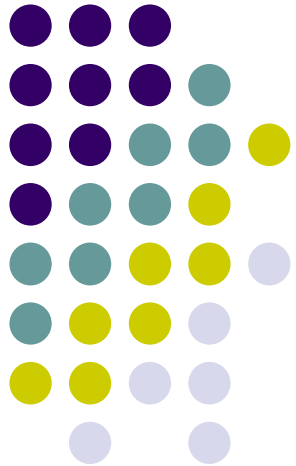
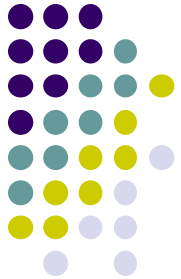
CS 528 Mobile and Ubicomp

Lecture 3a: Data-Driven Layouts & Android Components

Emmanuel Agu





Data-Driven Layouts

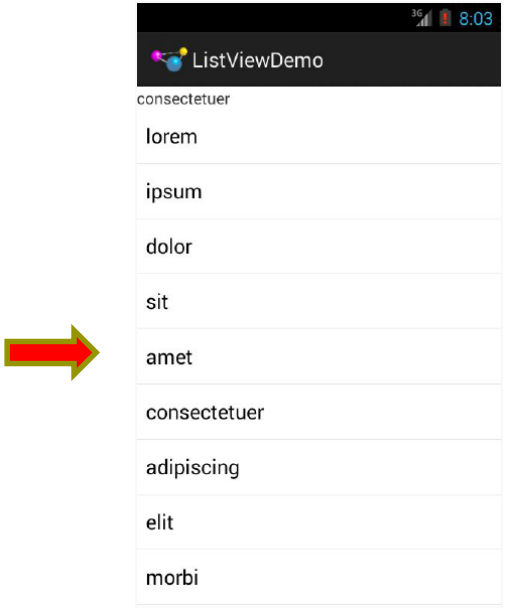
Data-Driven Layouts

- LinearLayout, RelativeLayout, TableLayout, GridLayout useful for positioning UI elements
  - UI data is **hard coded**

- Other layouts dynamically composed from data (e.g. database)
  - ListView, GridView, GalleryView
  - Tabs with TabHost, TabControl

