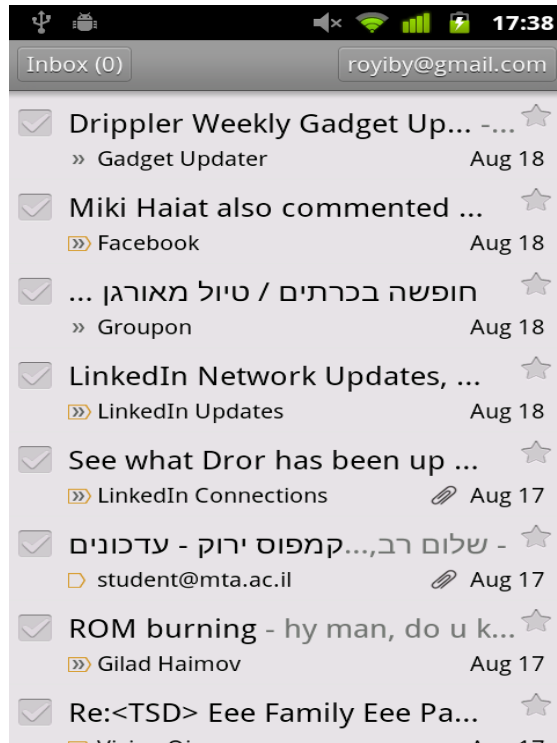


Activity life-cycle

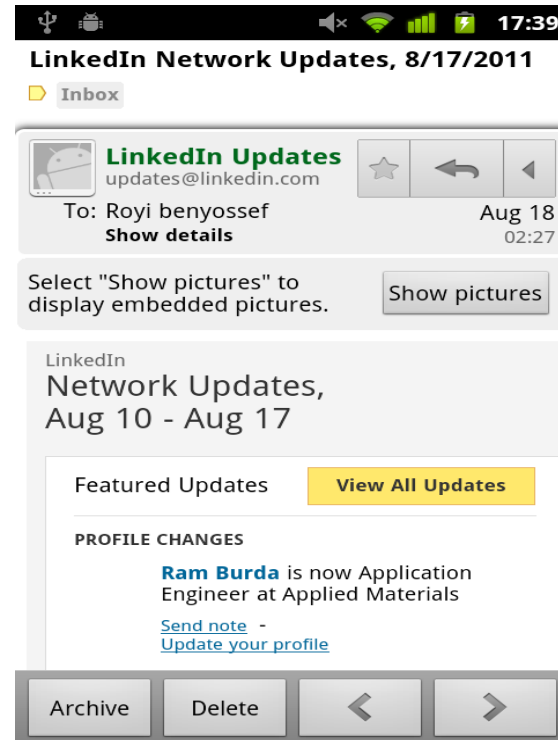


- An *activity* represents a single screen with a user interface (Java file + xml layout file).
- An application usually consists of multiple activities that are loosely bound to each other.
- One Activity is flagged as “main” and it is started at the application launch time.
- An Activity can launch other activities to create an app UI workflow.
- The Activity has an Intent attribute that determines how android treats it.

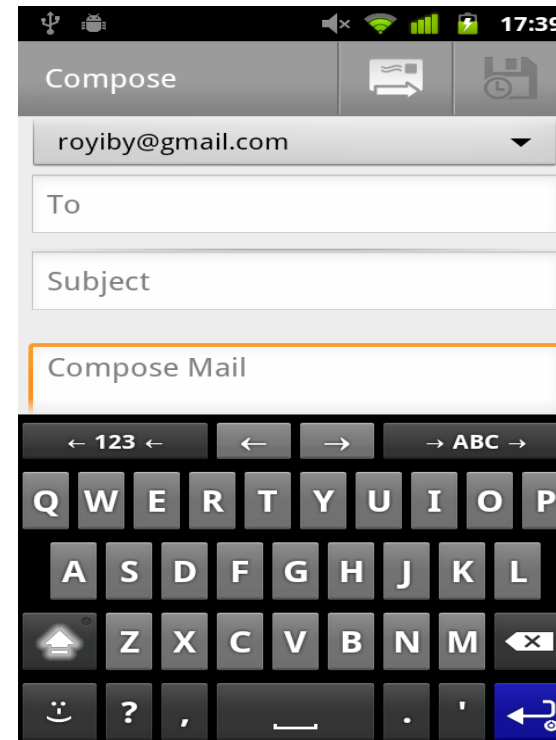
Each and every one of these is an Activity!



