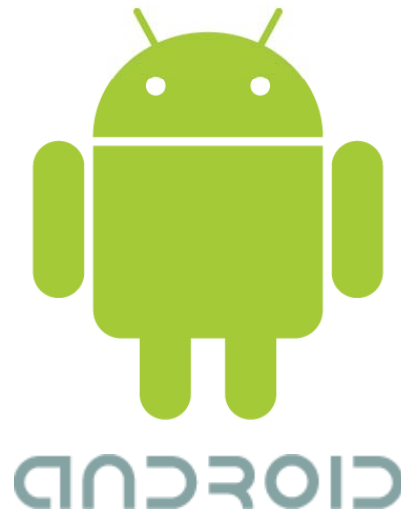


대학생(또는 안드로이드 초보자)을 위한 강의노트

# 안드로이드 교과과정 개요



2009. 9. 3.

[www.kandroid.org](http://www.kandroid.org) 운영자 : 양정수 (yangjeongsoo@gmail.com)

출처 : [www.kandroid.org](http://www.kandroid.org) → (탭메뉴)북스토어 → (왼쪽메뉴)대학생을 위한 강의노트

# 목 차

새로움에 대한 도전, 그 아름다움과 가치

오픈 소스의 시대

이동통신 산업

안드로이드 App. 개발 준비

안드로이드 App. 교육 (개요)





우리가 iPhone에 주목하는(했던) 이유는?

- Hardware 사양
- 사용자 인터페이스
- 탑재된 Software
- iPod 성공 이후의 애플 야심작에 대한 기대
- 폰 디자인 프로세스의 변화 또는 혁신



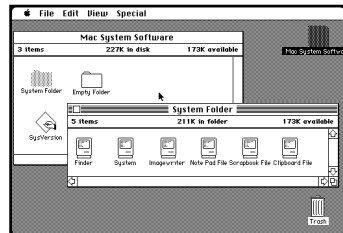
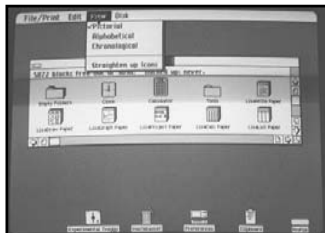
## 새로움에 대한 도전, 그 아름다움과 가치 : iPhone (간단한 역사)



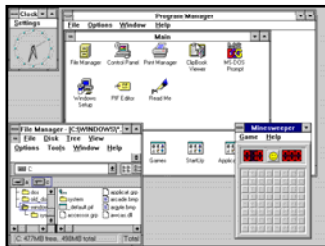
### - 1973년 : Xerox Alto - GUI 최초 구현 (Icon, Window, Mouse, etc)

※ Small Talk 란 프로그래밍언어를 개발한 것으로 유명한 앨런 케이는 1972년 제록사의 PARC(Palo Alto Research Center)에서 GUI에 대한 개념과 기술 구현에 대한 연구를 진행하였음. 케이는 1960년대 후반부터 맥루한의 미디어 이론과 제록 부루너의 인지이론을 정목해 어린이가 컴퓨터와 경험을 공유할 수 있는 환경으로 GUI를 제안했다. GUI는 문자를 대신해 정보를 전달하는 아이콘, 스크린을 효율적으로 활용하는 중첩 윈도우, 더글러스 엥겔바트가 개발한 마우스라는 입력장치로 구성되었다.

- 1976년 : Apple사 설립 및 Apple I 출시, 1977년 : Apple II 출시 (큰 성공을 거둠)
- 1979년 : Steve Jobs Xerox의 PARC 방문, 알토컴퓨터를 보고,  
차세대 컴퓨터에 적용할 '데스크탑 메타포'란 개념을 만들어 냄.
- 1983년 : Apple사 LISA 출시 - 현재 존재하는 GUI를 사실상 완성 (사업적으로는 실패)



- 1984년 : 애플사 Macintosh 출시 (큰 성공을 거둠)
- 1986년 : Steve Jobs Apple사를 떠남  
NeXT computer 설립  
(→ WWW, Doom, MacOS X)



- 1990년 : 마이크로소프트 윈도우 3.0 출시
- 1992년 : 마이크로소프트 윈도우 3.1 출시
- 1996년 : 애플사 NeXT 매입, 1997년 : Steve Jobs Apple CEO 복귀
- 2001년 : iPod 출시, 2003년 : iTunes, 2005년 iTunes Video



출처 : [http://www.youtube.com/watch?v=D5VTB7Lj\\_NA](http://www.youtube.com/watch?v=D5VTB7Lj_NA)



Alan Kay

## 새로움에 대한 도전, 그 아름다움과 가치 : **Android** (개발자 챌린지 공지)



출처 : <http://www.youtube.com/watch?v=XPD0cshaMy4>

# 새로움에 대한 도전, 그 아름다움과 가치 : Android (간단한 역사)



## Android 관련 주요 일정

2005.8	Google, Android Inc. 를 매입함.
2005.11	Google, Skia 를 매입함.
2007.10	Google, Jaiku Inc. 를 매입함.
2007.11.5	OHA 설립, 모바일 디바이스에 대한 오픈 스탠다드 개발 착수
2007.11	Ascender Corp, OHA를 위한 Droid Font Collection 공급
2007.11.12.	Google, Android SDK 릴리즈 Google, Android ADC 발표. (총 1,000만 달러 상금규모)
2007.12.3.	Google, 700MHz 주파수 경매 참여
2008.1.24.	美FCC, 700MHz 주파수 경매 시작
2008.1.28.	Google, Android ADC 일정 연기 발표
2008.2.13.	Google, Android SDK m5-rc14 릴리즈
2008.2.11~14.	The Mobile World Congress 2008 Barcelona Texas Instruments, Google Android 프로토타입 시연







Introducing T-Mobile G1™ with Google™



2008년 9월 23일

Introducing T-Mobile G1™ with Google™





Android 1.1



Android 1.5  
(Cupcake)



Android 1.6(?)  
(Donut)

<



Android 2.0  
(Eclair and Flan)

2008.09.23 : 안드로이드 1.0 SDK Release 1 배포

2008.10.21 : 안드로이드 전체 소스 공개

2008. 4/4 분기

- **Localization** : 문자열, 사용자 인터페이스, 날짜 및 숫자 표기
- Support for **multiple APNs**
- SIM application toolkit (**STK**)

2009. 1/4 분기

- Input method framework (**IMF**)
- Input method engines (**IME**)

2009. 02 : 안드로이드 1.1 SDK R1 배포

2009. 04 : 안드로이드 1.5 SDK R1, R2, R3 배포

2009. 06 : 안드로이드 1.5 NDK R1 배포



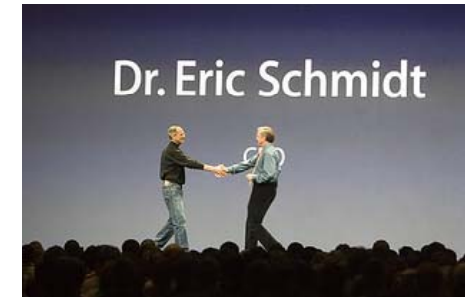


1964년 『미디어의 이해』라는 책을 통해  
'옷은 피부의 확장, 자동차의 바퀴는 다리의 확장, 책은 눈의 연장이며,  
전기는 중추신경의 연장' 이라고 언급했던 **마셜 맥루한**은  
'모든 기술을 인간기능(function)의 확장'이라고 정의했다.  
더불어 '미디어는 감각기능의 확장'이라고 정의하면서  
그 정세도(精細度/definition)와 참여도(參與度/participation)의 정도에 따라  
**Hot media**와 **Cool media**로 구분했다.



일반적으로 핫 미디어는 높은 정세도를 가지기 때문에 정보를 접하는 사람은 큰 상상력을 동원할 필요가 없다.  
반면에 쿨 미디어는 정세도가 낮은 대신 참여도는 높은 매체이다.

오늘날 수많은 매체들이 핫 미디어에서 쿨 미디어로 빠르게 진행 중에 있다.  
대표적인 것이 모바일, 휴대전화일 것이다.  
스마트폰의 쌍두마차, iPhone과 Android는 여전히 모바일의 화두이다.  
누군가는 아이폰은 핫미디어에, 구글폰은 쿨미디어에 비유했다.

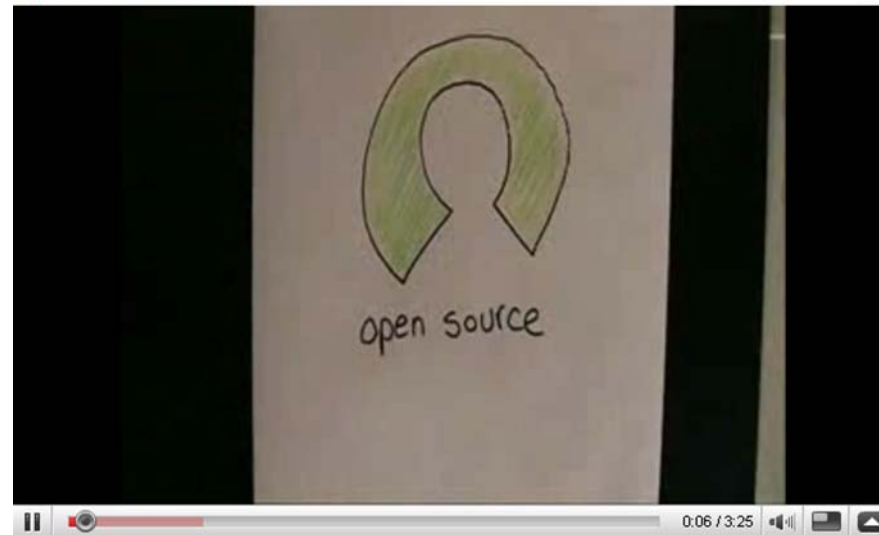


스스로를 혁신할 수 있고 그것을 통해 시장을 혁신할 수 있다고 믿는 애플 **iPhone**과,  
공개 시장을 통해 혁신의 가치를 수렴해서 궁극적인 혁신을 이루고자 하는 구글과 OHA의 **Android**는  
우리가 분명 주목할 만한 것이다.



**Richard Stallman**

Open Source Story Book



출처 : <http://www.youtube.com/watch?v=lxuXNM8a-n8>



**Linus Torvalds**



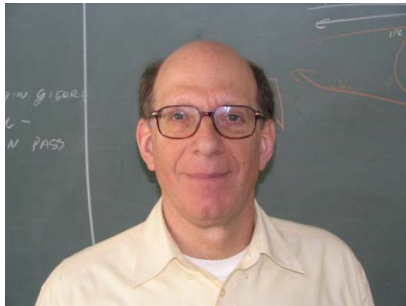
출처 : <http://www.freesoftwaremagazine.com/node/2564/pdf>



- Freedom 0:  
the freedom to run the program as you wish.
- Freedom 1:  
the freedom to study the source code, and change it to make the program do what you wish.
- Freedom 2:  
the freedom to redistribute copies of the program when you wish.
- Freedom 3:  
the freedom to distribute copies of your modified versions when you wish.



Ken Thompson & Dennis Ritchie



Andrew Stuart Tanenbaum

Tech Talk: Linus Torvalds on git



## comp.os.minix

[What would you like to see most in minix?](#) 토론으로부터 온 메시지

> [Linus Benedict Torvalds](#) [프로필 보기](#) [한국어로 번역](#)

[옵션 숨기기](#) 1991년8월26일, 오후3시12분

뉴스그룹: comp.os.minix

보낸사람: torva...@klaava.Helsinki.FI (Linus Benedict Torvalds)

날짜: 25 Aug 91 20:57:08 GMT

현지시간: 1991년8월26일(월) 오전5시57분

제목: What would you like to see most in minix?

[작성자에게 답글](#) | [전달](#) | [인쇄](#) | [메시지목록 보기](#) | [원문 보기](#) | [메시지 신고](#) | [작성자의 메시지 찾기](#)

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torva...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

[작성자에게 답글](#) | [전달](#) | [Report spam](#) | 게시물 평가: ☆☆☆☆☆





Richard Stallman



Linus Torvalds

구분	Copyleft
Apache License	No
BSD License	No
GPL	Yes
LGPL	Yes
MIT License	No
MPL	Yes
Common Public License	Yes
Eclipse Public License	No

Android is **open source**.

