

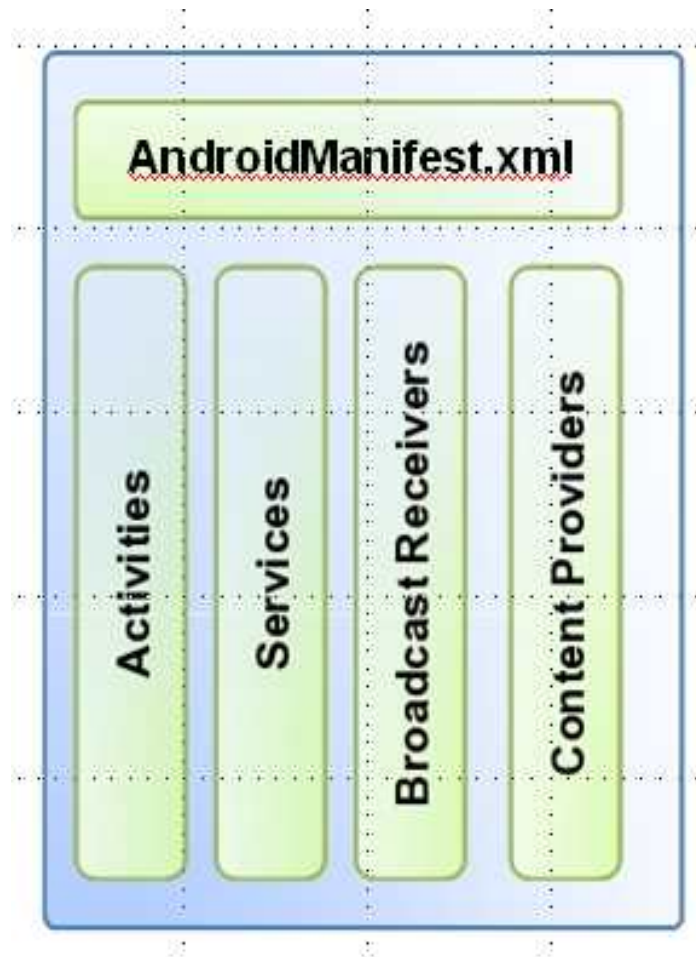
안드로이드의 경계 없는 메시지 전달 및 자료공유를 위한 방법과 보안

국가 소프트웨어 마에스트로
공식 멘토 정승일(netcoya@gmail.com)

안드로이드의 자료 공유의 다양한 관점

- 화면을 공유한다.
- 시스템의 기능을 공유한다.
- 브로드캐스트를 공유한다.
- 자료(데이터, 화일, 네트워크형자료 등)를 공유

안드로이드 어플리케이션의 구조



- Activity
 - UI, 즉 화면을 나타냄
 - `startActivity()`
- Service
 - 화면없고, 생명주기 있다
 - 1:1 , `startService()`
- BroadcastReceiver
 - 화면없고, 생명주기 없다
 - 1:N , `sendBroadcast()`
- ContentProvider
 - `ContentResolver`로 접근
- AndroidManifest.xml

안드로이드 컴포넌트간의 메시지전달, Intent



Activity

- startActivity(i)

