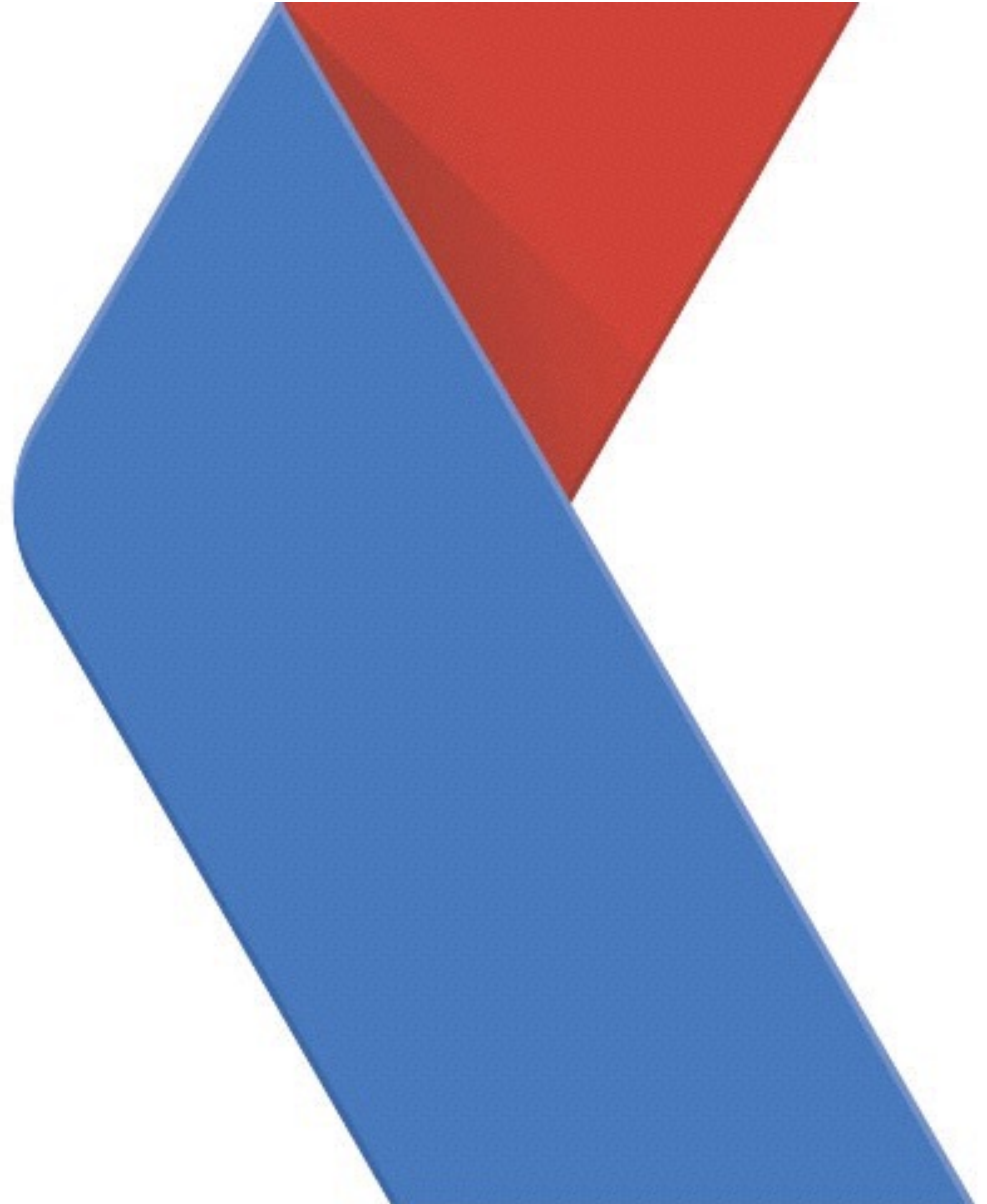


What's New in Android

December 2016





Developer Features (7.0)

Runtime

- Faster interpreter
- JIT
 - Faster install times
 - Less space consumed on device
- Apps use partial AOT
 - Only for hotspots



New Runtime Libraries

- ICU4J
 - Unicode and globalization support
 - Subset exposed in `android.icu`
 - Avoids conflicts with `com.ibm.icu`
 - Deprecated and unstable APIs omitted
- `FunctionalInterface`
- `java.util.function`
- `java.util.stream`

Java 8 Language Features

- Requires the Jack Compiler

```
android {  
    ...  
    defaultConfig {  
        ...  
        jackOptions {  
            enabled true  
        }  
    }  
    compileOptions {  
        sourceCompatibility JavaVersion.VERSION_1_8  
        targetCompatibility JavaVersion.VERSION_1_8  
    }  
}
```



Lambdas

- Implemented using Anonymous Class
- Compatible back to Gingerbread

```
Vertex[] v = new Vertex[100];  
// ...  
  
Arrays.sort(v, (Vertex v1, Vertex v2) -> {  
    return v1.length() - v2.length();  
});  
Arrays.sort(v, (v1, v2) -> v1.length() - v2.length());  
Arrays.sort(v, VertexFactory::compareVertices);
```



Default & Static Interface Methods

- Not Backwards Compatible

```
public interface Renderable {  
    void draw(Canvas c);  
  
    default bool isHardwareAccelerated() {  
        return false;  
    }  
  
    static int getMaxTextureSize() { return 2048; }  
}
```

```
public interface FrameListener {  
    default void onFrameStarted() { }  
    default void onFrameEnded() { }  
}
```



Repeating Annotations

- Not Backwards Compatible

```
@Repeatable(Exportables.class)
public @interface Exportable {
    String format() default "JSON";
}

public @interface Exportables {
    Exportable[] values();
}

@Exportable(format="JSON")
@Exportable(format="XML")
public class MyClass {
    // ...
}
```



Java Audio Latency

- Previous releases reduced native latency
- Lower latency **AudioTrack** (40-70ms reduction)

```
AudioAttributes attributes = new AudioAttributes.Builder()  
    .setFlags(AudioAttributes.FLAG_LOW_LATENCY)  
    .setContentType(AudioAttributes.CONTENT_TYPE_MUSIC)  
    .build();
```

- Dynamic resize of **AudioTrack** buffers
- Underruns query



RenderScript

- Single source
 - Many kernels in a single file
 - Launch kernels from kernels
- Reduction kernels
- Access multiple image buffers from video streams
- **Allocation.getBytesBuffer()** to reduce copies
- Enhanced **fp16** support

OpenGL ES 3.2

- GLES 3.1 + Android extension pack
 - Except EXT_texture_sRGB_decode
- Advanced blending equations
- Tessellation shaders
- Geometry shaders
- ASTC (LDR)
- Image atomics
- Floating point framebuffers



Vulkan

- Low-level, low-overhead, cross-platform 3D API
- Asynchronous/multithreaded command generation
- Intermediate shader binary format (SPIR-V)
- Offline shaders compilation
- Object-based API, no global state
- Explicit memory management and synchronization
- No error checking



adb shell

- Returns remote process exit status
 - Pass-through **stdin**
- ```
$ cat README | adb shell wc -l
```
- Handles window size and terminal type
  - Improved command line tools (**ls**, etc.)
  - Improved performance (push, pull)



# NDK

- Clang 3.8
- GCC 4.9
- Switch to clang, GCC is deprecated!
- ARM toolchains default to arm7 arch





# VR

```
Activity.setVrModeEnabled(boolean, ComponentName);
```

Android  
7.1

```
ActivityManager.setVrThread(int tid);
```

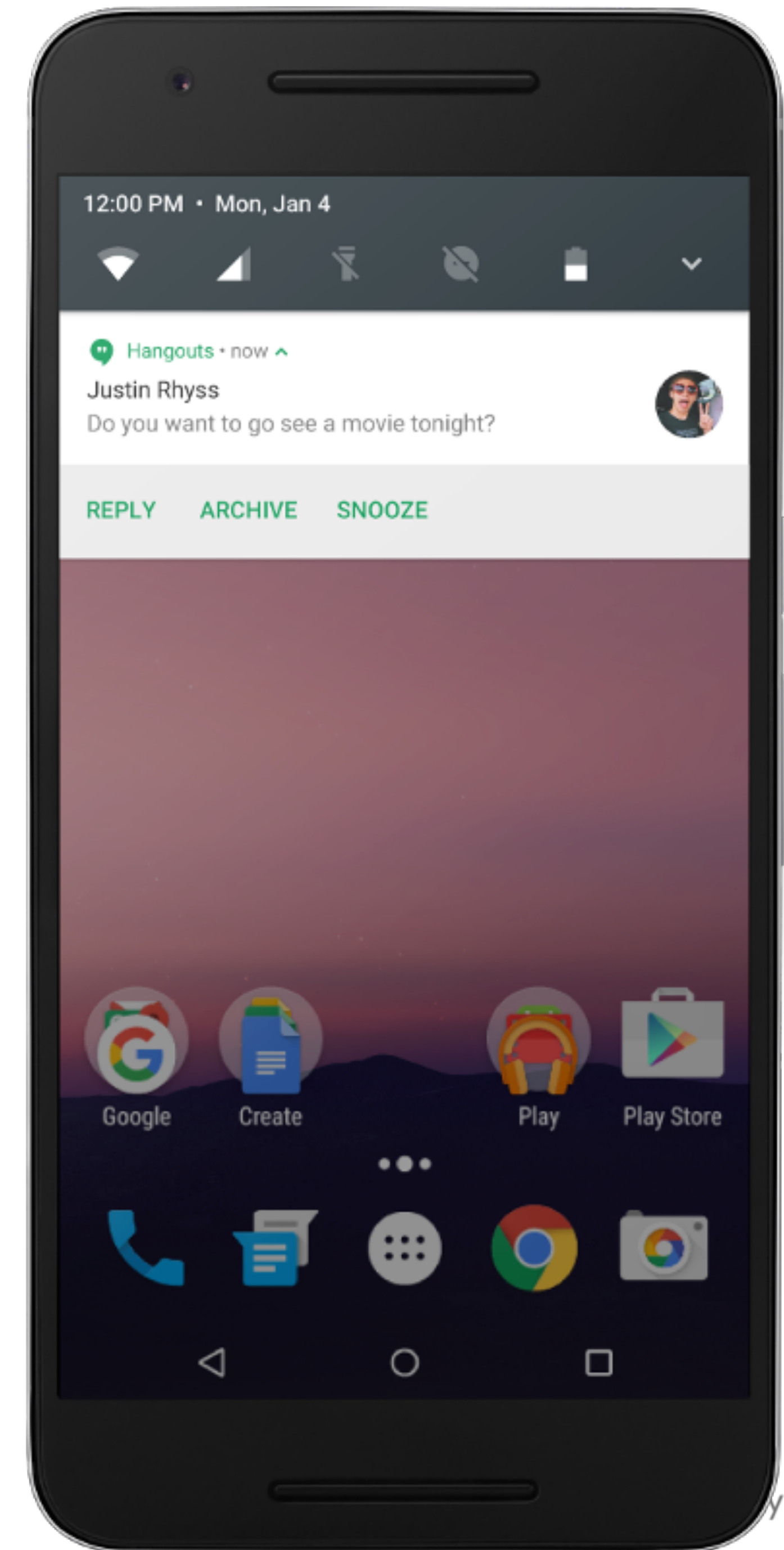




## **User-Facing Features (7.0)**

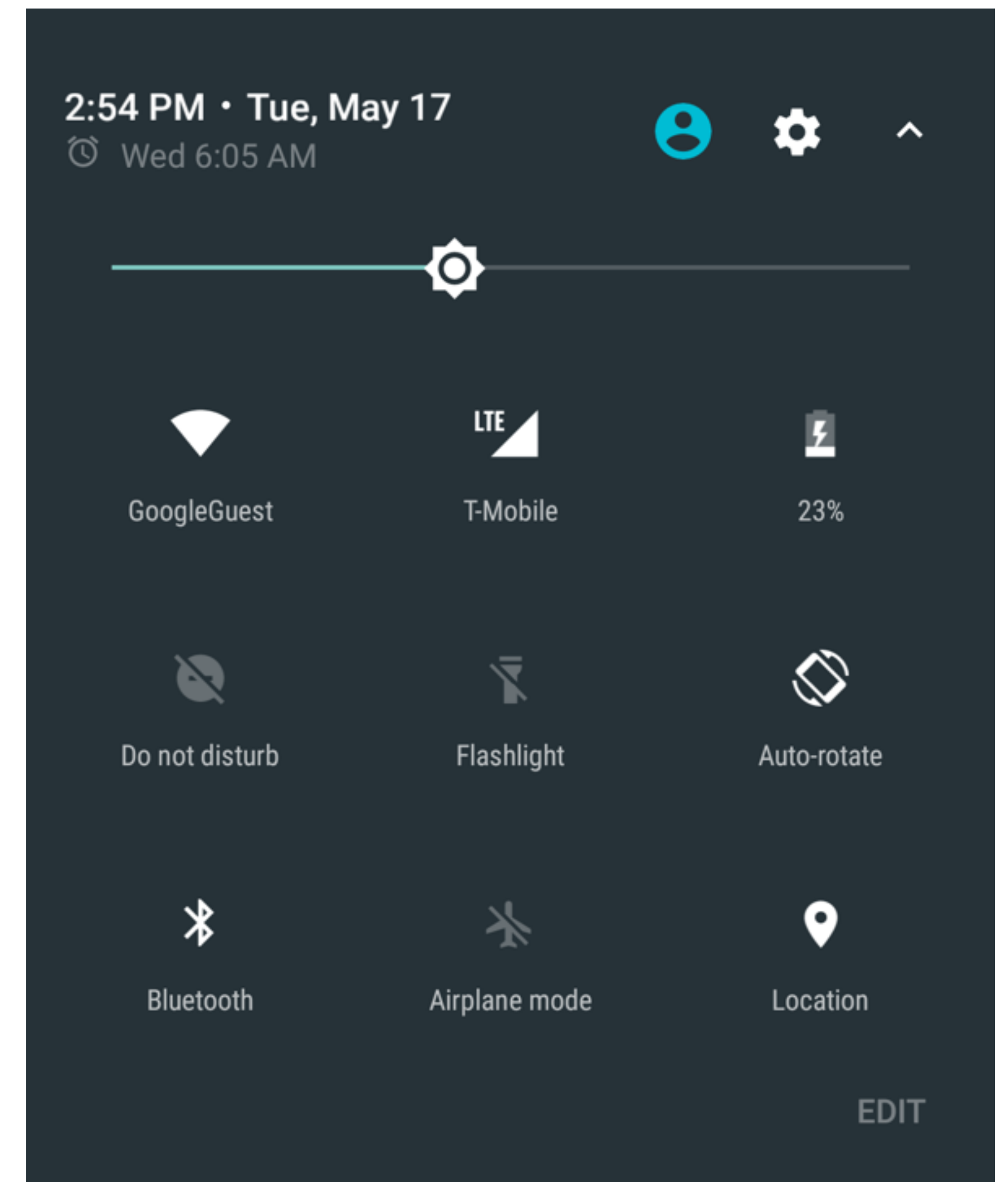
# Notifications

- New templates
- Bundled notifications
- Direct reply



# Quick Settings

- Users: Choose your favorite tiles!
  - Reorder them, too
  - Top 5 become your even-quicker-quick-settings