Project	Project
bionic	hardware/libhardware
bootloader/legacy	hardware/ril
build	kernel
dalvik	prebuilt
development	recovery
frameworks/base	system/bluetooth
frameworks /policies/base	system/core
system/extras	system/wlan/ti

External Project  
Packages  
Toolchain  
Emulator, Eclipse Plugin

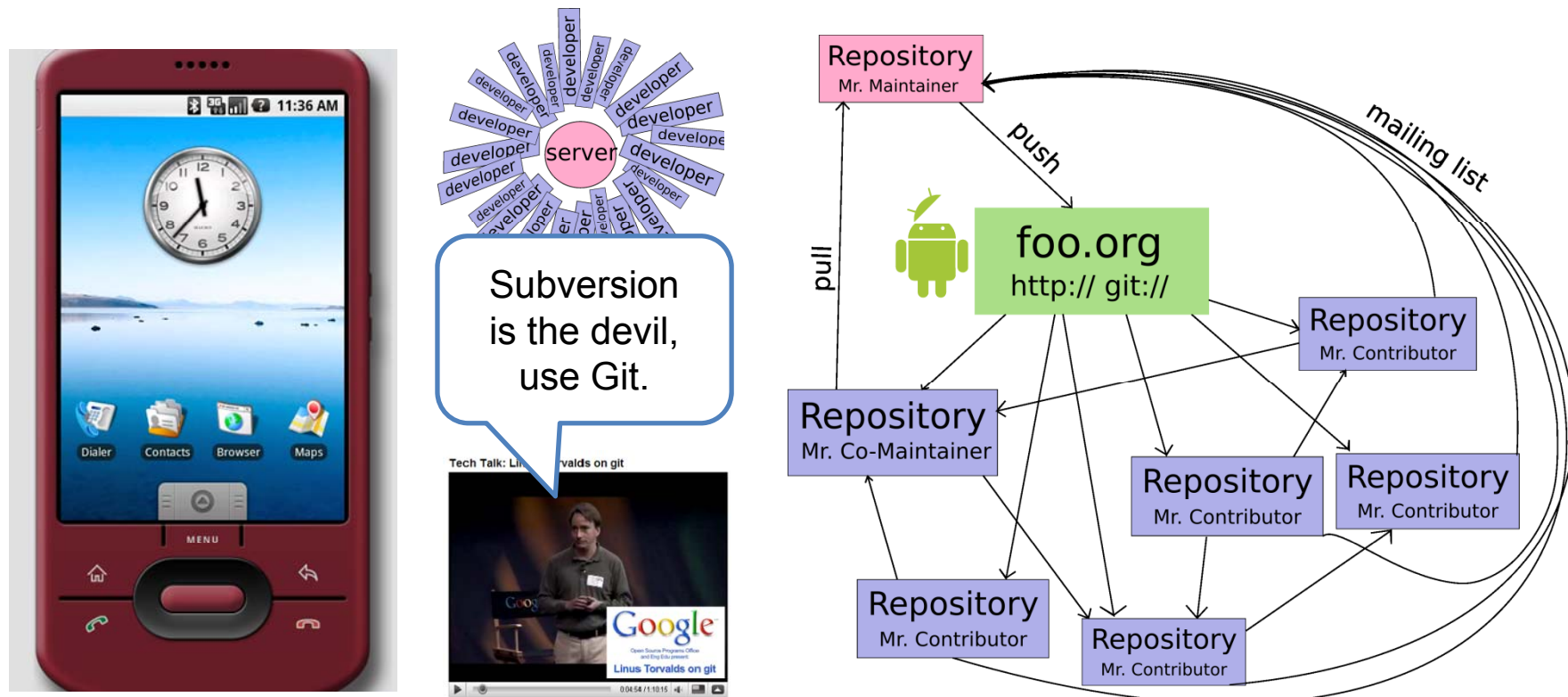
## 오픈 소스의 시대 : 우리가 오픈 소스 시대에 산다는 것은?



담론이란, 어떤 특수한 주제를 말하고, 생각하고 재현하는 방식임.

“담론의 생산을 통제하고, 선별하고, 조직화하고 나아가 재분배하는 일련의 과정들” 을 분석함.

- 미셸푸코, 1970년, 콜레주 드 프랑스 취임연설, 담론의 질서 中 에서

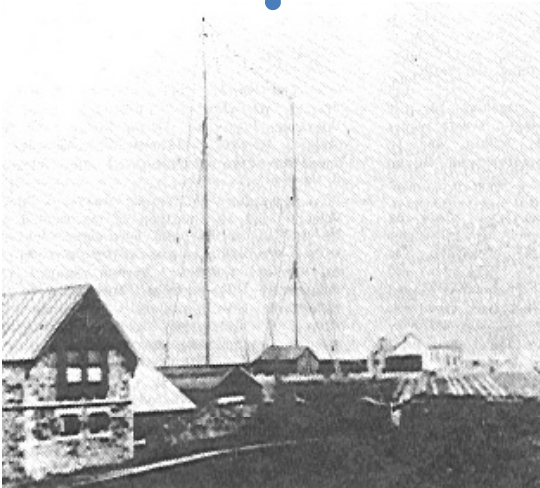


상세보기 : <http://www.kandroid.org/board/board.php?board=androidmarket&command=body&no=10>





dot-dot-dot(S)



Marconi's antenna system at Poldhu, Cornwall, December 1901. (John Belrose)



Guglielmo Marconi  
(1874~1937)



Marconi's 1901 receiving station was on the grounds of this castle-like structure on what's now called Signal Hill, overlooking St John's Harbor in Newfoundland. Signal Hill is home to VO1AA. [Rick Lindquist, N1RL, Photo]

**“The frequency spectrum is technology, industry, money, culture, and power.”**

“주파수는 기술임과 동시에 산업이며, 돈이며, 문화이며, 그리고 권력이다.”

- J.D. Bedin in Struzak 2000

공공성  
ITU  
국가  
방통위  
통신사업자

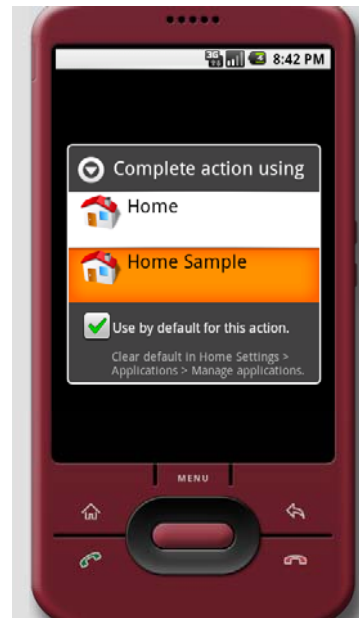
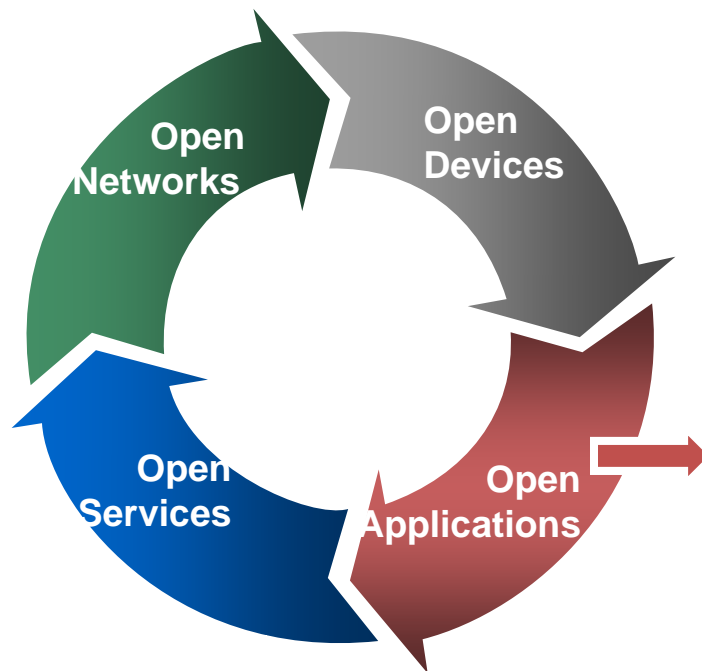


## The “**FOUR OPENS**” of Successful Open Access

- The letter of GOOGLE to FCC, July 18, 2007

“소비자는 지금의 무선 환경보다 더 많은 경쟁과 혁신을 경험할 가치가 있다. 누가 궁극적으로 이 경매에서 이기든 상관없이, 진정한 승리자는 인터넷을 접속하는 방법에서 이전보다 더 나은 선택을 경험할 미국의 소비자다.”

- 2007. 11. 30. 구글 CEO, 에릭 슈미츠



## 안드로이드 App. 개발 준비 : 이클립스, 안드로이드 SDK, ADT 등



The screenshot displays the Eclipse IDE interface for an Android project named 'HelloAndroid'. The left sidebar shows the project hierarchy, including the 'src' directory with 'HelloAndroid.java' and 'R.java', and the 'res' directory with 'drawable', 'layout', and 'values' subdirectories. The main editor window shows the 'HelloAndroid.java' file, which contains the following code:

```
package org.kandroid.helloandroid;

import android.app.Activity;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Below the code editor, the 'Console' window shows the following log output:

```
[2009-02-28 08:46:39 - HelloAndroid] Success!
[2009-02-28 08:46:40 - HelloAndroid] Starting
[2009-02-28 08:46:42 - HelloAndroid] ActivityM
```

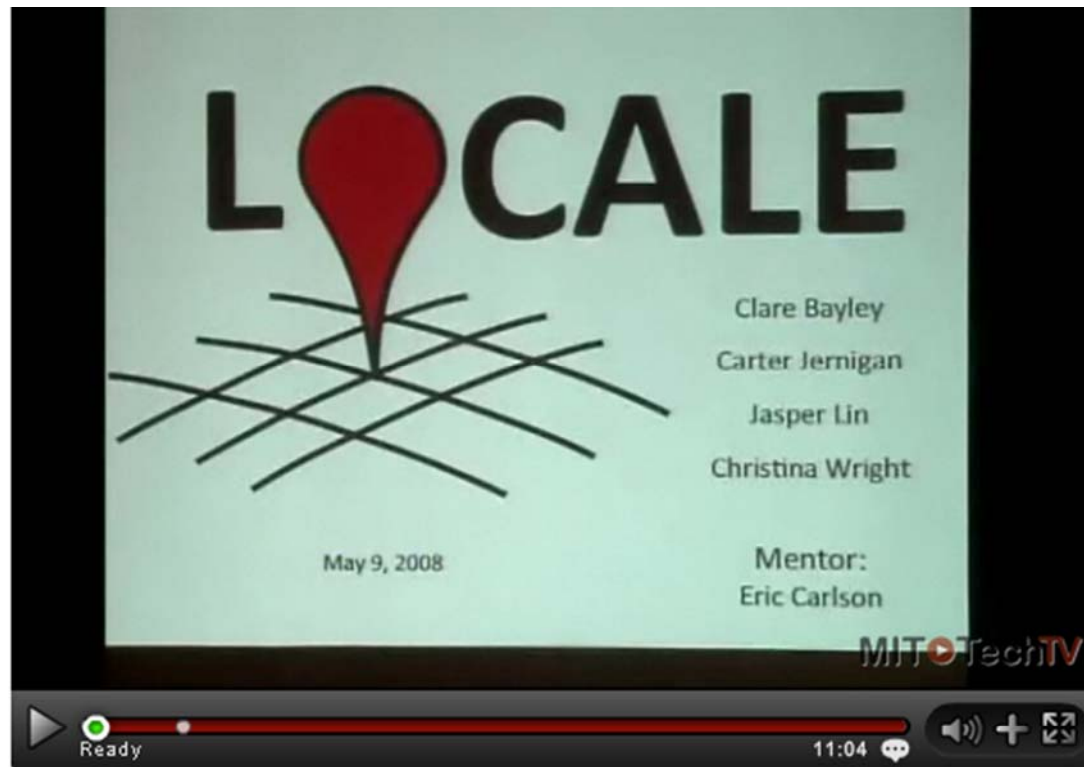
To the right of the IDE, a physical Android phone is shown displaying the home screen with a clock and app icons. Below the phone, an Android emulator (5554) is running, showing the same home screen and a virtual keyboard.



### A first hand look at building an Android application



출처 : <http://www.youtube.com/watch?v=l6ObTqliYfE>

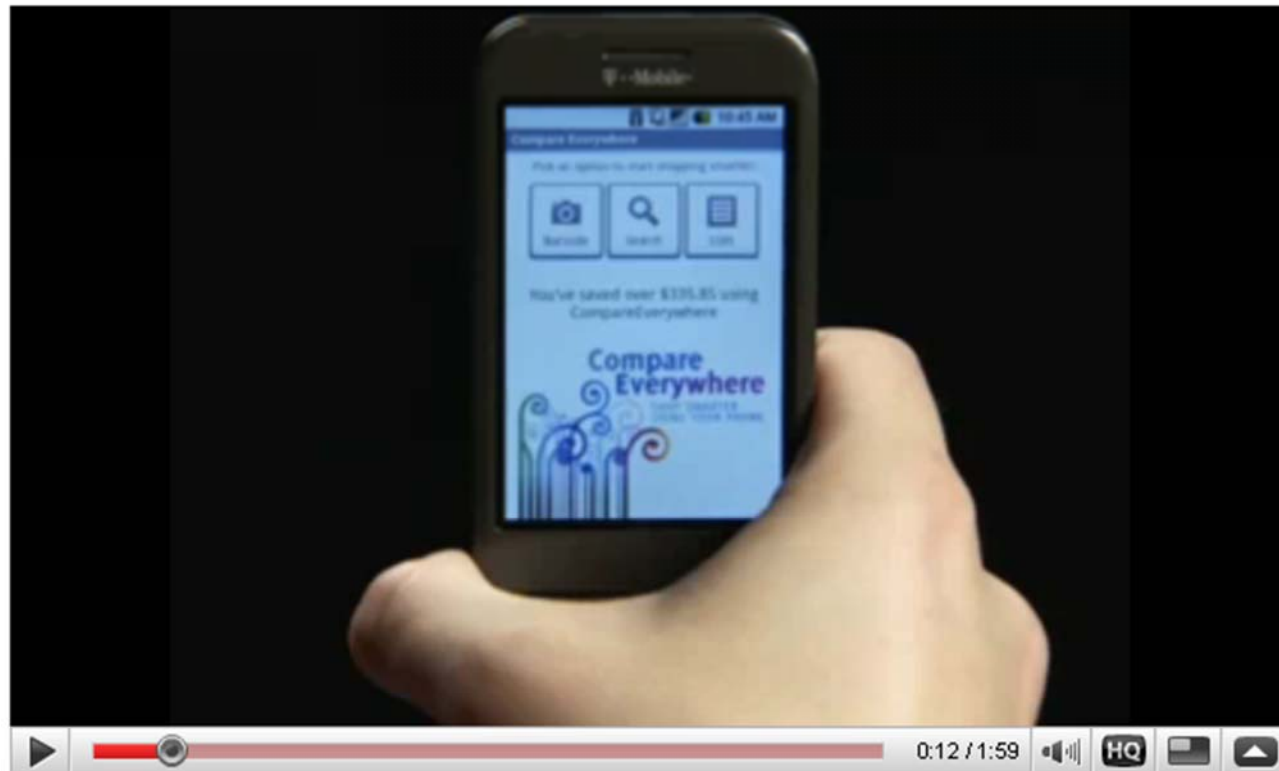


출처 : <http://techtv.mit.edu/videos/270-60876081-students-create-award-winning-mobile-platform-software>





### Android Developers - CompareEverywhere



출처 : <http://www.youtube.com/watch?v=V2-jLK0aNil>



<http://java.sun.com/docs/books/tutorial/java/TOC.html>

- What Is an Object?
- What Is a Class?
- What Is Inheritance?
- What Is an Interface?
- What Is a Package?



### 1. 과제의 내용(제한 없음) 및 형식(개인 또는 팀)

- 안드로이드 관련 문서 정리 (용어, 역사, 기술개념 정리 등)
- 안드로이드 관련 기획 (서비스, 콘텐츠, 애플리케이션 등)
- 안드로이드 애플리케이션 개발
- 안드로이드 Native 애플리케이션 개발
- 안드로이드 디바이스 프로토타이핑

### 2. 과제 제출 시점 (총2회)

- 중간고사 시점
- 기말고사 시점

### 3. 기타

- 모든 평가는 과제 제출로 대신합니다. (시험은 없습니다.)
- 과제 제출 방법은 별도로 공지할 예정입니다.





1. <http://www.kandroid.org/> 접속
2. 회원가입
3. 탭 메뉴에서 **북스토어** 클릭
4. 왼쪽 메뉴에서 **대학생을 위한 강의노트** 클릭
5. 게시글 중, **준비물 : 안드로이드 개발 환경 설정** 클릭 후 읽고 따라서 해 보세요.
6. 그런 다음, p20의 **Dan Morrill의 애플리케이션 개발 설명**을 보고 따라서 해 보세요.  
**(단, 실행해 보기 전에 AVD 생성하여야 함. p26에 생성방법 설명되어 있음)**



이것을 통해 이해해야 하는 것은,  
다음 페이지(p27, p28)에 나오는 **안드로이드 빌드 프로세스**(1,2) 입니다.



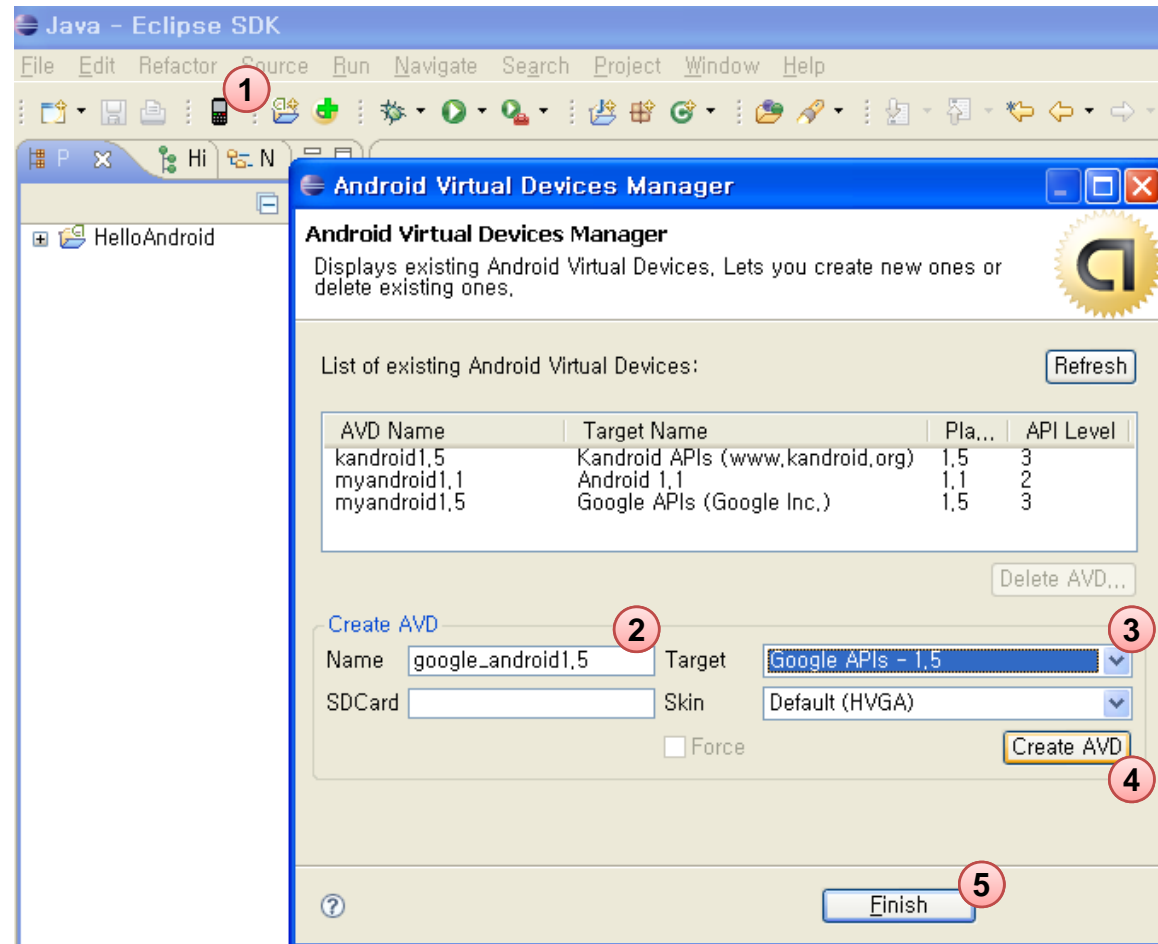
1. 클릭 후,

