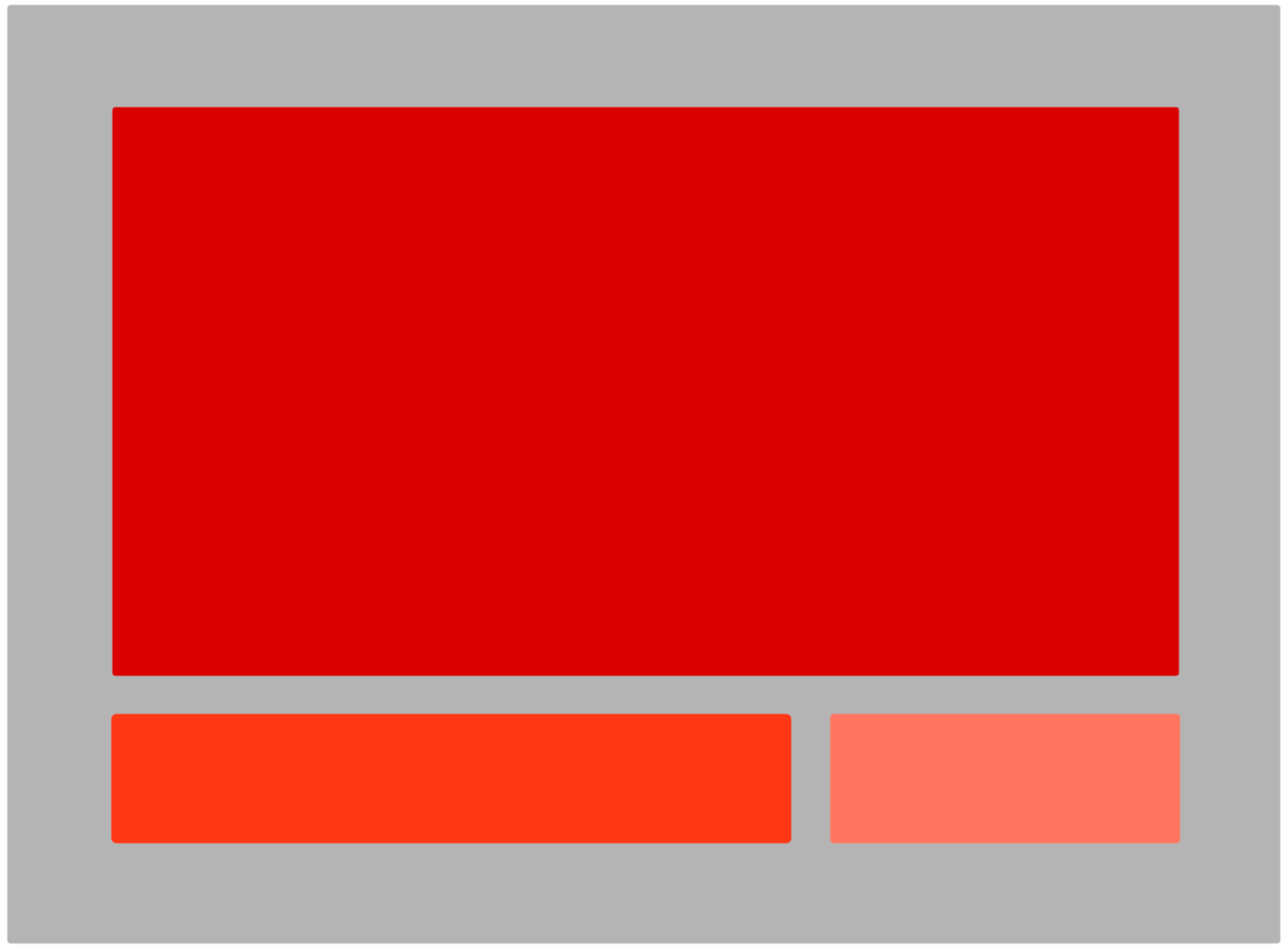


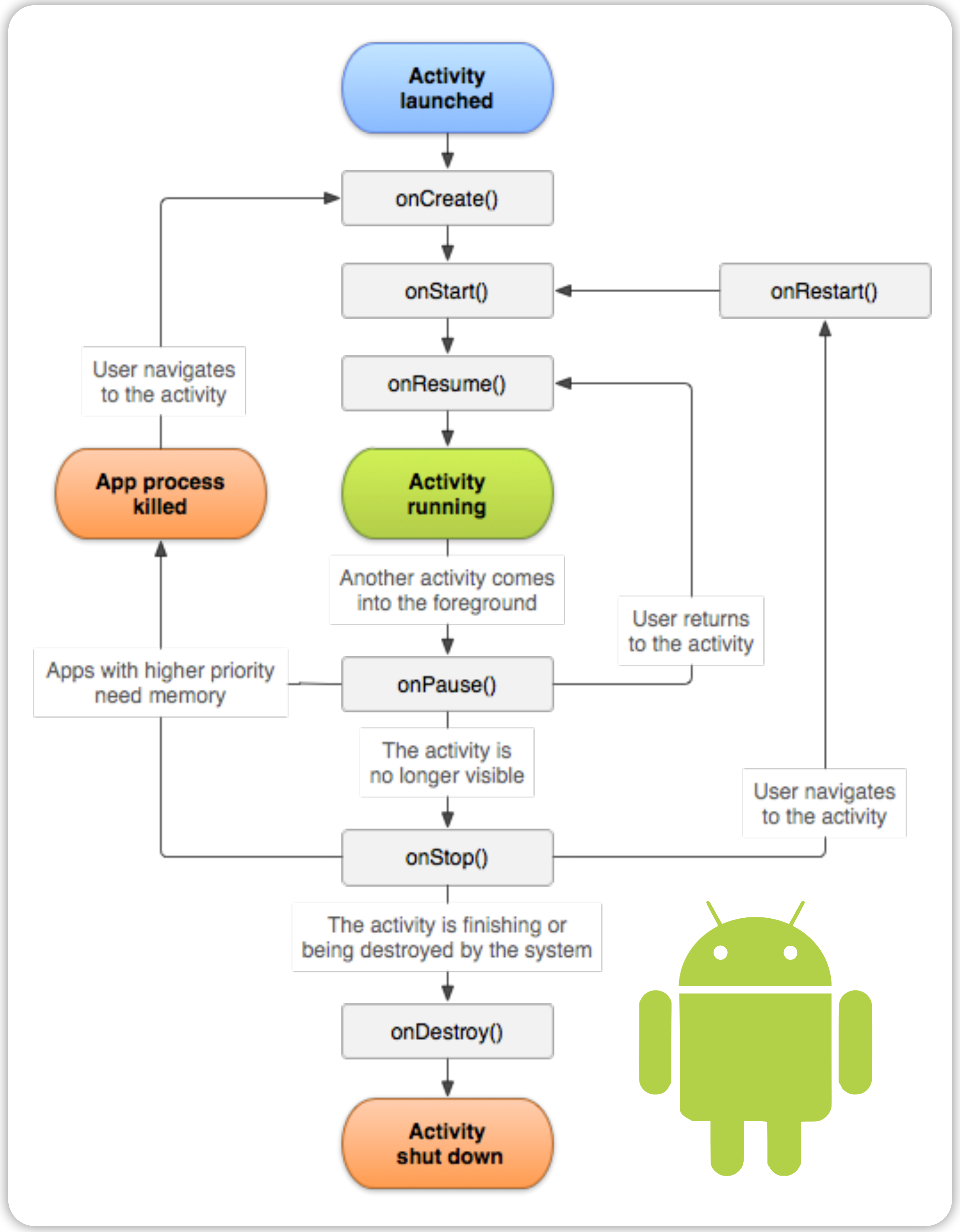
MOBILE APPLICATION DEVELOPMENT

ANDROID (2017)

