Mobile Application Development

Higher Diploma in Science in Computer Science

Produced by

Eamonn de Leastar (edeleastar@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

Navigation Principles

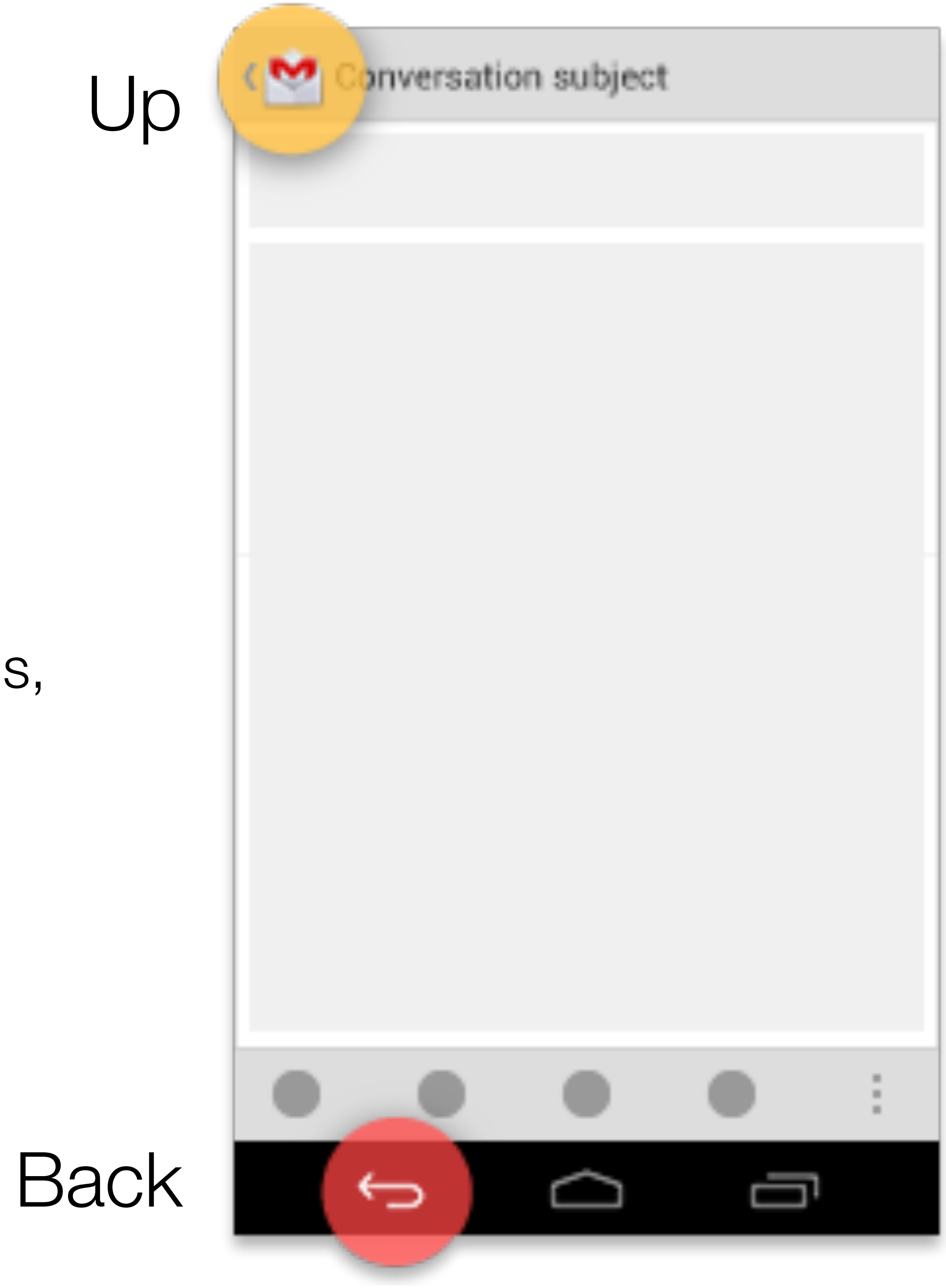
(from developer.android.com)

