Taking a Deeper Look at Activity Interaction and Lifecycle

Jim Wilson
MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim blog.jwhh.com

What to Expect from This Module

- Application Activity Relationship
- Implicit Intents
- Activities with Results
- Activity Tasks
- Activity Lifecycle
- Activity State Management

Application Activity Relationship

Android is a component-oriented platform

- Components run within a process

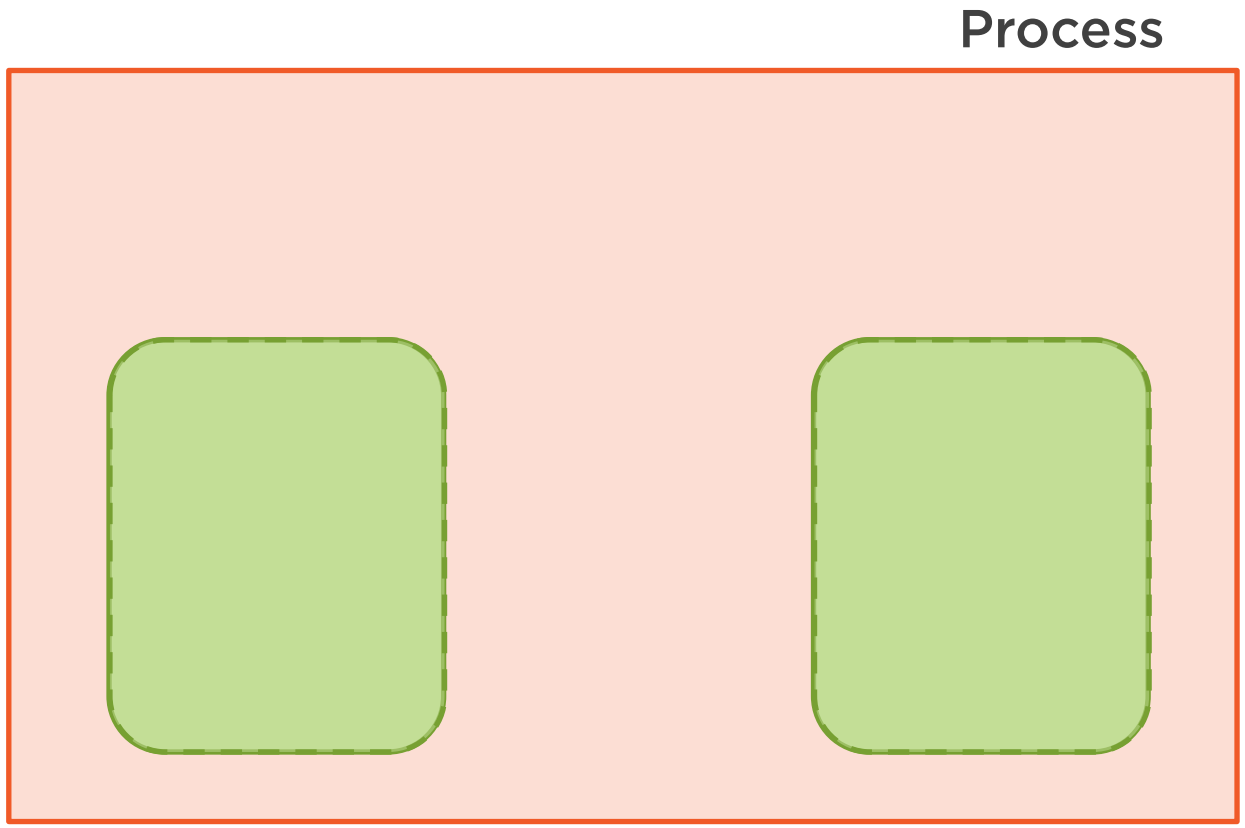
Process lifetime

- Driven by component lifetime
- Launched for first component accessed
- Terminates after last component exits

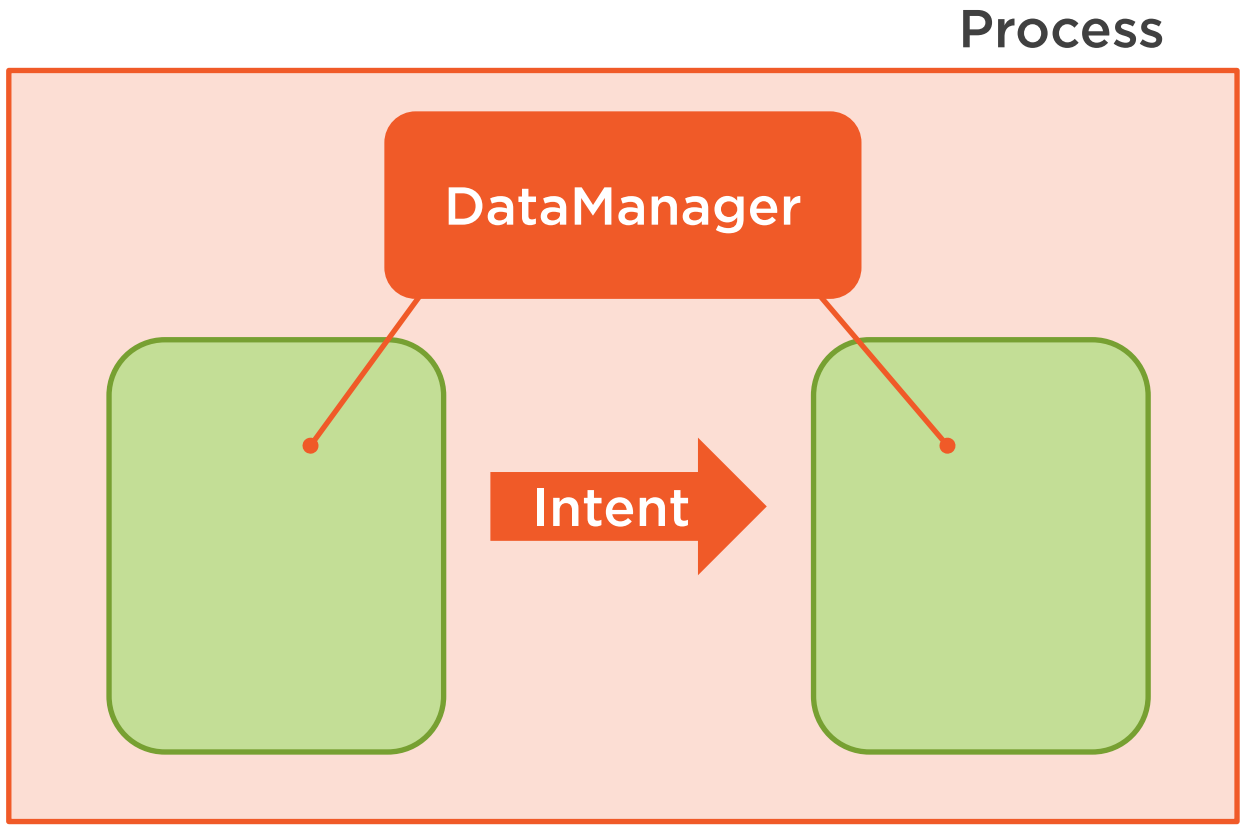
Application process

- Each app has its own process
- App components run in same process
  - When simultaneously active

Application Activity Relationship



Application Activity Relationship



Late-binding Components

Intents describe a desired operation

- Identify operation target

Explicit intent

- Target is explicitly identified
- Specify the Activity class to use

Implicit intent

- Target is implied
- Specify the Activity characteristics

Late-binding Components

Implicit intents provide late-binding

- Match is determined at runtime

System finds best match

- Often comes from another app
- Specific match may vary depending on apps installed on user device
- Prompts user if tie

Decouples sender and receiver

- Sender may not know receiver
- Receiver may not know sender

Implicit Intent Characteristics

Action

- Action string
- Many standard constants available
- Example: Intent.ACTION_VIEW
- Commonly set in Intent constructor
- Only required characteristic

Category

- Provides extended qualification
- Not normally used by sender

Implicit Intent Characteristics

Data

- URI of data to be acted upon
- Example: https://pluralsight.com
- Set with Intent.setData