# Quick Settings

```
<service
 android:name=".MyTileService"
 android:label="@string/starting_label"
 android:icon="@drawable/starting_icon"
 android:permission="android.permission.BIND_QUICK_SETTINGS_TILE">
 <intent-filter>
 <action android:name="android.service.quicksettings.action.QS_TILE" />
 </intent-filter>
</service>
```





# Quick Settings - TileService

android.service.quicksettings.TileService

onTileAdded()



onStartListening()



onStopListening()



onTileRemoved()

onClick()



# Quick Settings - Tile

`android.service.quicksettings.Tile`

`// Get the tile`

`Tile tile = TileService.this.getQsTile();`

`// Change the UI of the tile.`

`tile.setLabel(newLabel);`

`tile.setIcon(newIcon);`

`tile.setState(newState);`

`// Need to call updateTile for the tile to pick up changes.`

`tile.updateTile();`



# Doze

- Marshmallow
  - Screen off, on battery, stationary
  - Restricts network, services, defers jobs, syncs, alarms
- Nougat
  - All of the above, plus
  - Screen off, on battery, **not stationary**
  - Restricts network, defers jobs, syncs



# Project Svelte

`ConnectivityManager.CONNECTIVITY_ACTION`

`Camera.ACTION_NEW_VIDEO`

`Camera.ACTION_NEW_PICTURE`

`JobScheduler`

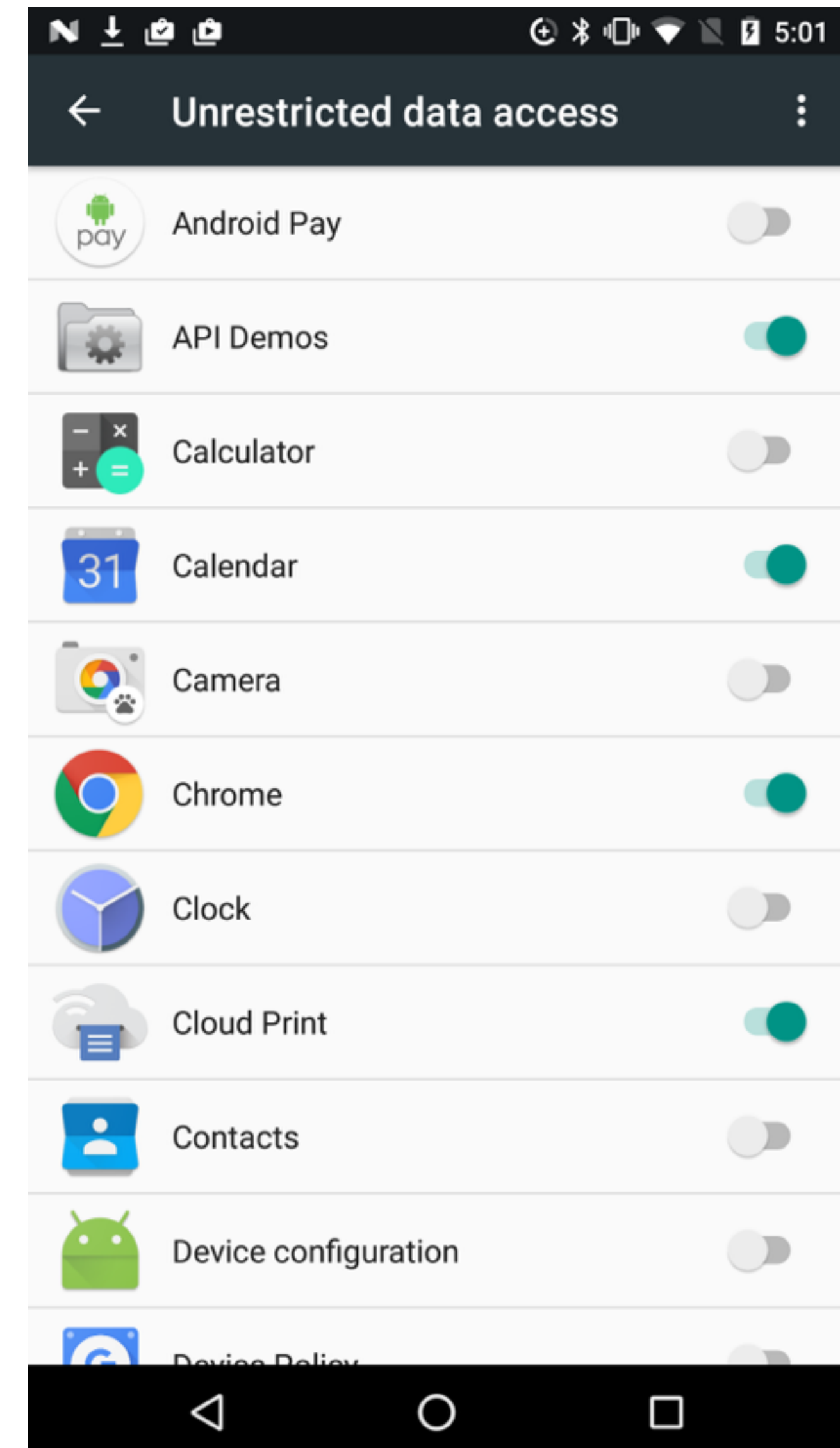
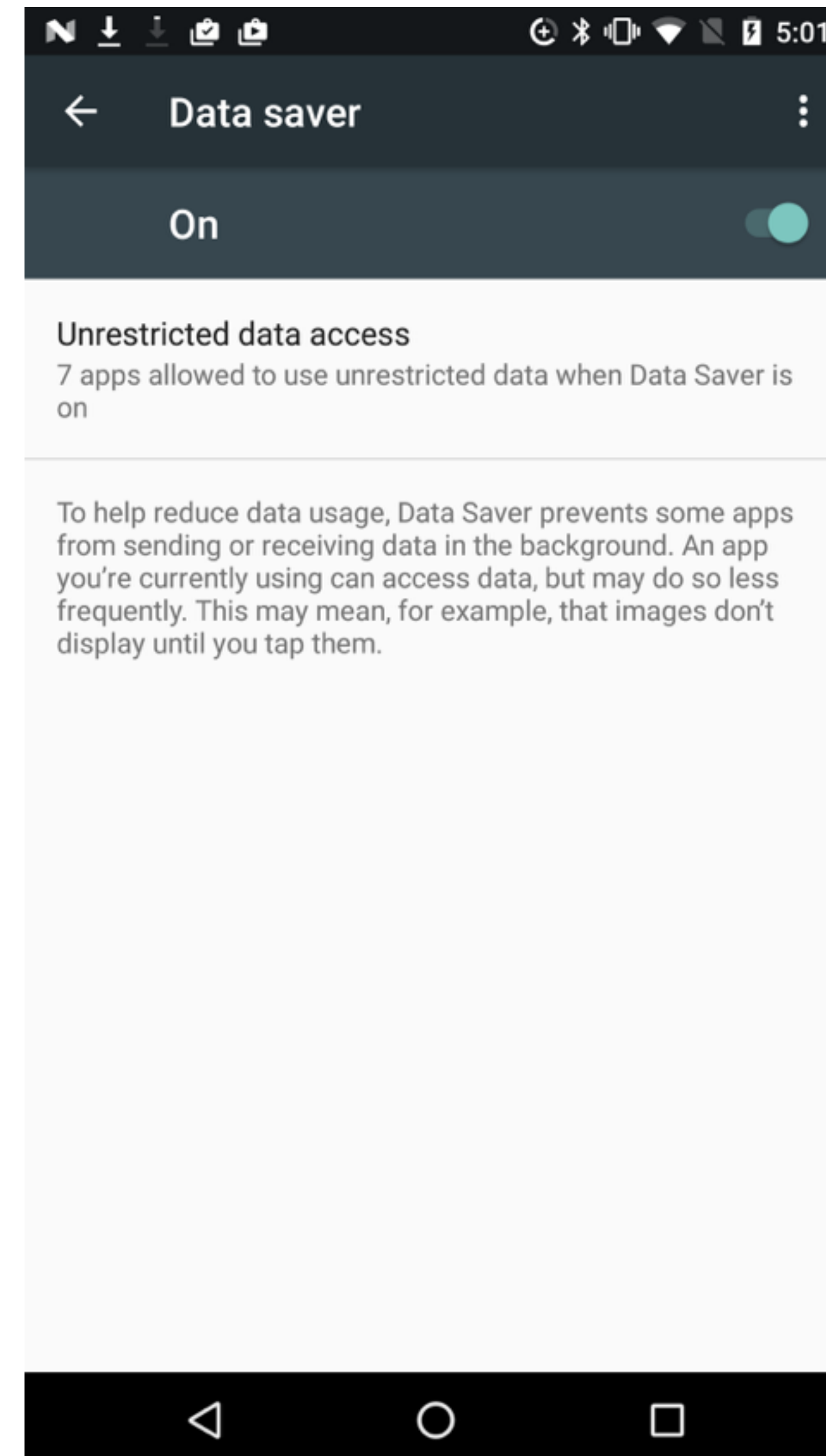
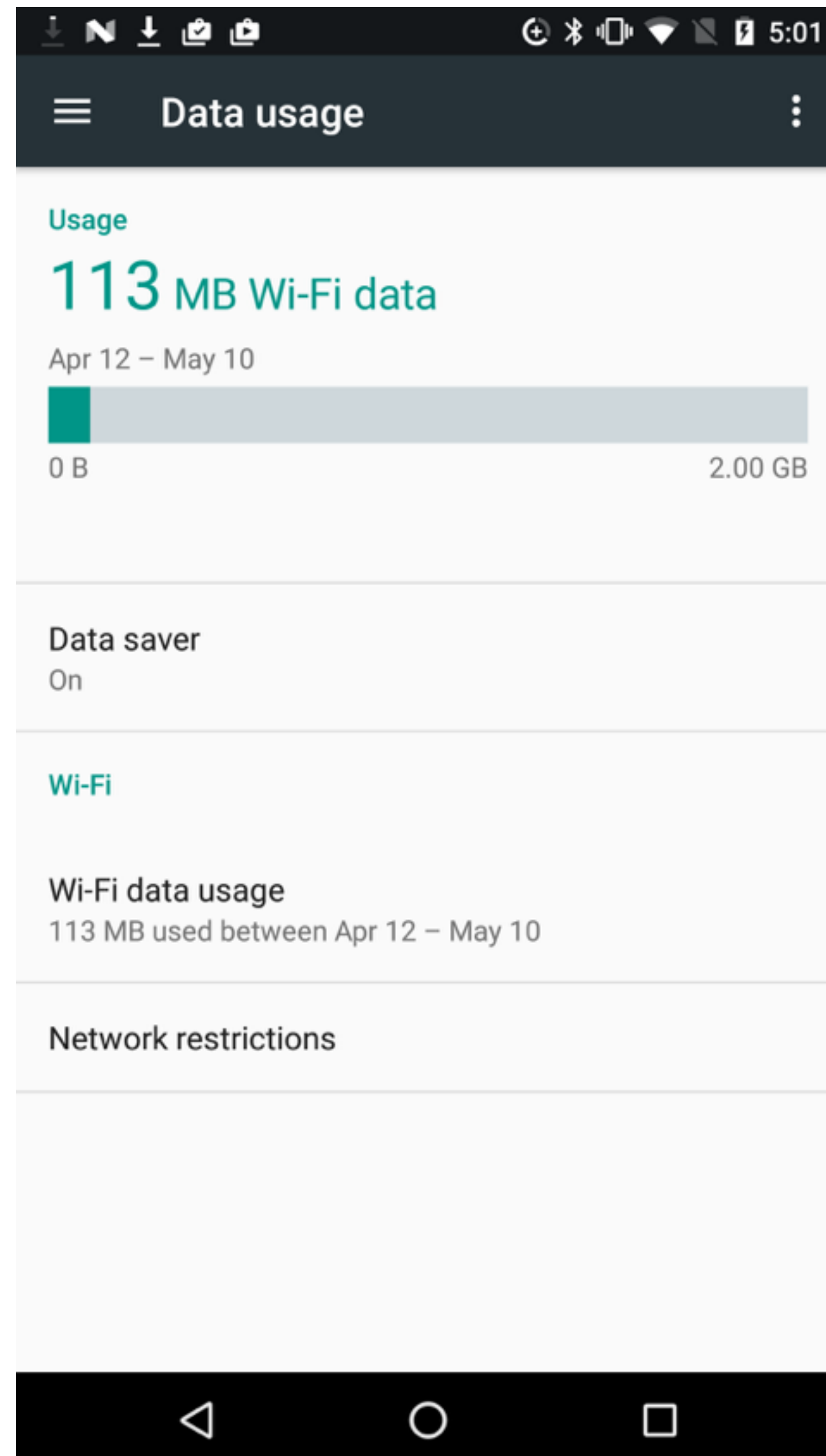
`JobScheduler.Builder.addTriggerContentUri()`

Goodbye!

