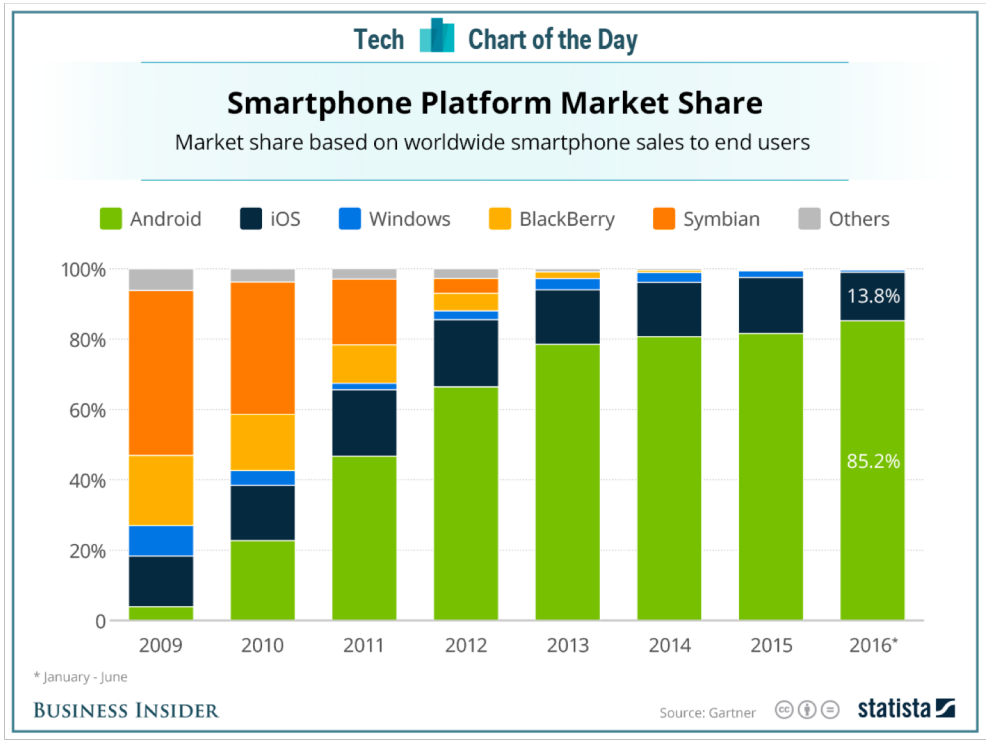
EECS 448 Smartphone Security
Android Basics

Xusheng Xiao

Electrical Engineering and Computer Science
Case Western Reserve University

Android Basics

- Architecture
- App Components
- Activity Lifecycle
- Inter-Component Communications



Android Architecture

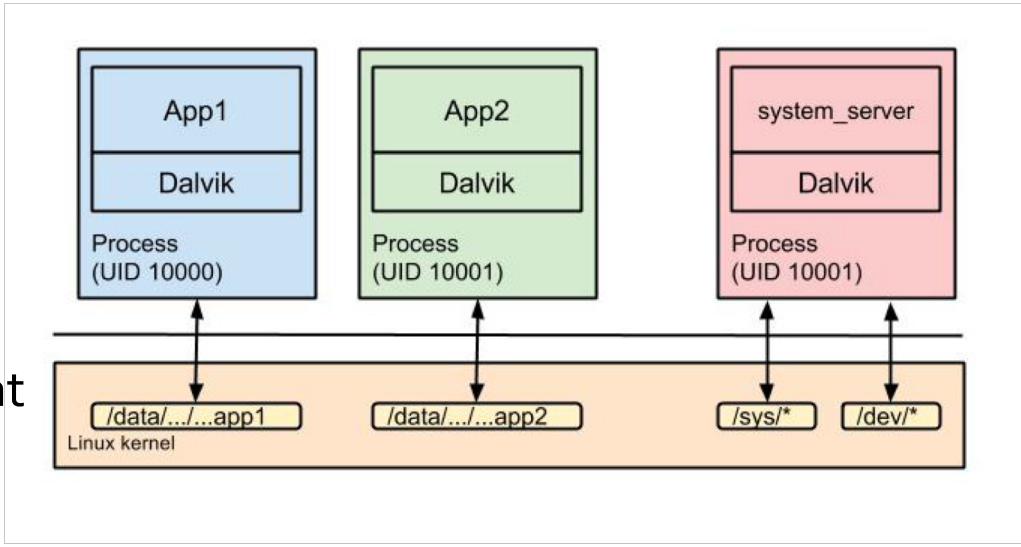
- Linux kernel
  - Threading, memory management
- Hardware Abstraction Layer (HAL)
  - Interfaces that expose device hardware capabilities
- Native C/C++ libraries
  - OpenGL
- Android Runtime
  - Running App using VM
- Java API Framework
  - APIs for the OS features
- System Apps
  - Email, SMS, Internet Browsing, etc.

