

18. Life Cycle

- 상황별 Life cycle callback method

- onCreate() > onStart() > onResume() 순으로 호출되는 경우

- (액티비티 미실행 상태) - 앱아이콘 클릭 - 액티비티 최초 실행
 - (최근 앱 목록 표시 중, 액티비티는 미실행 상태) - 액티비티 선택 - 액티비티 실행

onCreate() > onStart() > onResume() > onPause() > onStop() 순으로 호출되는 경우

- (화면 잠금 상태) - ADB 커맨드를 통한 액티비티 실행

onRestart() > onStart() > onResume() 순으로 호출되는 경우

- (액티비티 foreground 상태에서 화면 잠금 상태) - 화면 잠금 해제(홈이나 전원 버튼, 필요하다면 비밀번호 입력 등) - 액티비티 실행
 - (액티비티를 실행한 적이 있으며 숨겨져 있는 상태) - 액티비티 실행(앱 아이콘 클릭 등) - 액티비티 실행

onPause()만 호출되는 경우

- (액티비티 foreground 상태) - 다른 액티비티 실행

onPause() > onStop() 순으로 호출되는 경우

- (액티비티 foreground 상태) - 전원 버튼 누르기 - 화면 잠금
 - (액티비티 foreground 상태) - 홈 버튼 누르기 - 런처 홈으로 이동

onPause() > onStop() > onDestroy() 순으로 호출되는 경우

- (액티비티 foreground 상태) - 뒤로가기 버튼 눌러 액티비티 나가기

onDestroy()만 호출되는 경우

- (최근 앱 목록 표시 중) - 액티비티 제거하기(밀어내거나 'X' 버튼)

