Android for .NET Developers Series
Building Apps with Android Studio
Activity Lifecycle

Jim Wilson

jimw@jwhh.com

@hedgehogjim

http://facebook.com/hedgehogjim

http://hedgehogjim.wordpress.com

pluralsight

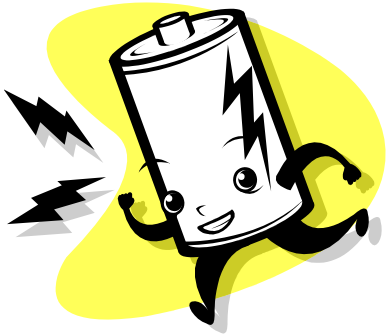
hardcore developer training

Outline

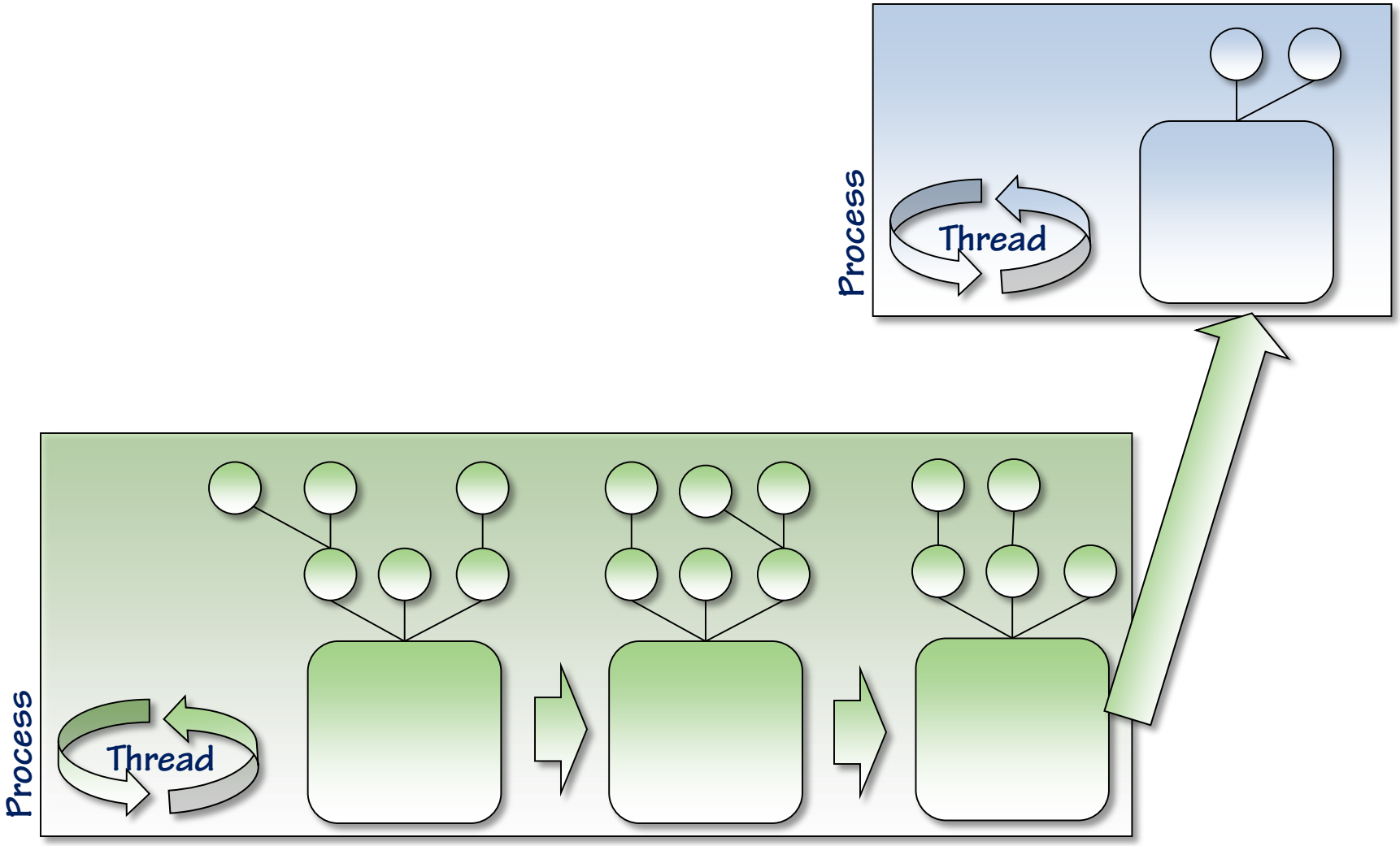
- Mobile resource challenge
- Android resource management
- Activity states
- Activity lifecycle callbacks
- Device orientation and Activity state

