Google

Developers

# Taking Advantage of Android Platform Features

Scott Kennedy, Vikram Aggarwal
Android Application Engineers

# Who are you?

- You have...
  - Apps in the Play Store
  - Real users with feature requests
  - And bug fixes (for someone else's bugs)
- Your goal
  - Happy users
  - Improve your productivity
  - More users

# You

- have...
  - Apps
  - Users
  - Ideas
- want...
  - Less work
  - More users
  - More $$$

We wrote the Gmail app

We wrote the Gmail app...

*Along with many other qualified engineers

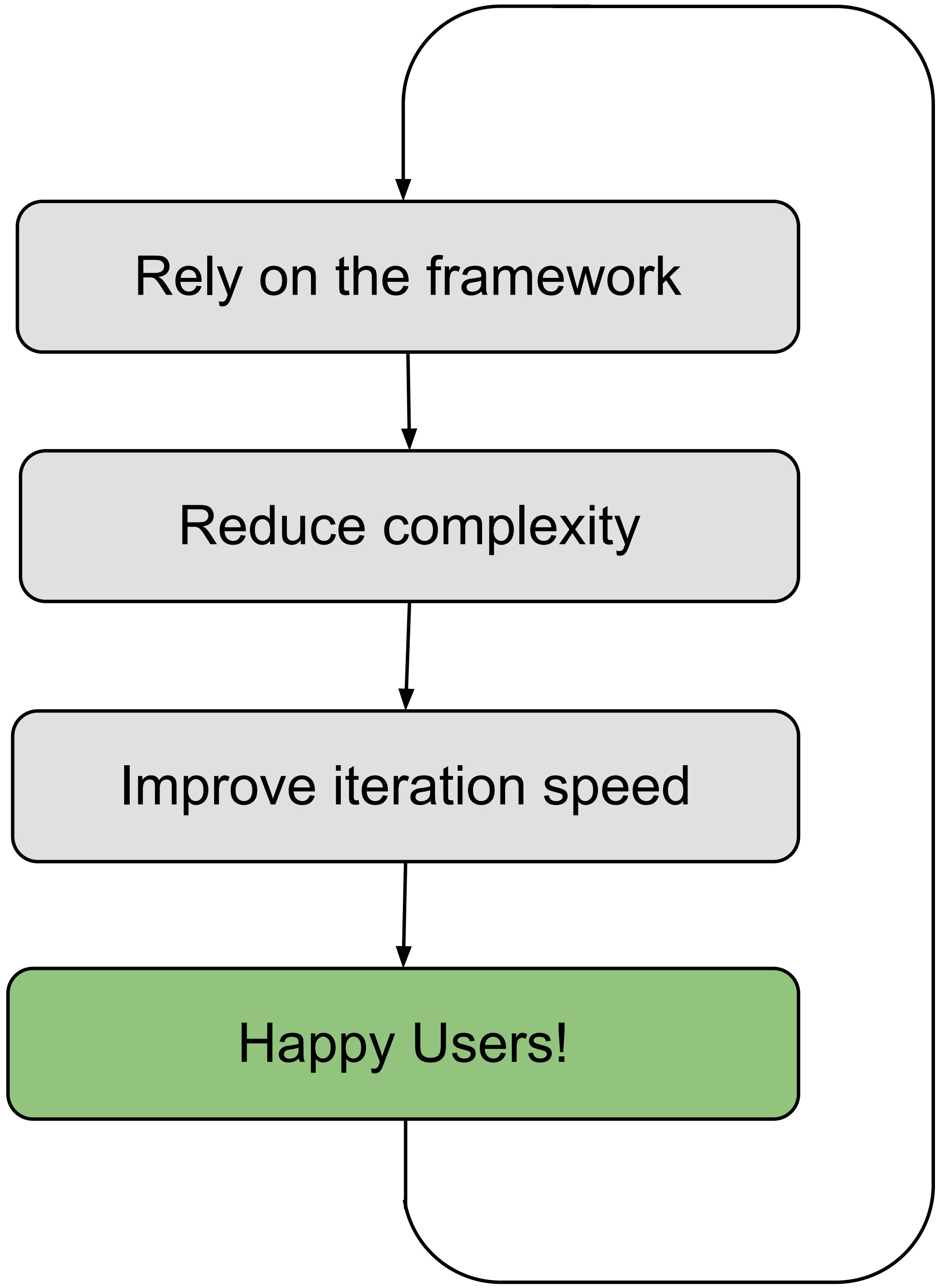
We wrote the Gmail app...

Along with many other qualified engineers**

**Everyone else is busy?

# Android Gmail

- Do less
- Rely on the framework
- Reduce complexity
- Make users happy



# Modularize UI: Fragments

**res/layout/topview.xml:**

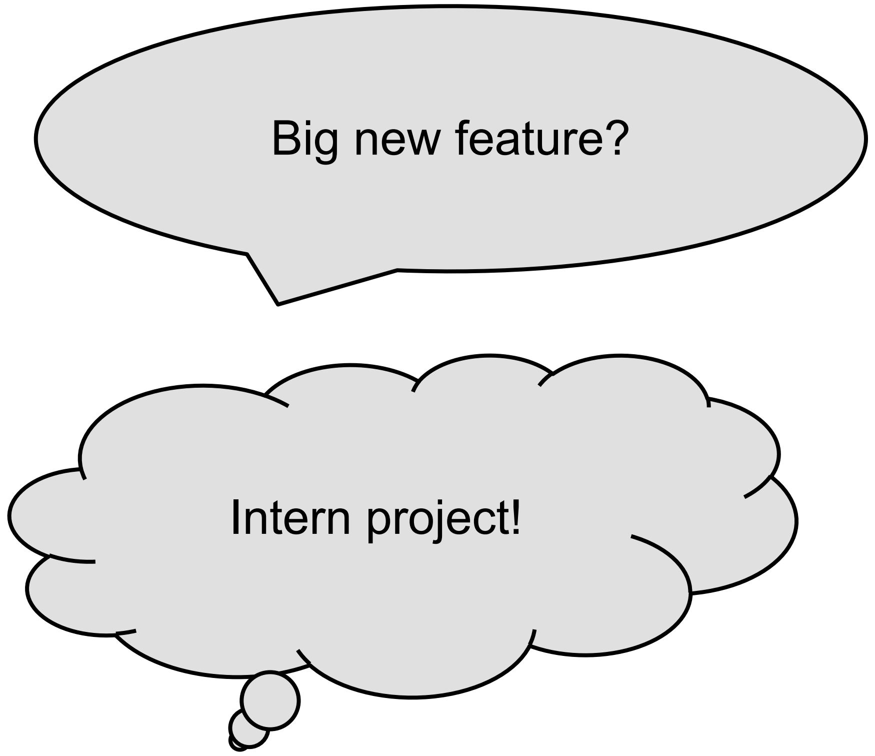
```
<fragment class="com.example.ListFragment"
    android:layout_height="match_parent"
    android:layout_gravity="start" />
```

---

**res/layout-sw600dp/topview.xml:**

```
<LinearLayout>
  <fragment class="com.example.ListFragment"
      android:layout_height="match_parent"
      android:layout_width="match_parent" />

  <fragment class="com.example.ContentFragment"
      android:layout_height="match_parent"
      android:layout_width="match_parent" />
</LinearLayout>
```



Big new feature?

Intern project!

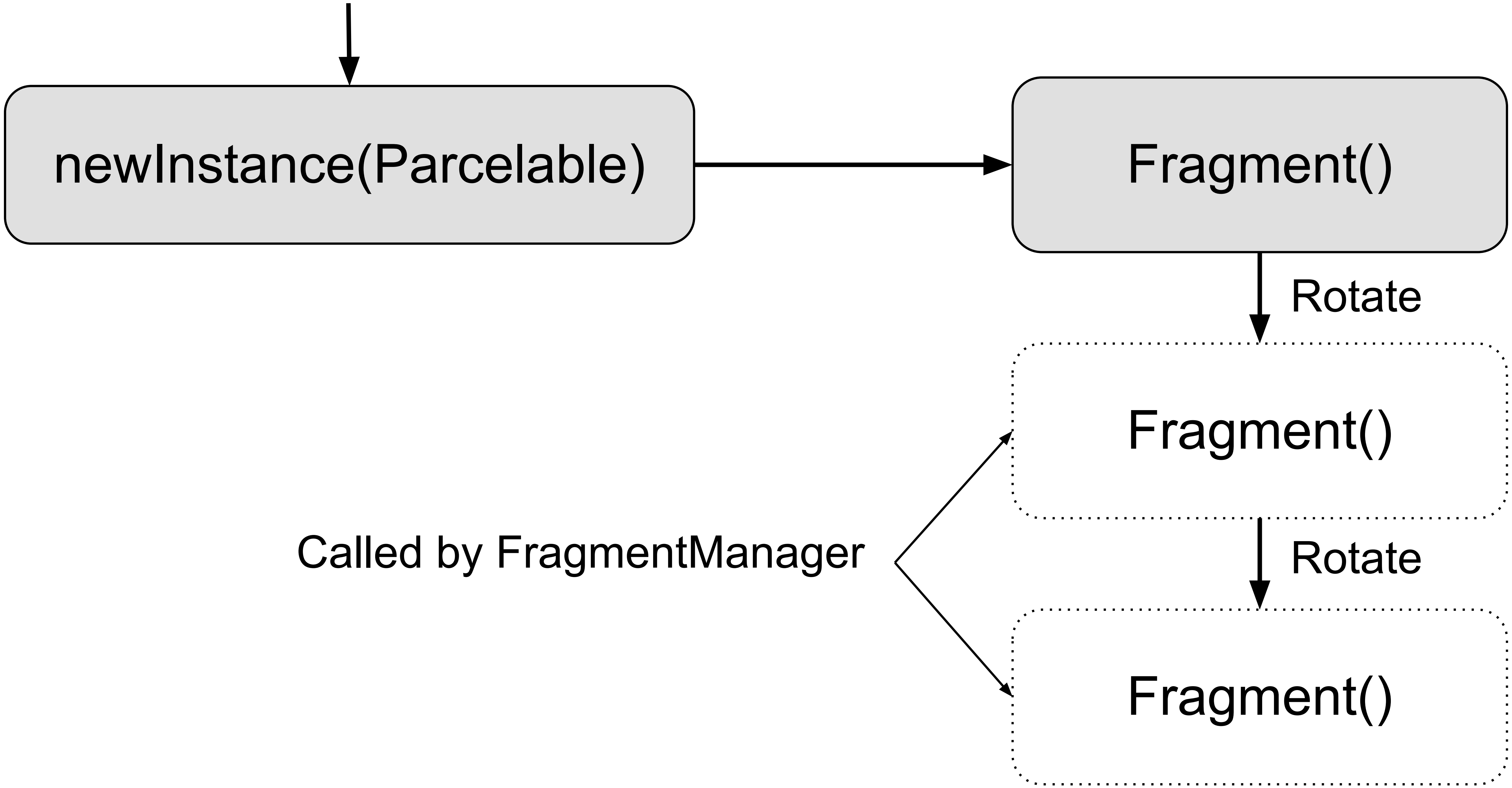
# Fragment techniques

```
class FolderList extends Fragment {
  /** Key to find our state in a bundle */
  private static String ARG_STATE = "arg-state";
  private Parcelable state;

  public FolderList() {
      // Do nothing.
      super();
  }

  public void onCreate(... ) {
      Bundle args = getArguments();
      state = args.getParcelable(ARG_STATE);
      ...
 }
```

```
/** The only constructor we ever call */
static FolderList newInstance(Parcelable state){
    Bundle b = new Bundle();
    b.putParcelable(ARG_STATE, state);
    FolderList f = new FolderList();
    f.setArguments(b);
    return f;
}
```