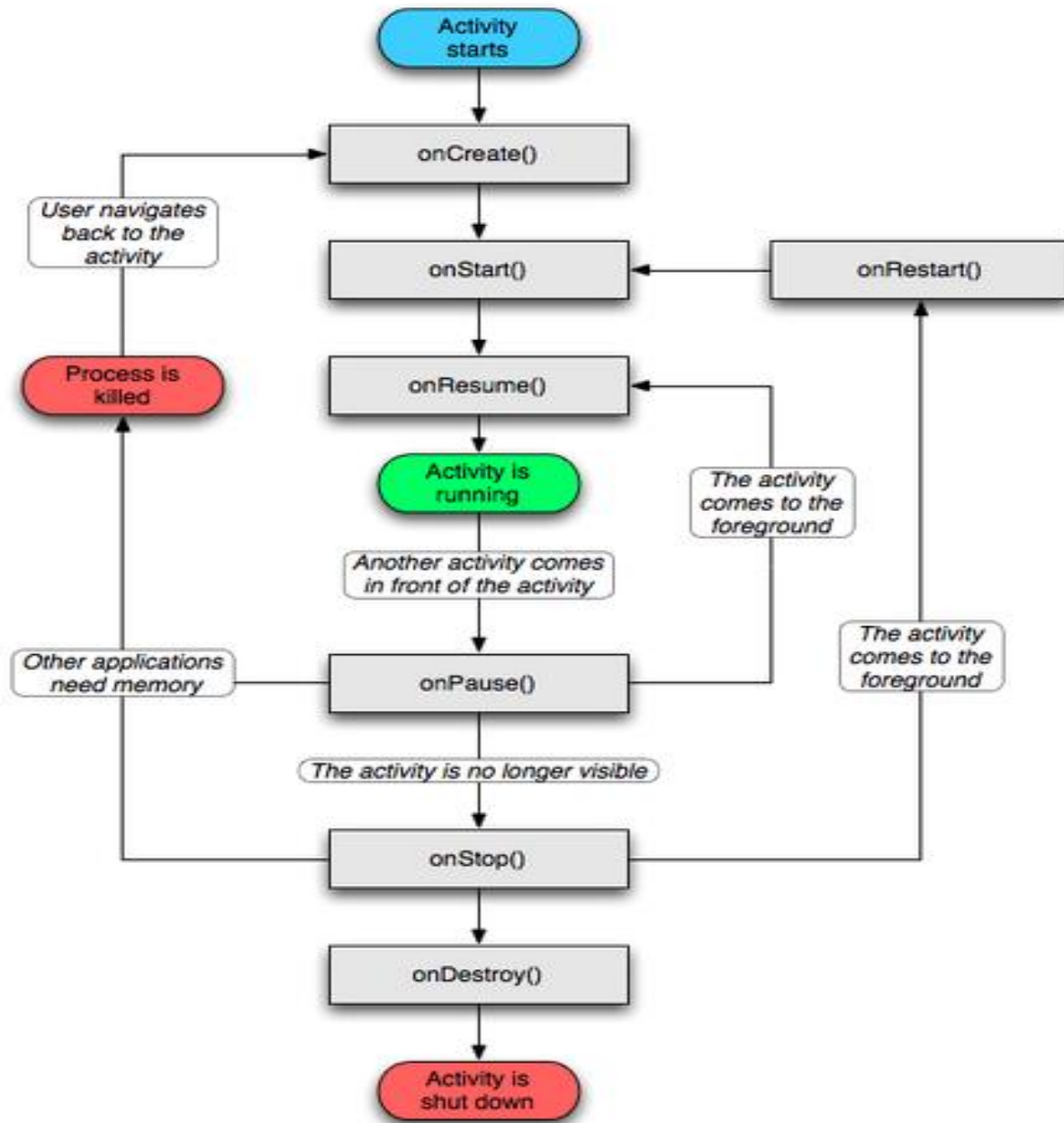
A screen that displays a list of emails.