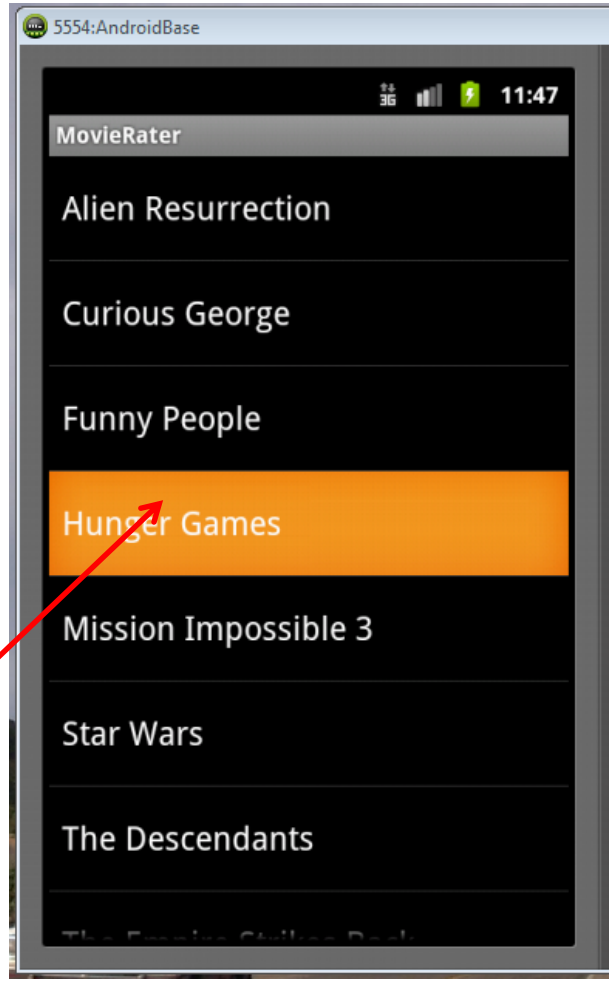
Generate widgets from data source

lorem
ipsum
dolor
amet
consectetuer
adipiscing
elit
morbi



Data Driven Layouts

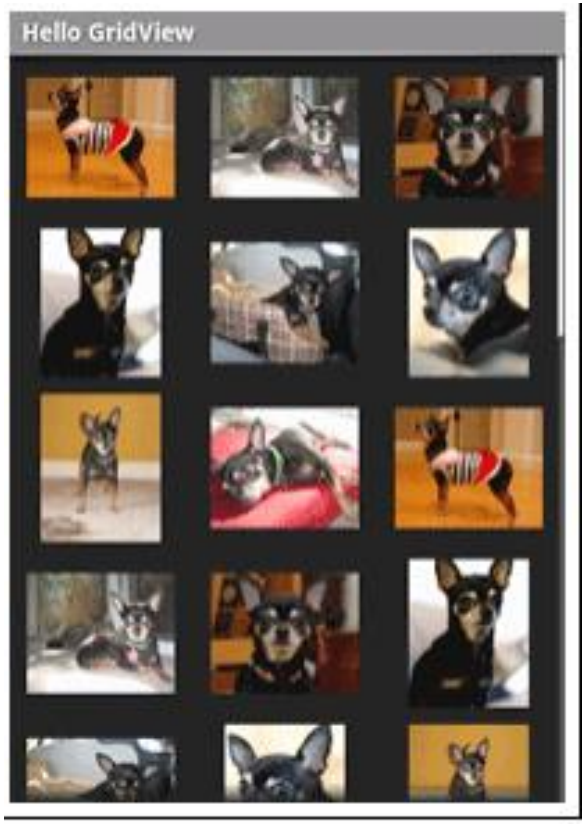
- May want to populate views from a data source (XML file or database)
- Layouts that display repetitive child widgets from data source
  - ListView
  - GridView
  - GalleryView
- ListView
  - Rows of entries, pick item, vertical scroll



Data Driven Containers

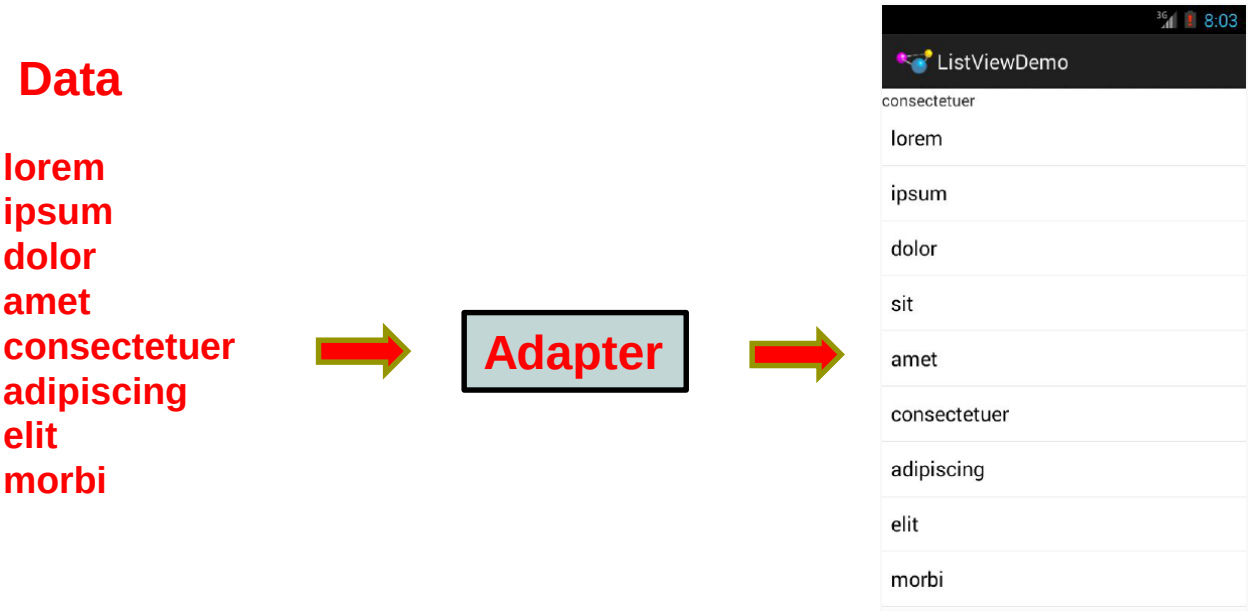
- GridView
  - List of items arranged in rows and columns

