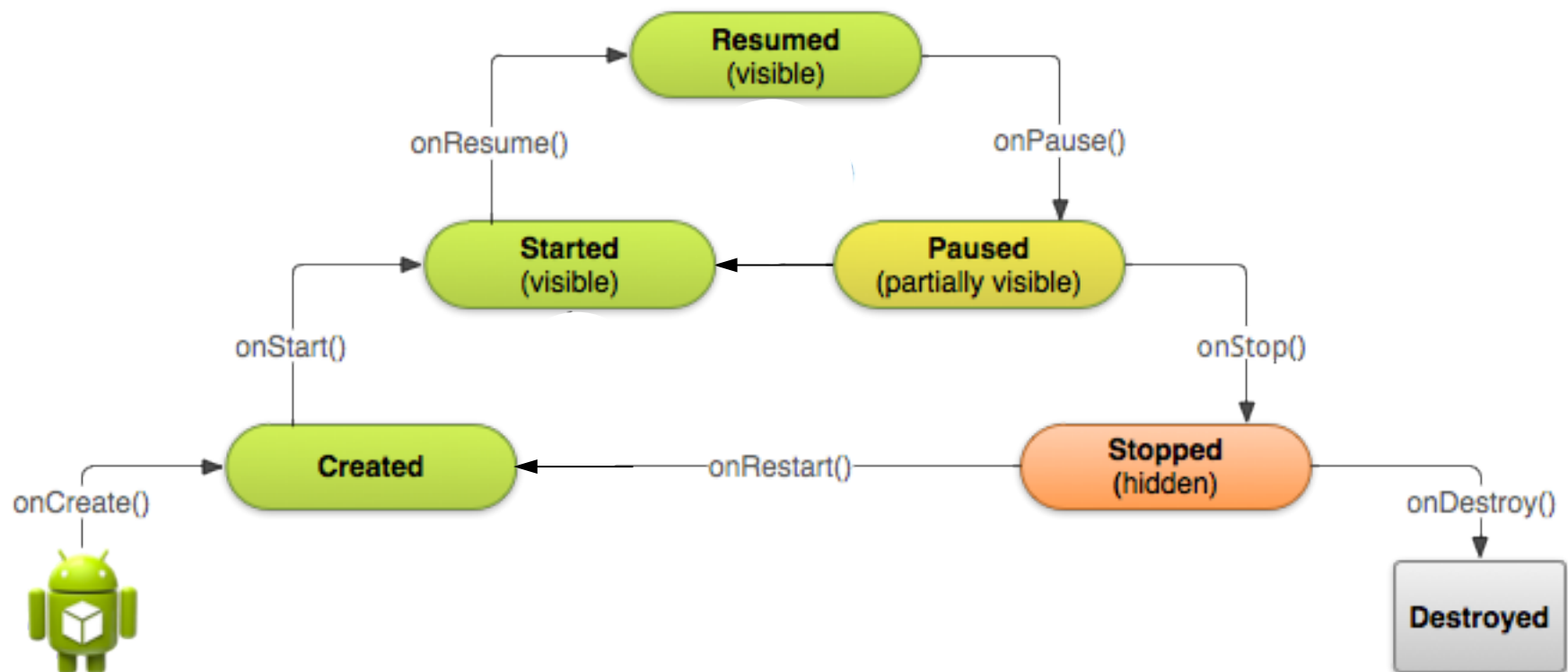
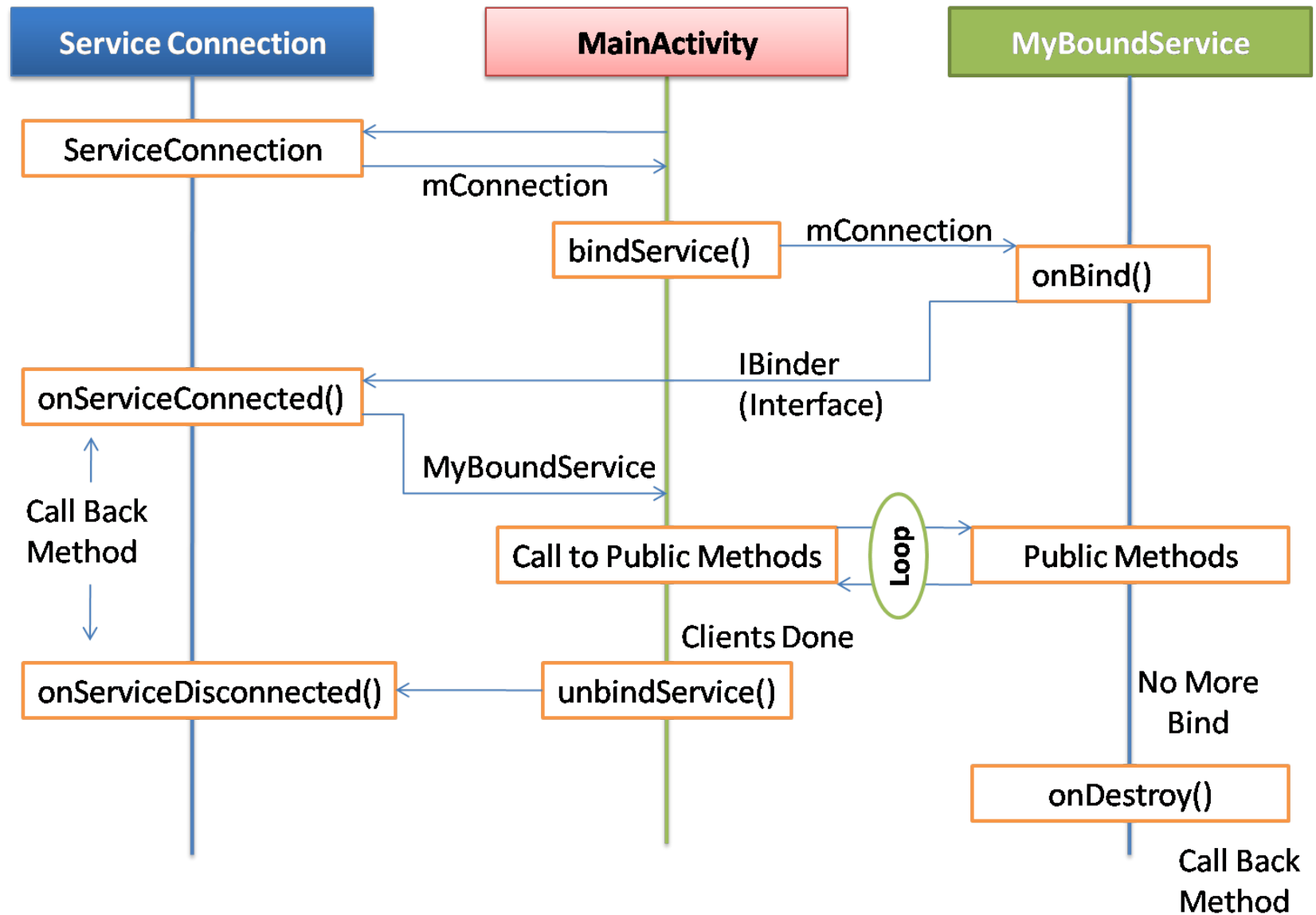