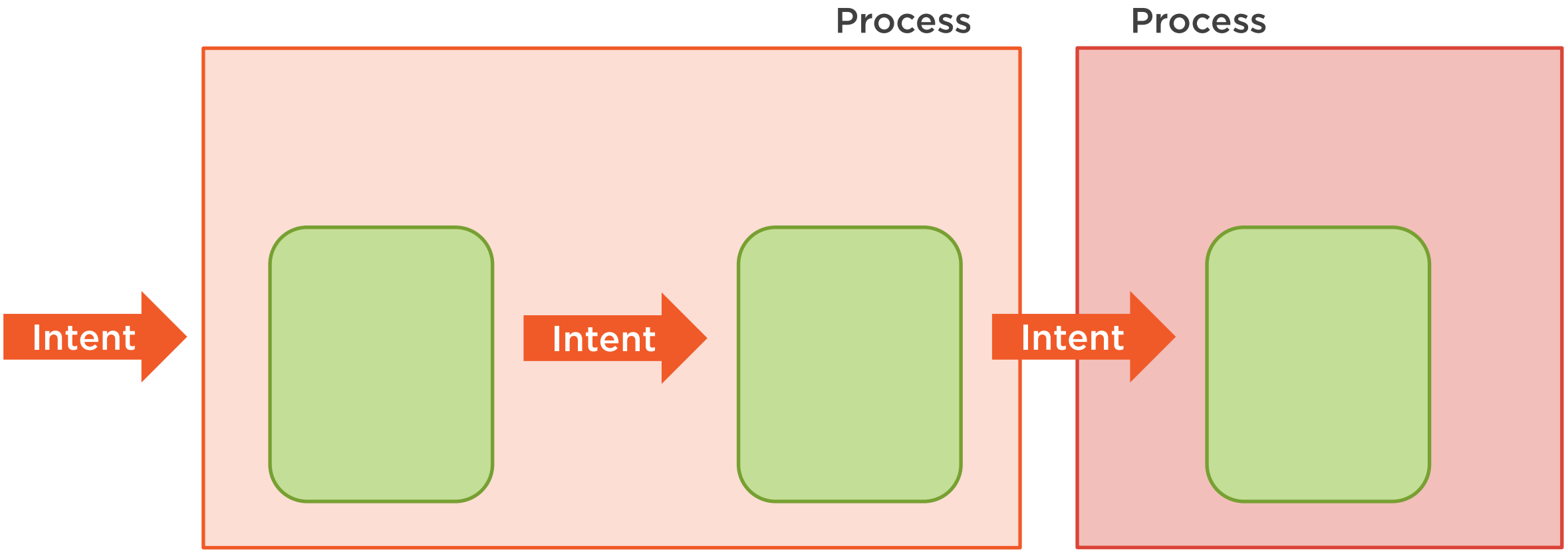
Mime type

- Common or app-specific mime type
- Example: text/html
- Set with Intent.setType

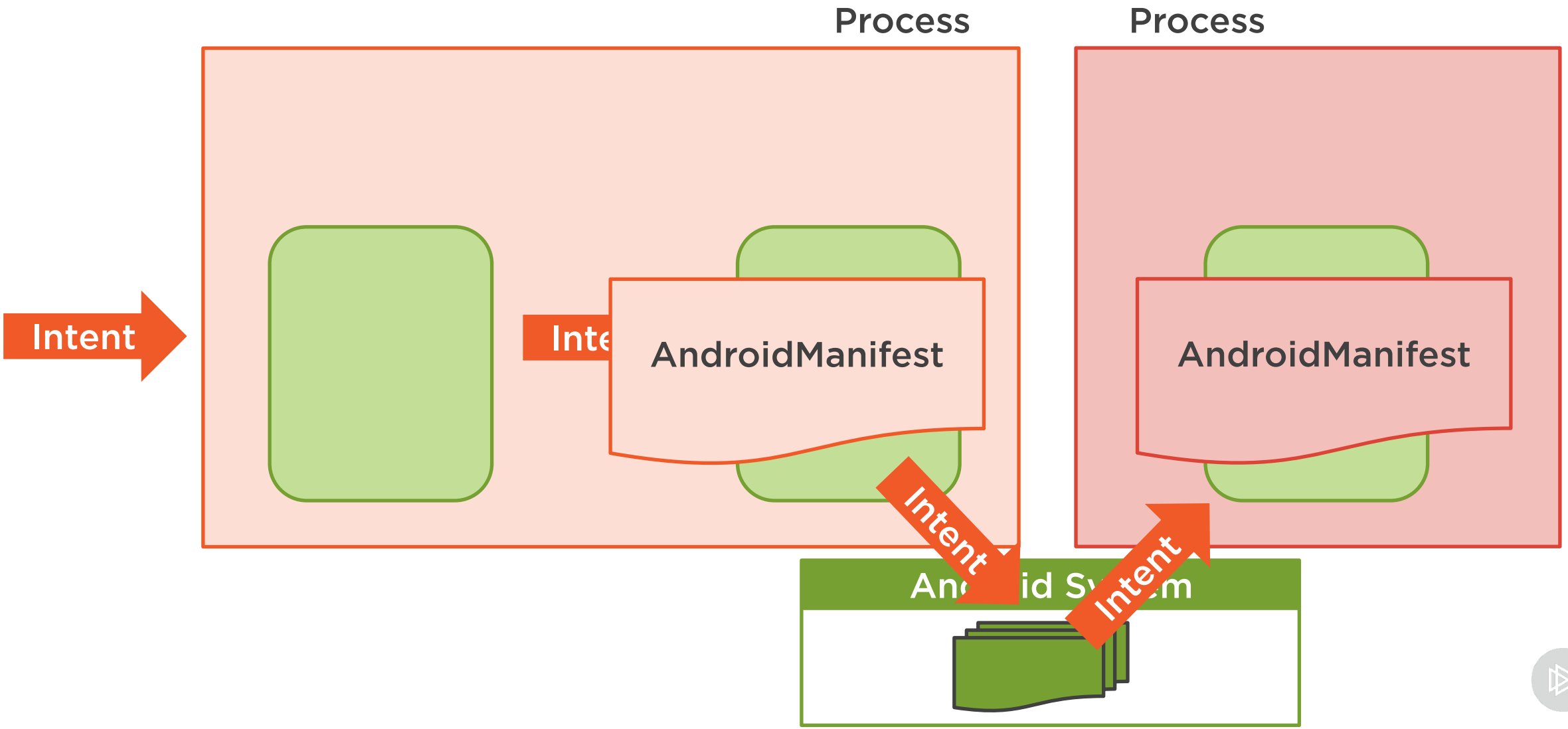
Setting data and mime type

- Use Intent.setDataAndType
- setType and setData cancel each other

Implicit Intents



Implicit Intents



Activities with Results

Some Activity classes return results

Camera Activity

- Presents camera functionality
- Returns image thumbnail

Contact Activity

- Presents contact UI
- Returns selected contact info

Many others

Activities with Results

Started differently than other activities

- Use startActivityForResult

Parameters passed to startActivityForResult

- Intent
- App defined integer identifier
  - Differentiates results within your app

Activities with Results

Receiving results

- Override your Activity's onActivityResult

Parameters received by onActivityResult

- App defined integer identifier
- Result code
  - RESULT_OK indicates success
- Intent
  - Contains activity results

Activity with Result Example

Camera

- Presents Camera UI
- Stores full quality image in a file
- Returns image thumbnail as a result

Starting the Activity

Intent action

- MediaStore.ACTION_IMAGE_CAPTURE

Extra

- MediaStore.EXTRA_OUTPUT
- File in which to save full quality image

Starting the Activity

```
public class MyActivity extends AppCompatActivity {
  private static final int SHOW_CAMERA = 1;
  private void showCamera(Uri photoFile) {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    intent.putExtra(MediaStore.EXTRA_OUTPUT, photoFile);
    startActivityForResult(intent, SHOW_CAMERA)
  }
  // other members elided for clarity
}
```

Receiving the Result

Check for request code of SHOW_CAMERA

- Identifies the result is for our request

Check for result code of RESULT_OK

- Indicates success
- Full quality image stored in file

Retrieve thumbnail

- Stored in result intent as a bitmap
- Name is “data”

Receiving the Result

```
public class MyActivity extends AppCompatActivity {
  private static final int SHOW_CAMERA = 1;
  @Override
  protected void onActivityResult(
        int requestCode, int resultCode, Intent result) {
    if(requestCode == SHOW_CAMERA && resultCode == RESULT_OK) {
      Bitmap thumbnail = result.getParcelable(“data”);
      // Do something
    }
  }
  // other members elided for clarity
}
```

Application Experience

Generally composed of multiple Activities

- Most probably come from your app
- Others may come from other apps
- Android needs to manage this flow
- Flow is managed as a task

What is a Task?