- GalleryView
  - List with horizontal scrolling, typically images



AdapterView

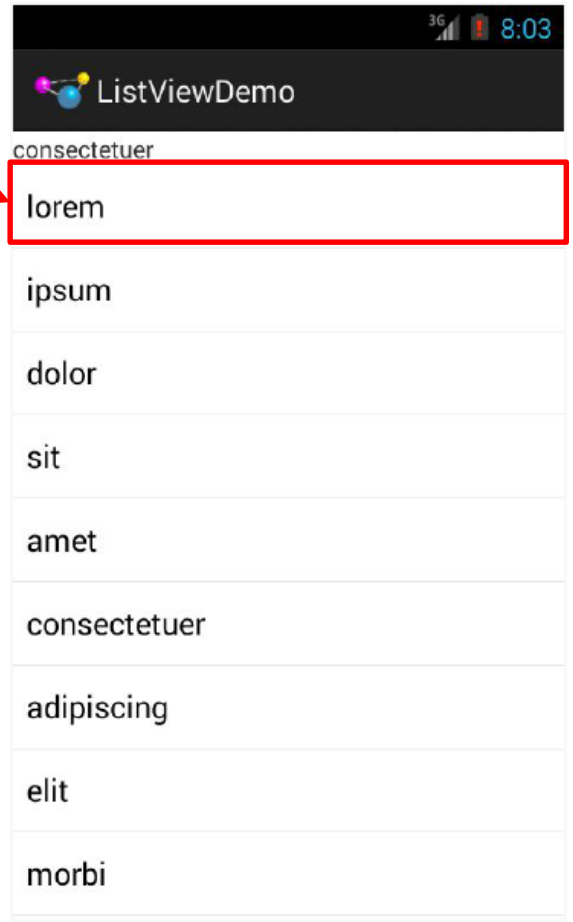
- ListView, GridView, and GalleryView are sub classes of AdapterView (variants)
- **Adapter:** generates widgets from a data source, populates layout
  - E.g. Data is adapted into cells of GridView



- Most common Adapter types:
  - **CursorAdapter:** read from database
  - **ArrayAdapter:** read from resource (e.g. XML file)

Adapters

- When using Adapter, a layout (XML format) is defined for each child element (View)



- The adapter
  - Reads in data (list of items)
  - Creates Views (widgets) using layout for each element in data source
  - Fills the containing layout (List, Grid, Gallery) with the created Views

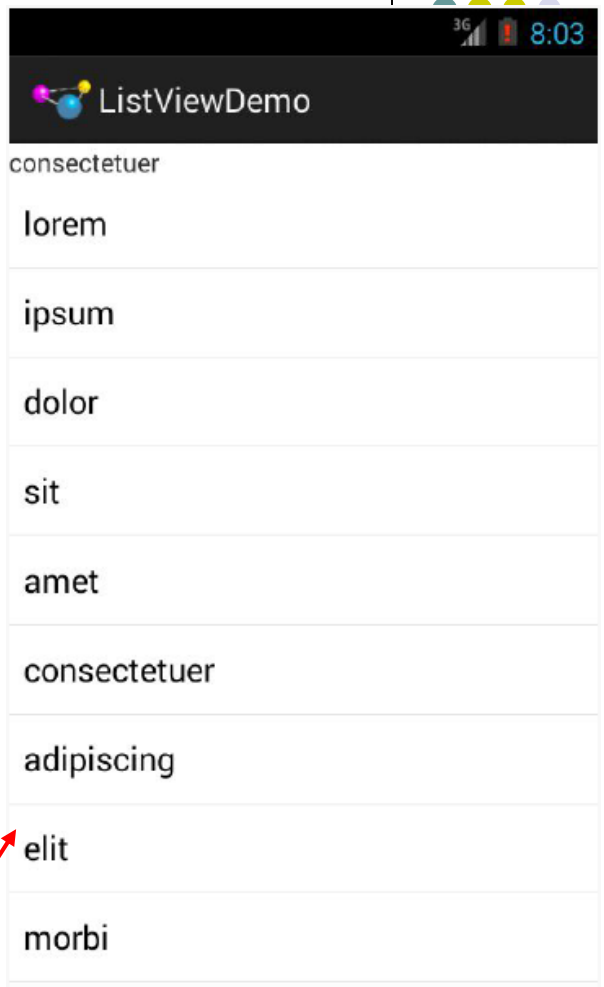
- Child widgets can be as simple as a TextView or more complex layouts / controls
  - simple views can be declared in a layout XML file (e.g. android.R.layout)

Example: Creating ListView using AdapterArray

- **Task:** Create listView (on right) from strings below

```
private static final String[] items={"lorem", "ipsum", "dolor",
        "sit", "amet",
        "consectetuer", "adipiscing", "elit", "morbi", "vel",
        "ligula", "vitae", "arcu", "aliquet", "mollis",
        "etiam", "vel", "erat", "placerat", "ante",
        "porttitor", "sodales", "pellentesque", "augue", "purus"};
```

