



**FPT POLYTECHNIC**

LẬP TRÌNH ANDROID NÂNG CAO

## **Bài 2: Broadcast Receiver**

---

[www.poly.edu.vn](http://www.poly.edu.vn)

---

## Nội dung bài học

- Tổng quan về Broadcast Receiver
- Phân loại Broadcast Receiver
- Miêu tả Broadcast Receiver
- Vòng đời Broadcast Receiver
- Hạn chế của Broadcast Receiver



# Broadcast Receiver

- Là một trong 4 thành phần của ứng dụng Android
- Chức năng dùng để nhận các sự kiện mà các ứng dụng hoặc hệ thống phát đi



# Broadcast Receiver



## Android Application Anatomy

### Activities

1. Provides **User Interface**
2. Usually represents a **Single Screen**
3. Can contain one/more **Views**
4. **Extends** the **Activity** Base class

### Services

1. **No User Interface**
2. Runs in **Background**
3. **Extends** the **Service** Base Class

Application= Set of Android Components

### Intent/Broadcast Receiver

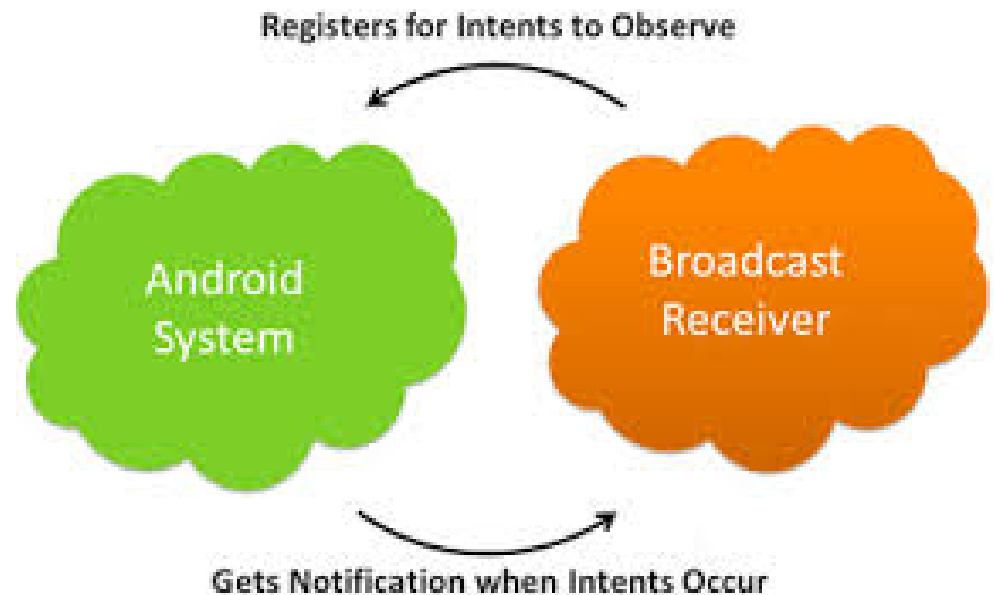
1. **Receives and Reacts to broadcast Intents**
2. No UI but **can start** an Activity
3. **Extends the BroadcastReceiver** Base Class

### Content Provider

1. **Makes application data available to other apps**
2. **Data stored in SQLite database**
3. **Extends the ContentProvider** Base class

