

# The Android Dual-Screen Landscape



# Getting To a TV

- TV as Android Primary Display
  - Amazon Fire TV
  - OUYA
  - Google TV
  - Android HDMI sticks
- Dual-Screen Models
  - TV as Android External Display
  - Remote Playback
    - Chromecast



# TV as Primary Display

- Good News: Standard Android Programming
  - Normal activities, fragments, widgets
- Issues
  - User input (no touchscreen)
  - 10-foot UI design
  - Overscan
  - Size / density
  - Color



# TV as External Display

- Connection Types
  - HDMI (full, micro)
  - MHL / MHL3
  - SlimPort
  - Miracast / AllShare Cast
  - Proprietary



# TV as External Display

- Technique #1: Presentation
  - Subclass of Dialog
  - Tie to a Display via DisplayManager or MediaRouter
    - PresentationHelper from CWAC-Presentation library makes this a bit easier to manage
  - Contents of “dialog” displayed on external display
  - Key limitation: must be shown by an Activity



# TV as External Display

- Technique #2: PresentationService
  - Subclass of Service
  - Tie to a Display via DisplayManager or MediaRouter
  - View returned from `buildPresoView()` displayed on external display
  - Key limitation: third-party library from some balding guy



# TV as External Display

- Conditional External Display
  - Wrap Presentation in a PresentationFragment
    - Subclass of DialogFragment from CWAC-Presentation library
  - show( ) the PresentationFragment on external display when available
  - Use as a regular Fragment in your main activity UI when no external display is available



# TV as External Display

- Issues
  - User input via phone or tablet
  - 10-foot UI
  - Size / density (use the dedicated Context)
  - Color





# TV as External Display

- Dual-Screen App Challenge
  - <http://siliconimage.challengepost.com/>
  - Offered by Silicon Image
  - \$100K in prizes
  - Deadline = July 21
  - Judges



# Remote Playback

- `MediaRouter`
  - Way to find out about speakers, displays, and remote playback devices like Chromecast
- `MediaRouteActionProvider`
  - The “cast button” that goes in your action bar to allow the user to choose a media route for your app to use
  - Google's requires `AppCompat`; cross-port works with native action bar



# Remote Playback

- RemotePlaybackClient
  - Interface to tell compatible devices what media to play and how to control it (pause, resume, etc.)
- Google Cast SDK
  - Interface to tell compatible devices what media to play and how to control it (pause, resume, etc.)
  - Offers ability to work with custom HTML “receiver”
  - Only works with Chromecast
  - Proprietary with icky license agreement



# Remote Playback

- **MediaRouteProvider**
  - Bridge code between RemotePlaybackClient and playback device/app
  - Represents the media route the user chooses
  - Can write own
    - To tie to your app running on stock Android hardware connected to the TV
    - To tie to other streaming media boxes that offer some sort of control API



# Remote Playback

- Issues
  - What if remote playback device is not there?
  - User input via phone or tablet
  - 10-foot UI



# Resources

- Official Documentation
  - Including undocumented sample code
- DevBytes Videos
- CWAC Libraries
  - Presentation
  - MediaRouter
- [The Busy Coder's Guide to Android Development](#)



# Webinars

- “MediaRouter and RemotePlaybackClient”
  - How to write a client app that can “cast” to a Chromecast, using standard Android SDK classes
- “Cast' To Your Own Device”
  - How to write a MediaRouteProvider to connect RemotePlaybackClient-using apps to your app or device

