Q3 2011

android developer lab

android
developer lab

Designing UIs for Phones and Tablets

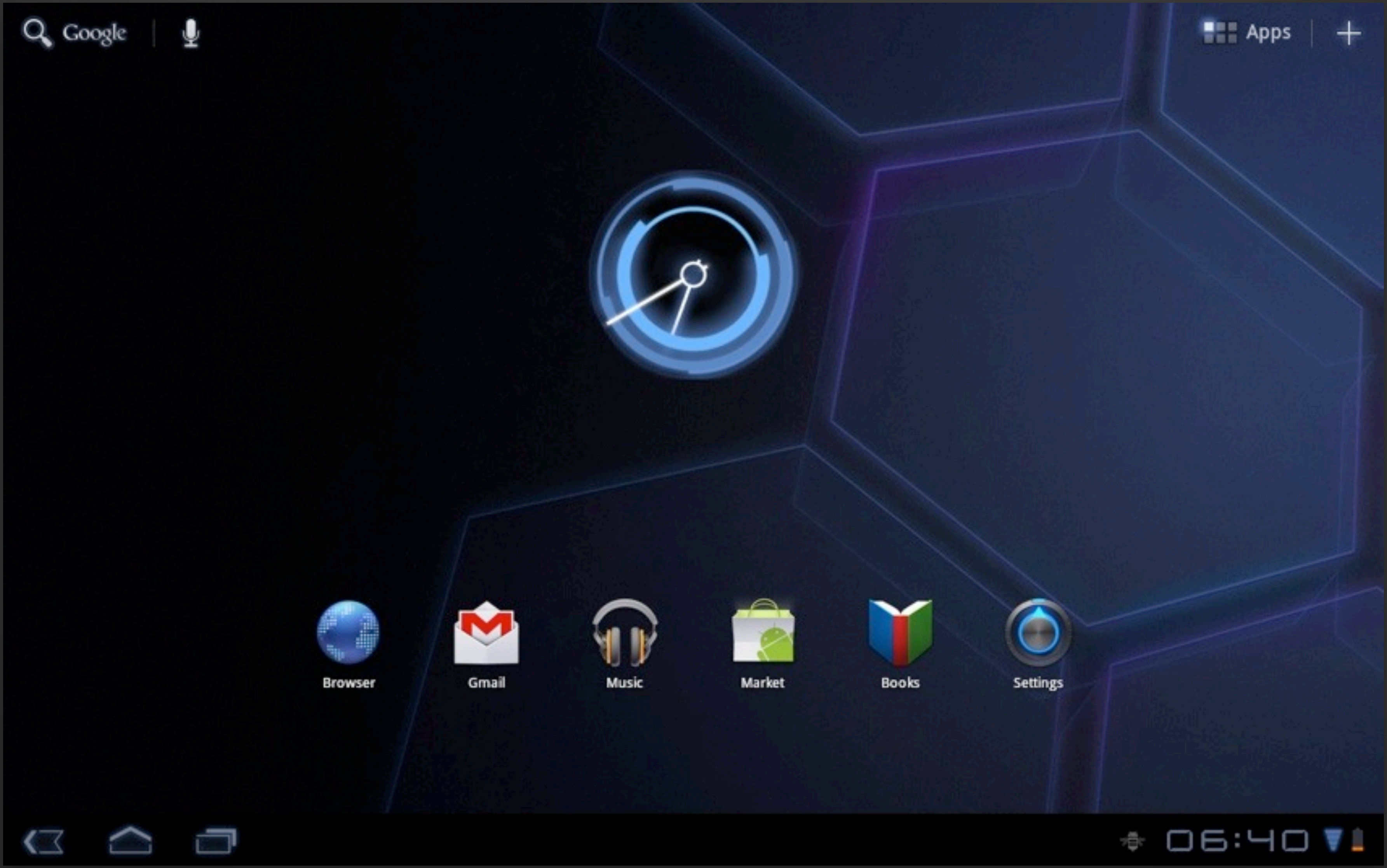
Q3 2011

Agenda

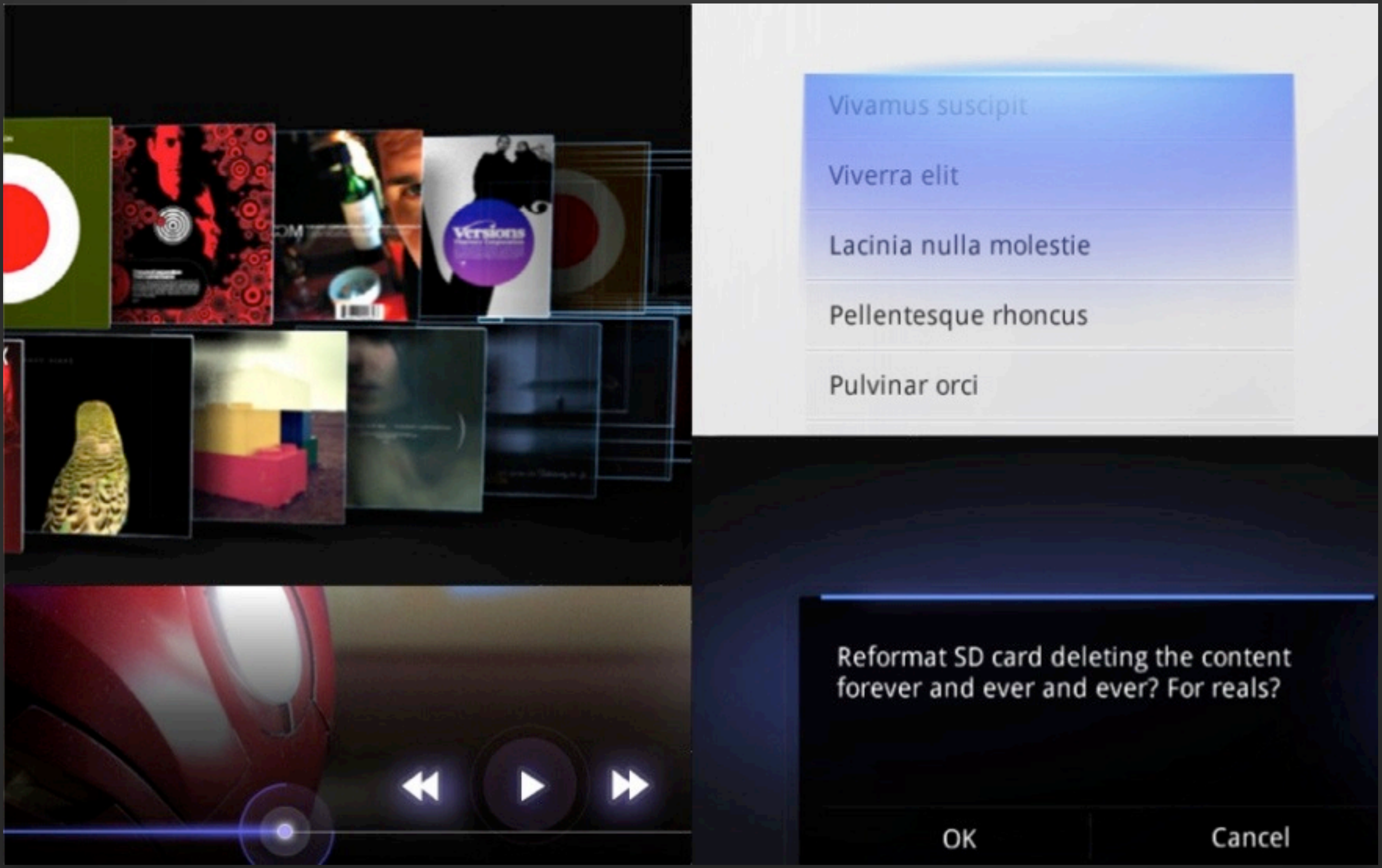
1. Honeycomb visual design
2. Tablet UI patterns + Honeycomb framework features
   - Interaction design
   - Implementation
3. Do’s and don’ts
4. Example — Google I/O 2011 App

