

# P82 Mobile Application Development - Android

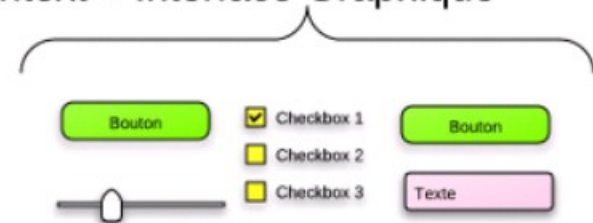
Activities

# Activity

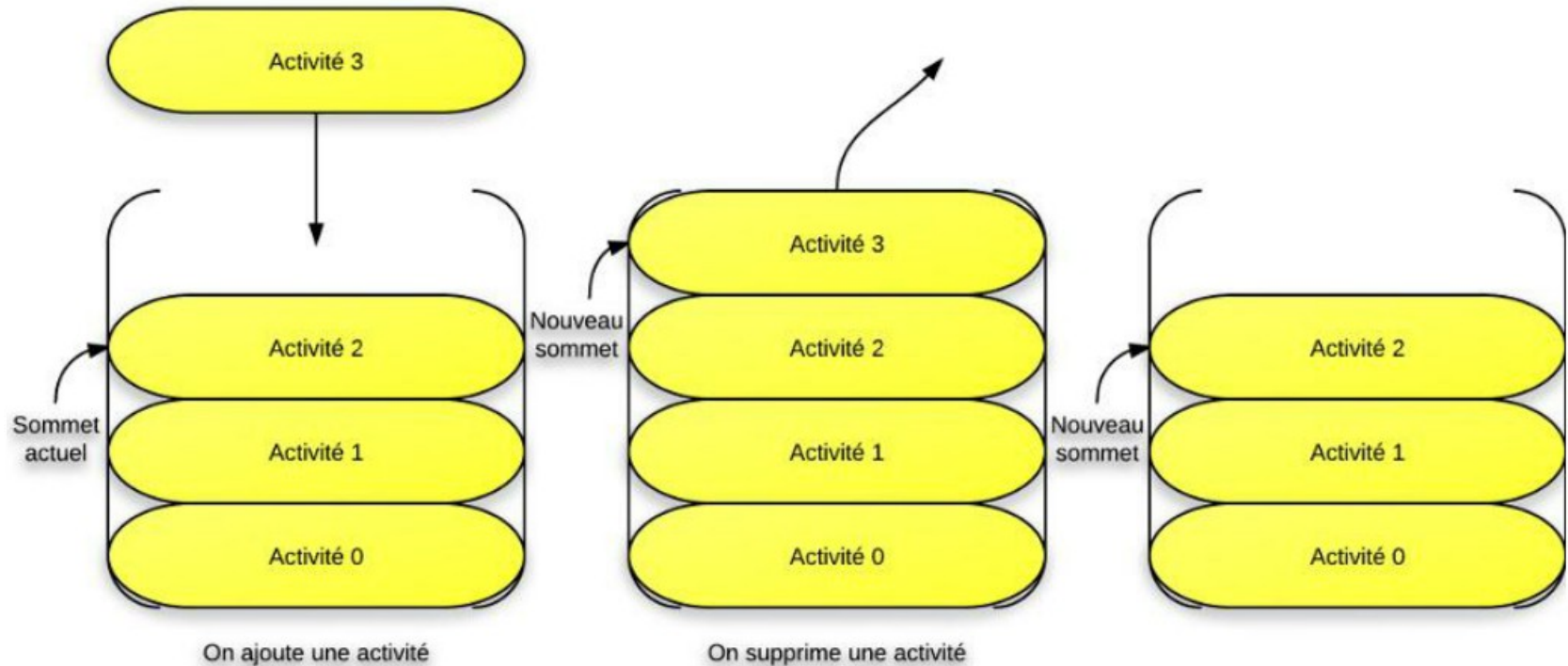
## Activity:

- basic block of an application
- represents the **screen that will be displayed** to the user
- class that inherits from Activity (or one of its descendants), and uses polymorphism on some of its methods.
- An Android application consists of a set of interrelated activities.
- An activity contains information about the current state of the application or the context. (link with Android system and other activities)