Hello!



# Data Saver





# Data Saver

```
getSystemService(Context.CONNECTIVITY_SERVICE);
```

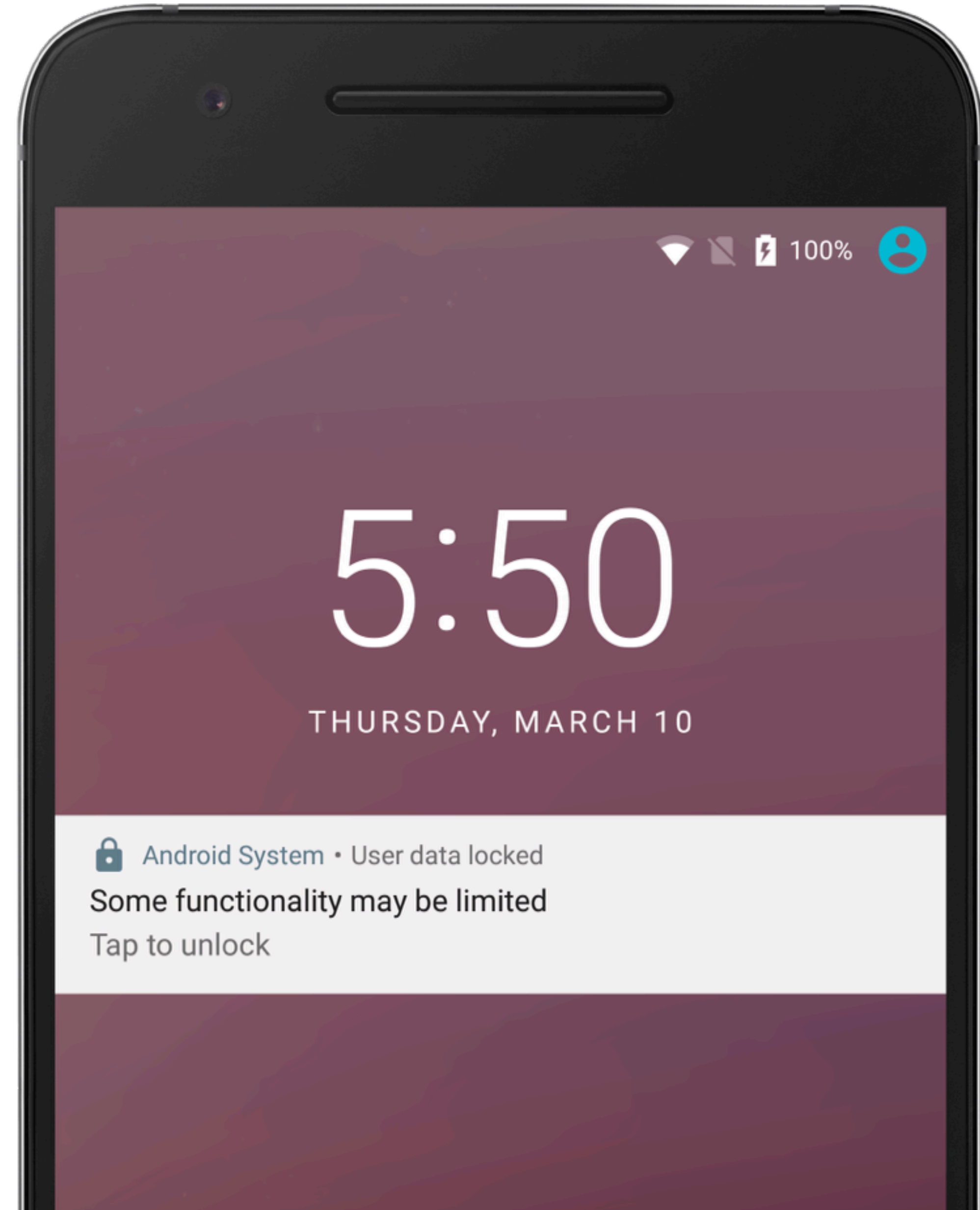
```
ConnectivityManager.isActiveNetworkMetered();
```

```
ConnectivityManager.getRestrictBackgroundStatus();
```



# Direct Boot

- Improves startup time
- Some app functionality after unexpected reboot



# Direct Boot - Manifest

```
<activity
 android:name=".DirectBootActivity"
 android:directBootAware="true">
</activity>

<receiver
 android:directBootAware="true" >
 ...
 <intent-filter>
 <action
 android:name="android.intent.action.ACTION_LOCKED_BOOT_COMPLETED" />
 </intent-filter>
 </receiver>
```



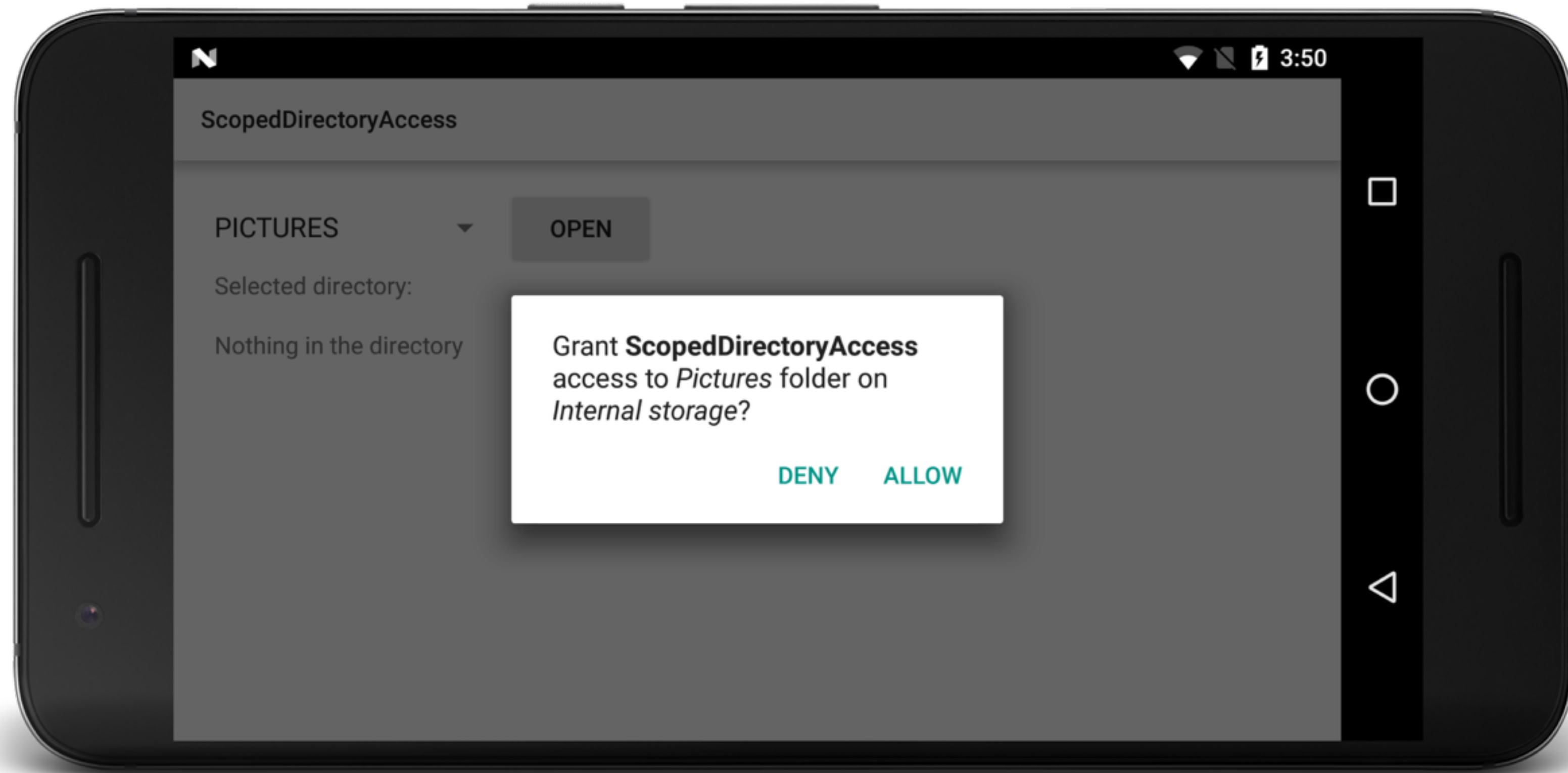
# Direct Boot - Storage

```
Context directBootContext =
 appContext.createDeviceProtectedStorageContext();
// Access appDataFilename that lives in device encrypted storage
FileInputStream inStream =
 directBootContext.openFileInput(appDataFilename);
// Use inStream to read content...
```

```
android.intent.action.USER_UNLOCKED
```

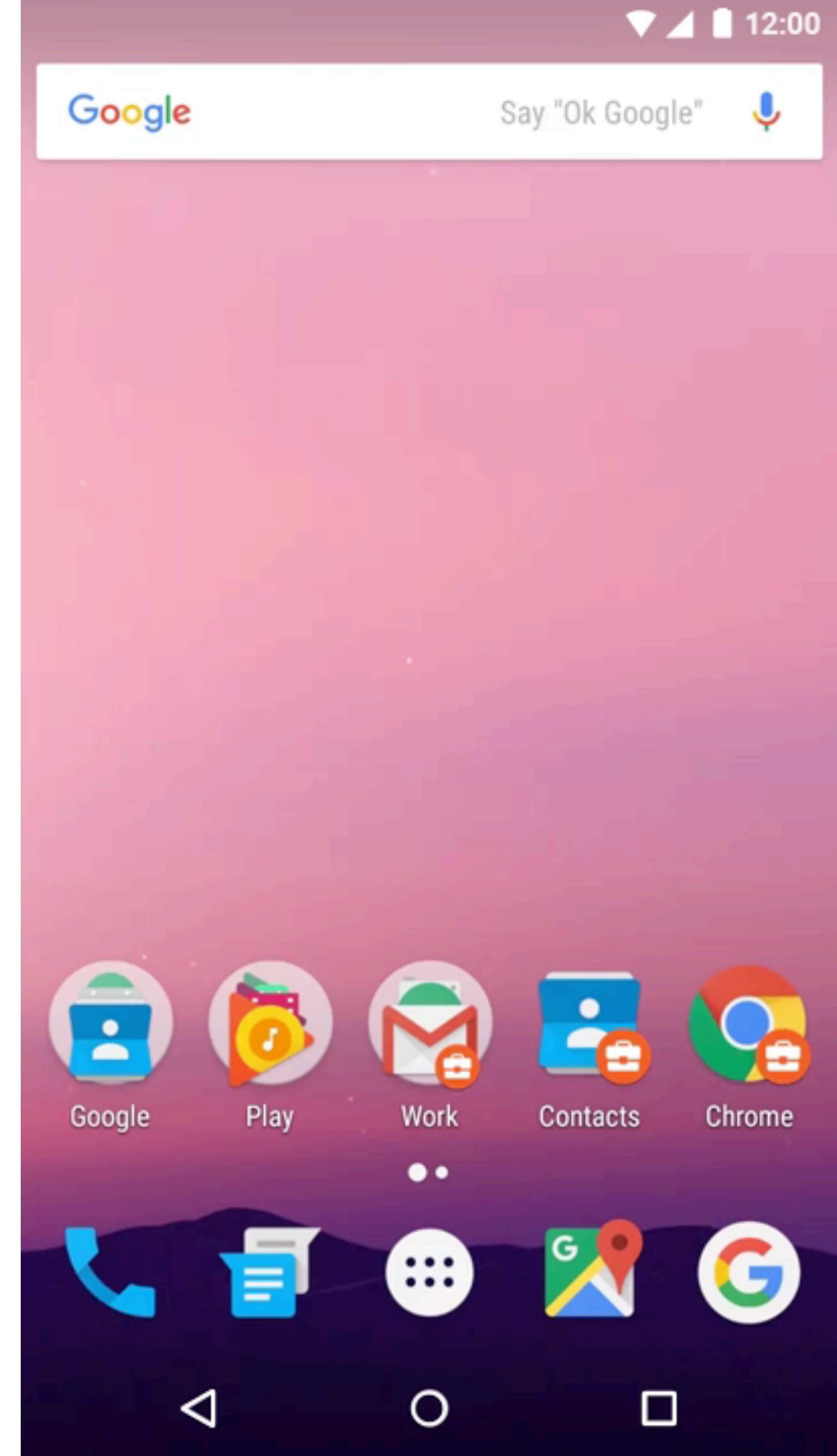


# Scoped Directory Access



# Android for Work

- Work mode
- Work challenge



# Multi-Window

```
android:resizeableActivity=["true" | "false"]
```

```
<activity android:name=".MyActivity">
 <layout android:defaultHeight="500dp"
 android:defaultWidth="600d"
 android:minimalHeight="200dp"
 android:minimalWidth="200dp"
 android:gravity="top|end"/>
</activity>
```

```
Activity.isInMultiWindowMode()
```

```
Activity.onMultiWindowModeChanged()
```

```
Intent.FLAG_ACTIVITY_LAUNCH_ADJACENT
```