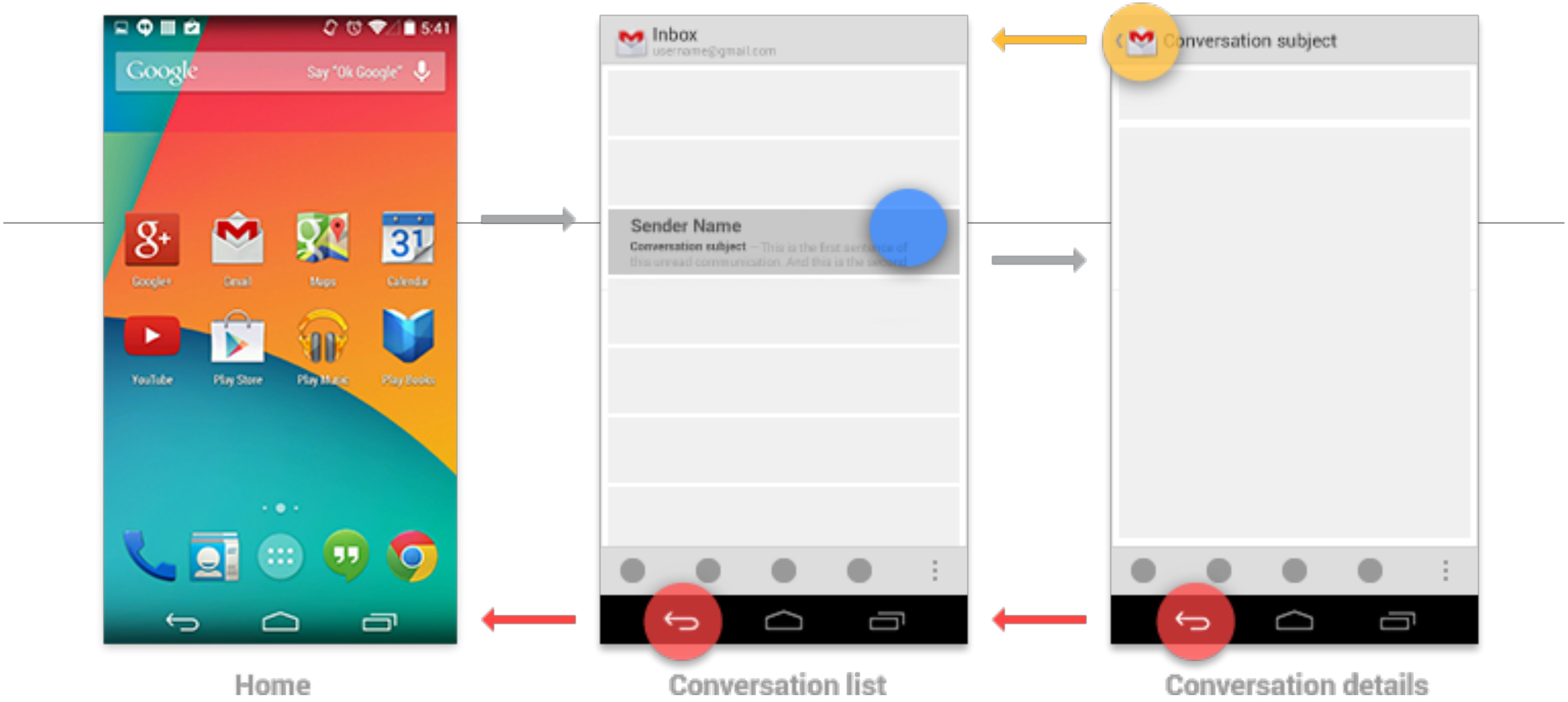
Navigation Within an App

- Back and Up
- Changing Screens
- Sibling Screens

“Up” and “Back” Buttons

- **Up:** used to navigate within an app based on the hierarchical relationships between screens.
  - 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.
- **Back:** 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.