Activity = Context + Interface Graphique

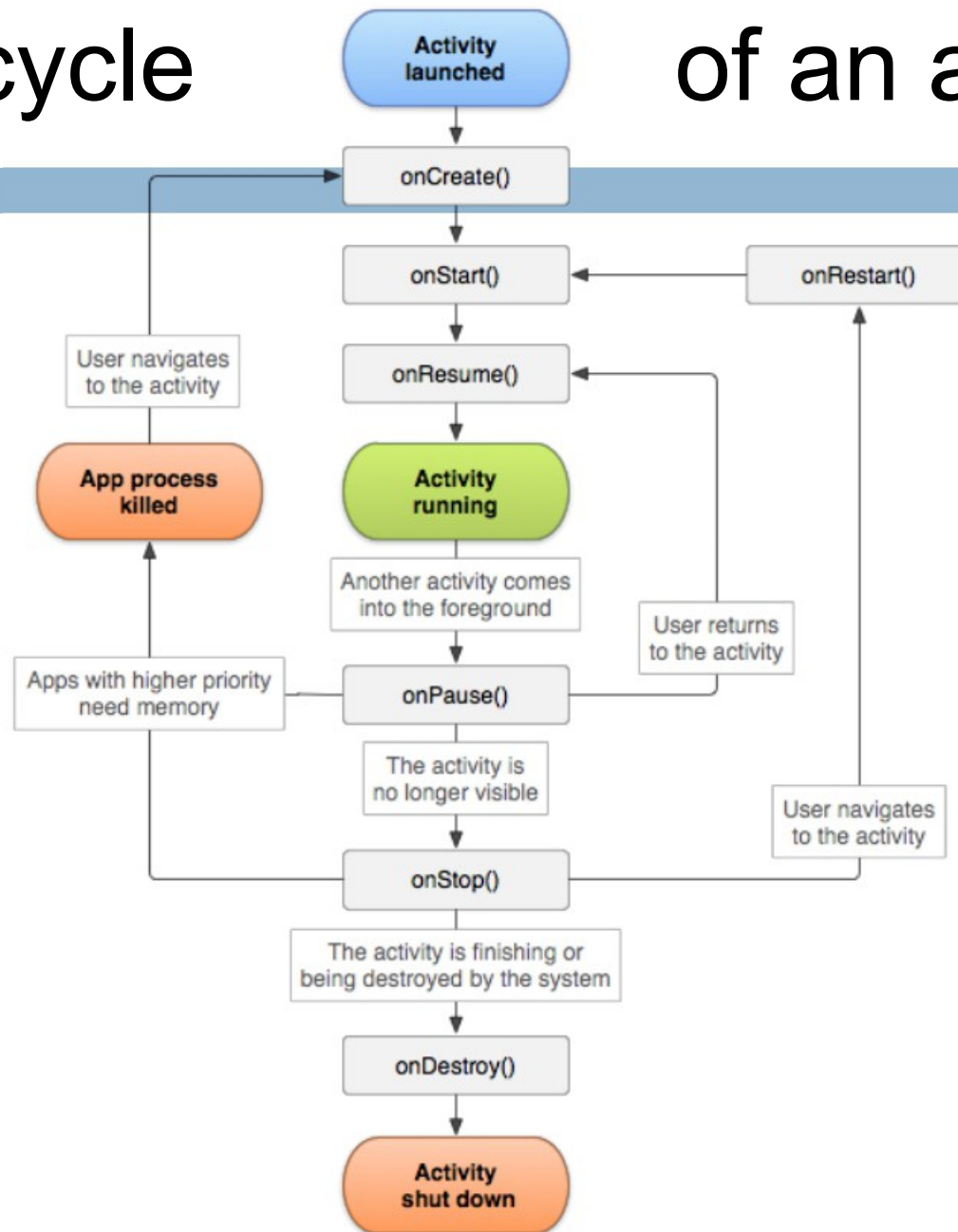


# Activities Behavior



# Life cycle

# of an activity



# Life Cycle (continued)

## □ onCreate()

- ⊙ First execution
- ⊙ Static declarations,
- ⊙ creating vies from XML,
- ⊙ loading data,
- ⊙ variable initialization,
- ⊙ Bundle object that contains the previous states of the activity

## □ onStart()

Called just **before the activity becomes visible to the user.**  
Retrieving the data saved during the last shutdown.