newInstance(Parcelable)

Fragment()

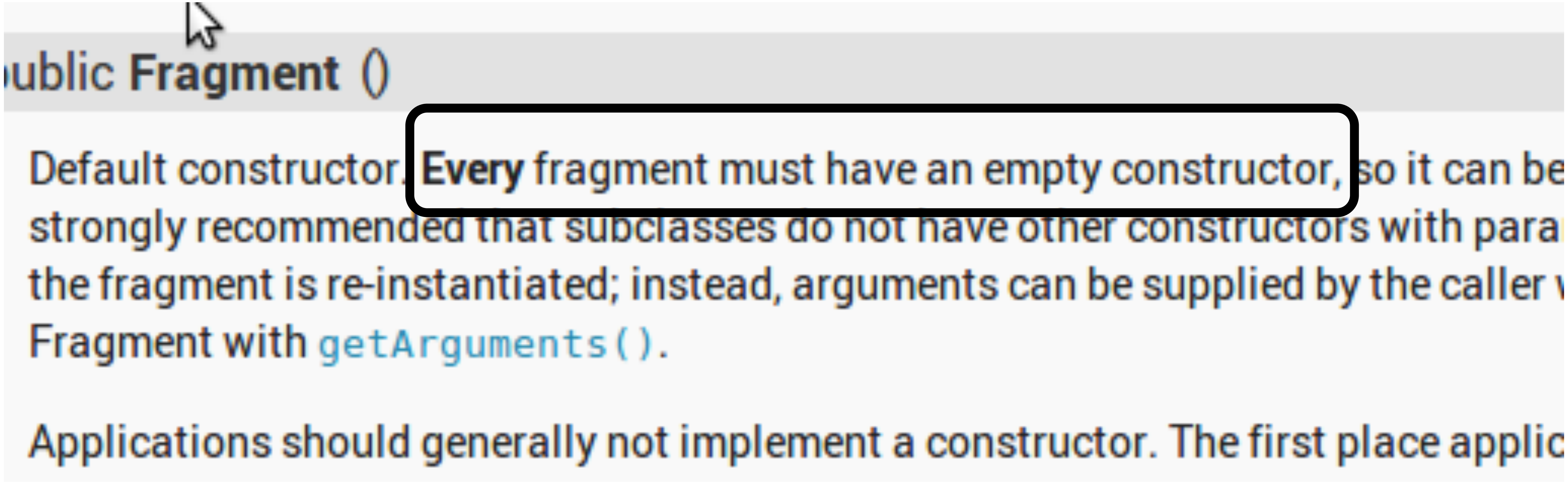
Rotate

Fragment()

Rotate

Fragment()

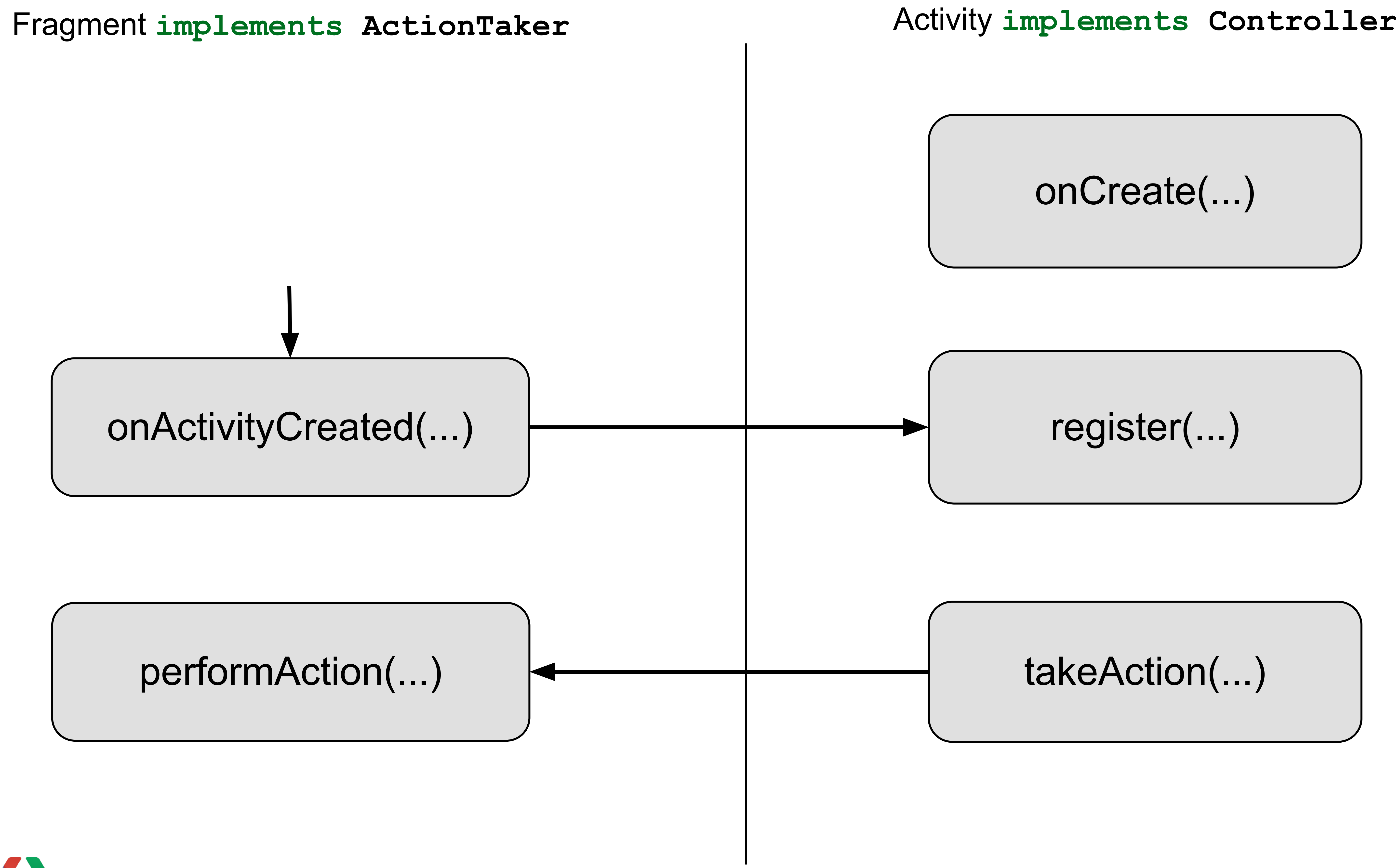
Called by FragmentManager



ublic **Fragment** ()

Default constructor. **Every** fragment must have an empty constructor, so it can be
strongly recommended that subclasses do not have other constructors with para
the fragment is re-instantiated; instead, arguments can be supplied by the caller
Fragment with `getArguments()`.

Applications should generally not implement a constructor. The first place applic



Fragment `implements` **`ActionTaker`**

Activity `implements` **`Controller`**

onCreate(...)

onActivityCreated(...)

register(...)

performAction(...)

takeAction(...)