- <http://sunphiz.me/wp/archives/3005>

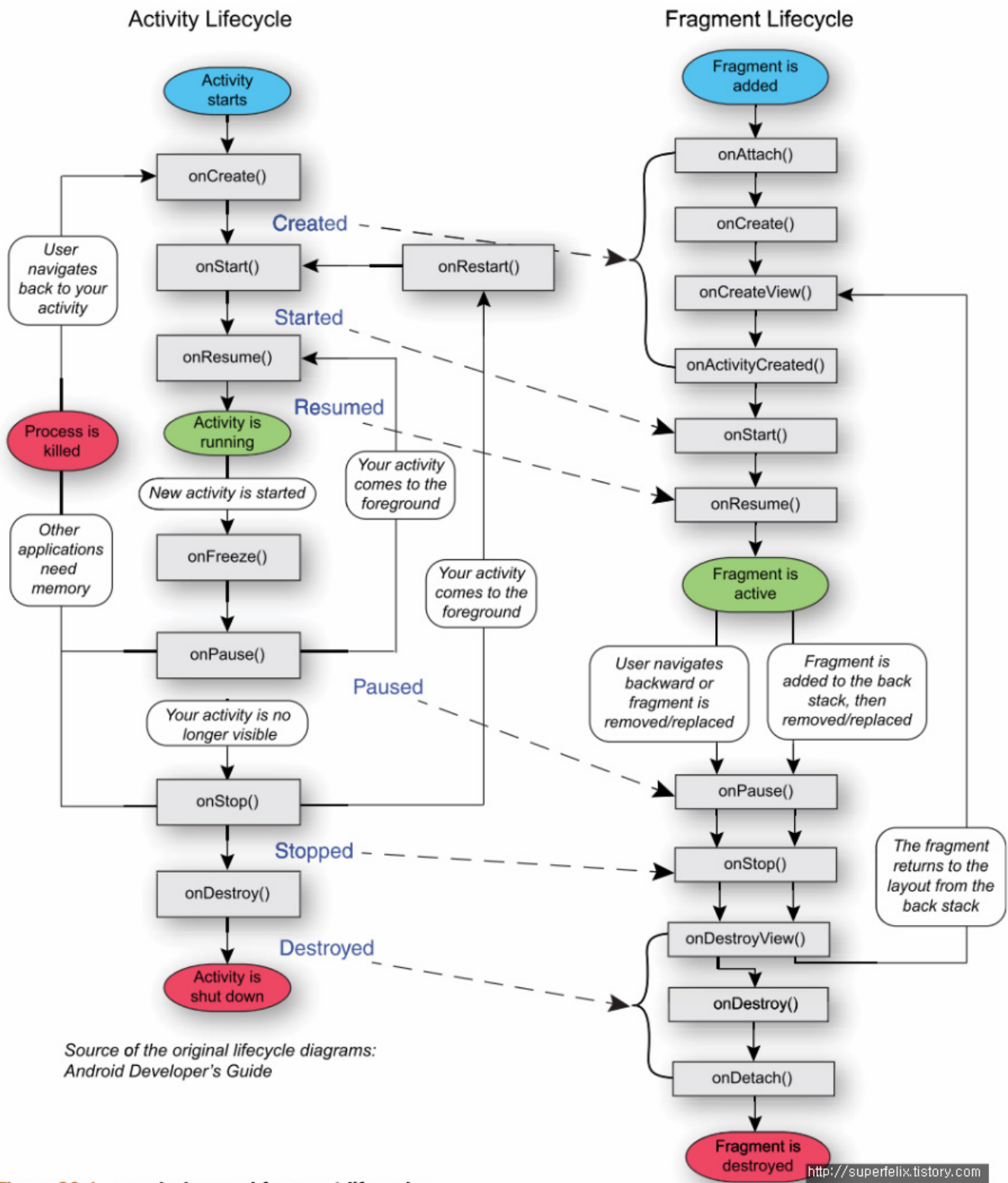


Figure 20.1 Activity and fragment lifecycles

- Activity Life Cycle - Activity Lifecycle

Activities in the system are managed as [activity stacks](#). When a new activity is started, it is usually placed on the top of the current stack and becomes the running activity -- the previous activity always remains below it in the stack, and will not come to the foreground again until the new activity exits. There can be one or multiple activity stacks visible on screen.

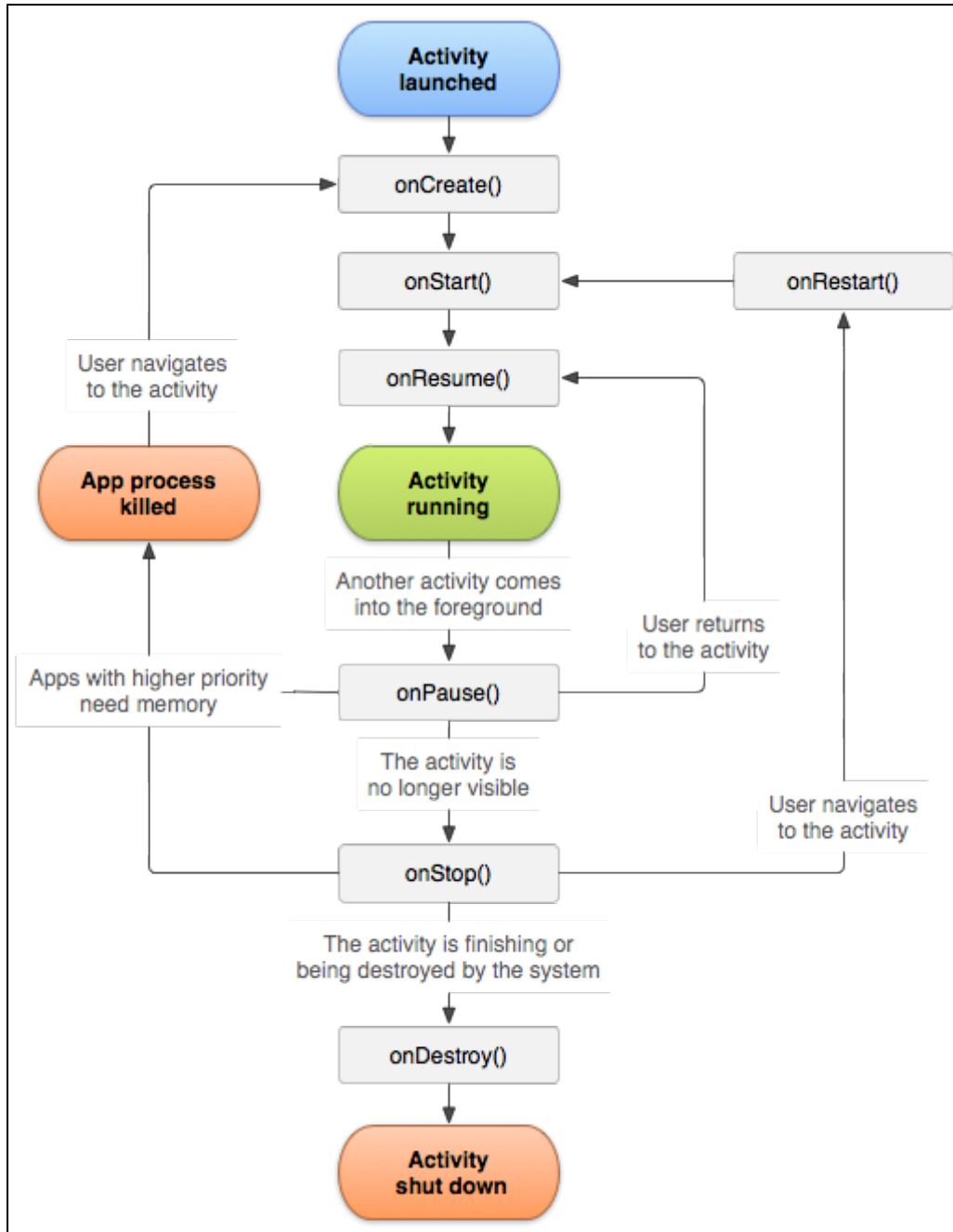
An activity has essentially four states:

- If an activity is in the foreground of the screen (at the highest position of the topmost stack), it is *active* or *running*. This is usually the activity that the user is currently interacting with.
- If an activity has lost focus but is still presented to the user, it is *visible*. It is possible if a new non-full-sized or transparent activity has focus on top of your activity, another activity has higher position in multi-window mode, or

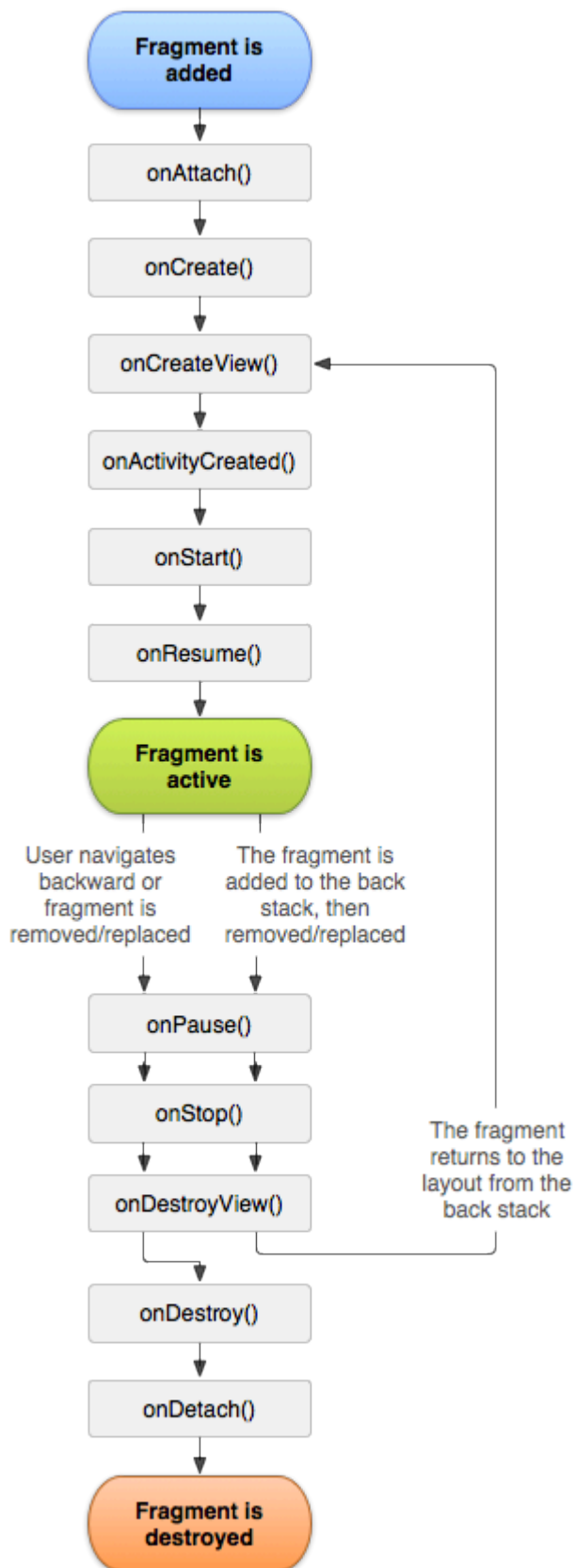
the activity itself is not focusable in current windowing mode. Such activity is completely alive (it maintains all state and member information and remains attached to the window manager).

- If an activity is completely obscured by another activity, it is *stopped* or *hidden*. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.
- The system can drop the activity from memory by either asking it to finish, or simply killing its process, making it *destroyed*. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

The following diagram shows the important state paths of an Activity. The square rectangles represent callback methods you can implement to perform operations when the Activity moves between states. The colored ovals are major states the Activity can be in.



<https://developer.android.com/reference/android/app/Activity.html?hl=ko>



onCreate() The system calls this when creating the fragment. Within your implementation, you should initialize essential components of the fragment that you want to retain when the fragment is paused or stopped, then resumed.

onCreateView() Called to create the view hierarchy associated with the fragment.

The system calls this when it's time for the fragment to draw its user interface for the first time. To draw a UI for your fragment, you must return a View from this method that is the root of your fragment's layout. You can return null if the fragment does not provide a UI.

Refer [Fragments Life Cycle](#)