The mobile resource challenge

- **Android must manage resources more carefully than non-mobile OS's**
  - Resources are comparatively very limited
    - Memory limits are absolute (no paging)
    - Need to closely manage power
  - Traditional OS's tie resource lifetime to processes/threads
    - Resources remain held even when the user isn't interested

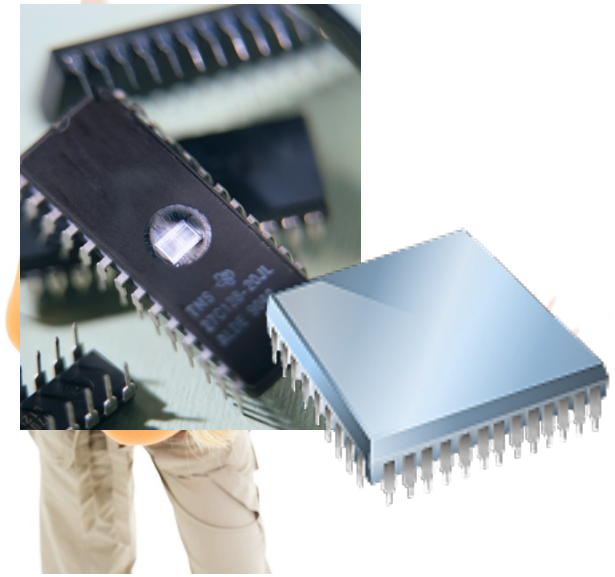


Traditional resource management

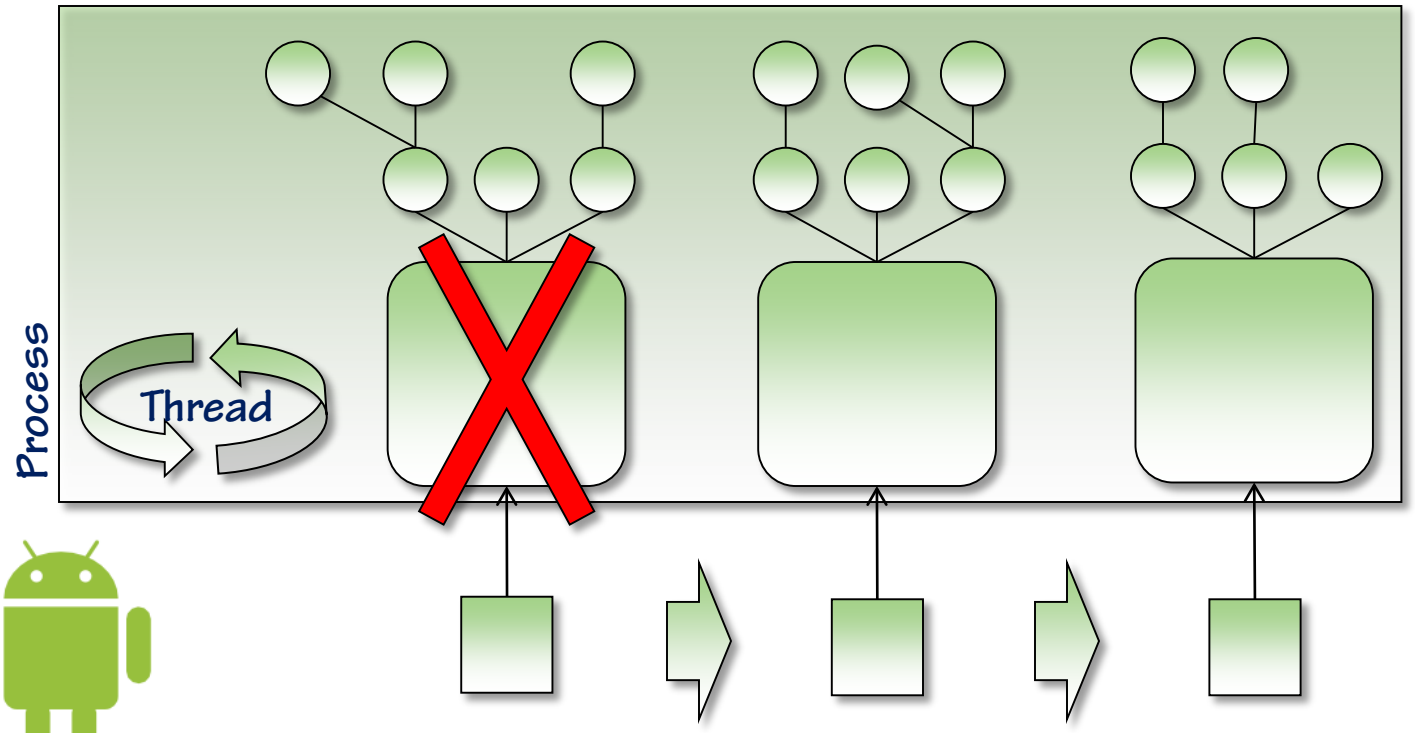


Moving beyond processes & threads