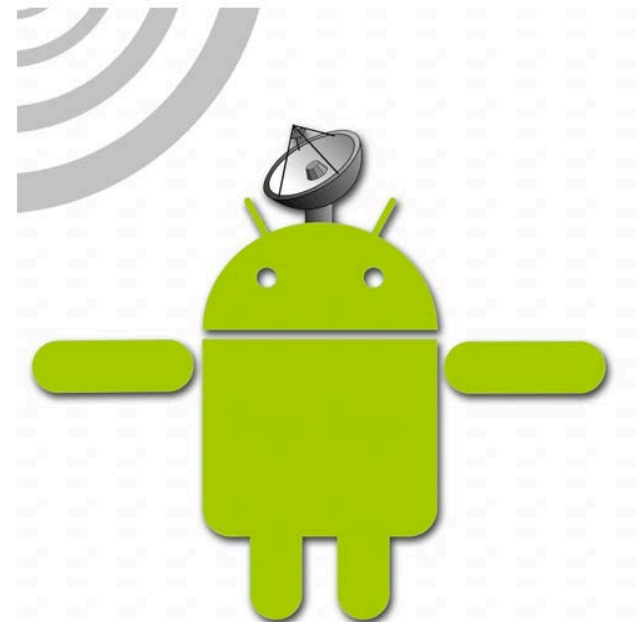
# Broadcast Receiver

- Có 2 cách phát-nhận:
  - Không có thứ tự: receiver nào đủ điều kiện thì nhận, không phân biệt và tách rời nhau
  - Có thứ tự: receiver nào đăng ký ưu tiên hơn thì nhận trước, và có thể truyền thêm thông tin xử lý cho các receiver sau



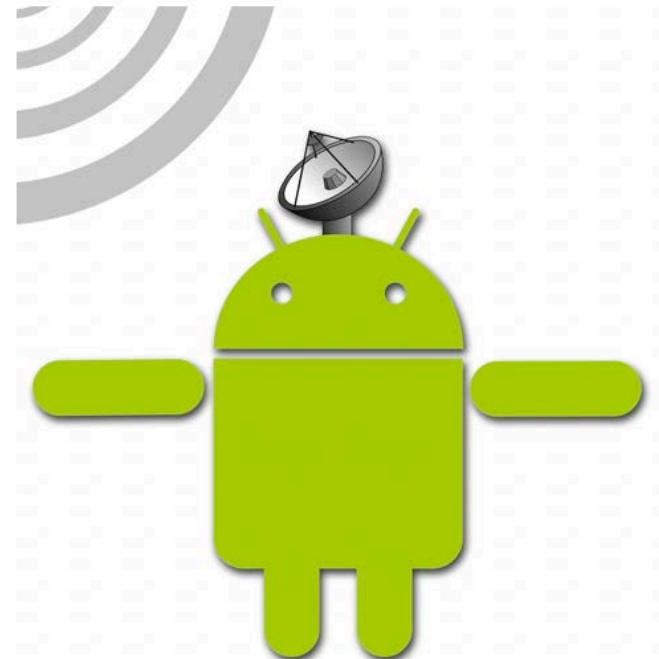
# Vòng đời của Broadcast Receiver

- Vòng đời của Broadcast Receiver gồm duy nhất một phương thức **onReceive()**
  - Khi có sự kiện mà Broadcast Receiver đã đăng ký nhận được phát đi, thì phương thức **onReceive()** của BroadcastReceiver đó sẽ được gọi
  - Sau khi thực hiện xong phương thức này, vòng đời của Broadcast Receiver kết thúc

