

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



Navigation Principles

(from developer.android.com)

More Navigation...

Navigating Into an App

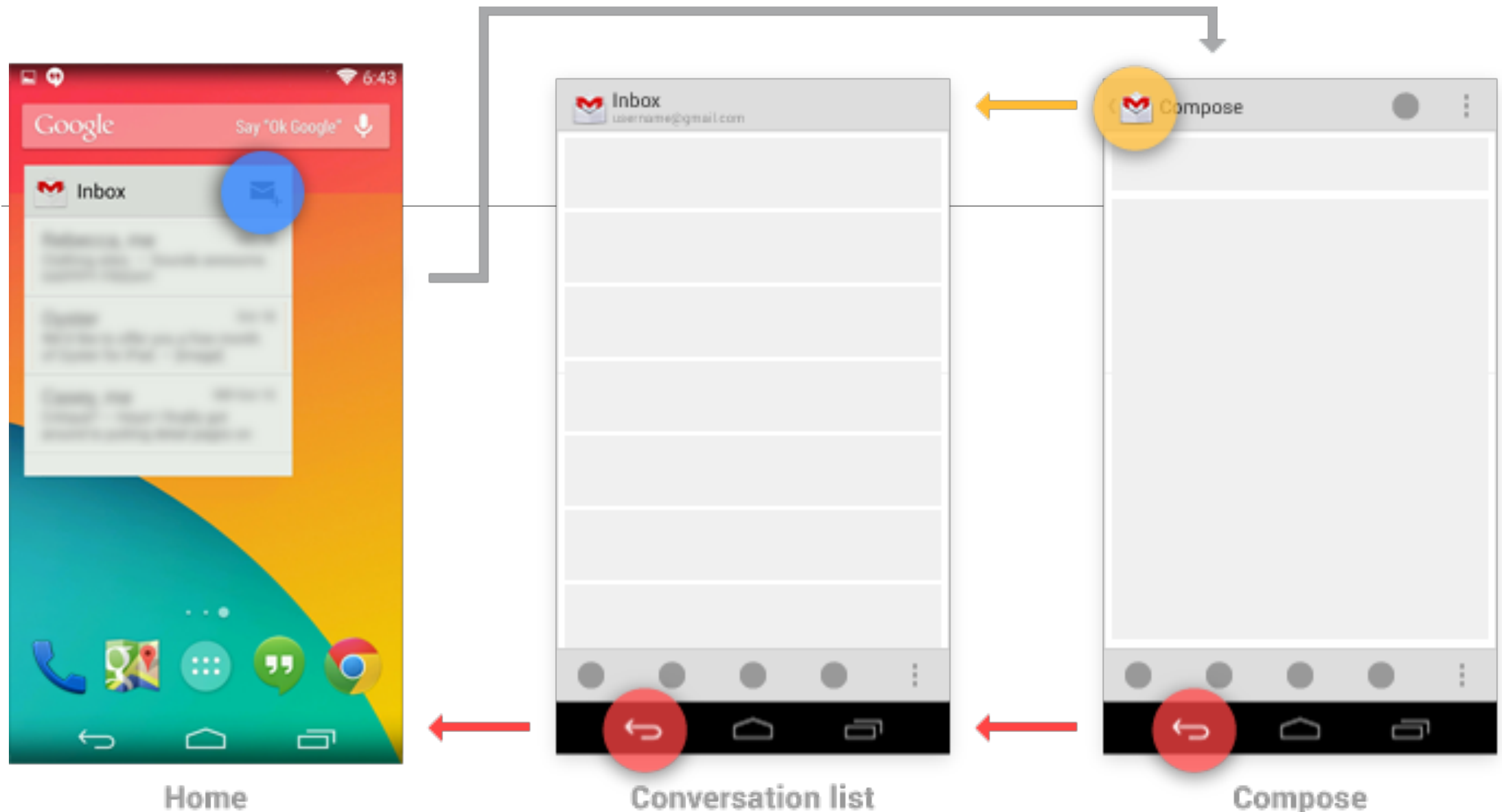
- Widgets
- Notifications
- Pop-ups

Navigation Between Apps

- Activities
- Tasks
- Intents

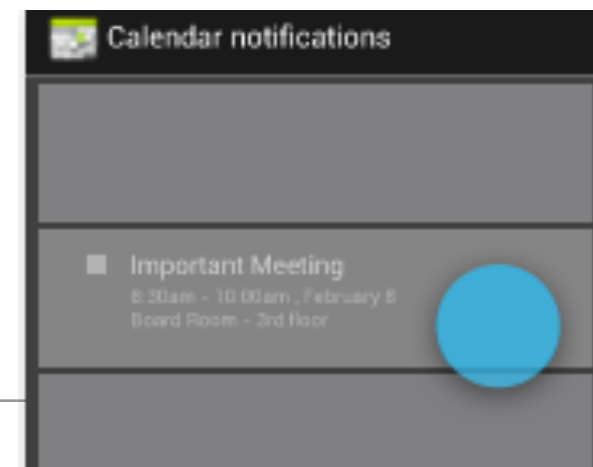
Navigation into an App via Home Screen Widgets and Notifications