Android Runtime

- Running Android apps using VMs
  - Optimized for low memory requirements
  - ART (after 5.0, with Ahead-Of-Time compilation)
  - Dalvik
- Running Dex bytecode
  - Legacy javac toolchain:
    javac (.java → .class) → **dx** (.class → .dex)
  - New Jack toolchain:
    jack (.java → .jack → .dex)

App Sandbox

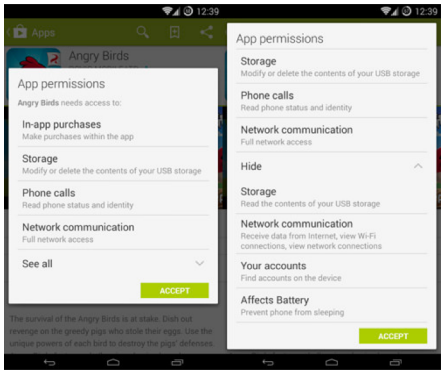
- Multi-user Linux system
- Each app is assigned a unique Linux user ID
  - User ID is unknown to the app
  - Only the user ID assigned to that app can access files in the app



- Each process has its own virtual machine (VM)
  - An app's code runs in isolation from other apps
- Every app runs in its own Linux process
  - Start a process for executing any of the app's components
  - Shut down the process when it's no longer needed or when the system must recover memory for other apps

Data Access

- Principle of least privilege
  - Each app has access only to the required components and no more

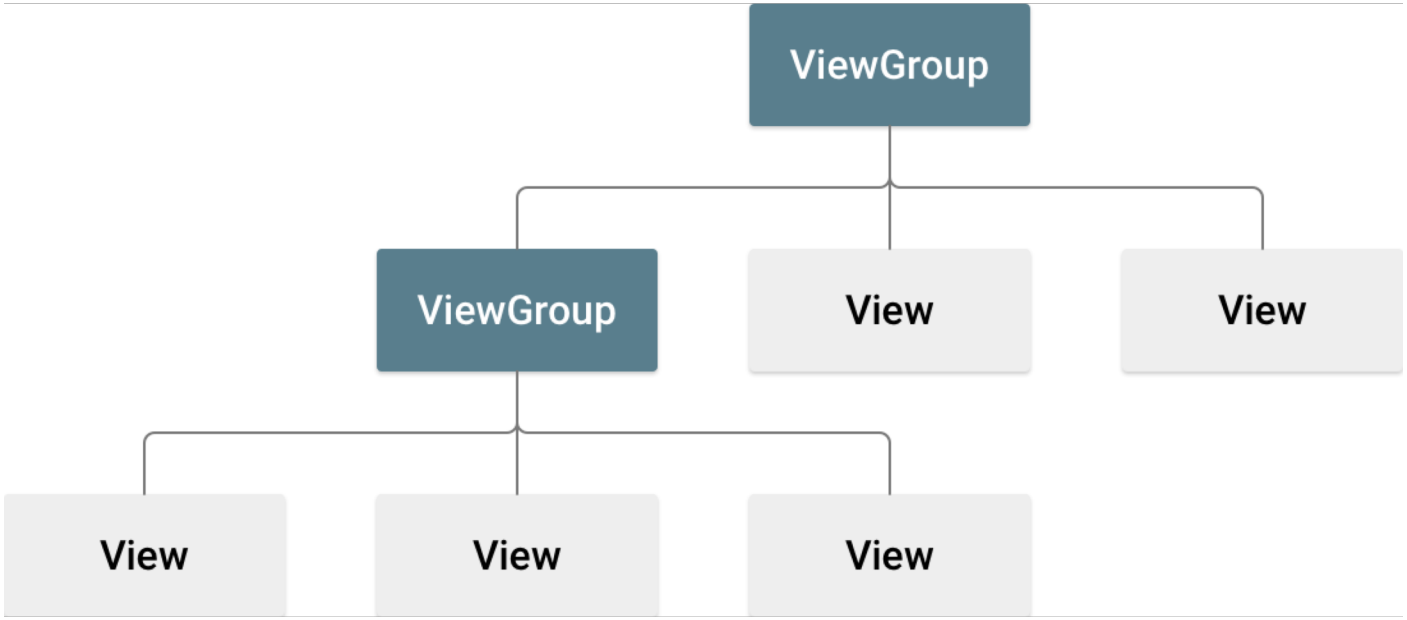


- Data sharing
  - Apps signed with the same certificate may run in the same process
  - **System permissions** are required to access device data (contacts, SMS)

Java API Framework

- View system
  - App's UI
- Resource manager
  - Access to resources: strings, images, etc.
- Notification manager
  - Custom alerts
- Activity manager
  - Manages the lifecycle of apps and provides a common navigation back stack
- Content provider
  - Access to other apps' data

View System



- ViewGroup: LinearLayouts, RelativeLayouts
- View: TextView, Button, RadioButton
- UI specification: XML layouts or code

Resource Manager

- Manage resource files
  - Located in res/ folder in the apk
  - Resource types: strings, images, layouts
  - Pre-compiled IDs for each resource

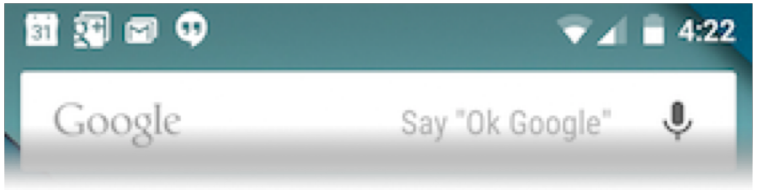
```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/submit" />
```