# Drag and Drop

```
android.view.DragAndDropPermissions
Activity.requestDragAndDropPermissions()
```

```
View.startDragAndDrop()
```

```
View.DRAG_FLAG_GLOBAL
```

```
View.DRAG_FLAG_GLOBAL_URI_READ
```

```
View.DRAG_FLAG_GLOBAL_URI_WRITE
```

```
View.cancelDragAndDrop()
```

```
View.updateDragShadow()
```



**Now,  
between  
activities!**

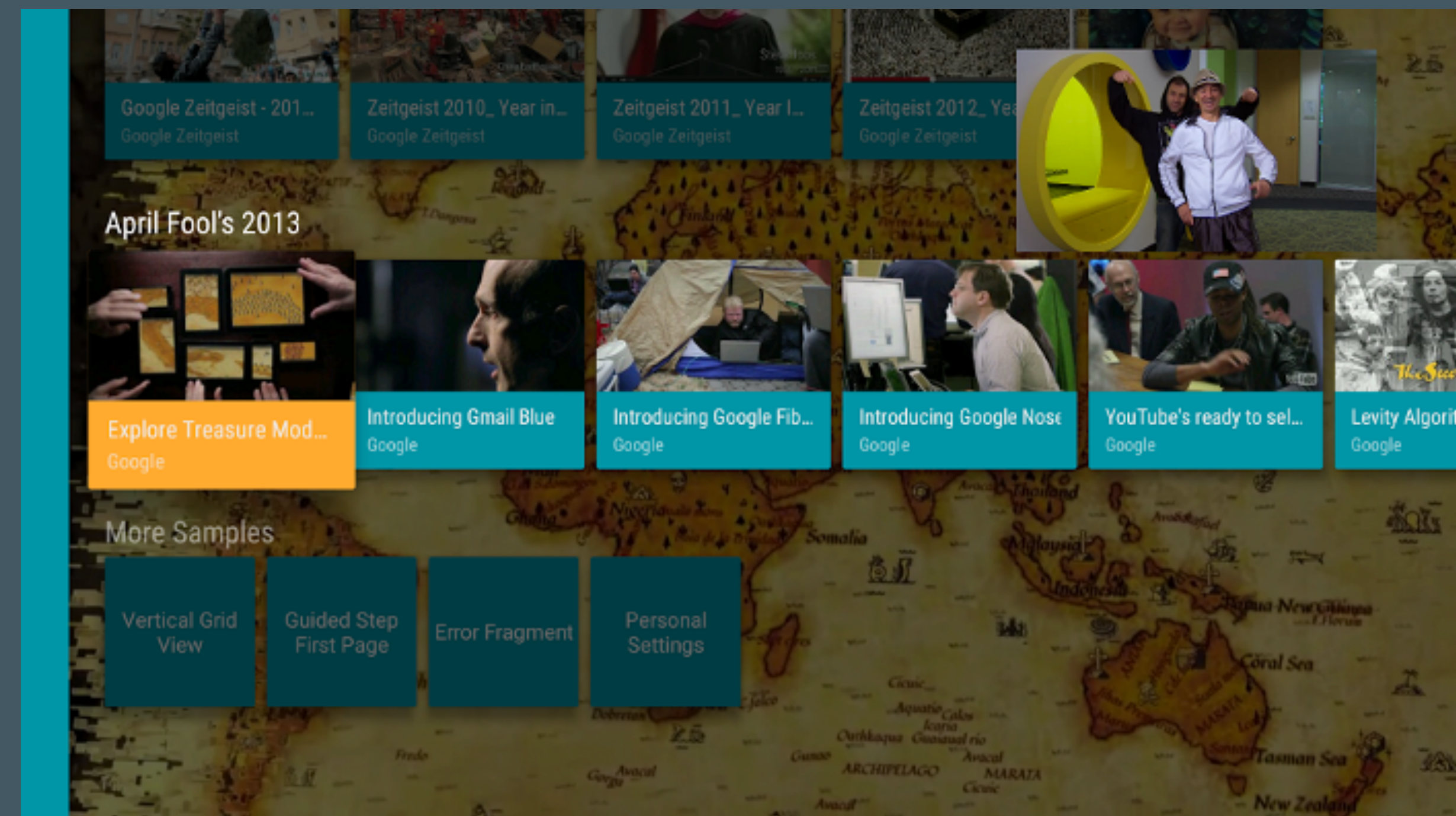
# Multi-Window: Picture in Picture

```
android:supportsPictureInPicture=["true" | "false"]
```

```
Activity.isInPictureInPictureMode()
```

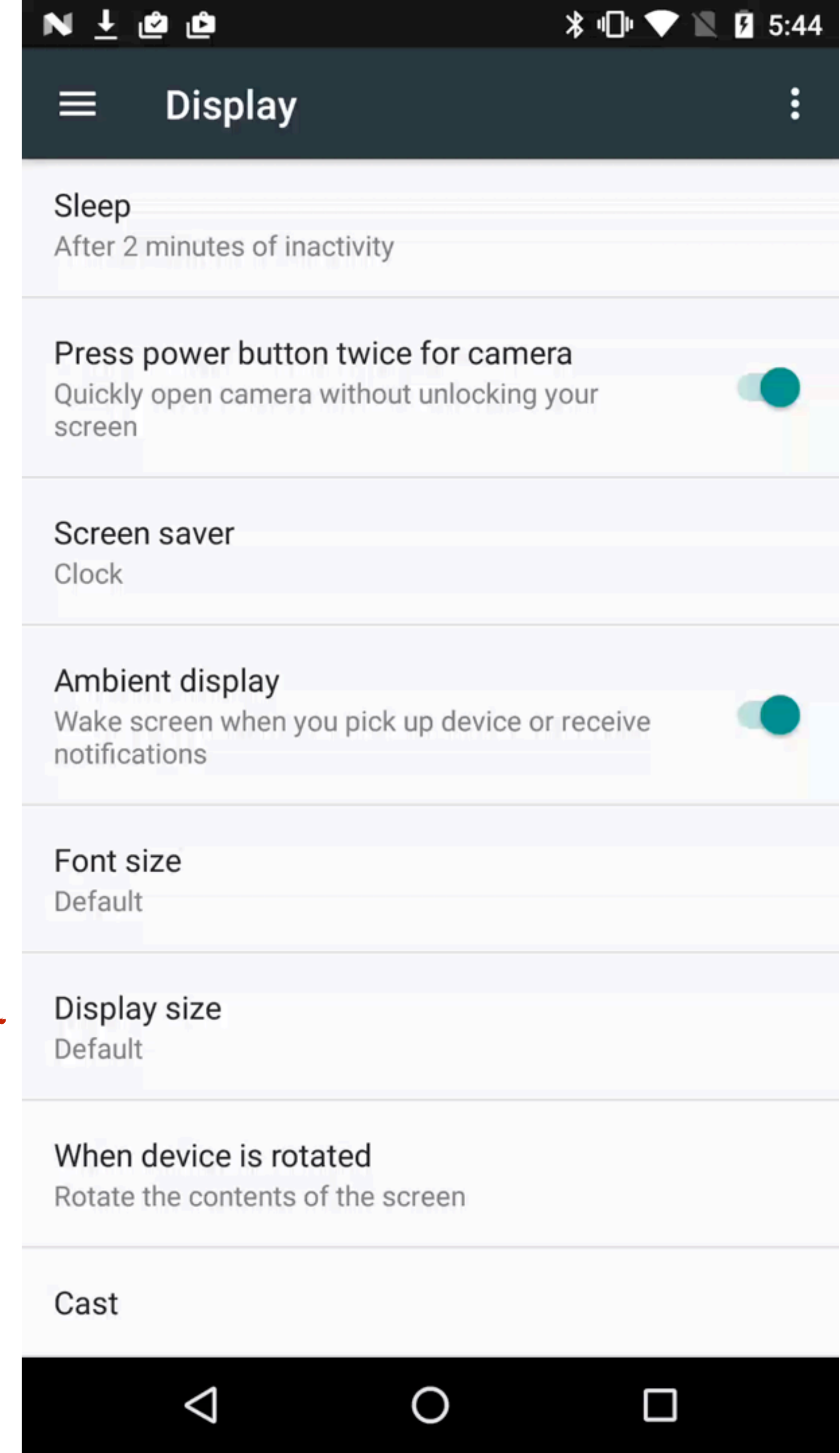
```
Activity.onPictureInPictureModeChanged()
```

```
Activity.enterPictureInPictureMode()
```



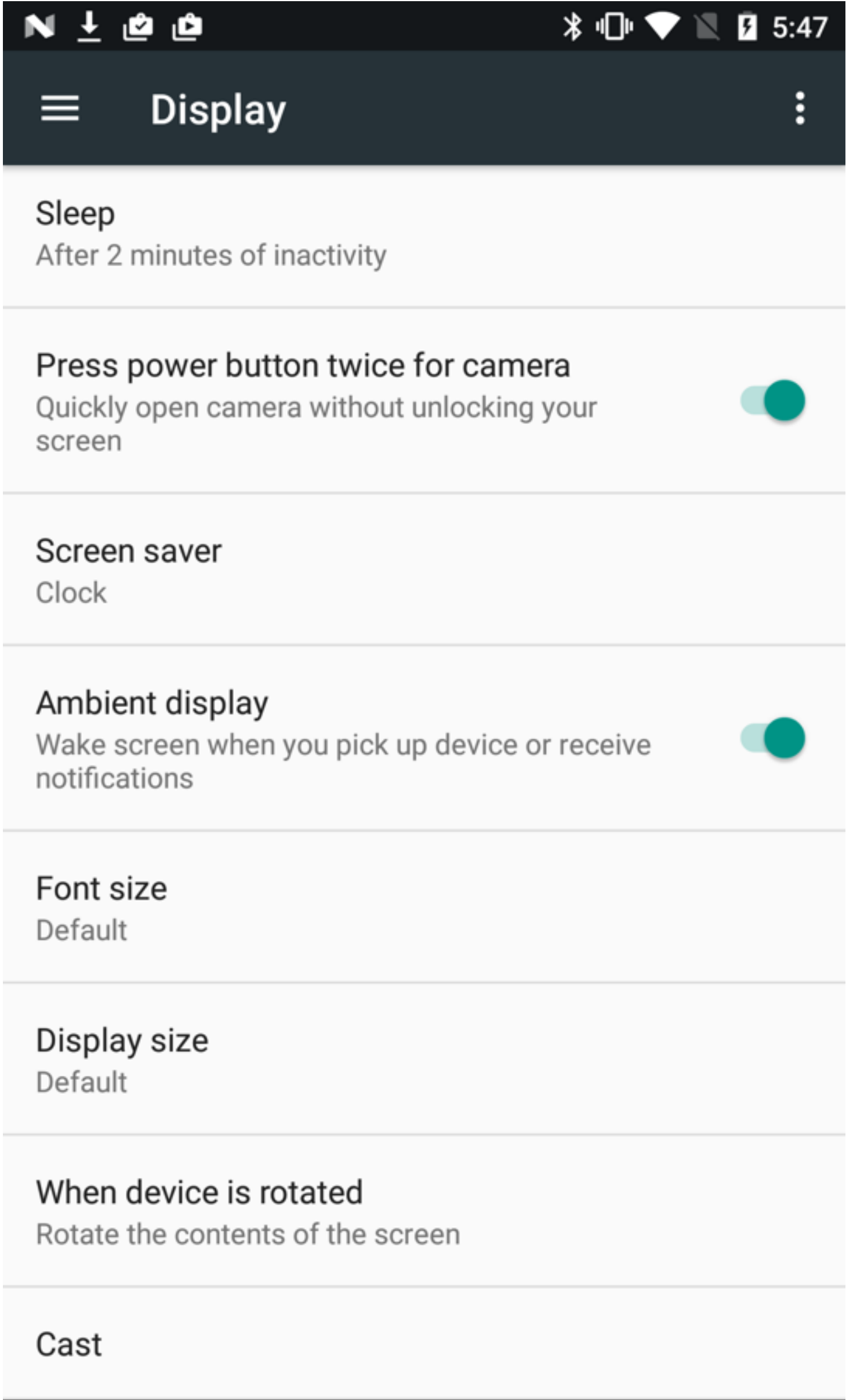
# Display Size

- User-determined display size
  - *All* UI, not just text
- .85x - 1.45x
- Avoid px
- Ensure your app works well on sw320dp