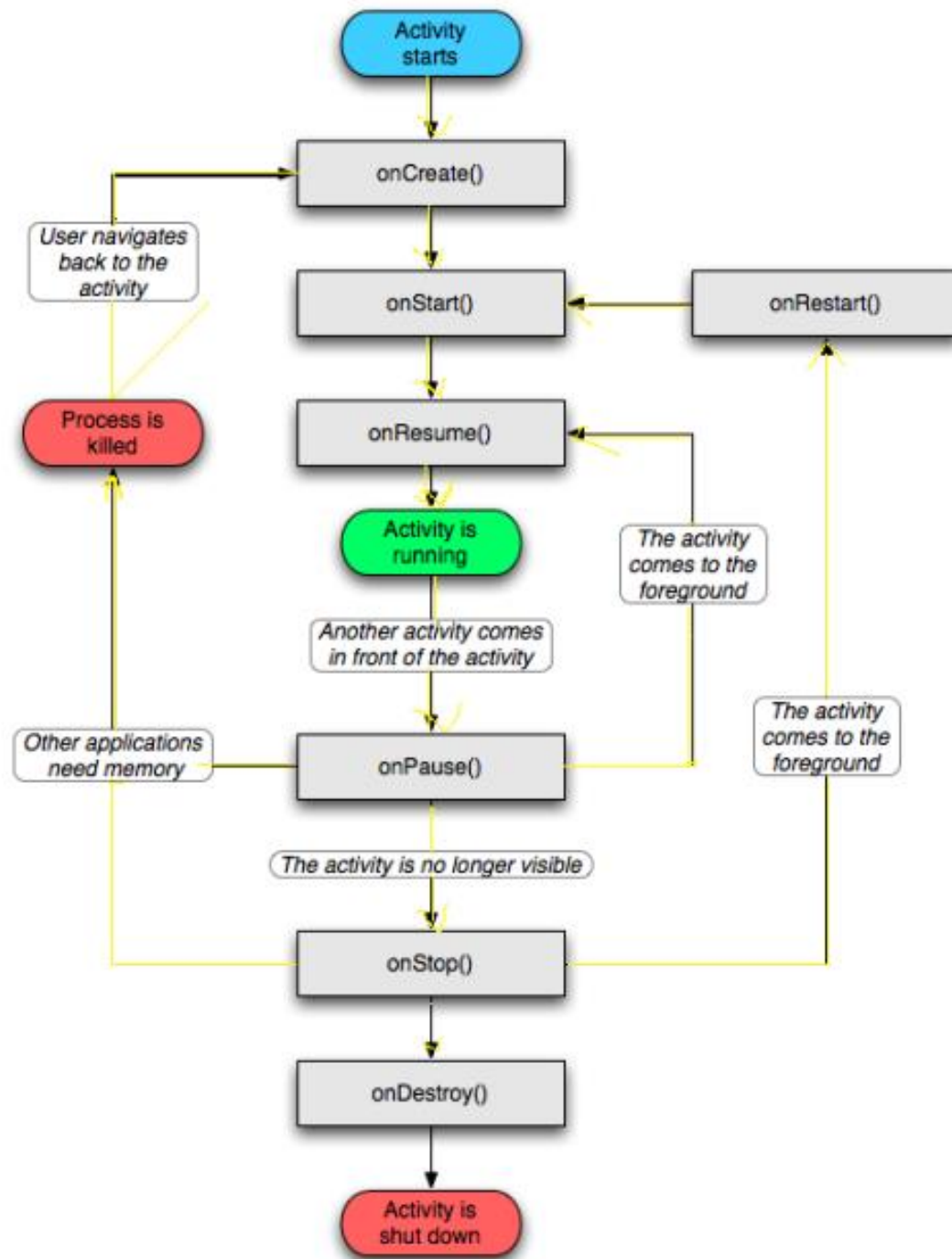
A screen that displays a single email.