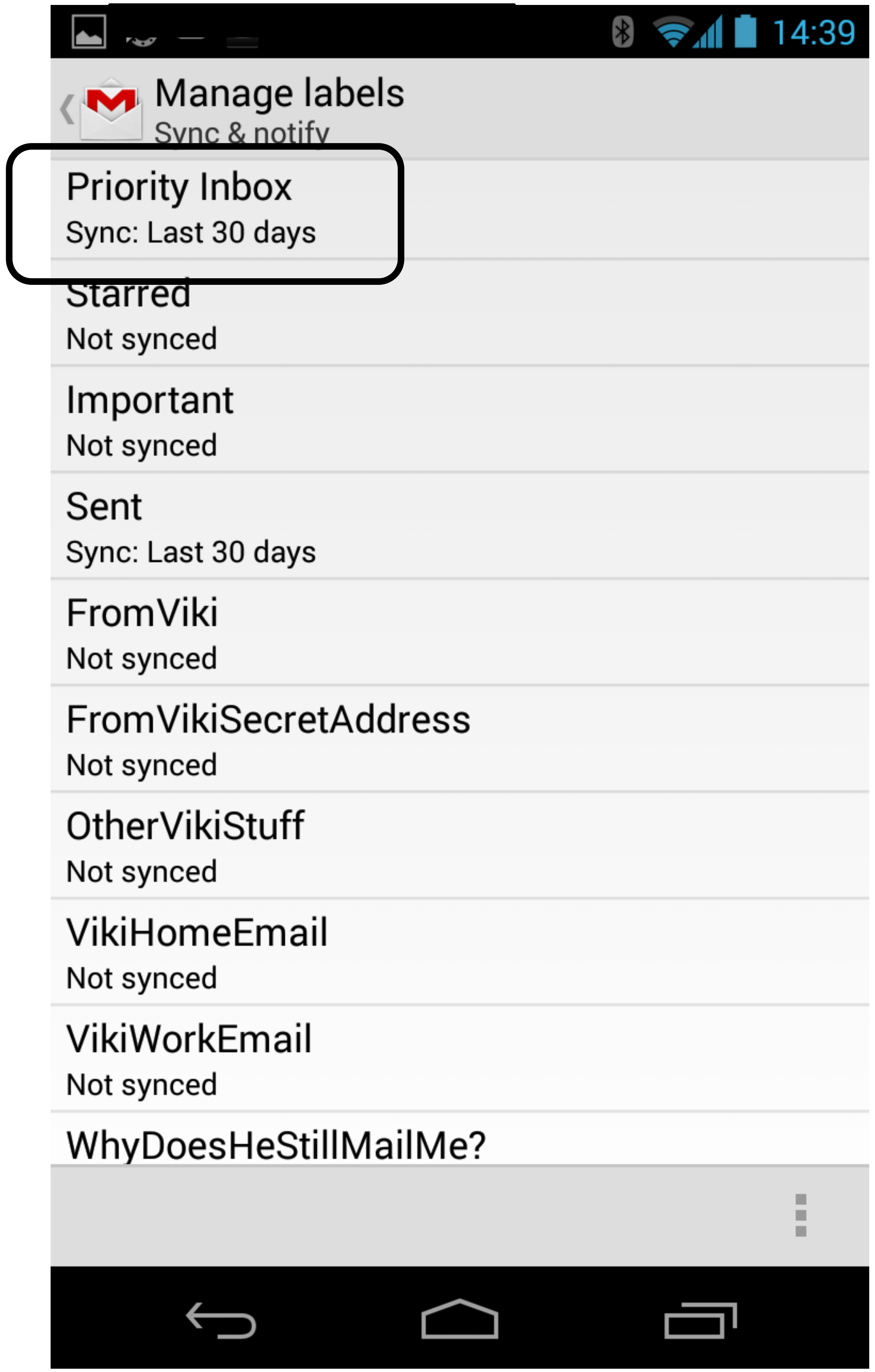
```
public class MyActivity extends Activity implements Controller {
    ActionTaker mDelegate;
   public void registerActionTaker(ActionTaker delegate) {
       mDelegate = delegate;
   }

   public void unregisterActionTaker() {
       mDelegate = null;
   }

   private void takeAction(Object information) {
       if (mDelegate != null) {
           mDelegate.performAction(information);
       }
  }
  ...
}
```

```java
public class ActionFragment extends Fragment implements ActionTaker {
    public void onActivityCreated(Bundle savedInstanceState) {
        final Controller controller = (Controller) getActivity();
        controller.registerActionTaker(this);
    }
    public void onDestroy() {
        final Controller controller = (Controller) getActivity();
        controller.unregisterActionTaker();
    }
    public void performAction(Object information) {
        // Perform the action
    }
....
}
```

# Do less, do it fast



Manage labels
Sync & notify

Priority Inbox
Sync: Last 30 days

Starred
Not synced

Important
Not synced

Sent
Sync: Last 30 days

FromViki
Not synced

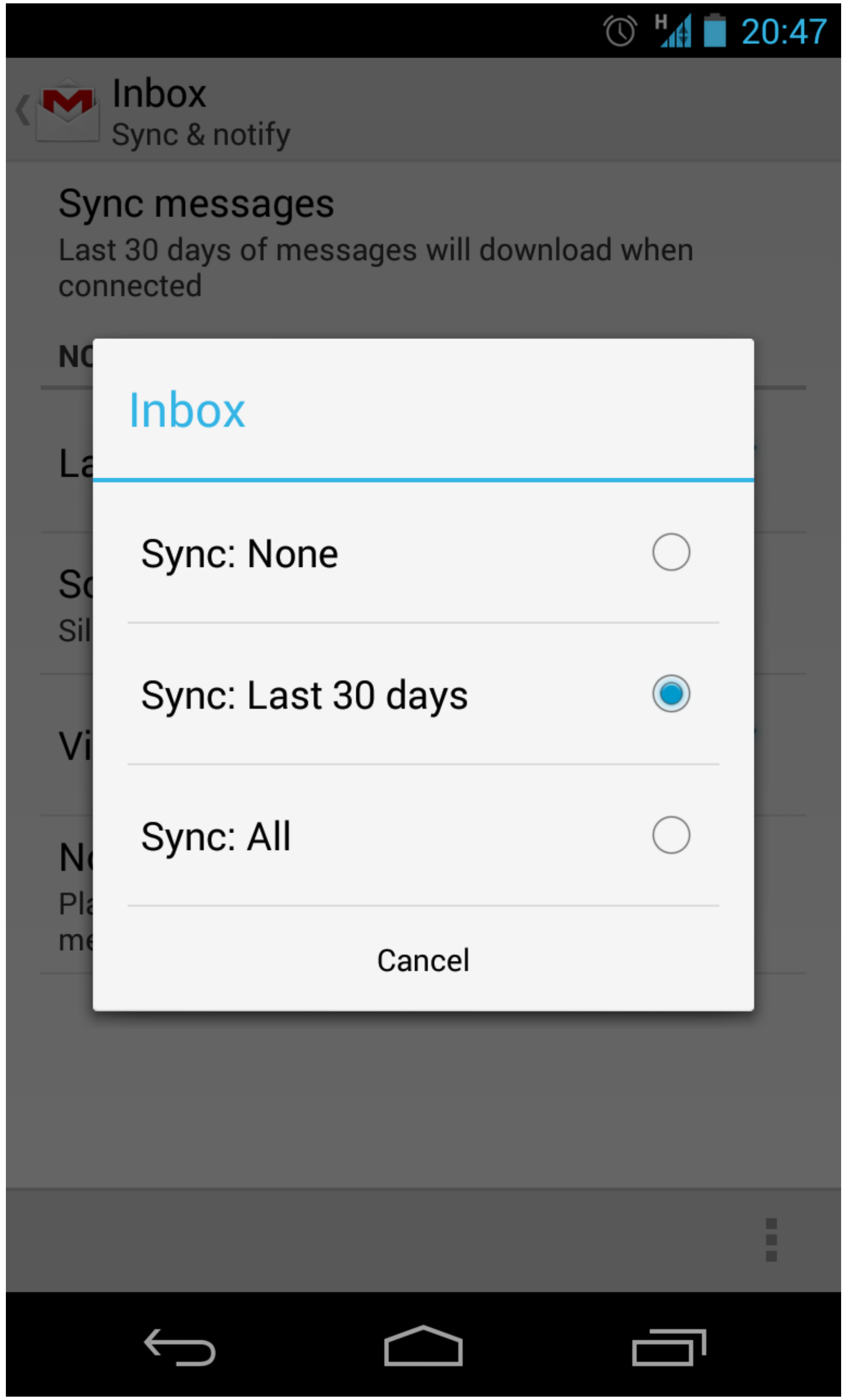
FromVikiSecretAddress
Not synced

OtherVikiStuff
Not synced

VikiHomeEmail
Not synced

VikiWorkEmail
Not synced

WhyDoesHeStillMailMe?