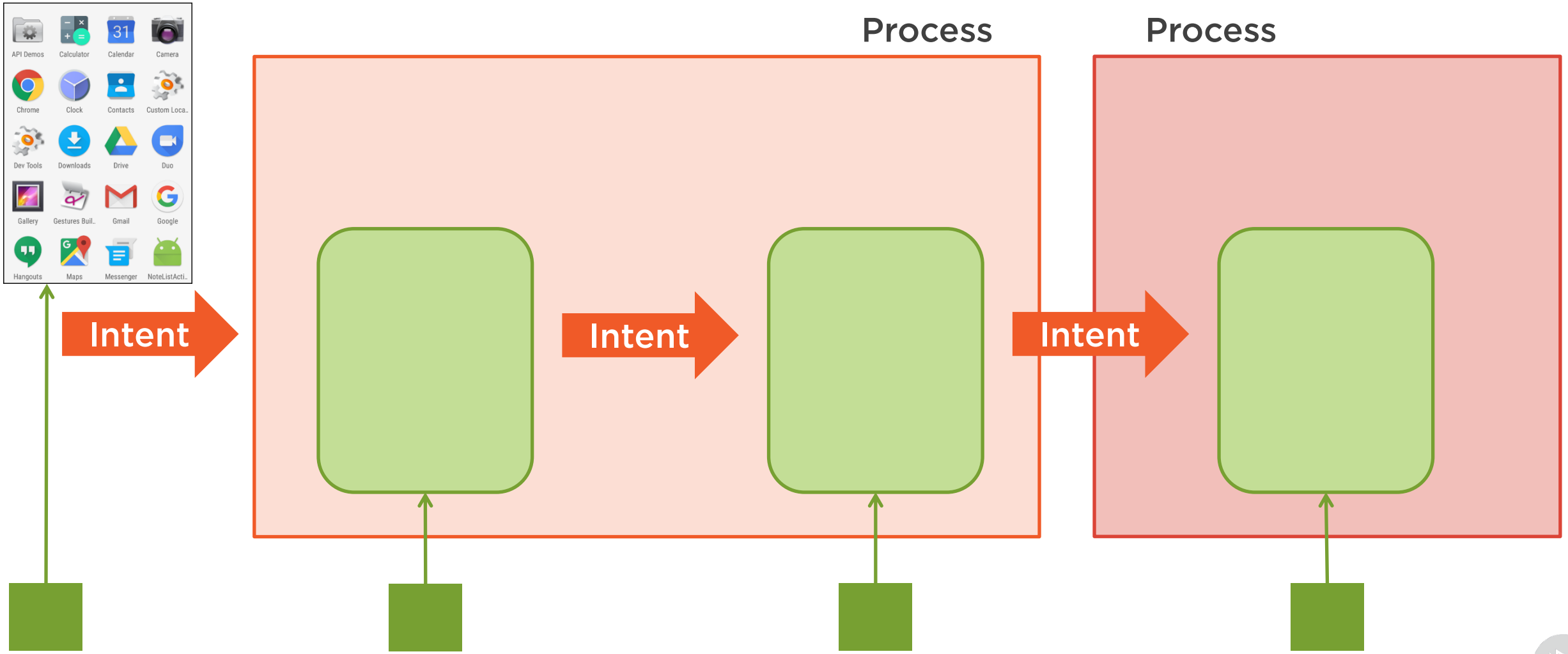
A task is a collection of activities that users interact with when performing a certain job.

Application Experience

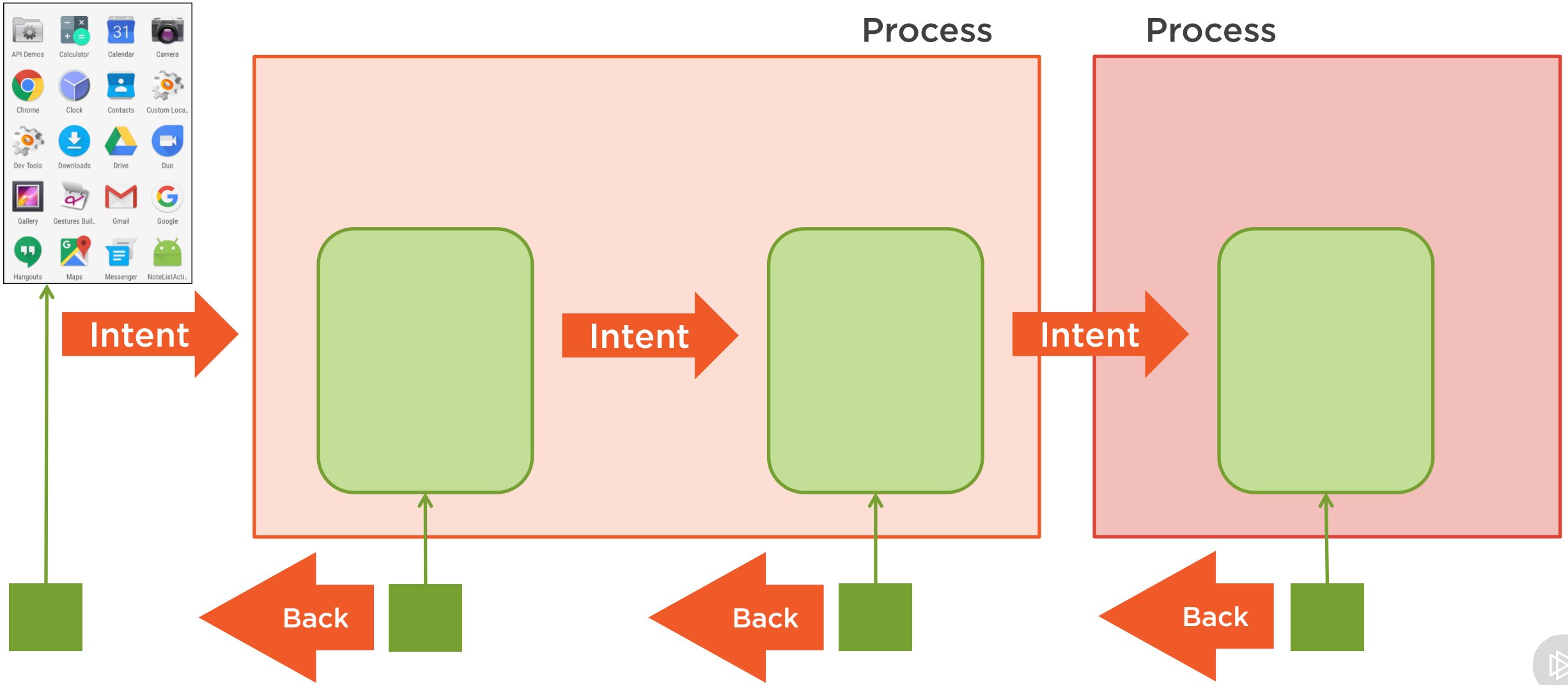
Task

- Managed as a stack
  - Known as the back stack
- Activities added going forward
- Back button removes Activities
  - Causes Activity to be destroyed

Task


API Demos
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Custom Loca..
Dev Tools
Downloads
Drive
Duo
Gallery
Gestures Buil..
Gmail
Google
Hangouts
Maps
Messenger
NoteListActi..
Intent
Process
Intent
Intent
Process