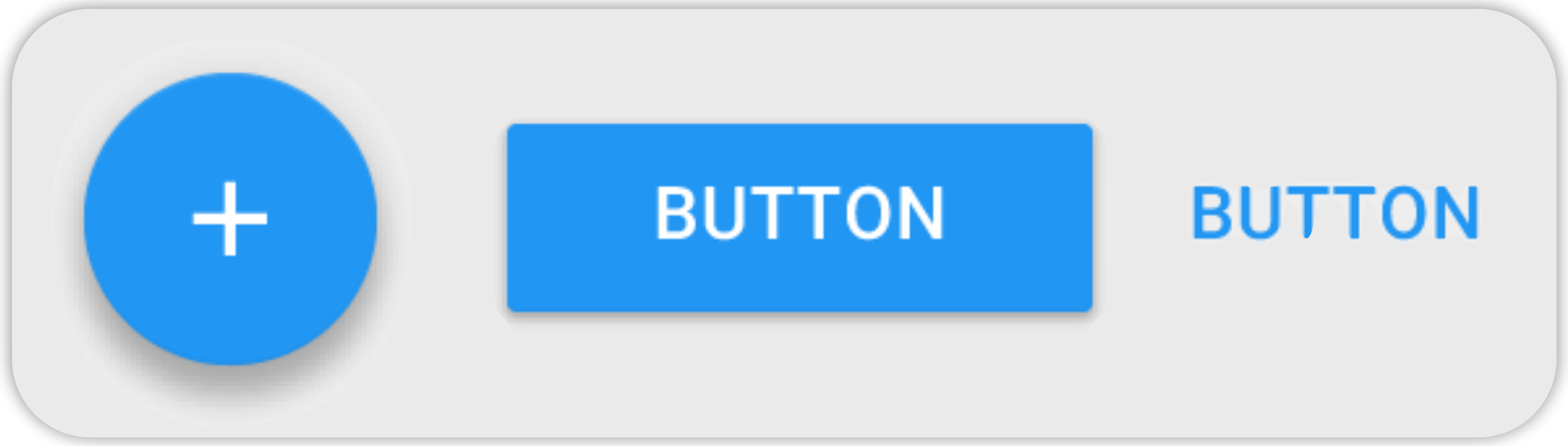
LECTURE 04: UI AND LAYOUT

ACTIVITIES

- Activities are the primary organizational unit in apps.
- An Activity is a self-contained task taking up one screen.
- Activities can start other Activities.
- Apps can use Activities from other applications.
- Activities control a single View which displays content.

