

시작하기

developer.android.com

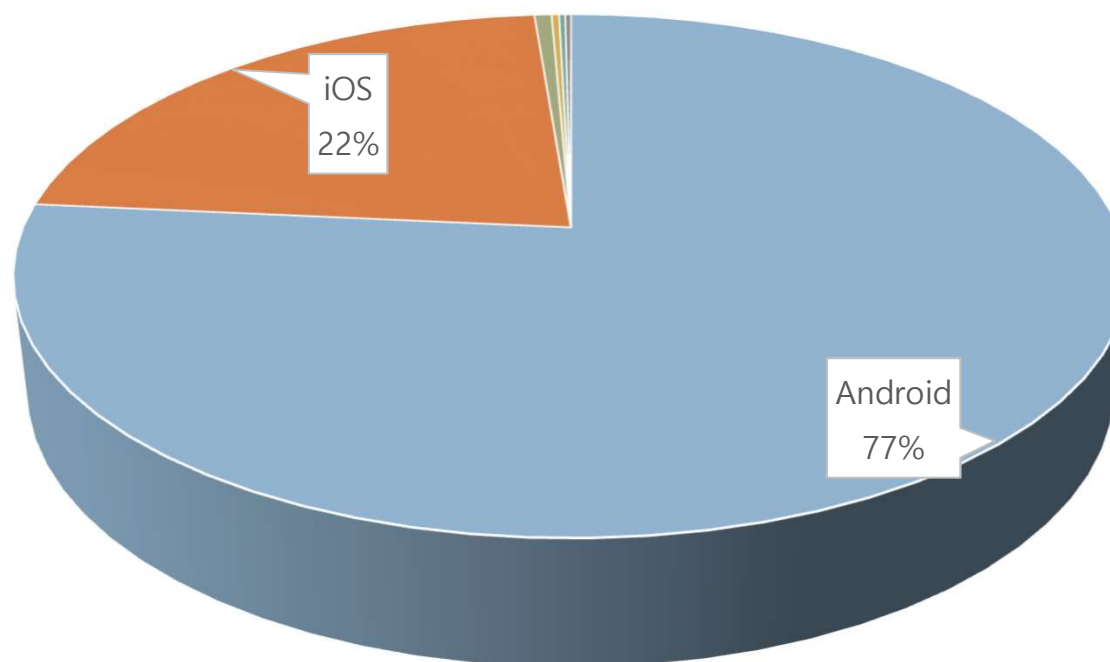
2

The screenshot shows the developer.android.com website. The header includes the Android Developers logo and navigation links for Platform, Android Studio, Google Play, Jetpack, Kotlin, Docs, and News. A search bar is located below the navigation. The main content area features a large banner with the text "build anything on android" and a search bar. Below the banner, there are three featured sections:

- FEATURED Kotlin/Everywhere**: A global series of community-led events - supported by Google and JetBrains - where you can learn more about Kotlin, including using it in Android, Google Cloud Platform, and multi-platform development. A "FIND OUT MORE" button is present.
- FEATURED Android needs your feedback**: Shape the future of Android by participating in a Google User Experience Research study. Sign up, participate, and get rewarded. A "SIGN UP" button is present.
- FEATURED Android Studio 3.5 Stable**: This release is the culmination of months of focus exclusively on fundamentals - fixing hundreds of critical and annoying bugs, from memory leaks to UI freezes to Lint analysis performance, and focusing on product stability along with some new features. Download it today and give it a spin! A "DOWNLOAD ANDROID STUDIO 3.5" button is present.

