



# ANDROID DEVELOPMENT

## BROADCASTS

Tan Cher Wah ([isstcw@nus.edu.sg](mailto:isstcw@nus.edu.sg))



# Agenda

- What are Broadcasts?
- Broadcast Receivers



# Broadcast Overview

- Android apps can send or receive broadcast messages
  - Android system (System events)
  - Other apps within the device (Custom events)
- System events
  - Arrival of SMS message
  - Low battery condition
  - Network availability
- App events
  - Download completed (e.g. from Service to Activity)



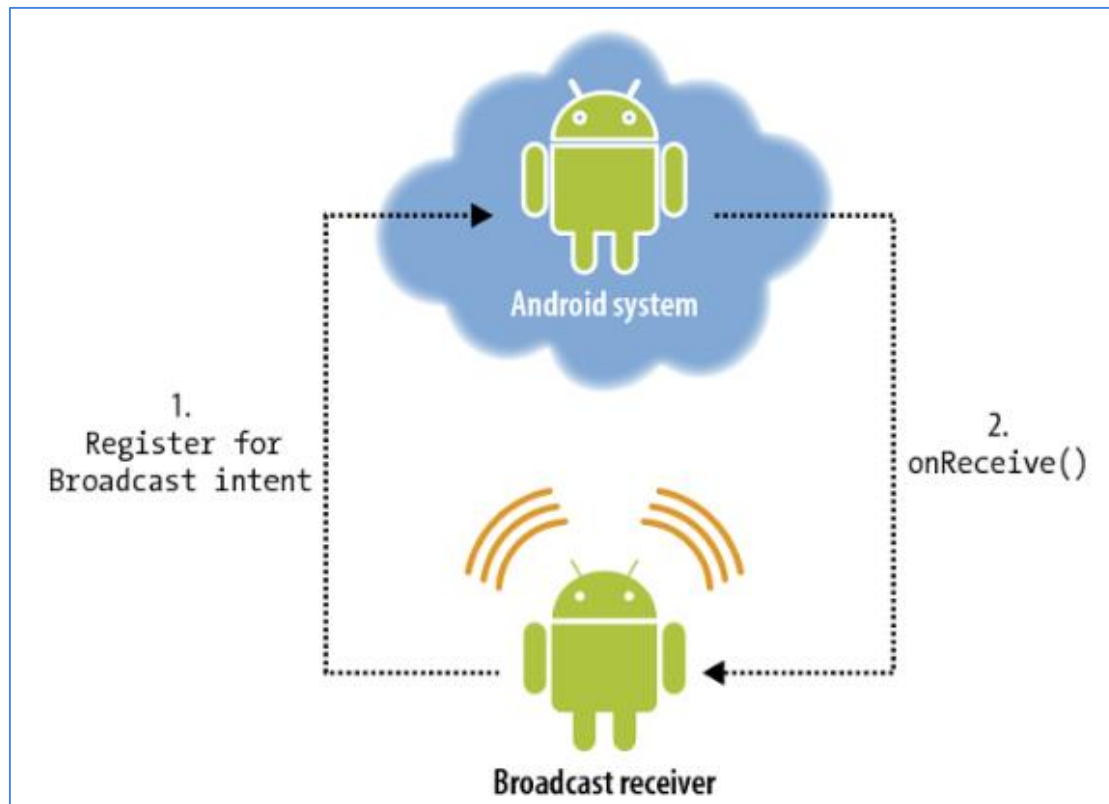
# Broadcasting

- Global Broadcast
  - System-Wide – any apps in the device can receive
  - Security concern
  - Performance concern
- Local Broadcast
  - App-Wide – broadcasts take place between app components within your app (e.g. broadcast from your Activity to your Services)
  - No concern on Data Leakage



# Broadcast Receiver

- To receive a broadcast
  - Register for the specific broadcast
  - Implement `onReceive()` to receive broadcast





- For System-Wide broadcasts, use a global broadcast register

```
protected MyBroadcastReceiver receiver = new MyBroadcastReceiver();
```

```
IntentFilter filter = new IntentFilter();  
filter.addAction(Intent.ACTION_POWER_CONNECTED);  
filter.addAction(Intent.ACTION_POWER_DISCONNECTED);  
filter.addAction(Telephony.Sms.Intents.SMS_RECEIVED_ACTION);  
registerReceiver(receiver, filter);  
  
String permission = Manifest.permission.RECEIVE_SMS;  
int grant = ContextCompat.checkSelfPermission(this, permission);  
if (grant != PackageManager.PERMISSION_GRANTED) {  
    String[] permission_list = new String[1];  
    permission_list[0] = permission;  
    ActivityCompat.requestPermissions(this, permission_list, 1);  
}
```

- Specify **permissions** in app's manifest

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

- Unregister receiver when done

```
unregisterReceiver(receiver);
```



# Local Broadcast

- For Local broadcasts, use a LocalBroadcastManager

```
IntentFilter customFilter = new  
IntentFilter(ACTION_CUSTOM_BROADCAST);  
  
LocalBroadcastManager.getInstance(this)  
    .registerReceiver(receiver, customFilter);
```

- Unregister receiver when done

```
LocalBroadcastManager.getInstance(this)  
    .unregisterReceiver(receiver);
```



# Broadcast Receiver example

```
public class MyBroadcastReceiver extends BroadcastReceiver {

    public final String ACTION_CUSTOM_BROADCAST = "ACTION1";

    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action == null)
            return;

        String msg = "";

        switch (action) {
            case Intent.ACTION_POWER_CONNECTED:
                msg = "Power connected...";
                break;

            case Intent.ACTION_POWER_DISCONNECTED:
                msg = "Power disconnected...";
                break;

            case Telephony.Sms.Intents.SMS_RECEIVED_ACTION:
                msg = "Received SMS...";
                break;

            case ACTION_CUSTOM_BROADCAST:
                msg = "Custom Broadcast received...";
                break;
        }

        Toast.makeText(context, msg, Toast.LENGTH_SHORT).show();
    }
}
```





# Broadcast Receiver Limitations

- Time Limit
  - Can only run for 10 secs once start
- Functionality Limit
  - Can **start** a Service, but not **bind** to it



# References

- BroadcastReceiver -  
<https://developer.android.com/guide/components/broadcasts>
- Understanding Android Broadcasts -  
<https://code.tutsplus.com/tutorials/android-from-scratch-understanding-android-broadcasts--cms-27026>
- Android BroadcastReceiver Tutorial -  
<https://en.proft.me/2017/03/10/android-broadcastreceiver-tutorial/>
- List of Broadcast Receivers -  
<https://stackoverflow.com/questions/4031740/where-is-the-all-android-broadcast-intent-list>



# Active Recall