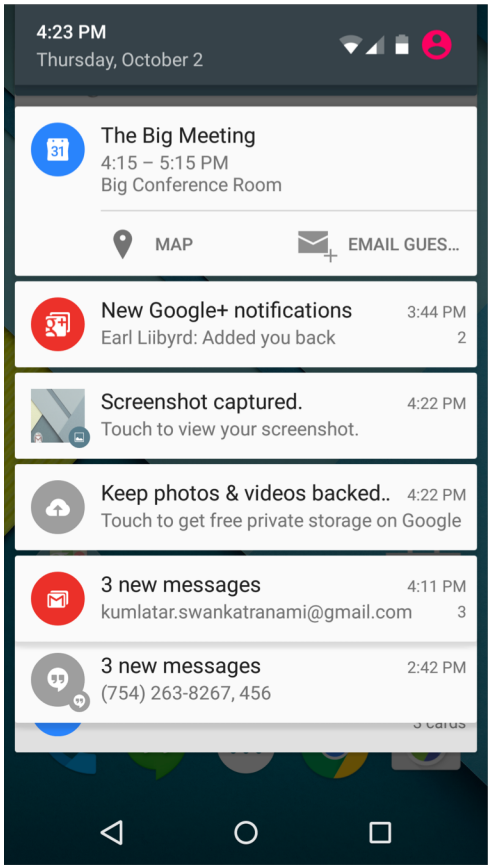
- Access to resources in code

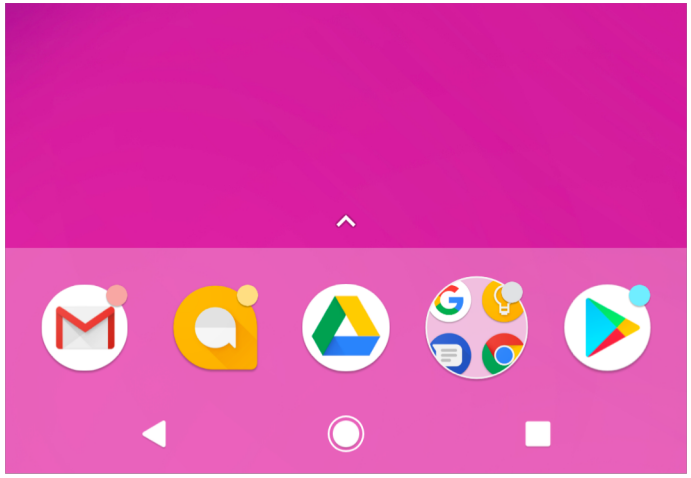
```
ImageView imageView = (ImageView) findViewById(R.id.myimageview);
imageView.setImageResource(R.drawable.myimage);
```

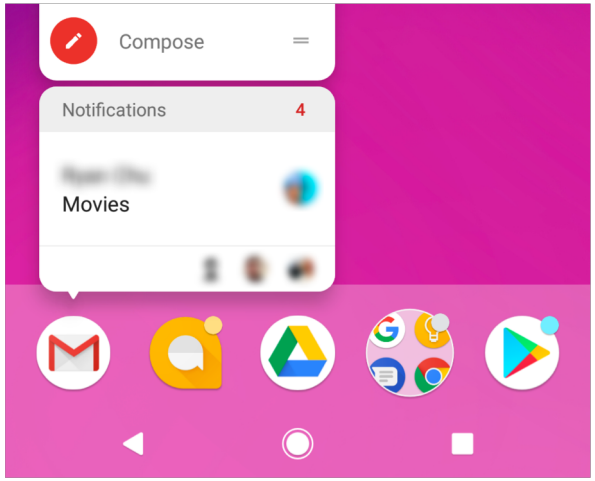
Notification Manager

- A message you display to the user outside of your app's normal UI.