- Screen widgets or notifications can allow users navigate directly to screens deep within an app's hierarchy.
 - Eg. Gmail's Inbox widget and new message notification can both bypass the Inbox screen, taking the user directly to a conversation view.
- Handle the Up button as follows:
 - If the destination screen is reached from one particular screen within the app, Up should navigate to that screen.
 - Otherwise, Up should navigate to the topmost ("Home") screen of the app.
- Back button: make navigation more predictable by inserting into the task's back stack the complete upward navigation path to the app's topmost screen. Allows users who've forgotten how they entered an app to navigate to the app's topmost screen before exiting.



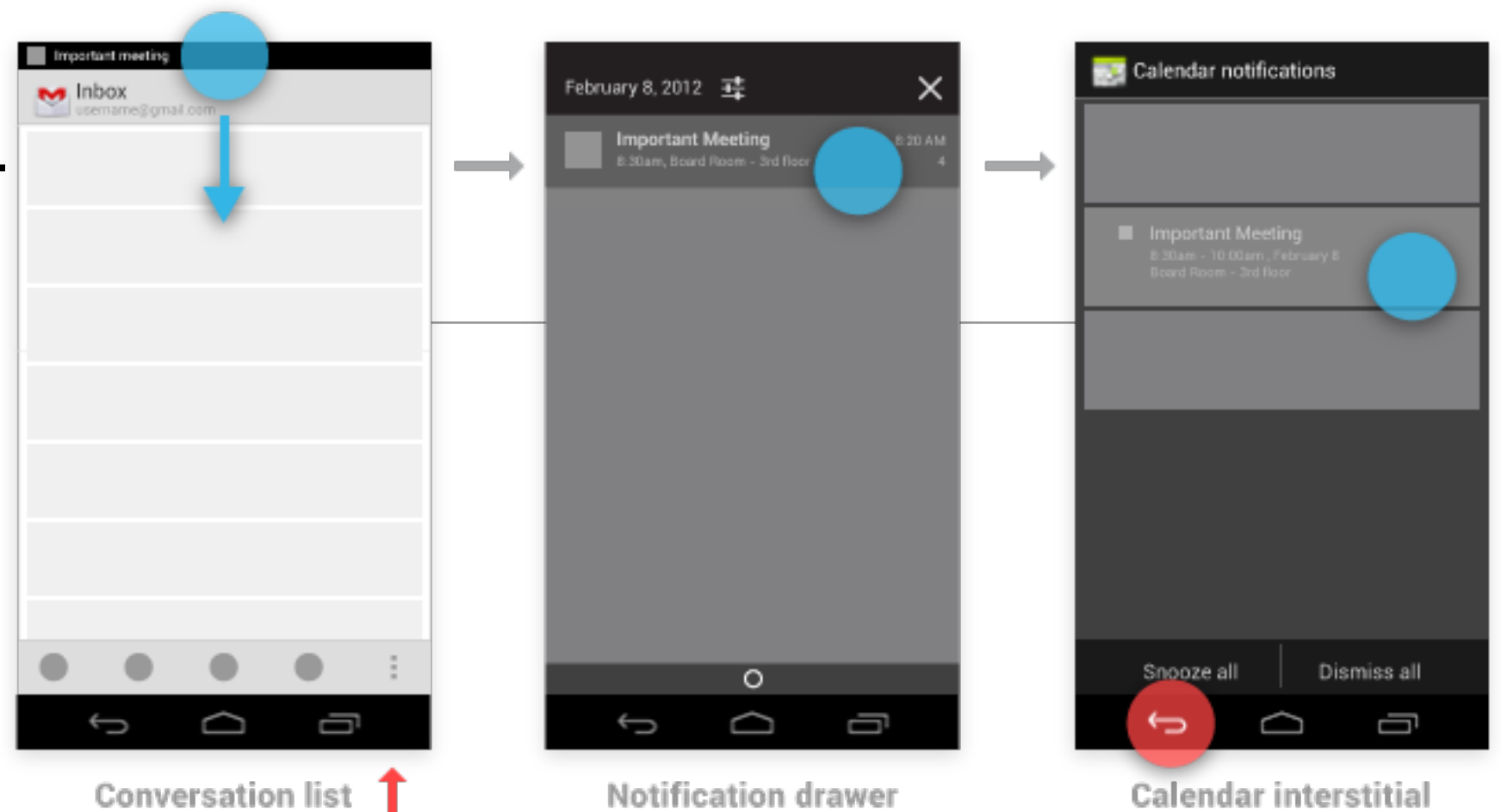
- Gmail's Home screen widget has a button for diving directly to its compose screen.
- Up or Back from the compose screen take the user to the Inbox, and from there the Back button continues to Home.