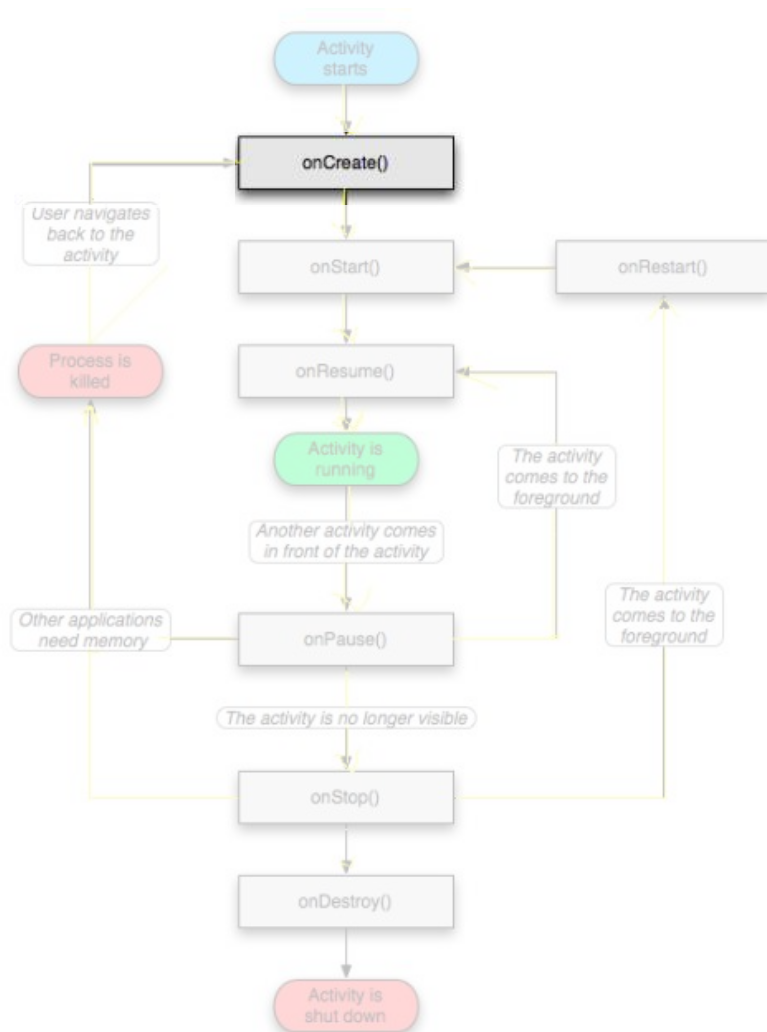


A screen that enables you to compose an email.