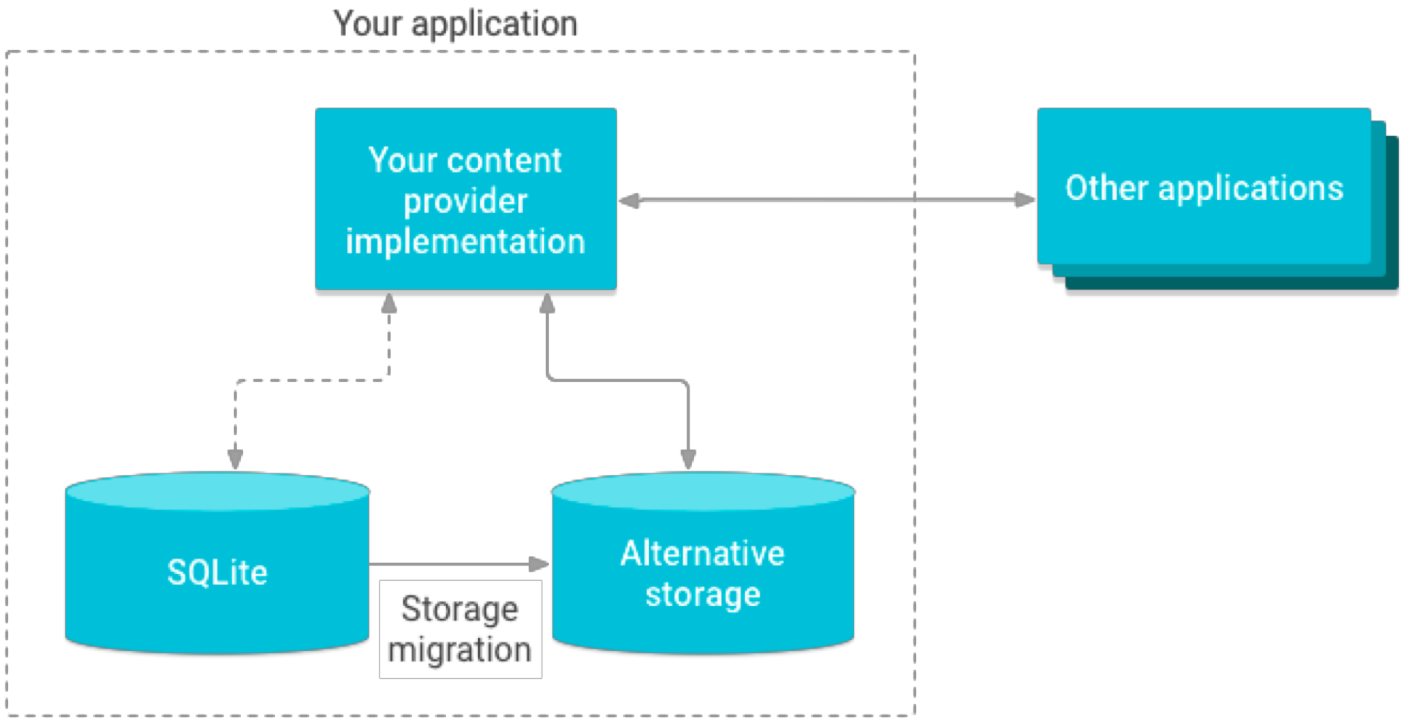


Activity Manager

- A standard application component is an **Activity**
  - Typically represents a single screen
  - Main entry point (equiv. to main() method)

- Activity manager manages activities' life cycles
  - One activity runs at a time, others are paused in the background.
  - As users navigate through your application, they switch activities.
  - Every activity has a state*: run*, *paused*, or *stopped.*
    - Changing state fires a corresponding activity method.

Content Provider



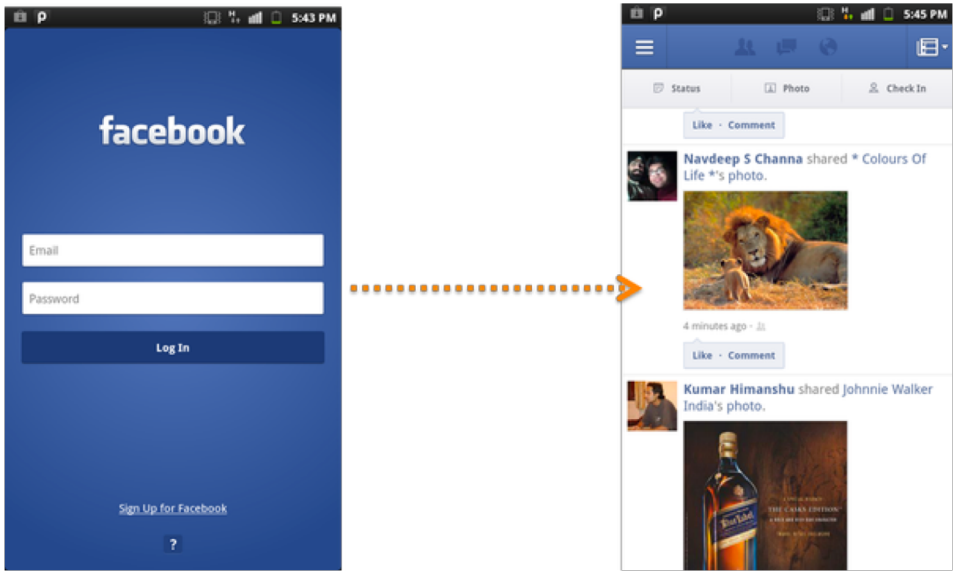
- Managing access to data stored by itself, stored by other apps, and provide a way to share data with other apps.
- Standard interface that connects data in one process with code running in another process.

App Components

- Essential building blocks of an Android app
  - An entry point for the system or a user to enter the app
- Four different types of app components:
  - Activities: a single screen with UI
  - Services: long-running operations in the background or services for remote processes
  - Broadcast receivers: responding to system-wide broadcast events
  - Content providers: shared data accesses

Activity

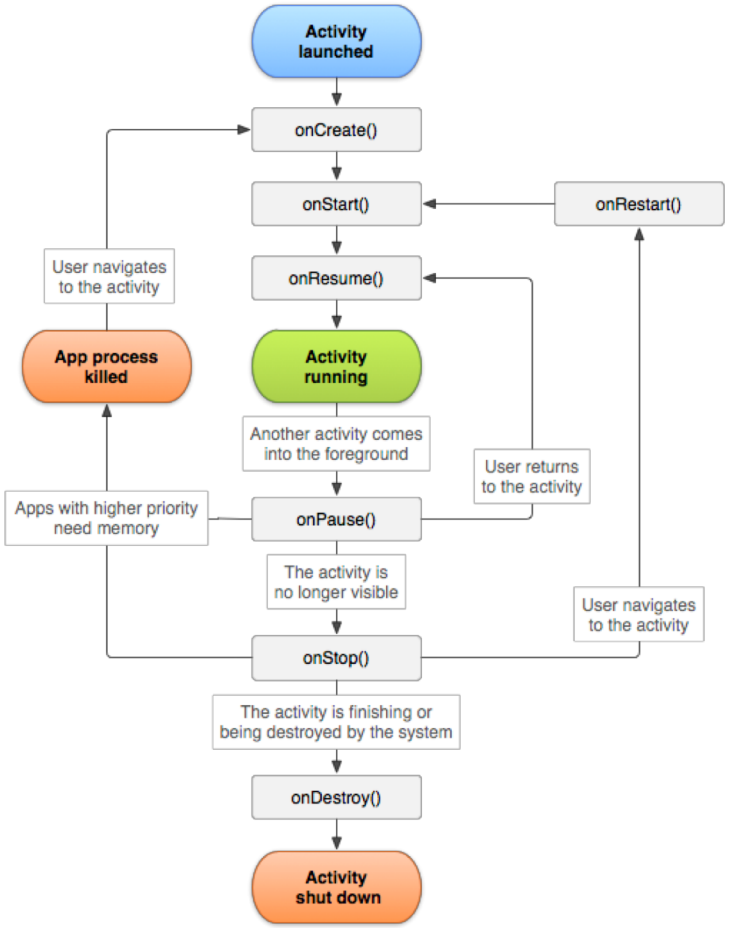
- Entry point for interacting with users
  - Representing a single screen