Indirect notifications

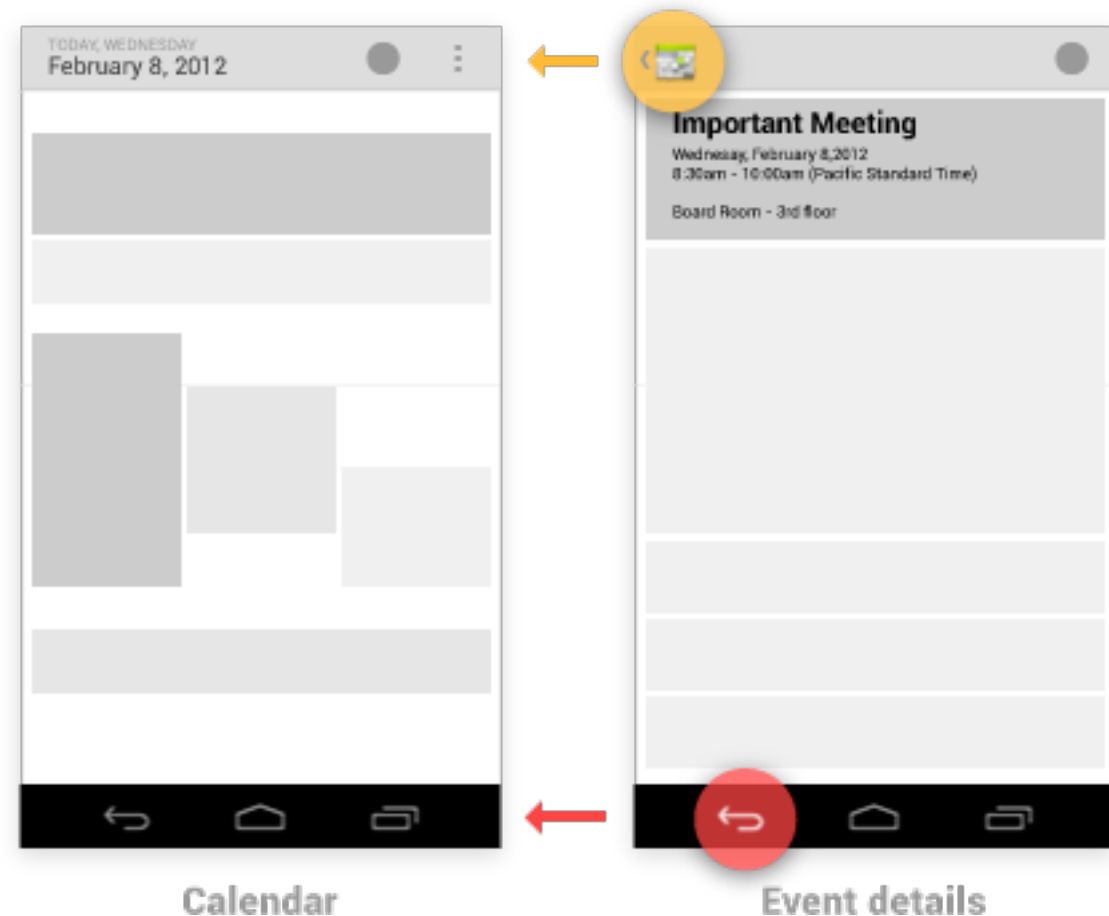


- When your app presents information about multiple events simultaneously, it can use a single notification that directs the user to an **interstitial** screen.
- This summarizes these events, and provides paths for the user to dive deeply into the app. Notifications of this style are called indirect notifications.
- Unlike standard (direct) notifications, pressing Back from an indirect notification's interstitial screen returns the user to the point the notification was triggered from—no additional screens are inserted into the back stack.
- Once the user proceeds into the app from its interstitial screen, Up and Back behave as for standard notifications - navigating within the app rather than returning to the interstitial.