## □ onResume()

Executed at **whenever the activity comes into the foreground;**

# Life Cycle (continued)

## □ **onPause()**

Called when the system **calls another activity**

- ⊙ Free resources
- ⊙ save data
- ⊙ Stop animations, Threads
- ⊙ Must be fast

## □ **onStop()**

**activity is no longer visible**

## □ **onDestroy()**

Called **before the activity is destroyed**

## □ **onRestart()**

the activity **was stopped and is being reactivated.**

# Life Cycle

---

- Redefining all methods is not Mandatory;
- if redefining, do not forget to call the `super.onX ()`

# Main Activity States

## ***active/running :***

- ☐ visible activity
- ☐ holds the user focus and waits for entries

## ***paused :***

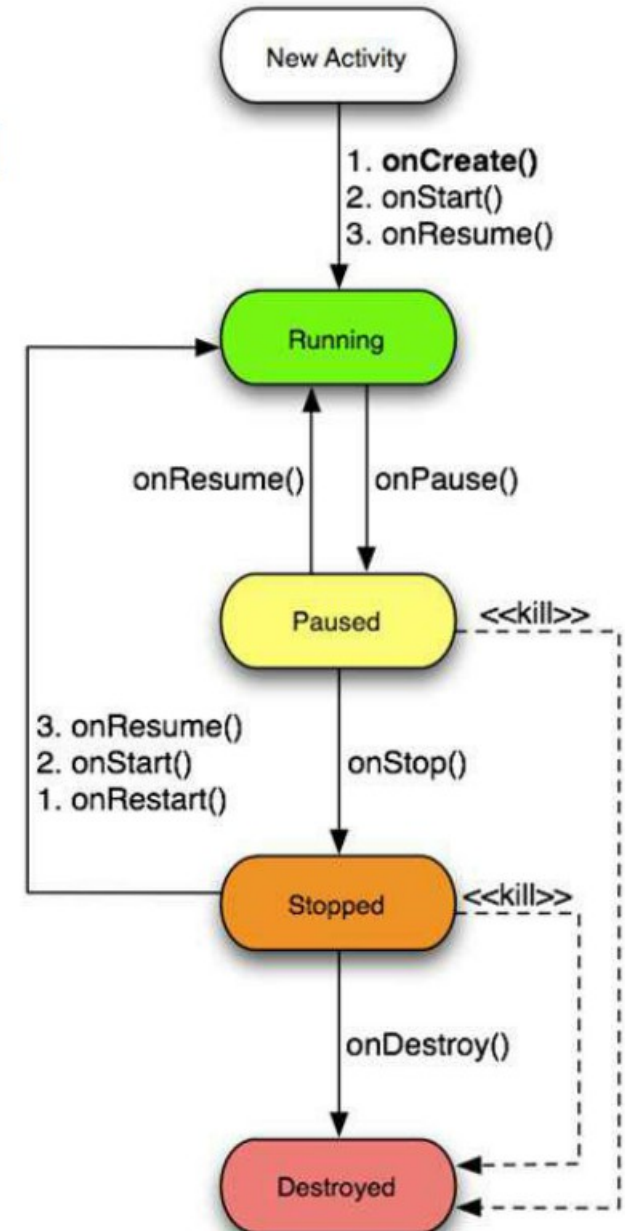
- ☐ partially visible activity on the screen
- ☐ BUT does not hold the focus (eg dialog box)
- ☐ we enter after the onPause method
- ☐ onResume or onStop methods make it possible to exit