Mobile Operating System Market Share Worldwide - August 2019

3						
Android	iOS	KaiOS	Unknown	Samsung	Windows	
76.22%	22.17%	0.59%	0.26%	0.21%	0.2%	



■ Android ■ iOS ■ KaiOS ■ Unknown ■ Samsung ■ Windows

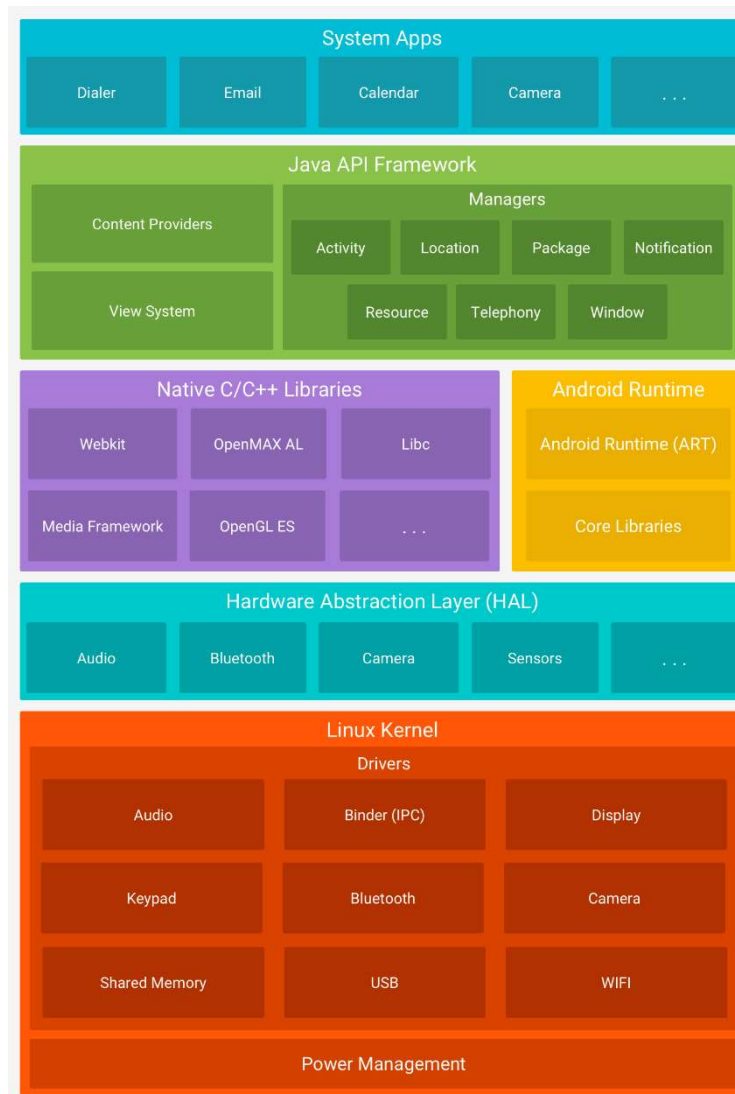
플랫폼 버전

4

Version ↕	Code name ↕	Release date ↕	API level ↕	Runtime ↕	First devices to run version ↕
9	Pie	August 6, 2018	28	ART	Essential Phone, Pixel, Pixel XL, Pixel 2, Pixel 2 XL, Nokia 7 Plus, OnePlus 6, Oppo R15 Pro, Sony Xperia XZ2, Vivo X21UD, Vivo X21, Xiaomi Mi Mix 2S ^[371]
8.1	Oreo	December 5, 2017	27	ART	Pixel, Pixel XL, Nexus 6P, Nexus 5X
8.0		August 21, 2017	26	ART	N/A
7.1	Nougat	October 4, 2016	25	ART	Pixel, Pixel XL
7.0		August 22, 2016	24	ART	Nexus 5X, Nexus 6P
6.0	Marshmallow	October 5, 2015	23	ART	
5.1	Lollipop	March 9, 2015	22	ART	Android One
5.0		November 3, 2014	21	ART 2.1.0	Nexus 6, Nexus 9
4.4	KitKat	October 31, 2013	19	Dalvik (and ART 1.6.0)	Nexus 5
4.3	Jelly Bean	July 24, 2013	18	Dalvik	Nexus 7 2013
4.2		November 13, 2012	17	Dalvik	Nexus 4, Nexus 10
4.1		July 9, 2012	16	Dalvik	Nexus 7
4.0	Ice Cream Sandwich	October 19, 2011	15	Dalvik	Galaxy Nexus
2.3	Gingerbread	February 9, 2011	10	Dalvik 1.4.0	Nexus S
Legend: Old version Older version, still supported Latest version Latest preview version Future release					