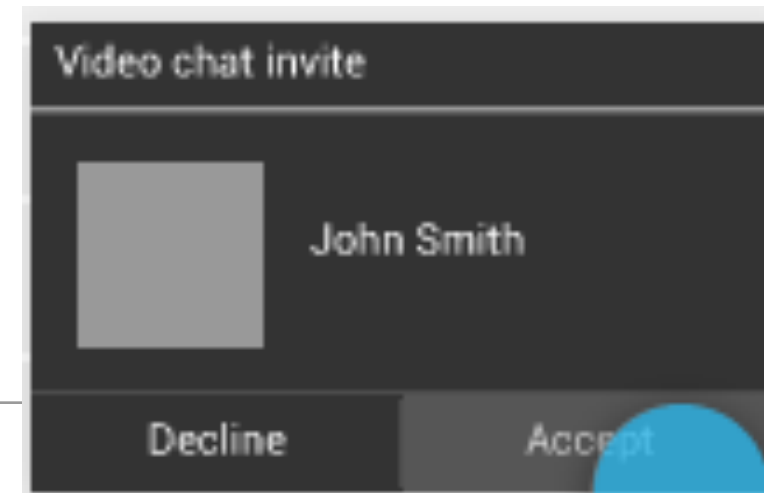
- Gmail receives an indirect notification from Calendar.
- Touching this notification opens the interstitial screen, which displays reminders for several different events.



- Touching Back from the interstitial returns the user to Gmail.
- Touching on a particular event takes the user away from the interstitial and into the full Calendar app to display details of the event.
- From the event details, Up and Back navigate to the top-level view of Calendar.