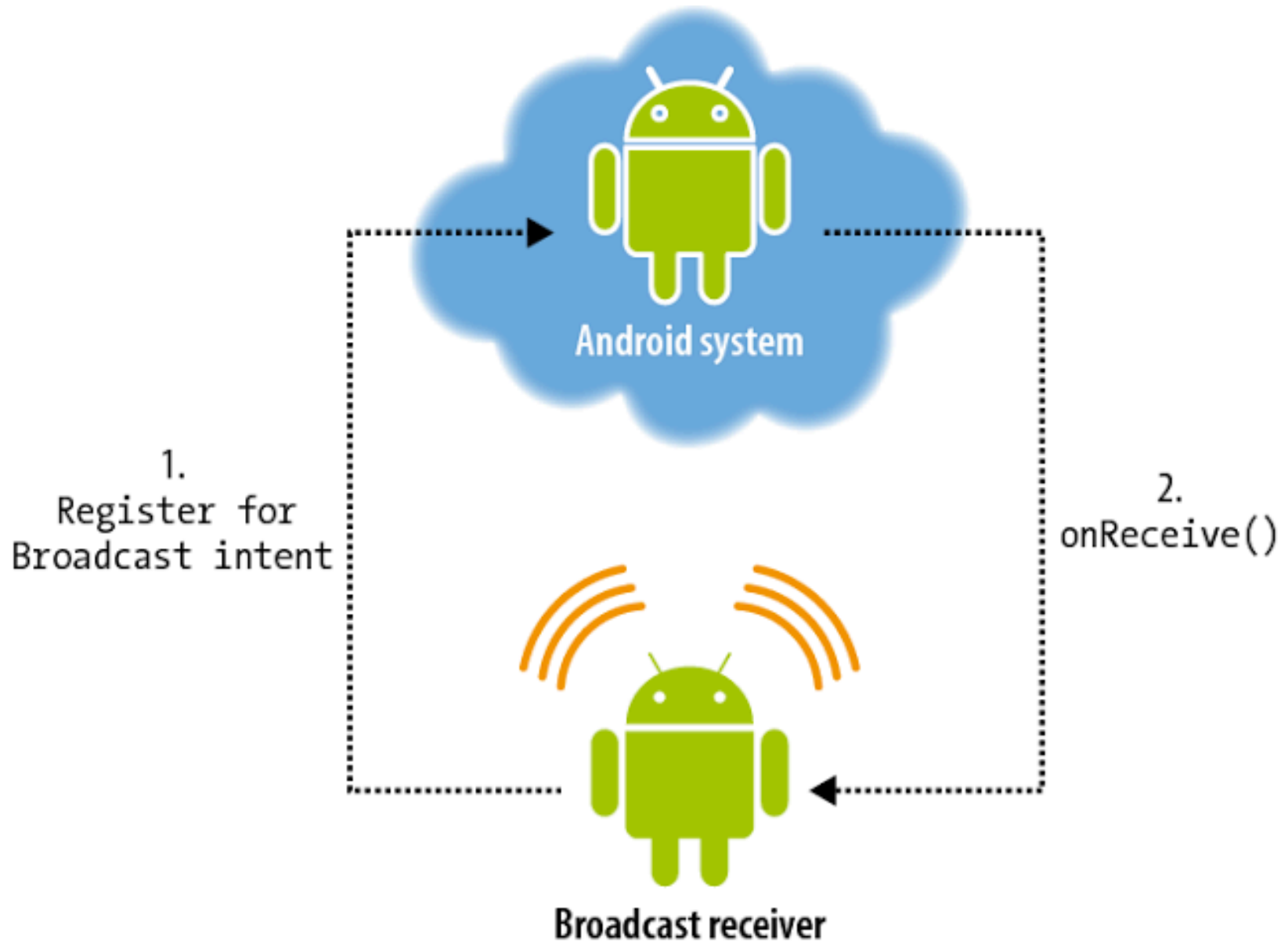


# Vòng đời của Broadcast Receiver

- Ngay khi onReceive() kết thúc, hệ thống có thể kill process chứa receiver bất kỳ lúc nào
- Bạn nên tránh xử lý tốn nhiều thời gian trong Broadcast Receiver
- Không nên có xử lý bất đồng bộ, chờ đợi callback... Trong Receiver (ví dụ như hiển thị Dialog, kết nối với service)



# Vòng đời của Broadcast Receiver





# Broadcast hệ thống

- Trong Android nhiều sự kiện hệ thống được hệ thống gửi đến toàn bộ ứng dụng trong hệ thống dưới dạng broadcast intent. Các intent này có các action được định nghĩa dưới dạng các biến final static trong lớp Intent cũng như các lớp khác (ví dụ TelephonyManager)
- Một số sự kiện quan trọng:
  - **Intent.ACTION\_BOOT\_COMPLETED**
  - **Intent.ACTION\_BATTERY\_LOW**
  - **Intent.ACTION\_BATTERY\_OKAY**

## Một số Broadcast phổ biến

- Báo hệ thống khởi động xong
- Bán pin có sự thay đổi
- Báo tắt máy
- Báo cắm sạc, rút sạc
- Báo có package mới được cài vào hay xóa đi



## Một số Broadcast phổ biến

- Thông báo tin nhắn tới
- Thông báo cuộc gọi đến
- Bạn có thể định nghĩa broadcast của riêng mình (mục tiêu chính là bạn có thể liên lạc giữa các ứng dụng bạn viết hoặc đưa ra thông báo một sự kiện quan trọng đến ứng dụng khác của bạn với các ứng dụng khác)



## onReceive()

- Phương thức này được gọi đến khi có sự kiện tương ứng được phát đi
- Trong phương thức này, bạn sẽ truyền vào context và intent

```
@Override  
public void onReceive(Context context, Intent intent) {  
  
}
```