## ***stopped :***

- ☐ activity is not visible
- ☐ it is the onStop method that leads to this state

***dropped : release memory***

## Activity Lifecycle



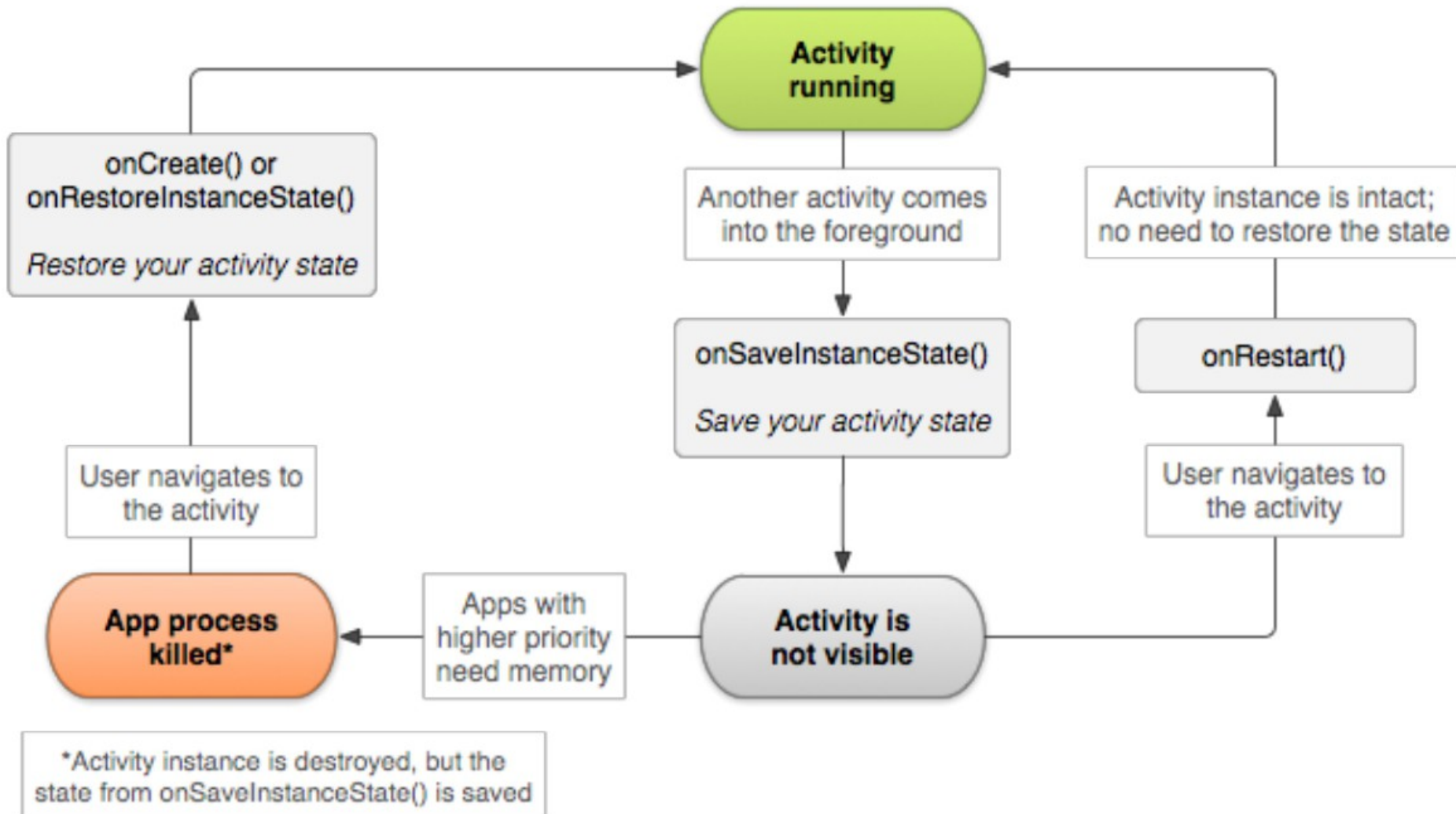


# Save activity status



- ❑ When Activity is paused or Stop all status remains the same
- ❑ If Activity destroyed by system everything is recreated
- ❑ A Callback method exists for the state backup

# Save activity status



# Save activity status

*onSaveInstanceState () saves the state of the activity to be able to restore it later in onRestoreInstanceState ()!*

```
Public final String ID_SAVE = "IDcomposant";
```

```
public void onSaveInstanceState(Bundle outState) { super.onSaveInstanceState(outState);  
    outState.putString(ID_SAVE, "1234567890");  
}
```

☐ Data recovery

```
public void onRestoreInstanceState(Bundle savedInstanceState) {  
    super.onRestoreInstanceState(savedInstanceState);  
    String ID = savedInstanceState.getString(ID_SAVE);  
}
```