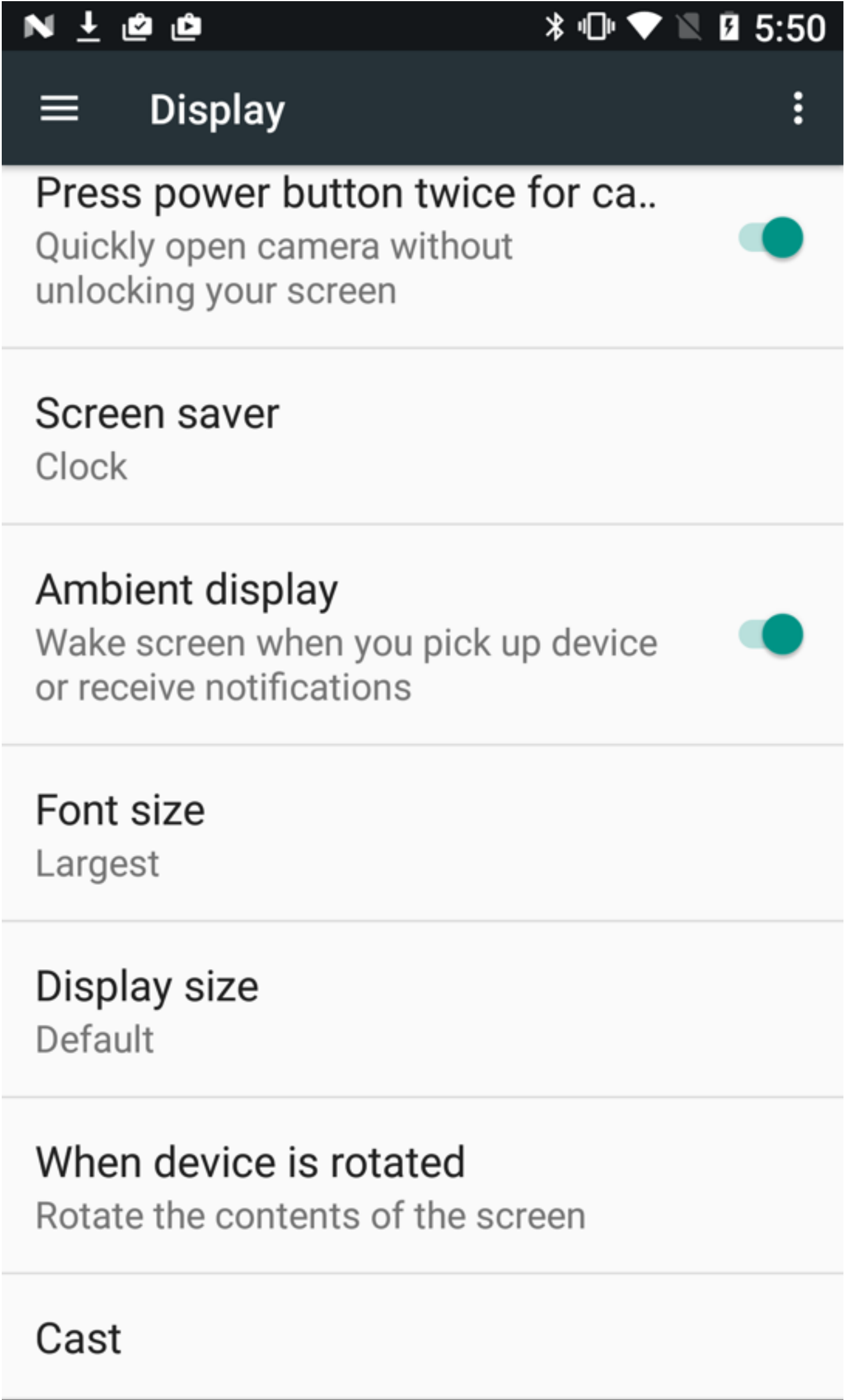




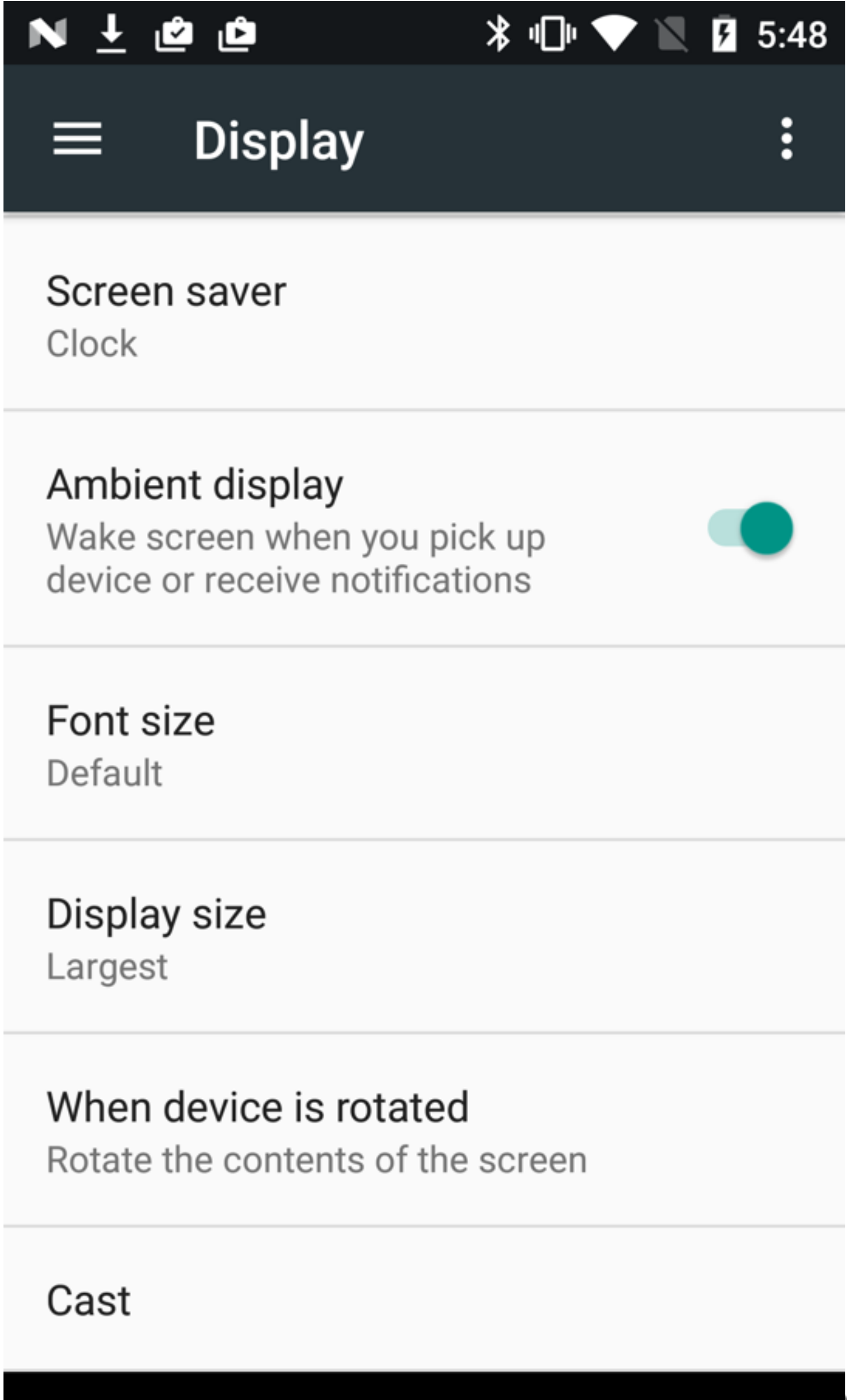
Default



Font size “Largest”



Display size “Largest”



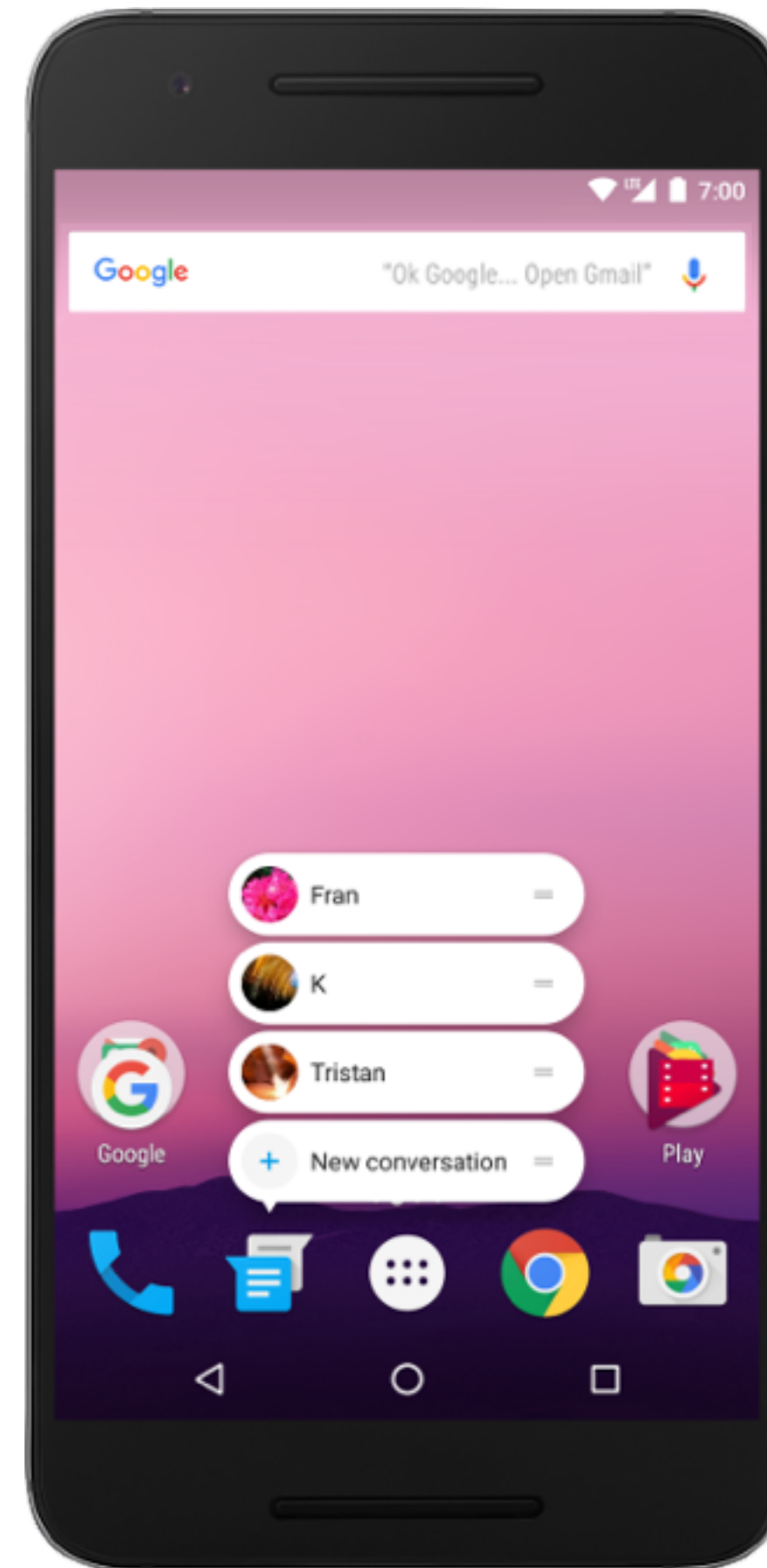
# Multi-Locale

- User can select multiple languages
  - And order them
- Also added new languages, variants