Inbox
username@gmail.com
Conversation subject
Sender Name
Conversation subject — This is the first sentence of this unread communication. And this is the second
Home
Conversation list
Conversation details

- 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

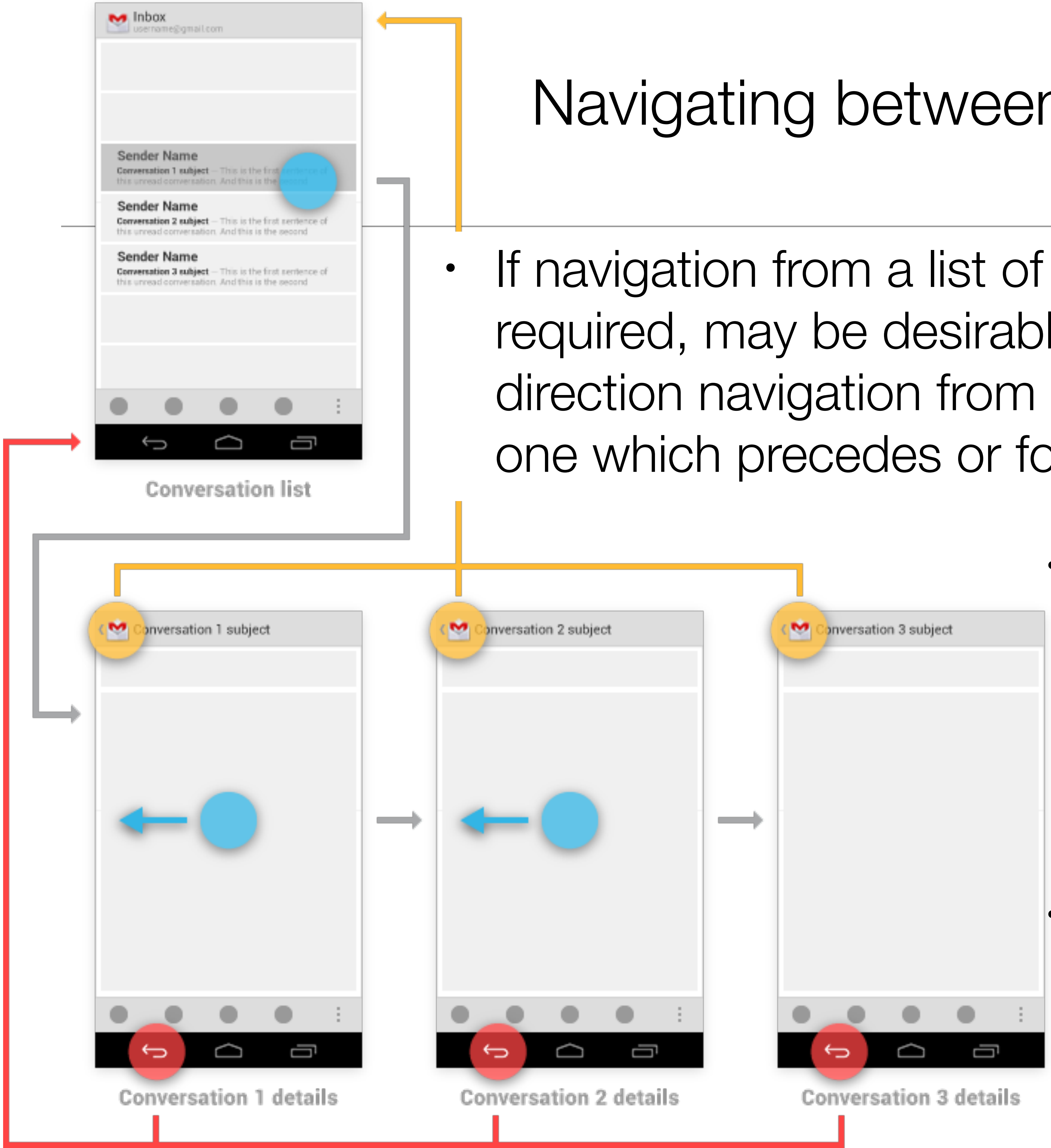
- The Back button also supports a few behaviors not directly tied to screen-to-screen navigation:
  - Dismisses floating windows (dialogs, popups)
  - Dismisses contextual action bars, and removes the highlight from the selected items
  - Hides the onscreen keyboard (IME)

Navigation Within Your App

- 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—such as a settings screen that can be reached from any other screen in your app.
  - In this case, the Up button should choose to return to the referring screen, behaving identically to Back.

