Mobile Application Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Dr. Siobhán Drohan (sdrohan@wit.ie)

Department of Computing, Maths & Physics

Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

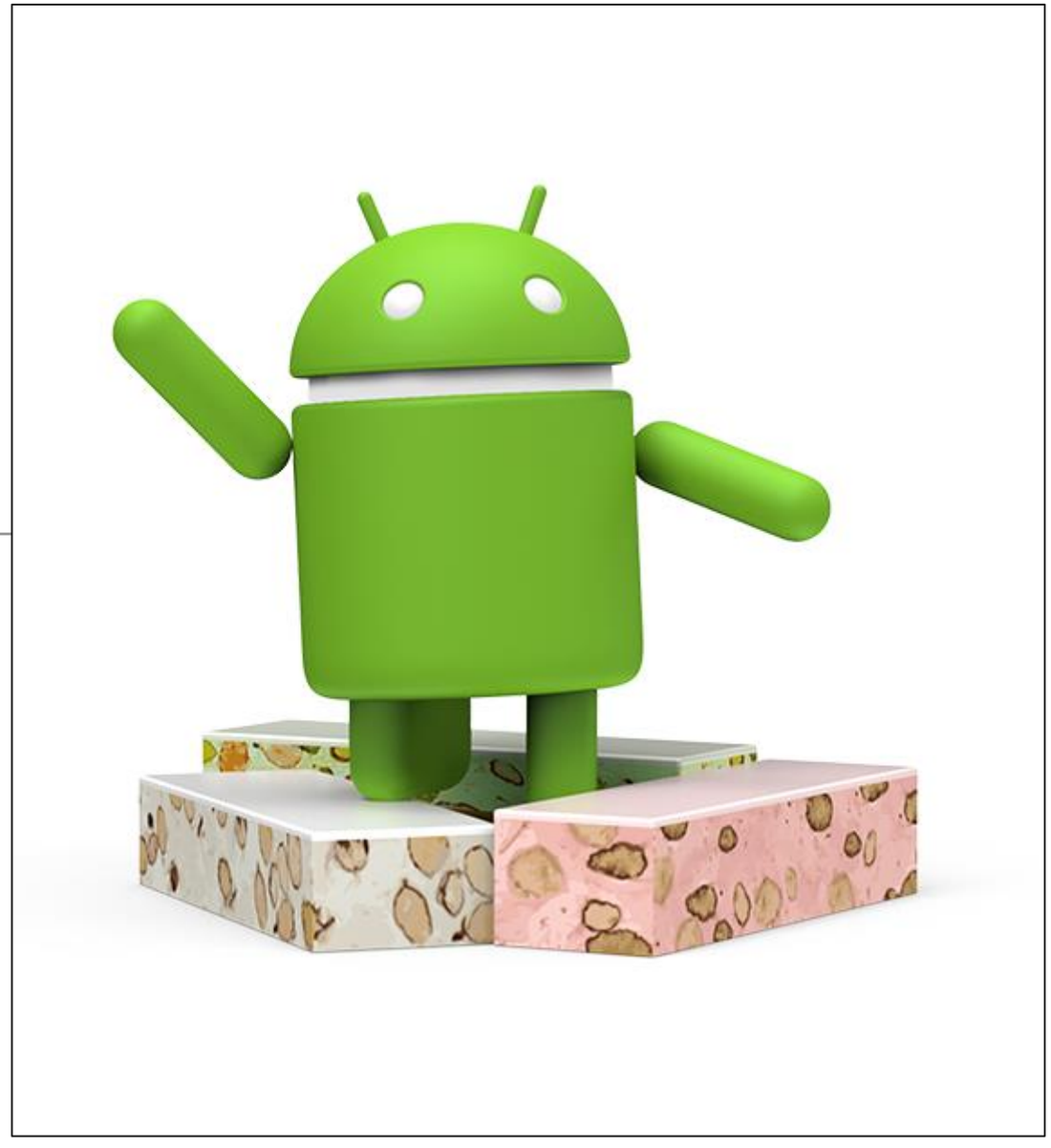
Waterford Institute *of* Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

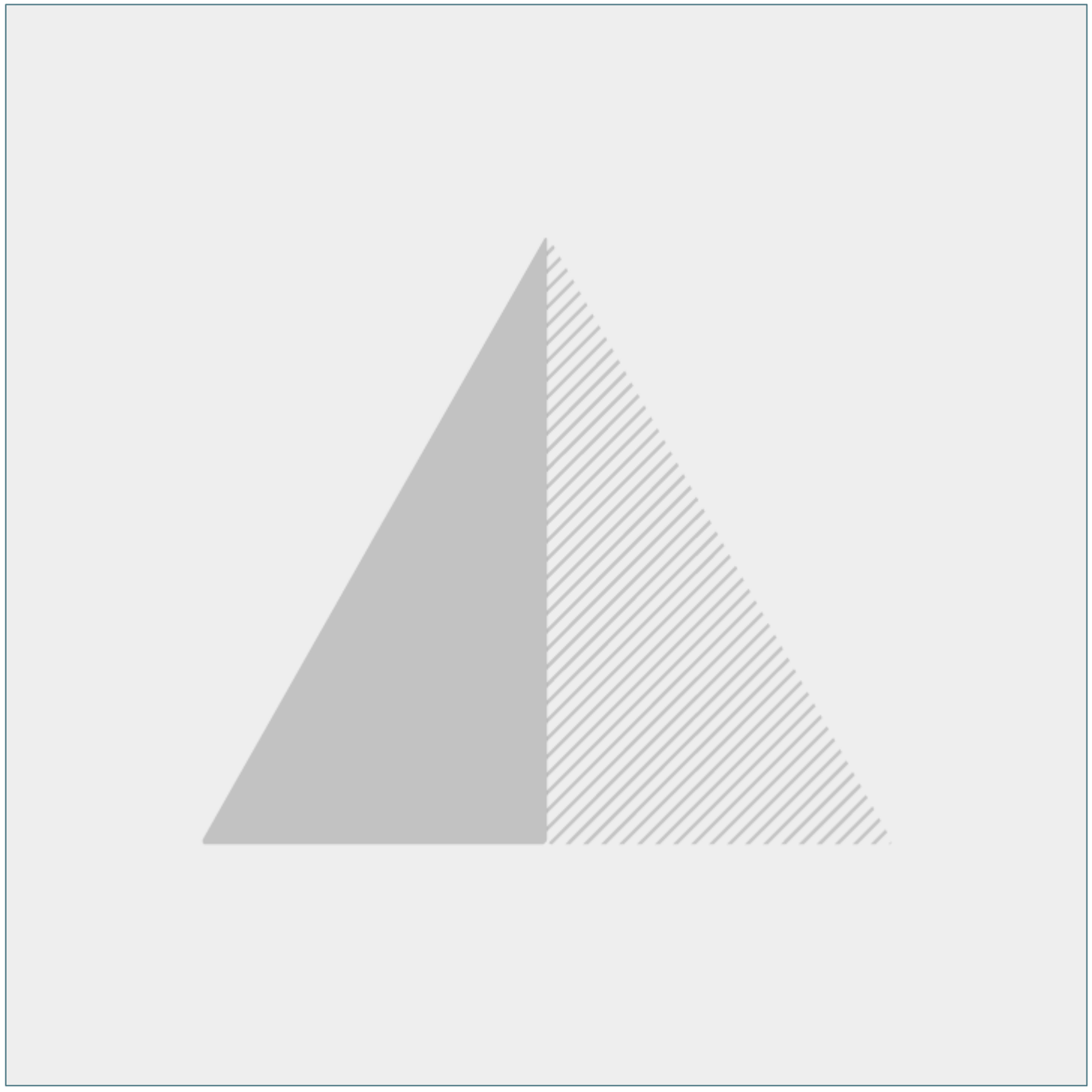
eLearning support unit

In-app Navigation Principles

(from developer.android.com
and material.io)



