Create class and extend it from BroadcastReceiver, where before doing something check if it is needed receive

if (“com.example.boris.broadcastreceiver”.equals(intent.getAction())) { … }

then add it dynamically or in manifest

manifest:

<receiver

android:name=".className"

exported=”false”> - can be used by another apps

<intent-filter>  
 <action android:name="android.intent.action.BOOT\_COMPLETED"/> - register action for calling receiver   
 <action android:name="android.intent.action.QUICKBOOT\_POWERON" />  
 </intent-filter>  
</receiver>

Dynamically:

Registration at first:

@Override  
protected void onStart() {  
 super.onStart();  
 //(can use all services, because they will happen only after yor app started, not all at once when you do something)

IntentFilter filter = new IntentFilter("com.example.boris.broadcastreceiver");  
 registerReceiver(classExampleVariable, filter);  
}  
  
@Override  
protected void onPause() {  
 super.onPause();  
 //unregister receiver !important  
 unregisterReceiver(classExampleVariable);  
}

Using in needed part of code

Intent intent = new Intent("com.example.boris.broadcastreceiver"); - value for checking, what was matched at start  
sendBroadcast(intent);

Many of them work only when you set them dynamically

Start personal broadcast from another app

In one app create class extend it from BroadcasrReeiver and register it in manifest like:

<receiver android:name=".ClassName"  
 android:exported="true">  
 </receiver>

Then in another app:

Intent intent = new Intent();  
ComponentName componentName = new ComponentName("Package name",  
 "Package name + class name");  
intent.putExtra("text","suka"); - addictional  
  
intent.setComponent(componentName);  
sendBroadcast(intent);

Start broadcast from this app

Intent intent = new Intent(this, ExampleBroadcastClass.class);  
sendBroadcast(intent);

Get result from broadcast

Create broadcastReceiver in your app and set it in manifest wthout fintent-filter, then when you will be starting broadcast set it like:

sendOrderedBroadcast(intent, null, new YourBroadcastClass(), null, 0, "Start", extras); - your class will start last with result, which you will get by (in another broadcast):

        Bundle resultExtras = getResultExtras(true);

        String resultData = getResultData();

        setResult(resultCode, resultData, resultExtras); - send result to next method