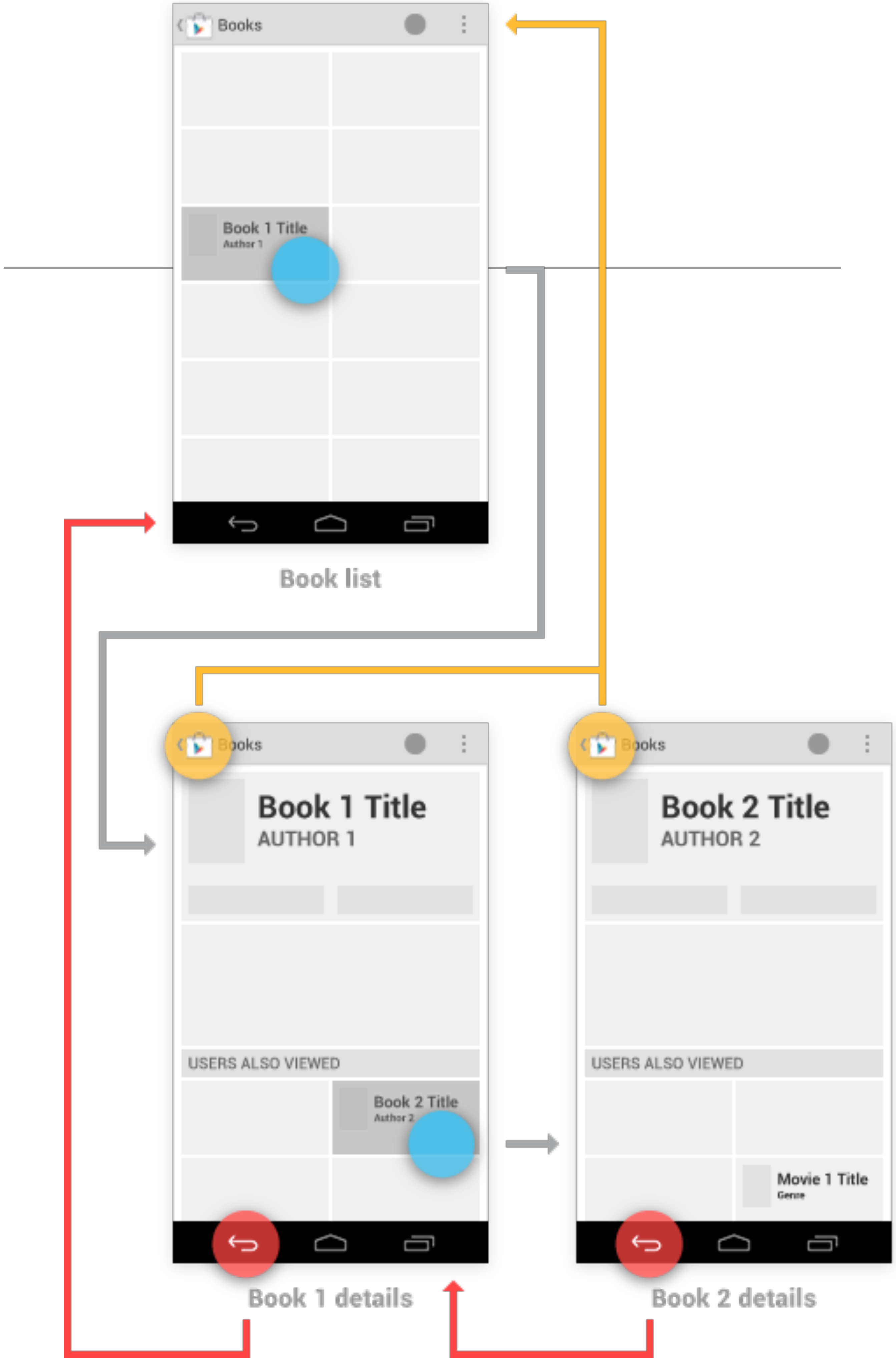
Navigating between sibling screens

- 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.