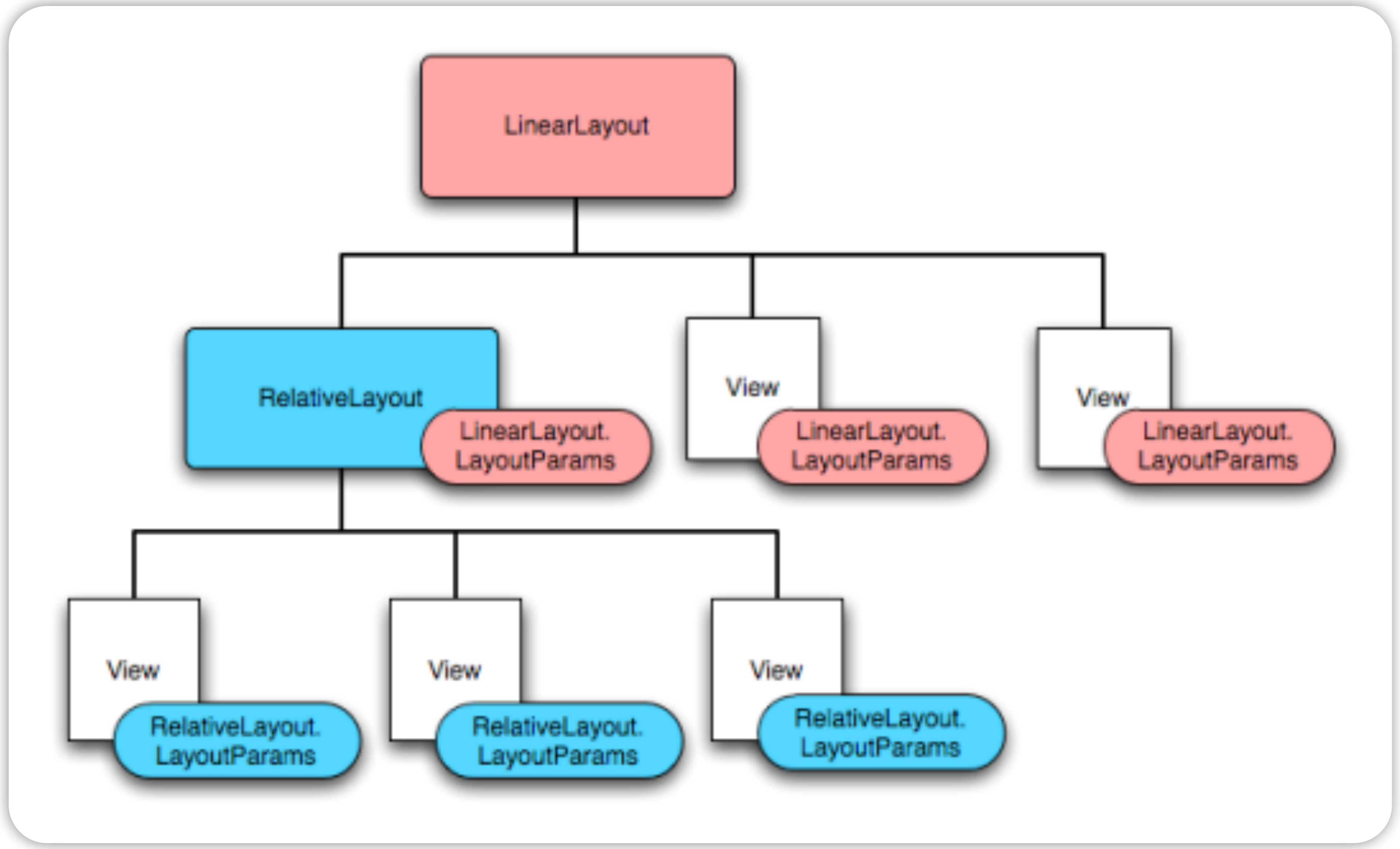

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

VIEWS AND VIEWGROUPS



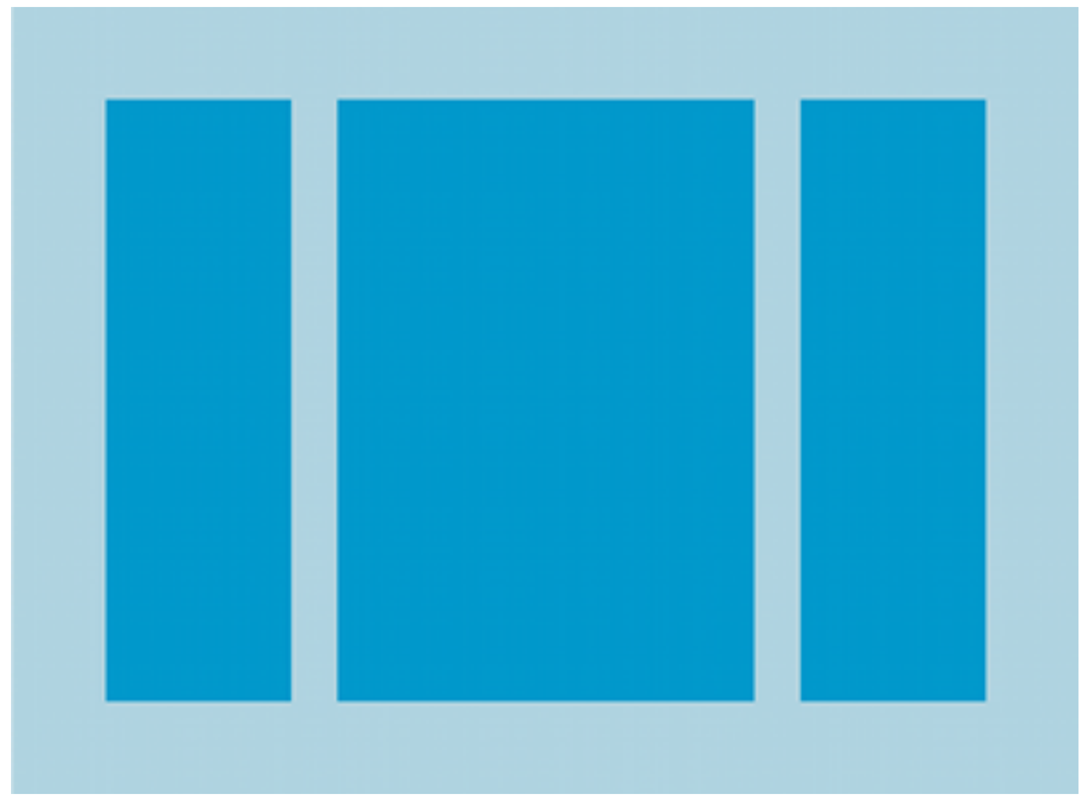
- A View presents information to the user visually.
- Each View occupies some portion of the user interface.
  - View dimensions and locations are specified in pixels.
  - Despite having coordinates and dimensions, the programmer generally does not set these properties directly.
- Views are organized within ViewGroups, which contain other Views (or ViewGroups) and define the layout for Views they contain.

ANDROID VIEW LAYOUT


LinearLayout
RelativeLayout
LinearLayout.LayoutParams
View
LinearLayout.LayoutParams
View
LinearLayout.LayoutParams
View
RelativeLayout.LayoutParams
View
RelativeLayout.LayoutParams
View
RelativeLayout.LayoutParams

- Android handles UI design with Layouts.
- Layouts are ViewGroups with defined rules.
- Layouts control how Views are positioned.
- Layouts manage Views and ViewGroups that they contain, and are managed by any Layouts or ViewGroups which contain them (think of a tree).
  - Most Layouts do not use absolute sizing for their Views.
  - Layouts allow many screen sizes to be supported in one flexible unit of code.