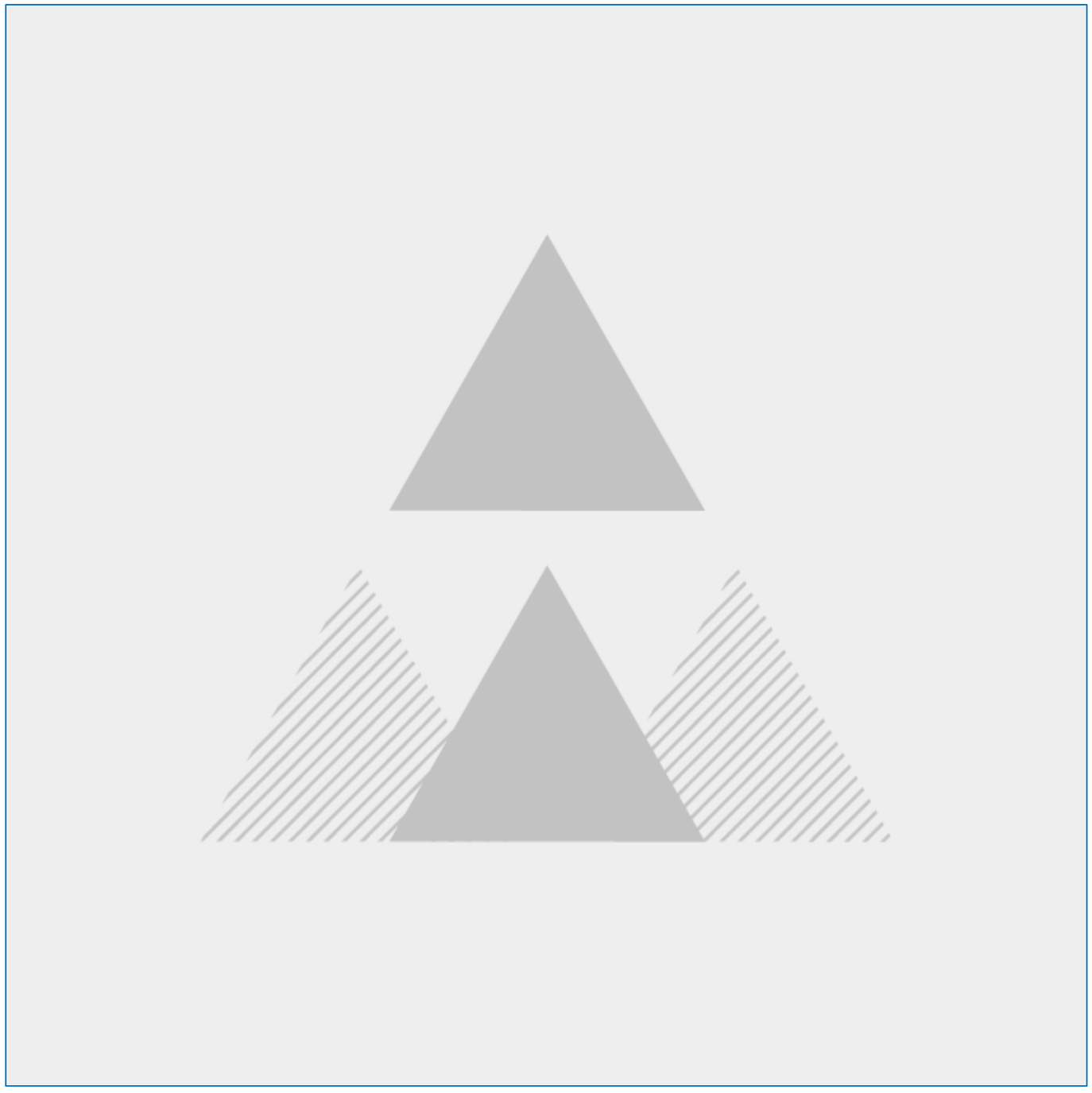
Hierarchy – Home



The **home** scene is the entrance to the app. It introduces the app and its navigation.

The home scene may vary the content it displays. A **stateful home** contains indications of a user's previous interaction with the app.

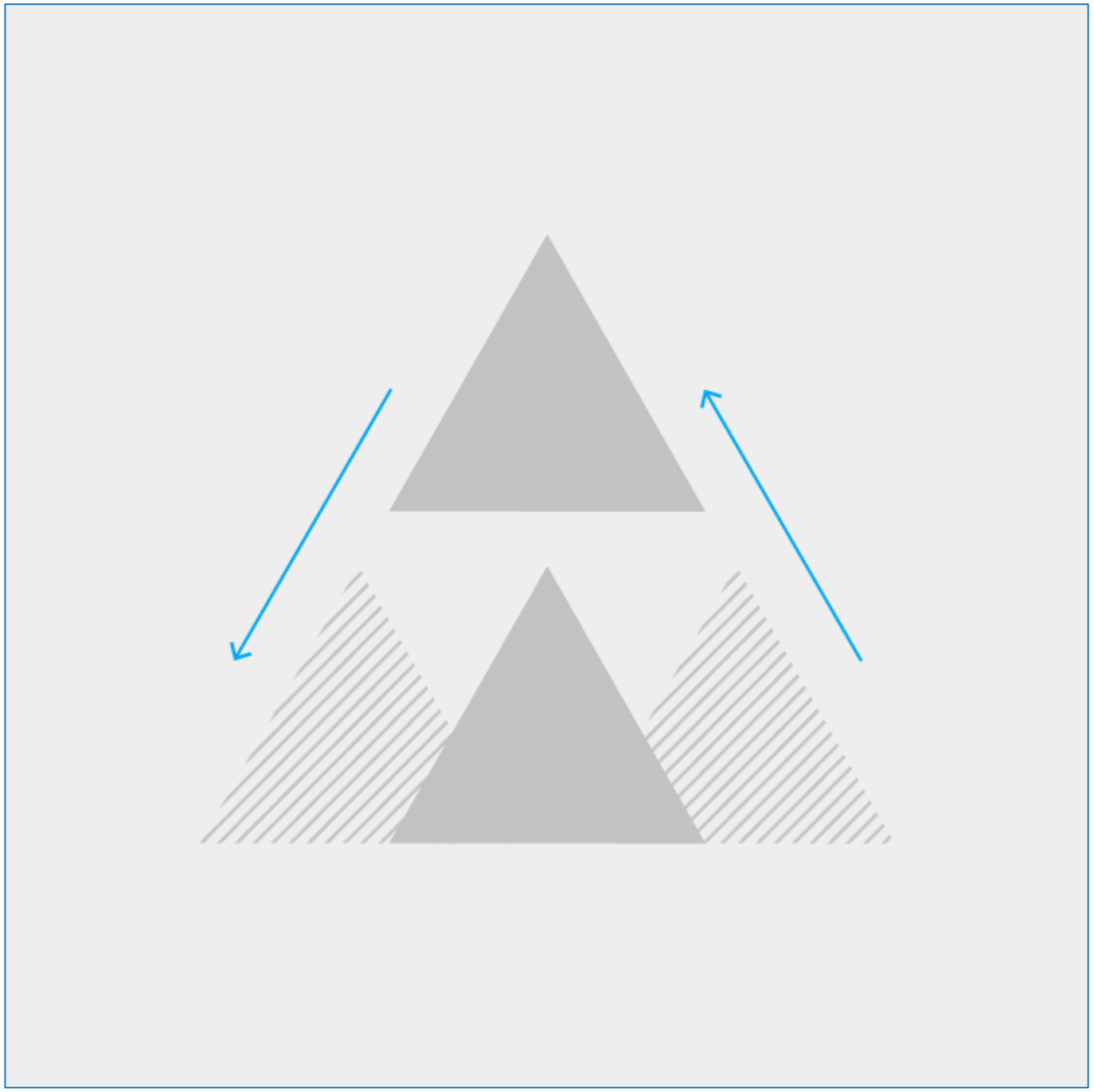
Hierarchy – Parent and Child



A higher level of navigational hierarchy is called a **parent**, and the level (or levels) below are referred to as **children**.

For example, the home screen is the parent to all other screens in an app.