Honeycomb visual design

Introducing the Holographic UI



Early style explorations



Vivamus suscipit

Viverra elit

Lacinia nulla molestie

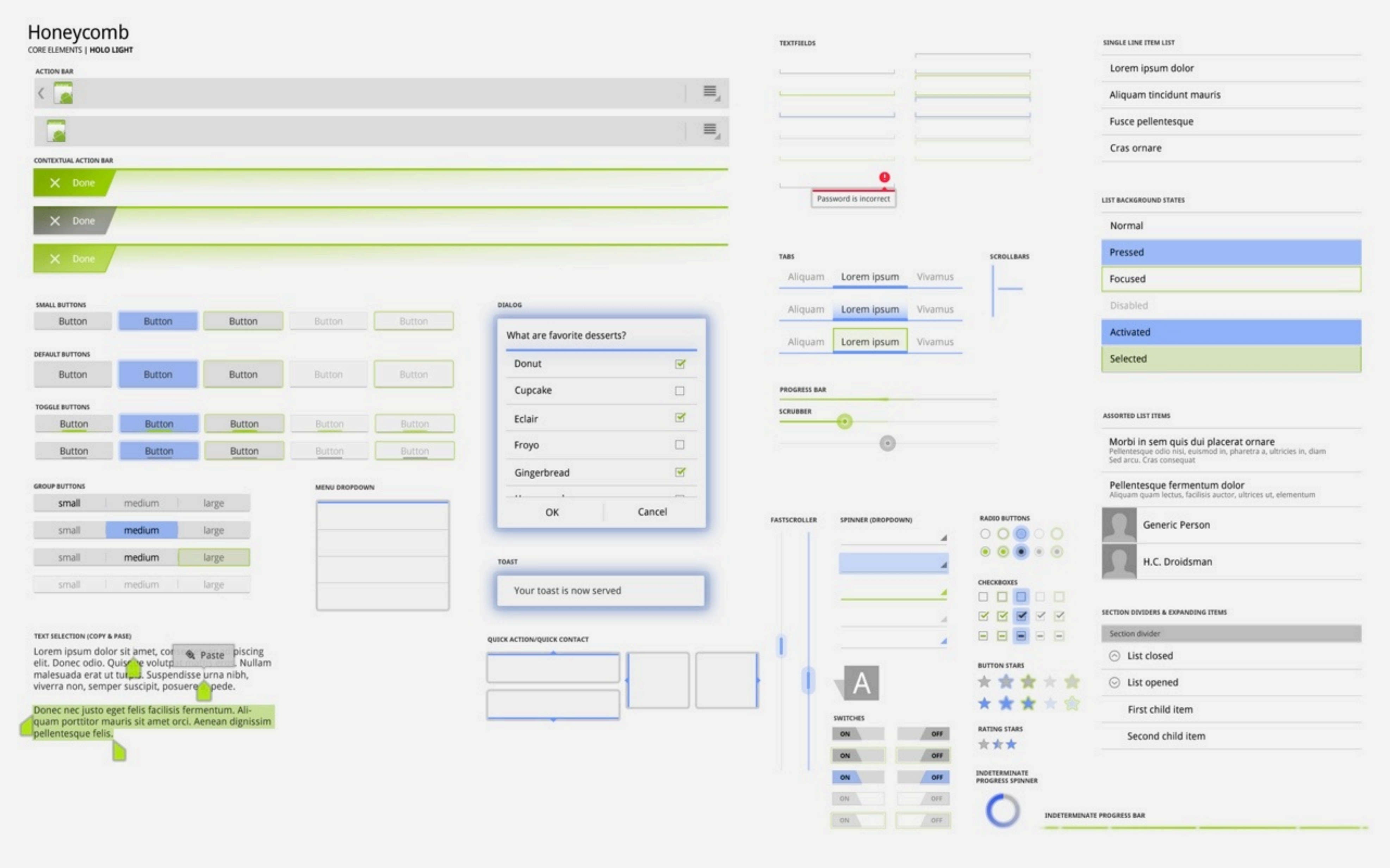
Pellentesque rhoncus

Pulvinar orci

Reformat SD card deleting the content forever and ever and ever? For reals?

Q3 2011

Widget styling – Theme.Holo.Light

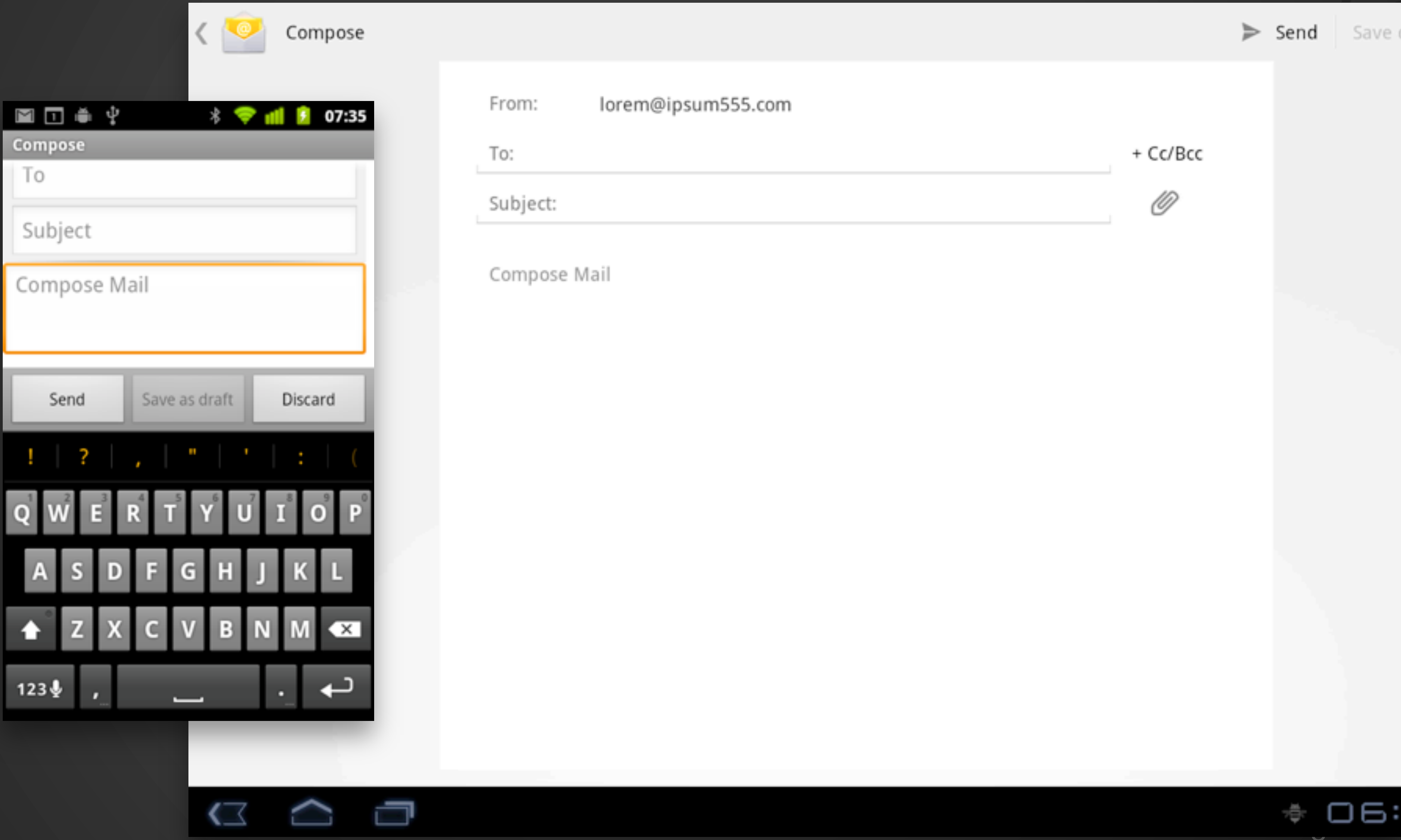


Lorem ipsum dolor sit amet, co piscing elit. Donec odio. Quis e volutp . Nullam malesuada erat ut tu . Suspendisse urna nibh, viverra non, semper suscipit, posuere pede.

Donec nec justo eget felis facilisis fermentum. Aliquam porttitor mauris sit amet orci. Aenean dignissim pellentesque felis.

Q3 2011

Simplify UI – Removing boxes

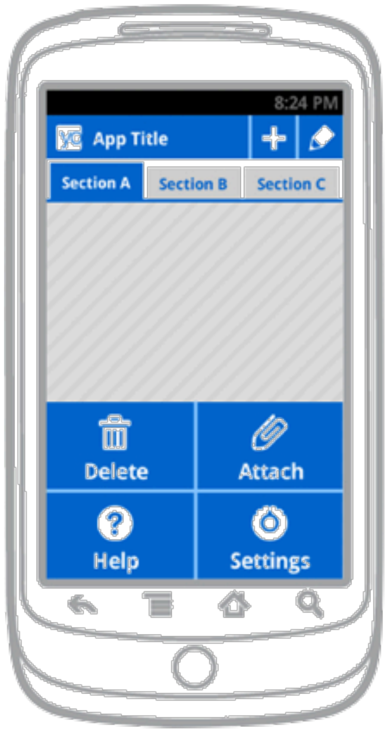


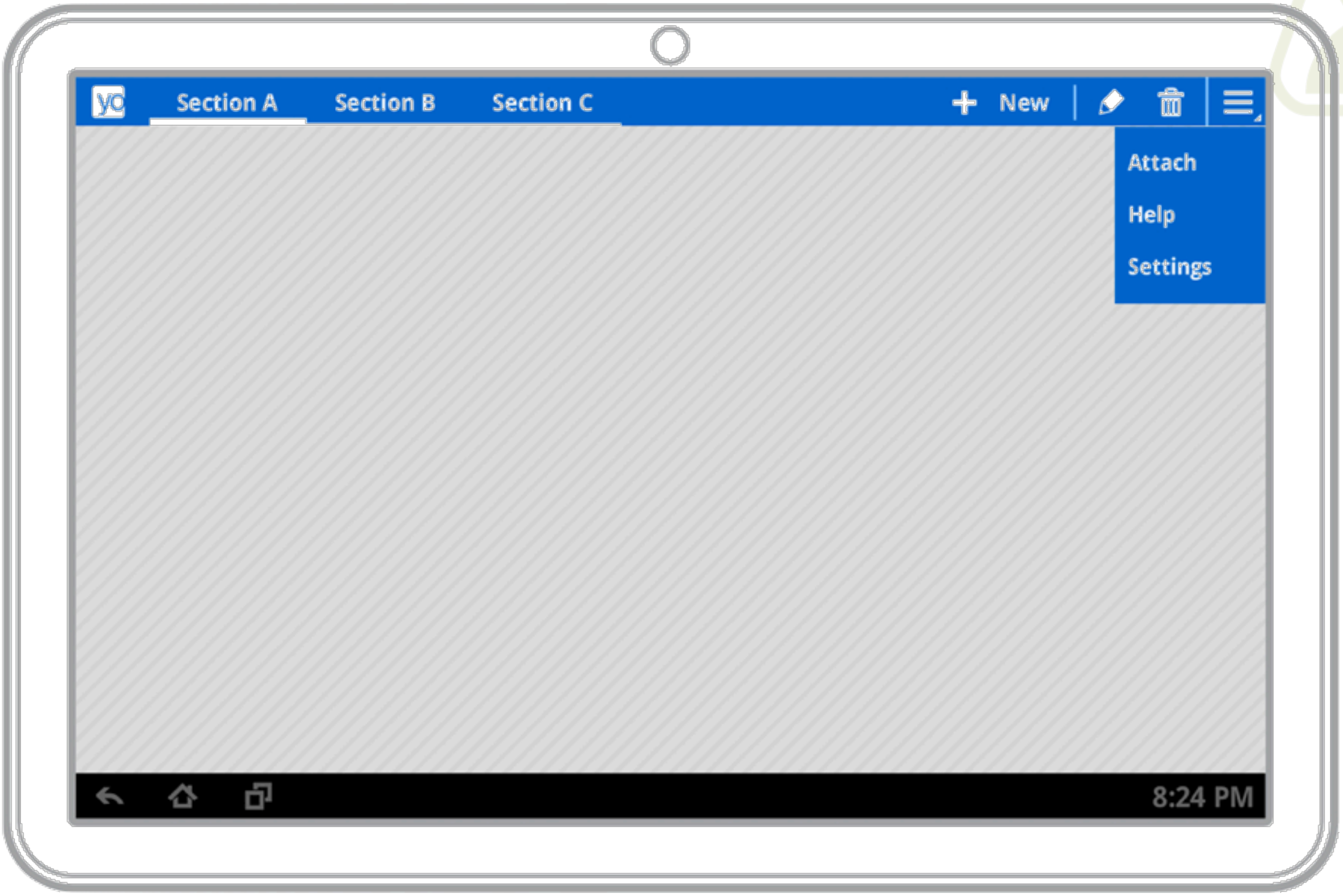
Honeycomb UI patterns and framework features