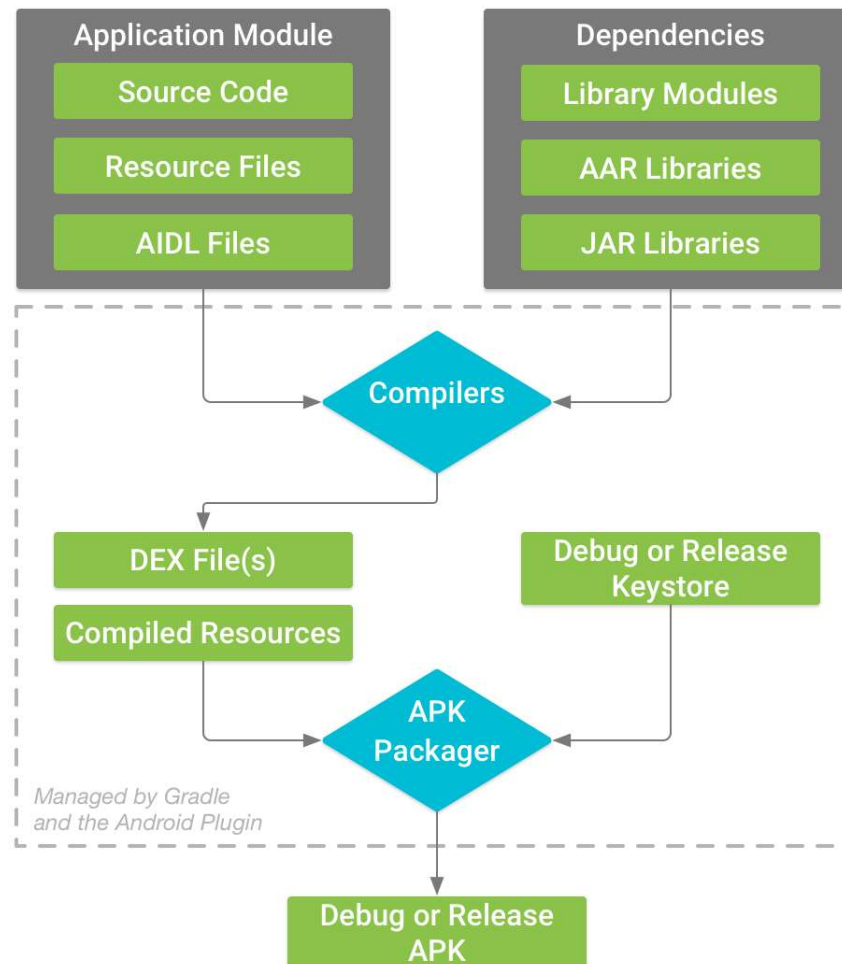
플랫폼 아키텍처

5



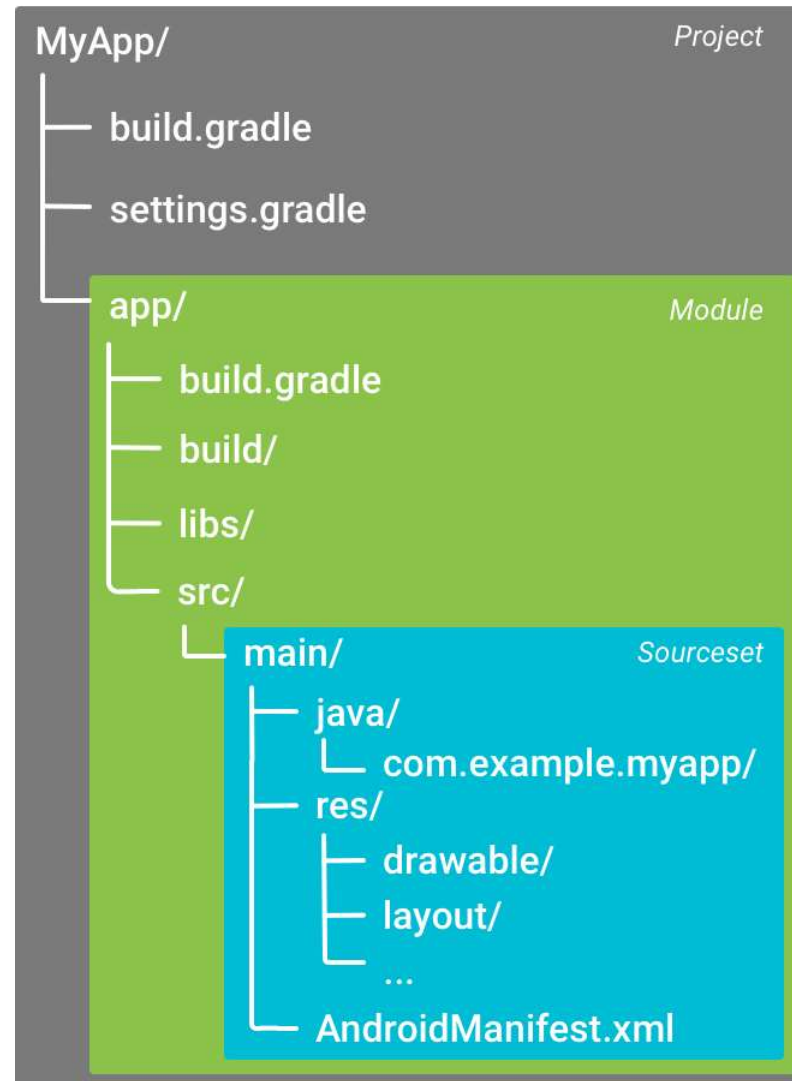
빌드 프로세스

6



Android 앱 모듈의 기본 프로젝트 구조

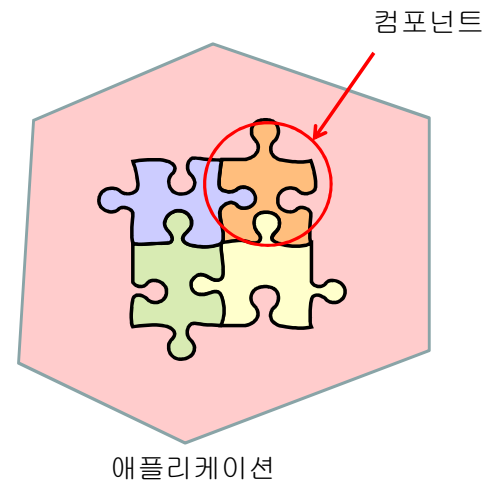
7



컴포넌트 기반 개발

8

- 애플리케이션은 컴포넌트로 이루어진다.
 - ▣ 액티비티(activity)
 - ▣ 서비스(service)
 - ▣ 방송 수신자(broadcast receiver)
 - ▣ 콘텐츠 제공자(content provider)



컴포넌트 기반 개발 - 액티비티

9

- 사용자 인터페이스 화면을 가지는 하나의 작업
- Activity 클래스 상속



액티비티

화면을 통하여
사용자를 상대
합니다.



액티비티

컴포넌트 기반 개발 - 액티비티

10

- 액티비티들이 모여서 애플리케이션이 된다.



이메일 애플리케이션

이메일 애플리케이션은 3개의 액티비티로 이루어진다.

컴포넌트 기반 개발 - 서비스

11

- 백그라운드에서 실행되는 컴포넌트로서 오랫동안 실행되는 작업이나 원격 프로세스를 위한 작업
 - ▣ Service 클래스 상속
 - ▣ (예) 배경 음악을 연주하는 작업