Task



Application Experience

Managing persistent state

- Use edit-in-place model
- Changes saved with no special action

Saving changes

- Write to backing store when leaving
- Handle in onPause

Handling new entries

- New entries created right away
- Handle in onCreate

Activity Lifecycle

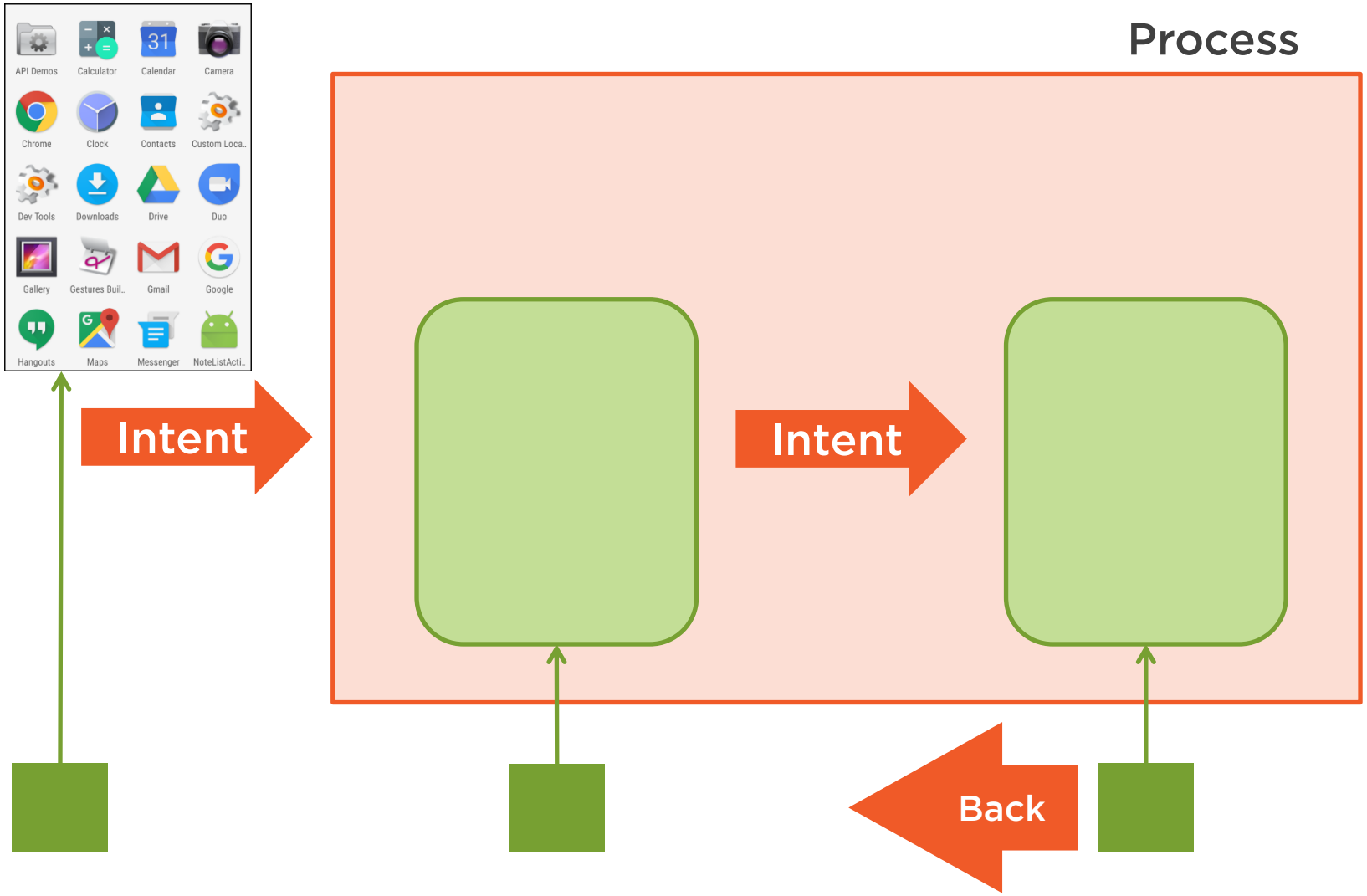
Common causes of Activity destruction

- Leaving with the back button
- Calling finish method
- System initiated

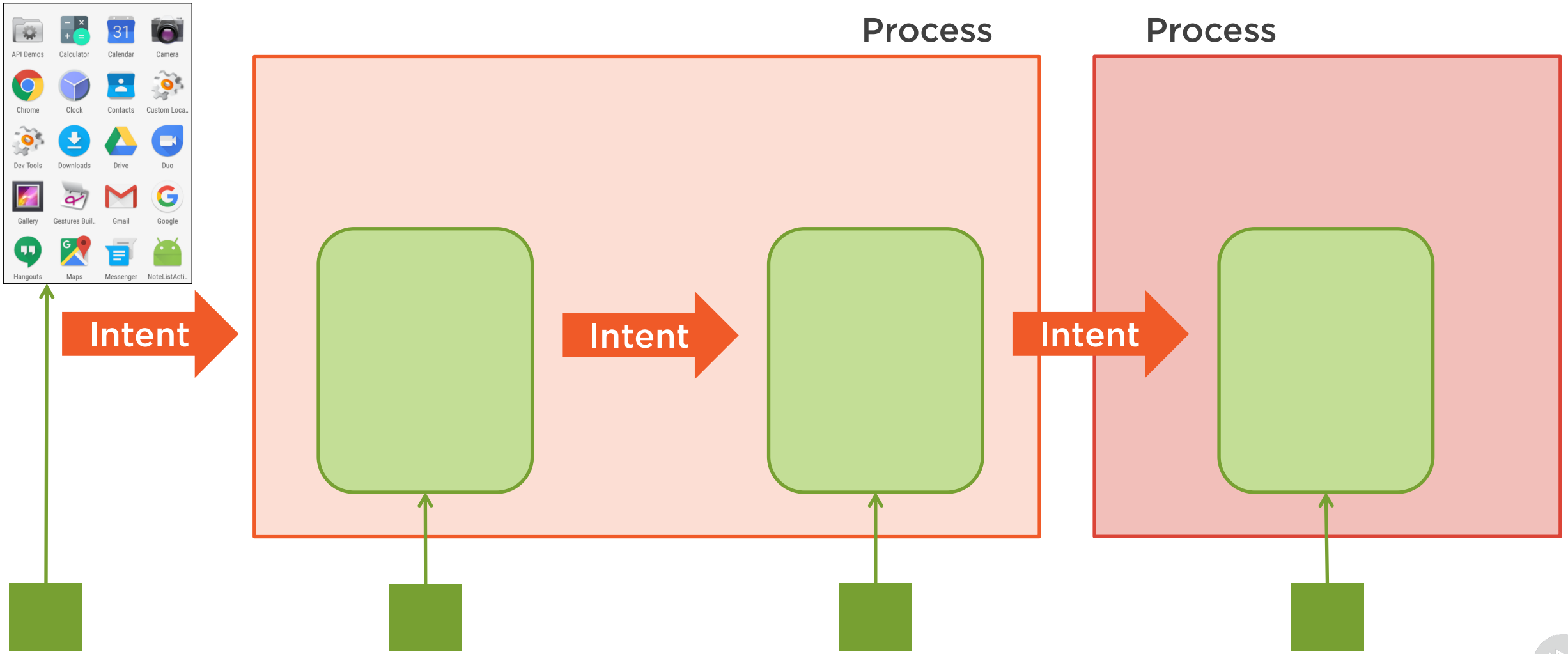
System initiated destruction

- Generally to reclaim resources
- Prolonged period in the background
- System experiencing resource pressure

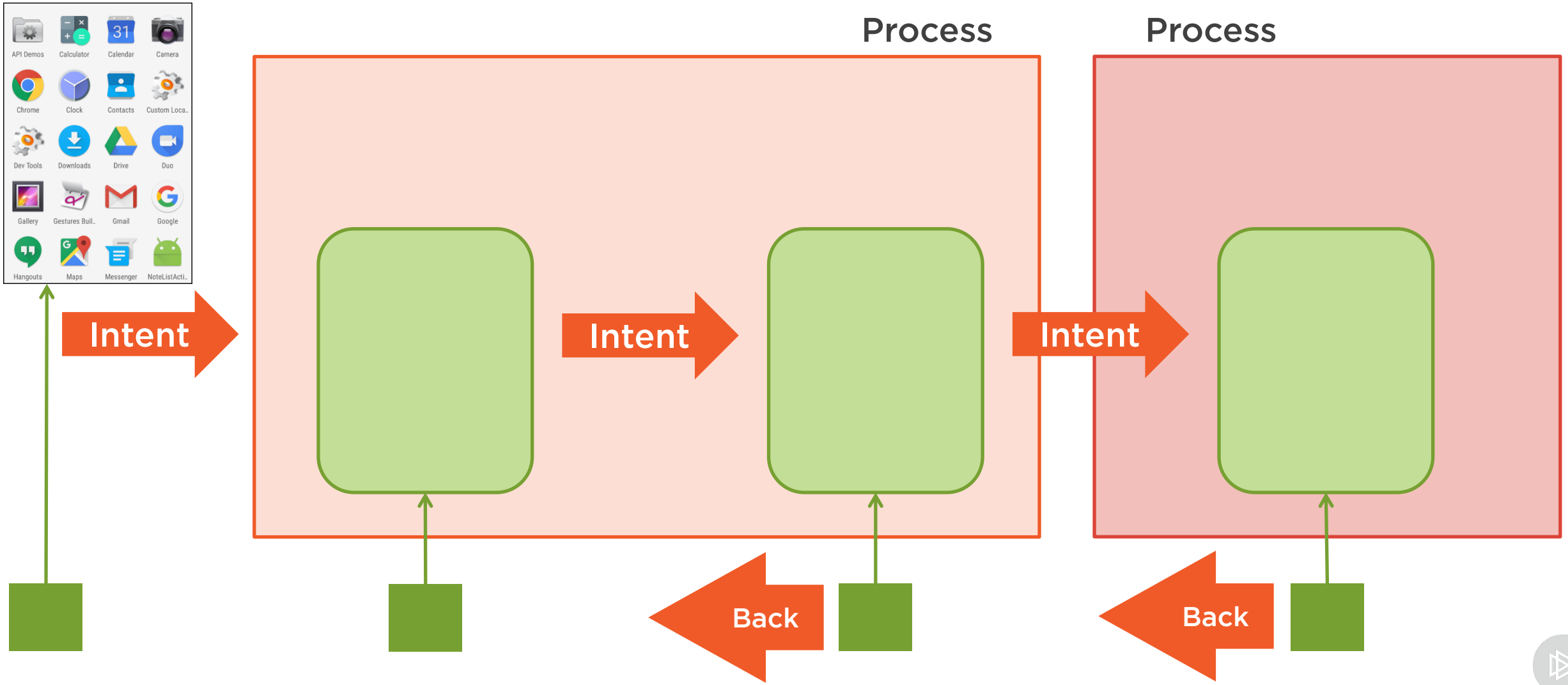
Task



Task


API Demos
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Custom Loca..
Dev Tools
Downloads
Drive
Duo
Gallery
Gestures Buil..
Gmail
Google
Hangouts
Maps
Messenger
NoteListActi..
Intent
Process
Intent
Intent
Process

