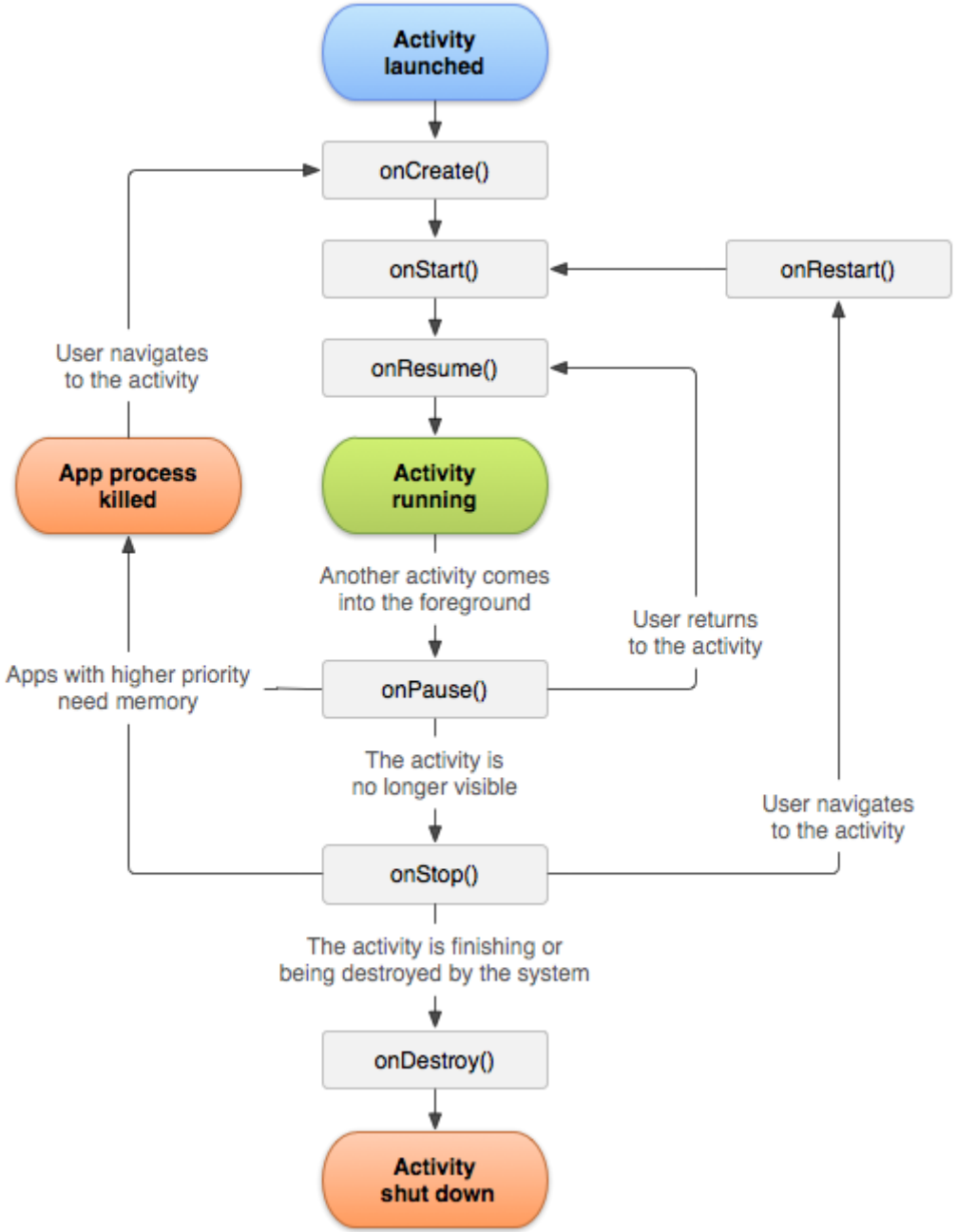
Activity Lifecycle

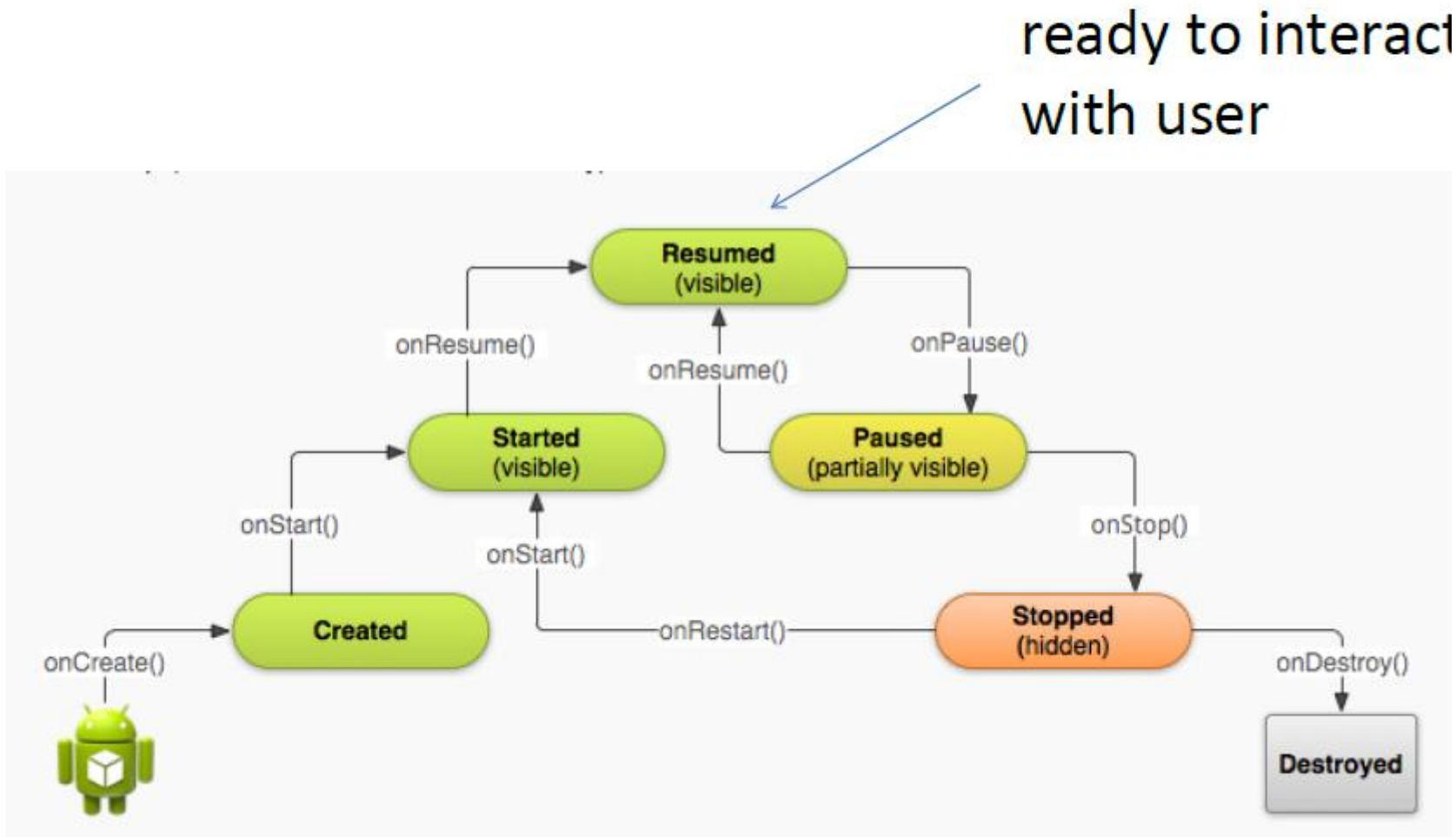


Activity Life Cycle

Starting Activities

- Android applications don't start with a call to main(String[])
- instead a series of callback methods are invoked
- each corresponds to specific stage of the Activity / application lifecycle
- callback methods also used to tear down Activity / application

Simplified Lifecycle Diagram



Understanding the Lifecycle

- Necessary to overload callback methods so your app behaves well:
- App should not crash if the user receives a phone call or switches to another app while using your app.
- App should not consume valuable system resources when the user is not actively using it.
- App should not lose the user's progress if they leave your app and return to it at a later time.
- App should not crash or lose the user's progress when the screen rotates between landscape and portrait orientation.

http://developer.android.com/training/basics/activity-lifecycle/starting.html

Primary States

- Active
  - activity is in the foreground and user can interact with it
- Paused
  - activity partially obscured by another activity and user cannot interact with it (for example when working with a menu or dialog)
- Stopped
  - activity completely hidden and not visible to user. It is in the background.
  - Activity instance and variables are retained but no code is being executed by the activity
- Dead, activity terminated (or never started)
- Two other states, Created and Started, but they are transitory onCreate -> onStart -> onResume