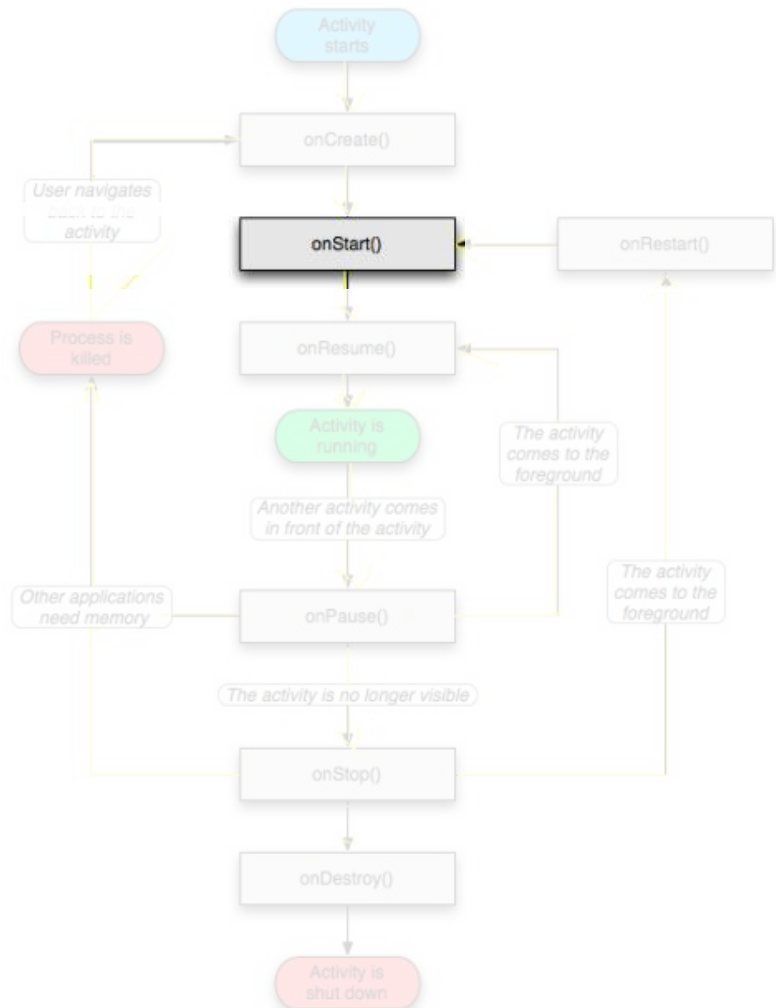
- There are 6 stages in the Android Activity life cycle.
- These stages allow you to perform tasks in their correct time of the workflow.





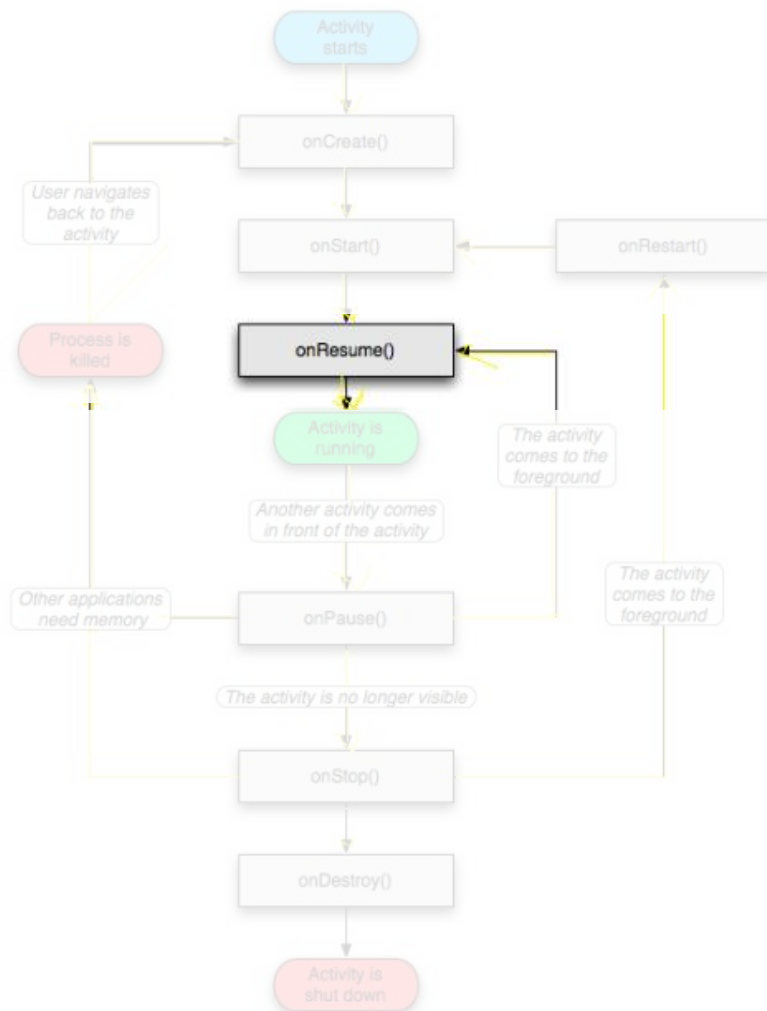
➤ onCreate()