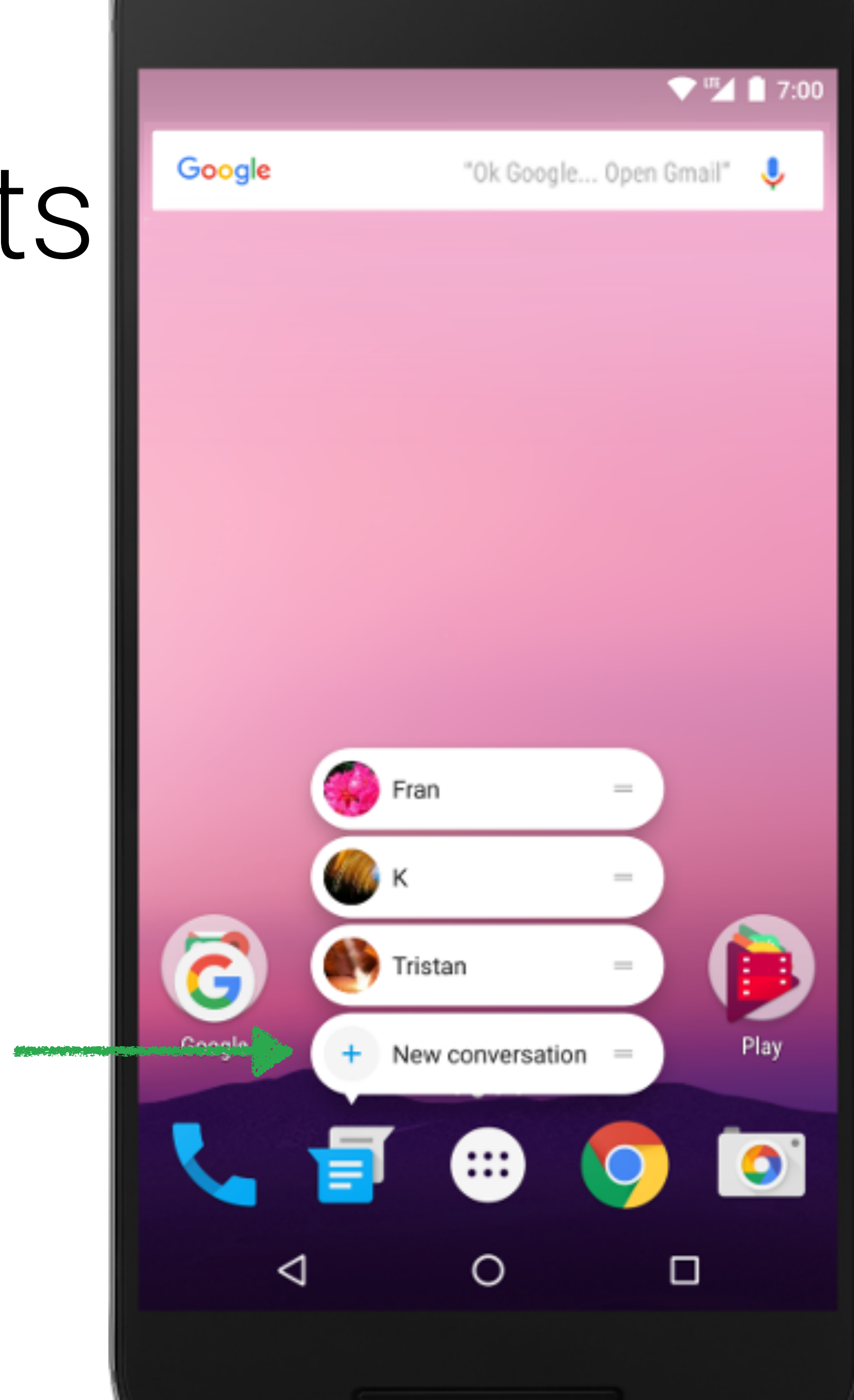
## **User-Facing Features (7.1)**

# App Shortcuts

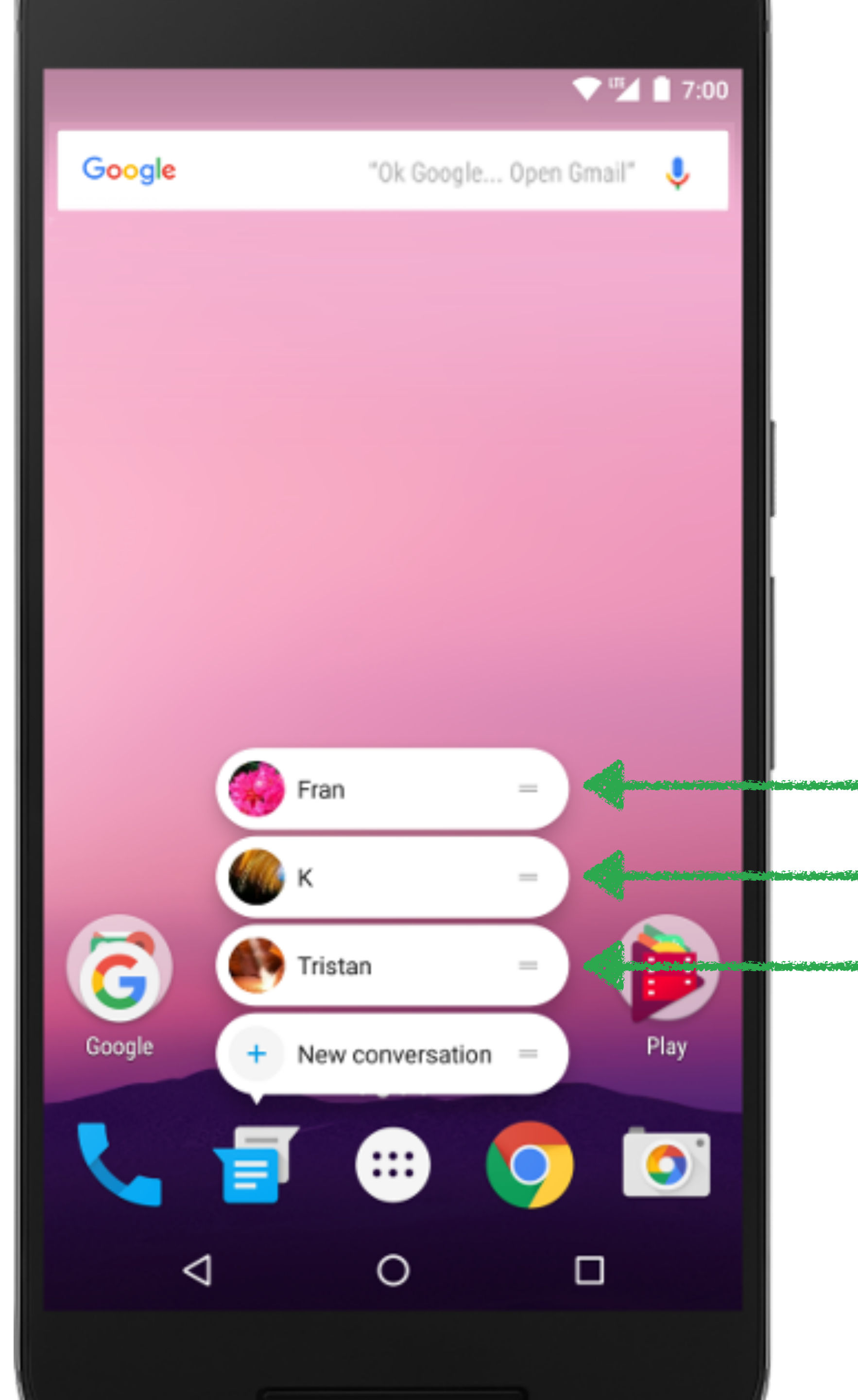


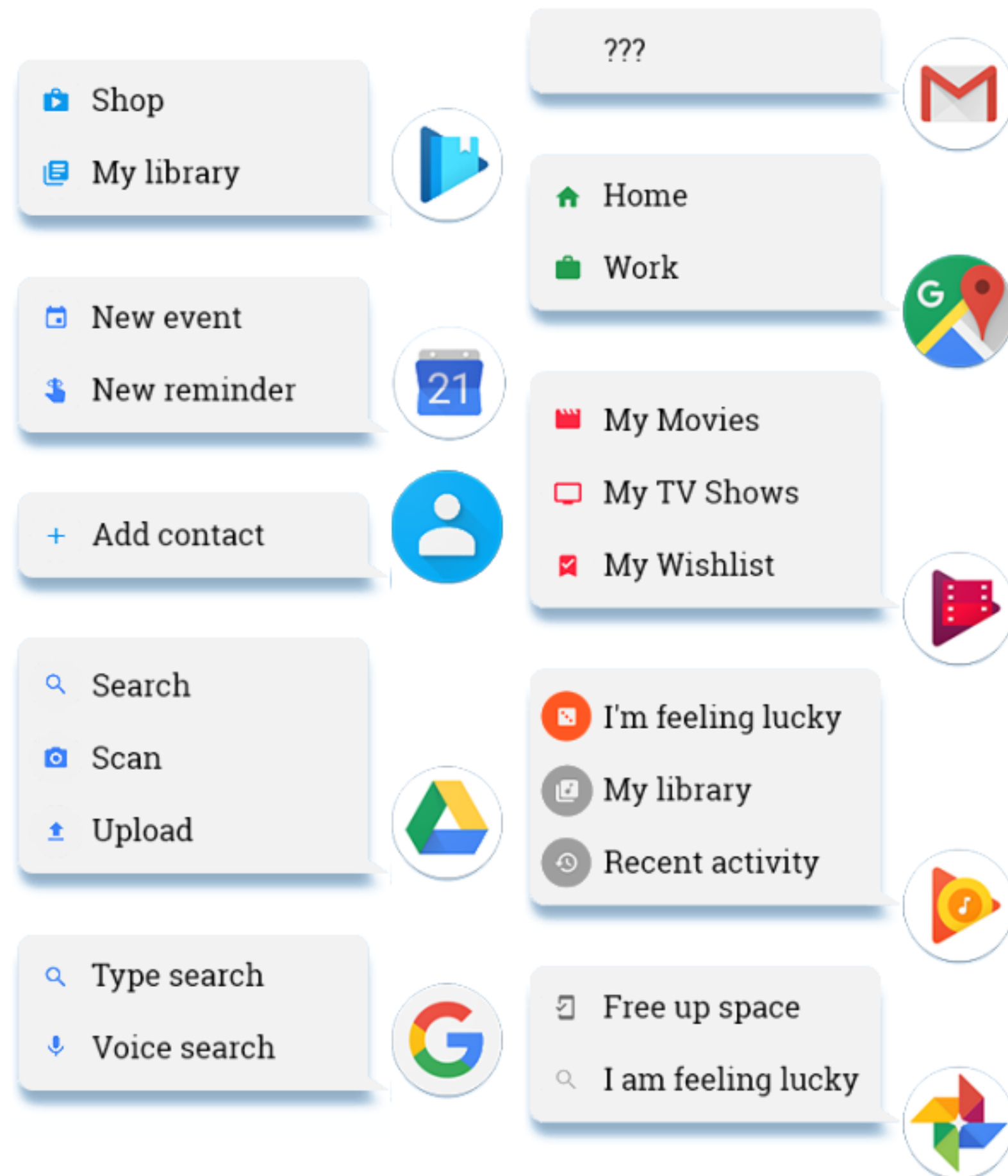


# App Shortcuts



# App Shortcuts





# Static App Shortcuts: Resource

```
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
 <shortcut
 → android:shortcutId="compose"
 android:enabled="true"
 → android:icon="@drawable/compose_icon"
 android:shortcutShortLabel="@string/compose_shortcut_short_label1"
 android:shortcutLongLabel="@string/compose_shortcut_long_label1"
 android:shortcutDisabledMessage="@string/compose_disabled_message1">
 → <intent
 android:action="android.intent.action.VIEW"
 android:targetPackage="com.example.myapplication"
 android:targetClass="com.example.myapplication.ComposeActivity" />
 <categories android:name="android.shortcut.conversation" />
 </shortcut>
 </shortcuts>
```



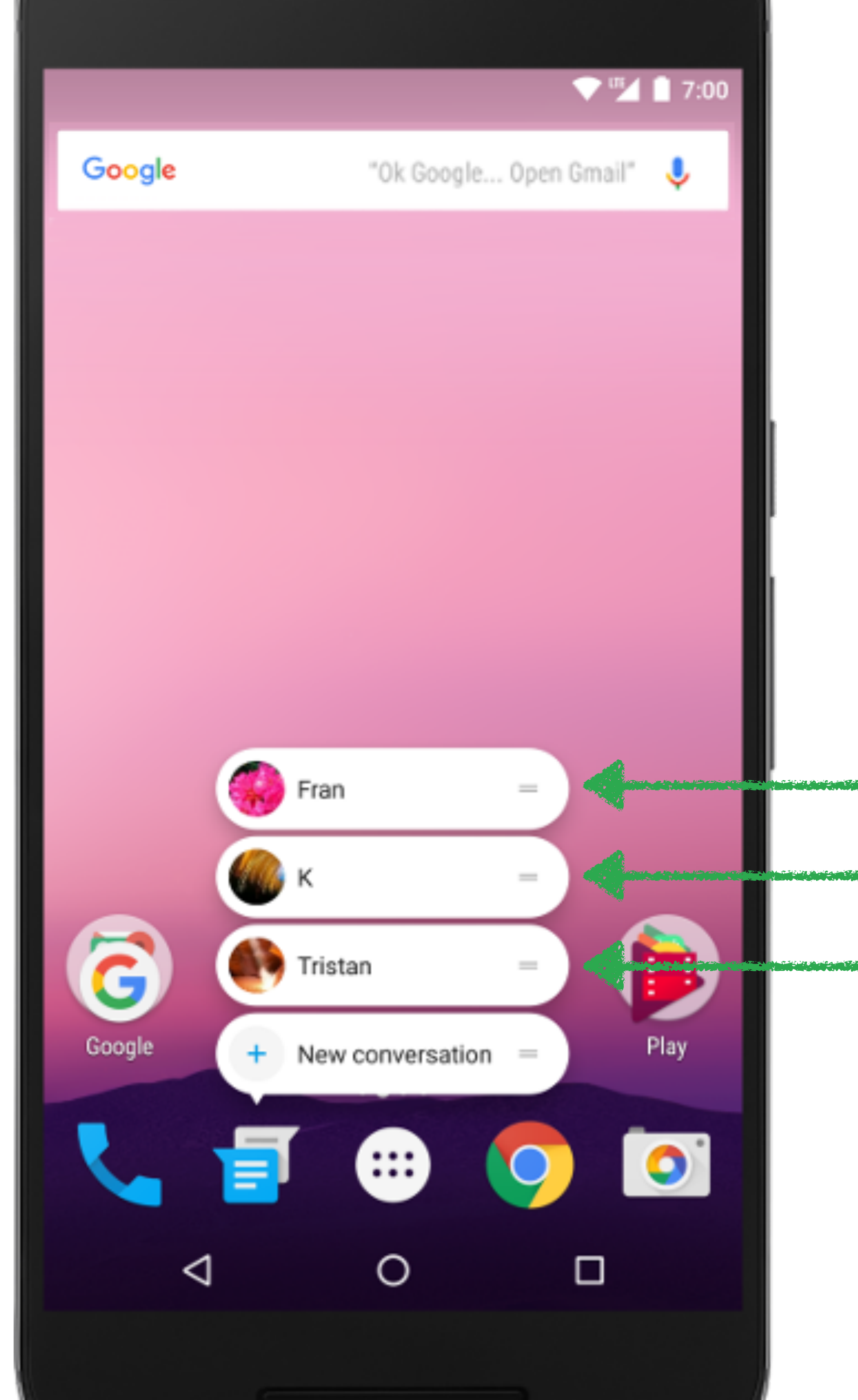
# Static App Shortcuts: Resource

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
 package="com.example.myapplication">
 <application ... >
 <activity android:name="Main">
 <intent-filter>
 <action android:name="android.intent.action.MAIN" />
 <category android:name="android.intent.category.LAUNCHER" />
 </intent-filter>
 ➔ <meta-data android:name="android.app.shortcuts"
 android:resource="@xml/shortcuts" />
 </activity>
 </application>
</manifest>
```





# App Shortcuts



# Dynamic App Shortcuts

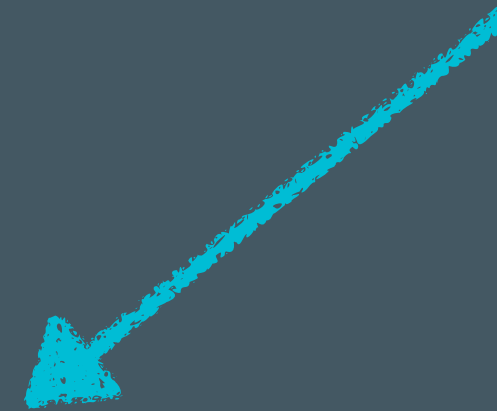
```
ShortcutInfo shortcut = new ShortcutInfo.Builder(this, shortcutId)
 .setShortLabel("Tristan")
 .setLongLabel("Tristan Jones")
 .setDisabledMessage("Contact Removed")
 .setIcon(Icon.createWithBitmap(tristanPic))
 .setIntent(new Intent(this, ComposeActivity.class)
 .setAction(Intent.ACTION_VIEW))
 .build();
```