미디어 플레이어 애플리케이션

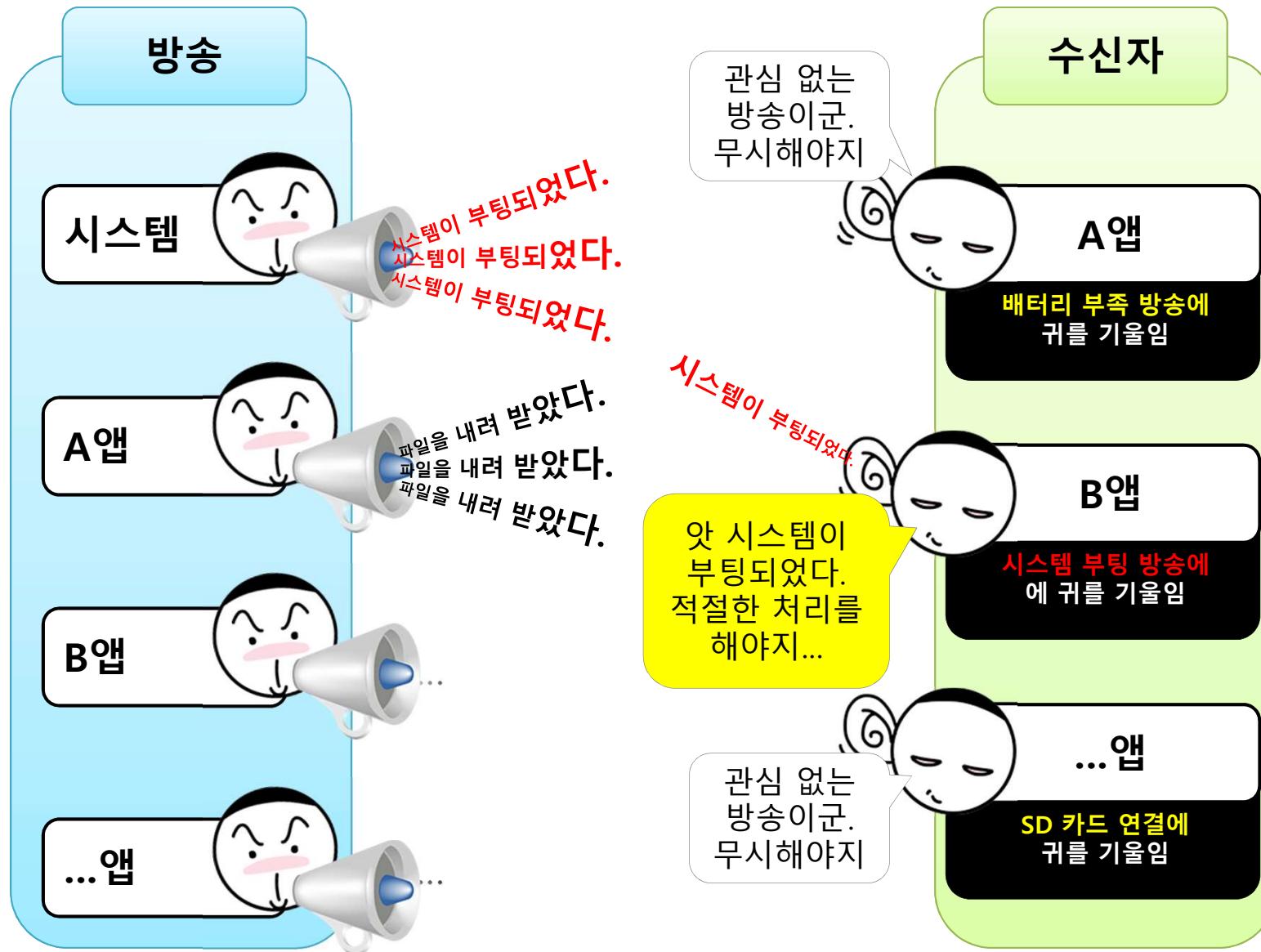
컴포넌트 기반 개발 - 방송 수신자

12

- 방송을 받고 반응하는 컴포넌트
 - ▣ BroadcastReceiver 클래스 상속



컴포넌트 기반 개발 - 방송 수신자



컴포넌트 기반 개발 - 콘텐츠 제공자

14

- 데이터를 관리 & 다른 앱에게 데이터를 제공하는 컴포넌트
 - ▣ ContentProvider 클래스 상속



전화번호부 애플리케이션

컴포넌트 기반 개발 - 콘텐츠 제공자

15