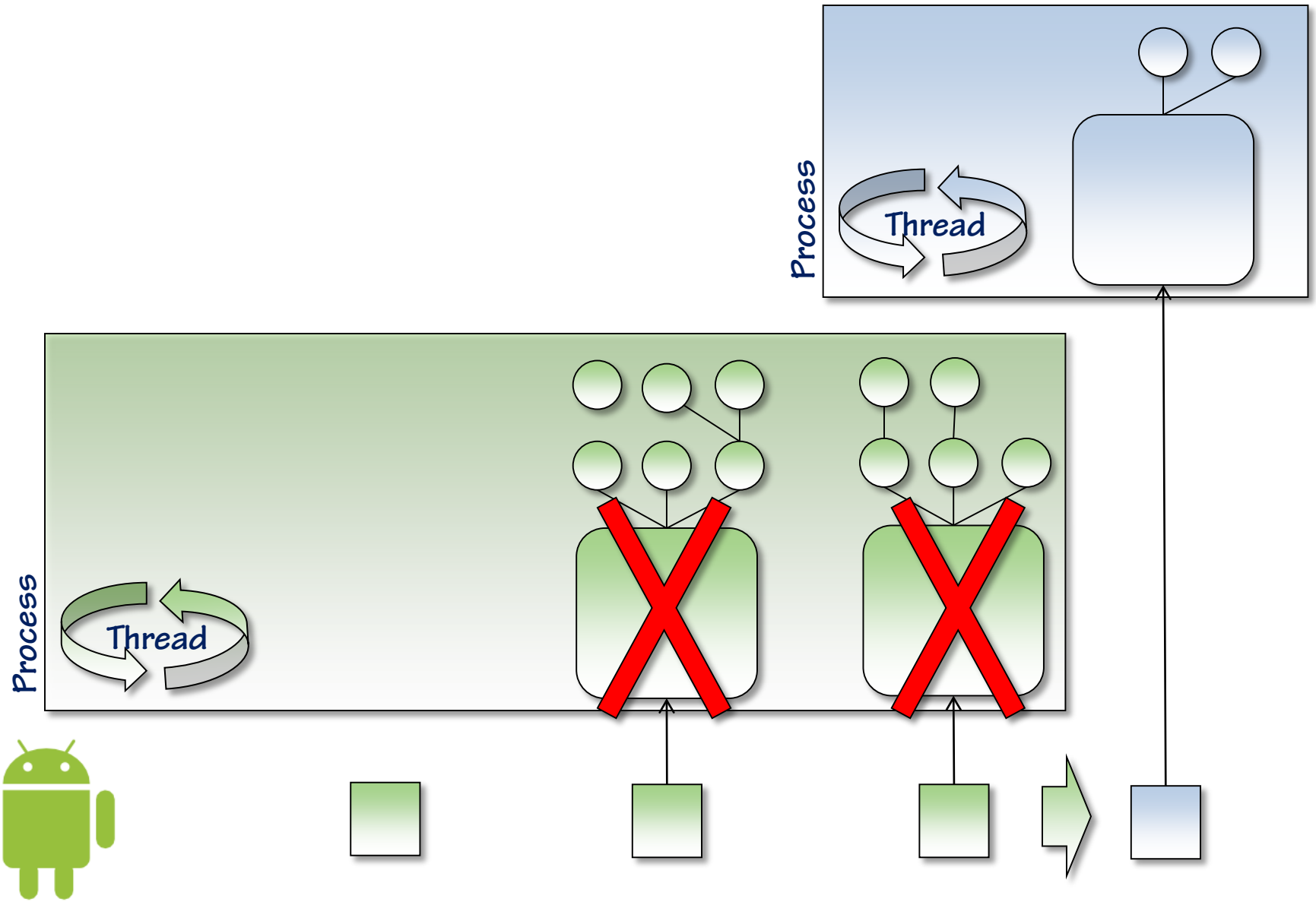
- **Android manages resources at the component level**
  - An Activity's right to resources is tied to user interaction
    - An Activity loses right to CPU when user moves to another Activity
    - An Activity may lose memory resources when user moves to another Activity
  - Details must be hidden from the user
    - User must be able to move between Activities freely
    - The fact that resources are lost should not impact user