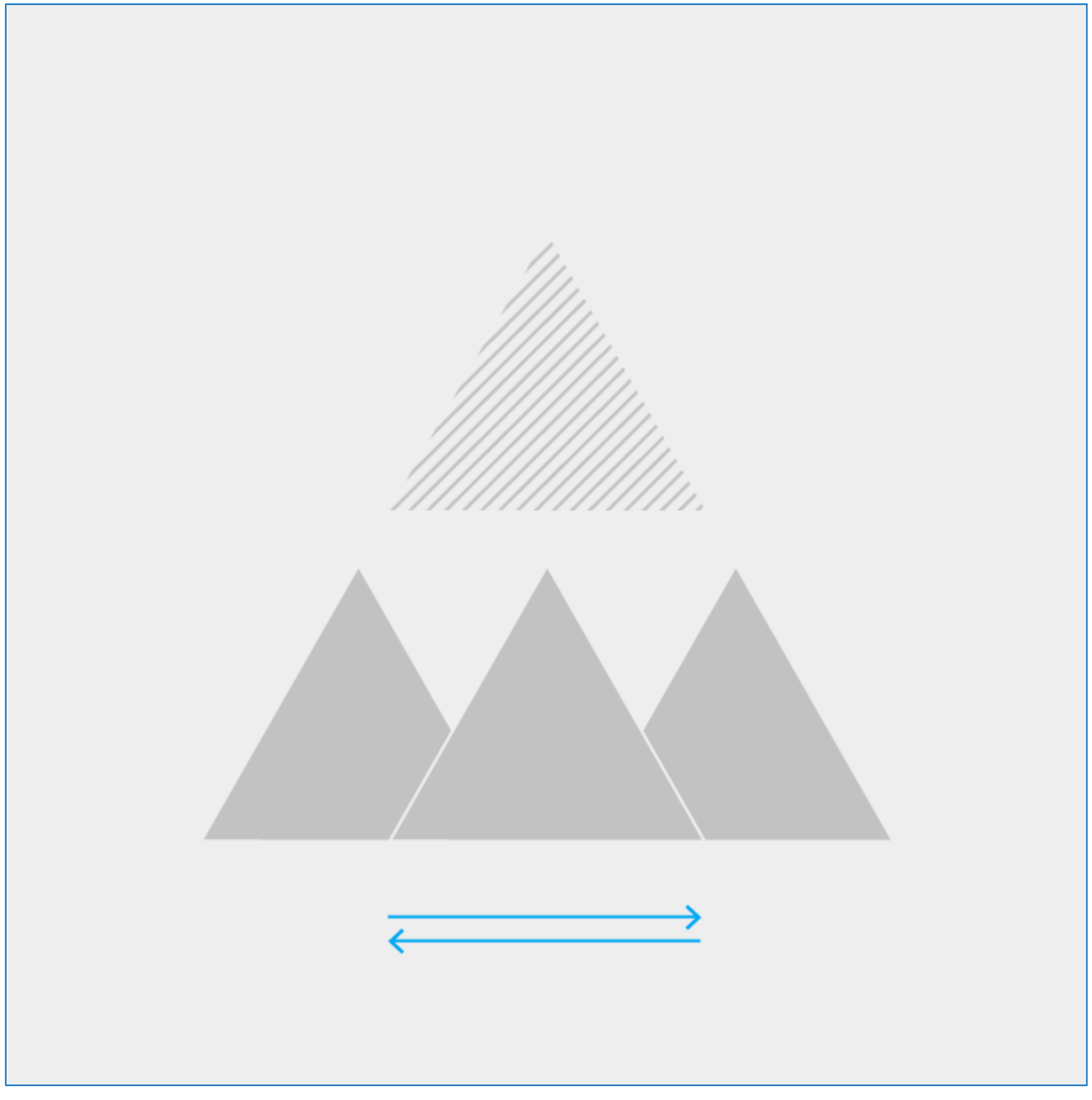
Hierarchy – Navigation



When you navigate more deeply into an app, you descend from a higher level of hierarchy to a lower one.

Moving from a parent scene to a child scene is **descending navigation**. Moving from child to parent is **ascending navigation**.

Hierarchy – Siblings



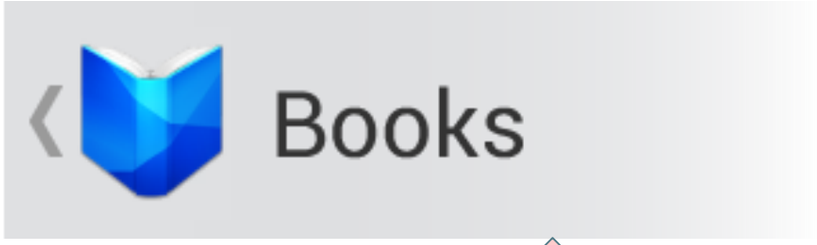
Scenes that have the same parent are called **siblings**. Apps that have multiple use cases, all of equal priority, usually contain many sibling scenes.

Lateral navigation is movement between siblings.

Navigation Patterns

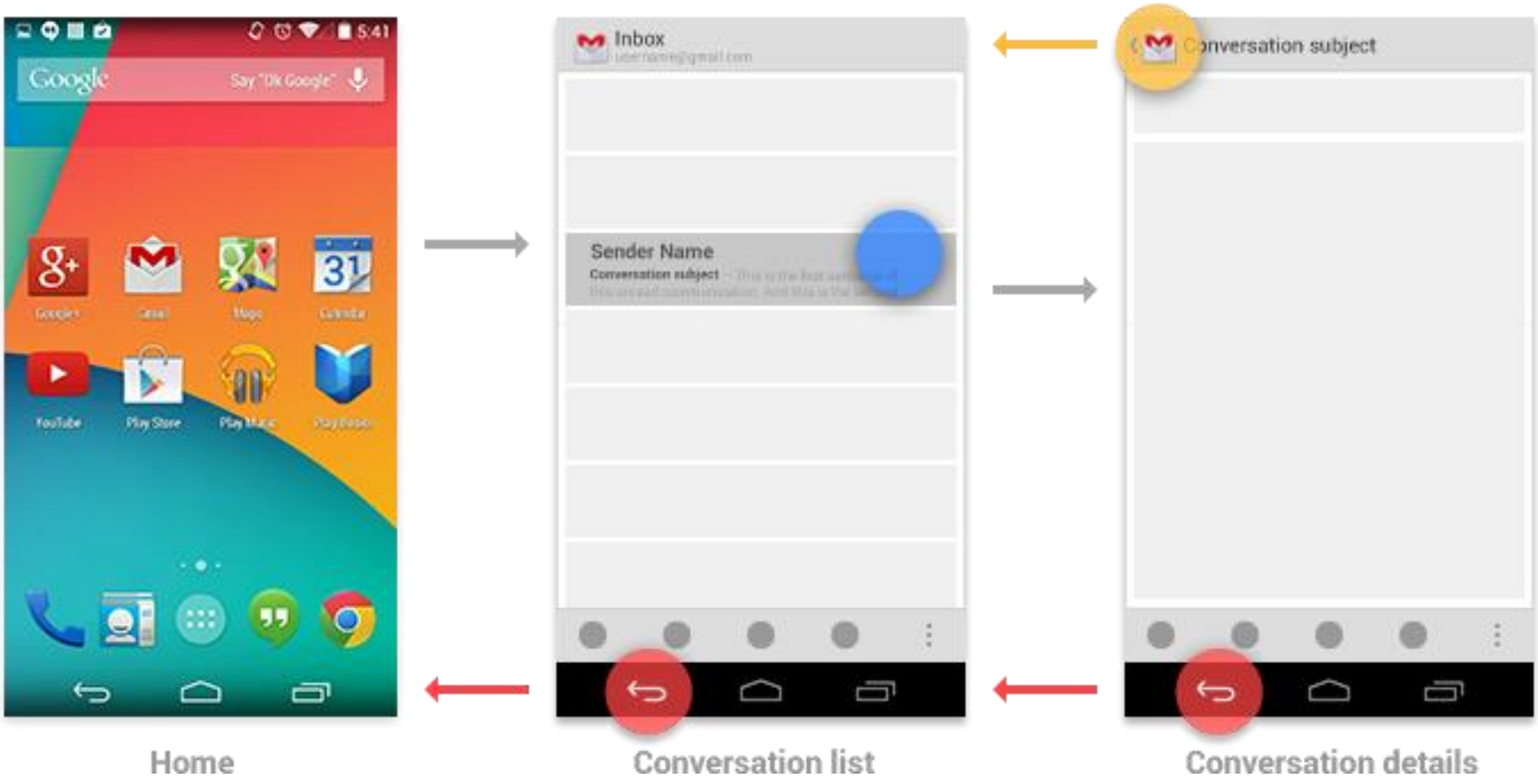
- Consistent Navigation is essential for good UX.
- Follow the guidelines for “Back” and “Up” makes your app predictable and reliable.

Back: navigates in reverse chronological order through the history of recently viewed screen. Back might take you out of your app.



Up: returns the user to the previous screen they viewed; keeps going up the hierarchy until the app’s home screen is reached.