2. google\_android1.5 입력  
3. Google APIs – 1.5 선택  
4. Create AVD 클릭

2. my\_android1.1 입력  
3. Android 1.1 – 1.1 선택  
4. Create AVD 클릭

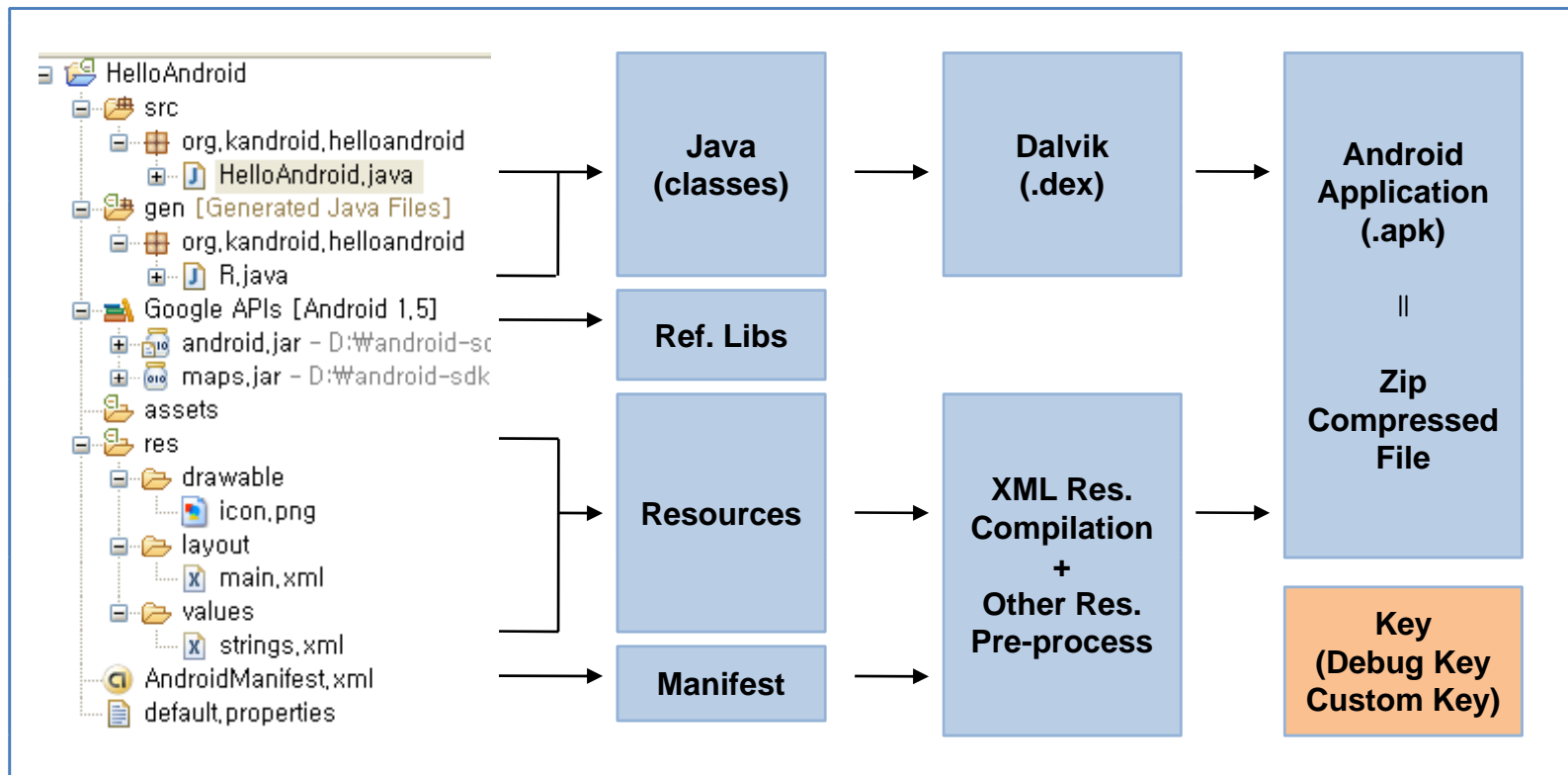
2. my\_android1.5 입력  
3. Android 1.5 – 1.5 선택  
4. Create AVD 클릭

5. Finish 클릭



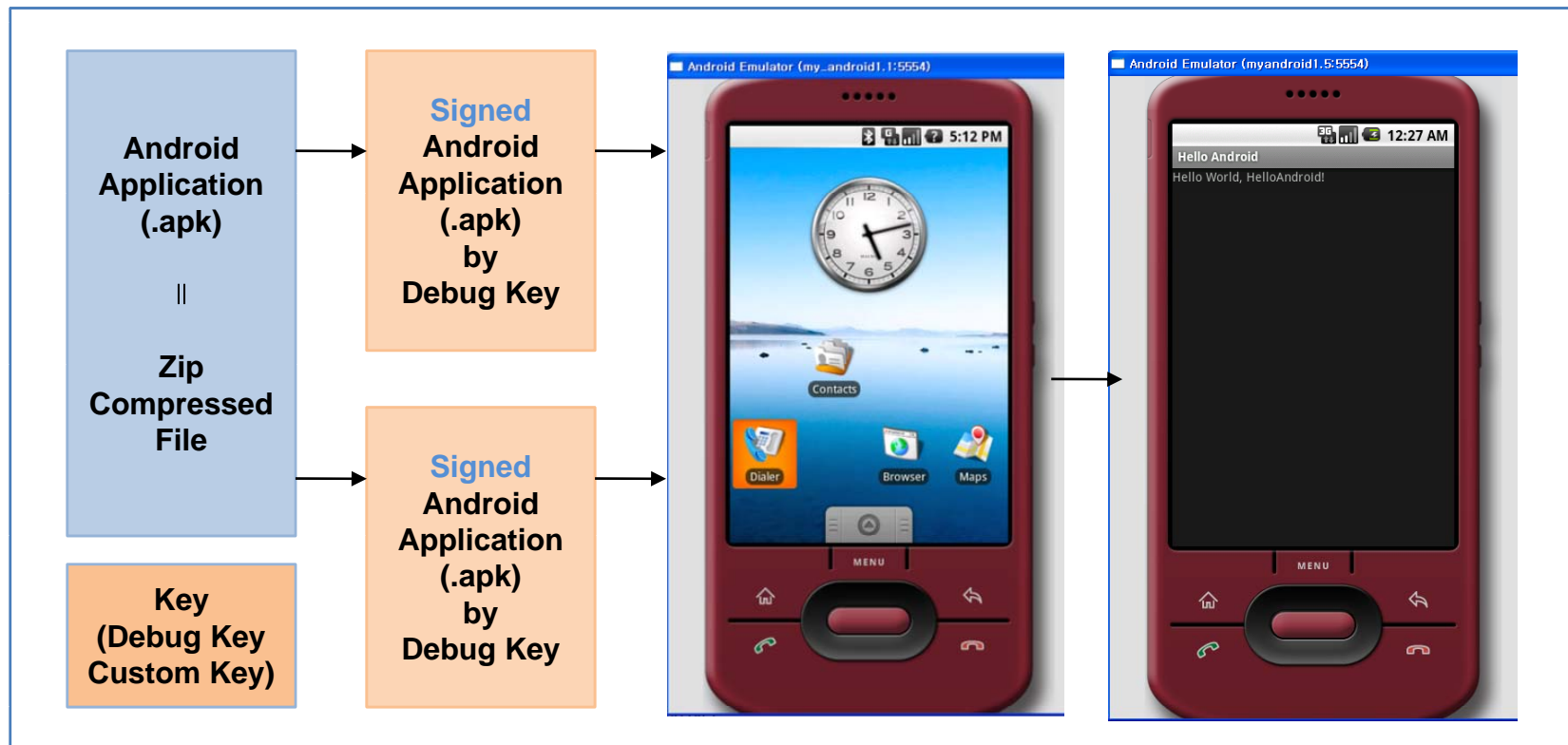


## 안드로이드 애플리케이션 빌드 프로세스(1)

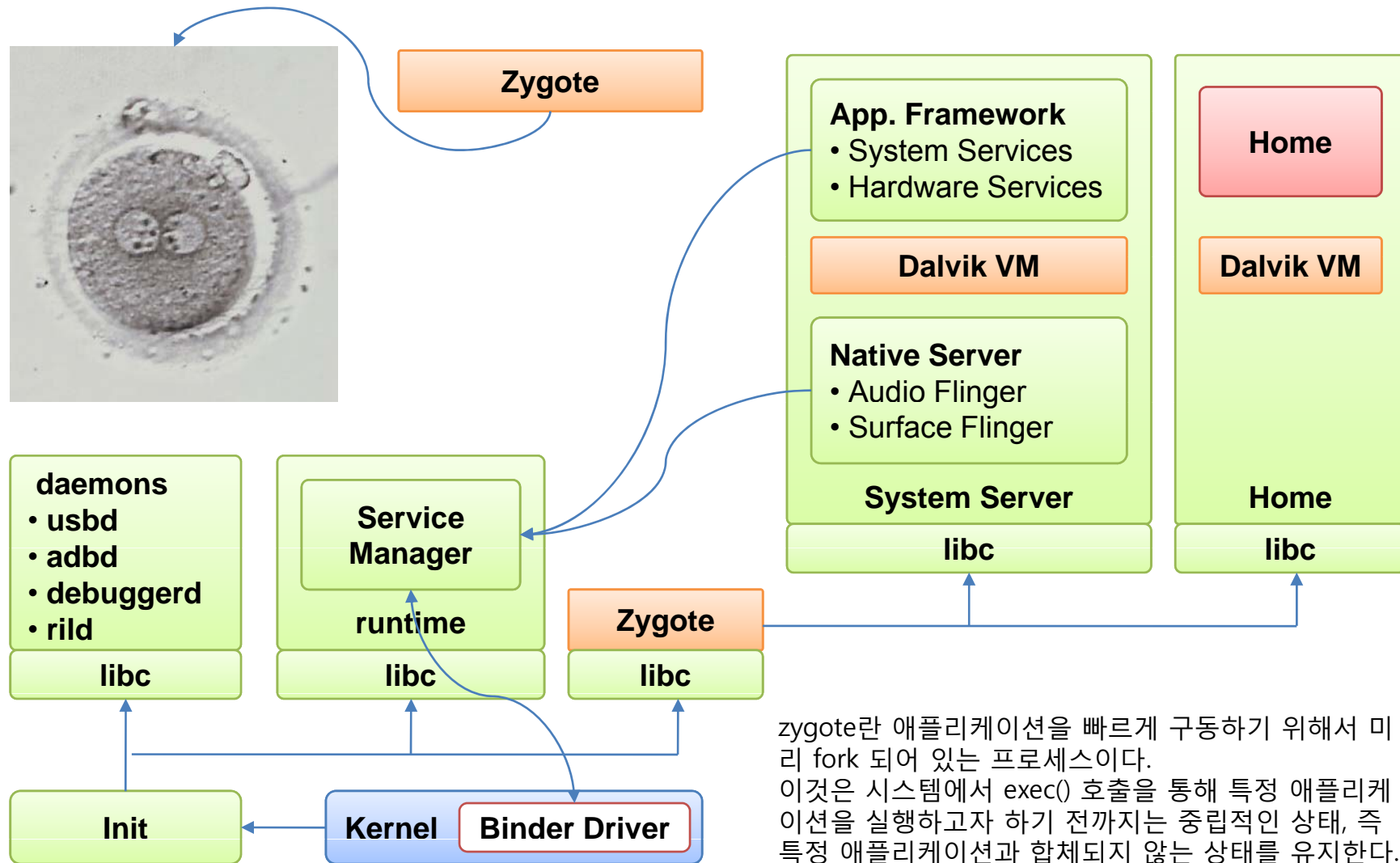




## 안드로이드 애플리케이션 빌드 프로세스(2)







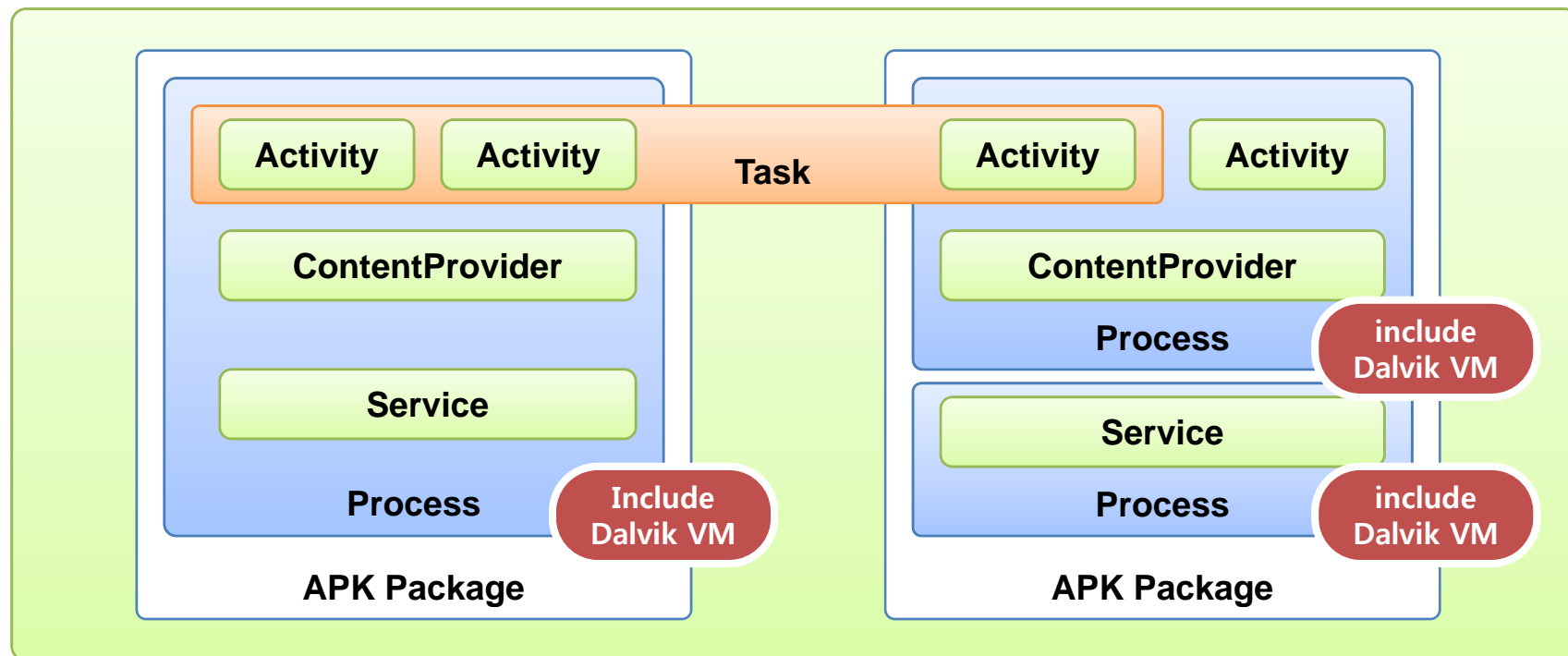


## Application Building Block

- AndroidManifest.xml
- Activity
- ContentProvider
- Service
- BroadcastReceiver

## Intent : Bundle of Information

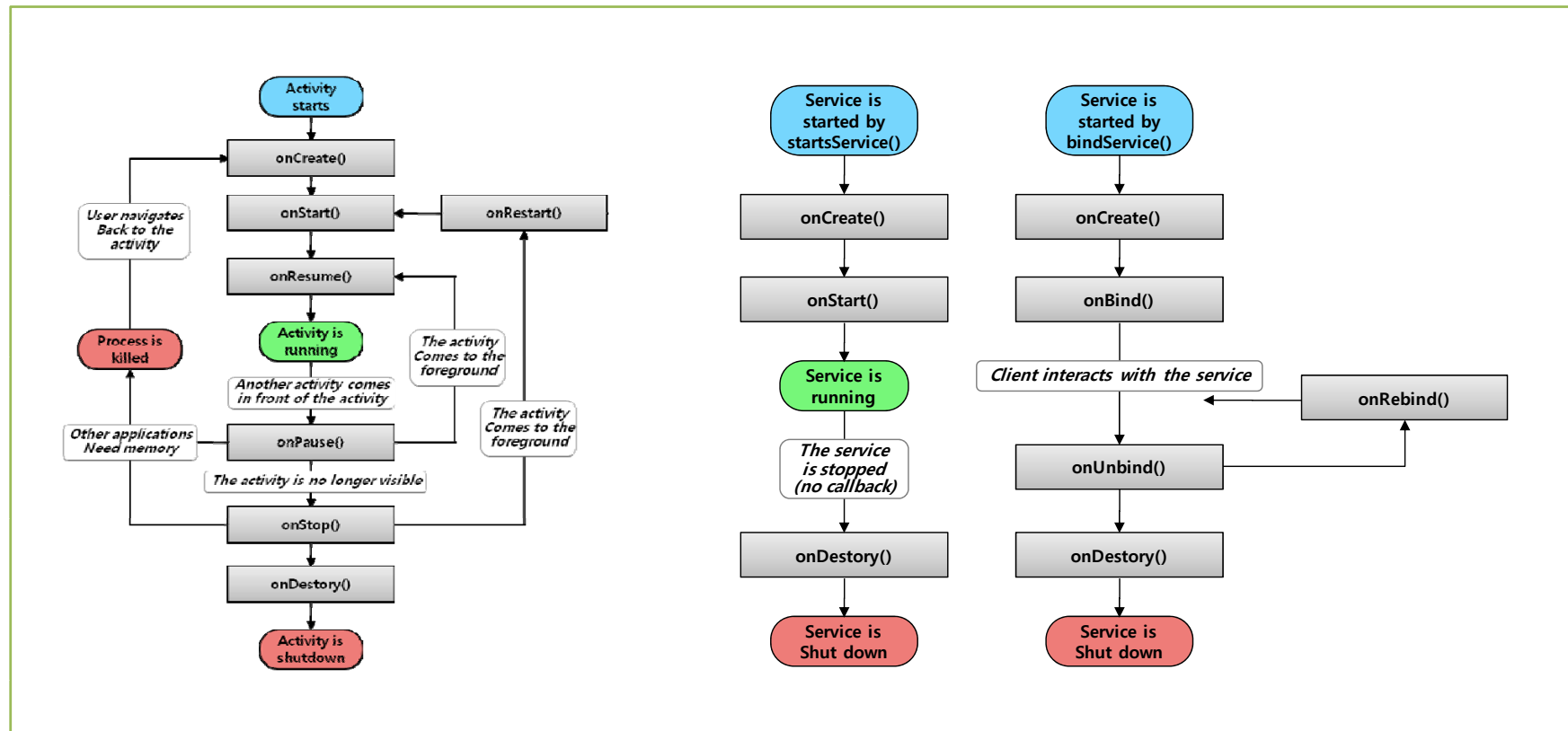
- Explicit Method : Call Class
- Implicit Method : IntentFilter
  - Action, Data, Category
  - Declared at AndroidManifest.xml



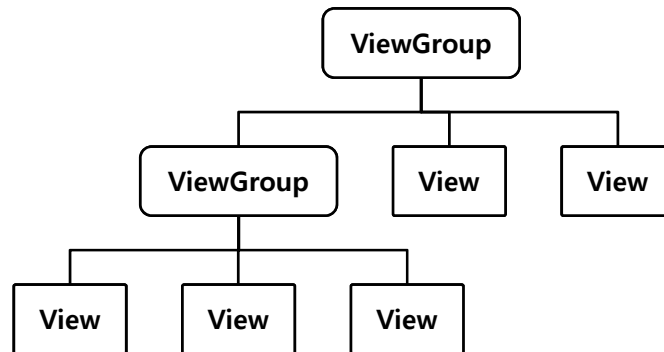


## Application Lifecycle

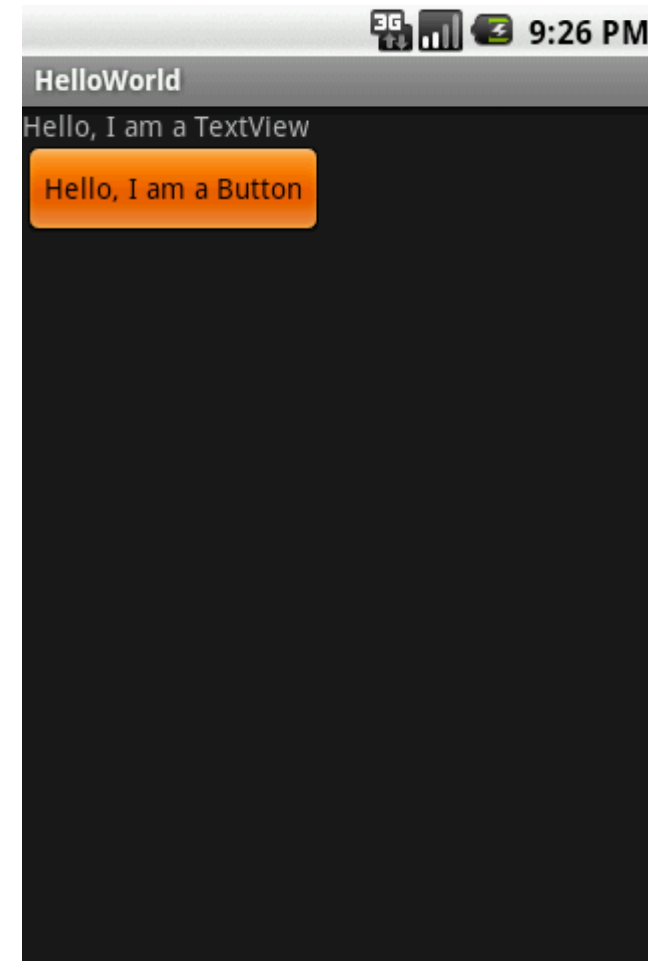
- Activity Lifecycle
- Service Lifecycle







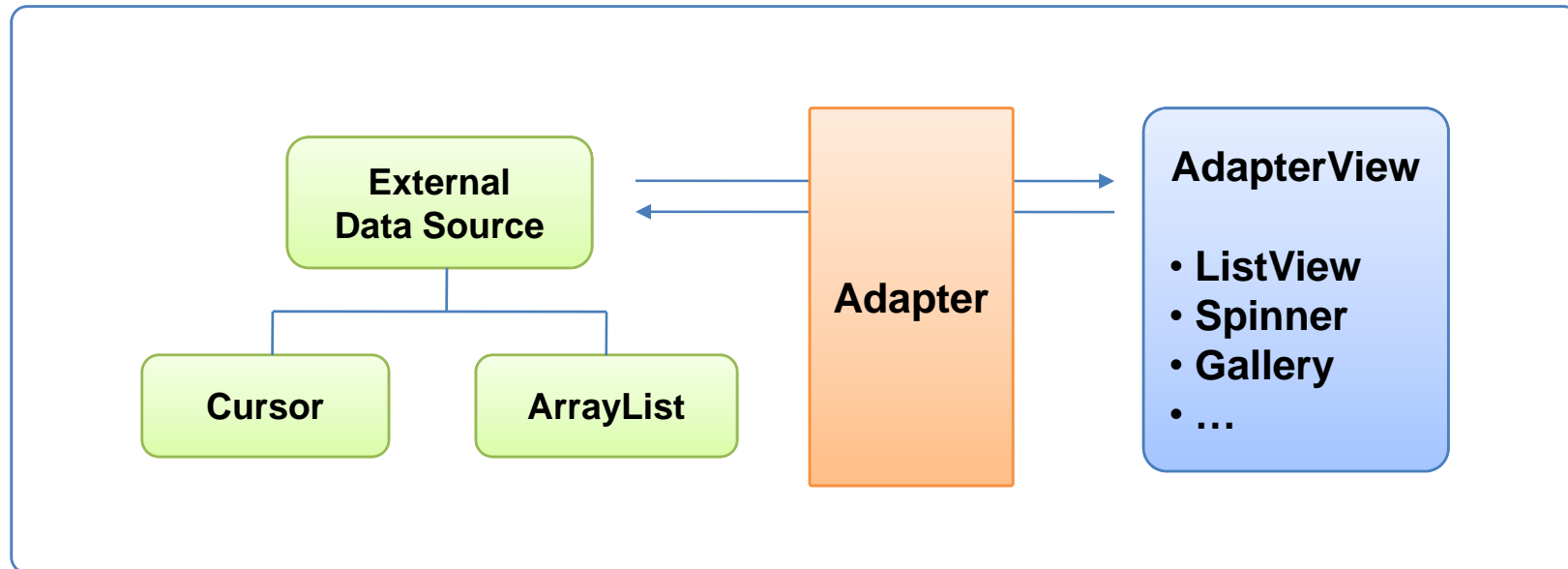
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/..."
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

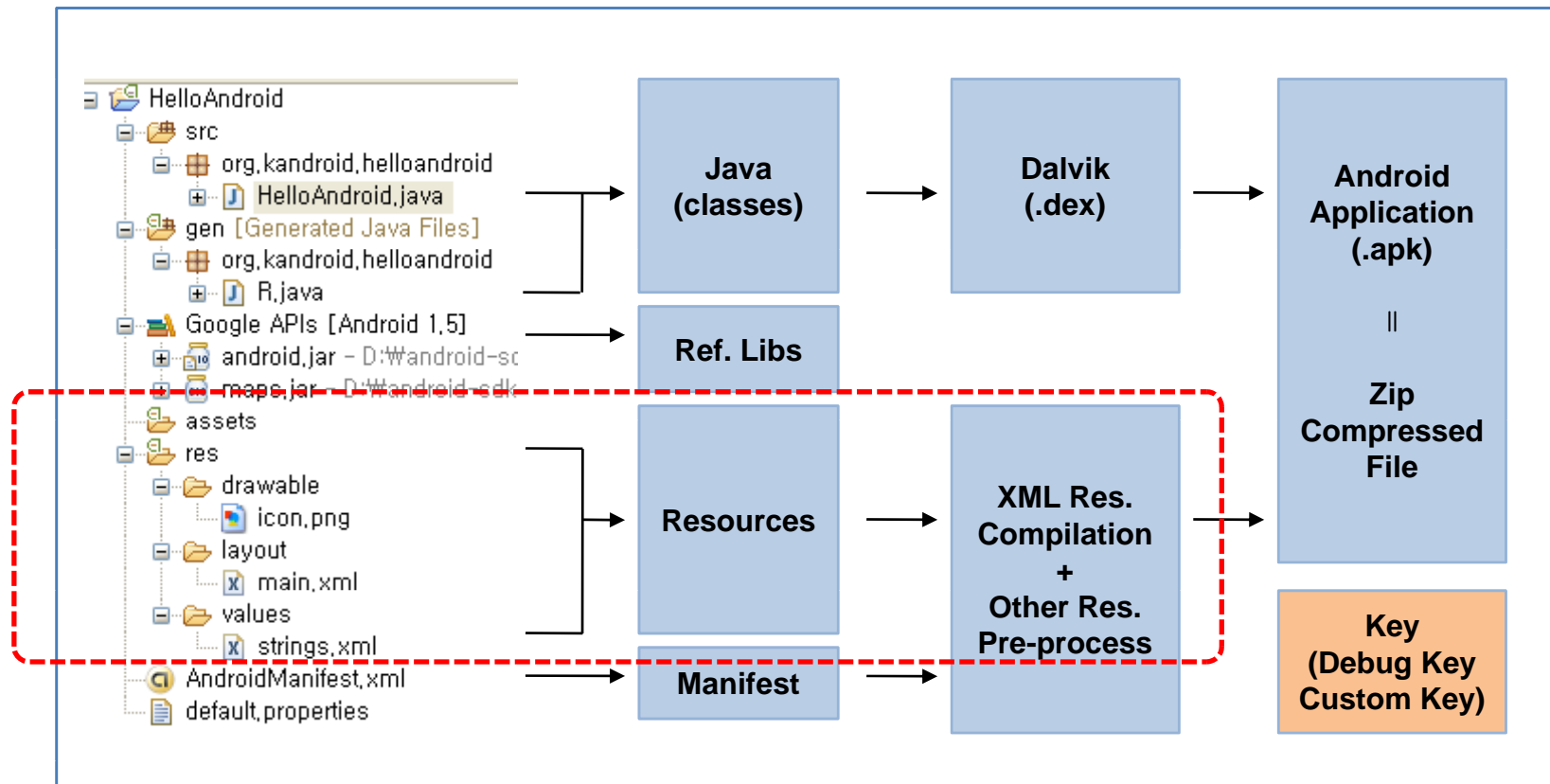




### Dynamic Data Binding Mechanism

- AdapterView
- Adapter
- Data Source
- Handling User Selection Event

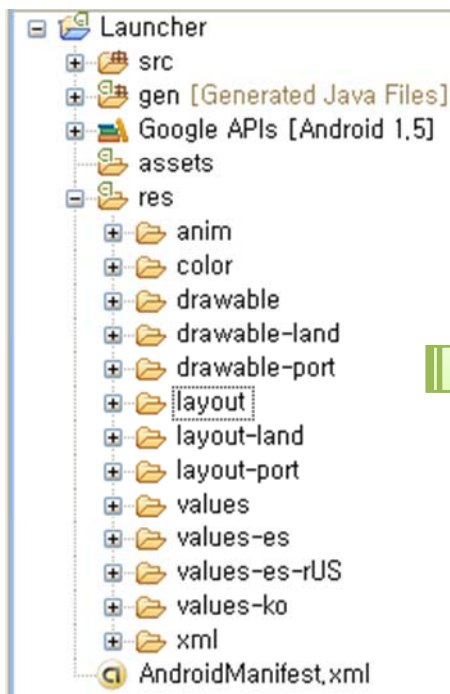




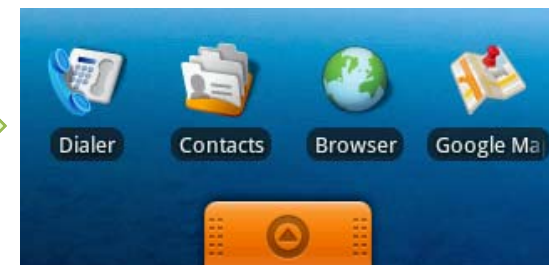


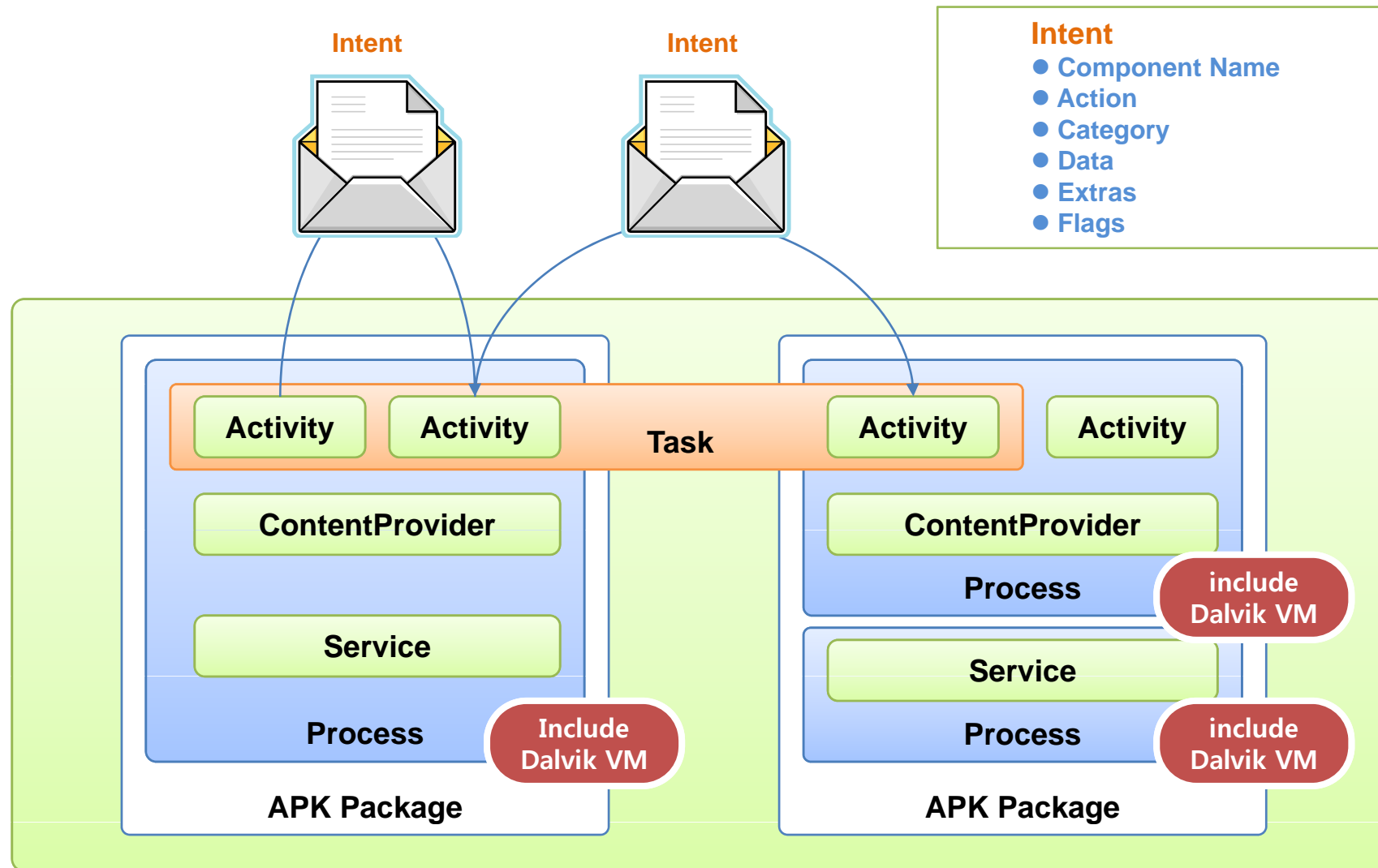
## Alternate Resources

- Basic Resource Folder
- Add Qualifier



Qualifier
MCC and MNC
Language and region
Screen orientation
Screen pixel density
Touchscreen type
Whether the keyboard is available to the user
Primary text input method
Primary non-touchscreen navigation method