LAYOUTS: LINEAR LAYOUT

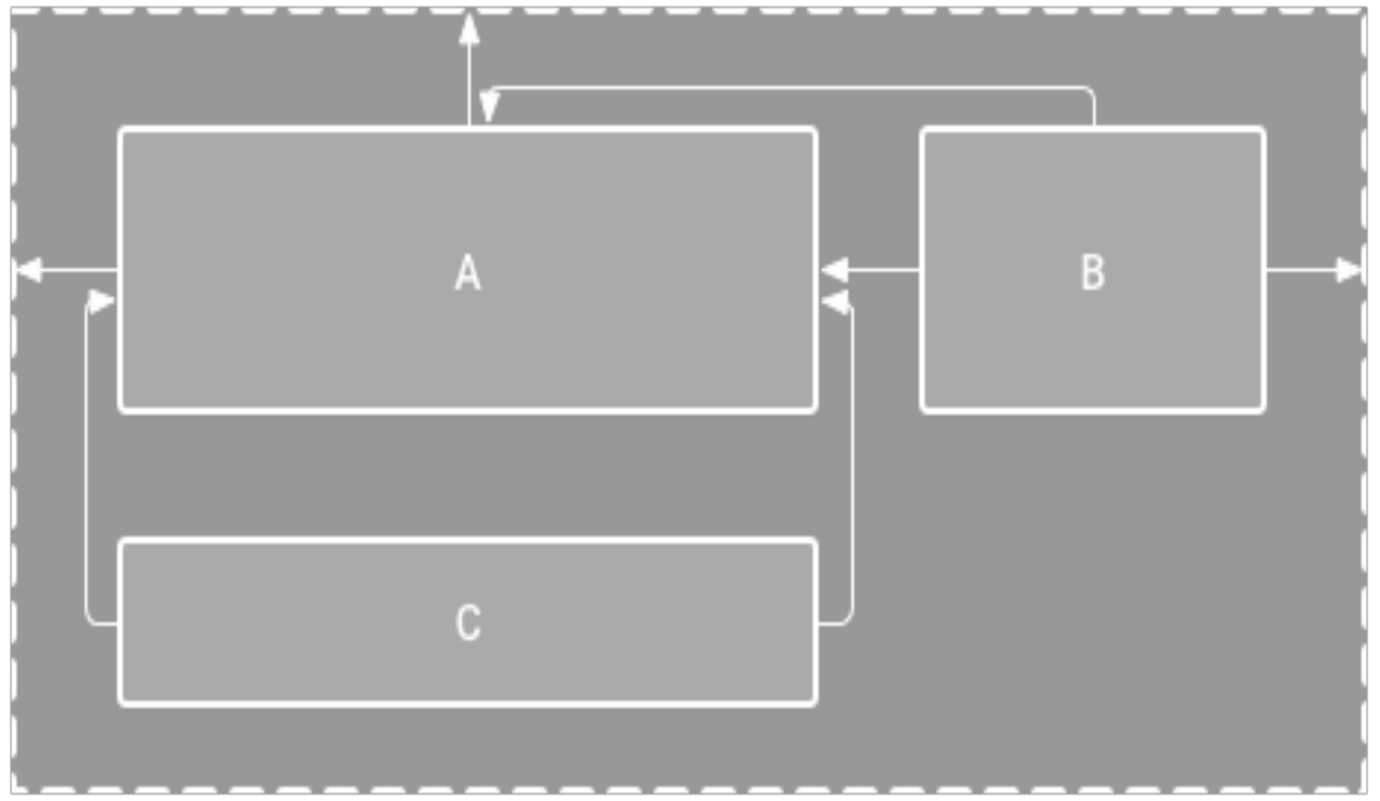


- A LinearLayout organizes child Views in a single direction.
  - Can be either horizontal or vertical.
  - Usually attempts to fill its available space, adjusting child Views to fit.
- Allows the programmer to define spacing (dividers) between child Views.
- Uses the concept of 'weight' to determine how much space child Views are given.
- Uses the concept of 'gravity' to describe where child Views should be 'pushed' to within their containing layout.