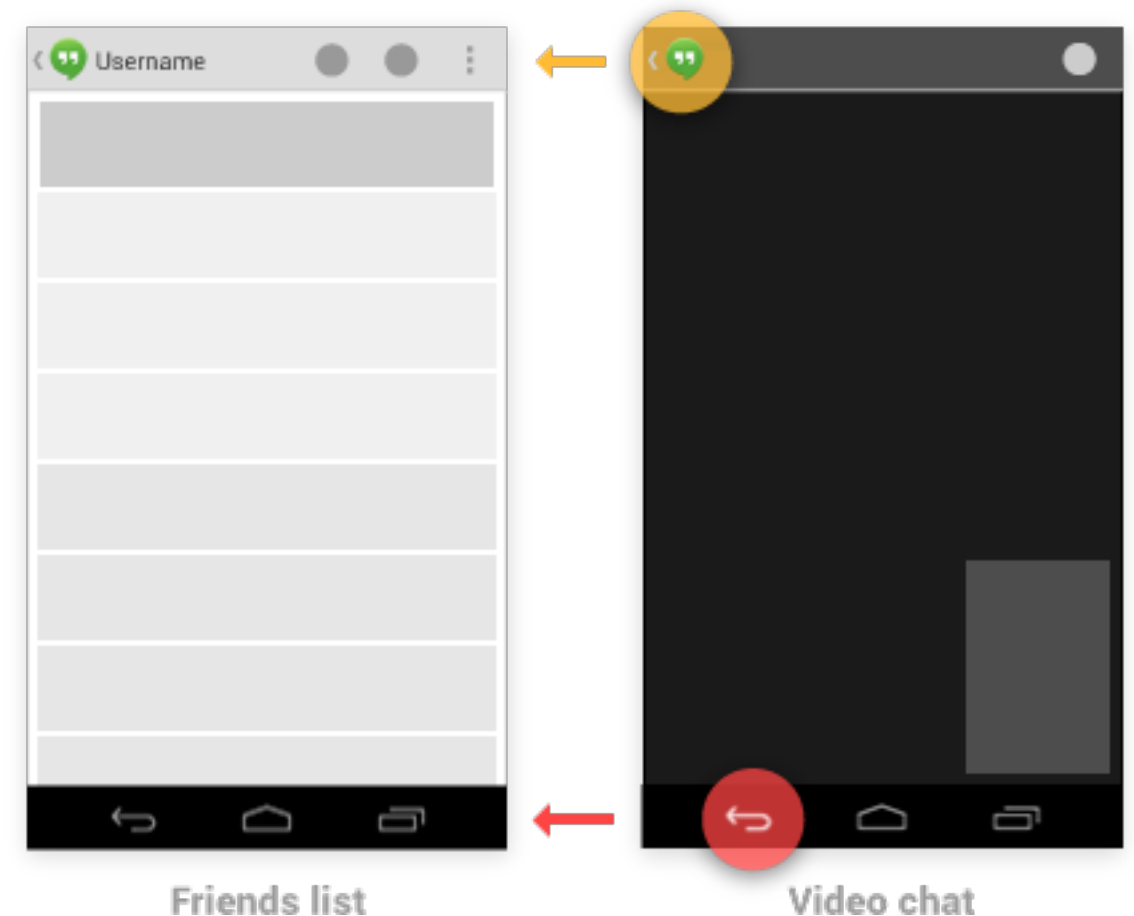
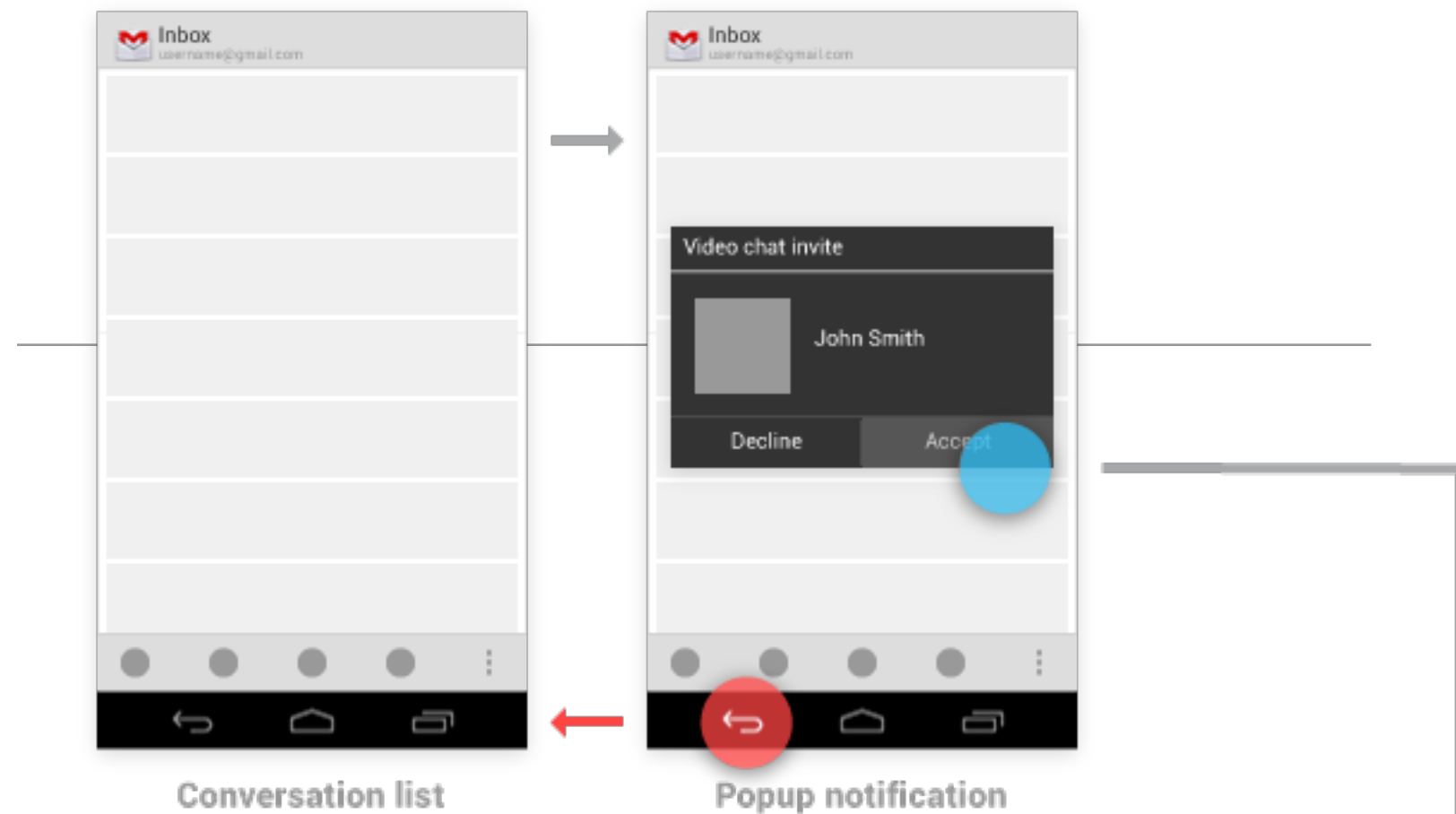


Pop-up notifications



- Pop-up notifications bypass the notification drawer, appearing directly in front of the user.
- They are rarely used, and should be reserved for occasions where a timely response is required and the interruption of the user's context is necessary.
- Eg: Talk uses this style to alert the user of an invitation from a friend to join a video chat, as this invitation will automatically expire after a few seconds.

- Pop-up notifications follow the behavior of an indirect notification's interstitial screen.
- Back dismisses the pop-up notification.
- If the user navigates from the pop-up into the notifying app, Up and Back follow the rules for standard notifications, navigating within the app.