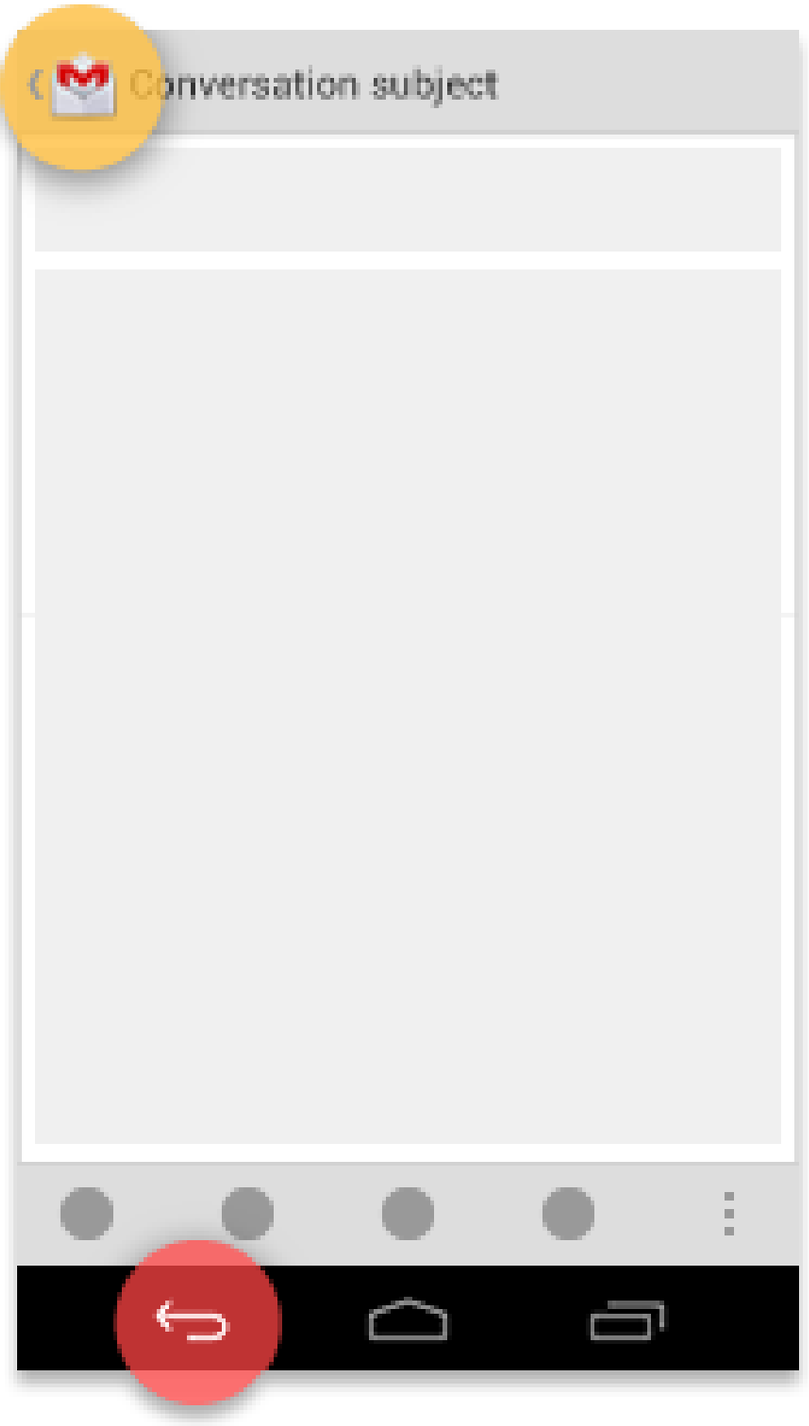
“Up” and “Back” Buttons



“Up” and “Back” Buttons

- **Up:** returns the user to the previous screen they viewed; keeps going until the Home page is reached.
  - e.g: if screen A displays a list of items, and selecting an item leads to screen B (which presents that item in more detail), then screen B should offer an Up button that returns to screen A.
- If a screen is the topmost one in an app (that is, the app's home), it should not present an Up button.