Screen dimensions
SDK version
(Minor version)







## Implicit Intent

- Component Name
- Action
- Category
- Data
- Extras
- Flags



## AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.kandroid.helloandroid" android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloAndroid" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

## Intent Resolution

- Component Name
- Action Test
- Category Test
- Data Test
- Extras
- Flags

Activities

Services

Broadcast Receivers

Content Providers

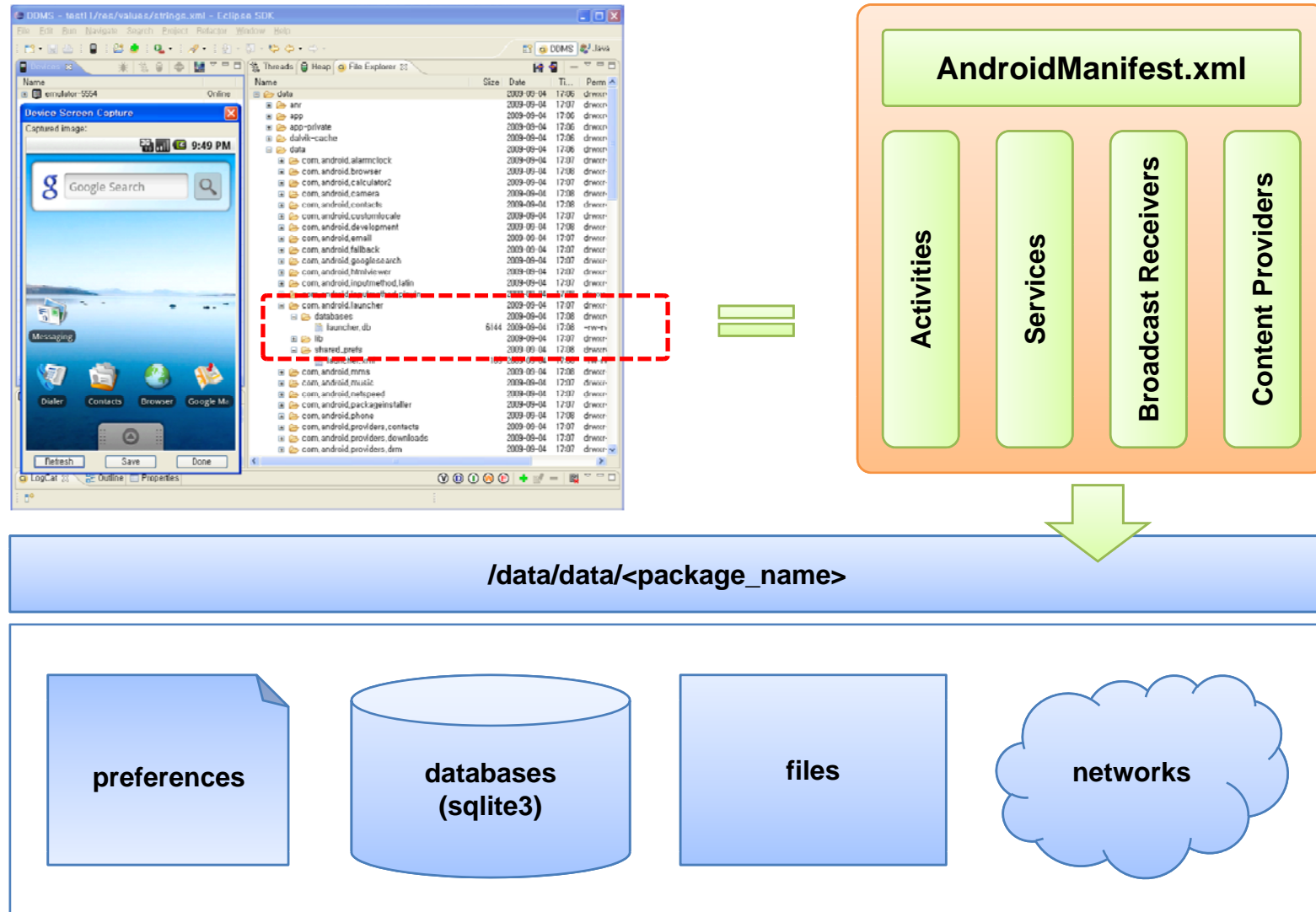


## All Application

- Has their own directory
- /data/data/<package\_name>

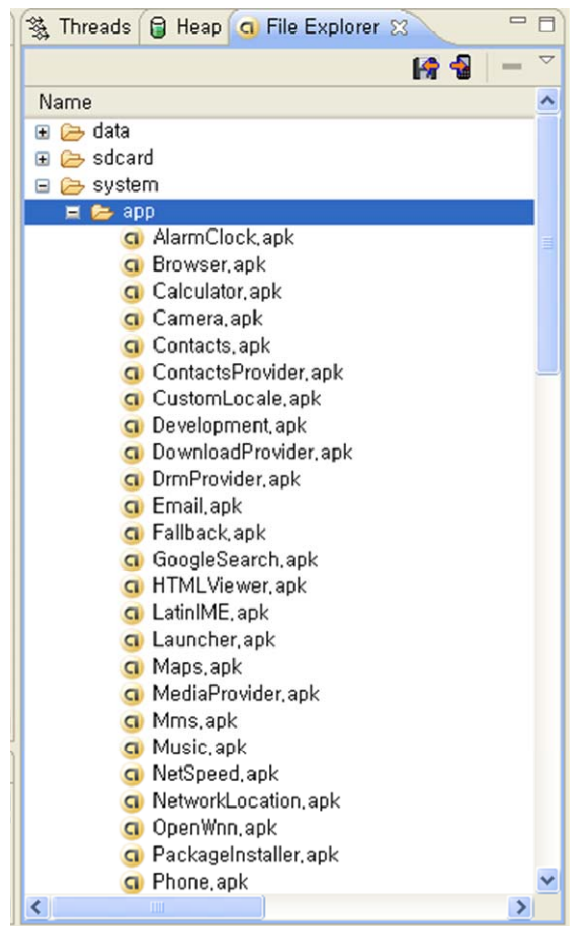
The screenshot displays the Eclipse IDE interface. The 'File Explorer' view on the right shows the file system of an Android device. The 'data' directory is expanded, revealing a list of application-specific directories. Each entry includes its name, size, date, time, and permissions. The 'data' directory itself has permissions 'drwxr-xr-x'. Sub-directories include 'anr', 'app', 'app-private', 'dalvik-cache', and a list of application-specific directories such as 'com.android.alarmclock', 'com.android.browser', 'com.android.calculator2', 'com.android.camera', 'com.android.contacts', 'com.android.customlocale', 'com.android.development', 'com.android.email', 'com.android.fallback', 'com.android.googlesearch', 'com.android.htmlviewer', 'com.android.inputmethod.latin', 'com.android.inputmethod.pinyin', 'com.android.launcher', 'databases', 'launcher.db', 'lib', 'shared\_prefs', 'launcher.xml', 'com.android.mms', 'com.android.music', 'com.android.netspeed', 