Inbox
Sync & notify

Sync messages
Last 30 days of messages will download when connected

Inbox

○ Sync: None

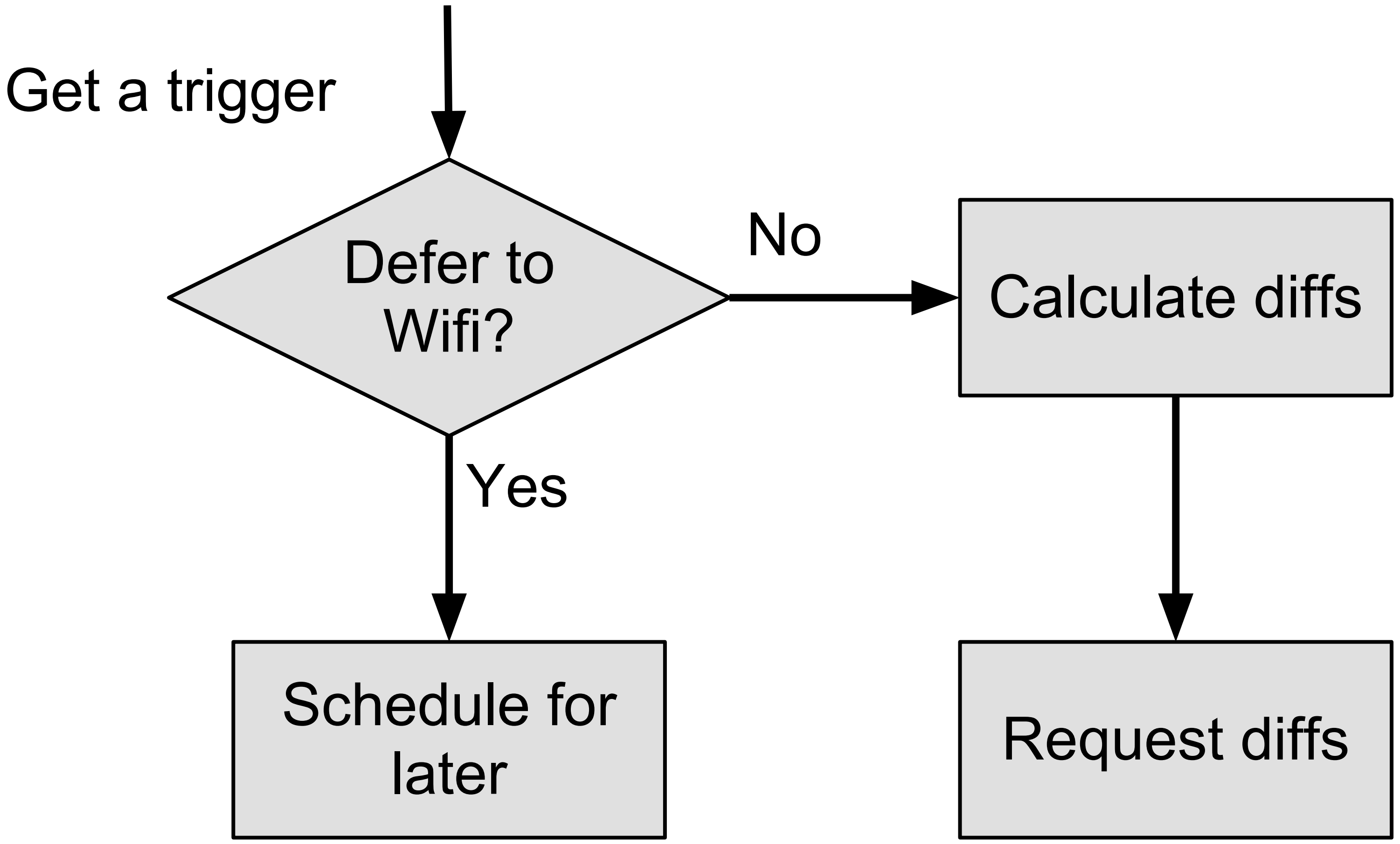
● Sync: Last 30 days

○ Sync: All

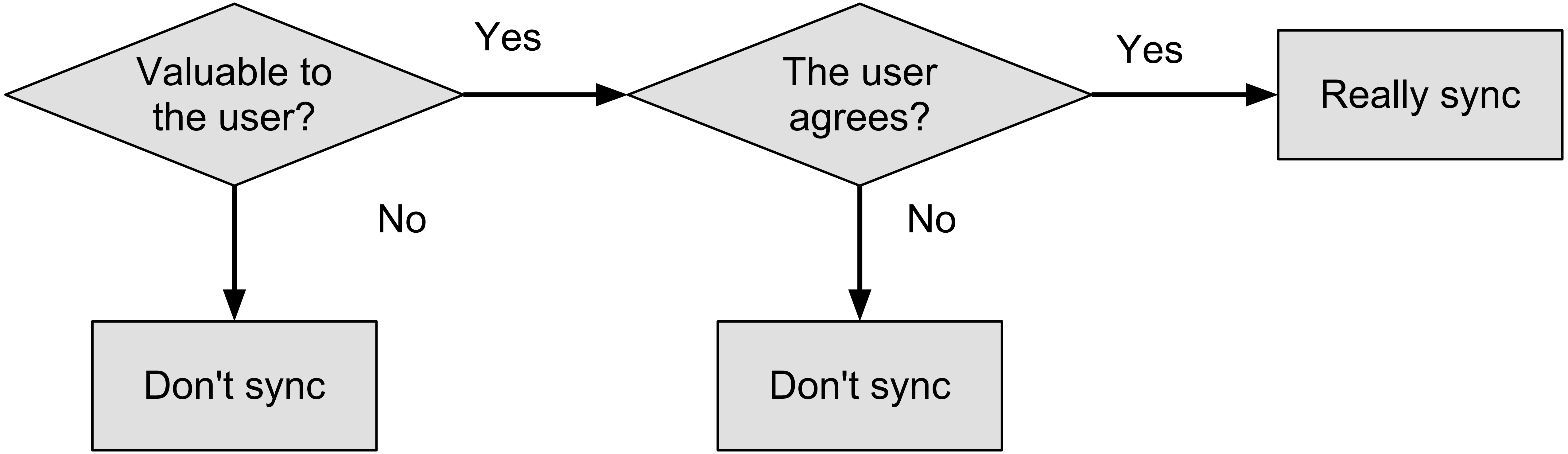
Cancel

```
GCMRegistrar.checkDevice(this);
GCMRegistrar.checkManifest(this);
String regId = GCMRegistrar.getRegistrationId(this);
if ("".equals(regId)) {
    GCMRegistrar.register(this, SENDER_ID);
}
// Use the name GCMIntentService!
public class GCMIntentService extends GCMBaseIntentService {
...
    @Override
    protected void onMessage(Context context, Intent intent) {
        requestSync();
    }
...
}
```

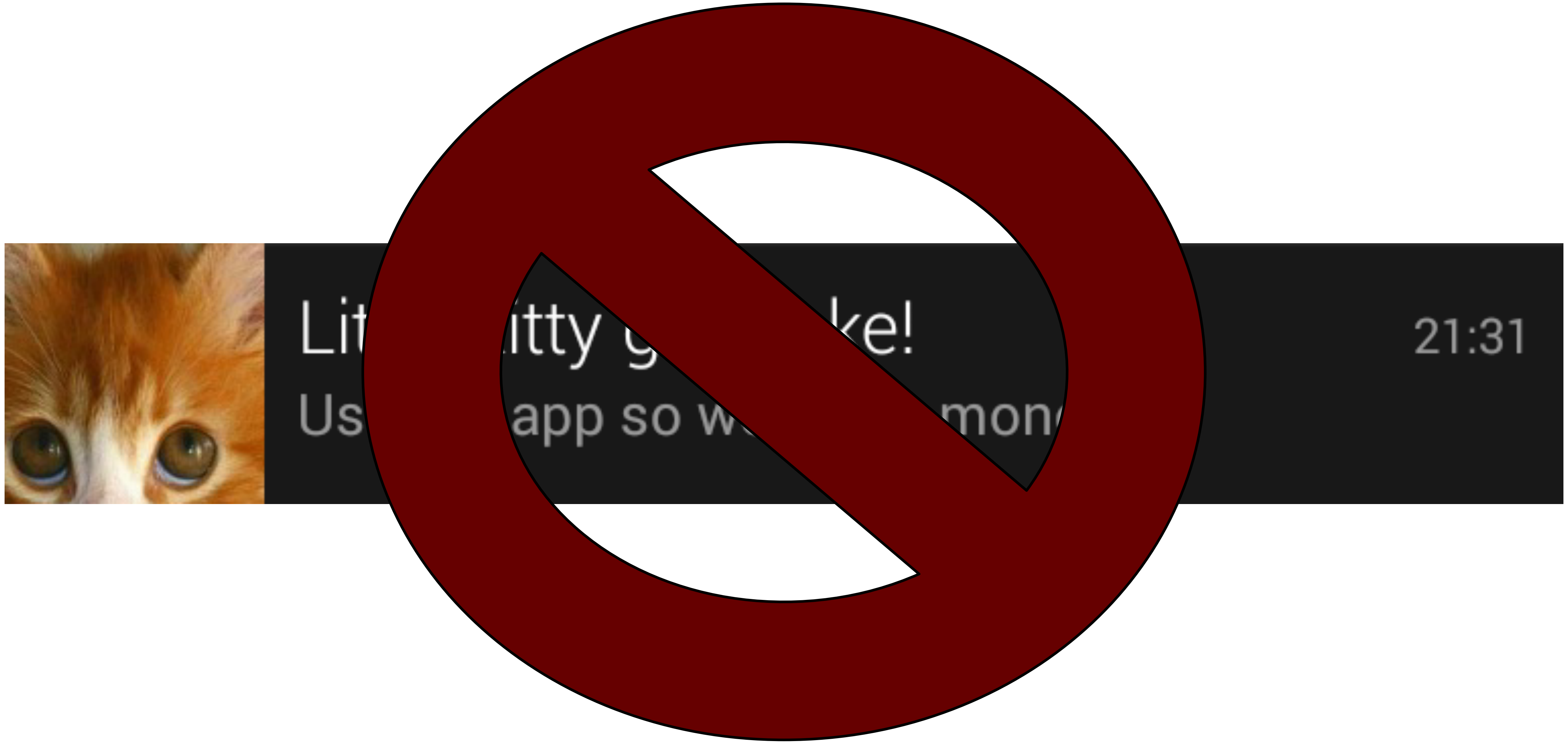
```
public class IOIntentService extends IntentService {
    @Override
    protected void onHandleIntent(final Intent intent) {
        if (ACTION_SYNC.equals(intent.getAction())) {
            requestSync();
        }
    }
}
```


Get a trigger
Defer to Wifi?
No
Calculate diffs
Yes
Schedule for later
Request diffs

# Only sync critical data



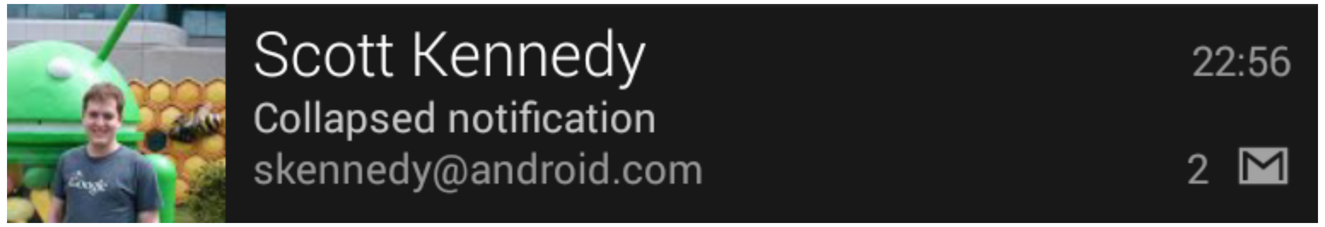
# Notifications



Lit itty g ke!