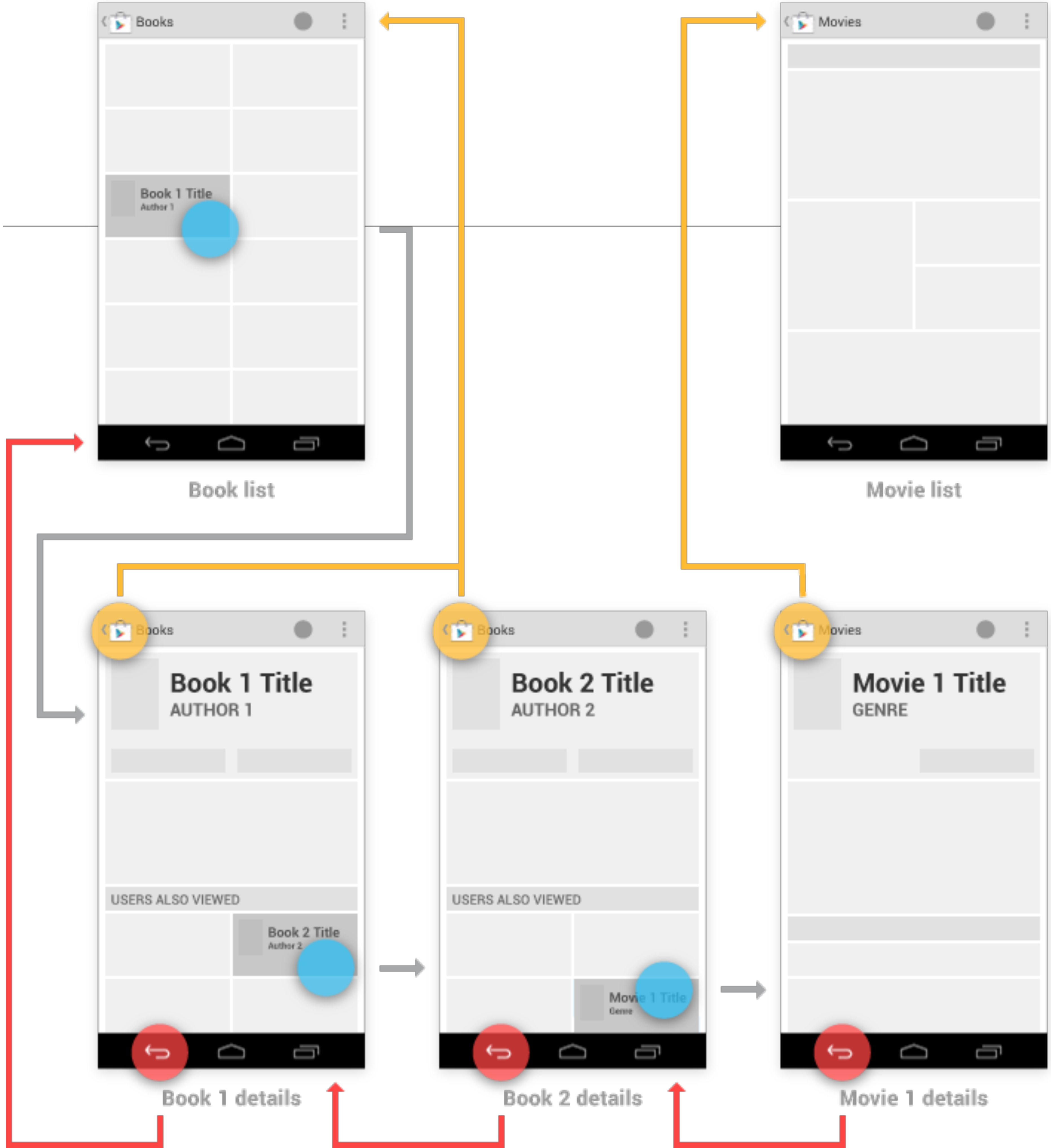

Inbox
username@gmail.com
Sender Name
Conversation 1 subject — This is the first sentence of this unread conversation. And this is the second
Sender Name
Conversation 2 subject — This is the first sentence of this unread conversation. And this is the second
Sender Name
Conversation 3 subject — This is the first sentence of this unread conversation. And this is the second
Conversation list
Conversation 1 subject
Conversation 2 subject
Conversation 3 subject
Conversation 1 details
Conversation 2 details
Conversation 3 details

- 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
    - Eg: 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.
- 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.



(Material drawn from developer.android.com)