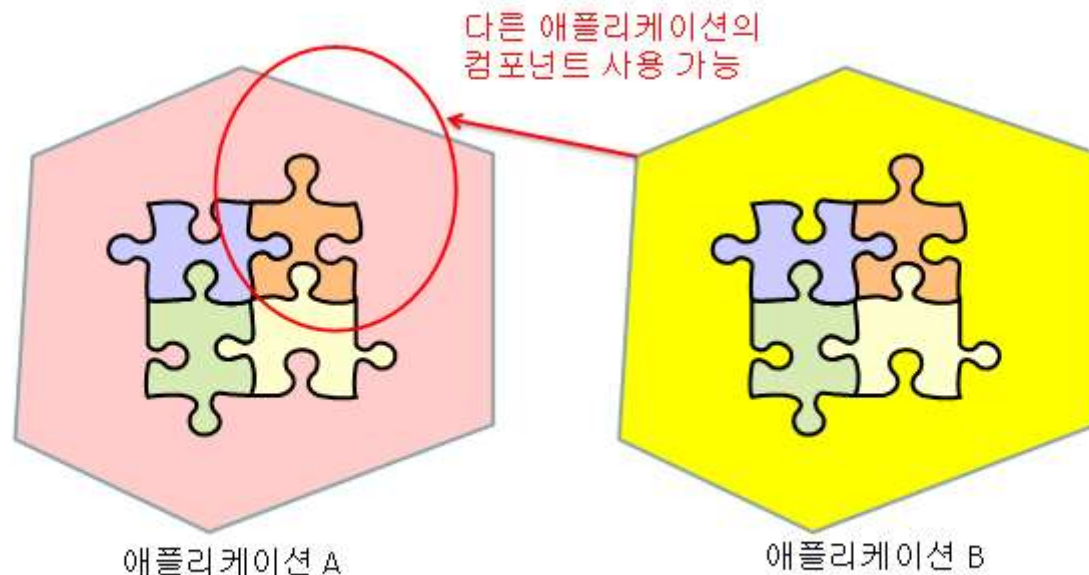
- 콘텐츠 제공자는 외부에 데이터를 공유할 때 사용한다.
 - ▣ 앱의 데이터를 외부에 공개하지 않는다면, 콘텐츠 제공자를 사용할 필요는 없다.



컴포넌트 기반 개발

16

- 안드로이드 앱은 single entry point를 갖지 않는다.
 - ▣ 내부에 포함된 컴포넌트들을 통하여 여러 갈래로 실행될 수 있다.

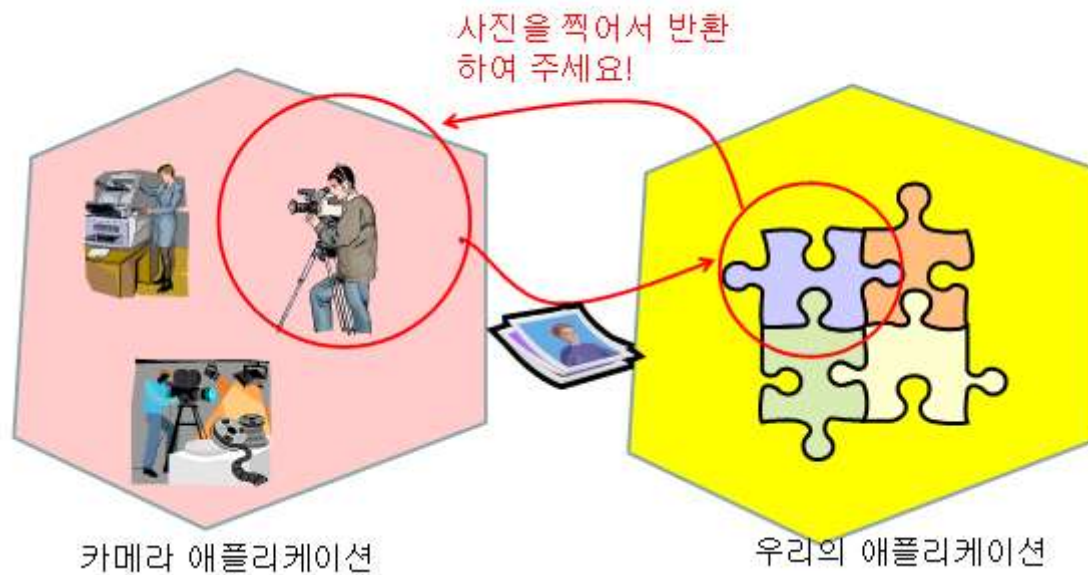


안드로이드에서는 다른 애플리케이션이 가지고 있는 컴포넌트를 사용할 수 있다.