- Called when the activity is created
- Should contain the initialization operations
- Has a Bundle parameter
- If `onCreate()` successfully terminates, it calls `onStart()`



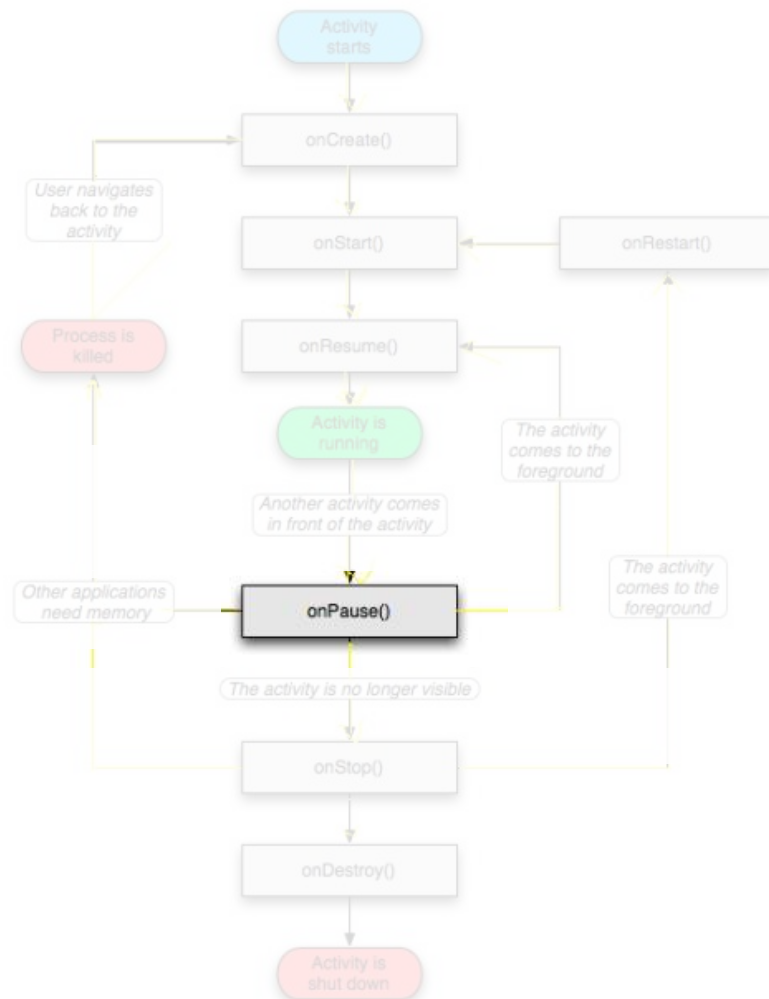
➤ onStart()

- Called when onCreate() terminates
- Called right before it is visible to user
- If it has the focus, then onResume() is called
- If not, onStop() is called