Up

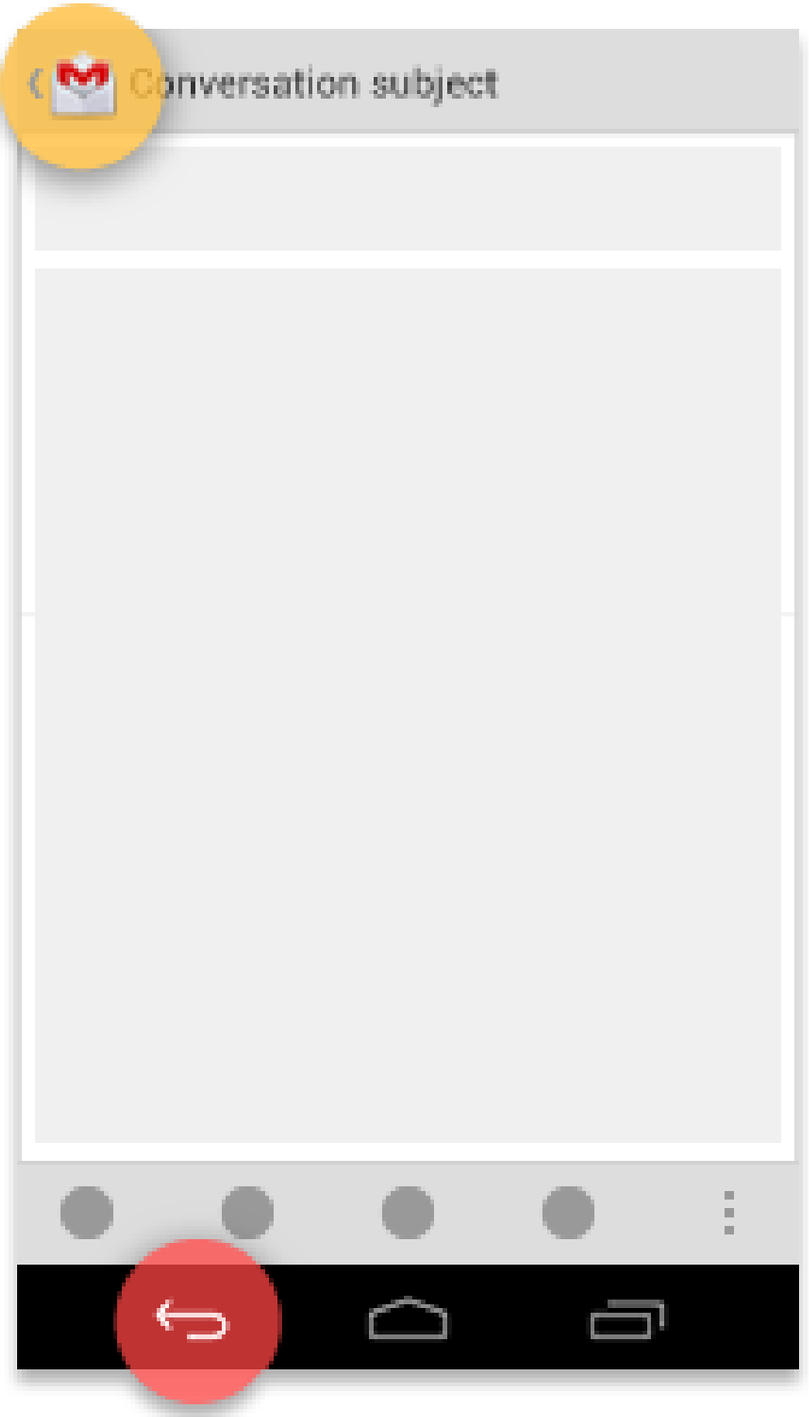


Back

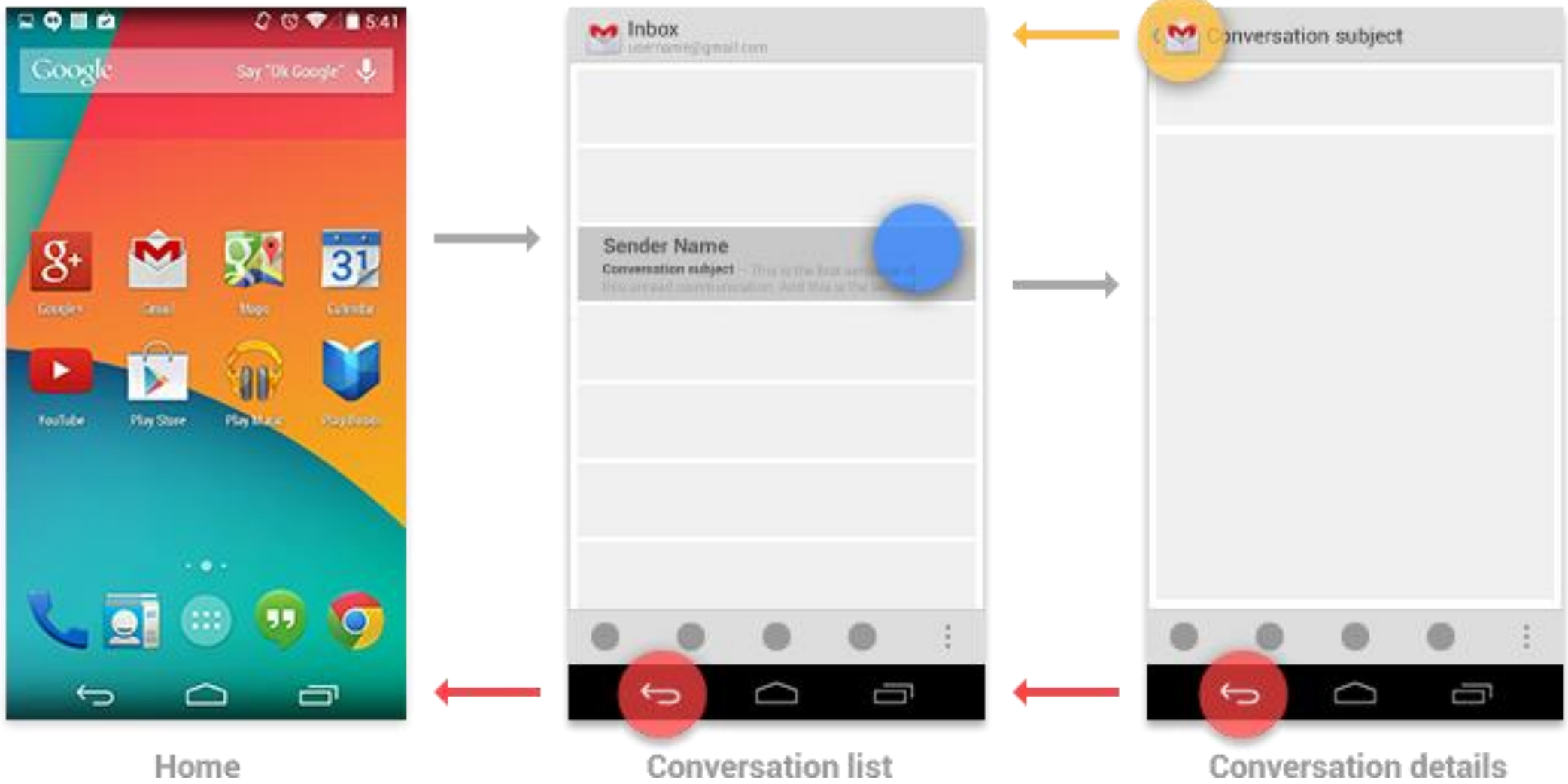
“Up” and “Back” Buttons

- **Back:** The System back button is used to navigate, in reverse chronological order, through the history of screens the user has recently worked with.
- Based on the temporal relationships between screens, rather than the app's hierarchy.

Up



Back



- When the previously viewed screen is also the hierarchical parent of the current screen, pressing the Back button has the same result as pressing an Up button—this is a common occurrence.

- However, unlike the Up button, which ensures the user remains within your app, the Back button can return the user to the Home screen, or even to a different app.

“Back" Button - Behaviours

- The Back button also supports a few behaviors not directly tied to screen-to-screen navigation e.g.:
  - Dismisses floating windows (dialogs, popups).
  - Dismisses contextual action bars, and removes the highlight from the selected items.
  - Hides the onscreen keyboard.

Navigating to screens with multiple entry points

- Sometimes a screen doesn't have a strict position within the app's hierarchy, and can be reached from multiple entry points
  - *e.g. a settings screen that could be reached from any other screen in your app → “Up" button should choose to return to the referring screen, behaving identically to "Back".*

Changing view within a screen

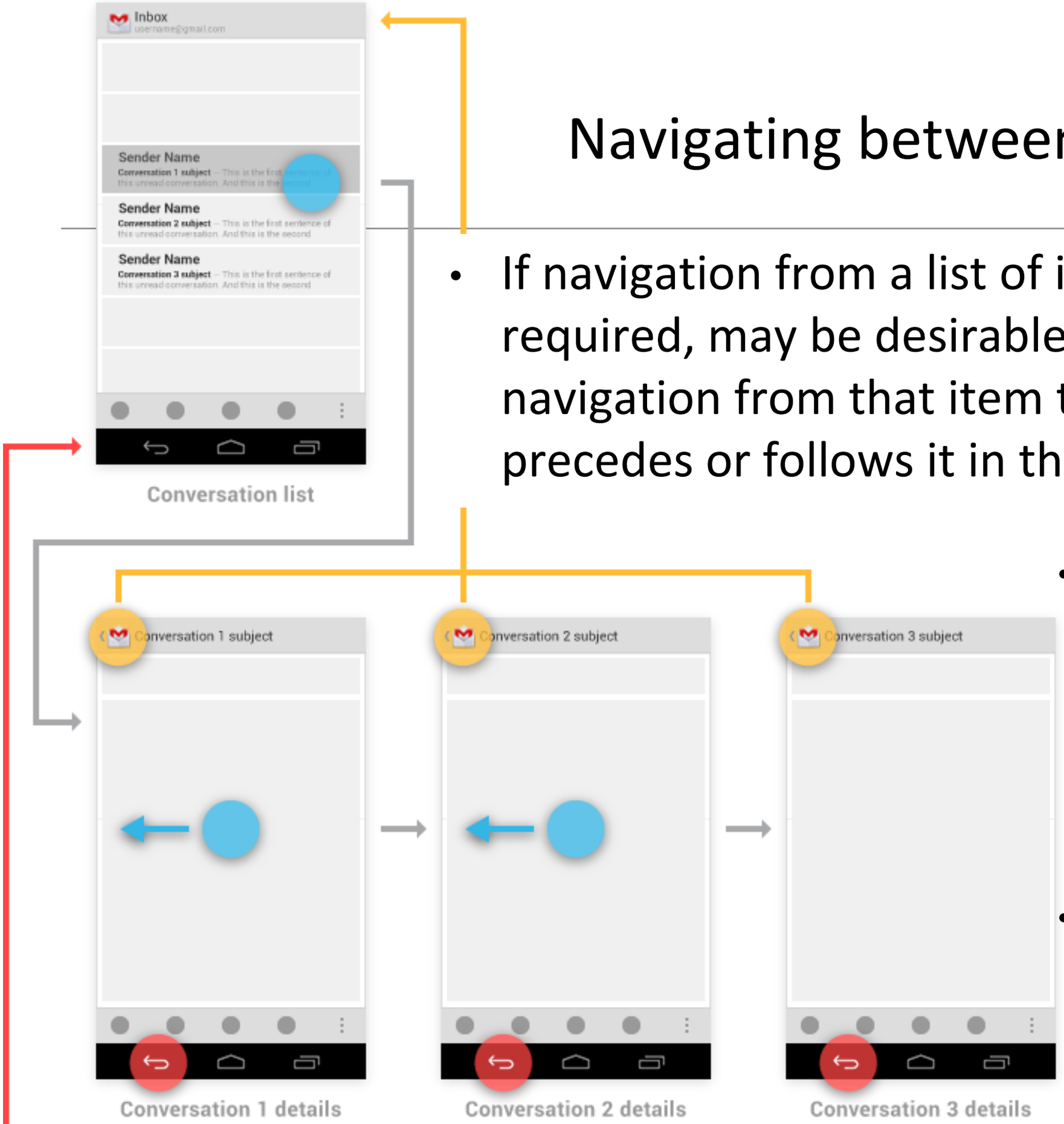
Changing view options for a screen does not change the behaviour of “Up” or “Back”:

- the screen is still in the same place within the app's hierarchy, and
- no new navigation history is created.