Us app so w mon

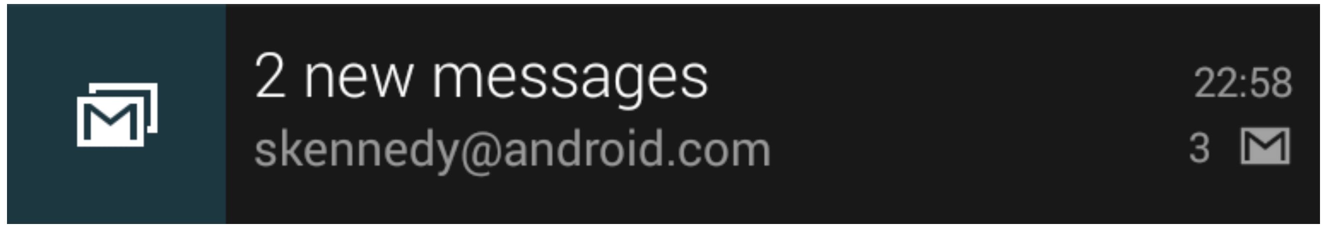
21:31



Scott Kennedy
Collapsed notification
skennedy@android.com

```
NotificationCompat.Builder b = new Builder(context);

b.setSmallIcon(notifIcon);
b.setLargeIcon(contactPhoto);
b.setContentTitle(contactName);
b.setContentText(subject);
b.setSubText(account);
b.setNumber(unreadMessageCount);
```



2 new messages

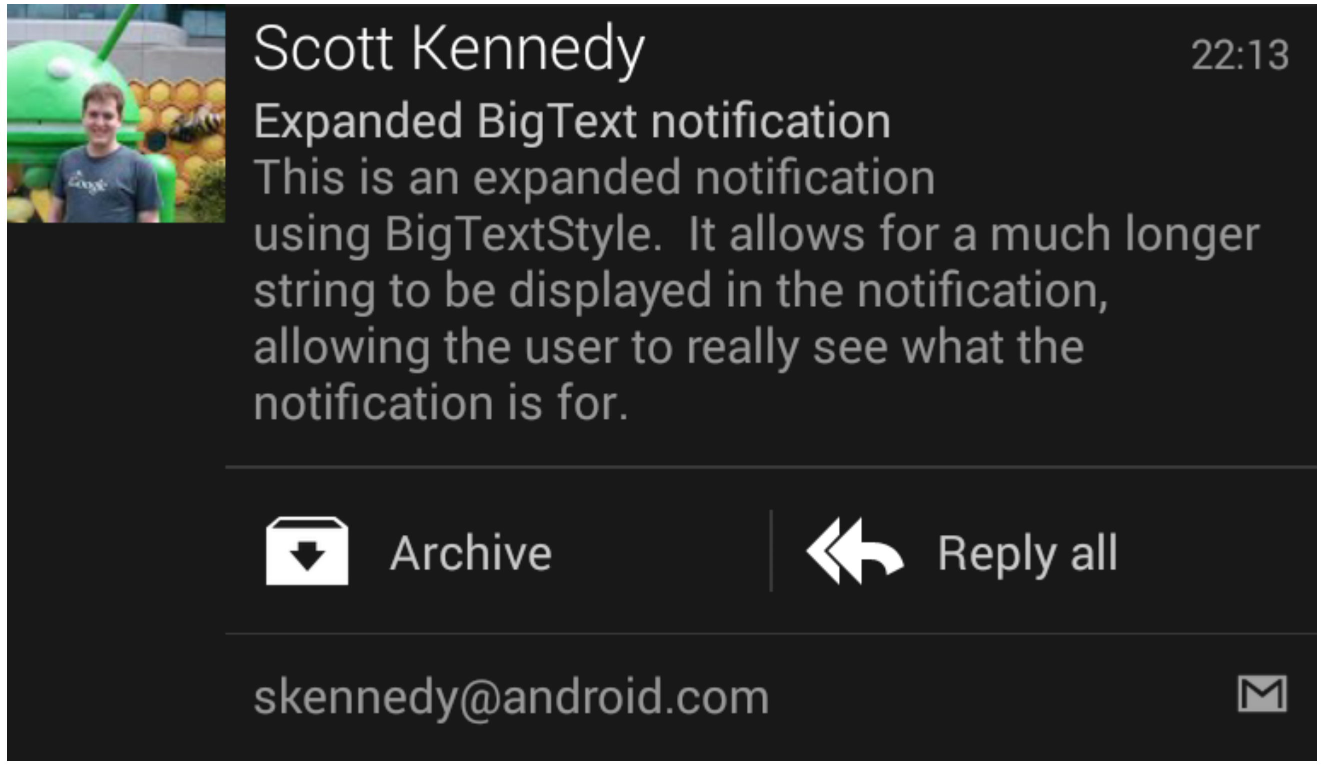
skennedy@android.com

22:58

3

```
NotificationCompat.Builder b = new Builder(context);

b.setSmallIcon(notifIcon);
b.setLargeIcon(multiMailIcon);
b.setContentTitle(notifTitle);
b.setSubText(account);
b.setNumber(unreadMessageCount);
```



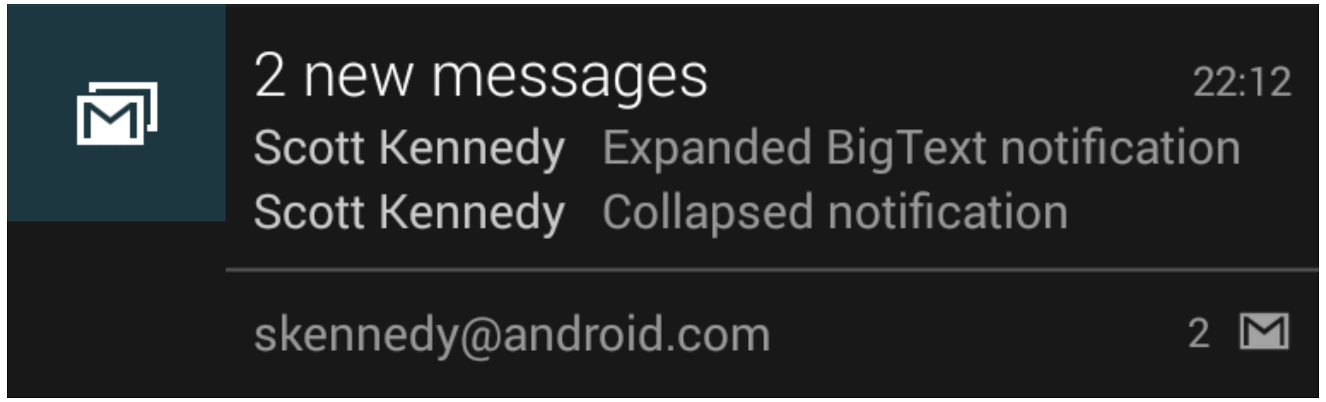
Scott Kennedy 22:13

Expanded BigText notification

This is an expanded notification using BigTextStyle. It allows for a much longer string to be displayed in the notification, allowing the user to really see what the notification is for.

skennedy@android.com

```
NotificationCompat.Builder b = new Builder(context);
...
BigTextStyle bts = new BigTextStyle(b);
bts.bigText(messageBody);
b.addAction(R.drawable.archive, "Archive", archivePI);
b.addAction(R.drawable.reply_all, "Reply All", replyAllPI);
```



2 new messages 22:12

Scott Kennedy Expanded BigText notification

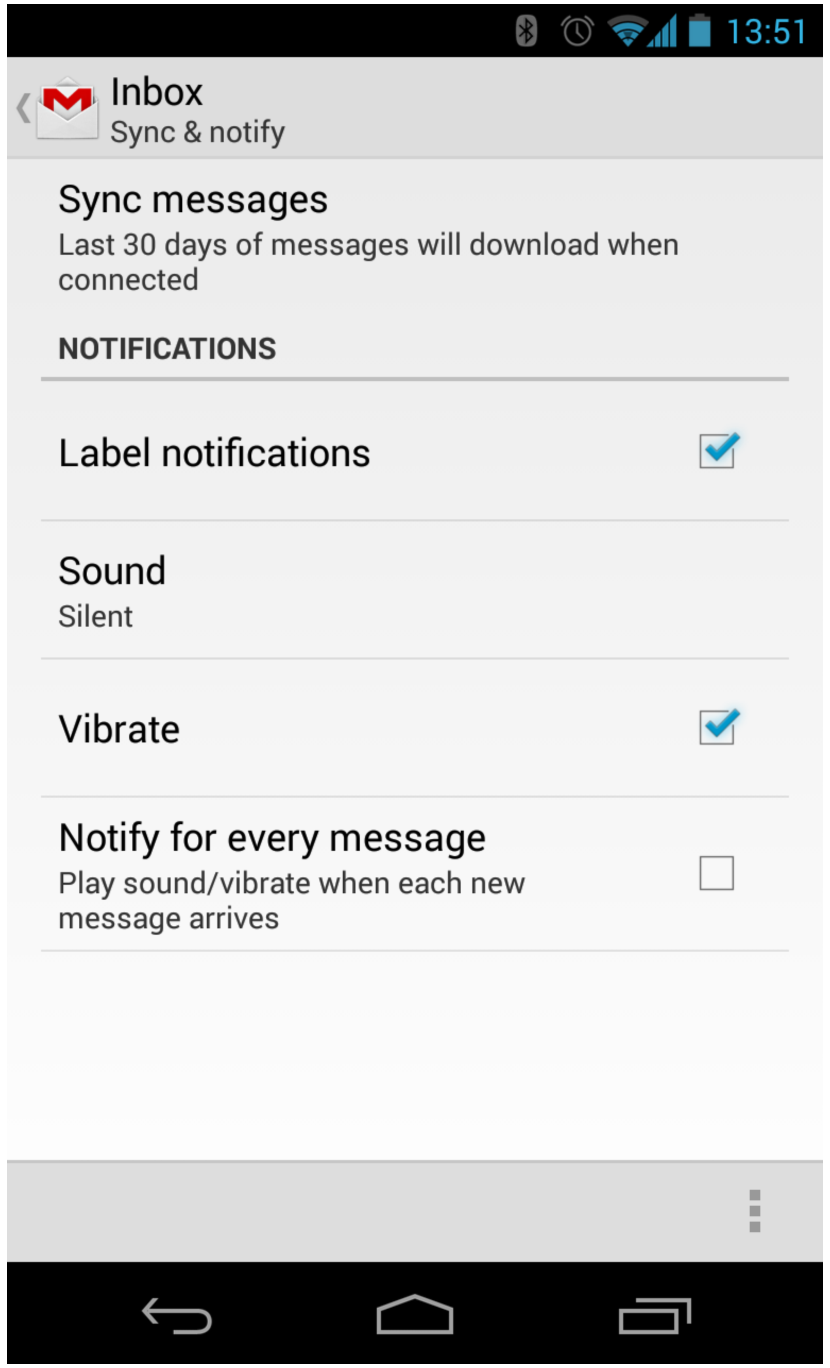
Scott Kennedy Collapsed notification

skennedy@android.com 2

```
NotificationCompat.Builder b = new Builder(context);
...
InboxStyle is = new InboxStyle(b);
is.addLine(msg1);
is.addLine(msg2);
```