Enumerated list



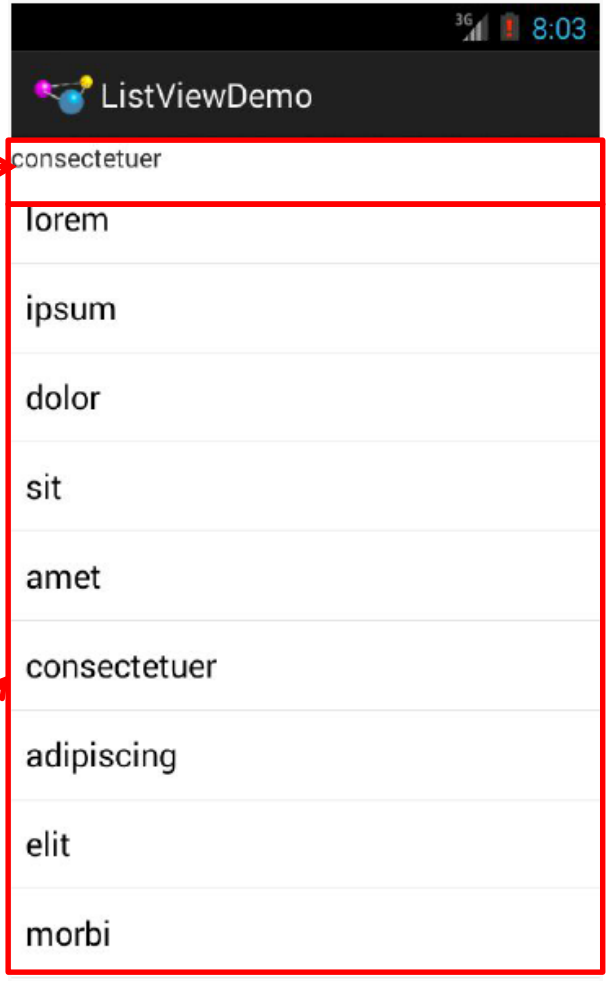
ListView of items

Example: Creating ListView using AdapterArray

- First create Layout file (e.g. LinearLayout)

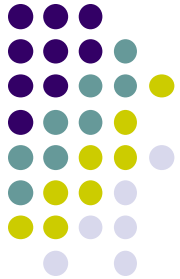
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.co
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
  <TextView
    android:id="@+id/selection"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
  <ListView
    android:id="@android:id/list"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />
</LinearLayout>
```

TextView Widget for selected list item



ListView for list of options

Using ArrayAdapter



- Command used to wrap adapter around array of menu items or **java.util.List** instance

```
String[] items={"this", "is", "a", "really", "silly", "list"};
new ArrayAdapter<String>(this,
                         android.R.layout.simple_list_item_1,
                         items);
```

Context to use.
(e.g app's activity)

Array of items to display

Resource ID of View for formatting

- E.g. **android.R.layout.simple_list_item_1** turns strings into textView widgets

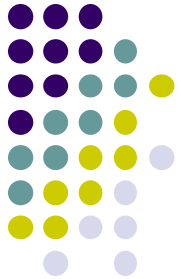
Example: Creating ListView using AdapterArray

```java
package com.commonsware.android.list;

import android.app.ListActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
public class ListViewDemo extends ListActivity {
  private TextView selection;
  private static final String[] items={"lorem", "ipsum", "dolor",
          "sit", "amet",
          "consectetuer", "adipiscing", "elit", "morbi", "vel",
          "ligula", "vitae", "arcu", "aliquet", "mollis",
          "etiam", "vel", "erat", "placerat", "ante",
          "porttitor", "sodales", "pellentesque", "augue", "purus"};

  @Override
  public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);
    setListAdapter(new ArrayAdapter<String>(this,
                        android.R.layout.simple_list_item_1,
                        items));
    selection=(TextView)findViewById(R.id.selection);
  }