Examples of such view changes are:

- Switching views using tabs and/or left-and-right swipes
- Switching views using a dropdown (aka collapsed tabs)
- Filtering a list
- Sorting a list
- Changing display characteristics (such as zooming)

Navigating between sibling screens



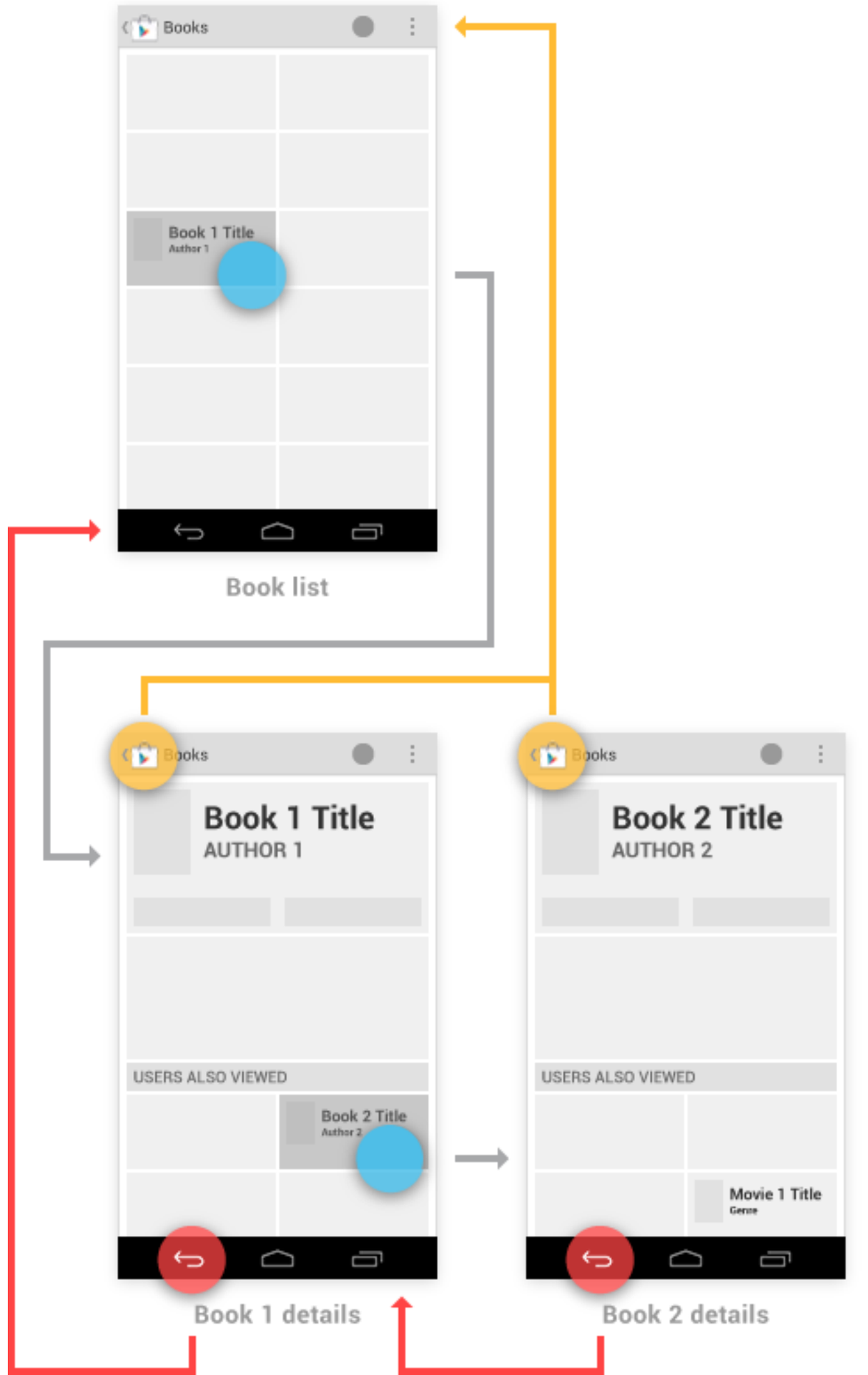
- If navigation from a list of items to a detail required, may be desirable to support direction navigation from that item to another one which precedes or follows it in the model.
  - E.g. in Gmail, swipe left or right from a conversation to view a newer or older message
  - Such navigation does not change the behavior of Up or Back.

Exception: when browsing between related detail views not tied together by the referring list.

- e.g.: when browsing in the Play Store between apps from the same developer, or albums by the same artist.

- In this case, following each link **does** create history, causing the "Back" button to step through each previously viewed screen.

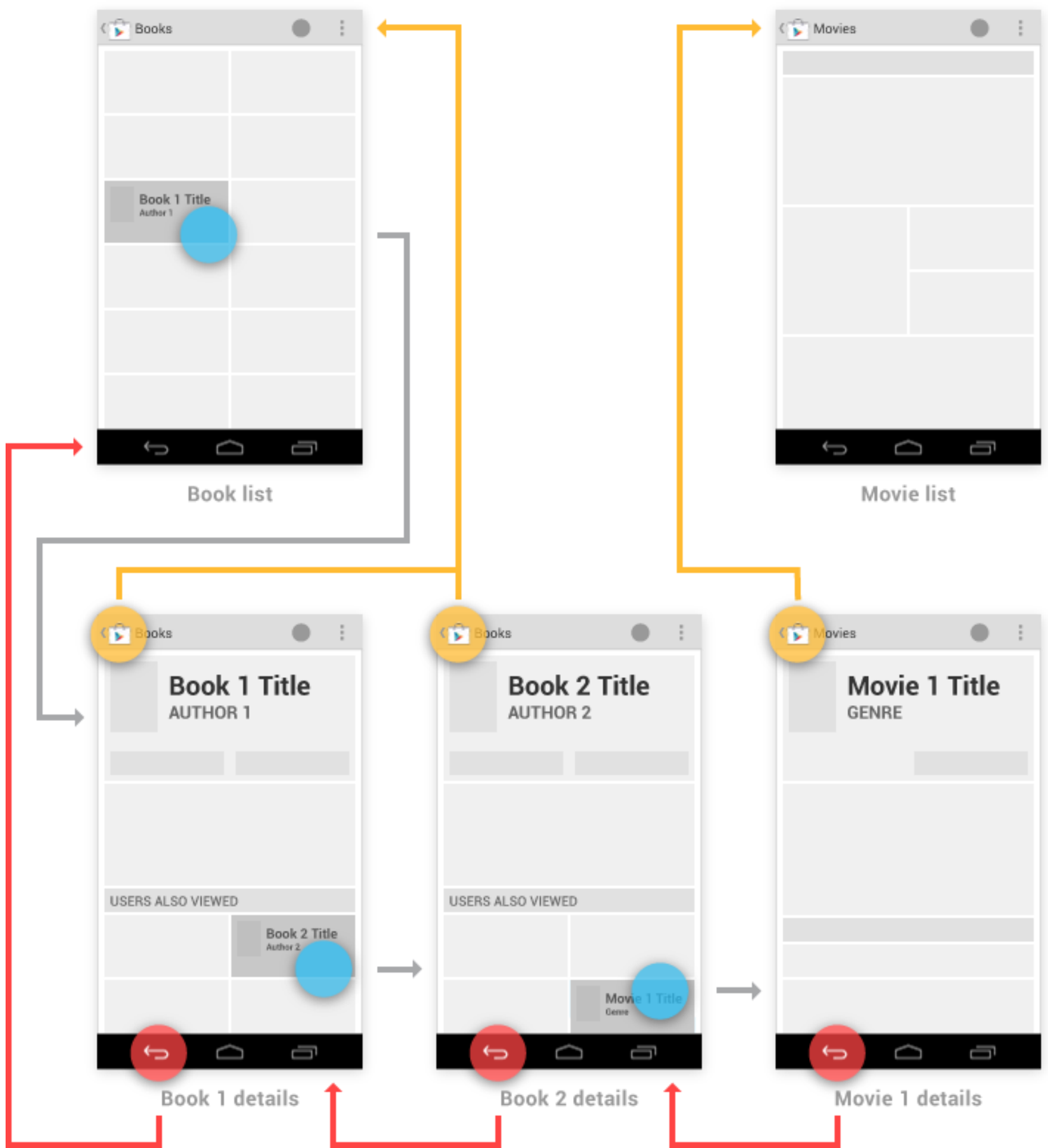
- "Up" should continue to bypass these related screens and navigate to the most recently viewed container screen.