UI patterns

- General solutions to recurring problems
- Framework-supported
- Guidelines, not restrictions
- Topics we’ll discuss today:
    1. Action Bar
    2. Multi-pane Layouts
    3. App Navigation
    4. Beyond the List

Action Bar


8:24 PM
App Title
Section A
Section B
Section C
Delete
Attach
Help
Settings


Section A
Section B
Section C
New
Attach
Help
Settings
8:24 PM

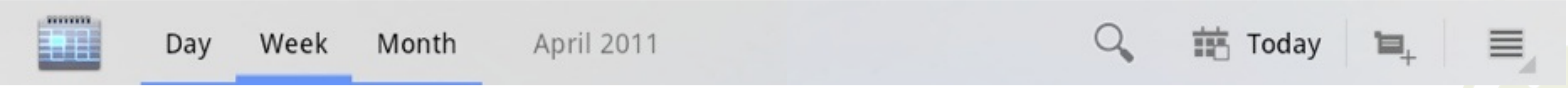
Action Bar – Introduction

- Not a new pattern
  - Presented as phone UI pattern at Google I/O 2010
  - Used in many apps through Android Market
  - Honeycomb has greatly extended its usefulness
- Dedicated real estate at the top of each screen
  - Generally persistent throughout application

Action Bar – Introduction

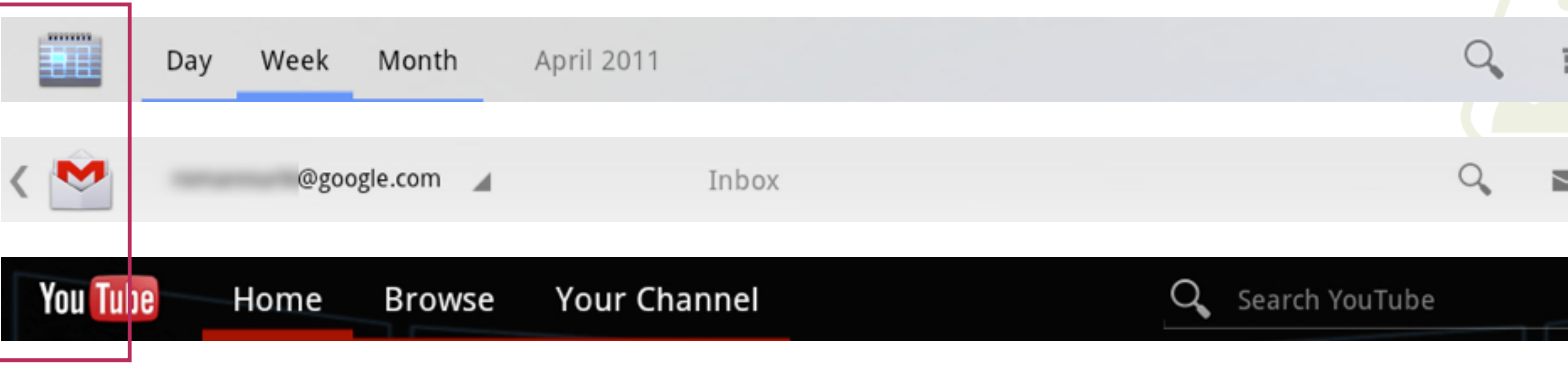
- Used to make frequently used actions prominent
- Supports navigation, give users a sense of place
- Convenient means of handling **Menu** and **Search**

Action Bar – General organization



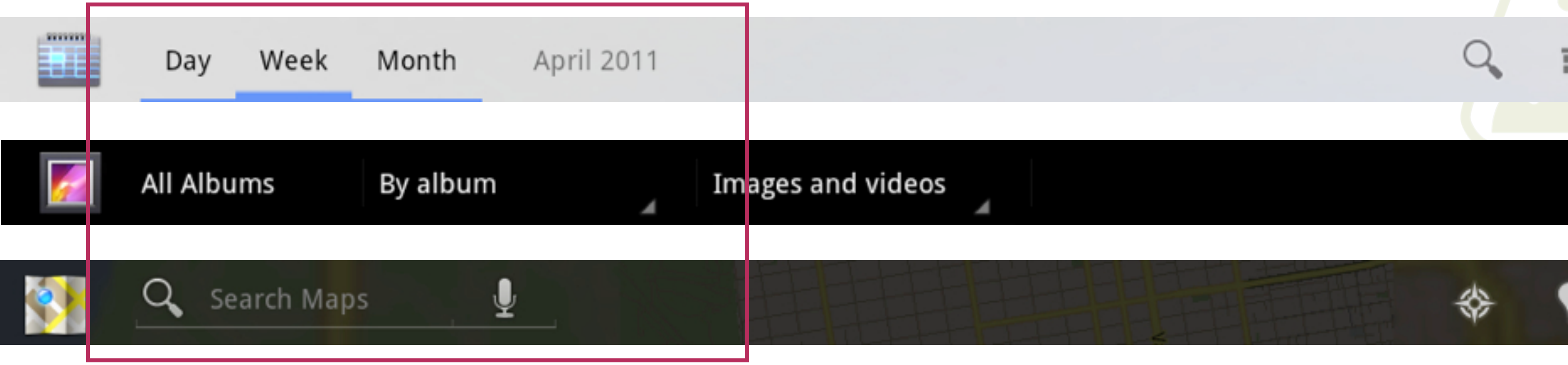
- App icon — where am I?
- View details — what can I see?
- Action buttons — what can I do here?

Action Bar – General organization



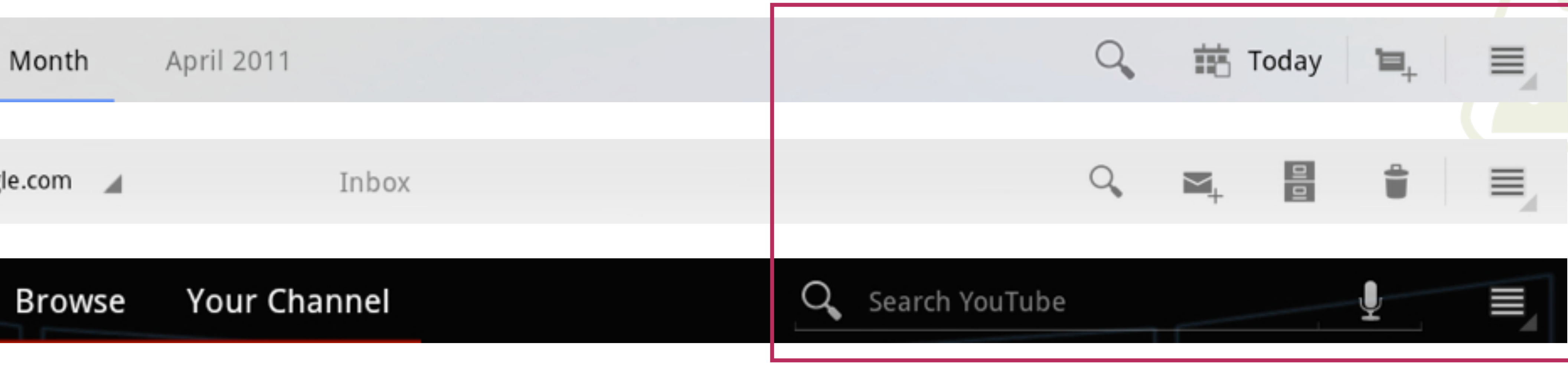
- App icon
  - Can be replaced with logo or other branding
  - Give the user a sense of place
  - Used to support “upward” navigation within the app

Action Bar – General organization



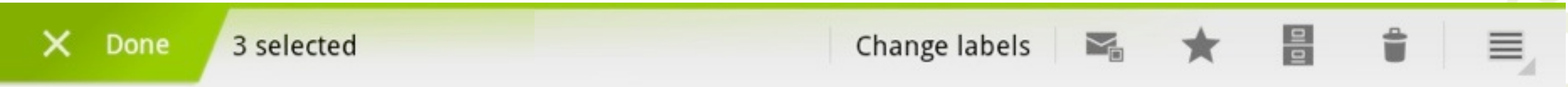
- View details
  - Simple: non-interactive title bar replacement
  - Richer: Tabs, drop-down menus, breadcrumbs

Action Bar – General organization



- Action buttons
  - More important / frequently-accessed action at left
  - Buttons can be icon-only, text-only, or icon-and-text
  - Overflow menu

Action Bar – Contextual actions



- Action bar can transform its appearance when items are selected
  - Useful for single or multiple selection
  - Typically invoking via *touch and hold*
- Like normal action bar, three sections:
  - *Done* button (for releasing selection)
  - Selection details
  - Contextual action buttons
- Implemented using `ActionMode`

