Mobile Applications
CSCI 448
Lecture 35

Broadcast Receivers

Learning Outcomes For Today

- Discuss how a Broadcast Receiver operates and concerns when listening to broadcasted events

On Tap For Today

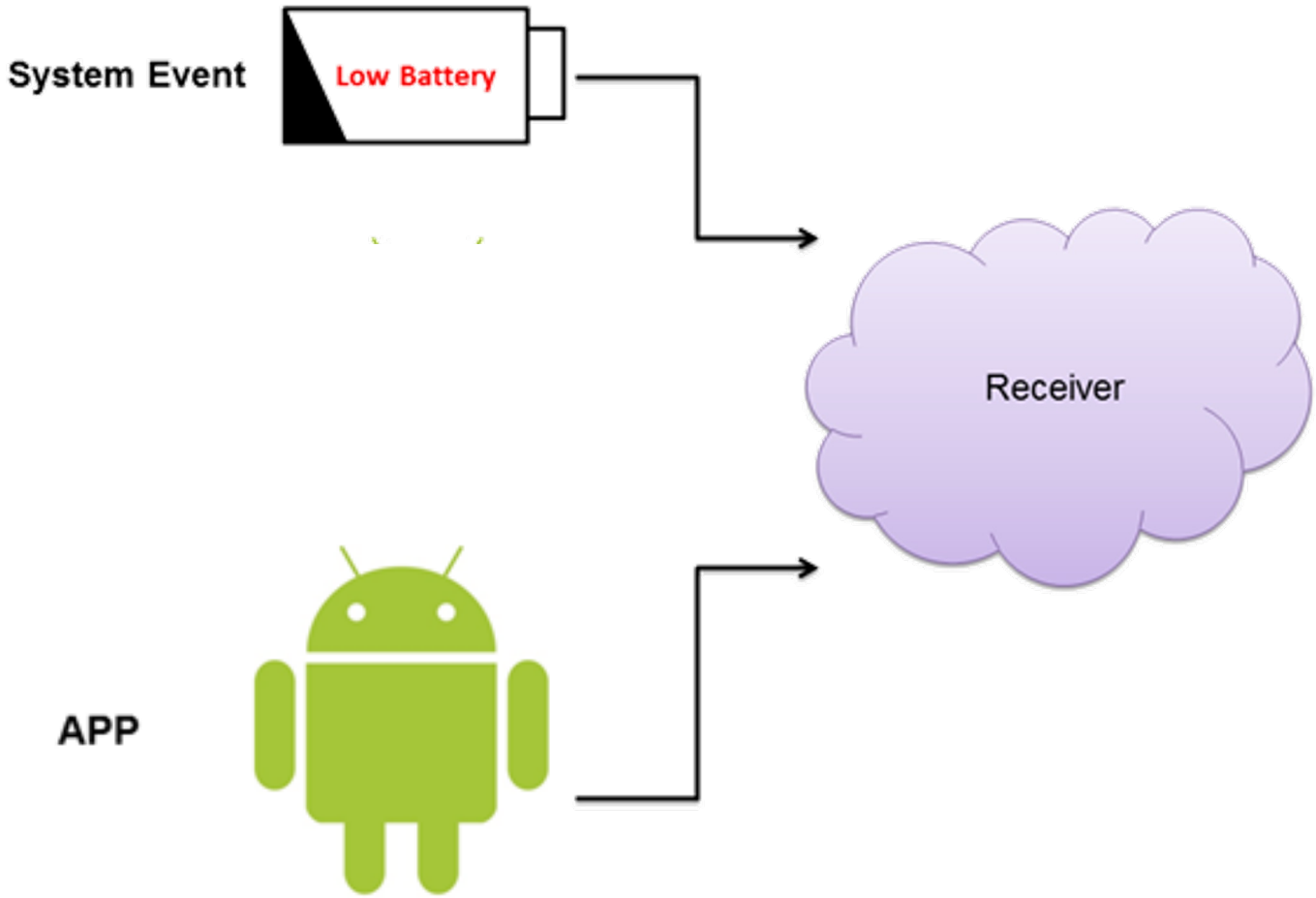
- Broadcast Receivers
- Final Project Remainders