- Activities have an explicit lifecycle
  - One activity runs at a time, others are paused in the background.
  - As users navigate through your application, they switch activities.

Activity Lifecycle

- onCreate()
  - First creating the activity
- onStart()
  - Making UI visible
- onResume()
  - Bringing back to foreground
- onPause()
  - Receiving a phone call, navigating to another activity, or screen's turning off.
- onStop()
  - No longer visible to users
- onDestroy()
  - Destroying the activity

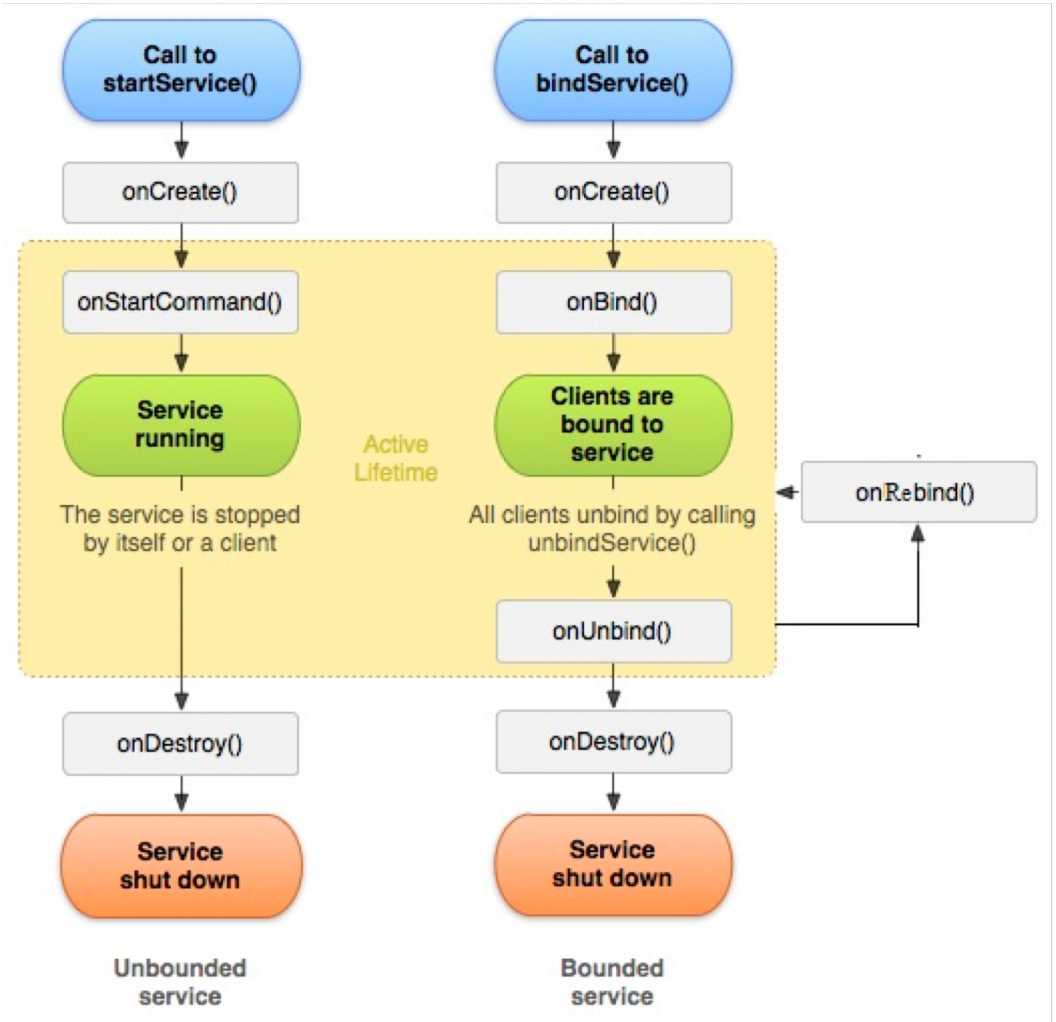


Service

- Performing long-running operations in the background while not interacting with users
  - Playing music, location update
- Providing functionality for other apps to use
  - Playing music and get data from the music