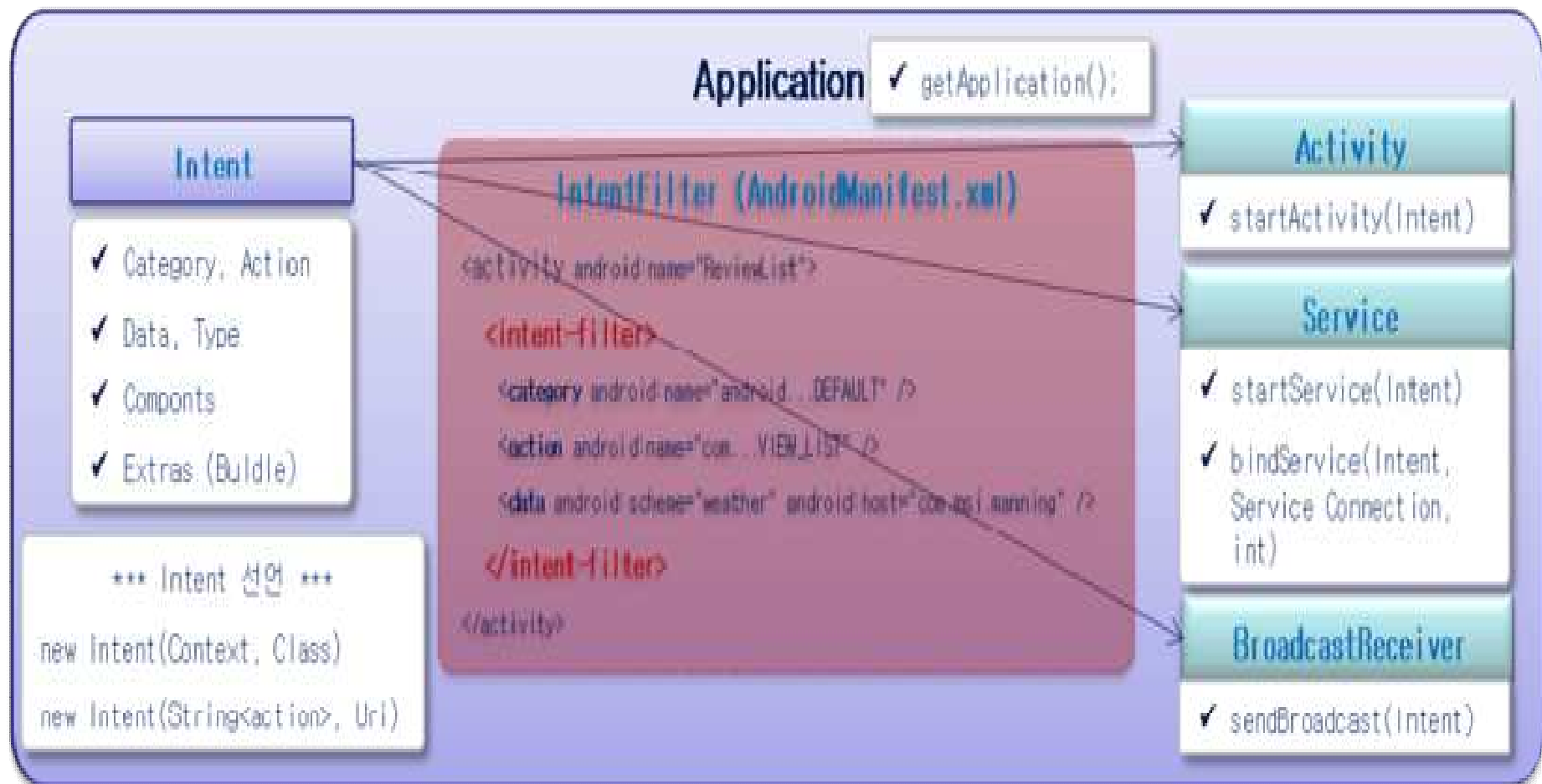
Service

- startService(i)

BroadcastReceiver

- sendBroadcast(i)