You have the ability to make the Up behavior even smarter based on your knowledge of detail view e.g.:

- Play Store: imagine the user has navigated from the last Book viewed to the details for the Movie adaptation.

- In that case, Up can return to a container (Movies) which the user hasn't previously navigated through.



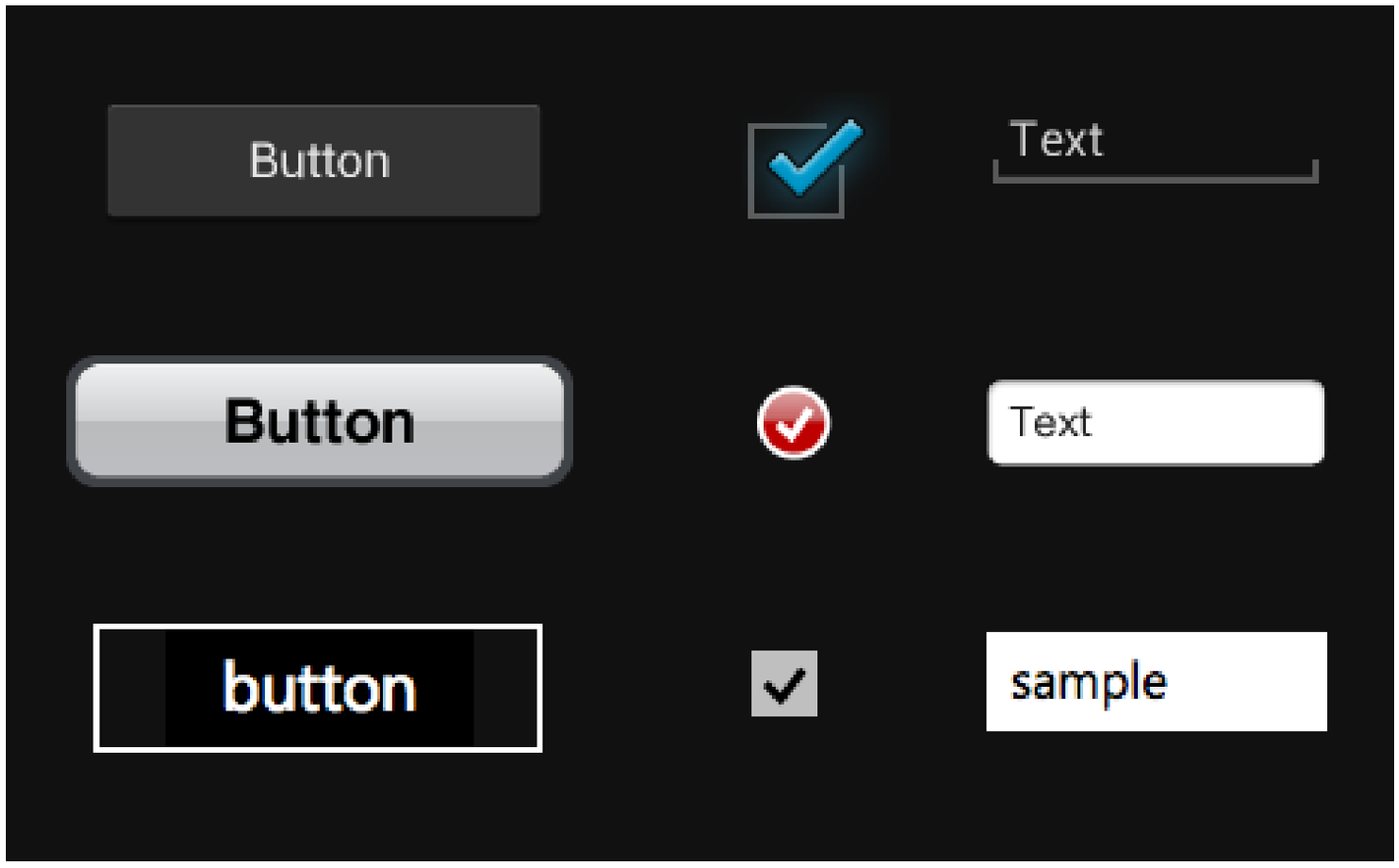
Pure android…

- Different platforms play by different rules and conventions.
- Design decisions that make perfect sense on one platform will look and feel misplaced in the context of a different platform.

https://developer.android.com/design/patterns/pure-android.html

Pure android...Tip 1

Don't mimic UI elements from other platforms.

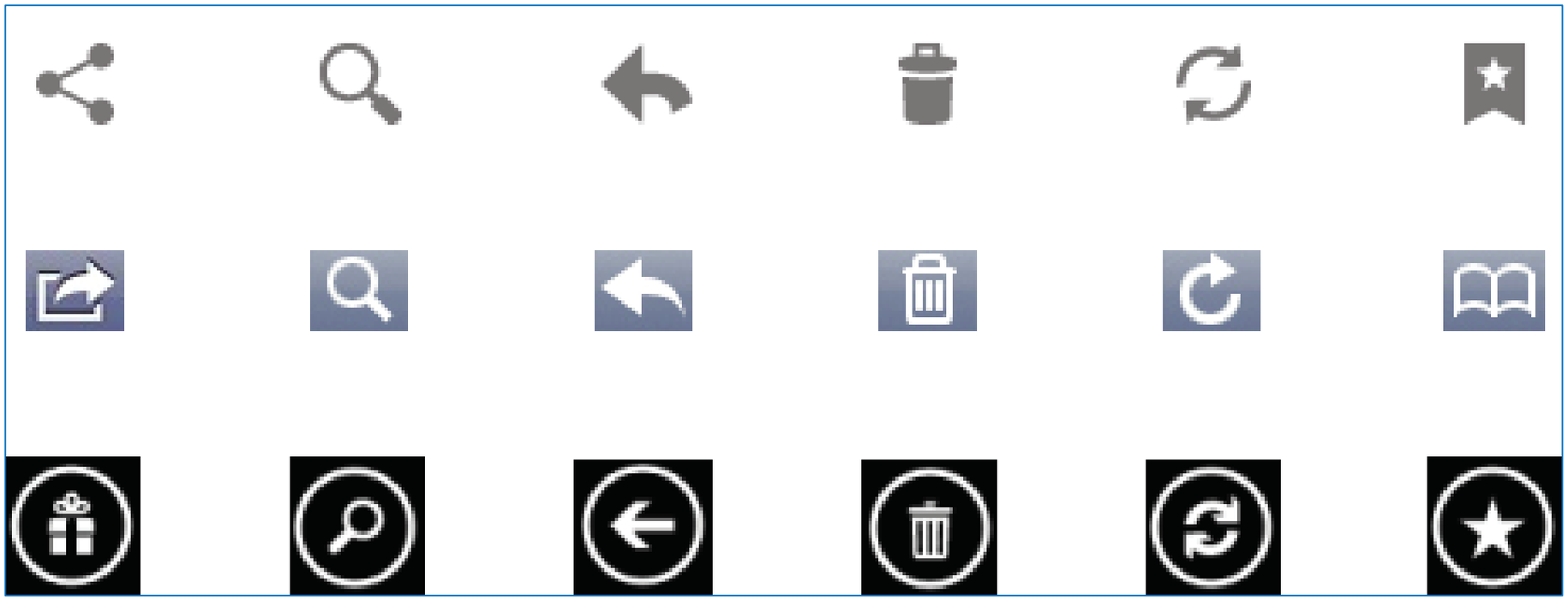


Sampling of UI elements from Android, iOS, and Windows Phone.