Action Bar – Implementation

- Basic action bar
  - `Theme.Holo` (default if `targetSdkVersion` ≥ 11)
  - Action items from `res/menu/` with `showAsAction`

- Customizing the action bar
  - `ActionBar` class

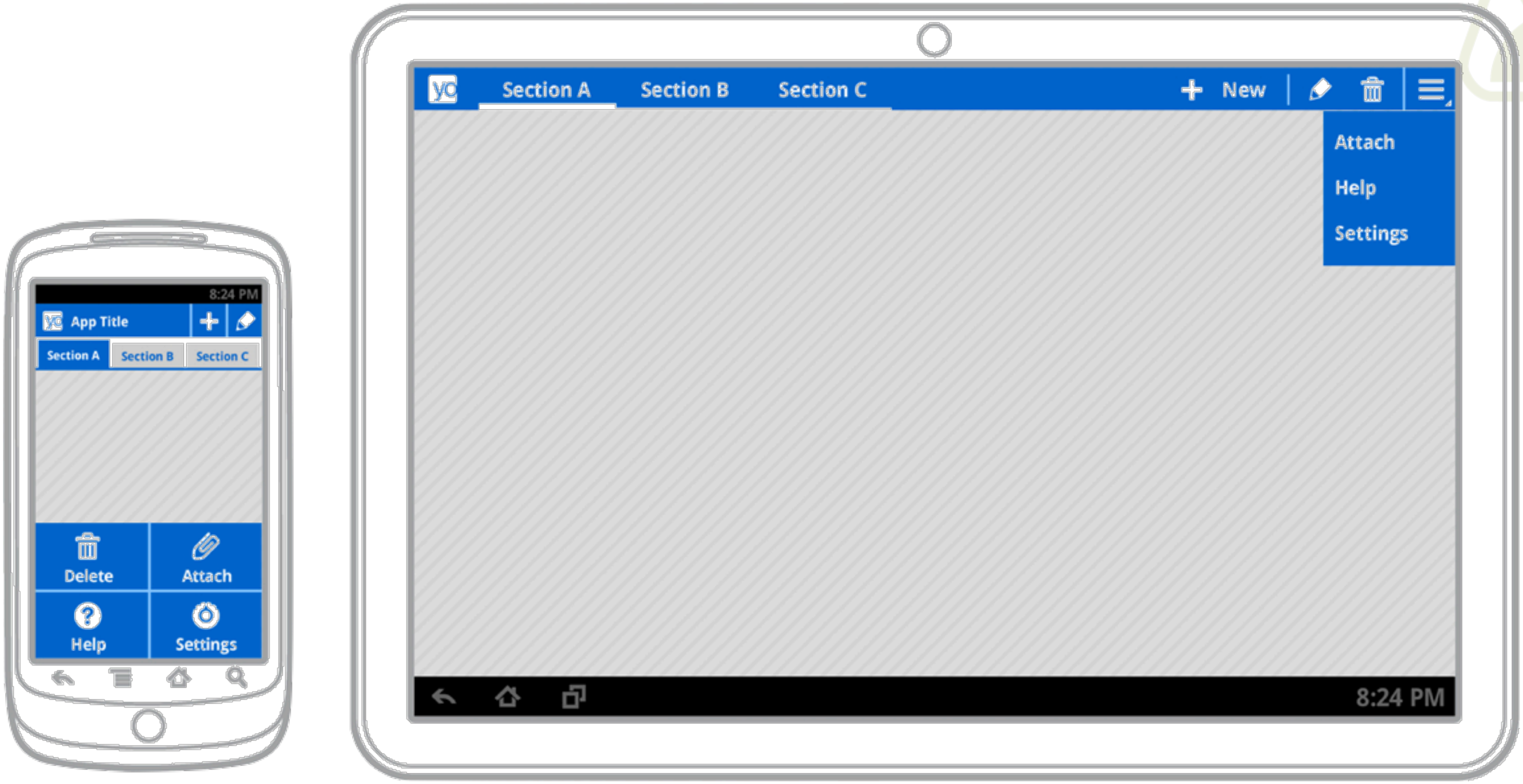
j.mp/customizing-action-bar

Action Bar – Compatibility

1. Write a custom action bar implementation pre-Honeycomb
   - iosched
   - GreenDroid
   - ActionBarSherlock

2. Alternatively, defer to the standard Options menu

Action Bar – Phones and tablets



Multi-pane Layouts – Introduction

- Take advantage of vastly increased real estate
  - Avoids excessively long line lengths
- Consolidate multiple related phone screens into a single compound view
- Provide more context (e.g. Settings)

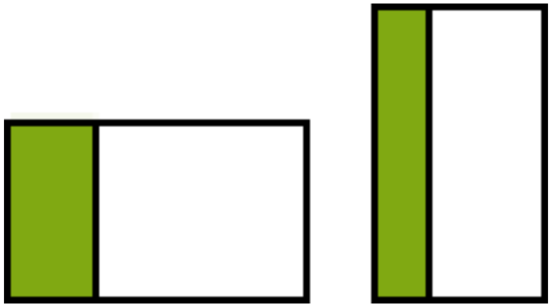
Multi-pane Layouts – Tips

- Panes to the right should generally present more content or details for items selected in the panes on the left.

Multi-pane Layouts – Implementation

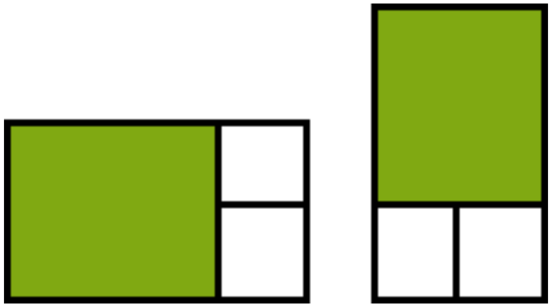
- `Fragment` class
- Optionally use the `<fragment>` tag in layout XML
- But **fragments are a lifecycle construct**, not necessarily a visual construct

Multi-pane Layouts – Orientation change



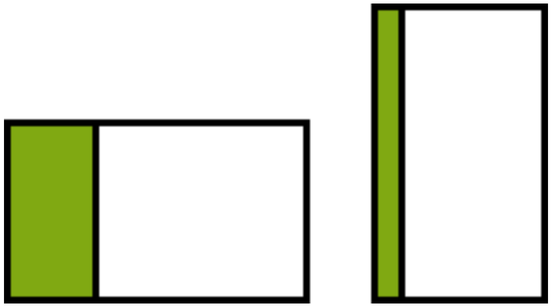
Stretch

(e.g. Settings)



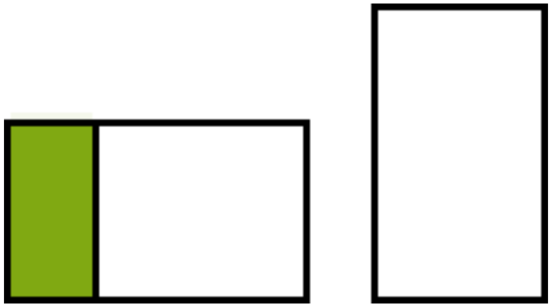
Stack

(e.g. Calendar)



Expand/collapse

(e.g. Google Talk)



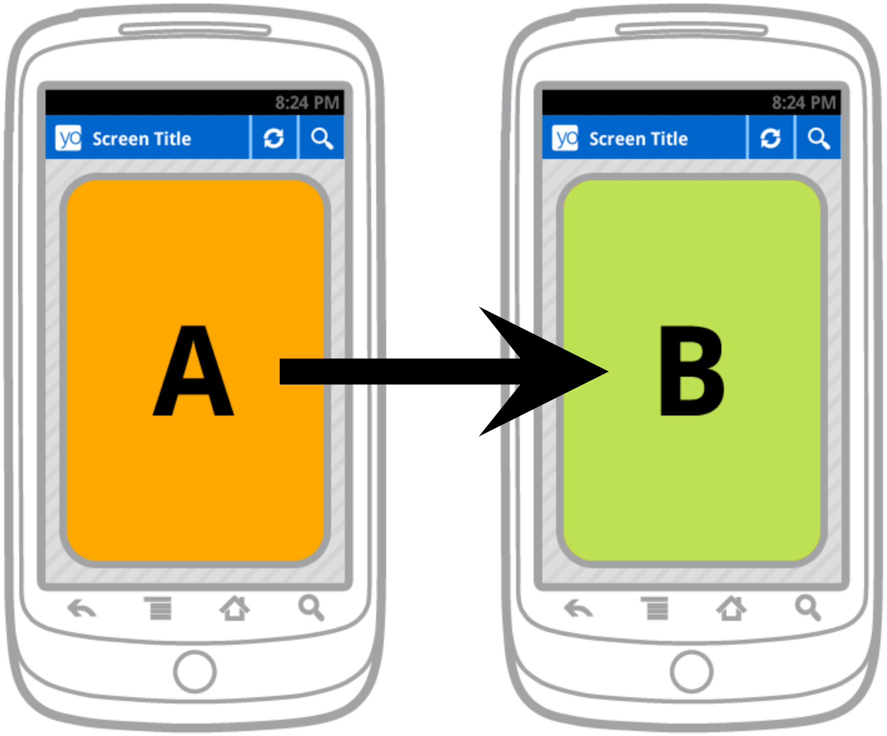
Show/hide

(e.g. Gmail)