On Tap For Today

- Broadcast Receivers

- Final Project Remainders

Events That Are Broadcast



Create a Receiver

```
class MyReceiver : BroadcastReceiver() {
  override fun onReceive(context: Context, intent: Intent) {
    if (intent.action == ACTION_LISTENING_FOR) {
      // perform action
    }
  }
}
```

Context Register Receiver

```
// onCreate
val myReceiver = MyReceiver()
val filter = IntentFilter(ACTION_LISTENING_FOR)
val listenToBroadcastsFromOtherApps = false
val receiverFlags = if (listenToBroadcastsFromOtherApps) {
  ContextCompat.RECEIVER_EXPORTED
} else {
  ContextCompat.RECEIVER_NOT_EXPORTED
}
ContextCompat.registerReceiver(context, myReceiver, filter, receiverFlags)

// onDestroy
ContextCompat.unregisterReceiver(myReceiver)
```

Statically Register Receiver

```
<!-- in AndroidManifest.xml -->
<manifest ... >
  <application ...>
    <activity ...> </activity>
    <receiver
      android:name=".MyReceiver">
    </receiver>
  </application>
</manifest>
```

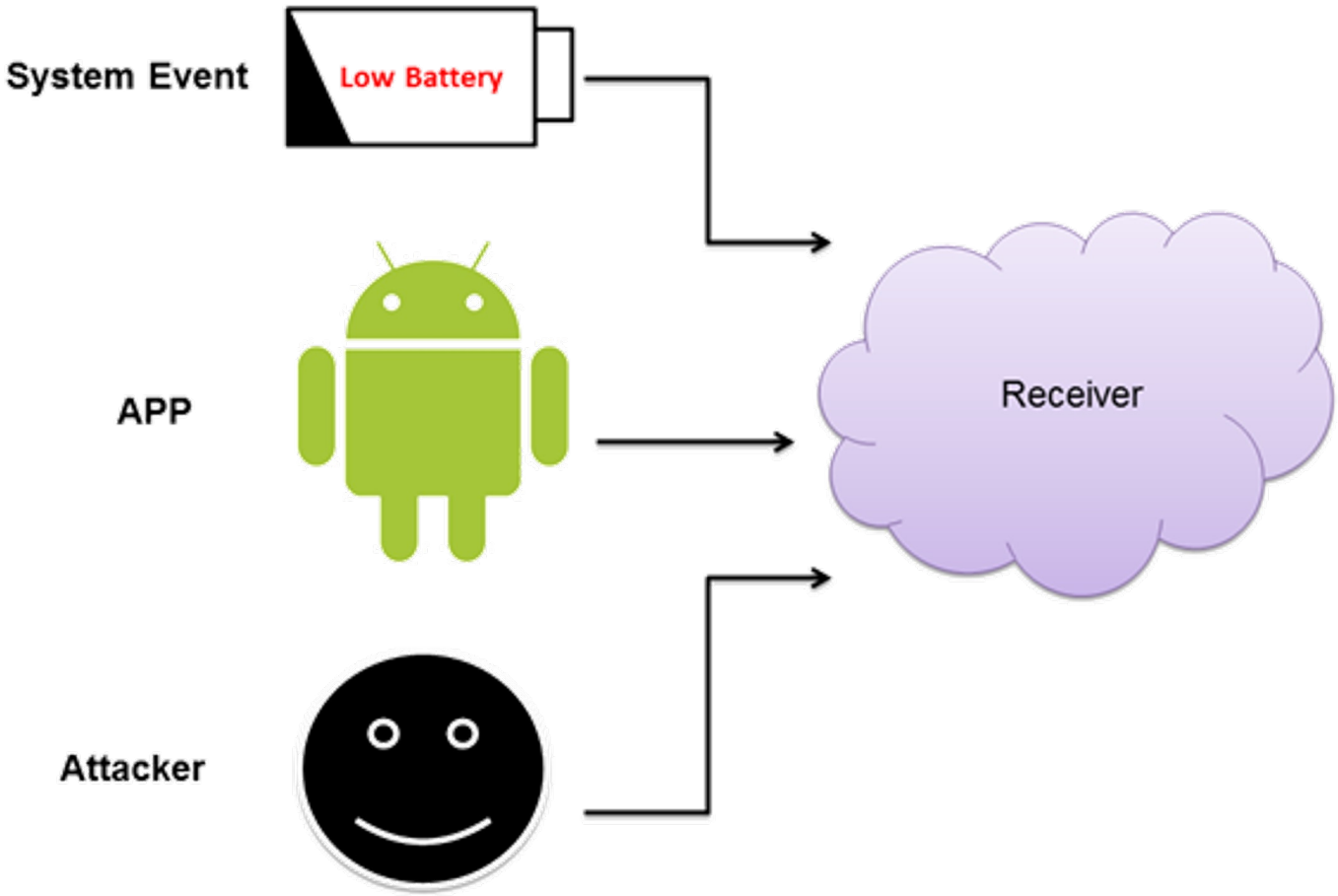
Broadcasting an Event

```
// create intent with Action set
val intent = Intent(ACTION_LISTENING_FOR)
// send broadcast!
activity.sendBroadcast(intent)
```