  @Override
  public void onListItemClick(ListView parent, View v, int position,
                              long id) {
    selection.setText(items[position]);
  }
}
```

Set list adapter (Bridge Data source and views)

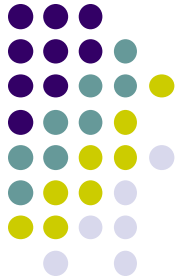
Get handle to TextView of Selected item

Change Text at top to that of selected view when user clicks on selection



Android App Components

Android App Components



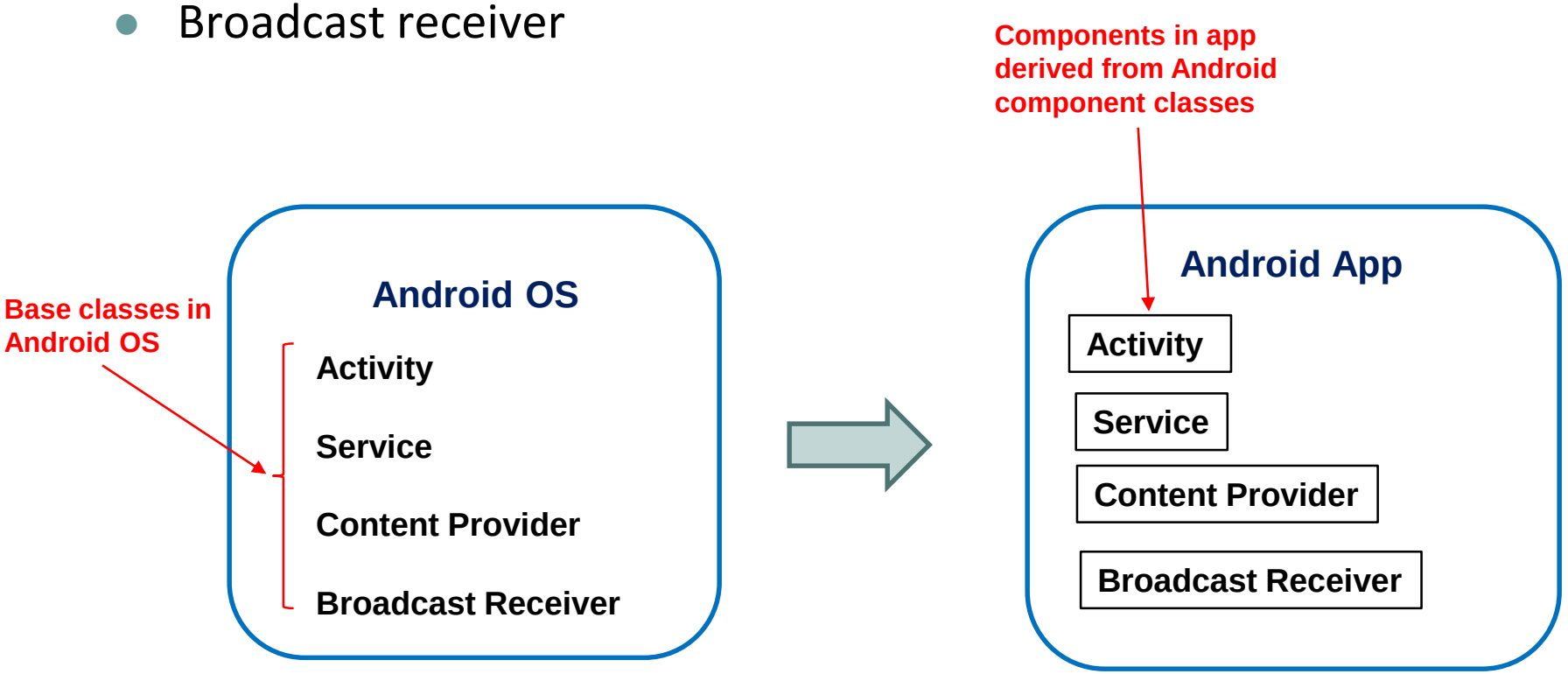
- Typical Java program starts from main( )

```
class SillyApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

- Android app: No need to write a main
- Just define app components derived from base classes already defined in Android

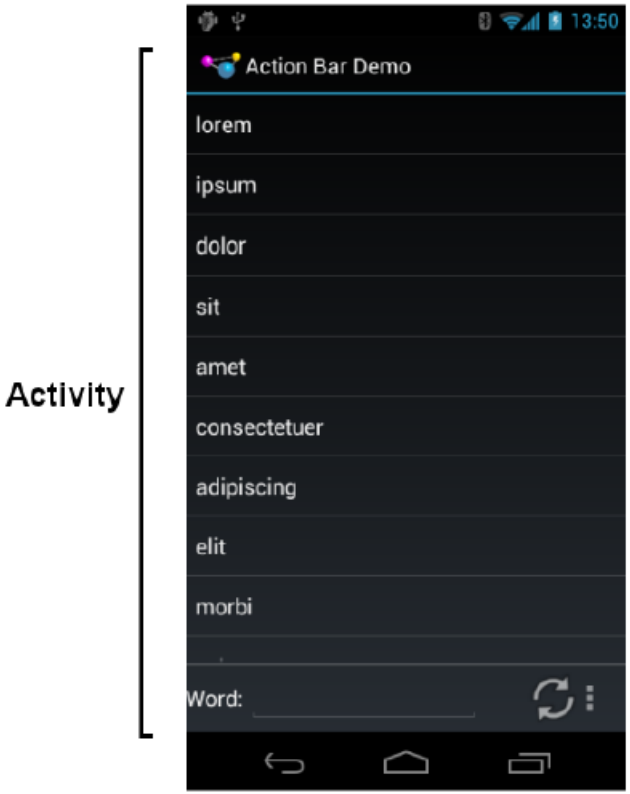
Android App Components

- 4 main types of Android app components:
  - Activity (already seen this)
  - Service
  - Content provider
  - Broadcast receiver