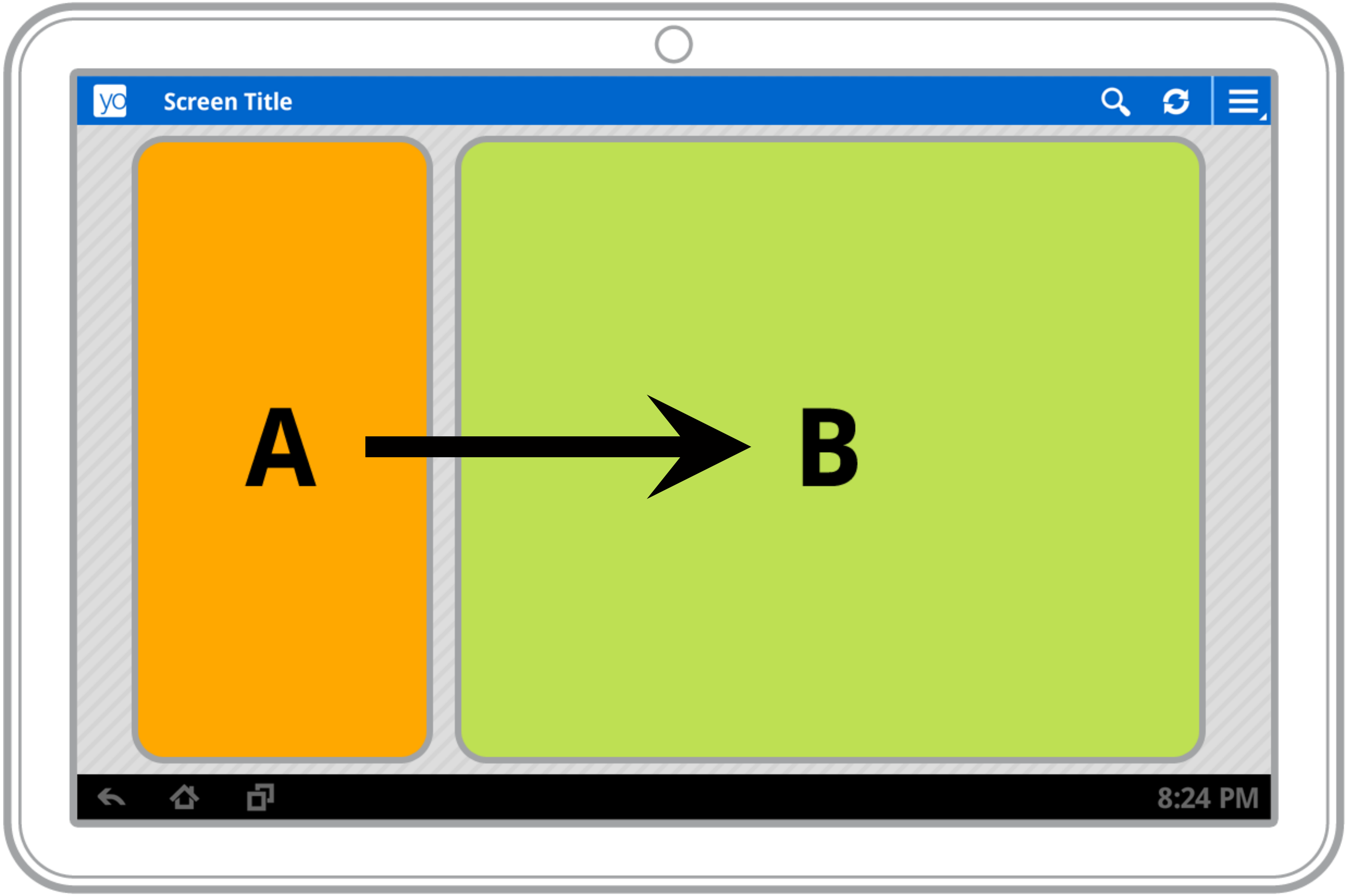
Multi-pane Layouts – Orientation change

- Orientation changes should **preserve functional parity**
  - User shouldn’t have to rotate device to achieve a task
- Strategies apply per-screen, not per app
- For the **show/hide** orientation strategy, use **UP** navigation to show the master pane
  - e.g. Gmail conversation view

Multi-pane Layouts – Intents



Multi-pane Layouts – Intents



Multi-pane Layouts – Intents

- If implementing **A** → **B** with multiple activities, need a strategy to “connect” fragments
  - Activity 1 (Phone, A)
  - Activity 2 (Phone, B)
  - Activity 3 (Tablet, A & B)

Strategies for “connecting” fragments

1. Phone + tablet activities implement a common interface
2. Fragments hold references to each other, or use `setTargetFragment`
   - Defer to startActivity if no target fragment

…

Strategies for “connecting” fragments

…

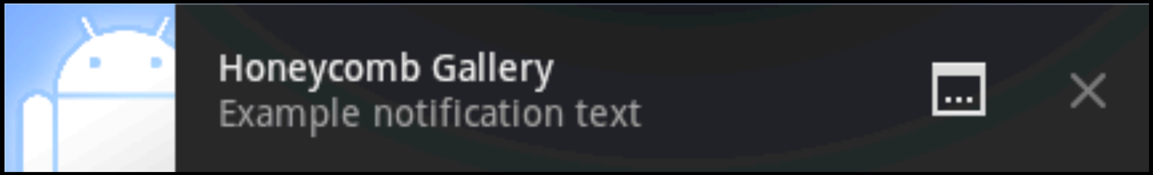
3. Fragments call `startActivity`, tablet activity intercepts/overrides it

4. Fragments call `startActivity`, tablet activity is `singleTask` (or `singleTop`) + routes intent to correct fragment in `onNewIntent`.

App Navigation – Introduction

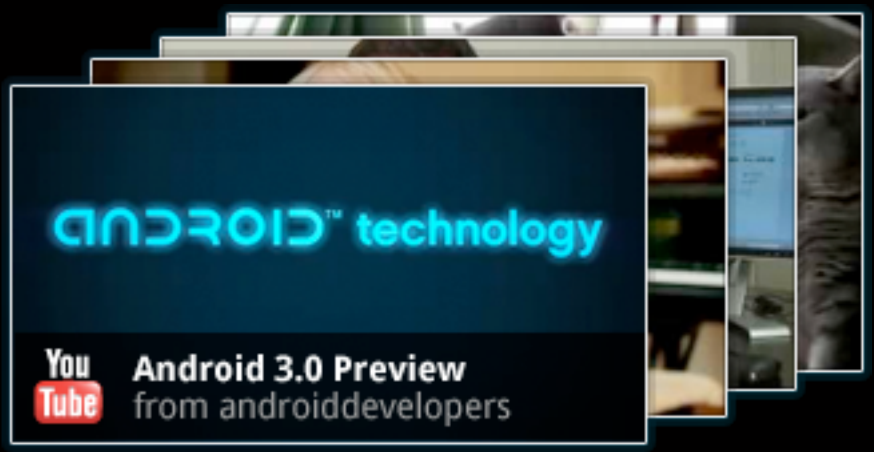
- One of the more dramatic changes in Honeycomb
- Increased variety of mechanisms for direct, deep navigation into an app

App Navigation – Highlights



Honeycomb Gallery
Example notification text

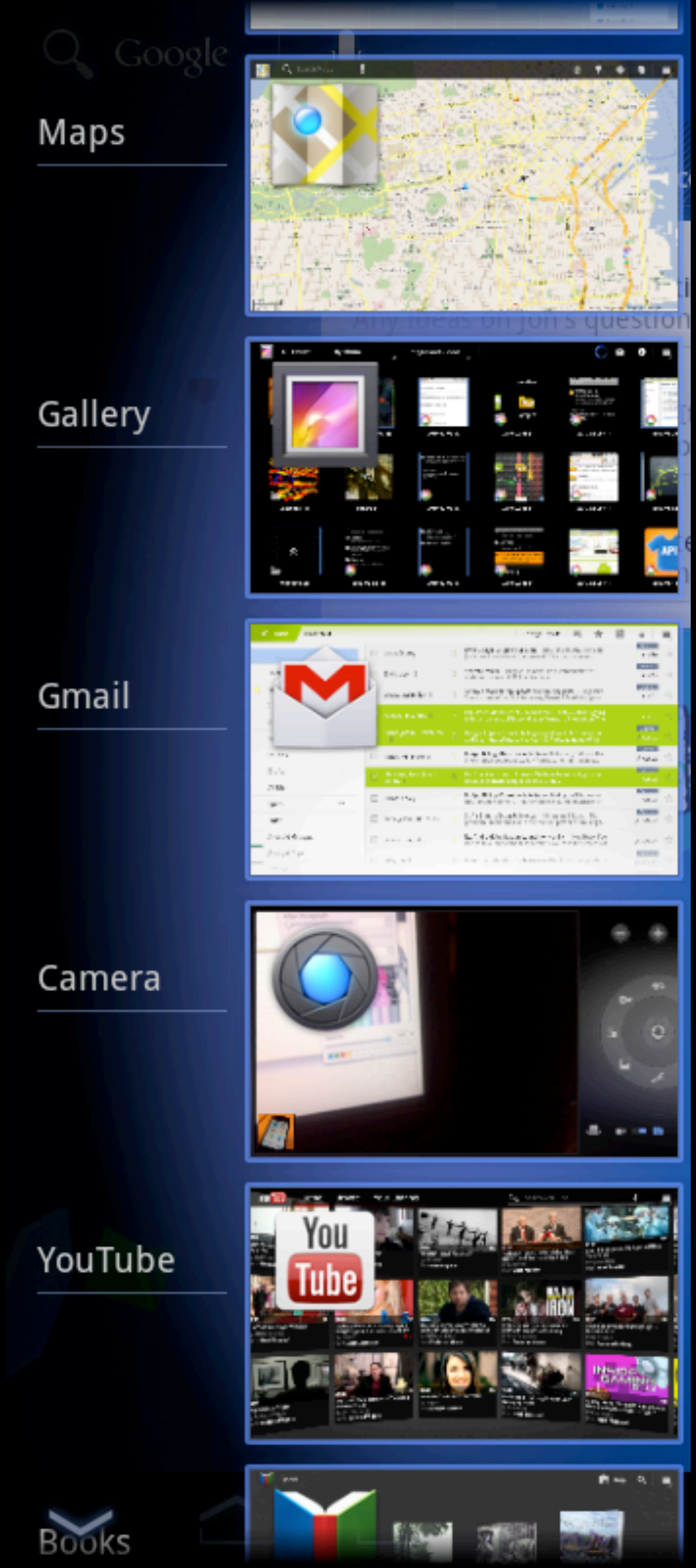
Richer notifications



android™ technology

Android 3.0 Preview
from androiddevelopers

Richer home screen widgets



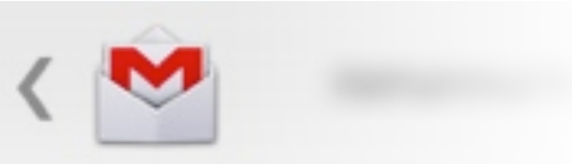
'Recents'