Intent를 이용한 컴포넌트간에 메시지 전달 방법



Intent의 Action의 의미에 대해

android.intent.action.VIEW

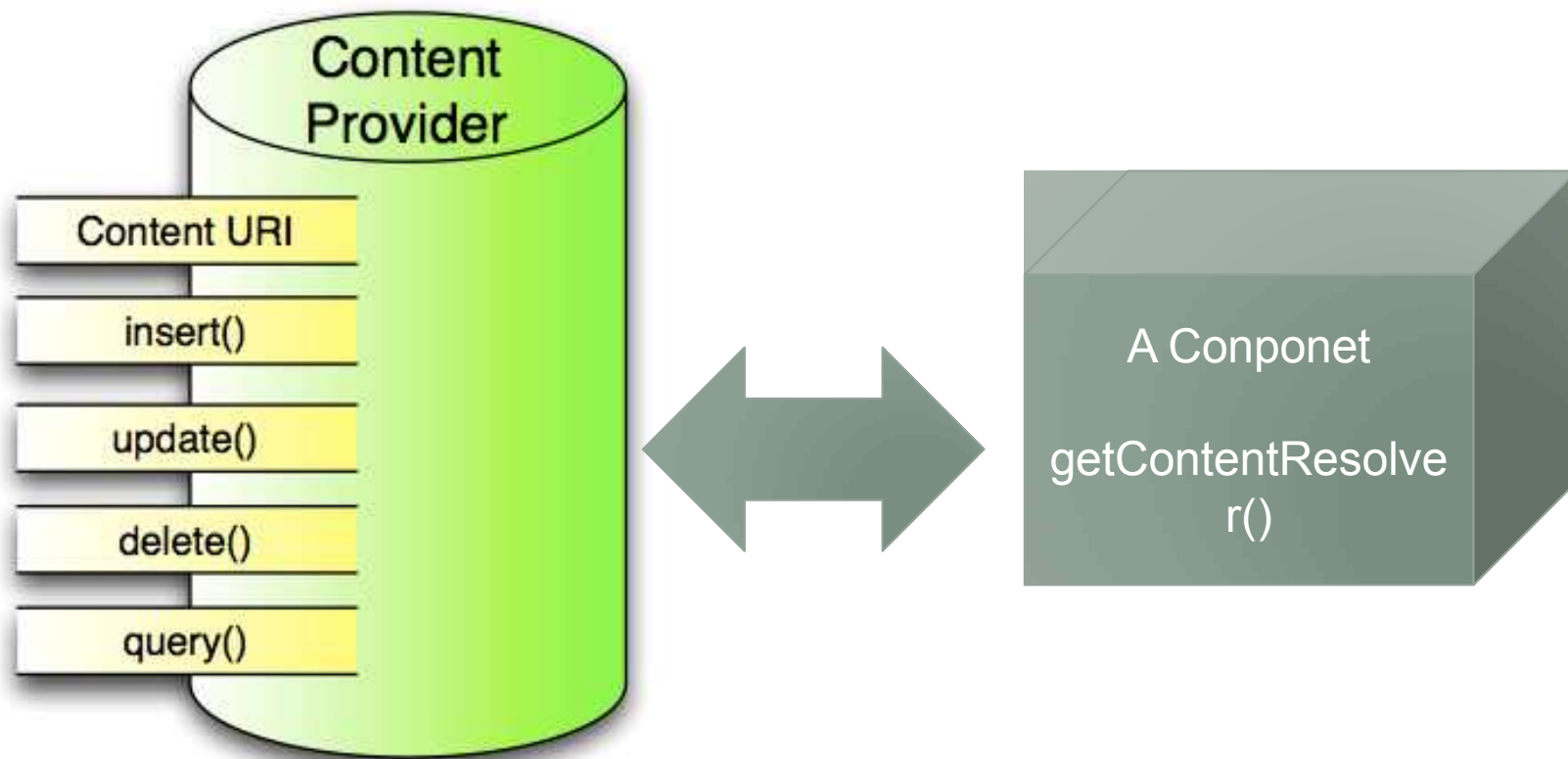
android.intent.action.PICK

android.intent.action.DIAL

...

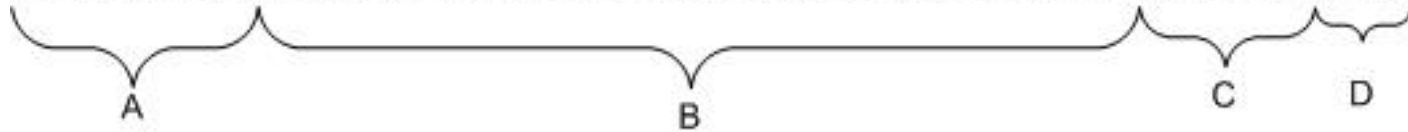


어플리케이션 간의 데이터 공유 ContentProvider



ContentURI

`content://com.example.transportationprovider/trains/122`



A

`content://`

ContentProvider 접근 프로토콜

B

`com.example.transportationprovider`

`androidManifest.xml`

```
<provider android:name=".TransportationProvider"
          android:authorities="com.example.transportationprovider"
          . . . >
```