Navigation Between Apps

- Activities
- Tasks
- Intents

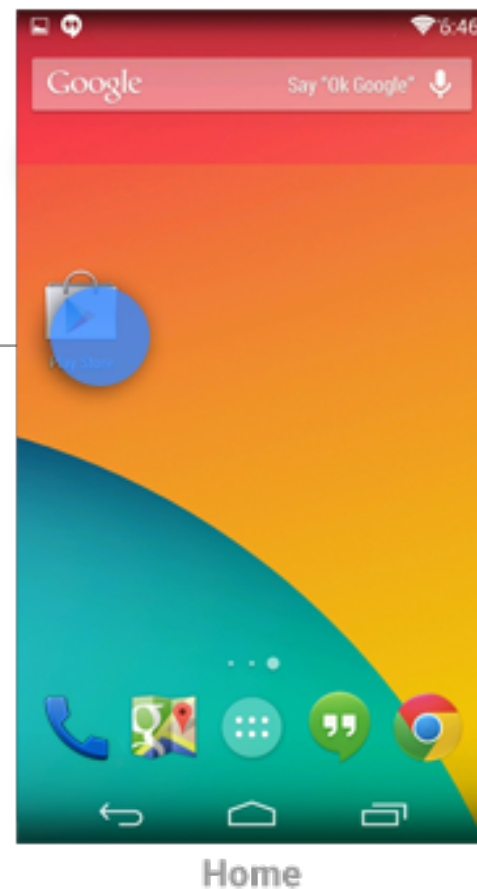
Motivation

- Give apps the ability to activate each other, navigating directly into another app.
- Eg, an app that needs to capture a photo can activate the Camera app, which will return the photo to the referring app.
- Enables the developer to leverage code from other apps
- Delivers a consistent experience for commonly performed actions.

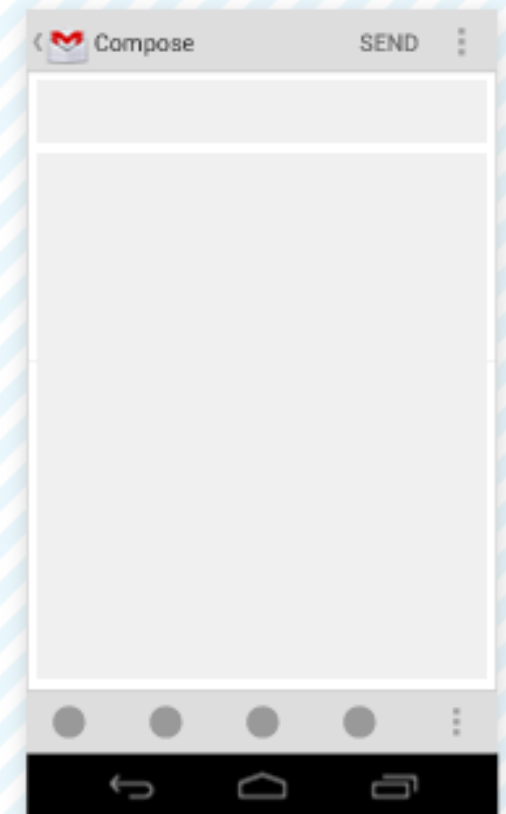
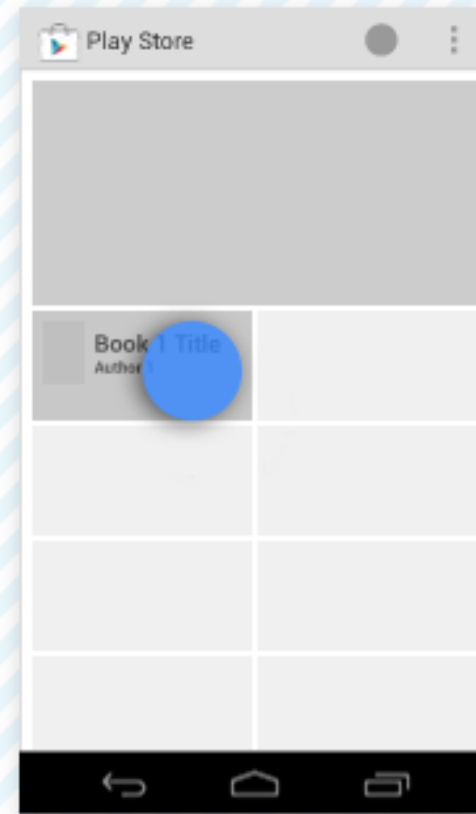
Activities, tasks, and intents

- **Activity:** an application component that defines a screen of information and all of the associated actions the user can perform. An app is a collection of activities, consisting of both the activities created by the app and those re-used from other apps.
- **Task:** is the sequence of activities a user follows to accomplish a goal. A single task can make use of activities from just one app, or may draw on activities from a number of different apps.
- **Implicit Intent:** is a mechanism for one app to signal it would like another app's assistance in performing an action. The other app can indicate which intents they can respond to. For common intents such as "Share", the user may have many apps installed that can fulfill that request.