Navigation and user memory

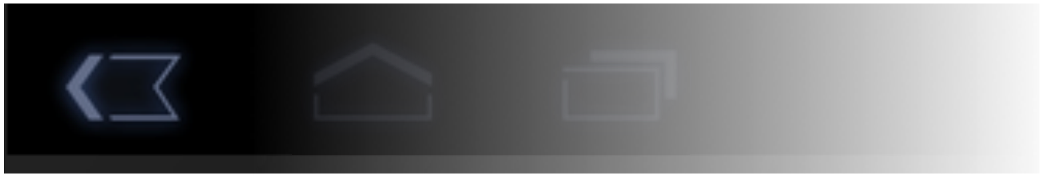
- Android has traditionally relied on **temporal** memory:
  - We’re good at remembering what *just* happened
  - Not so good with order of events from a while ago
  - Potential for error, surprise
- Users have strong **structural** memory
  - Good at relationships between screens in an app
  - Used to going “Home” in web apps
  - Clearer expectations for behavior

Back versus Up

- **APPLICATION UP**
  navigates hierarchy within a single app



- **SYSTEM BACK**
  navigates history between related screens



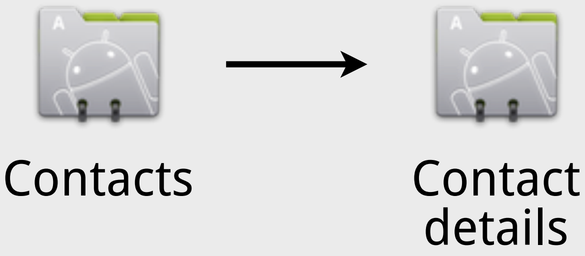
Example Flows

Contacts Task

Contacts

Example Flows

Contacts Task

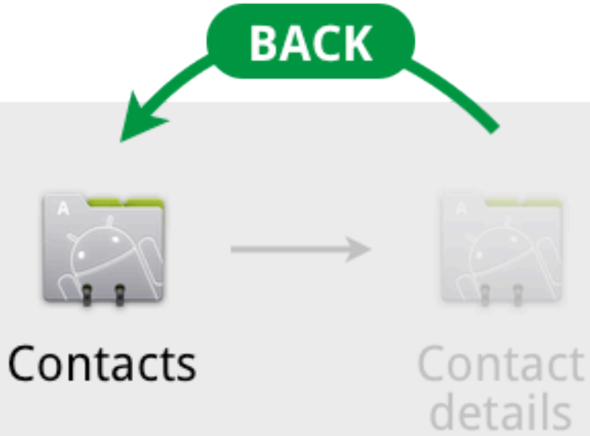


Contacts

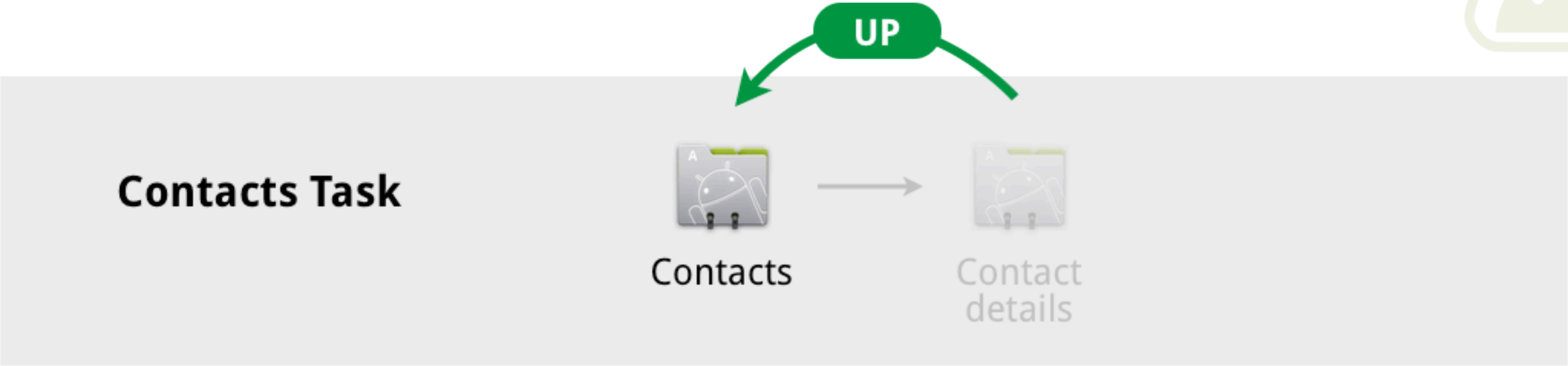
Contact details

Example Flows

Contacts Task



Example Flows



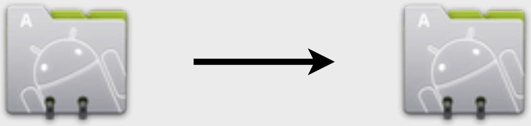
Example Flows

Contacts Task

Contacts

Example Flows

Contacts Task

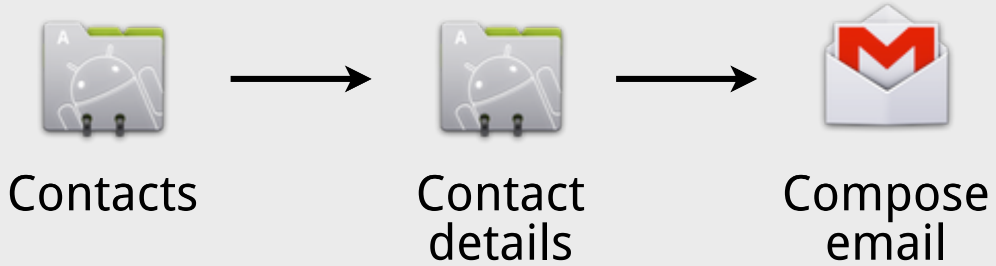


Contacts

Contact details

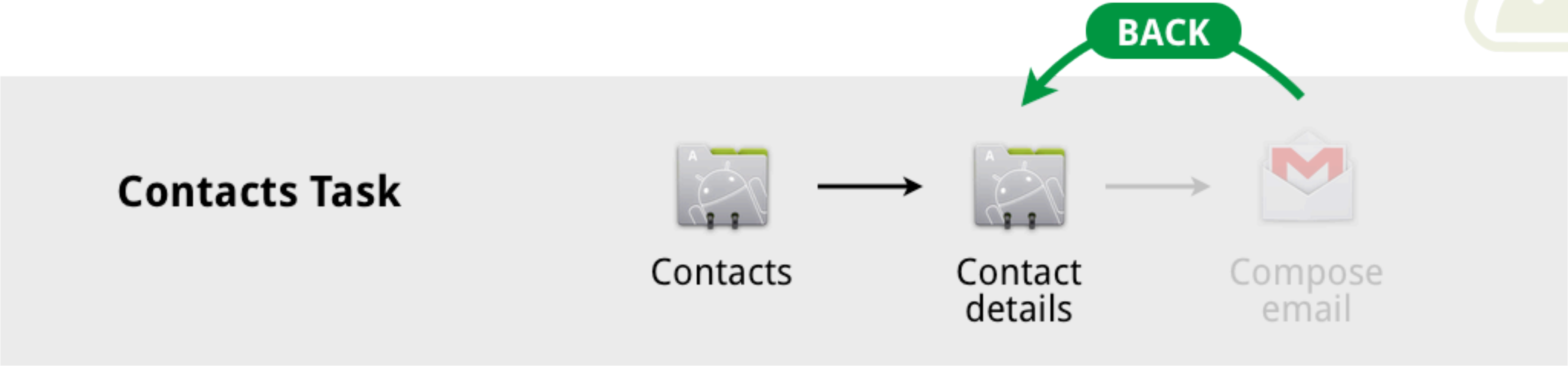
Example Flows

Contacts Task



Contacts → Contact details → Compose email