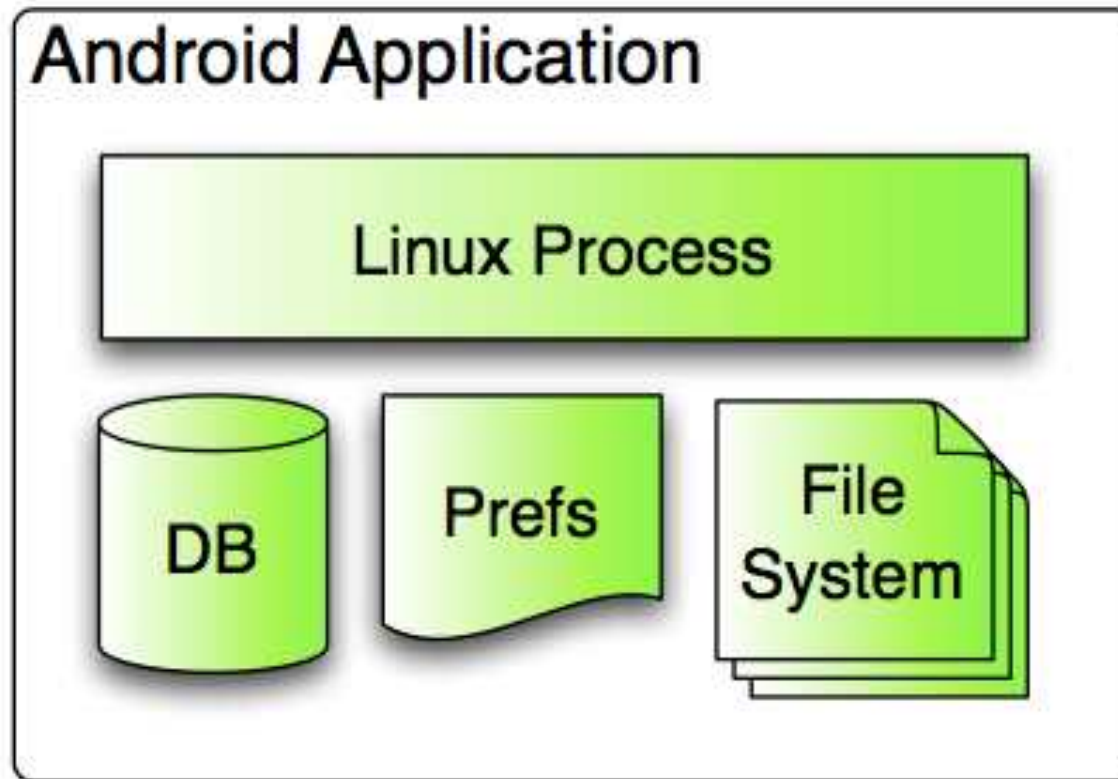
C,D

Extra Informations

HelloContentProvider

시연

경계없는 Android의 보안



어플리케이션의 자료공유에
대한 명시적인 허가가 없다면,
다른 어플리케이션의 접근 불가

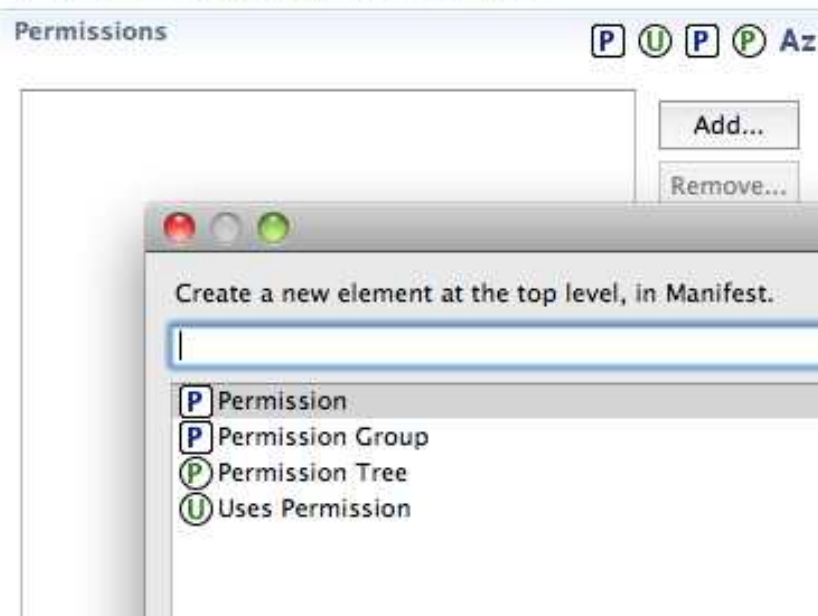
기본적으로, 허가되도록 설정됨

읽기 쓰기 권한에 대해

- 안드로이드 어플리케이션은 각각의 UserID 자동으로 생성
- 그 자료에 접근해서, Read / Write / Execute하는 방식은 Linux의 Permission에 의해서 관리가 된다.
- 어플리케이션은 다른 어플리케이션이 접근할 수 있도록 퍼미션을 설정할 수 있다.
- 그 어플리케이션이 가진 퍼미션을 만들고, 이 퍼미션을 적용하고, 그 적용된 퍼미션을 이용하는 3단계 과정이 중요.