https://developer.android.com/design/patterns/pure-android.html

Pure android…Tip 2

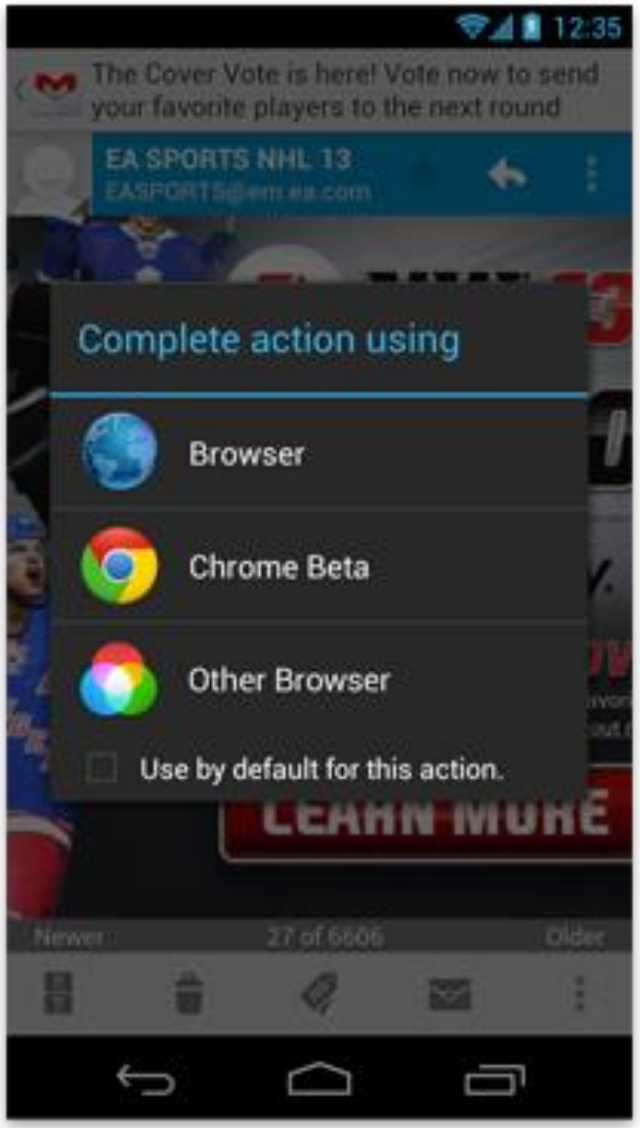
Don't carry over platform-specific icons.

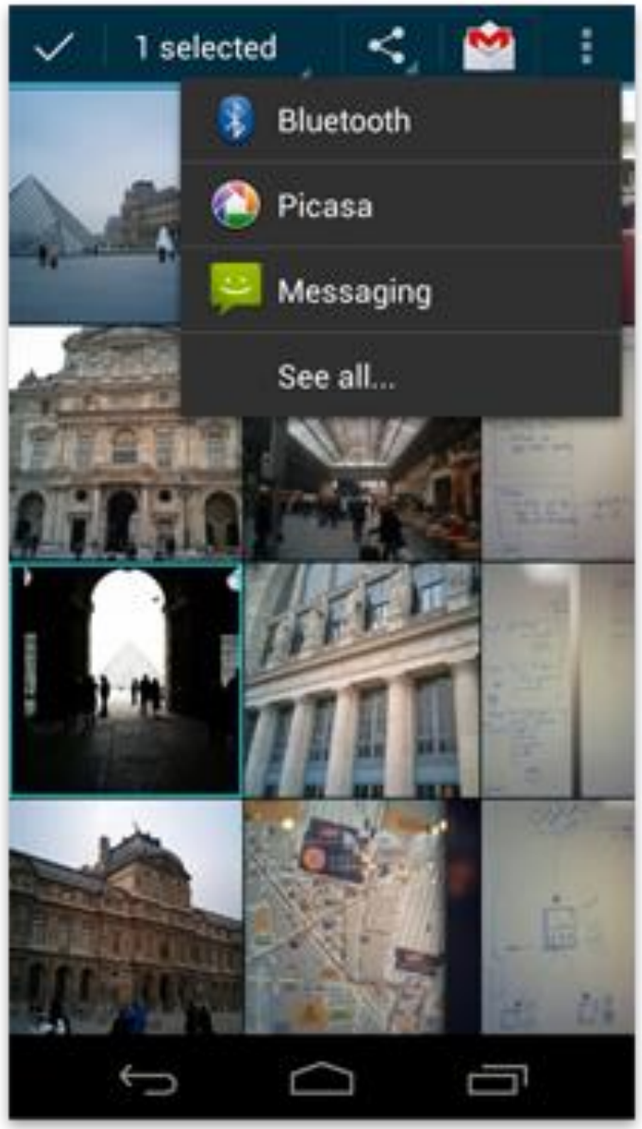


Sampling of icons from Android, iOS, and Windows Phone.

https://developer.android.com/design/patterns/pure-android.html

Pure android…Tip 3





Don’t hardcode links to other apps…instead use Android's intent API to launch an activity chooser which lists all applications that are set up to handle the particular request.

https://developer.android.com/design/patterns/pure-android.html

Pure android...Tip 4

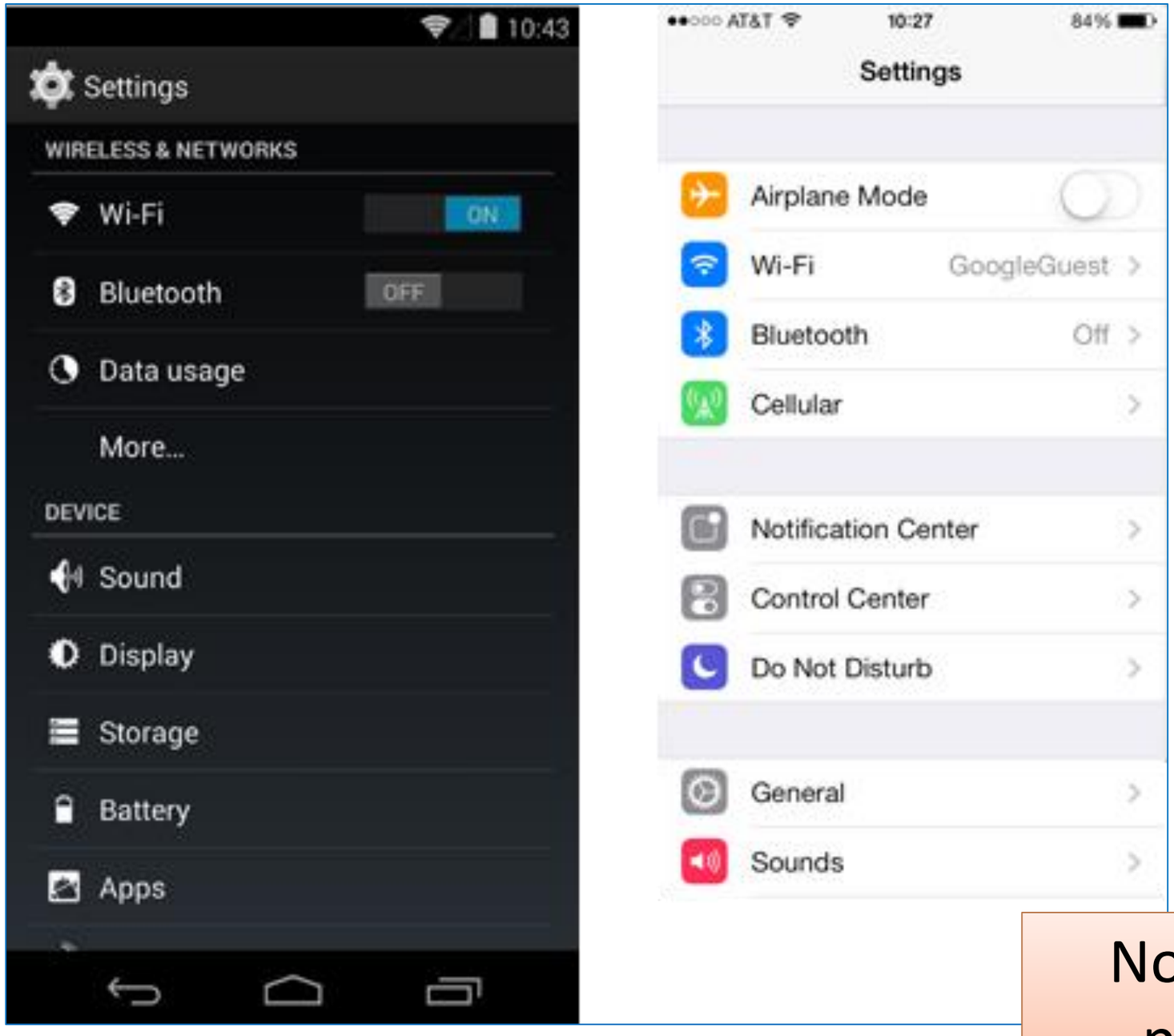
Don't use labelled back buttons on action bars.

Note: iphone uses labelled back buttons.

https://developer.android.com/design/patterns/pure-android.html

Pure android…Tip 4

Don't use right-pointing carets on line items.



Note: iphone uses pointing carets.

https://developer.android.com/design/patterns/pure-android.html

(Material drawn from
developer.android.com and material.io)

Questions?