Example Flows



Contacts Task

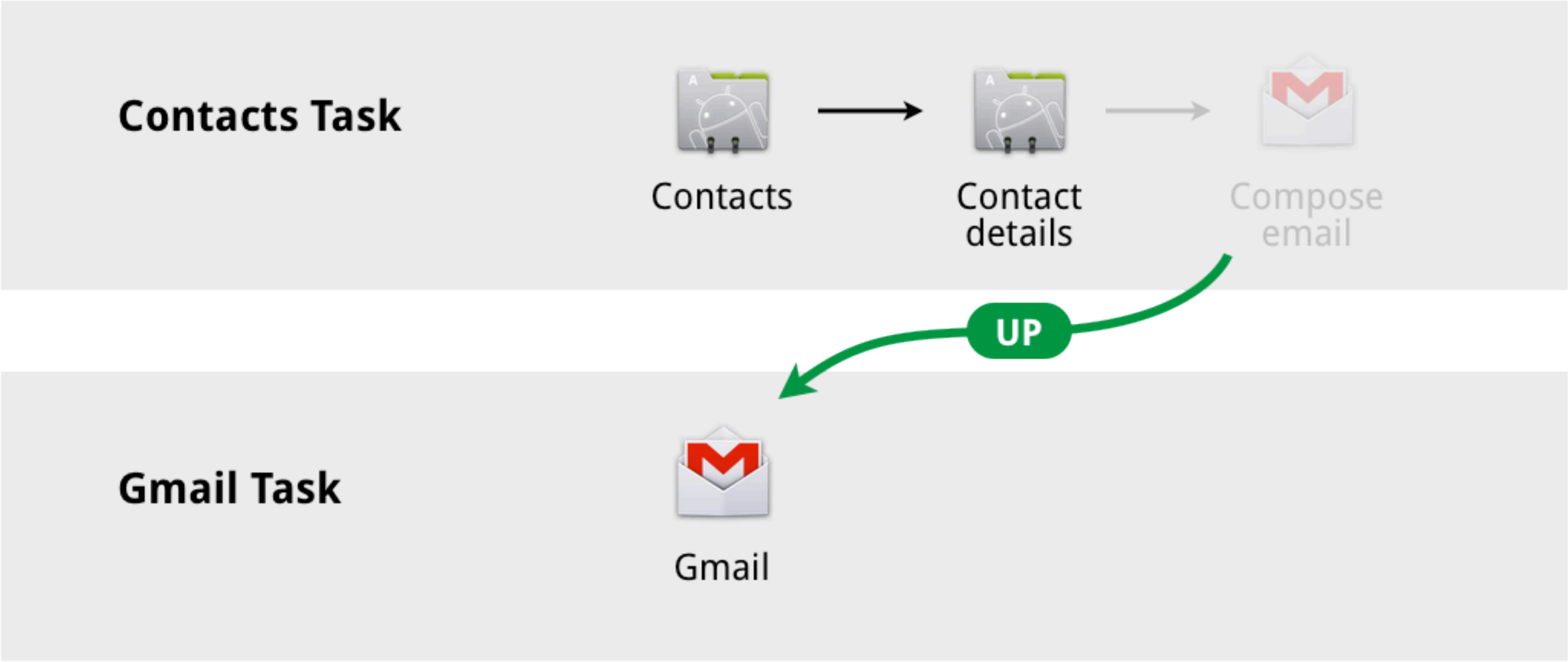
Contacts

Contact details

Compose email

BACK

Example Flows



Contacts Task

Contacts

Contact details

Compose email

UP

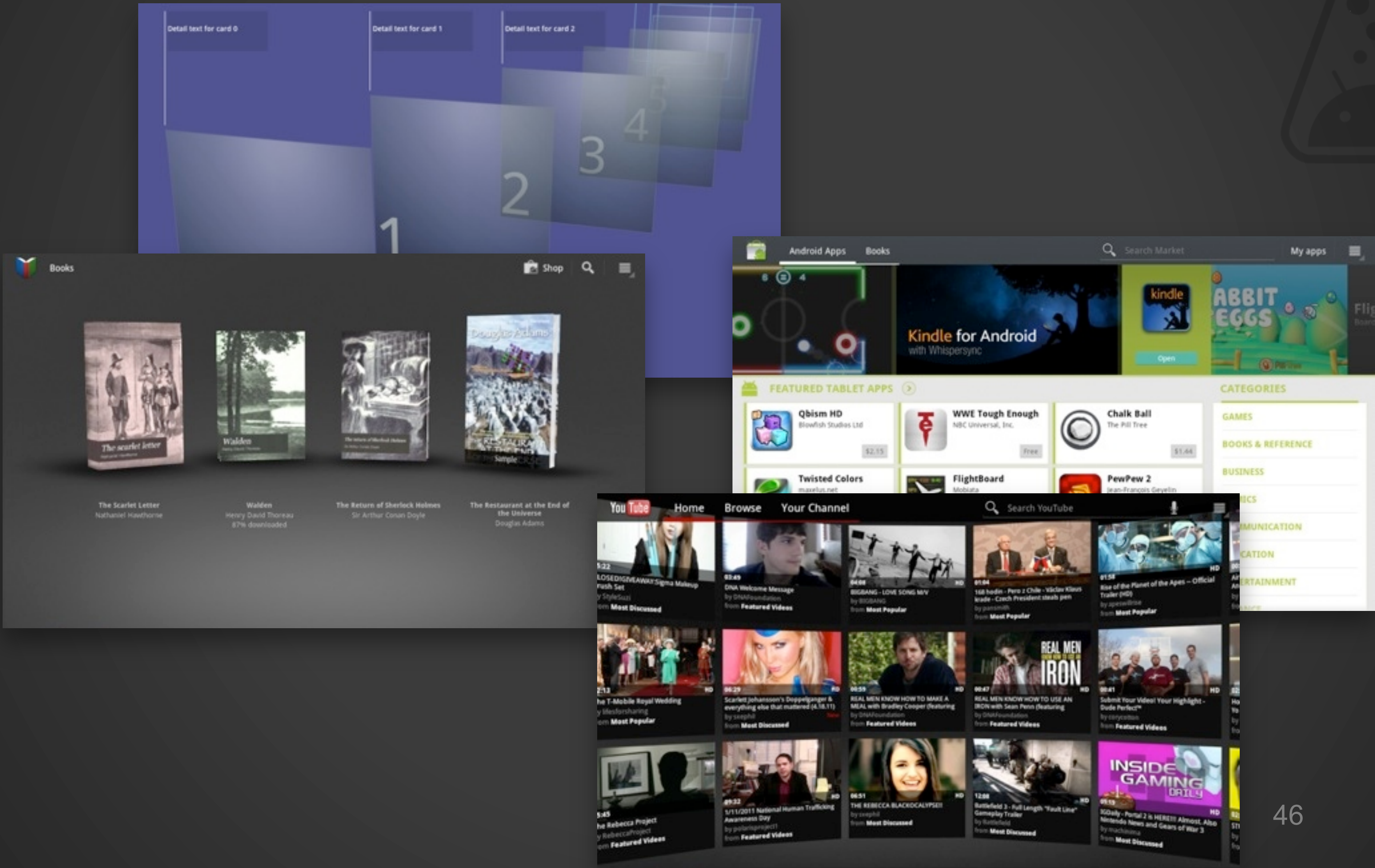
Gmail Task

Gmail

Beyond the List – Introduction

- Views for media-rich applications
- **“Hero moments”** to break the monotony of list views
- Encourage more engaged exploration, akin to flipping through a magazine

Beyond the List – Examples



Beyond the List – Implementation

- `CarouselView` (3D)
  - Renderscript
  - Intended for customization

j.mp/io2011-carousel-sample

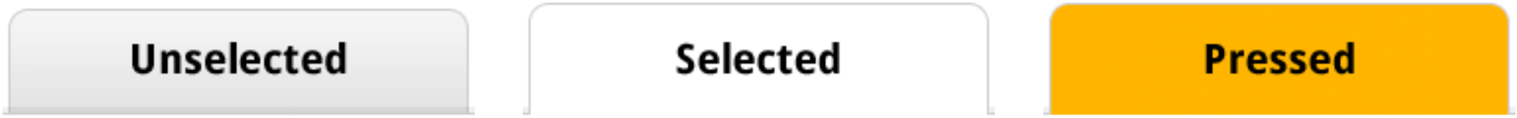
- `ViewPager` (2D) for showing one item or page at a time

Do’s and don’ts

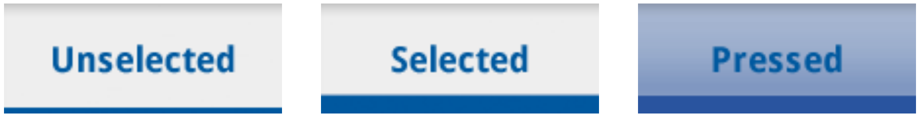
- **DO** aim for a single APK
- **DO** use the compatibility library
- **DO** customize visual design completely, if straying from Holo theme
- **DO** support both landscape and portrait
- **DO** extract dimensions for phones and tablets
  - `values/dimens.xml`
  - `values-large/dimens.xml`
- **DO** use theme/style/ etc. to reduce redundancy