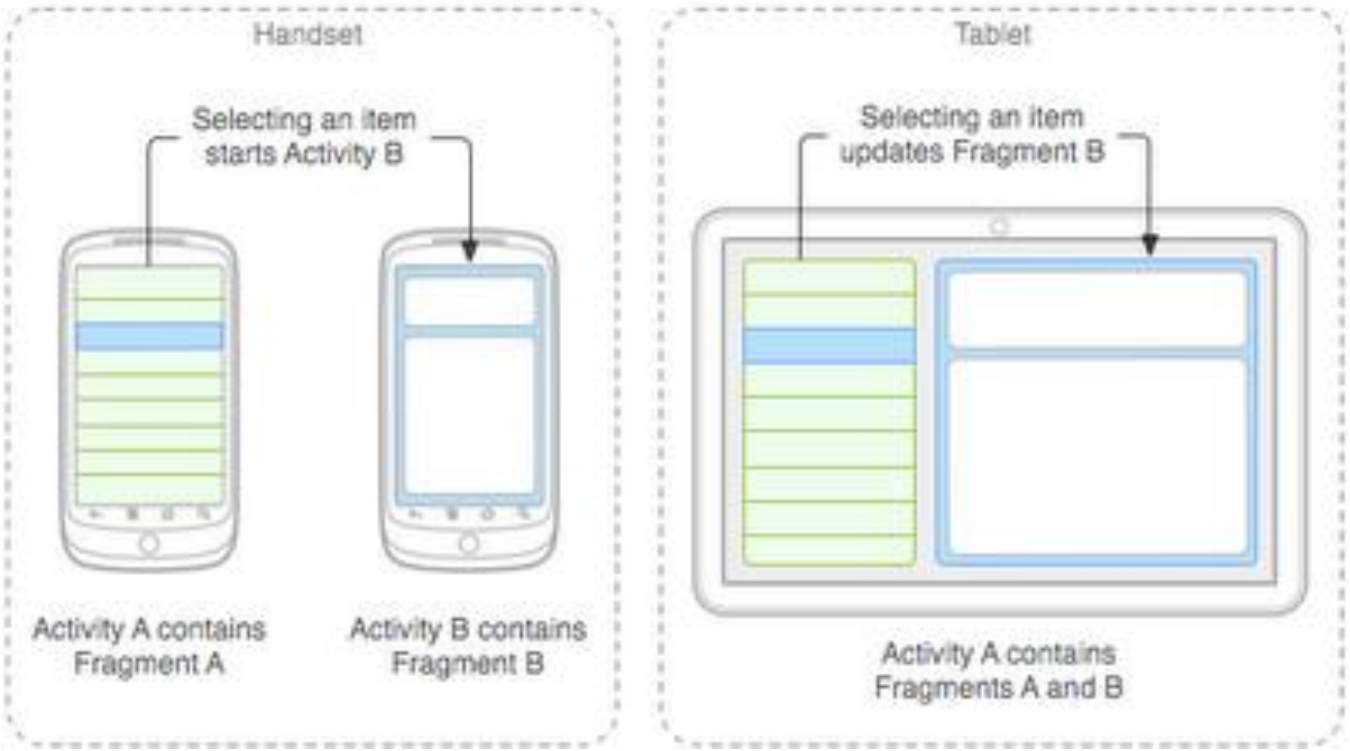
Recall: Activities

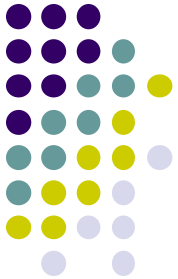
- Activity: main building block of Android UI
- Analogous to a window or dialog box in a desktop application
- Apps
  - have at least 1 activity that deals with UI
  - Entry point of app similar to **main( )** in C
  - typically have multiple activities
- Example: A camera app
  - **Activity 1:** to focus, take photo, start activity 2
  - **Activity 2:** to present photo for viewing, save it



Fragments

- Fragments
  - UI building blocks (pieces), can be arranged in Activities in different ways.
  - Enables app to look different on different devices (e.g. phone vs tablet)
- An activity can contain multiple fragments that are organized differently on different devices (e.g. for phone vs tablet)
- More later





