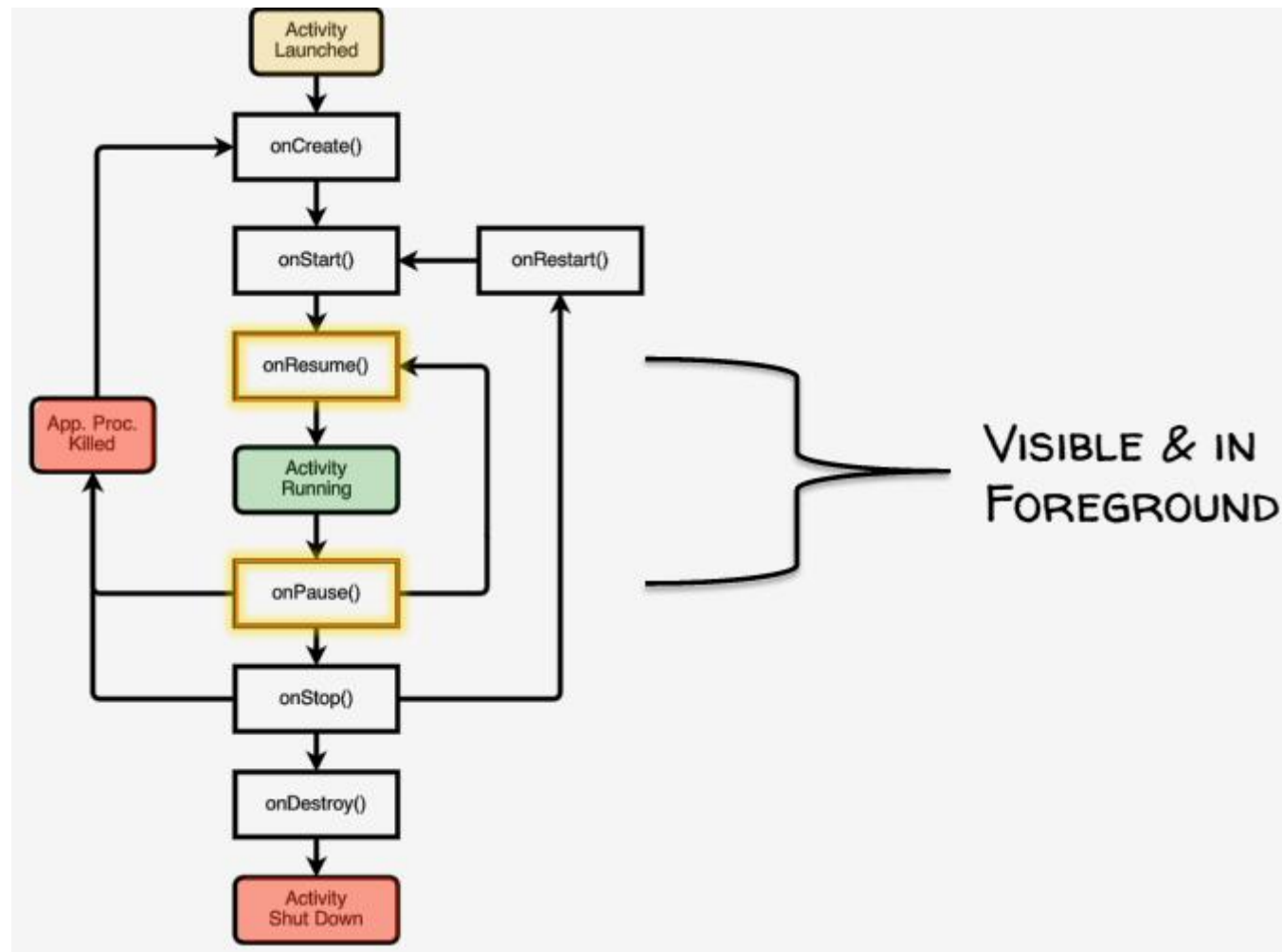
Le cycle de vie d'une activité (2)