Sharing

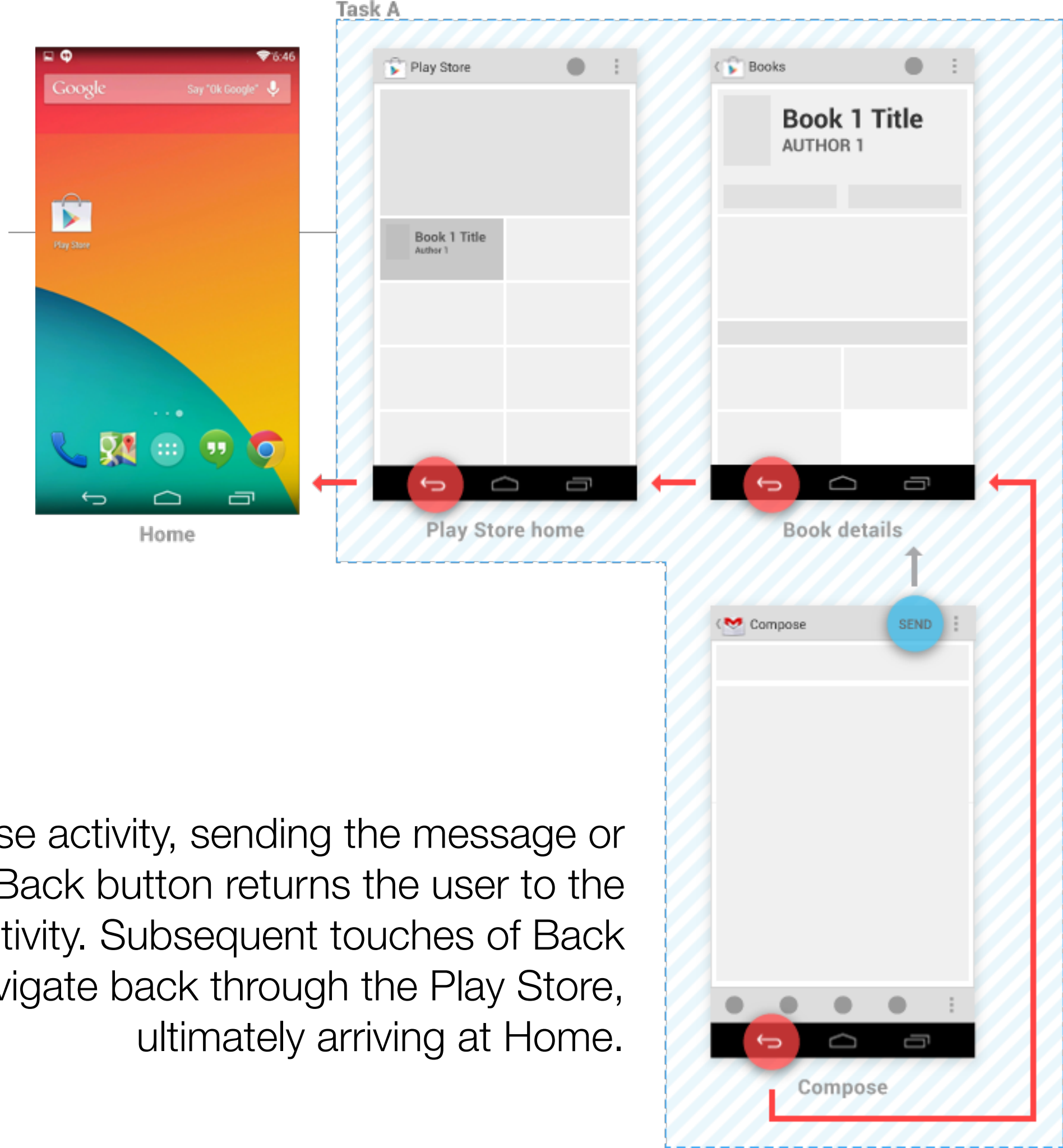


Task A



- Launching the Play Store app from Home begins new Task A.
- After navigating through the Play Store and touching a promoted book to see its details, the user remains in the same task, extending it by adding activities.
- Triggering the Share action prompts the user with a dialog listing each of the activities (from different apps) which have registered to handle the Share intent.