'com.android.packageinstaller', 'com.android.phone', 'com.android.providers.contacts', 'com.android.providers.downloads', and 'com.android.providers.drm'. The 'Device Screen Capture' window on the left shows a simulated Android phone screen with a Google search bar, a 'Messaging' app icon, and a dock with 'Dialer', 'Contacts', 'Browser', and 'Google Mail' icons. The status bar at the top of the device screen shows '3G', signal strength, and the time '9:49 PM'.

# 안드로이드 App. 교육 : 강의 개요 (6강. 데이터 저장공간)

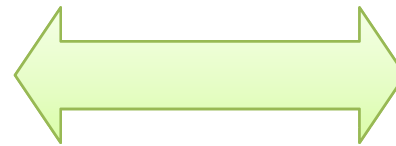




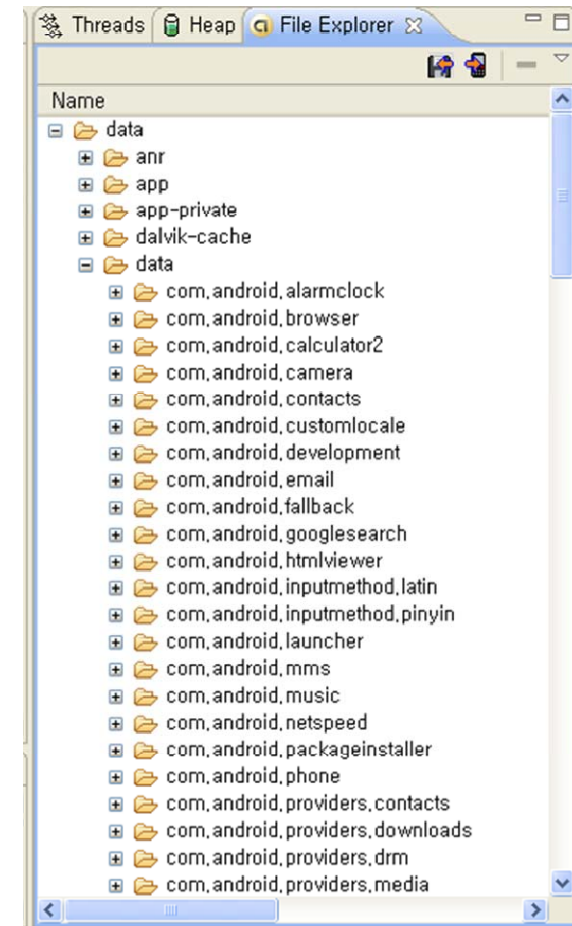
## 안드로이드 App. 교육 : 강의 개요 (7강. 콘텐츠 프로바이더)

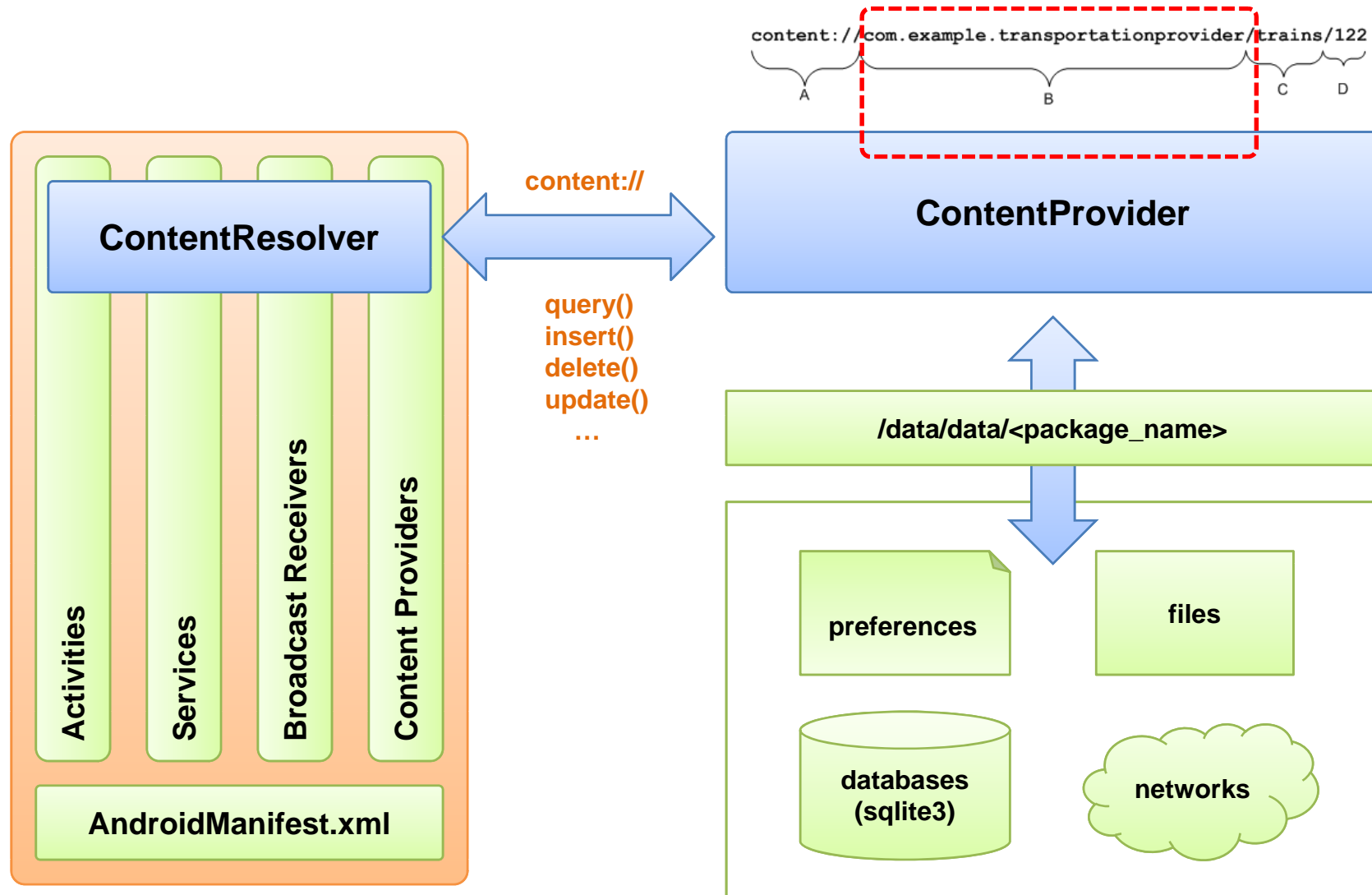


All application data are private to that application.



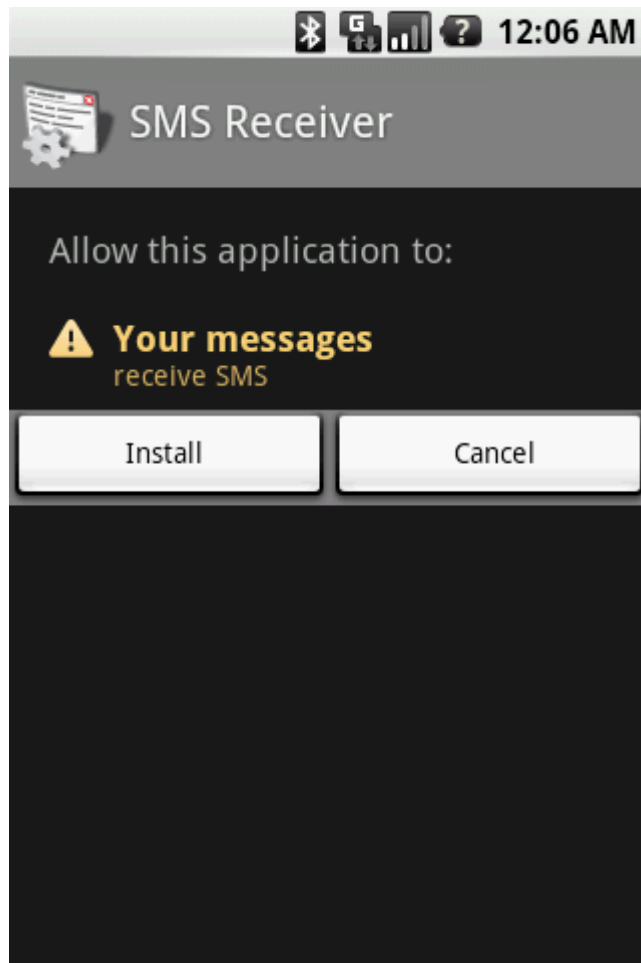
Data sharing mechanism between apps.







USER	PID	PPID	VSIZE	RSS	PC	NAME
-----						
root	1	0	264	176	0000c36c	S /init
root	17	1	724	300	afe0b50c	S /system/bin/sh
system	18	1	796	252	afe0b74c	S /system/bin/servicemanager
root	19	1	1824	324	afe0b50c	S /system/bin/mountd
root	20	1	652	252	afe0c0bc	S /system/bin/debuggerd
radio	22	1	5308	636	afe0bdbc	S /system/bin/rild
root	23	1	69732	18780	afe0b874	S zygote
media	24	1	16052	3068	afe0b74c	S /system/bin/mediaserver
bluetooth	26	1	1160	560	afe0c69c	S /system/bin/dbus-daemon
root	27	1	788	288	afe0b50c	S /system/bin/installd
root	30	1	820	316	afe0c61c	S /system/bin/qemud
root	33	1	3328	144	0000e794	S /sbin/adbd
system	46	23	196516	23832	afe0b74c	S system_server
radio	84	23	103296	16628	afe0c824	S com.android.phone
app_5	90	23	121588	21136	afe0c824	S android.process.acore
app_11	108	23	92172	11984	afe0c824	S com.android.mms