Permission 만들기

Android Manifest Permissions



Attributes for Permission

[P] The `permission` tag declares a security permission that can be used to control access from other packages to specific components or features in your package (or other packages).

Name	com.gkiil.permission.ReadTest	Browse...
Label		Browse...
Icon		Browse...
Permission group		Browse...
Description		Browse...
Protection level	dangerous normal dangerous signature signatureOrSystem	

<permission .. />

```
<permission android:protectionLevel="dangerous"  
    android:name="com.gkiil.permission.ReadTest"  
    android:description="@string/description"/>
```

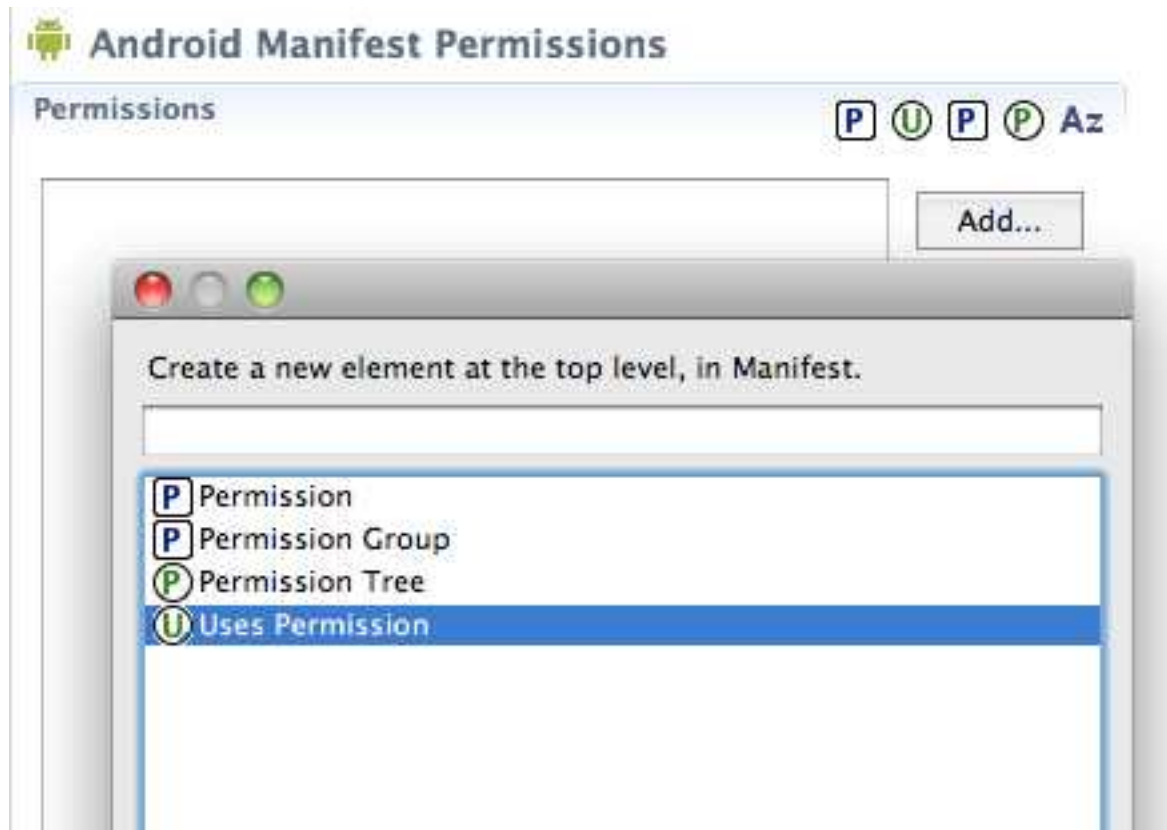
Permission을 컴포넌트에 적용하기



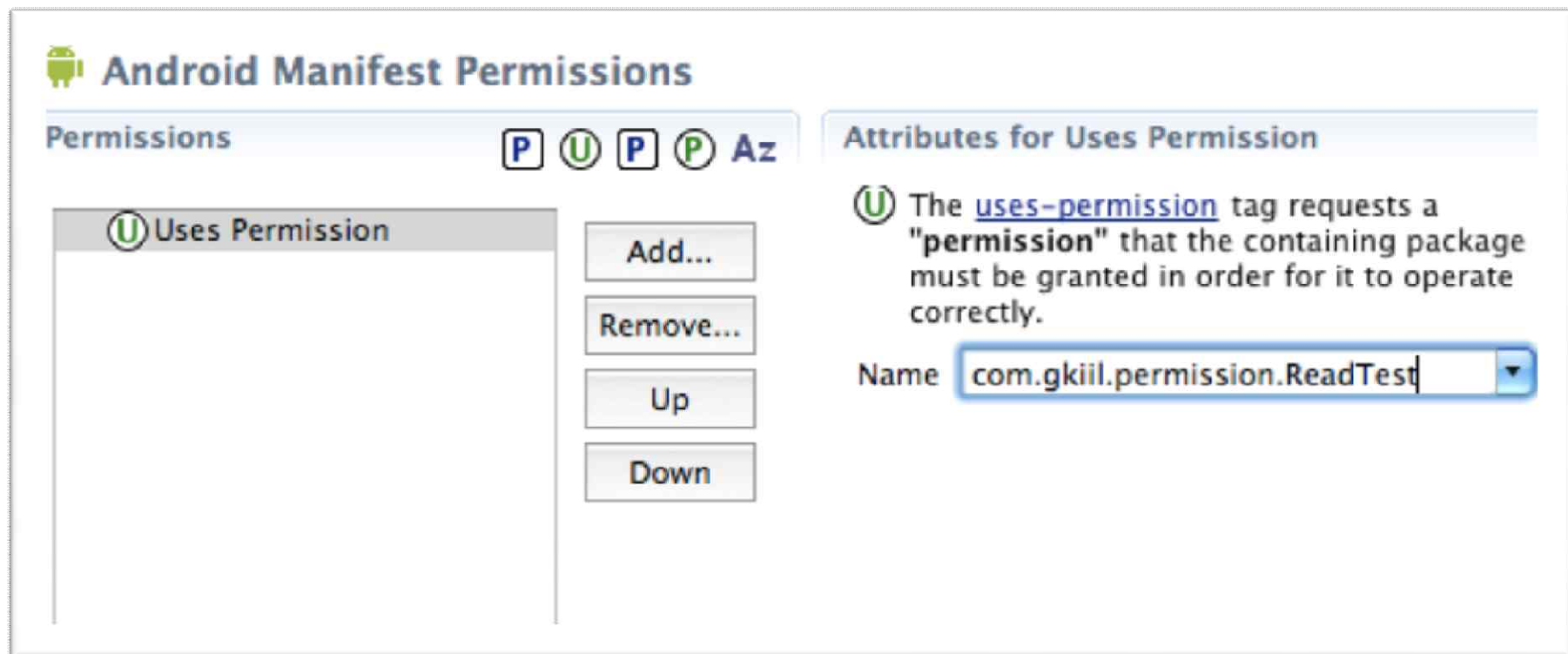
```
<application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".HelloContentProvider" android:label="@string/app_name"
        android:permission="com.gkiil.permission.ReadTest">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <provider android:name="MyContentProvider"
        android:authorities="com.gkiil.contentProivider.test"
        android:readPermission="com.gkiil.permission.ReadTest"></provider>

</application>
```

다른 어플리케이션에서 컴포넌트에 접근하기 - 퍼미션 사용하기



다른 어플리케이션에서 컴포넌트에 접근하기 - 퍼미션 사용하기



```
<uses-permission android:name="com.gkiil.permission.ReadTest">  
</uses-permission>
```

결론

- 안드로이드는 경계없는 컴포넌트간의 공유를 위한 디자인
- 컴포넌트를 공유하기 위해, 인텐트를 사용하는 방법
 - startActivity, startService, sendBroadcast 메소드를 이용
- 어플리케이션의 자료를 공유하기 위해서,
 - ContentProvider 객체가 필요
 - 접근을 위해서 먼저 Content URI 만들고,
 - ContentResolver 객체로 접근함
 - 접근해서, insert, delete, update, query를 통해 자료를 제어함
- 컴포넌트와 어플리케이션의 공유를 위한 퍼미션 적용 3단계,
 - Permission 생성
 - Permission 적용
 - Permission 사용

감사합니다.

Q & A