LAYOUTS: RELATIVE LAYOUT

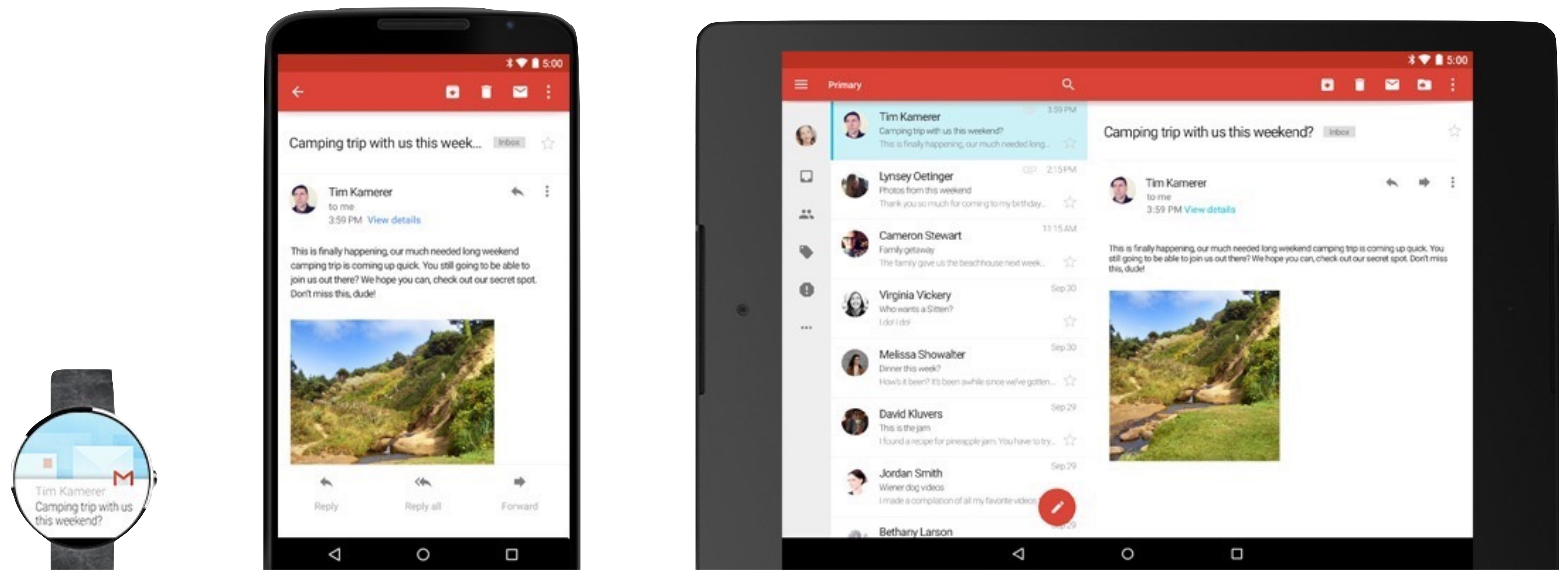
- A RelativeLayout organizes child Views relative to each other.
  - Requires child Views to have assigned IDs.
  - Uses child View IDs to specify where child Views are in relation to others.
- Allows the programmer to define rules for child Views such as 'child View 1 is centered in its parent View and is located to the left of child View 2'.
- Uses the concept of 'gravity' to describe where child Views should be 'pushed' to within their containing layout.

LAYOUTS: CONSTRAINT LAYOUT



- A ConstraintLayout organizes child Views with constraints.
  - Similar to RelativeLayout, but more precise.
  - Uses child View IDs to specify where child Views are in relation to others.
- Allows the programmer to define precise rules for child Views such as 'child View 1 is 10 pixels to the left of child View 2, and 5 pixels above child View 3, which is attached to the bottom left corner of the layout'.
- Uses the concept of 'weight' to determine how much space child Views are given.
- Uses the concept of 'gravity' to describe where child Views should be 'pushed' to within their containing layout.

INFORMATIONAL VIEWS


Tim Kamerer
Camping trip with us this weekend?

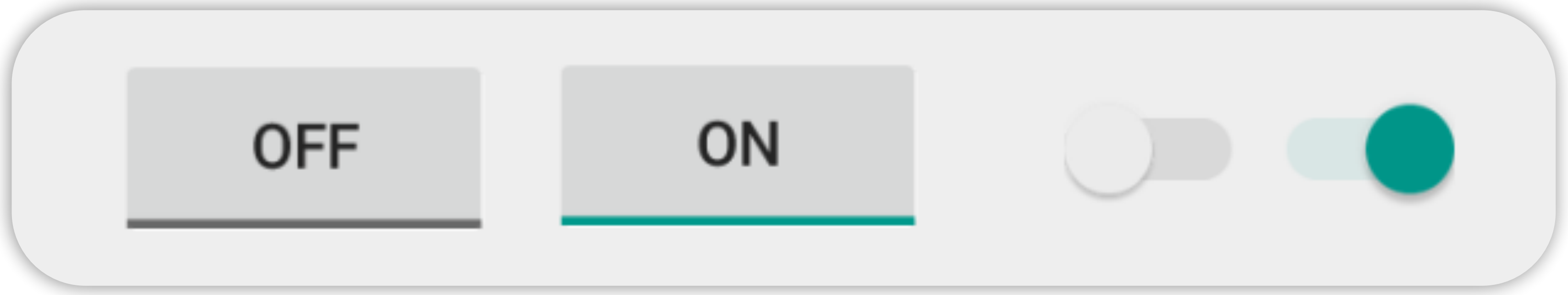
Camping trip with us this week... Inbox
Tim Kamerer
to me
3:59 PM View details
This is finally happening, our much needed long weekend camping trip is coming up quick. You still going to be able to join us out there? We hope you can, check out our secret spot. Don't miss this, dude!

