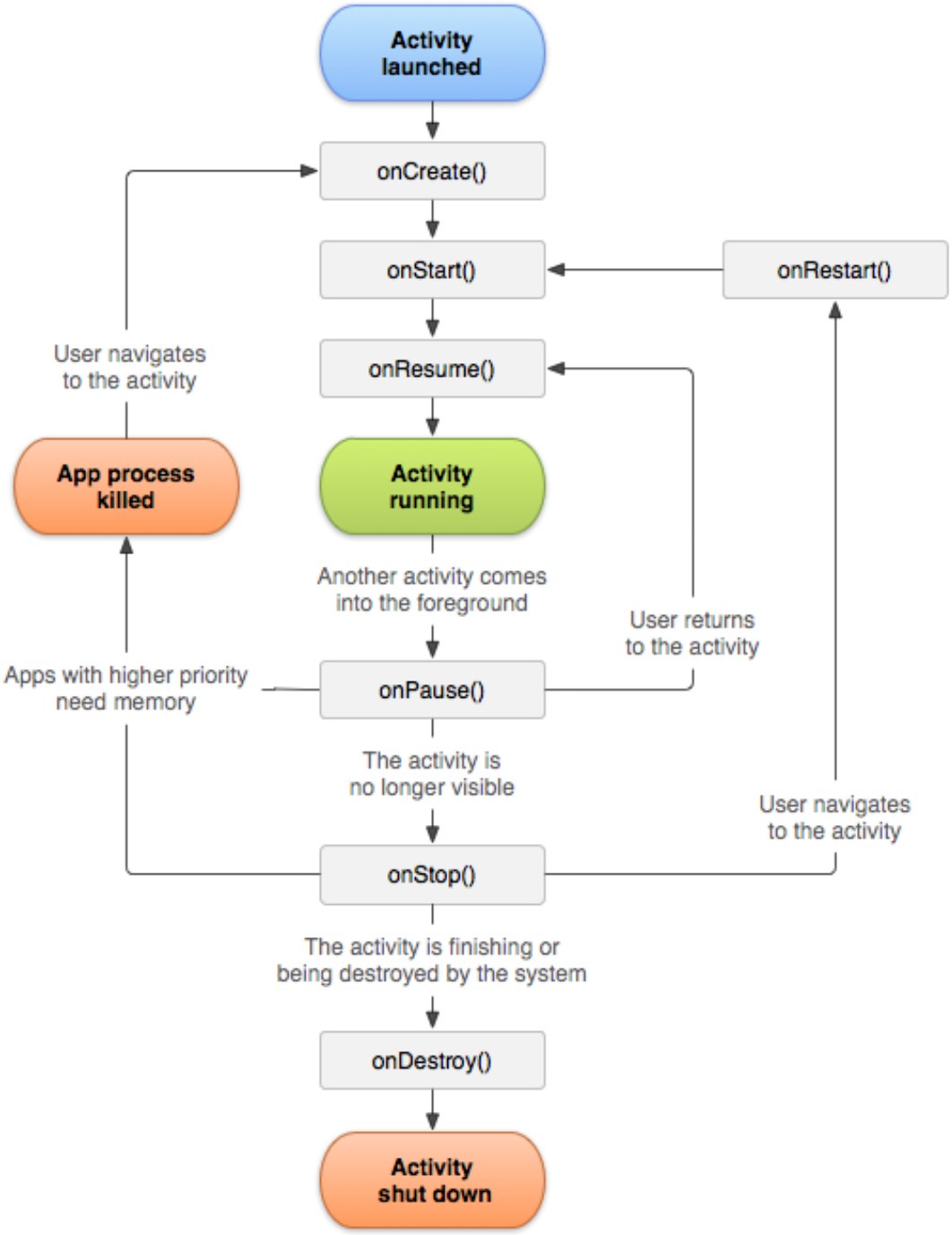
Activities

Dr Tadhg O’Sullivan

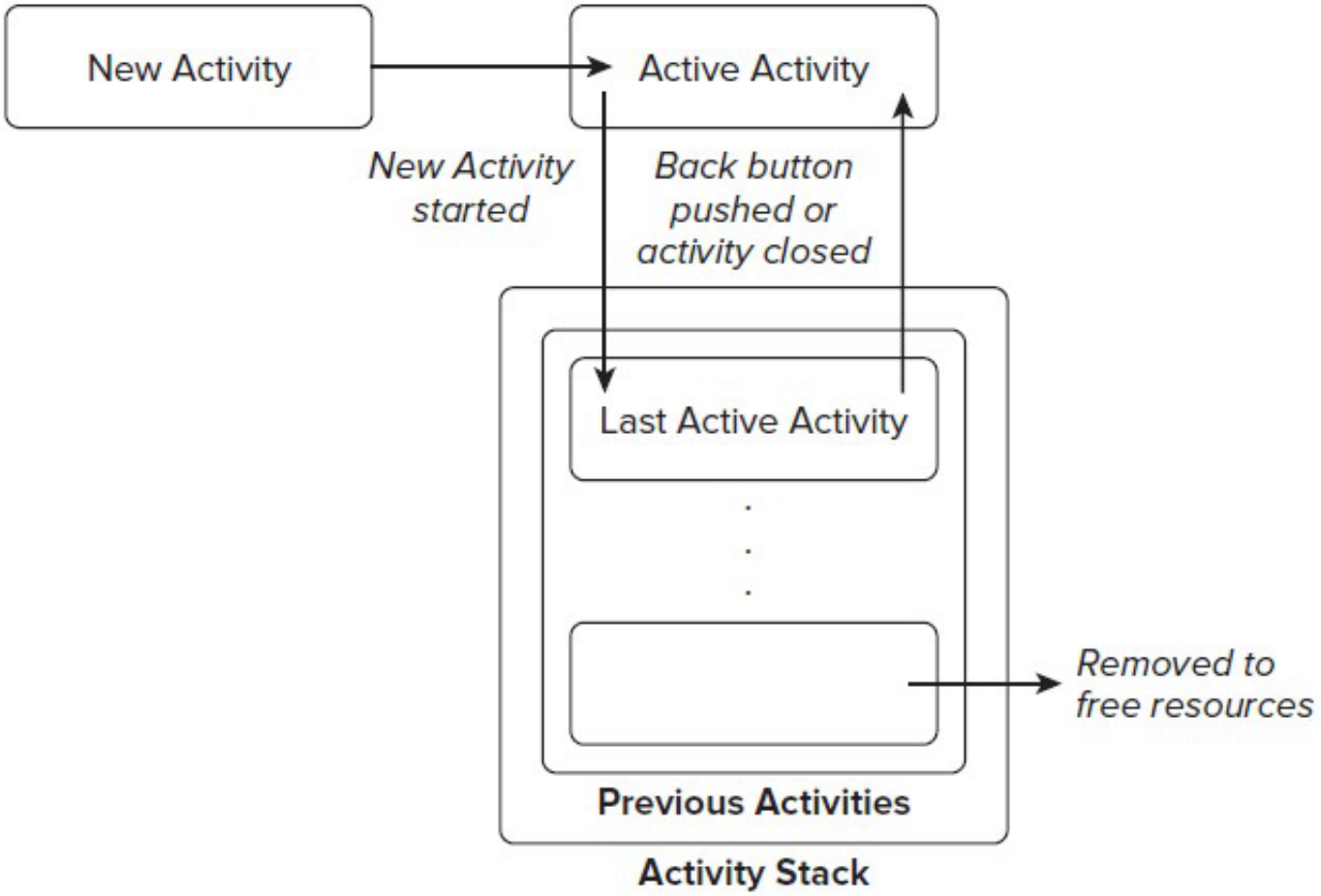
Activity Lifecycle

- Active – running in a foreground
- Paused – started, running and visible, but notification or popup overlaying part of the screen
- Stopped – started and running, but got hidden by other activities
- Inactive/dead – either never started (phone reset) or killed due to lack of memory



http://developer.android.com/guide/components/activities.html

Activity Stack



© Reto Meier

Developer Tip on Activities

- To ensure seamless experience, make transition of states invisible to the user, *i.e.* It should make no difference to user whether activity got paused and resumed, stopped and restarted again.
- We do so by saving activity's state and restoring it again

Working with Activity State

- onCreate()
- onCreate(Bundle savedInstanceState)
- onRestoreInstanceState(Bundle savedInstanceState)
- onRestart()
- onStart()
- onResume()
- onSaveInstanceState(Bundle savedInstanceState)
- onPause()
- onStop()
- onDestroy()

Remember to Clean up

- onCreate() - upon first startup
- onCreate(Bundle...) - restoring UI state after it got destroyed

- onDestroy() - end of full lifetime. End threads, close DB connections, etc.

Save/Restore Activity State

- onRestoreInstanceState(Bundle savedInstanceState) – invoked upon activity's recreation
- onSaveInstanceState(Bundle savedInstanceState) – insurance when activity stops/destroys

Changes of Activity State within Active Lifetime

- onPause() : Remember onPause() will be called when our activity is partially covered such as due to a Popup window, Alarms, Notifications, etc
- OnResume() : will be called when our activity comes back to the front.
- Also we can use these methods to develop a pause function in our app e.g. for games

Stopped, but not destroyed

- onRestart() : Deal with your app being restarted.
- onStart() : Called on either Create or Restart
- onStop(): Finished state either going to clean by memory or directly destroyed.

Questions

- Please ask in the Student Forum