```
ShortcutManager shortcutManager =
 getSystemService(ShortcutManager.class);
shortcutManager.setDynamicShortcuts(Arrays.asList(shortcut));
```





# Dynamic App Shortcuts - Backup/Restore



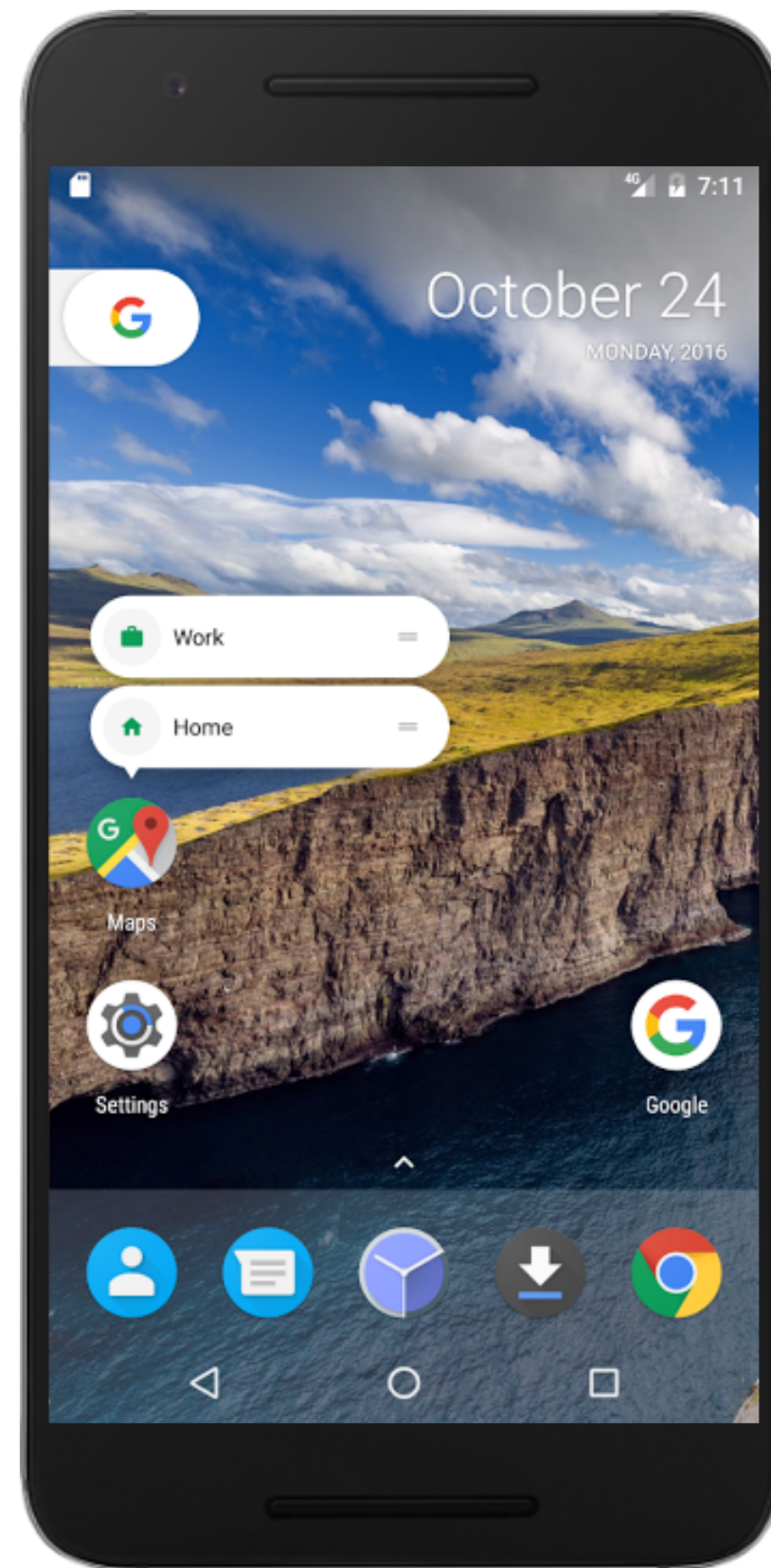
```
if (shortcutManager.getDynamicShortcuts().size() == 0) {
 // Application restored. Need to re-publish dynamic shortcuts.
 if (shortcutManager.getPinnedShortcuts().size() > 0) {
 // Pinned shortcuts have been restored. Use
 // updateShortcuts(List) to make sure they
 // contain up-to-date information.
 }
}
```



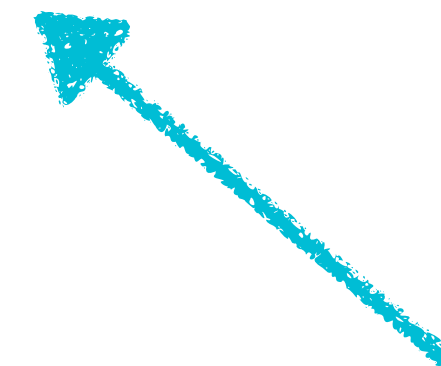
# App Shortcuts - Things to Consider

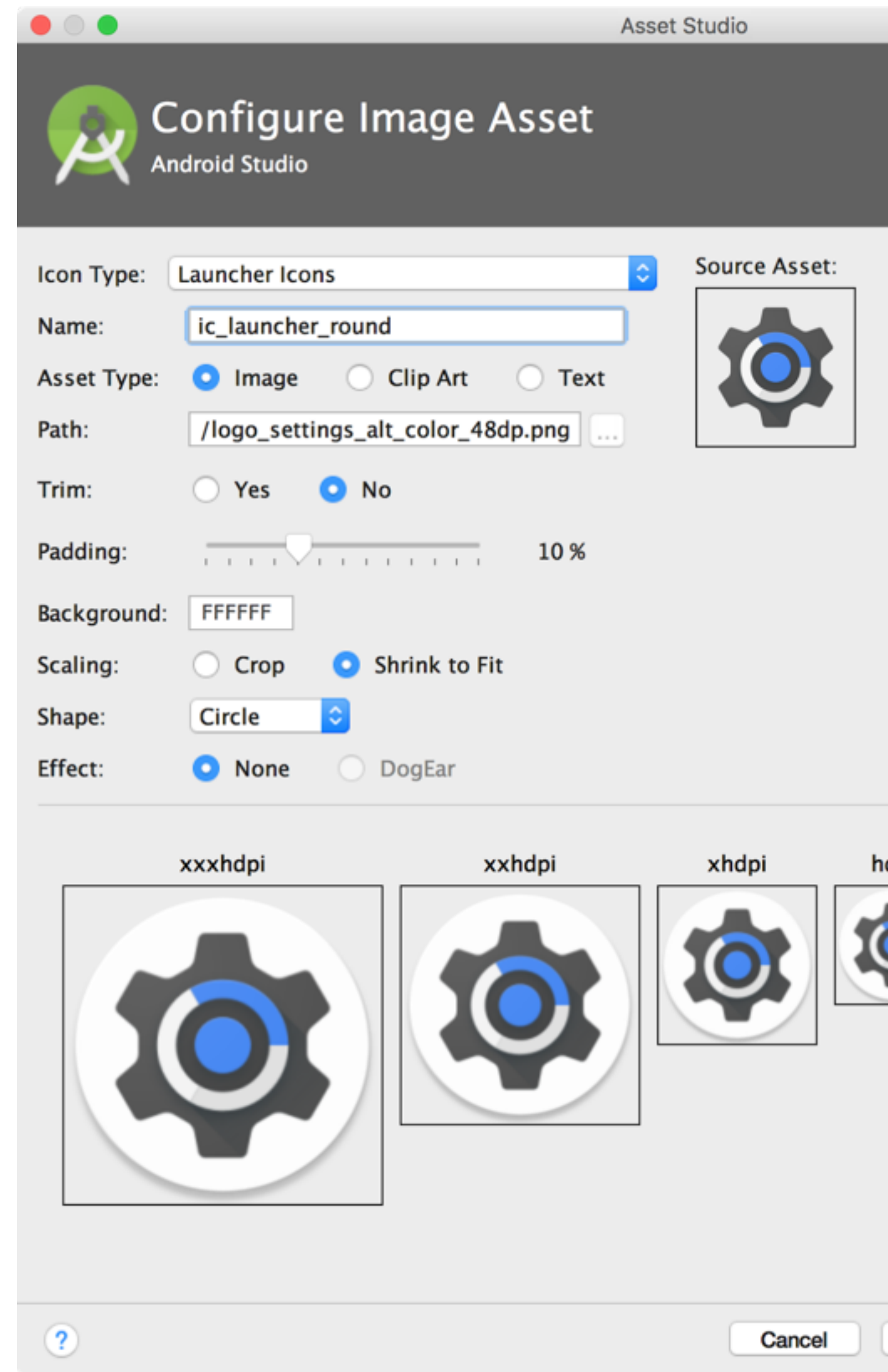
- Updates are rate-limited for background applications
  - **setDynamicShortcuts, addDynamicShortcuts, updateShortcuts**
  - Functions return false when rate-limited
  - Rate limiting resets when app returns to foreground
- Make sure to call **reportShortcutUsed**
  - Whenever the underlying functionality is used
  - System uses this to order shortcuts

# Round Icons



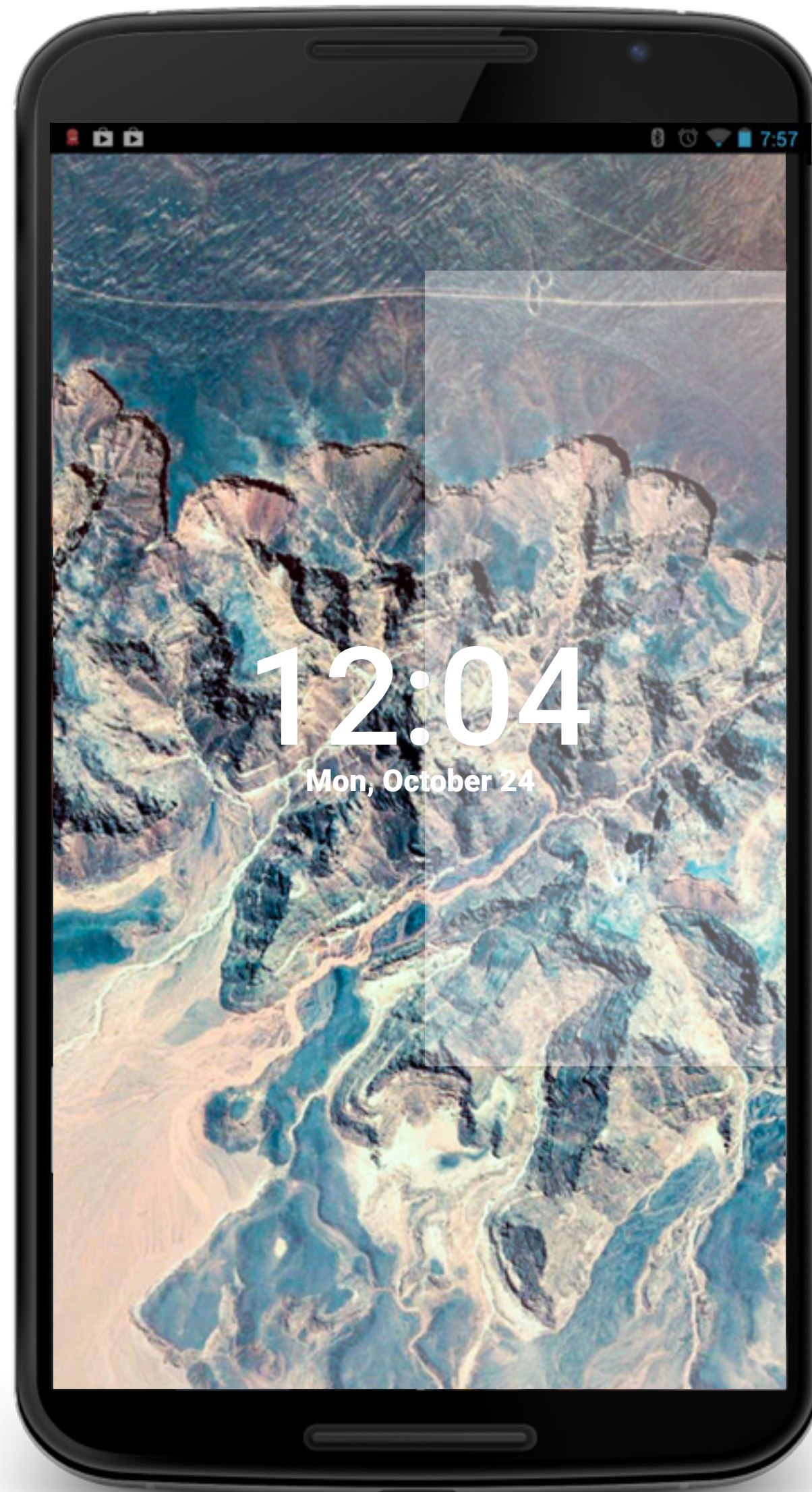
```
android:icon="@mipmap/ic_launcher"
android:roundIcon="@mipmap/ic_launcher_round"
```







# Enhanced Live Wallpaper Metadata

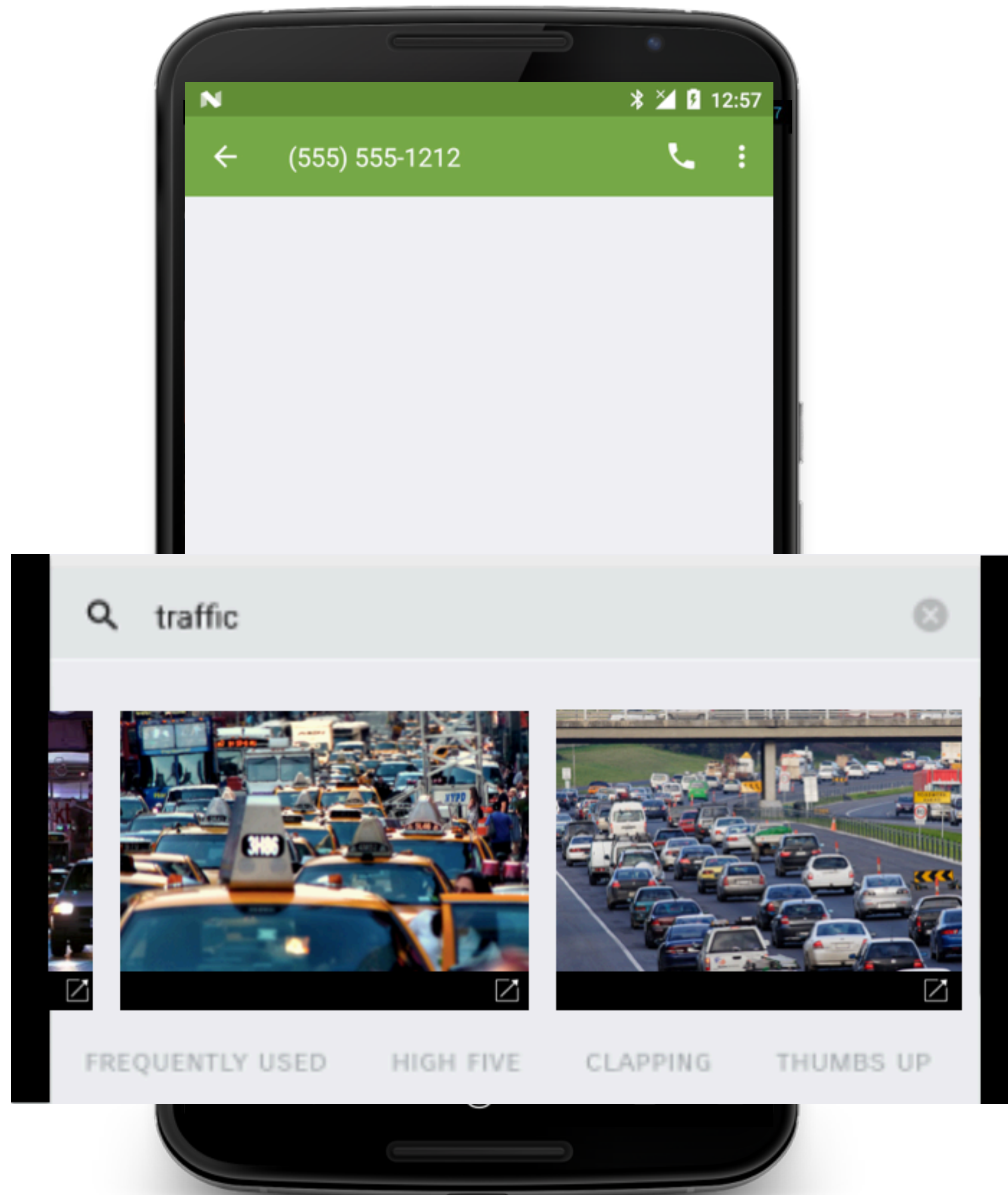