Task



Activity Lifecycle Methods

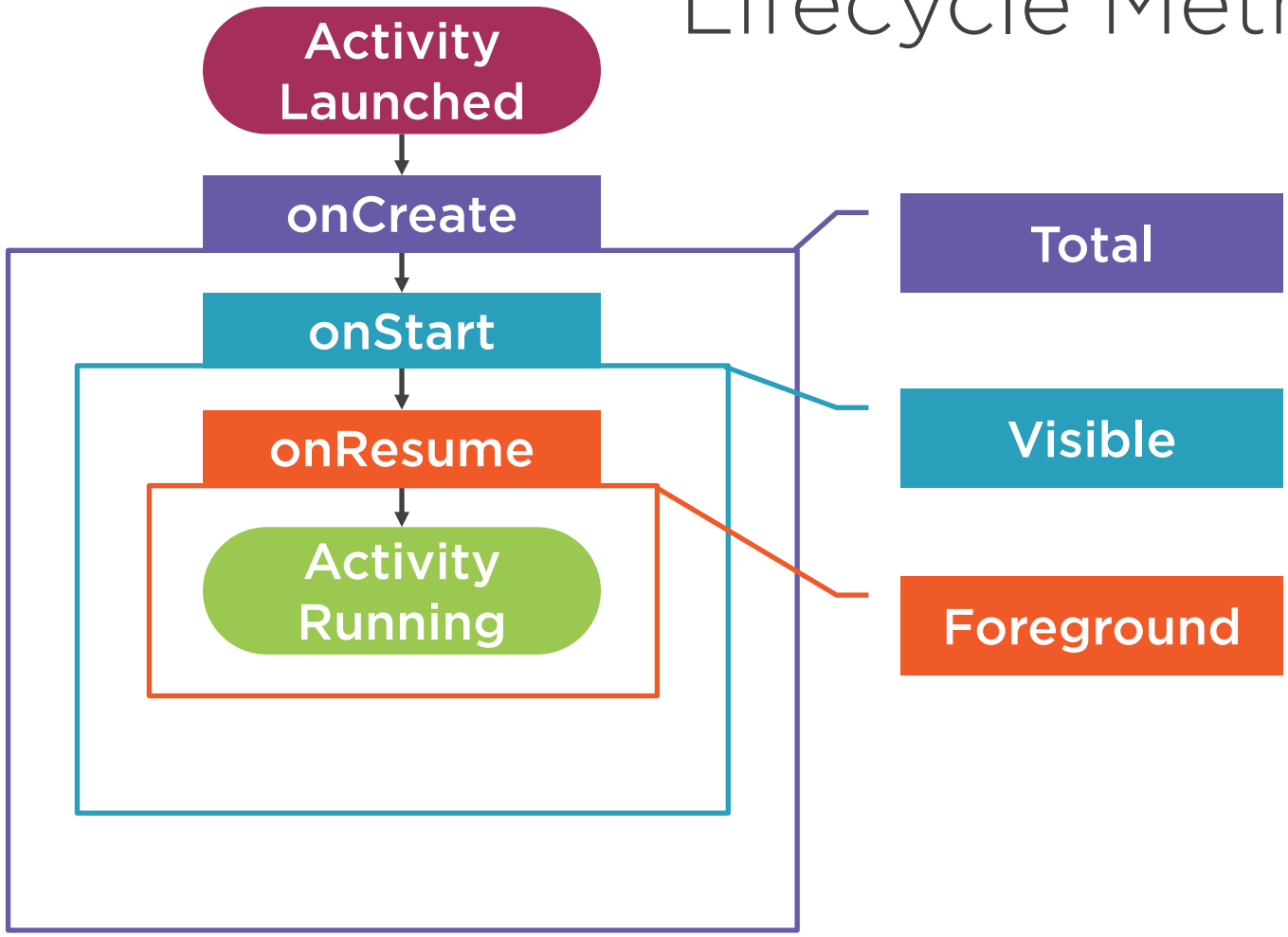
Lifetimes within Activity lifecycle

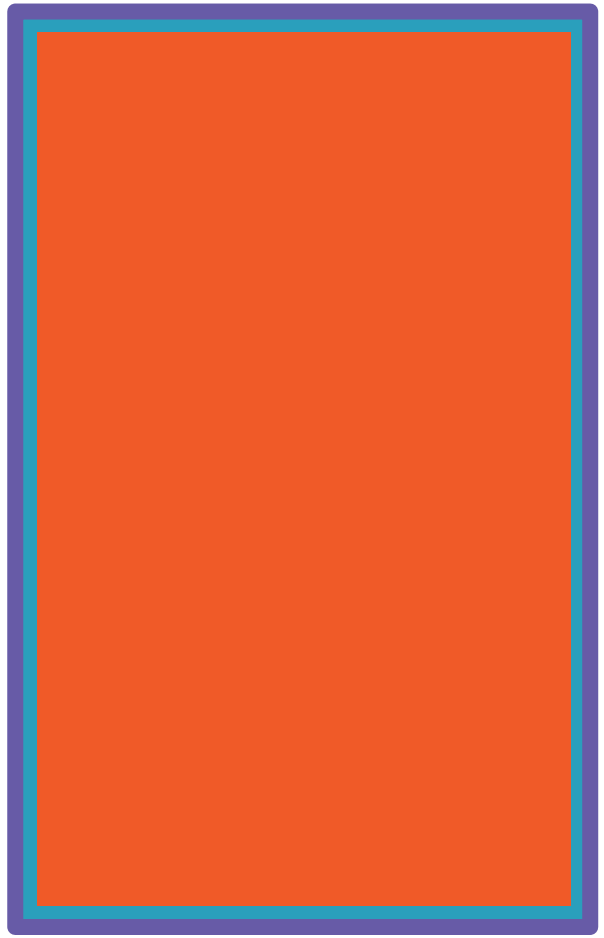
- Total lifetime
- Visible lifetime
- Foreground lifetime

Activity lifecycle methods

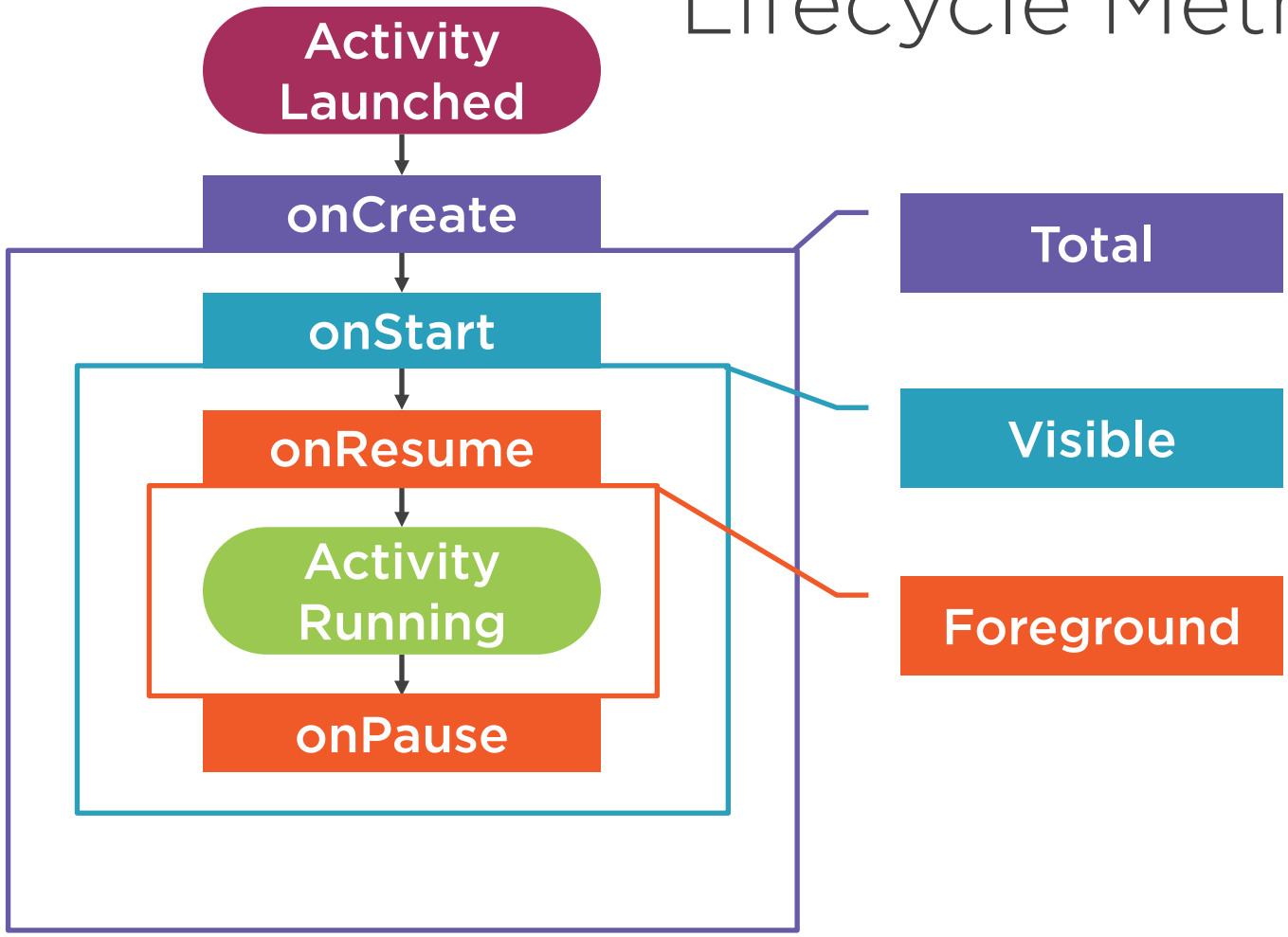
- Methods for start/end of each lifetime
- A few additional methods for transitions