Services

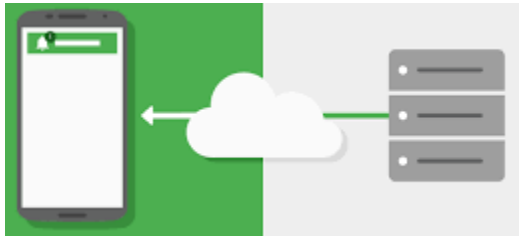
- Activities are short-lived, can be shut down anytime (e.g when user presses back button)
- Services keep running in background
- Similar to Linux/Unix CRON job
- Example uses of services:
  - Periodically check/update device's GPS location
  - Check for updates to RSS feed
- Independent of any activity, minimal interaction
- Typically an activity will control a service -- start it, pause it, get data from it
- Services in an App are sub-class of Android's **Services** class

Android Platform Services

- Android Services can either be on:
  - On smartphone or Android device (local)
  - Remote, on Google server/cloud

- Android platform local services examples (on smartphone):
  - **LocationManager:** location-based services.
  - **ClipboardManager:** access to device's clipboard, cut-and-paste content
  - **DownloadManager:** manages HTTP downloads in background
  - **FragmentManager:** manages the fragments of an activity.
  - **AudioManager:** provides access to audio and ringer controls.

Android services on smartphone

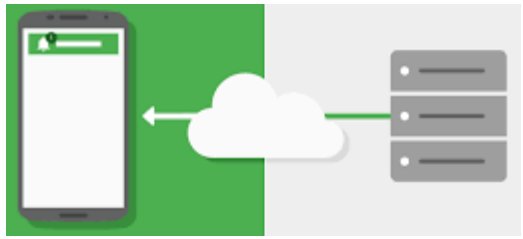


Android services In Google cloud

Google Services (In Google Cloud)

- Maps
- Location-based services
- Game Services
- Authorization APIs
- Google Plus
- Play Services
- In-app Billing
- Google Cloud Messaging
- Google Analytics
- Google AdMob ads

Typically need
Internet connection

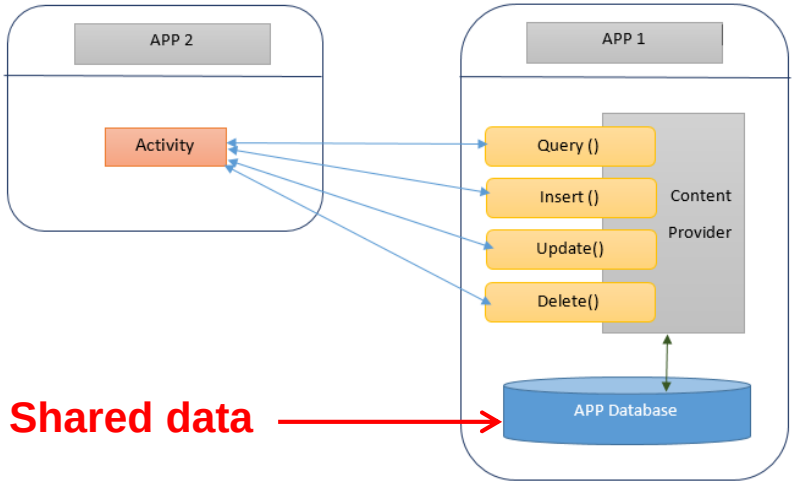


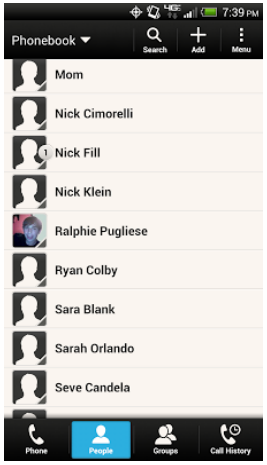
Android services
on smartphone

Android services
In Google cloud

Content Providers

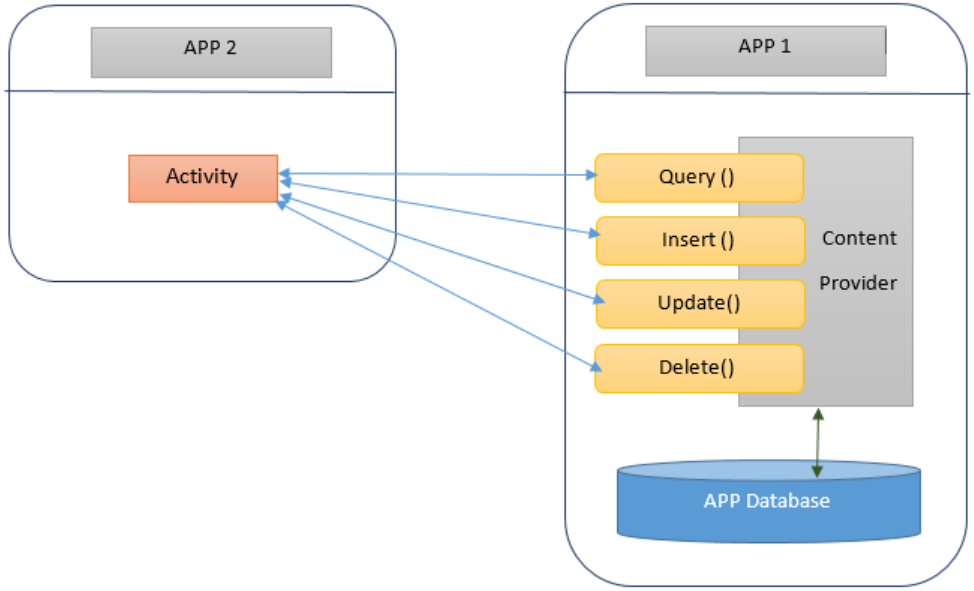
- Android apps can share data (e.g. User's contacts) as content provider
- Content Provider:
  - Abstracts shareable data, makes it accessible through methods
  - Applications can access that shared data by calling methods for the relevant **content provider**
  - E.g. Can query, insert, update, delete shared data (see below)