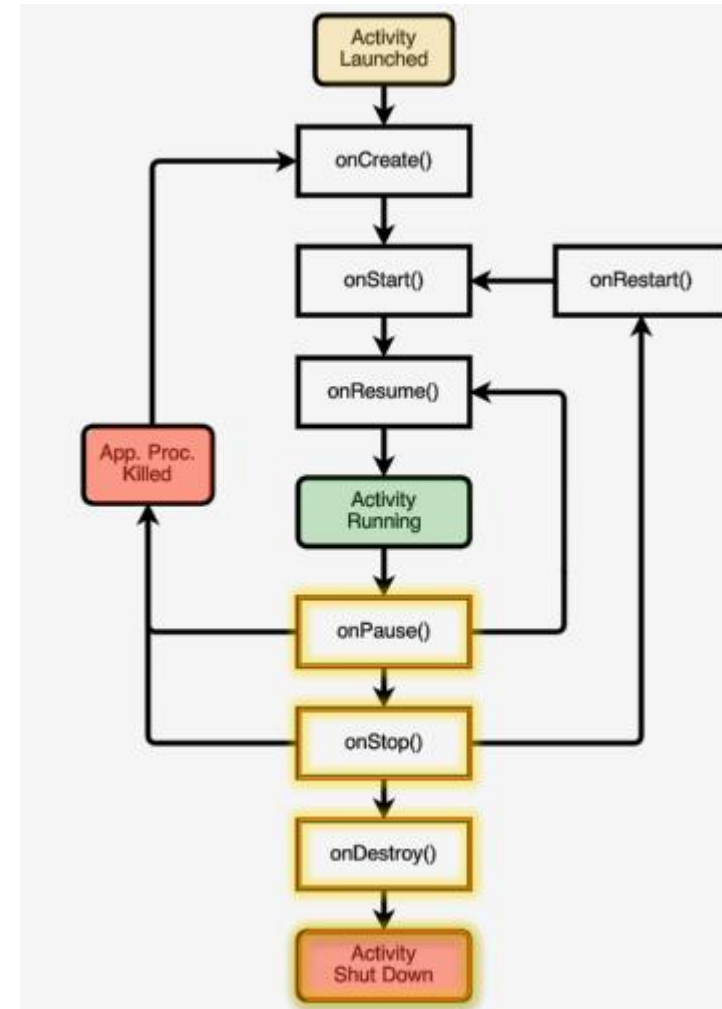
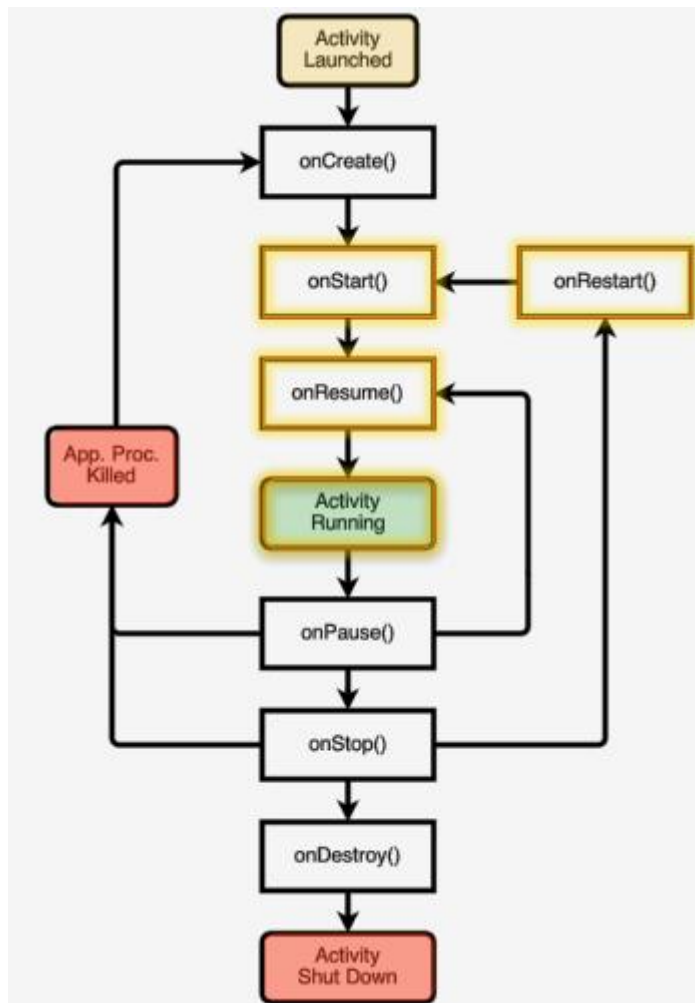
Le cycle de vie d'une activité (3)



Le cycle de vie d'une activité (4)



Le cycle de vie d'une activité (5)



OnCreate()

- Invoquée quand une activité est créée
- Met en place l'état initial
 - ▣ Invoque `super.OnCreate()`
 - ▣ Règle l'affichage et le contenu de l'activité
 - ▣ Conserve une référence aux différentes vues de l'UI, quand c'est nécessaire
 - ▣ Configure les vues, quand c'est nécessaire

OnRestart()

- Invoquée si une activité vient d'être stoppée et sur le point d'être démarrée à nouveau
- Scénario d'actions typiques
 - ▣ L'activité à besoin d'un certain traitement qu'après d'être stoppée

OnStart()

- L'activité est sur le point d'être visible
- Scénarios d'actions typiques
 - ▣ Démarre le «**when visible-only behaviors**»
 - **Exemple:** Charger les coordonnées d'un emplacement quand « maps » est sur le point d'afficher la carte.
 - ▣ Charger l'état persistant d'une application
 - **Exemple:** mettre à jour la liste des emails non-lus dans une application de messagerie.

OnResume()

- L'activité est visible et sur le point de commencer à interagir avec l'user
- Scénario d'actions typiques
 - Démarre le «Foreground-only behaviors»

OnPause()

- Se focalise sur basculer vers une autre activité
- Scénario d'actions typiques
 - ▣ Arrête le «Foreground-only behaviors»
 - ▣ Sauvegarder un état persistant

OnStop()

- L'activité n'est plus visible à l'utilisateur

- ▣ Peut être redémarrer plus tard.

- Scénario d'actions typiques

- ▣ Mettre l'état de l'activité en cache.

N.B.: peut ne pas être invoquée si Android tue l'application.

- Ne sauvegarder pas l'état à persister dans cette méthode (il faut le faire dans **onPause()**)

OnDestroy()

- L'activité est sur le point d'être détruite
 - Scénario d'actions typiques
 - ▣ Libère les ressources utilisées par l'activité
 - **Exemple:** arrêter des threads lancés par l'activité
- N.B.** peut ne pas être invoquée si Android tue l'application

Lancer des activités

□ C'est très simple !

1. Créer un objet « `Intent` » et spécifier l'activité à lancer
2. Faire passer en paramètre le « intent » créé récemment à des méthodes telles que:
 - `startActivity()`
 - `startActivityForResult()`
 - Va fournir un résultat pour l'activité

Voir l'exemple ;)



- MapLocation