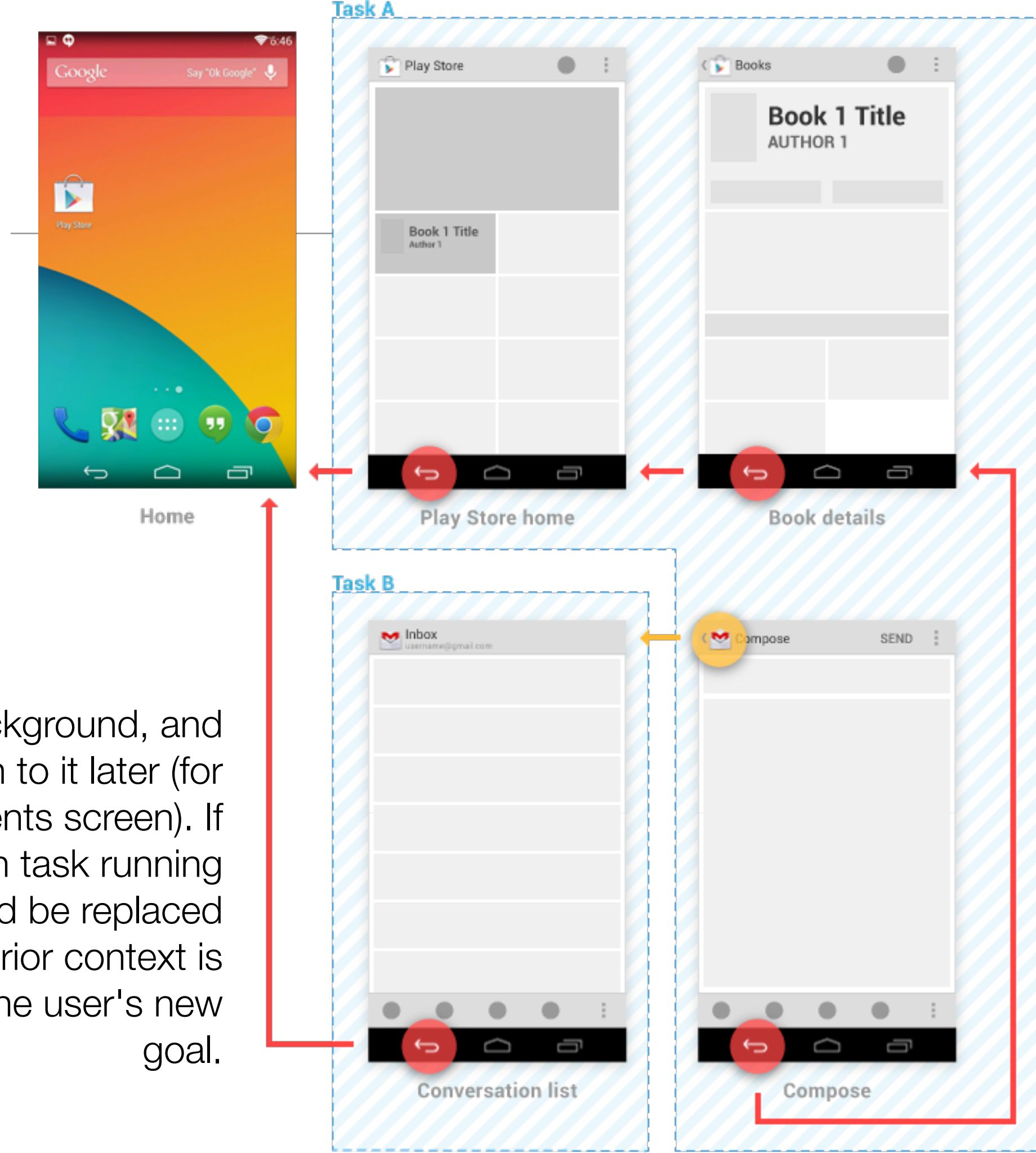
- When the user elects to share via Gmail, Gmail's compose activity is added as a continuation of Task A—no new task is created. If Gmail had its own task running in the background, it would be unaffected.



From the compose activity, sending the message or touching the Back button returns the user to the book details activity. Subsequent touches of Back continue to navigate back through the Play Store, ultimately arriving at Home.

- However, by touching Up from the compose activity, the user indicates a desire to remain within Gmail. Gmail's conversation list activity appears, and a new Task B is created for it. New tasks are always rooted to Home, so touching Back from the conversation list returns there.

Task A persists in the background, and the user may return to it later (for example, via the Recents screen). If Gmail already had its own task running in the background, it would be replaced with Task B—the prior context is abandoned in favour of the user's new goal.



(Material drawn from developer.android.com)