DO marry OS visual style with your brand/identity

- `drawable-hdpi`



- `drawable-large-mdpi-v11`



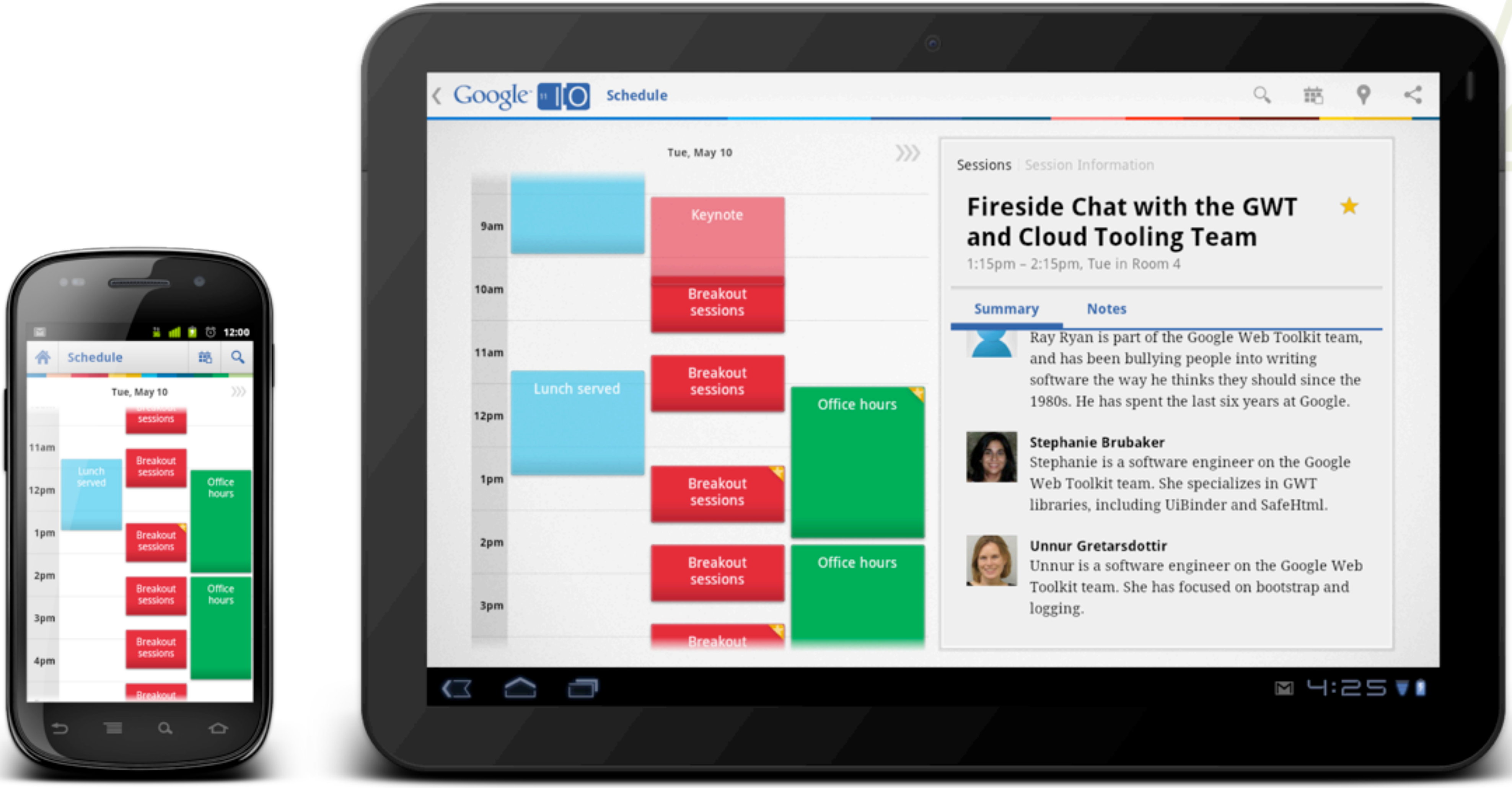
- **DON’T** assume API level ≥11 == tablet
- **DON’T** assume xlarge == tablet
  - 7" tablet is large
- **DON’T** use small font sizes
- **DON’T** overuse `fill_parent`; avoid excessively long lines of text

DON’T think that tablets are just big phones

Tablets fulfill a very different need.

Example:
Google I/O 2011 App

Q3 2011



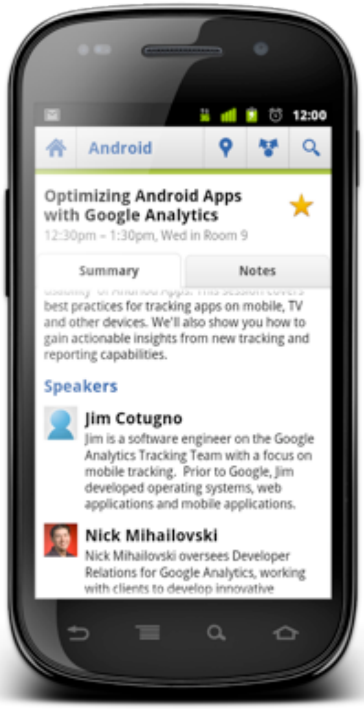
Fireside Chat with the GWT and Cloud Tooling Team

1:15pm – 2:15pm, Tue in Room 4

Ray Ryan is part of the Google Web Toolkit team, and has been bullying people into writing software the way he thinks they should since the 1980s. He has spent the last six years at Google.

Stephanie Brubaker
Stephanie is a software engineer on the Google Web Toolkit team. She specializes in GWT libraries, including UiBinder and SafeHtml.

Unnur Gretarsdottir
Unnur is a software engineer on the Google Web Toolkit team. She has focused on bootstrap and logging.



12:00

Android

Optimizing Android Apps with Google Analytics ★

12:30pm – 1:30pm, Wed in Room 9

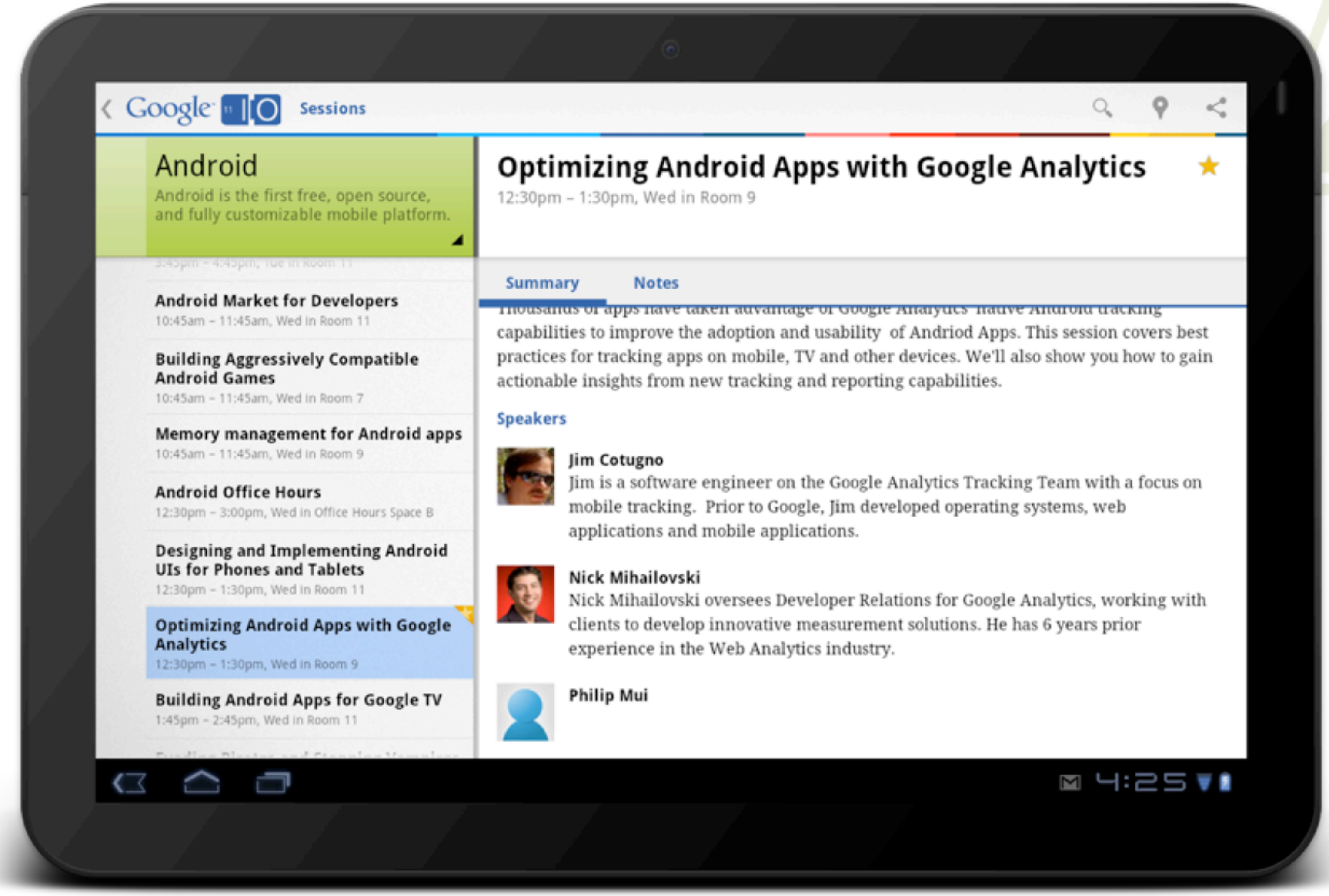
Summary | Notes

best practices for tracking apps on mobile, TV and other devices. We'll also show you how to gain actionable insights from new tracking and reporting capabilities.

Speakers

Jim Cotugno
Jim is a software engineer on the Google Analytics Tracking Team with a focus on mobile tracking. Prior to Google, Jim developed operating systems, web applications and mobile applications.

Nick Mihailovski
Nick Mihailovski oversees Developer Relations for Google Analytics, working with clients to develop innovative



Android

Android is the first free, open source, and fully customizable mobile platform.

3:45pm – 4:45pm, Tue in Room 11

Android Market for Developers
10:45am – 11:45am, Wed in Room 11

Building Aggressively Compatible Android Games
10:45am – 11:45am, Wed in Room 7

Memory management for Android apps
10:45am – 11:45am, Wed in Room 9

Android Office Hours
12:30pm – 3:00pm, Wed in Office Hours Space B

Designing and Implementing Android UIs for Phones and Tablets
12:30pm – 1:30pm, Wed in Room 11

Optimizing Android Apps with Google Analytics
12:30pm – 1:30pm, Wed in Room 9

Building Android Apps for Google TV
1:45pm – 2:45pm, Wed in Room 11

Evading Pirates and Stopping Vampires

Optimizing Android Apps with Google Analytics ★

12:30pm – 1:30pm, Wed in Room 9

Summary | Notes

Thousands of apps have taken advantage of Google Analytics' native Android tracking capabilities to improve the adoption and usability of Andriod Apps. This session covers best practices for tracking apps on mobile, TV and other devices. We'll also show you how to gain actionable insights from new tracking and reporting capabilities.

Speakers

Jim Cotugno
Jim is a software engineer on the Google Analytics Tracking Team with a focus on mobile tracking. Prior to Google, Jim developed operating systems, web applications and mobile applications.

Nick Mihailovski
Nick Mihailovski oversees Developer Relations for Google Analytics, working with clients to develop innovative measurement solutions. He has 6 years prior experience in the Web Analytics industry.

Philip Mui

4:25

Get the code at

code.google.com/p/iosched

For more, visit

developer.android.com

Copyrights and trademarks

- Android, Google are registered trademarks of Google Inc.
- All other trademarks and copyrights are the property of their respective owners.