컴포넌트 기반 개발

17

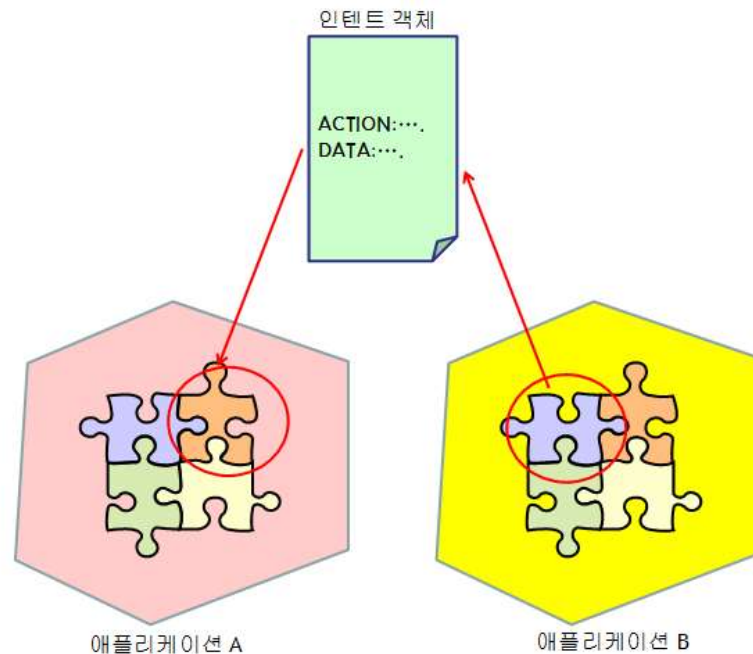
- 애플리케이션에서 사용자가 사진을 촬영하도록 하고 싶은 경우



컴포넌트 기반 개발 - 인텐트

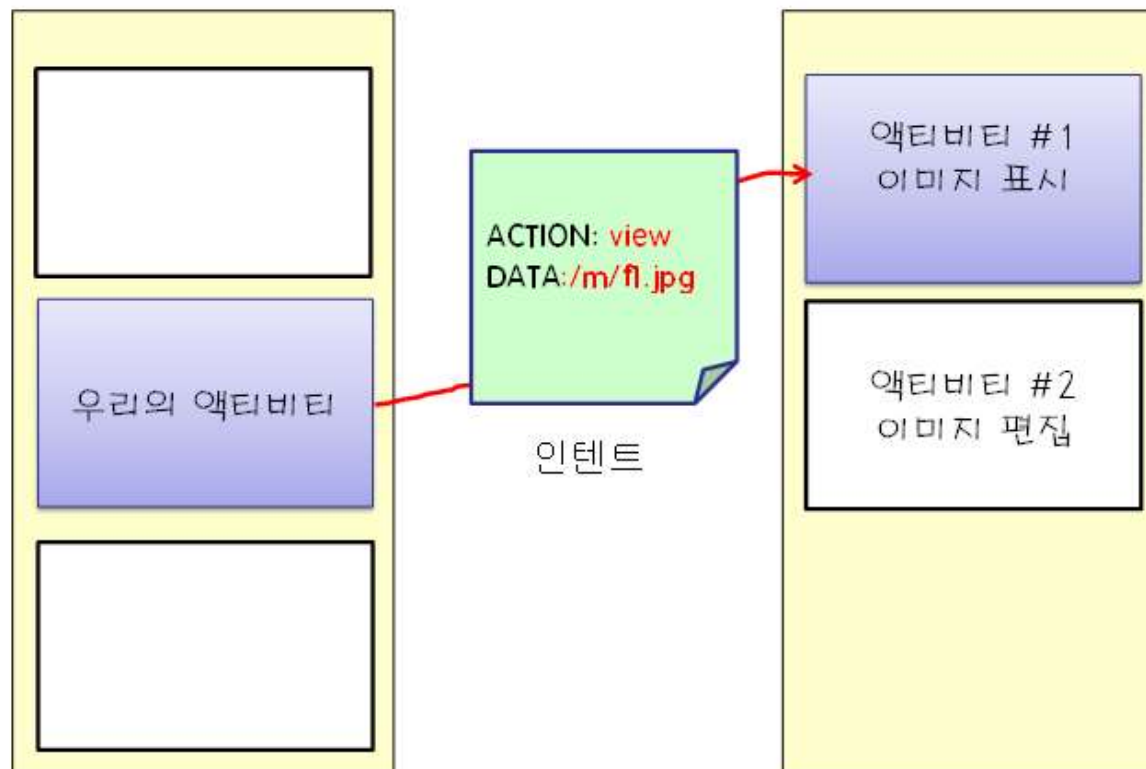
18

- 애플리케이션의 의도를 적어서 안드로이드에 전달하면 안드로이드가 가장 적절한 컴포넌트를 찾아서 활성화하고 실행
 - ▣ Intent 클래스의 객체
- 외부 앱에서 콘텐츠 제공자를 사용하기 위해서는 콘텐츠 리졸버(ContentResolver) 객체를 사용해야 한다.