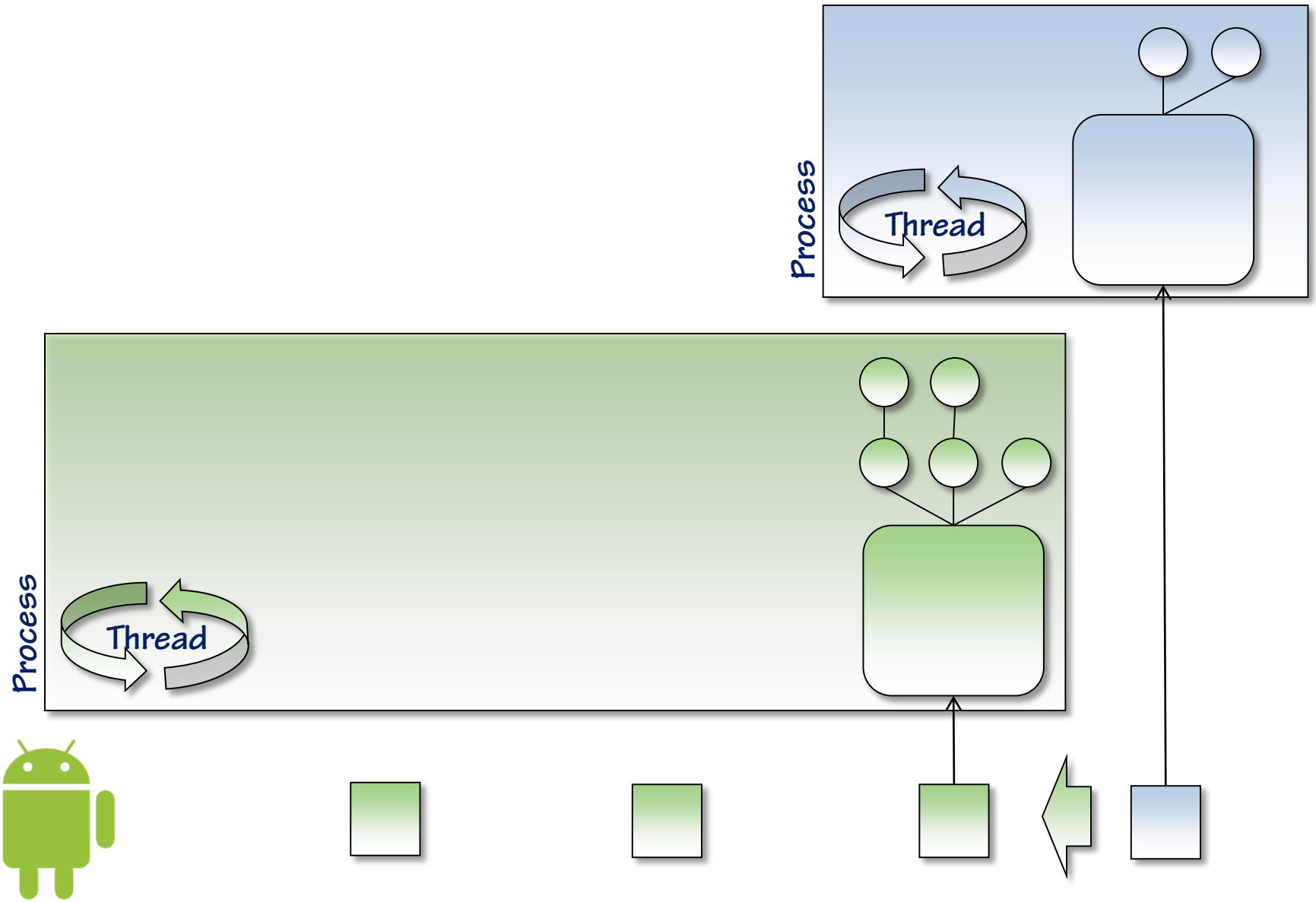
Android resource management



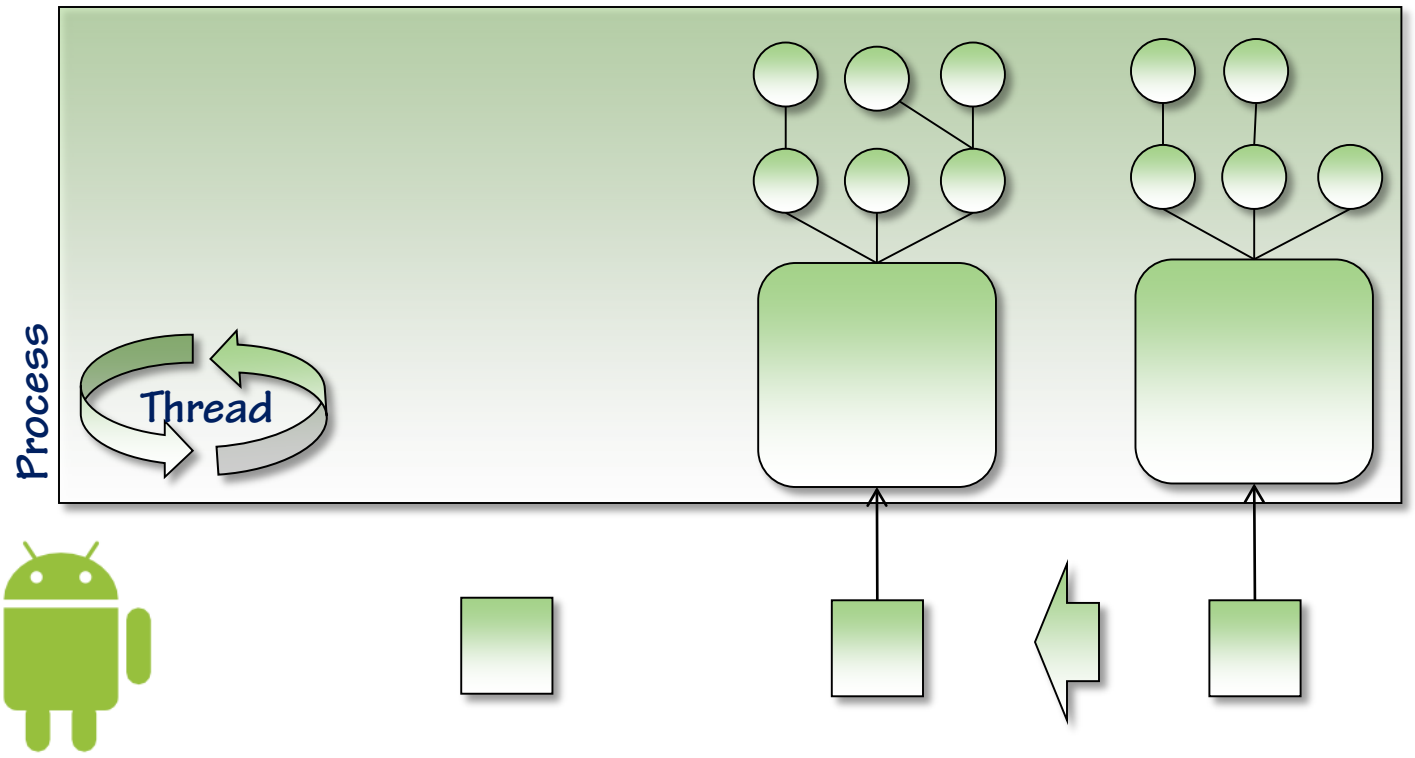
Android resource management



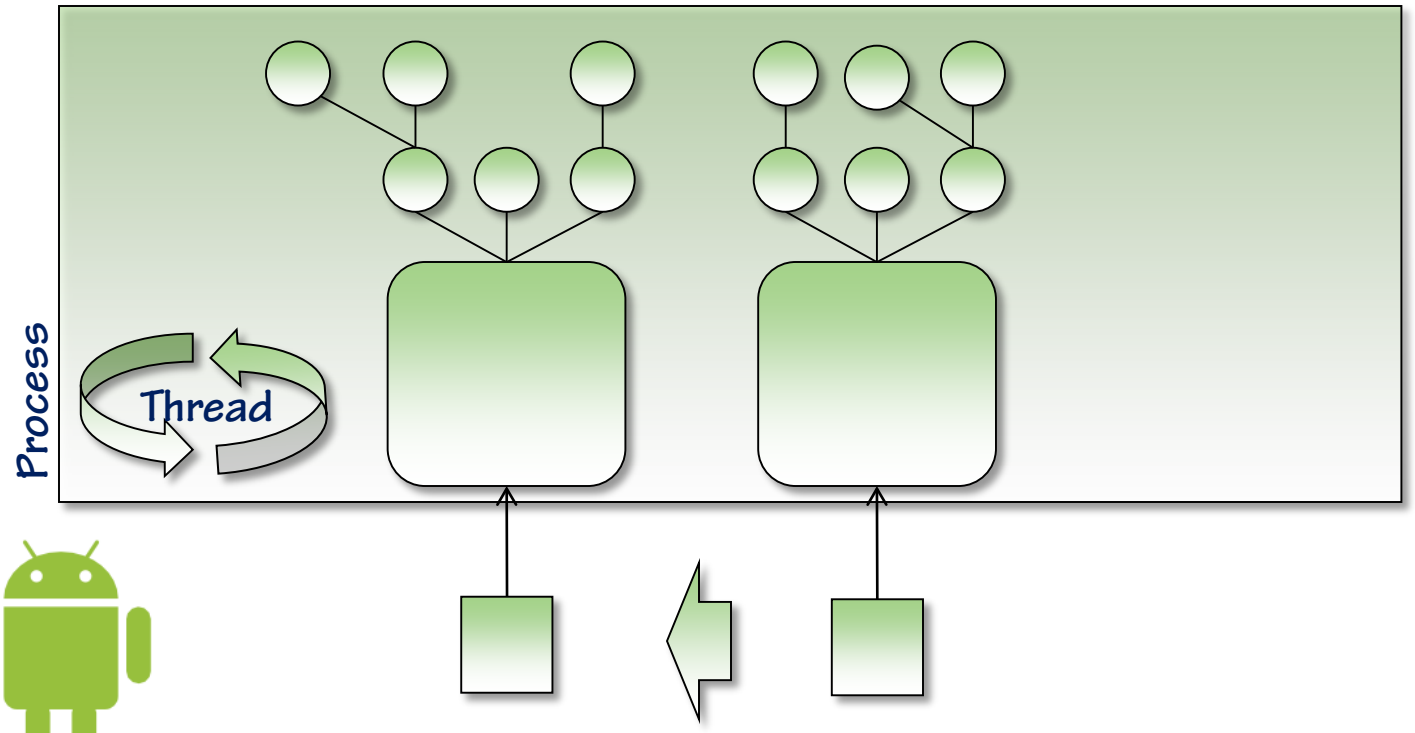
Android resource management



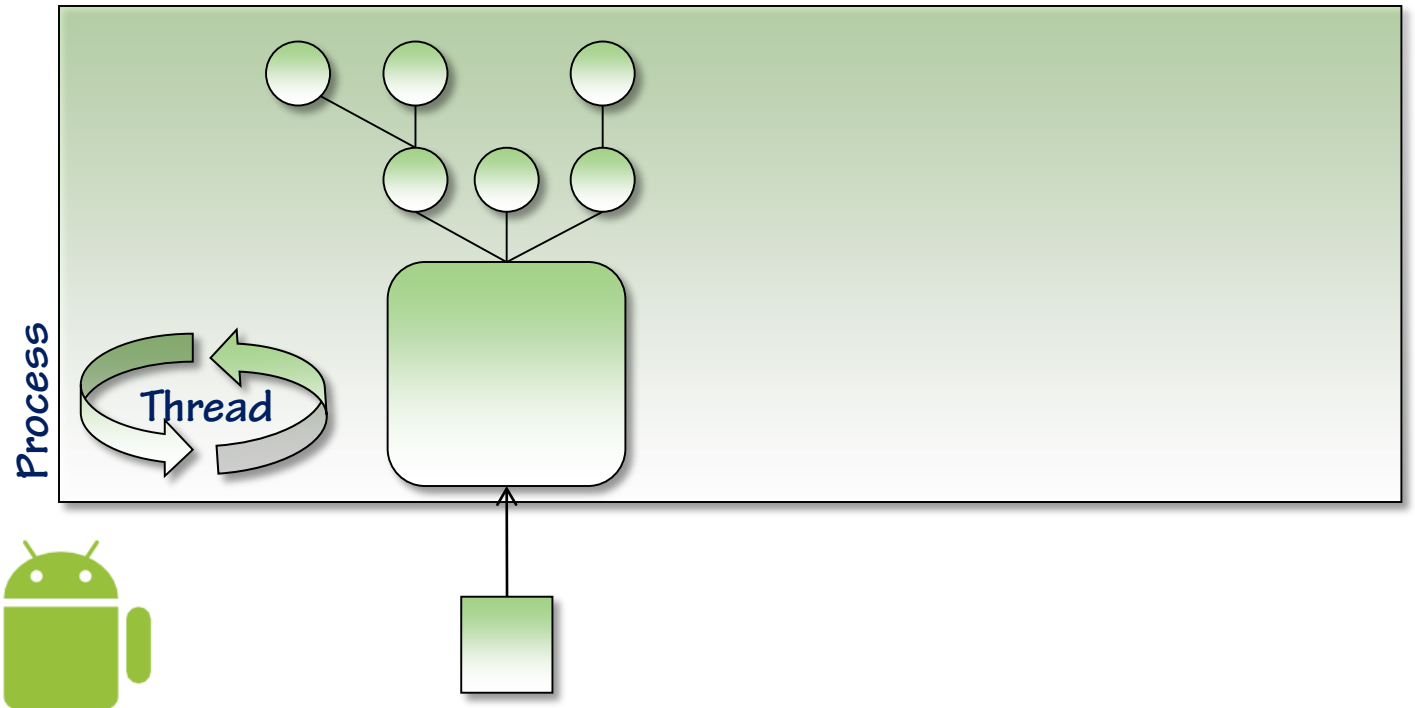
Android resource management



Android resource management



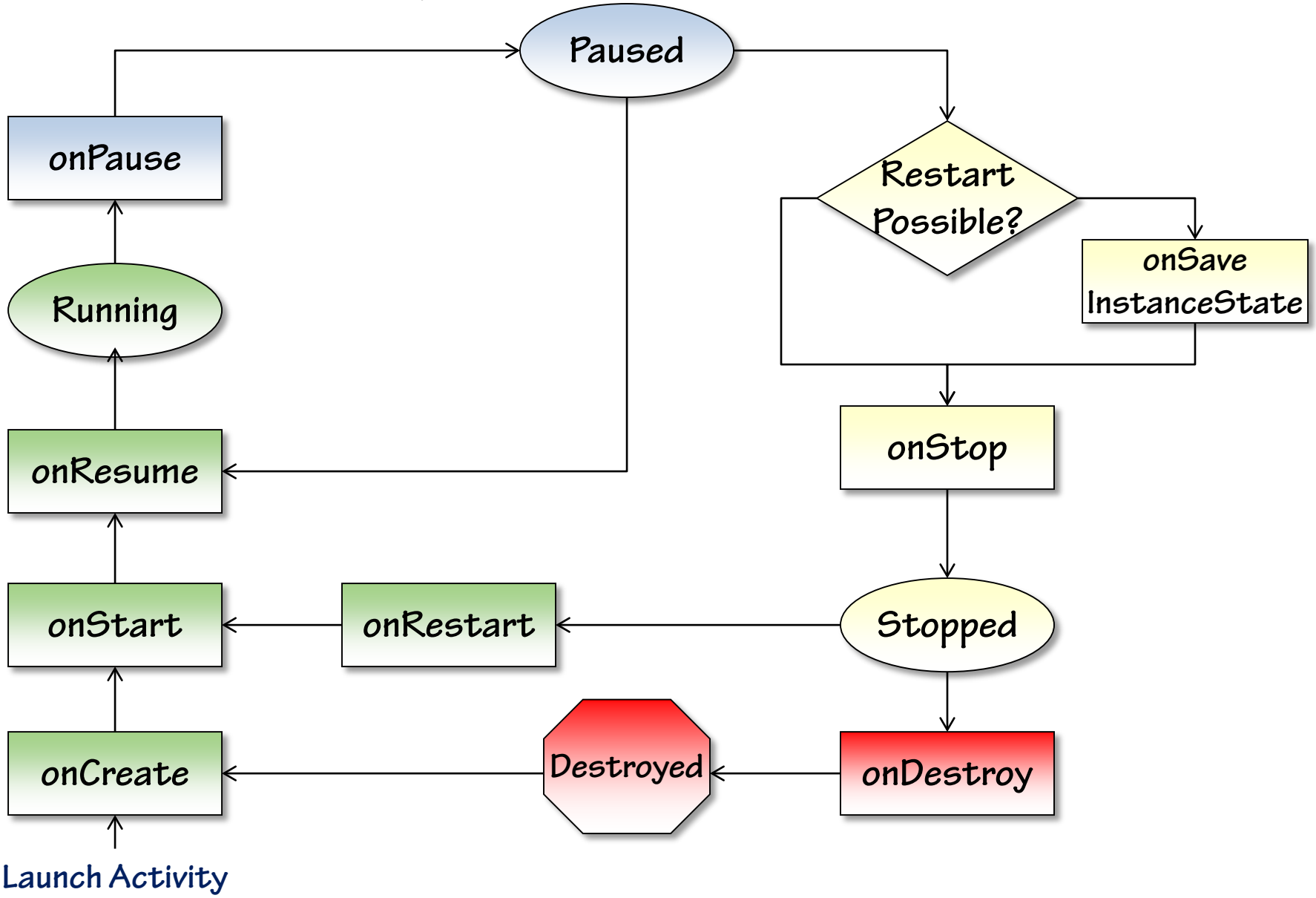
Android resource management



Activity states