# Notifications

- Don't enter the app
- Act on it right there
- One notification per event



Inbox
Sync & notify

Sync messages
Last 30 days of messages will download when connected

NOTIFICATIONS

Label notifications ☑

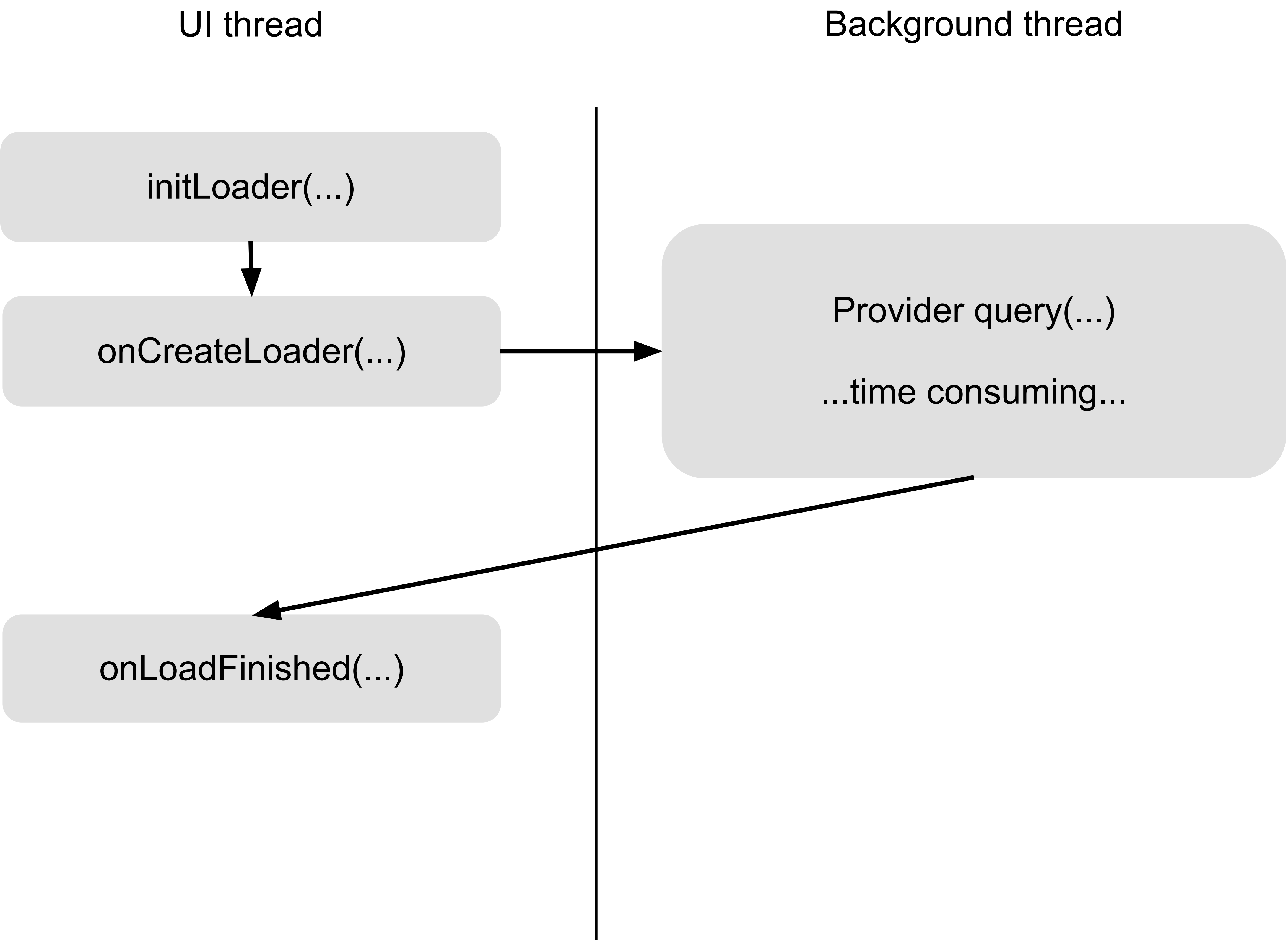
Sound
Silent

Vibrate ☑

Notify for every message
Play sound/vibrate when each new message arrives ☐

# Loaders

938G
[redacted]
[redacted]
27215
Keeping the Air Cleaner in Texas



UI thread

Background thread

initLoader(...)

onCreateLoader(...)

Provider query(...)

...time consuming...

onLoadFinished(...)

```
class FolderLoads implements LoaderManager.LoaderCallbacks<Cursor> {
    Loader<Cursor> onCreateLoader(int id, Bundle args) {
        if (id == FOLDER_LOADER) {
            Uri uri = Uri.parse(args.getString(TARGET));
            return new CursorLoader(authority, uri, ...);
        }
    }
    public void onLoadFinished(Loader<Cursor> loader, Cursor data) {
        if (loader.getId() == LOADER_FOLDER) {
            // Use the folder cursor here
        }
    }
    public void onLoaderReset(Loader<Cursor> loader) {
        if (loader.getId() == LOADER_FOLDER) {
            // Data gone. Do something sensible.
        }
    }
```

```
/** Loads {@link Folder} objects */
final FolderLoads mCallback = new FolderLoads();
final String TARGET = "target-uri";
final int LOADER_FOLDER = 42;

/** Start async data fetch */
void needData(Uri location) {
    final LoaderManager lm = getLoaderManager();
    final Bundle args = new Bundle();
    args.putString(TARGET, location);
    lm.initLoader(LOADER_FOLDER, args, mCallback);
}
```

# Making objects off the UI

Producer

```
public class ObjectCursor <T> extends CursorWrapper {
    /** Returns an object at the current cursor position. */
    public final T getModel() { .... }
    /** Populates the cache with objects. */
    final void fillCache() { .... }
}


class ObjectCursorLoader<T> extends AsyncTaskLoader<ObjectCursor<T>> {
    public ObjectCursor<T> loadInBackground() {
        final Cursor hidden = getContentResolver().query(....);
        ...
        final ObjectCursor<T> cursor = new ObjectCursor<T>(hidden, mFactory);
        cursor.fillCache(); // Create the objects here.
        return cursor;
    }
}
```

```
class HandleLoads implements LoaderManager.LoaderCallbacks<...> {
    public void onLoadFinished(..., ObjectCursor<T> data) {
        if (data != null && data.moveToFirst()) {
            final BigHeavy log = data.getModel();
            // Do something with log object.
        }
    }
...
}
```

Consumer

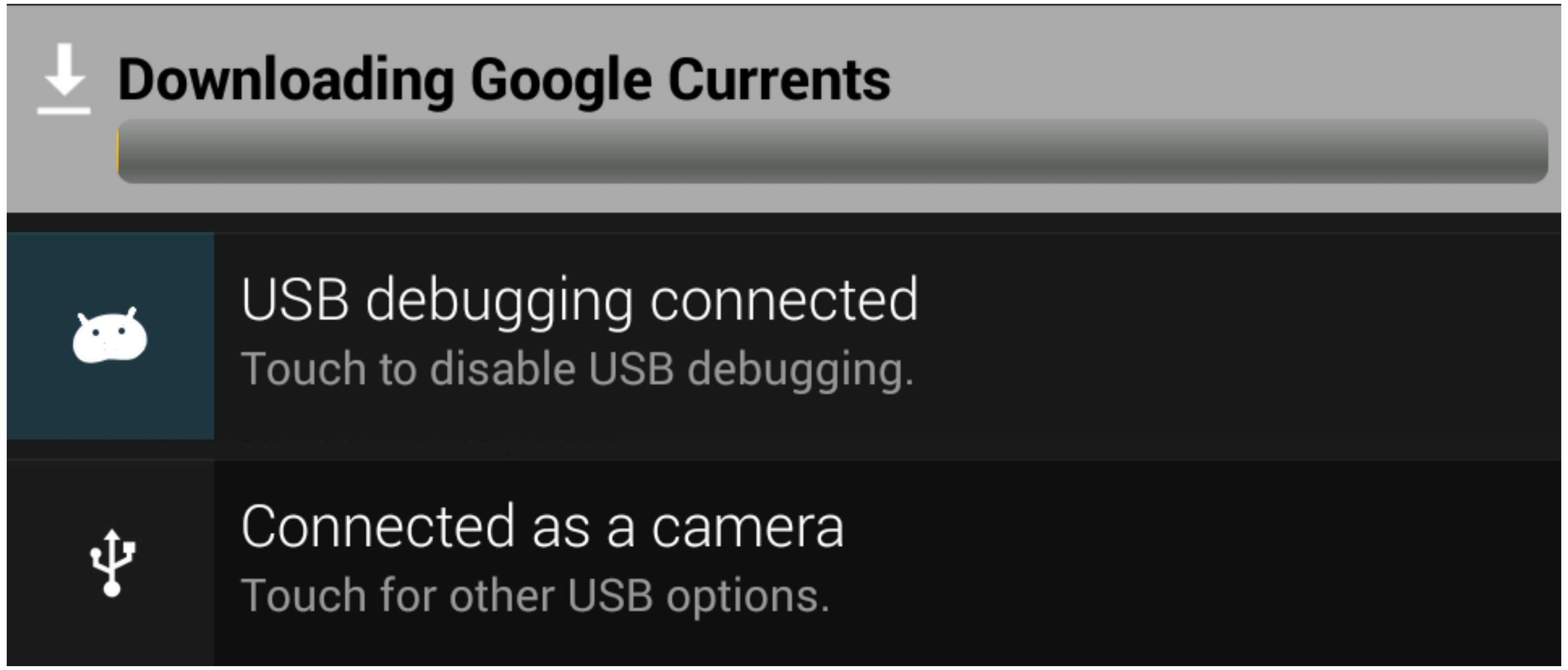
Consumer

```
/** Loads {@link BigHeavy} objects */
final HandleLoads mLoadCallback = new HandleLoads();

...
public boolean onCreate(Bundle savedState) {
    final Bundle args = new Bundle();
    args.putString(KEY_URI, uri);
    lm.initLoader(LOADER_BIG_HEAVY, args, mLoadCallback);
    ...
}
```