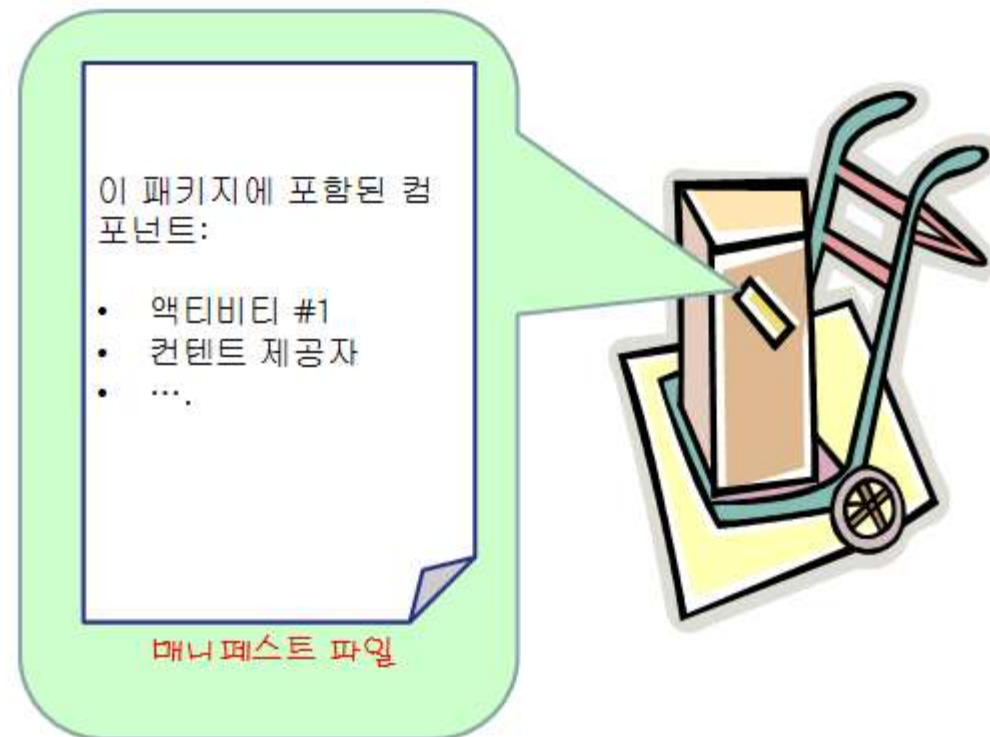
컴포넌트 기반 개발 - 인텐트

19



AndroidManifest.xml

20



AndroidManifest.xml

21

- XML을 사용한다.
 - <activity> 요소 : 액티비티 선언
 - <service> 요소 : 서비스 선언
 - <receiver> 요소 : 방송 수신자
 - <provider> 요소 : 콘텐츠 제공자

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.user.androidlab">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

리소스 외부화를 이용한 개발

22

□ 코드 내부에서:

```
msgTextView.setText(R.string.hello);
```

```
getWindow().setBackgroundDrawableResource(R.drawable.graphic);
```

```
public final class R {  
    ...  
    public static final class string {  
        ...  
        public static final int hello=0x7f050001;  
        ...  
    }  
}
```

□ XML 내부에서:

```
<?xml version="1.0" encoding="utf-8"?>  
<EditText xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:textColor="@color/opaque_red"  
    android:text="@string/hello" />
```

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <color name="opaque_red">#f00</color>  
    <string name="hello">Hello!</string>  
</resources>
```

```
MyProject/  
    src/  
        MyActivity.java  
    res/  
        drawable/  
            graphic.png  
        layout/  
            main.xml  
            info.xml  
        mipmap/  
            icon.png  
        values/  
            strings.xml
```