Tim Kamerer 3:59 PM
Camping trip with us this weekend?
This is finally happening, our much needed long...
Lynsey Oetinger 2:15 PM
Photos from this weekend
Thank you so much for coming to my birthday...
Cameron Stewart 11:15 AM
Family getaway
The family gave us the beachhouse next week...
Virginia Vickery Sep 30
Who wants a Sitten?
I do! I do!
Melissa Showalter Sep 30
Dinner this week?
How's it been? It's been awhile since we've gotten...
David Kluvers Sep 29
This is the jam
I found a recipe for pineapple jam. You have to try...
Jordan Smith Sep 29
Wiener dog videos
I made a compilation of all my favorite videos
Bethany Larson

Camping trip with us this weekend? Inbox
Tim Kamerer
to me
3:59 PM View details
This is finally happening, our much needed long weekend camping trip is coming up quick. You still going to be able to join us out there? We hope you can, check out our secret spot. Don't miss this, dude!

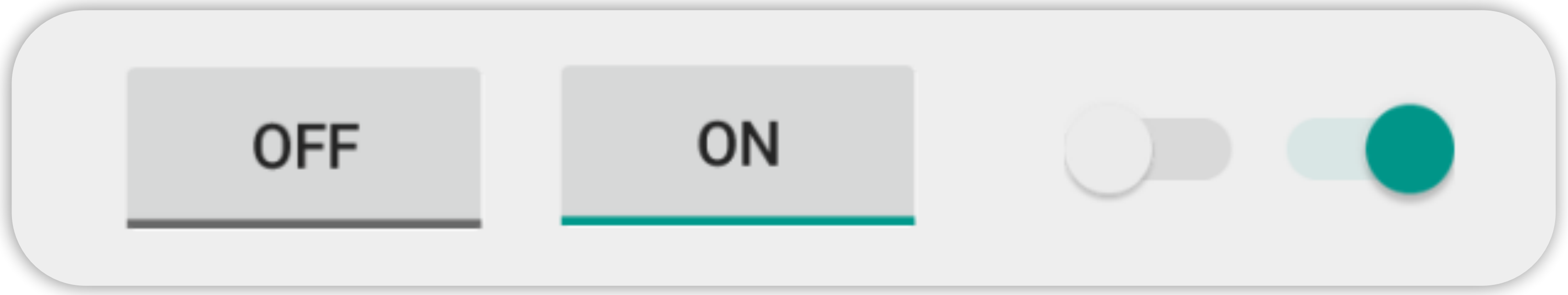
- Many Views are primarily for displaying data.
- Examples include:
  - TextView
  - ImageView
  - VideoView
  - ProgressBar
  - AnalogClock / DigitalClock

INPUT CONTROLS



- Controls are Views which allow the user to interact with the application directly.
- Examples include:
  - Button
  - CheckBox
  - ToggleButton
  - RadioButton
  - EditText

INTERACTING WITH CONTROLS



- Controls generally manage a value and notify the program when it changes.
  - Buttons detect when they are pressed.
  - EditTexts detect when their text changes.
  - CheckBoxes detect when their checked state changes.
- The program responds to control events by associating listeners with controls.
  - Each control defines interfaces for the delegates which can handle its events.
  - Generally, objects implementing interfaces are used as control handlers.