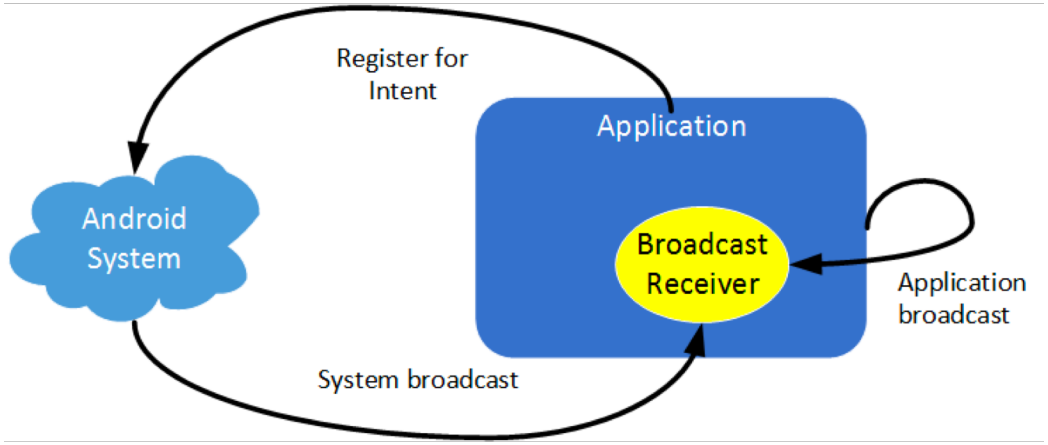
Service Lifecycle

- Unbounded
  - Started by startService()
  - Destroyed by stopSelf() or another component calling stopService().
- Bounded
  - Started by bindService()
  - Destroyed if unbound from all of its clients

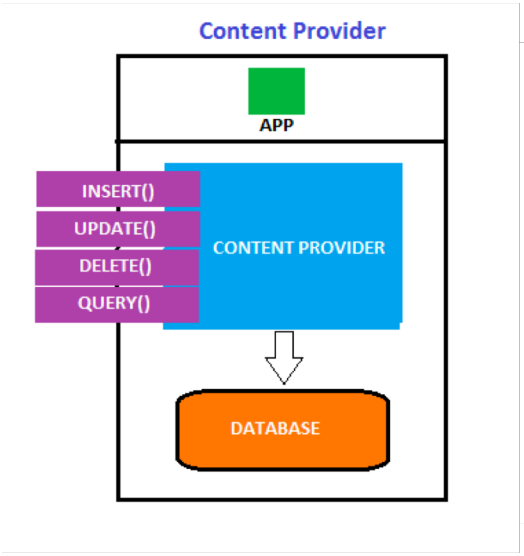


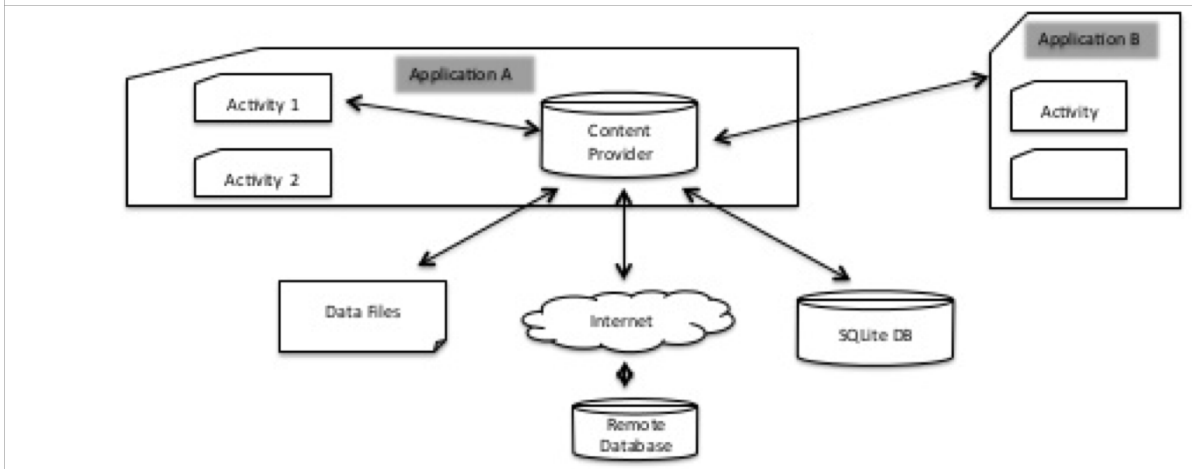
Broadcast Receiver

- A component that allows apps to respond to system-wide broadcast announcements
  - Screen turned off, phone call coming, SMS, etc.
- No UI, but may create a status bar notification



Content Provider





- Manage accesses to data sources
  - SQLLite, files, remote databases, etc.
- Expose interfaces for other apps to use
  - Access by URI: content://<authority>/<data_type>/<id>
    - E.g., content://contacts/people/5
  - APIs: intert(), update(), delete(), query()