Lifecycle Methods

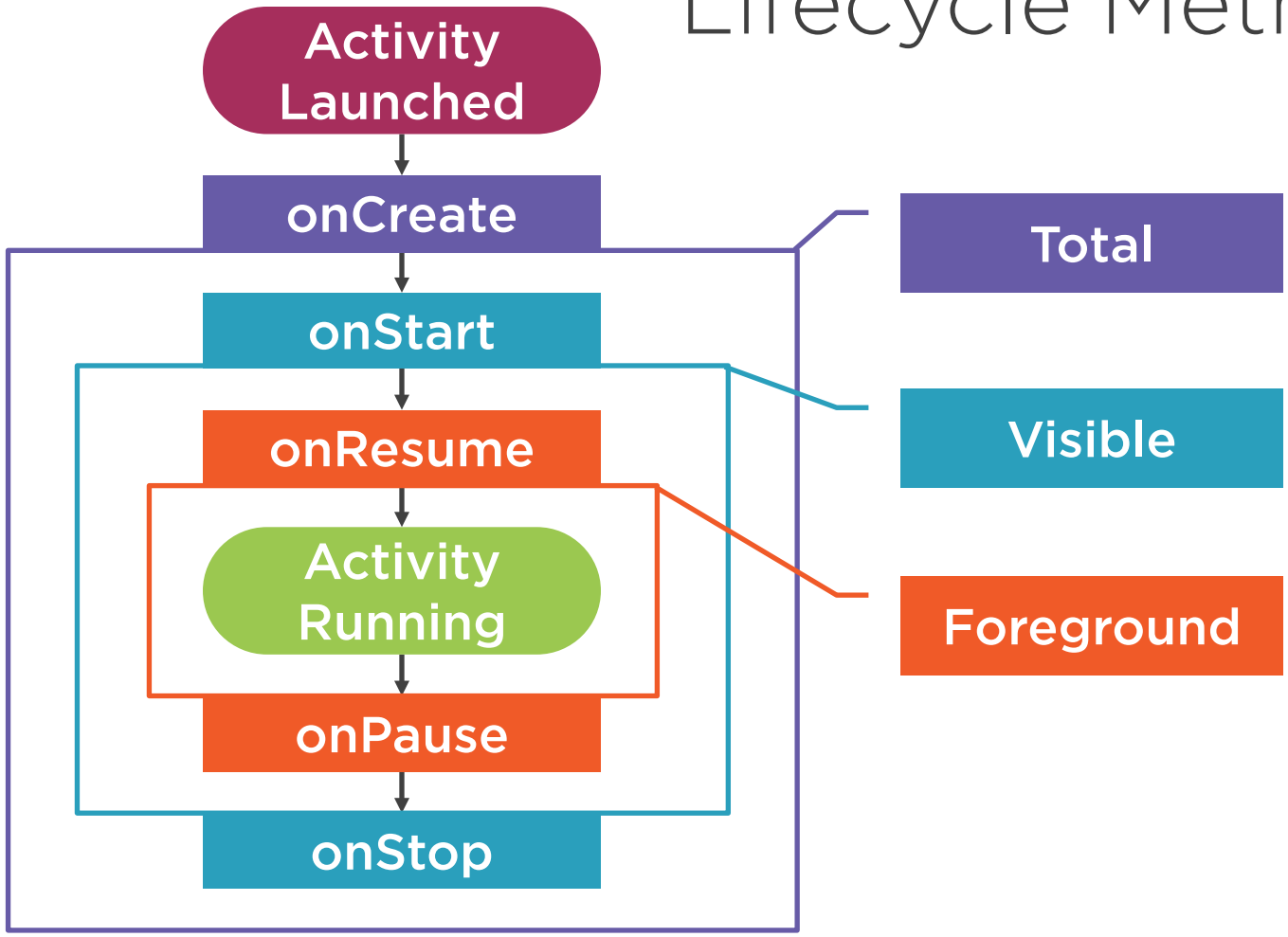




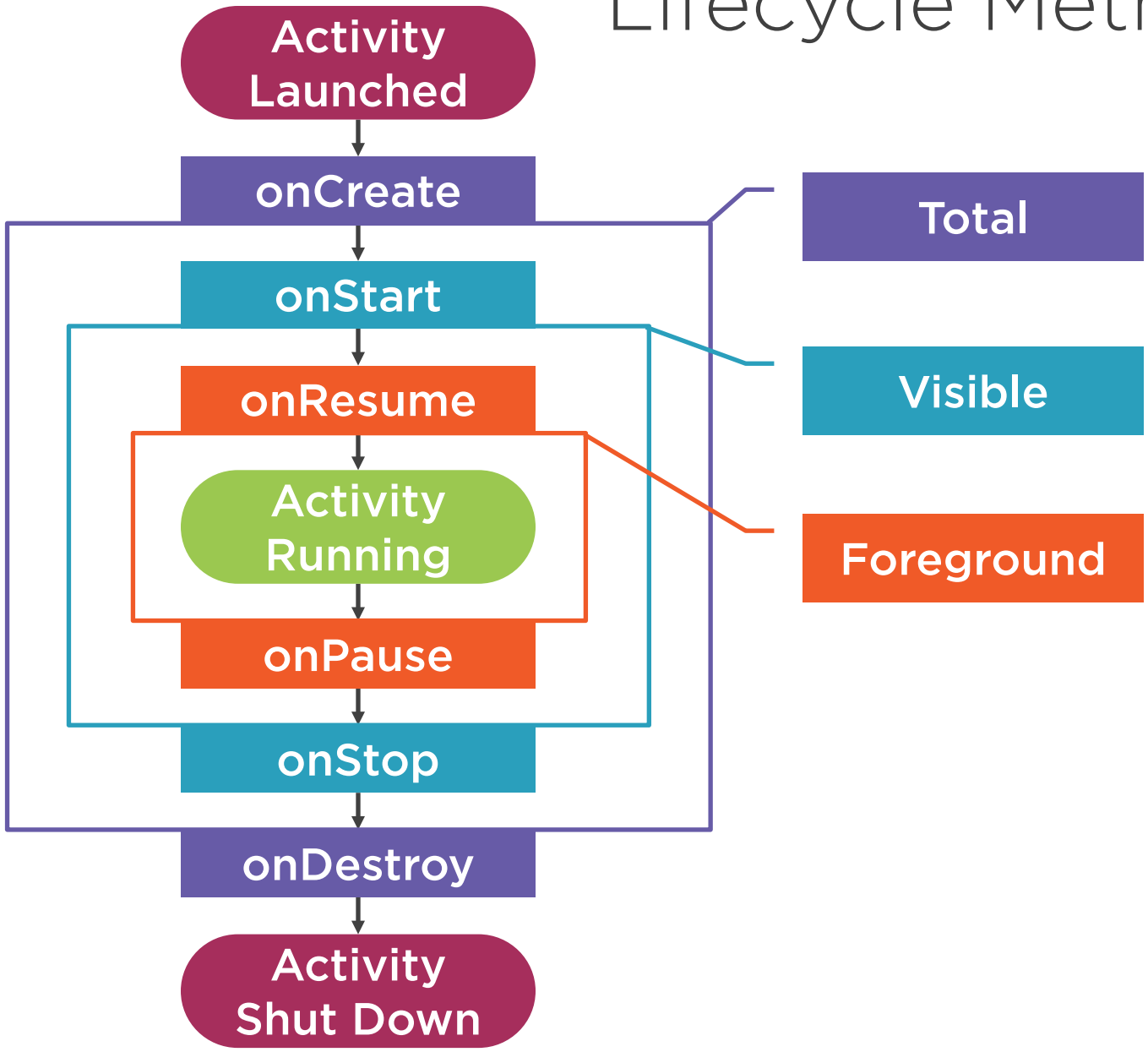
Lifecycle Methods

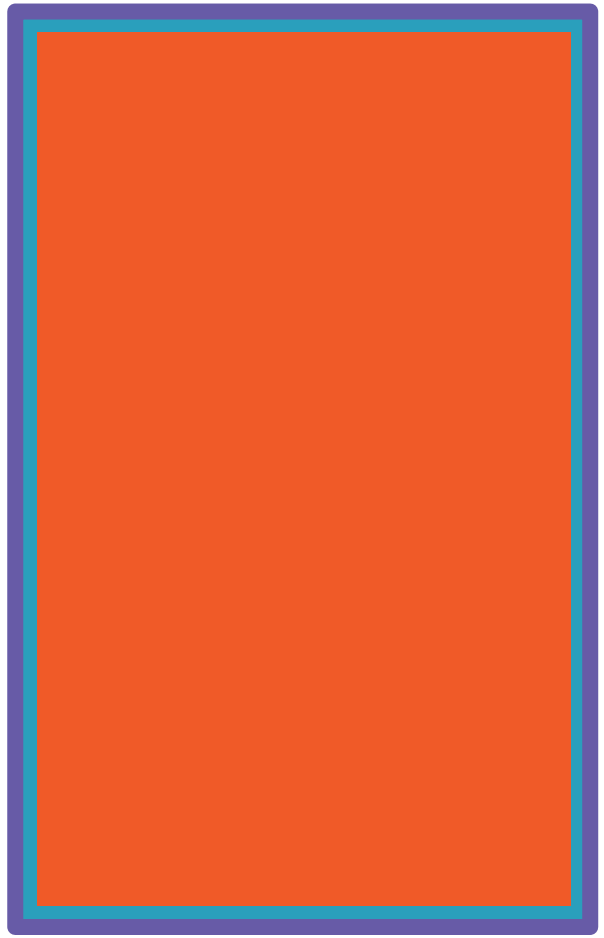


Lifecycle Methods

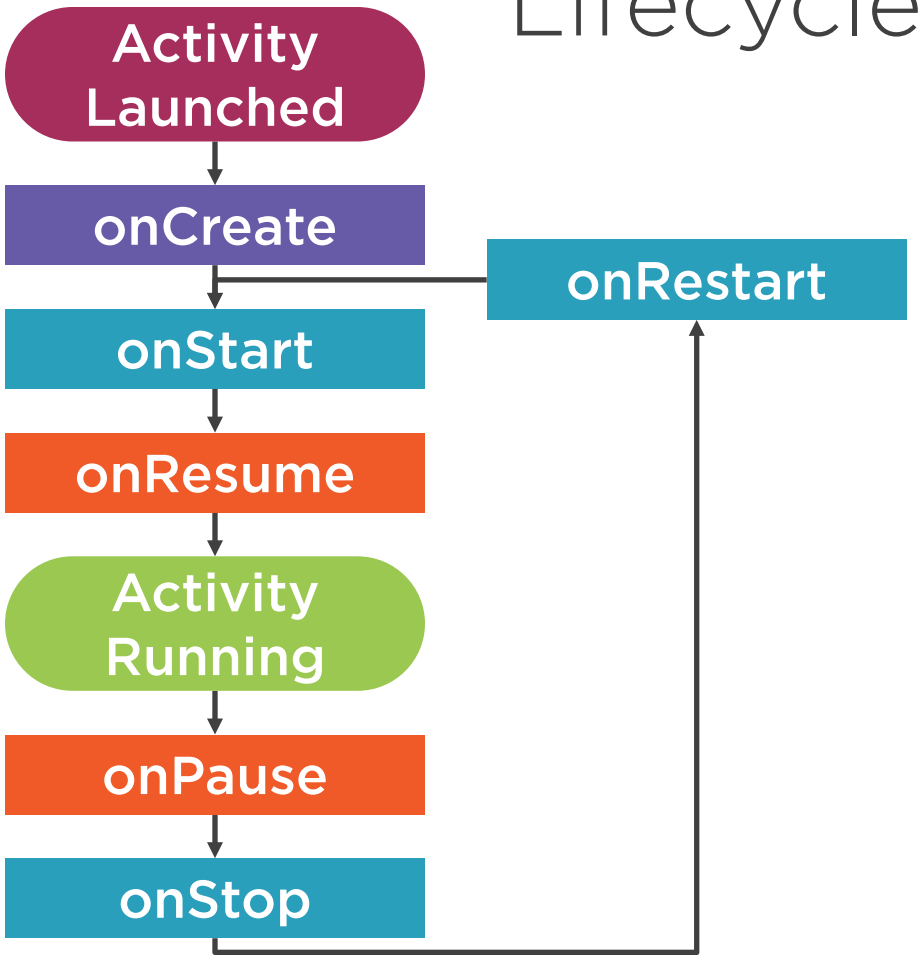


Lifecycle Methods

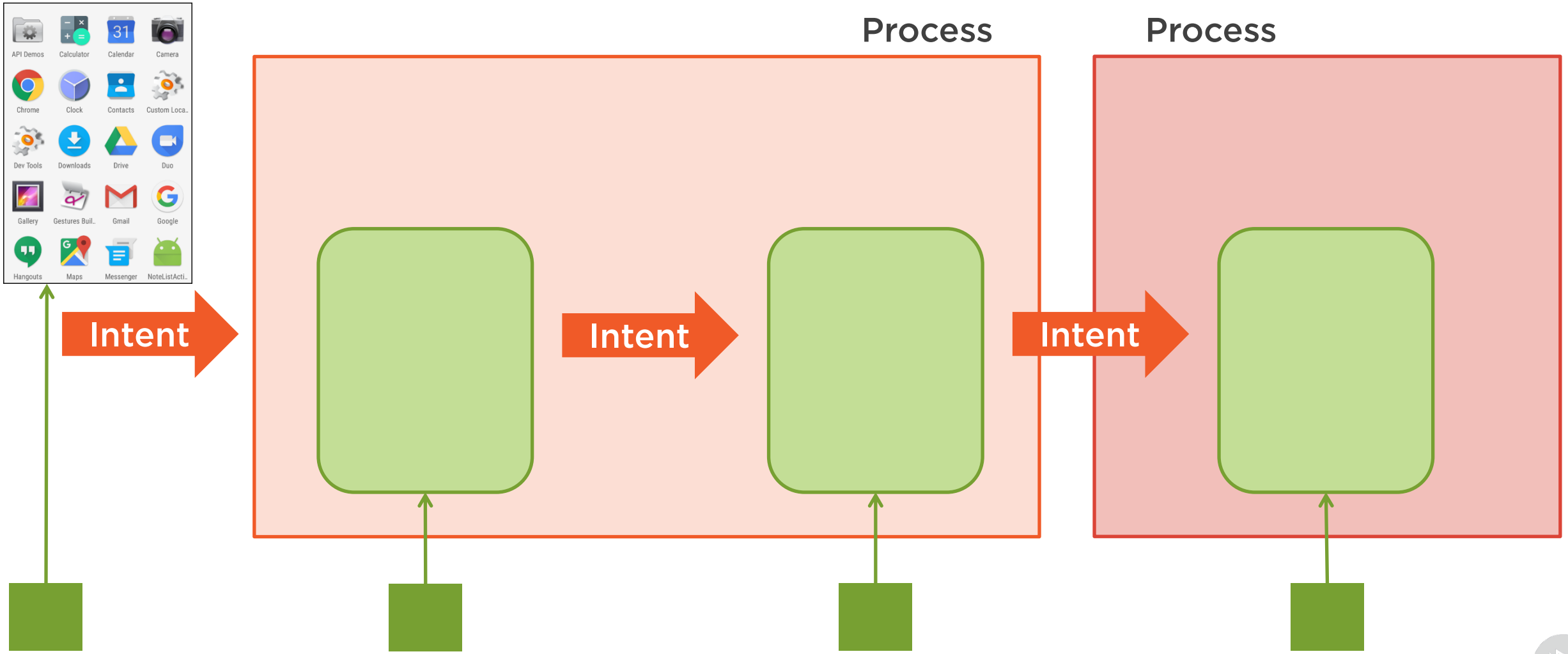




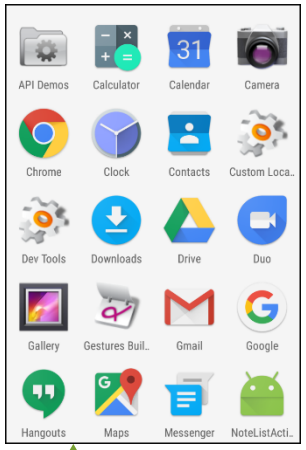
Lifecycle Methods

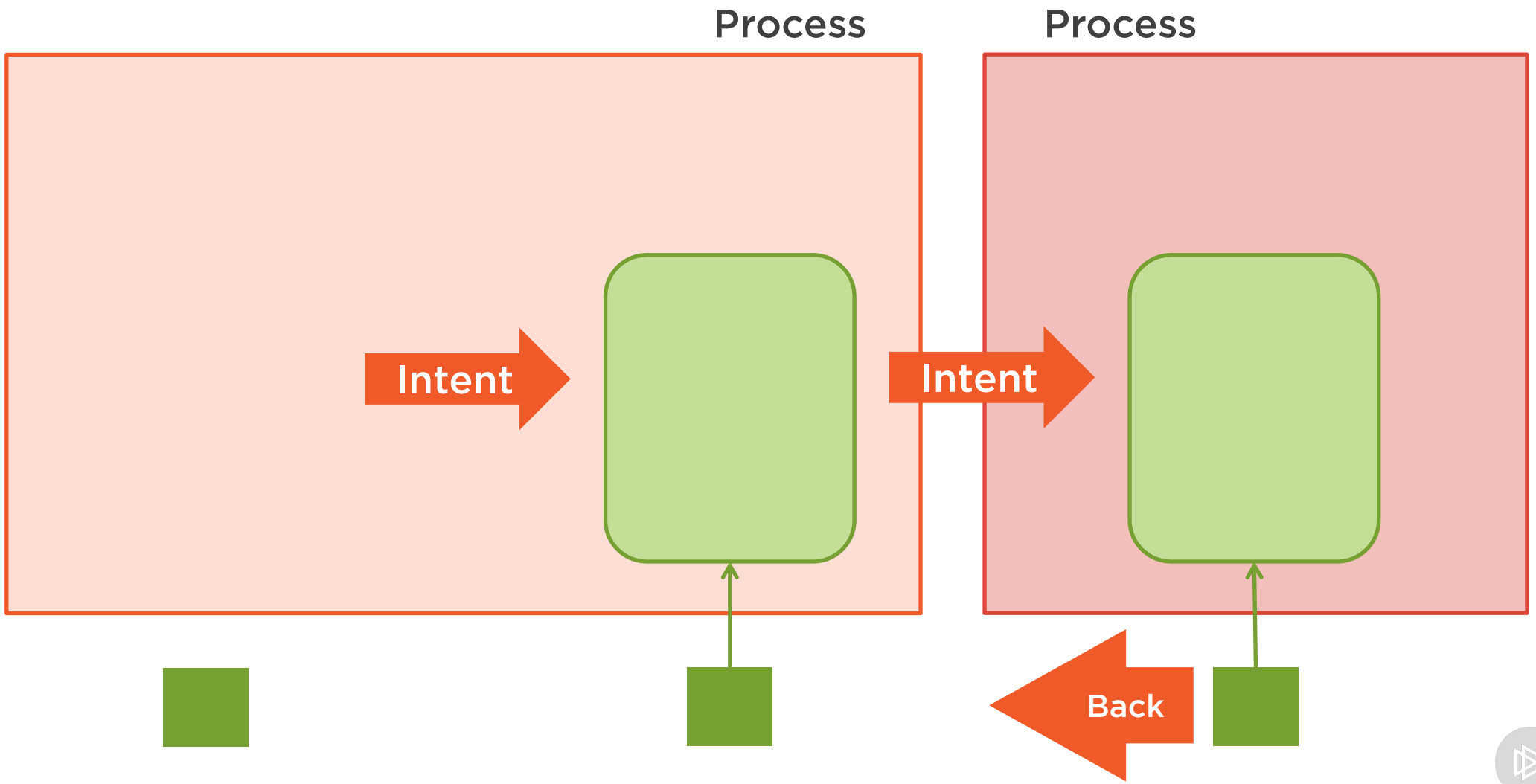


Activity State Management

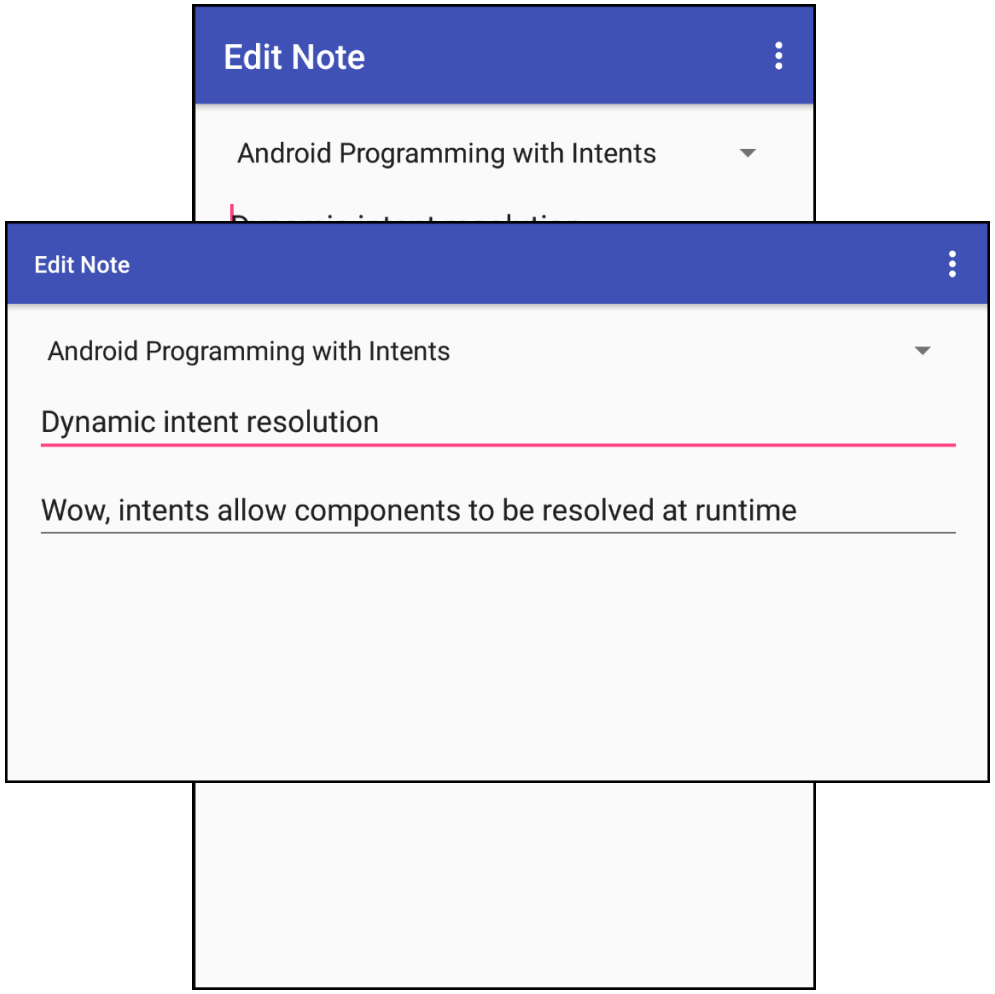


Activity State Management





Activity State Management



Activity State Management

Activities provide state management

- Opportunity to save before destroy
- Saved state provided on restore

Saving state

- onSaveInstanceState
- Write Activity state to passed Bundle

Restoring state

- onCreate
- Receives saved Bundle on restore
- Bundle is null on initial create
- Intent remains available on restore

Summary

App components run in the same process

- When simultaneously active

Explicit intents

- Target type explicitly provided

Implicit intents

- Target identified using characteristics
- System finds best match
- Match may vary depending on apps installed on user device

Summary

Some Activity types return results

- Use startActivityForResult
- Results received in onActivityResult

Activity task

- Collection of activities to perform a job
- Organized as the back stack

Managing persistent state

- Use edit-in-place model
- Write to backing store when leaving
- New entries created right away

Summary

Activity lifecycle

- Total lifetime
- Visible lifetime
- Foreground lifetime
- Methods for start/end of each lifetime

Activity state management

- Activities often destroyed & restored
- Save state in onSaveInstanceState
- Saved state passed to onCreate