# Large file downloads

# DownloadManager



Downloading Google Currents

USB debugging connected

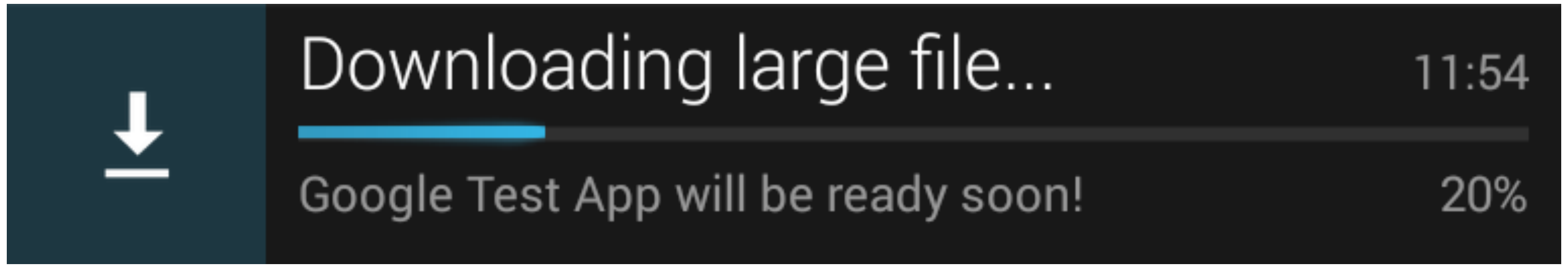
Touch to disable USB debugging.

Connected as a camera

Touch for other USB options.

```
getSystemService(DOWNLOAD_SERVICE).enqueue(new DownloadManager.Request(uri));
```

```
Request request = new Request(URI);
request.setAllowedNetworkTypes(Request.NETWORK_WIFI);
request.setAllowedOverMetered(false);
request.setAllowedOverRoaming(false);
request.setDestinationUri(destUri);
request.setTitle("Downloading large file...");
request.setDescription("Google Test App will be ready soon!");
request.setNotificationVisibility(
        Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
request.setVisibleInDownloadsUi(true);
downloadManager.enqueue(request);
```



Downloading large file...

11:54

Google Test App will be ready soon!

20%

# Cloud Backup, Restore

```
<manifest ... >
    ...
    <application android:label="MyApplication"
                 android:backupAgent="MyBackupAgent">
        <activity ... >
            ...
        </activity>
    </application>
</manifest>
```

```
public class MyBackupAgent extends BackupAgentHelper {
    private static final String PREFS_NAME = "mail_preferences";
    private static final String PREFS_BACKUP_KEY = "prefs";
    private static final String FILE_NAME = "app.state";
    private static final String FILE_BACKUP_KEY = "file";

    @Override
    public void onCreate() {
        SharedPreferencesBackupHelper prefsHelper =
            new SharedPreferencesBackupHelper(this, PREFS_NAME);
        addHelper(PREFS_BACKUP_KEY, prefsHelper);

        FileBackupHelper fileHelper = new FileBackupHelper(this, FILE_NAME);
        addHelper(FILE_BACKUP_KEY, fileHelper);
    }
}
```

```java
public void onBackup(ParcelFileDescriptor oldState, BackupDataOutput data,
            ParcelFileDescriptor newState) throws IOException {
    ...
    ByteArrayOutputStream bufStream = new ByteArrayOutputStream();
    DataOutputStream outWriter = new DataOutputStream(bufStream);

    SenderInfo senderInfo = getSenderInfo();
    outWriter.writeUTF(senderInfo.name);
    outWriter.writeUTF(senderInfo.signature);

    byte[] buffer = bufStream.toByteArray();
    int len = buffer.length;
    data.writeEntityHeader(BACKUP_KEY, len);
    data.writeEntityData(buffer, len);
}
```

# Support Library

## Before

```
import android.app.Activity;
import android.app.Fragment;
import android.app.Notification;

getFragmentManager()
```

→

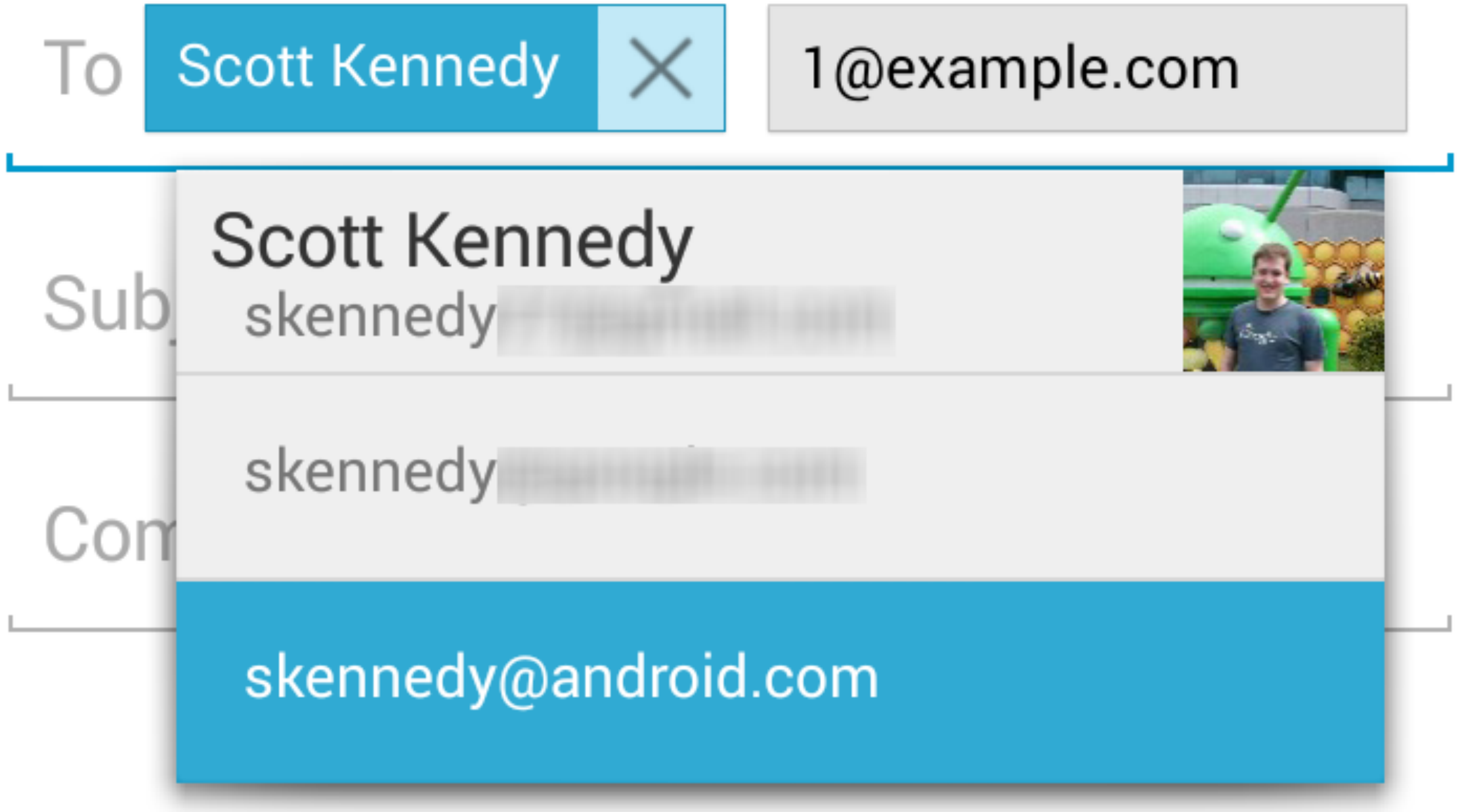
## After

```
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.Fragment;
import android.support.v4.app.NotificationCompat;

getSupportFragmentManager()
```

# Read (some) source

# Chips



To Scott Kennedy × 1@example.com

Sub

Com

Scott Kennedy
skennedy[redacted]

skennedy[redacted]

skennedy@android.com

## Chips

```
BaseRecipientAdapter adapter =
  new BaseRecipientAdapter(context) {};
RecipientEditTextView retv =
  new RecipientEditTextView(context,
    null);
retv.setTokenizer(
  new Rfc822Tokenizer());
retv.setAdapter(adapter);
retv.append("skennedy@android.com");
retv.append("1@example.com");
```

```
Rfc822Token[] tokens =
  Rfc822Tokenizer.tokenize(
  retv.getText());
int count = tokens.length;
String[] result = new String[count];
for (int i = 0; i < count; i++) {
    result[i] = tokens[i].toString();
}
```