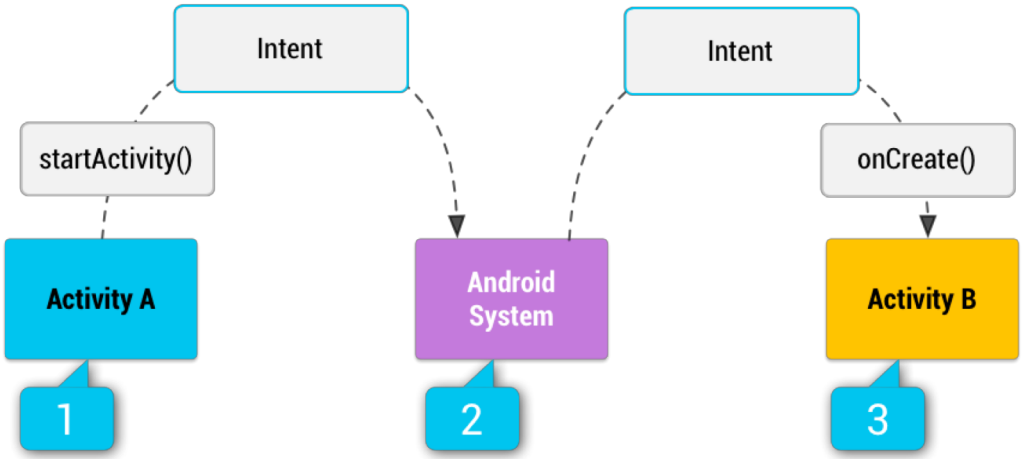
Inter Component Communication

- Intent: a messaging object to request an action from another app component
  - Explicit intent: specify the component to start by name, e.g., activity or service names
  - Implicit intent: declare a general action to perform, e.g., a specified location in a map
- Intent filters: an expression to specify what intents that the component would like to receive

```
<activity android:name="ShareActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
    </intent-filter>
</activity>
```

Starting Another Activity

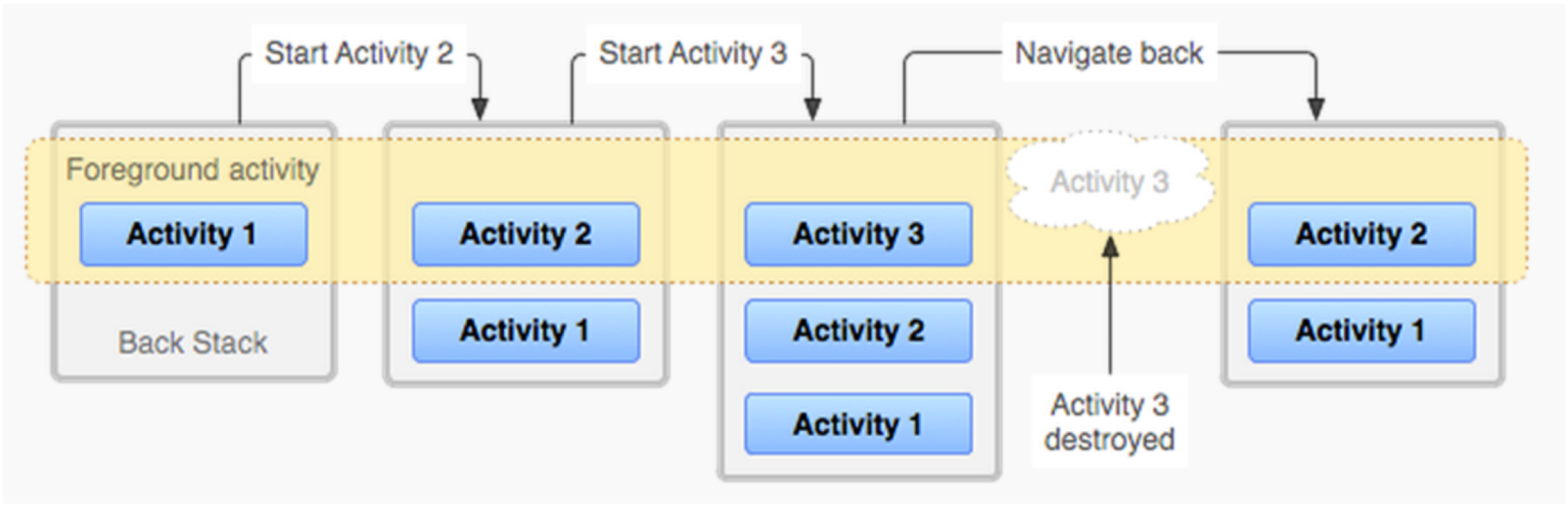
- Build an intent, and then use startActivity() to start another activity




```
Intent intent = new Intent(this, DisplayMessageActivity.class);
EditText editText = (EditText) findViewById(R.id.editText);
String message = editText.getText().toString();
intent.putExtra(EXTRA_MESSAGE, message);
startActivity(intent);
```

"Back Stack"

- Navigation forward/back through activities is typically triggered by user actions
  - An activity creating another activity
  - Pressing Back button



Starting A Service

- Start a service from an activity or other application component by passing an Intent to startService() or startForegroundService().
- The Android system calls the service's onStartCommand() method and passes it the Intent

```
Intent intent = new Intent(this, HelloService.class);
startService(intent);
```

Registering Broadcast Receivers

- Manifest file

```
<receiver android:name=".MyBroadcastReceiver"  android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <action android:name="android.intent.action.INPUT_METHOD_CHANGED" />
    </intent-filter>
</receiver>
```