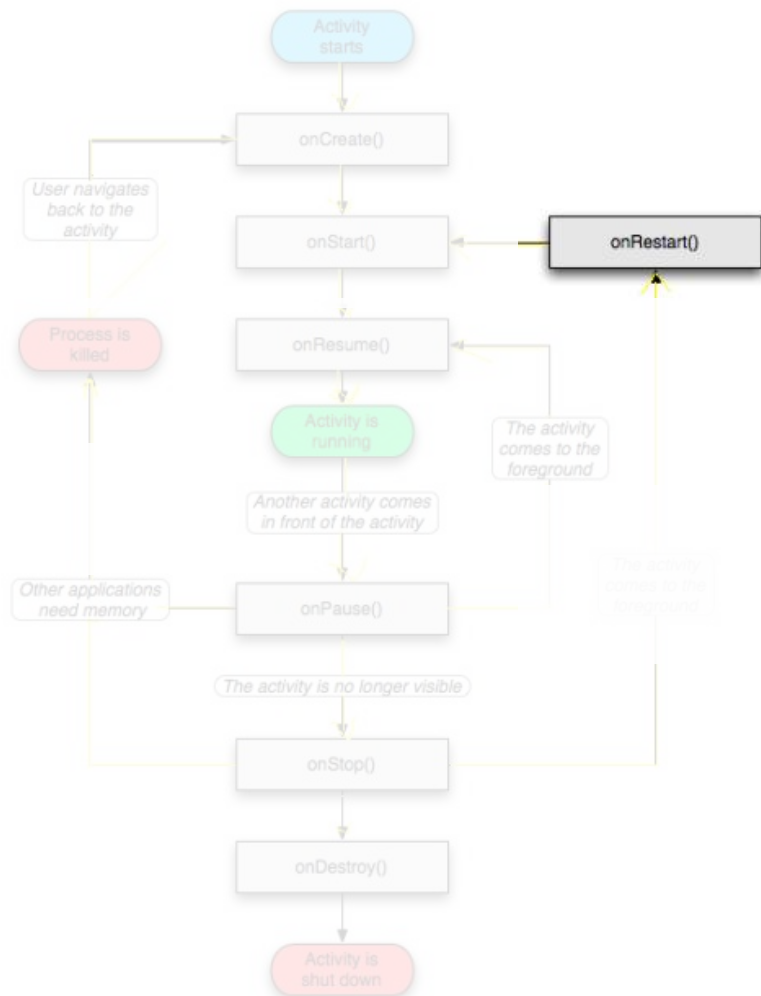
➤ OnResume()

- Called when the activity is ready to get input from users
- Called when the activity is resumed too
- If it successfully terminates, then the Activity is RUNNING



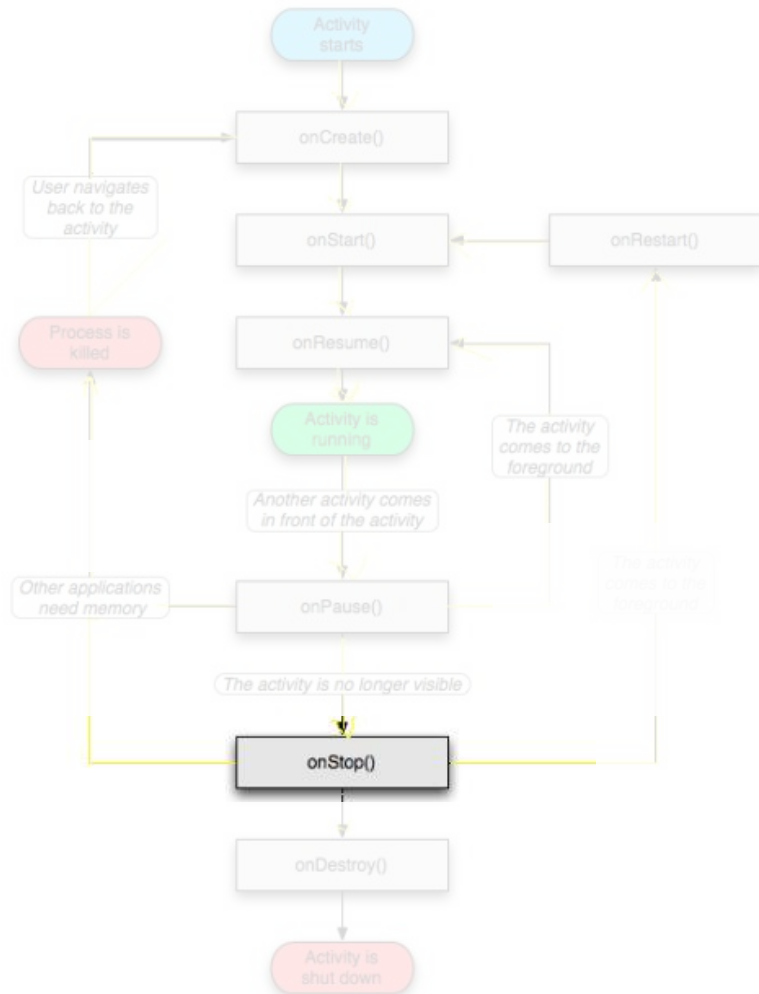
➤ onPause()