```
<wallpaper
 android:showMetadataInPreview="true"
 android:contextUri=
 "@string/wallpaper_context_uri"
 android:contextDescription=
 "@string/wallpaper_context"
/>
```



# Image Keyboard Support

Backwards  
Compatible:  
API 13



- New **CommitContent** API added in N MR1 allows keyboards to send images and other rich content into text fields.
- MIME-based: Declare what media types your application accepts (image/png, image/gif, image/jpeg, etc...)
- Open API: Third party developers can develop custom keyboards.

# Accepting image input...

```
private EditText createEditTextWithContentMimeTypes(final String[] mimeTypes) {
 → EditText exitText = new EditText(this) {
 @Override
 → public InputConnection onCreateInputConnection(EditorInfo editorInfo) {
 final InputConnection ic = super.onCreateInputConnection(editorInfo);

 EditorInfoCompat.setContentMimeTypes(editorInfo, mimeTypes);

 final InputConnectionCompat.OnCommitContentListener callback =
 new InputConnectionCompat.OnCommitContentListener() {
 int flags, Bundle opts) {
 // ...
 };
 return InputConnectionCompat.createWrapper(ic, editorInfo, callback);
 }
 };
 return exitText;
}
```





# Accepting image input...

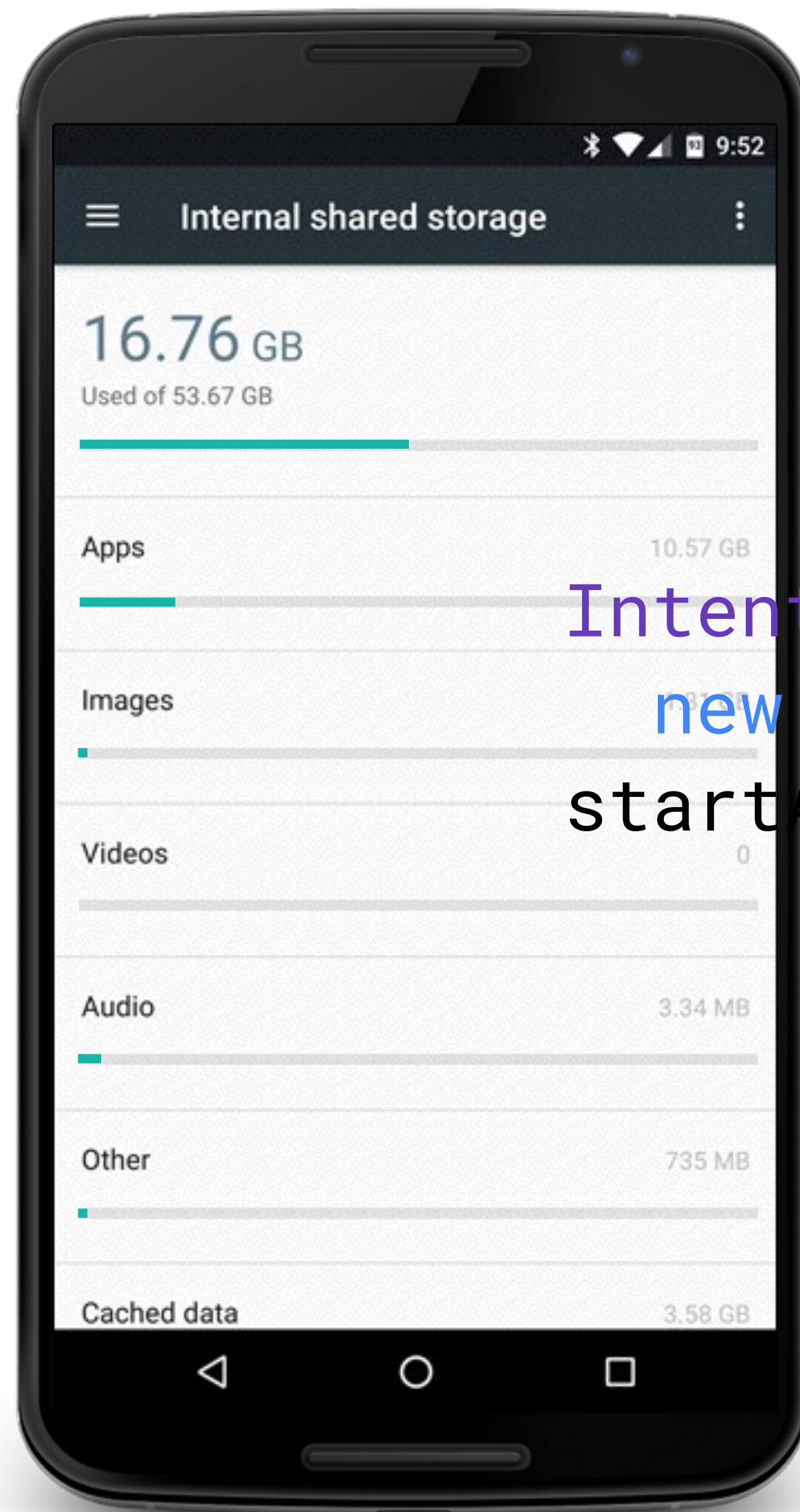


```
@Override
public boolean onCommitContent(InputContentInfoCompat inputContentInfo,
 int flags, Bundle opts){
 if ((flags&InputConnectionCompat.INPUT_CONTENT_GRANT_READ_URI_PERMISSION) != 0) {
 try {
 inputContentInfo.requestPermission();
 } catch (Exception e) {
 Log.e(TAG, "InputContentInfoCompat#requestPermission() failed.", e);
 return false;
 }
 }
 // Render image asynchronously
 Uri uri = inputContentInfo.getContentUri();
 renderImage(uri);

 // Public link for sending off-device
 Uri publicUri = inputContentInfo.getLinkUri();
 return true;
}
```

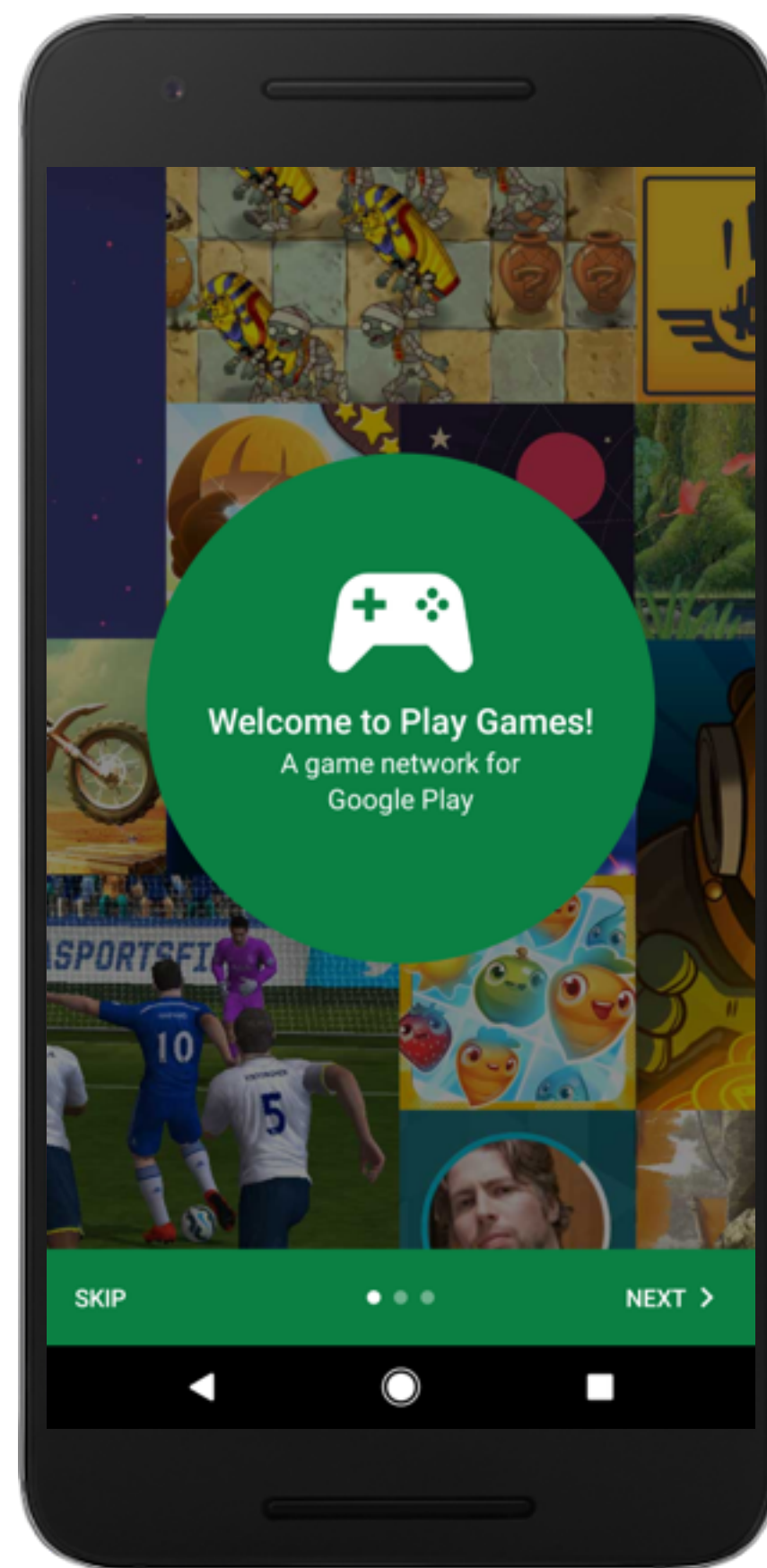


# Manage Storage Intent



```
Intent intent =
 new Intent(StorageManager.ACTION_MANAGE_STORAGE);
startActivity(intent);
```

# Demo User



```
userManager.isDemoUser();
```



Google Developer Day

# More Information

- Get the Android 7.1 Preview  
[d.android.com/preview](http://d.android.com/preview)
- See the API Diff Report  
[d.android.com/sdk/api\\_diff/25/changes.html](http://d.android.com/sdk/api_diff/25/changes.html)