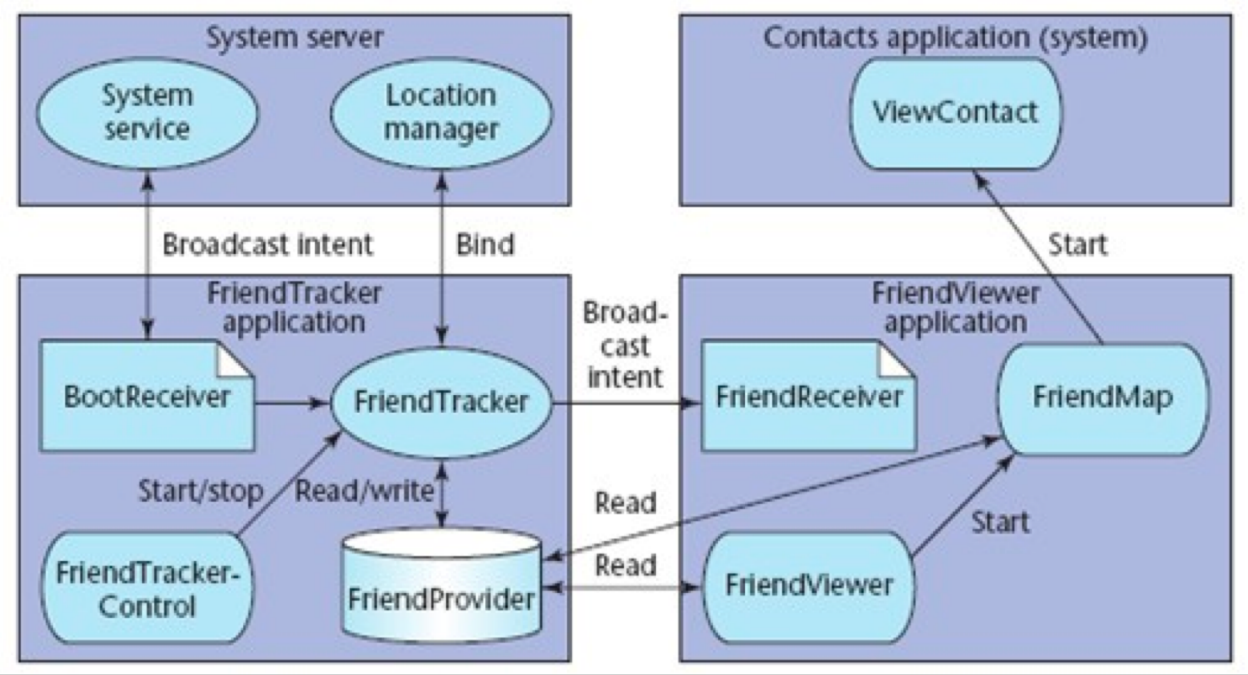
- App code

```
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
intentFilter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
this.registerReceiver(br, filter);
```

- Broadcasting

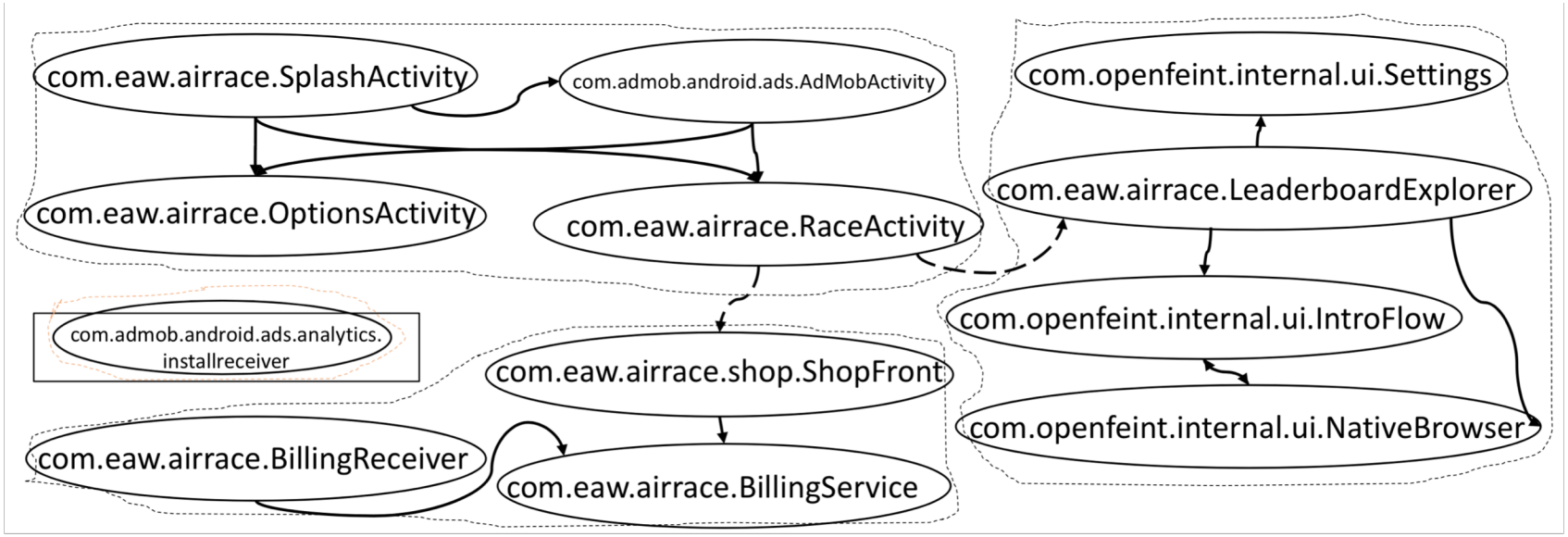
```
Intent intent = new Intent();
intent.setAction("com.example.broadcast.MY_NOTIFICATION");
intent.putExtra("data","Notice me senpai!");
sendBroadcast(intent);
```

ICCs in An Example App



- Activity: FriendTrackerControl, FriendViewer, FriendMap, ViewContact
- Service: System service, FriendTracker, Location manager
- Broadcast Receiver: BootReceiver, FriendReceiver
- Content Provider: FriendProvider

ICCs in An Real App - RacePilot



- Gaming: SplashActivity, OptionsActivity, RaceActivity
- Social: LeaderboardExplorer, IntroFlow, NativeBrowser, Settings
- Purchasing: ShopFront, Billing Receiver, BillingService
- Analytic: InstallerReceiver

Thank You !

Questions ?