- Called when another activity comes to the foreground, or when someone presses back
- Commit unsaved changes to persistent data
- Stop cpu-consuming processes
- Make it fast



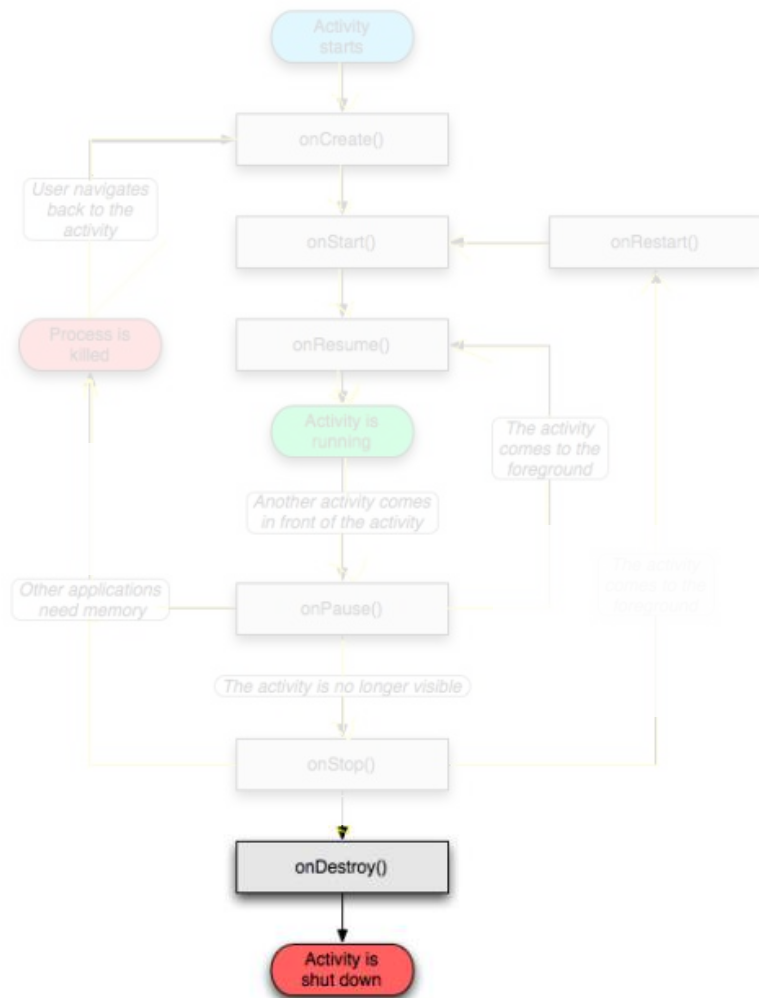
➤ OnRestart()

- Similar to `onCreate()`
- We have an activity that was previously stopped



➤ OnStop()

- Activity is no longer visible to the user
- Could be called because:
 - the activity is about to be destroyed
 - another activity comes to the foreground



➤ OnDestroy()

- The activity is about to be destroyed
- Could happen because:
- The systems need some stack space
 - Someone called `finish()` method on this activity
 - Could check with `isFinishing()`