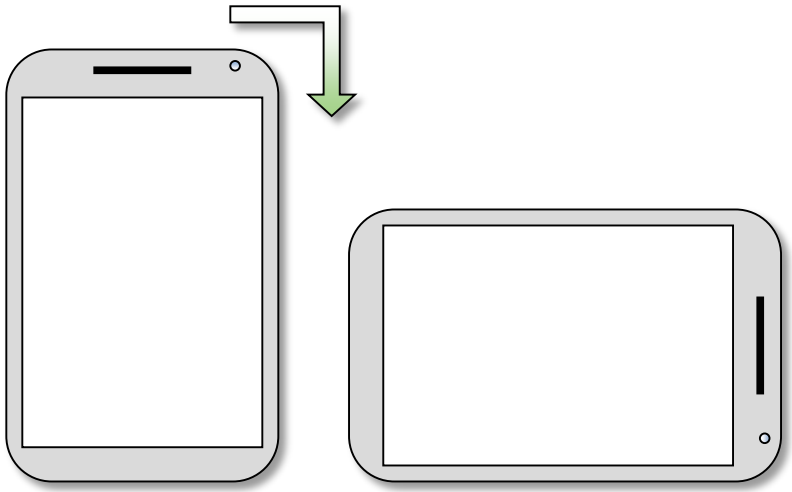
- **An Activity's access to resources depends on its current state**
  - Running state (also known as Active or Resumed)
    - Activity is in the foreground
    - Full access to resources
    - Will not be destroyed
  - Paused state
    - Activity is visible but not in the foreground
    - Retains memory resources
    - Limited opportunity to perform processing
    - Not likely to be destroyed