# DEMO

Sử dụng phương thức onReceive



## **Hạn chế của Broadcast Receiver**

- Broadcast Receiver phải kết thúc việc thực thi trong 10 giây nếu không sẽ có ANR exception
- Từ Android 3.1, mặc định hệ thống sẽ loại bỏ tất cả receiver từ intent nếu như ứng dụng tương ứng với các receiver này chưa được start bởi người dùng bao giờ hoặc các ứng dụng này bị stop một cách tường minh bởi người dùng thông qua menu (Manage Application)

# Đăng ký Broadcast Receiver qua AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.fpolycallbroadcastapp"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="21" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver
            android:name=".MyCallReceiver"
            android:enabled="true"
            android:exported="true" >
        </receiver>
    </application>
</manifest>
```



# DEMO

Xây dựng ứng dụng sử dụng Broadcast  
Receiver





# Đăng ký Broadcast Receiver qua code

- Bạn có thể đăng ký Broadcast Receiver bằng code sử dụng phương thức **Context.registerReceiver()** trong onResume của Activity
- Trong onPause của Activity, gọi phương thức **Context.unregisterReceiver()**



# Disable receiver

- Để disable receiver đã đăng ký trong AndroidManifest.xml, chúng ta sử dụng **PackageManager**

```
ComponentName mReceiver = new ComponentName(getApplicationContext(),  
    MyCallReceiver.class);  
PackageManager mManager = getApplicationContext().getPackageManager();  
mManager.setComponentEnabledSetting(mReceiver,  
    PackageManager.COMPONENT_ENABLED_STATE_DISABLED,  
    PackageManager.DONT_KILL_APP);
```

# Bảo mật trong Broadcast Receiver

- Nếu bạn không muốn gửi broadcast trong các ứng dụng khác, cần nhắc sử dụng **LocalBroadcastManager**
- Lớp **LocalBroadcastManager** giúp cho bạn không phải lo về các vấn đề bảo mật liên quan đến việc ứng dụng khác có gửi và nhận broadcast của bạn



# Bảo mật trong Broadcast Receiver

- Intent namespace là global do đó bạn phải đảm bảo rằng tên của intent action và các xâu khác được viết theo đặc trưng riêng của bạn để đảm bảo tránh xung đột với các ứng dụng khác
- Khi bạn sử dụng **registerReceiver(BroadcastReceiver, IntentFilter)**, bất kỳ ứng dụng nào có thể gửi broadcast tới receiver đã được đăng ký
- Khi bạn publish một receiver trong AndroidManifest.xml và xác định intent filter cho receiver, bất kỳ ứng dụng nào cũng có thể gửi broadcast đến cho dù intent filter có nội dung như thế nào

## Bảo mật trong Broadcast Receiver

- Để ngăn cản ứng dụng khác gửi yêu cầu, sử dụng **android:exported="false"**
- Bắt đầu từ ICE\_CREAM\_SANDWICH, bạn có thể hạn chế broadcast tới một ứng dụng cụ thể sử dụng **Intent.setPackage**

```
<receiver
    android:name=".MyCallReceiver"
    android:enabled="true"
    android:exported="false" >
</receiver>
```

# Sticky Broadcast Intent

- Broadcast intent thông thường sẽ bị mất sau khi hệ thống xử lý xong
- Nếu chúng ta sử dụng **Context.sendStickyBroadcastIntent** để gửi broadcast intent thì intent này sẽ không mất đi sau khi hệ thống xử lý
- Chúng ta có thể truy xuất đến intent này bằng phương thức **Context.registerReceiver(BroadcastReceiver, IntentFilter)**
- Ví dụ Intent.ACTION\_BATTERY\_CHANGED là một sticky broadcast intent. Chúng ta có thể truy xuất đến nó bất kỳ lúc nào sau khi nó được gửi



# DEMO

Sử dụng sticky broadcast intent



# Tổng kết nội dung bài học

- Tổng quan về Broadcast Receiver
- Phân loại Broadcast Receiver
- Miêu tả Broadcast Receiver
- Vòng đời Broadcast Receiver
- Hạn chế của Broadcast Receiver