**result is now:**

```
  "<skennedy@android.com>",
  "<1@example.com>"
```

# PhotoViewer

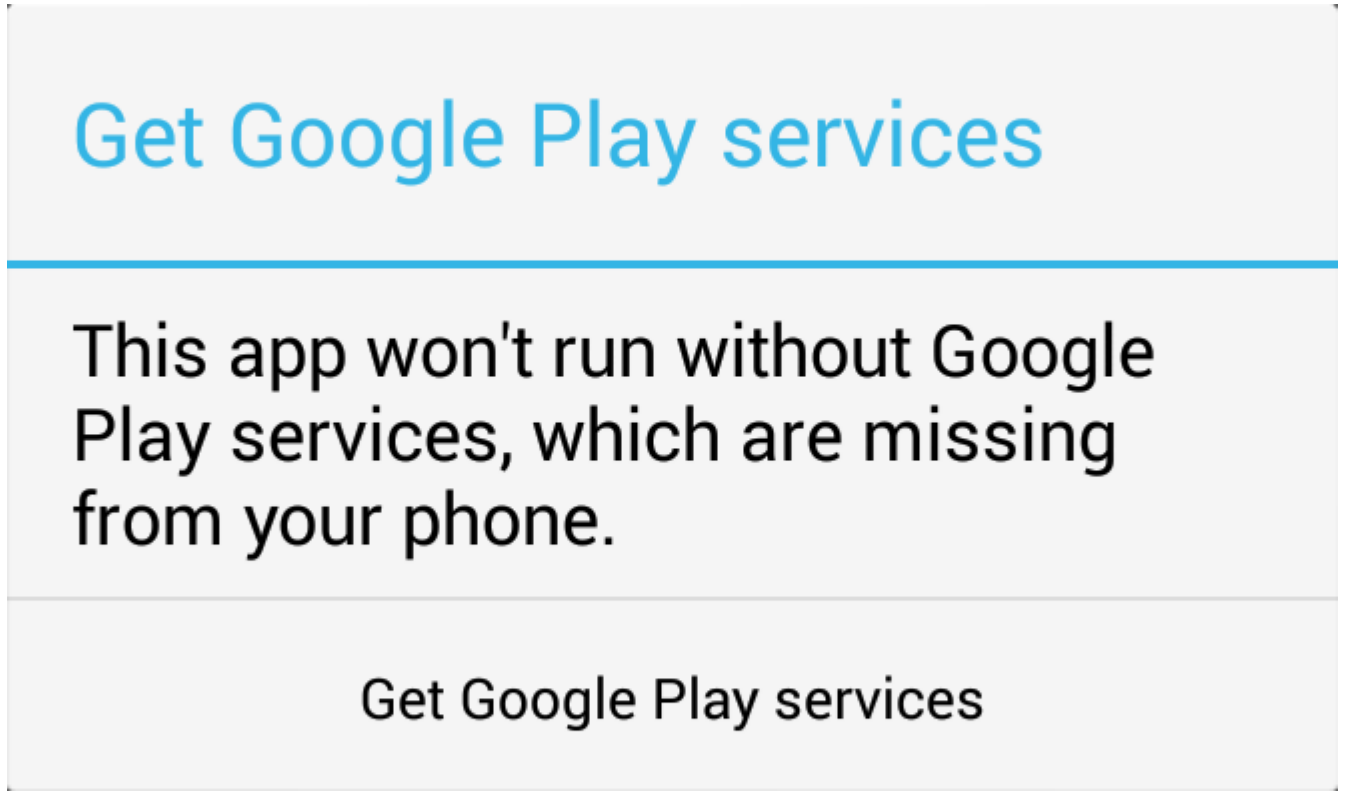
```java
PhotoViewIntentBuilder builder =
        Intents.newPhotoViewActivityIntentBuilder(this);
builder.setPhotosUri(
        "content://com.example.photoviewersample.SampleProvider/photos");
startActivity(builder.build());
```

# PhotoViewer

```java
public final static String[] PROJECTION = {
    PhotoViewColumns.URI,
    PhotoViewColumns.NAME,
    PhotoViewColumns.CONTENT_URI,
    PhotoViewColumns.THUMBNAIL_URI,
    PhotoViewColumns.CONTENT_TYPE,
};
```

# Google Play services

```
int result = GooglePlayServicesUtil.isGooglePlayServicesAvailable(context);
if (result != ConnectionResult.SUCCESS) {
    GooglePlayServicesUtil.getErrorDialog(result, activity,
        REQUEST_CODE).show();
}
```



Get Google Play services

This app won't run without Google Play services, which are missing from your phone.

## Photo Sphere

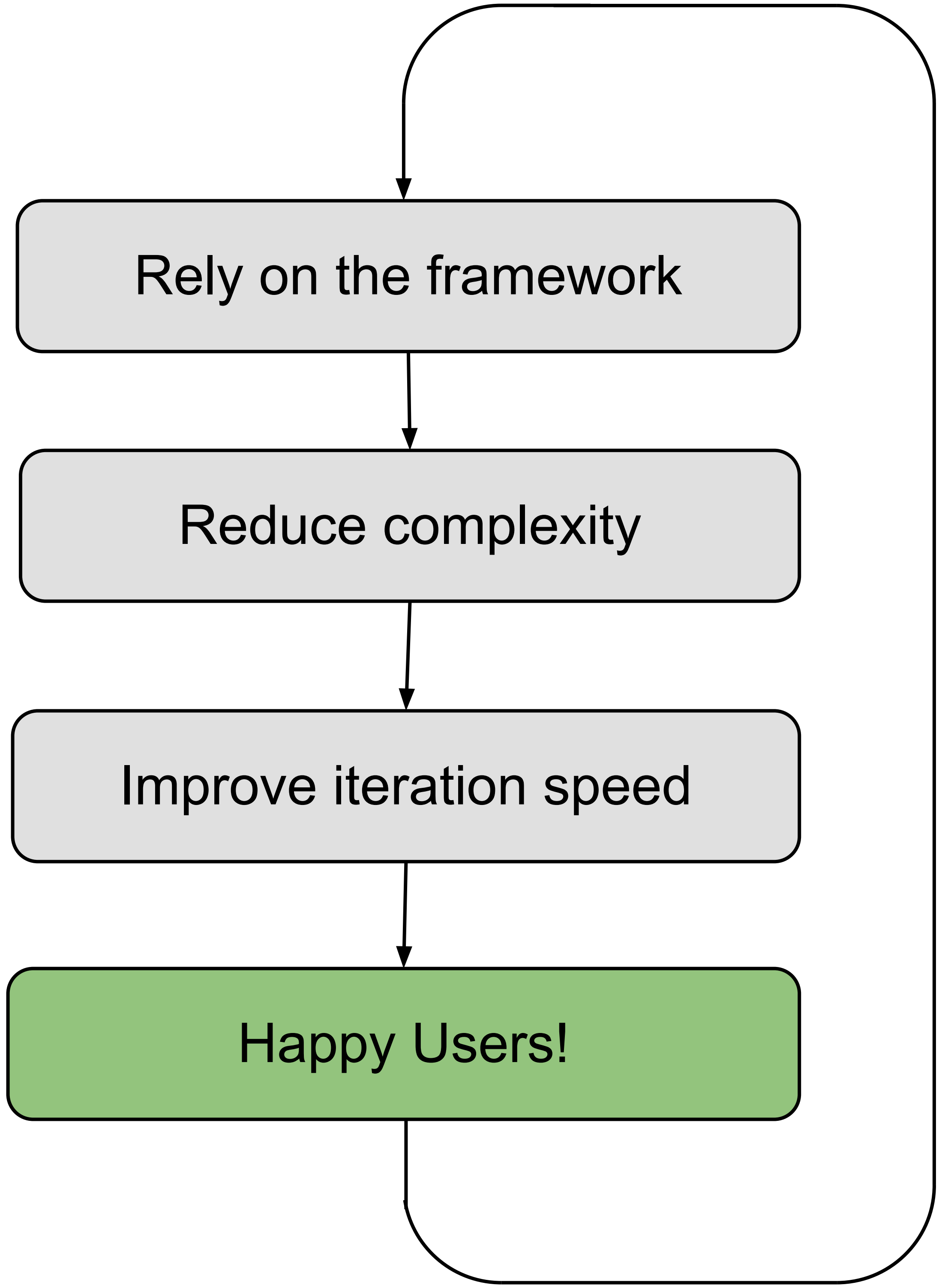
```
PanoramaClient client = new PanoramaClient(context, callbacks, listener);
...
client.loadPanoramaInfoAndGrantAccess(new OnPanoramaInfoLoadedListener() {
  @Override
  public void onPanoramaInfoLoaded(
           ConnectionResult result, Intent viewerIntent) {
    if (result.isSuccess() && viewerIntent != null) {
      context.startActivity(viewerIntent);
    }
  }
}, uri);
```

## Code

http://goo.gl/**PJ1Vj**

# Rely on the framework for better apps

- Fragments for reusable UI components
- Fragment Activity coupling
- Loaders for obtaining data
- Notifications: time critical information
- Download Manager:
- Cloud Backup: sync common settings
- Google Cloud Messaging: know when data is available
- Support Library: seamless functionality on older devices
- Photoviewer in AOSP
- Chips in AOSP
- Google Play Services:



!

Use the framework

# Fit the system well

Do less

Make your users happy

your users... thank you!

Scott skennedy@android.com

Viki viki@android.com

# We'd love your feedback!

Room 12

## License, reuse, rights

- Loader image file rights allow reuse.
- The kitten is free for use too!