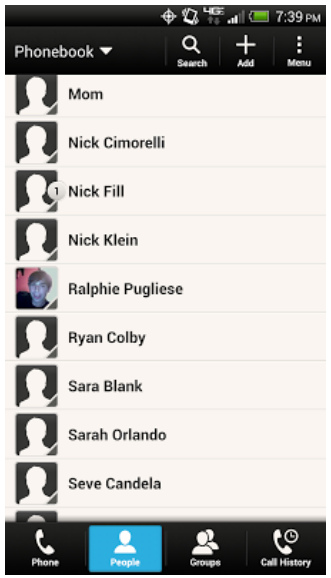




Content Providers

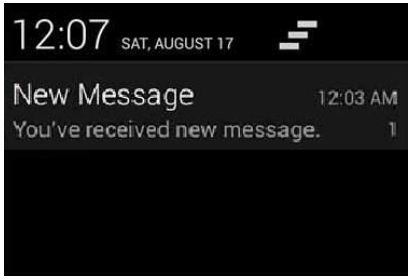
- **E.g.** Data stored in Android Contacts app can be accessed by other apps
- **Example:** We can write an app that:
  - Retrieve's contacts list from contacts content provider
  - Adds contacts to social networking (e.g. Facebook)
- Apps can also **ADD** to data through content provider. E.g. Add contact
- E.g. Our app can also share its data
- Content provider in an App are sub-class of Android's **ContentProvider** class





Broadcast Receivers

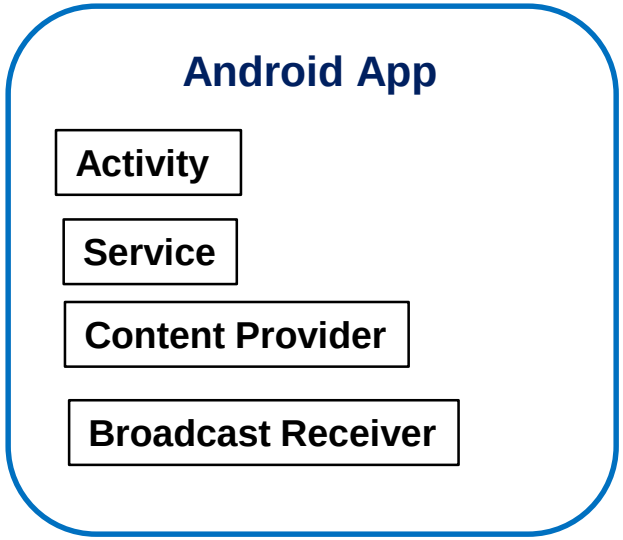
- Android OS (system), or applications, periodically ***broadcasts*** events
- Example broadcasts:
  - Battery getting low
  - Download completed
  - New email arrived
- Any app can create broadcast receiver to listen for broadcasts, respond
- Our app can also initiate broadcasts
- Broadcast receivers typically
  - Doesn't interact with the UI
  - Creates a status bar notification to alert the user when broadcast event occurs
- Broadcast Receiver in an App are sub-class of Android's **BroadcastReceiver** class



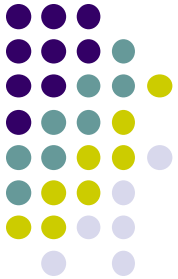
Quiz

- Pedometer App has the following Android components:
  - **Component A:** continously counts user's steps even when user closes app, does other things on phone (e.g. youtube, calls)
  - **Component B:** Displays user's step count
  - **Component C:** texts user's friends (from contacts list) every day with their step totals

- What should component A be declared as?
  - Activity, service, content provider, broadcast receiver?
- What of component B?
- Component C?



References



- Busy Coder’s guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014