app_16	121	23	95200	11588	afe0c824	S com.google.process.gapps
app_1	134	23	91844	11188	afe0c824	S com.android.alarmclock
app_0	148	23	94584	13200	afe0c824	S com.google.android.apps.maps
app_4	155	23	91908	12100	afe0c824	S android.process.media
root	174	33	724	320	afe0c4cc	S /system/bin/sh



### Permission enforcement cases

At the time of a call into the system

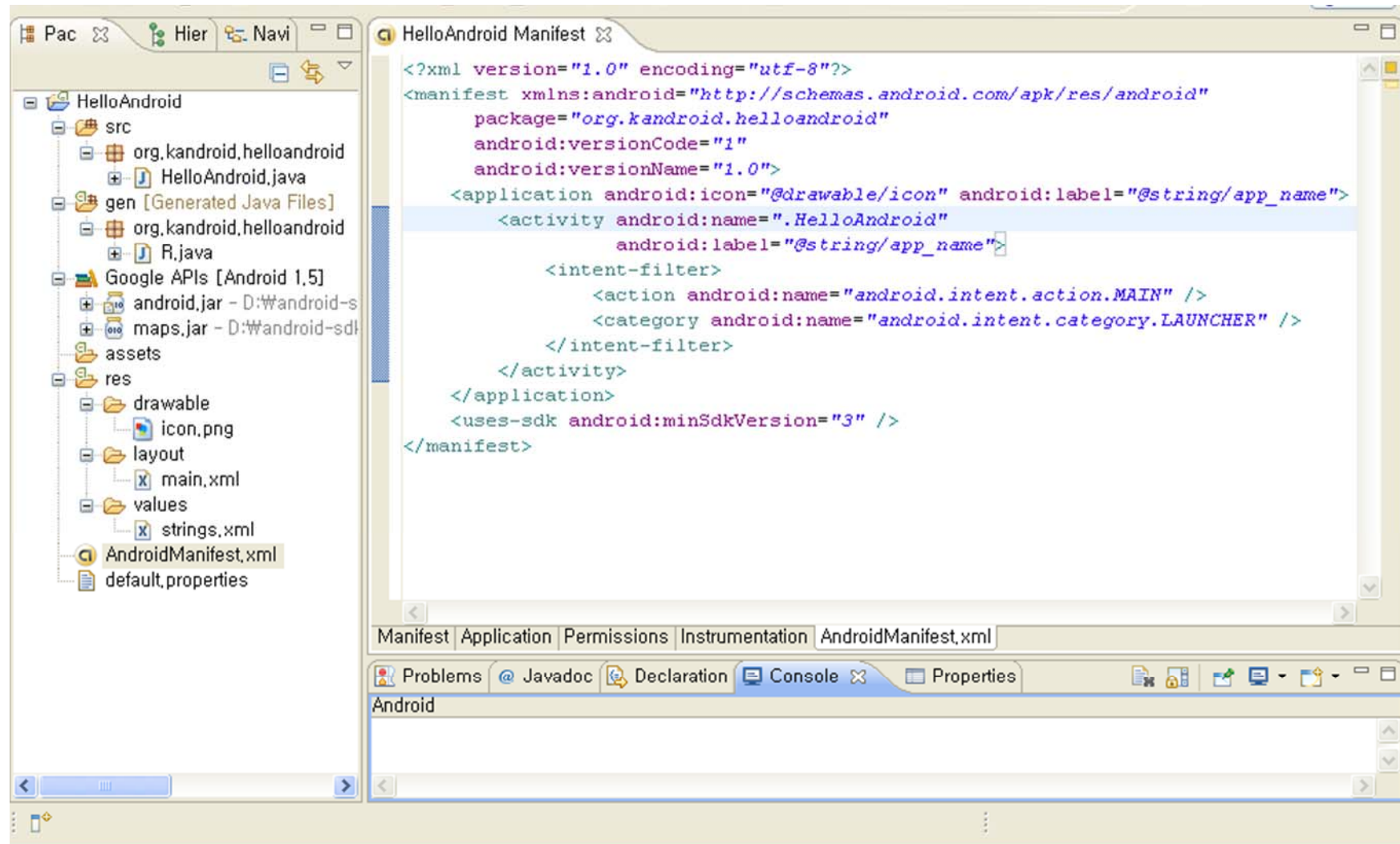
When starting an activity

Both sending and receiving broadcast

When accessing and  
Operating on a content provider

URI Permission

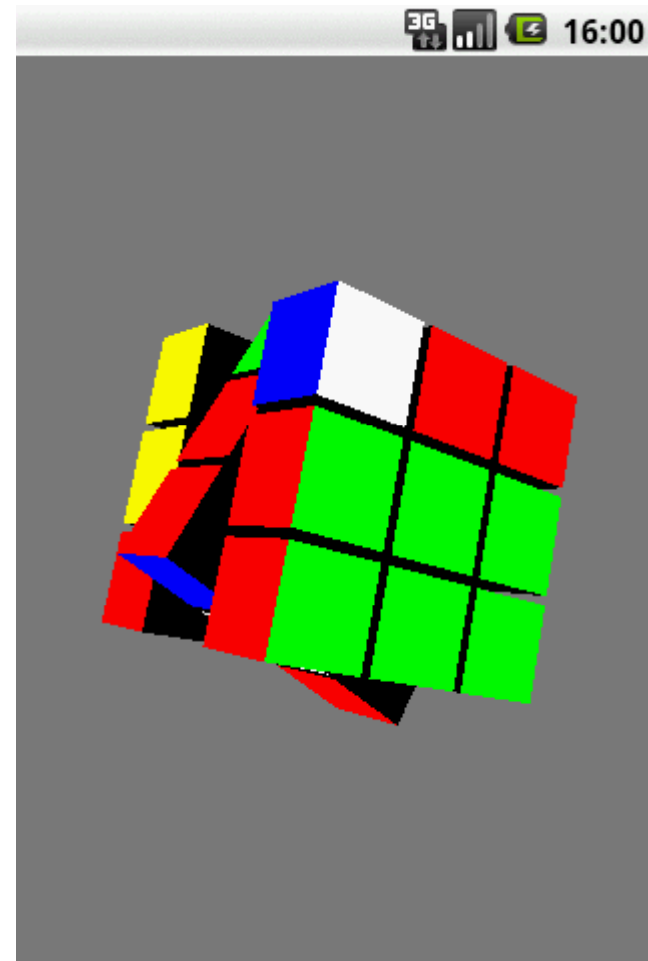
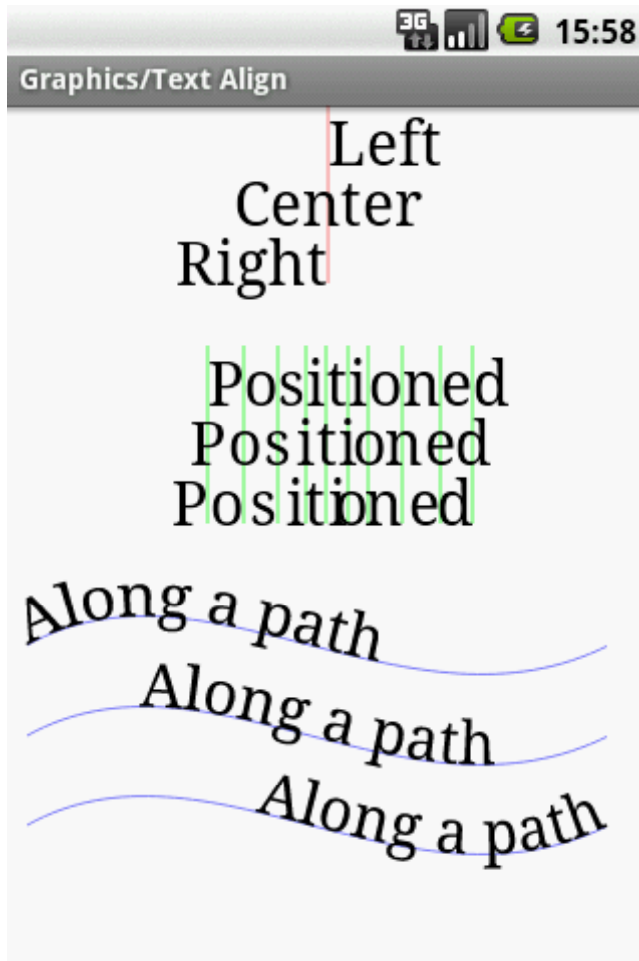
Binding or starting a service



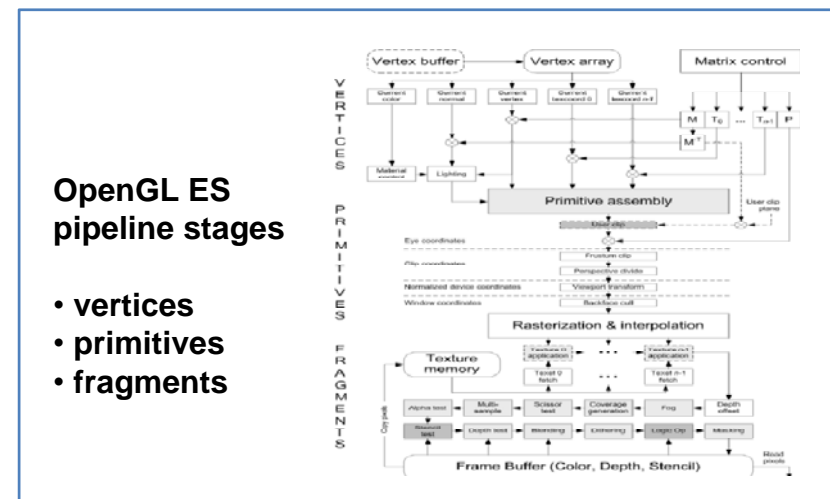
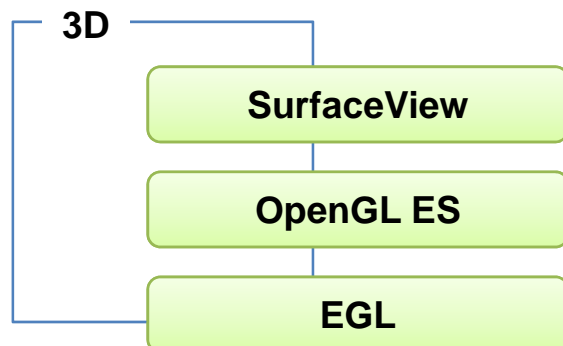
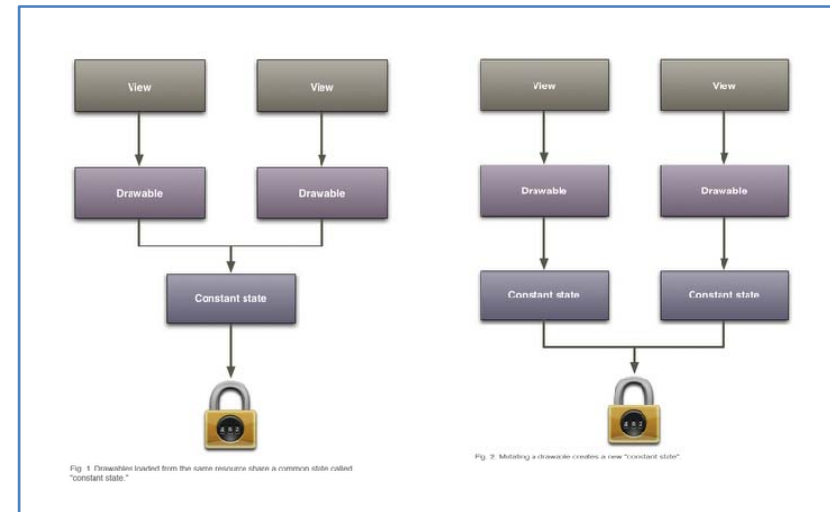
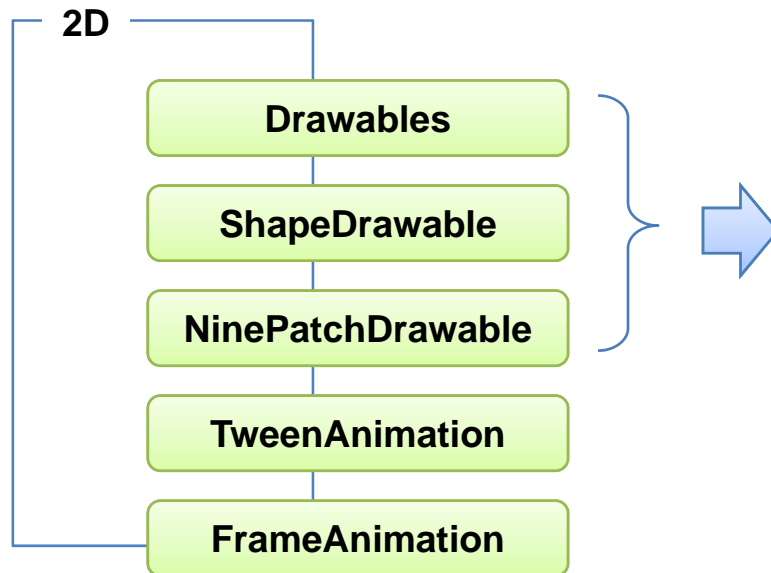


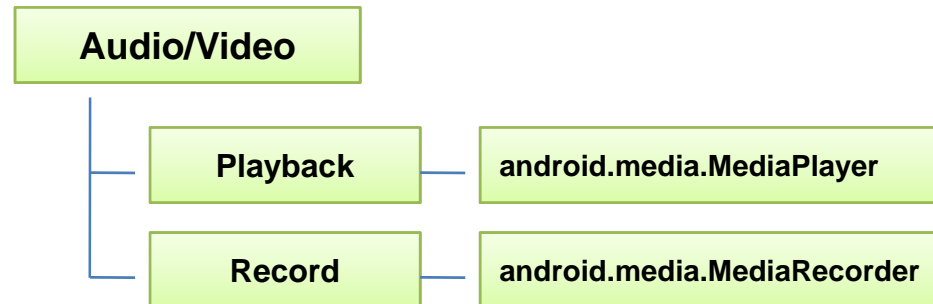
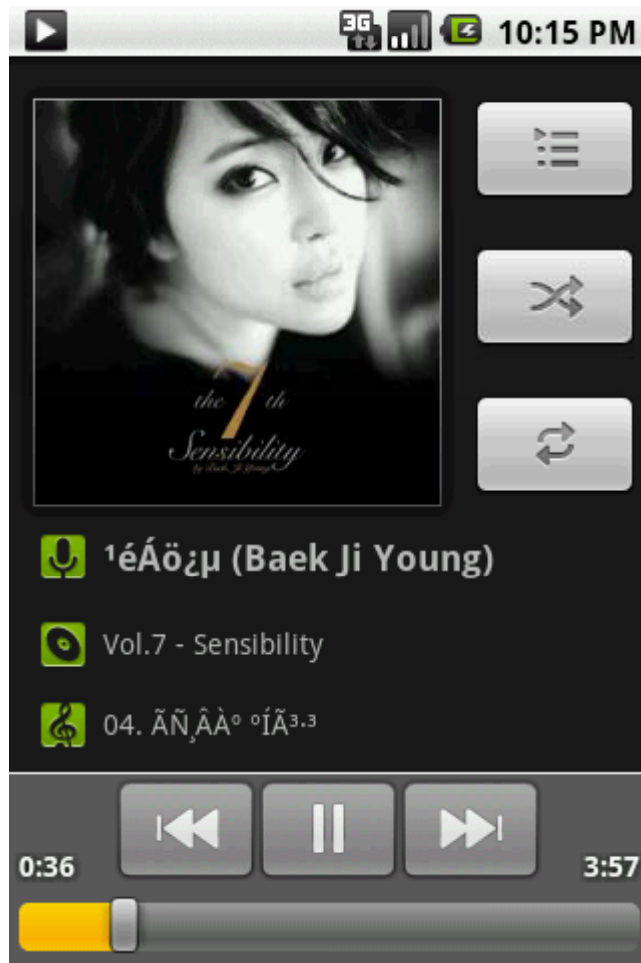
### Structure of the Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <uses-permission />
  <permission /> <permission-group /> <permission-tree />
  <instrumentation /> <uses-sdk />
  <application>
    <activity>
      <intent-filter><action /><category /><data /></intent-filter><meta-data />
    </activity>
    <activity-alias>
      <intent-filter> .. </intent-filter> <meta-data />
    </activity-alias>
    <service>
      <intent-filter> .. </intent-filter> <meta-data />
    </service>
    <receiver>
      <intent-filter> .. </intent-filter> <meta-data />
    </receiver>
    <provider>
      <grant-uri-permission /> <meta-data />
    </provider>
    <uses-library />
  </application>
</manifest>
```