  - Stopped state
    - Activity is not visible
    - Should be prepared to lose memory resources
    - Limited opportunity to perform processing
    - Very likely to be destroyed

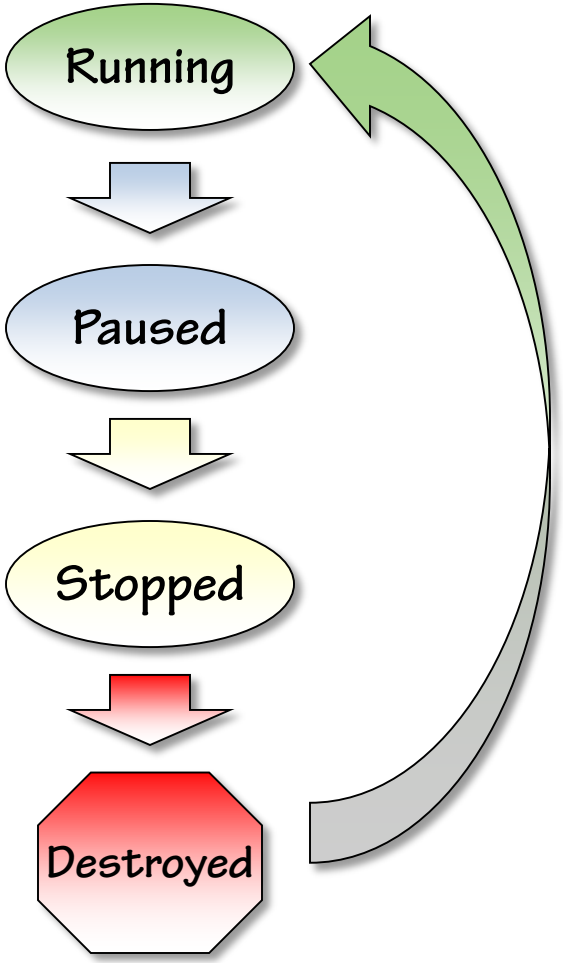
Activity state callback methods



Device orientation and state

- **Rotating the device affects Activity state**
  - On a change of orientation, Android complete tears down an Activity
    - Activity is destroyed
    - You must save your Activity's state
  - Completely recreates the Activity
    - You must restore your Activity's state





Summary

- **Android ties resource lifetime to components**
  - More fine-grain management than processes & threads
- **An Activity's right to resources is tied to user interaction**
  - Android aggressively reclaims resources when not accessed by user
- **Details of resource management hidden from users**
  - An Activity may be completely destroyed between user access
  - Activities are responsible to save and restore state
- **Activities cooperate with resource mgmt through callback methods**
  - Callback methods provide hooks for state management
- **Orientation change completely tears down and rebuilds an Activity**
  - Use callback methods to manage state