Android Design Patterns

- Behavioral Patterns
  1. Command – UI Event Handling, Retrofit Request Callback, Activity Result Callback, Permissions Callback, Location
  2. Observer – State, Flow, LiveData, Broadcast Receiver
  3. Template Method - IScreenSpec
- Creational Patterns
  4. Builder – Compose NavGraph, WorkRequest, Constraints, Retrofit, LocationRequest
  5. Factory – ViewModelFactory
  6. Singleton – ViewModelProvider, Repository, Room Database
- Structural Patterns
  7. Decorator – View Model
  8. Façade – DAO, Repository

Events That Are Broadcast



Statically Register Receiver

```
<!-- in AndroidManifest.xml -->
<manifest ... >
  <application ...>
    <activity ...> </activity>
    <receiver
      android:name=".MyReceiver"
      android:exported="false"
      android:permission="PERMISSION"> <!-- optional -->
        <intent-filter> <!-- optional -->
          <action android:name="ACTION_LISTENING_FOR"/>
        </intent-filter>
    </receiver>
  </application>
</manifest>
```

android:exported

- Where can receiver listen?
  - false : listen from
    - System
    - Components of same application
    - Applications with same User ID
  - true : listen from
    - Non-system sources outside its application

On Tap For Today

- Broadcast Receivers
- Final Project Remainders

Final Project

- Final Release: finish the app
- Website & Video: advertise your app!
- Final Report: Documentation to date + change log and reflection on development
- During final exam time slot:
  - Final Presentation
  - Team Evaluation
- Extra Credit: Deploy to Play Store

Final Release

- Complete the app
  - Fully functional
    - Navigation
    - Calculations
    - Logic
  - Contains 3+ components

- Due: Thursday, May 4 11:59pm

Final Report

- Guide on how to use app
- Reflect on the development process you went through
  - What changes were made to original design? Why?
- Challenges faced

- Due: Thursday, May 4 11:59pm

Website & Video

- Together, both should "sell" your app. Why should people download your app? What makes it cool?
  - Website can be simple
  - Video demonstrates your app running
    - Can use Zoom to record emulator with voice over
- See past examples https://cs-courses.mines.edu/csci448/homework/appstore.html
- Due: Thursday, May 4 11:59pm

Team Feedback

- Provide feedback on your team members
- Form to come, will be done during final slot
- Due: Monday, May 8
  - Individual deliverable

Final Presentation

- During final exam slot, give a demo of your app
  - What does it do?
  - What are the components being used?
  - Anything cool about it?

- Due: Monday, May 8

Extra Credit: Deploy to Play Store

- Deploy your app to the Google Play Store
- More downloads received == more extra points
- Due: Monday, May 8

Questions?

To Do For Next Time

- Lab11 due today
- A3 due tomorrow

- Everything's posted for remainder of semester

- Lab12
  - A: Create alarm to trigger sending broadcast
  - B: Receive broadcast and post notification