Bound Services & BroadcastReceivers

Activity Lifecycle





Bound Service

- Bound Service
 - extends Service
 - implements onBind()
 - returns IBinder object
 - called only once, by the first client
- Client
 - call bindService()
 - pass ServiceConnection instance
 - with onServiceConnection(): gets IBinder object
- Only works if service and client belongs to the same App
 - Otherwise, use Messenger or AIDL

```
public class MyService extends Service {
    private final IBinder myBinder= new MyBinder();

    public class MyBinder extends Binder {
        MyService getService() {
            return MyService.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        return myBinder;
    }

    public f() {}
    public g() {}
    ....
}
```

```
public class MyActivity extends Activity {
    MyService myService;

    @Override
    protected void onStart() {
        super.onStart();
        Intent intent = new Intent(this, MyService.class);
        bindService(intent, myCon, Context.BIND_AUTO_CREATE);
    }

    @Override
    protected void onStop() {
        super.onStop();
        unbindService(myCon); }

    private ServiceConnection mConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName className, IBinder service) {
            MyBinder binder = (MyBinder) service;
            myService = binder.getService();
        }
        @Override
        public void onServiceDisconnected(ComponentName arg0) {};
    }
}
```

Broadcast Receiver

- Receiver
 - Create an intent filter
 - `new IntentFilter(intent.<action>)`
 - Create a BroadcastReceiver (inner class)
 - implement `onReceive`
 - Register the receiver (in `onResume()`)
 - `registerReceiver(<receiver>, <filter>)`
 - Unregister the receiver in `onPause()`
 - `unregisterReceiver(<receiver>)`

Broadcast Receiver by Manifest

- Create a receiver class extending BroadcastReceiver
 - implement onReceive
- `<receiver android:name=".<receiver class>" android:exported="true">`
 - `<intent-filter>`
 - `<action android:name="android.intent.action.*action*" />`
 - `</intent-filter>`
- `</receiver>`

Send a Broadcast

- Create a class
 - new Intent()
- Set action
 - setAction(<action name>)
- Add extra information
 - putExtra()
- sendBroadcast(<intent>)

Implicit Intent vs Broadcast Receiver

- Implicit Intent
 - Sent by App
 - Only received by one App/Service
 - Wants a specific action to be performed
 - Cause an Activity or Service to react
- Broadcast Receiver
 - Sent by App or System
 - Received by any interested Apps/Services
 - Not asking a specific action to be performed
 - Just notifying the events
 - Cause minor state updates