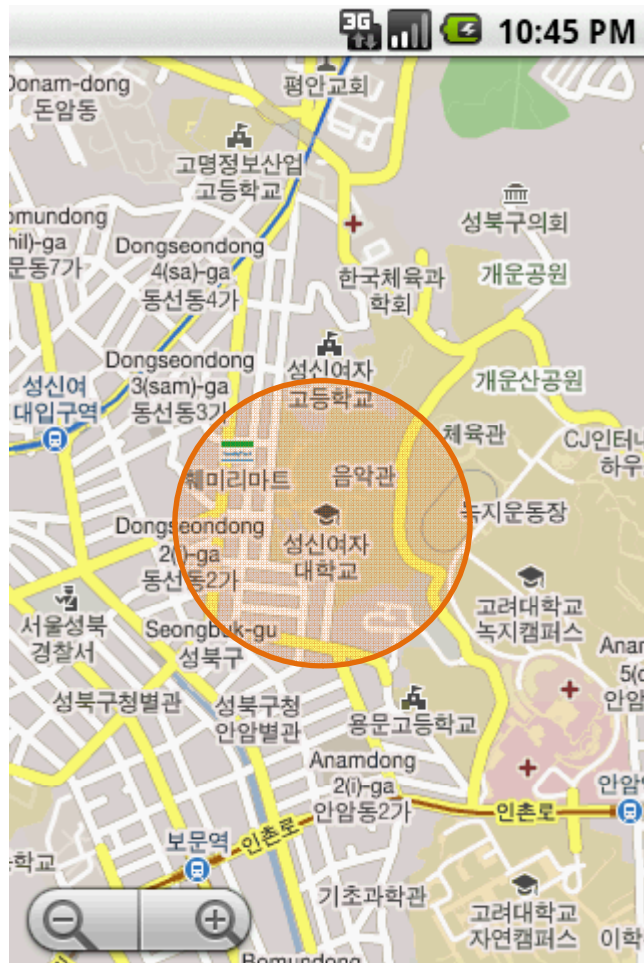


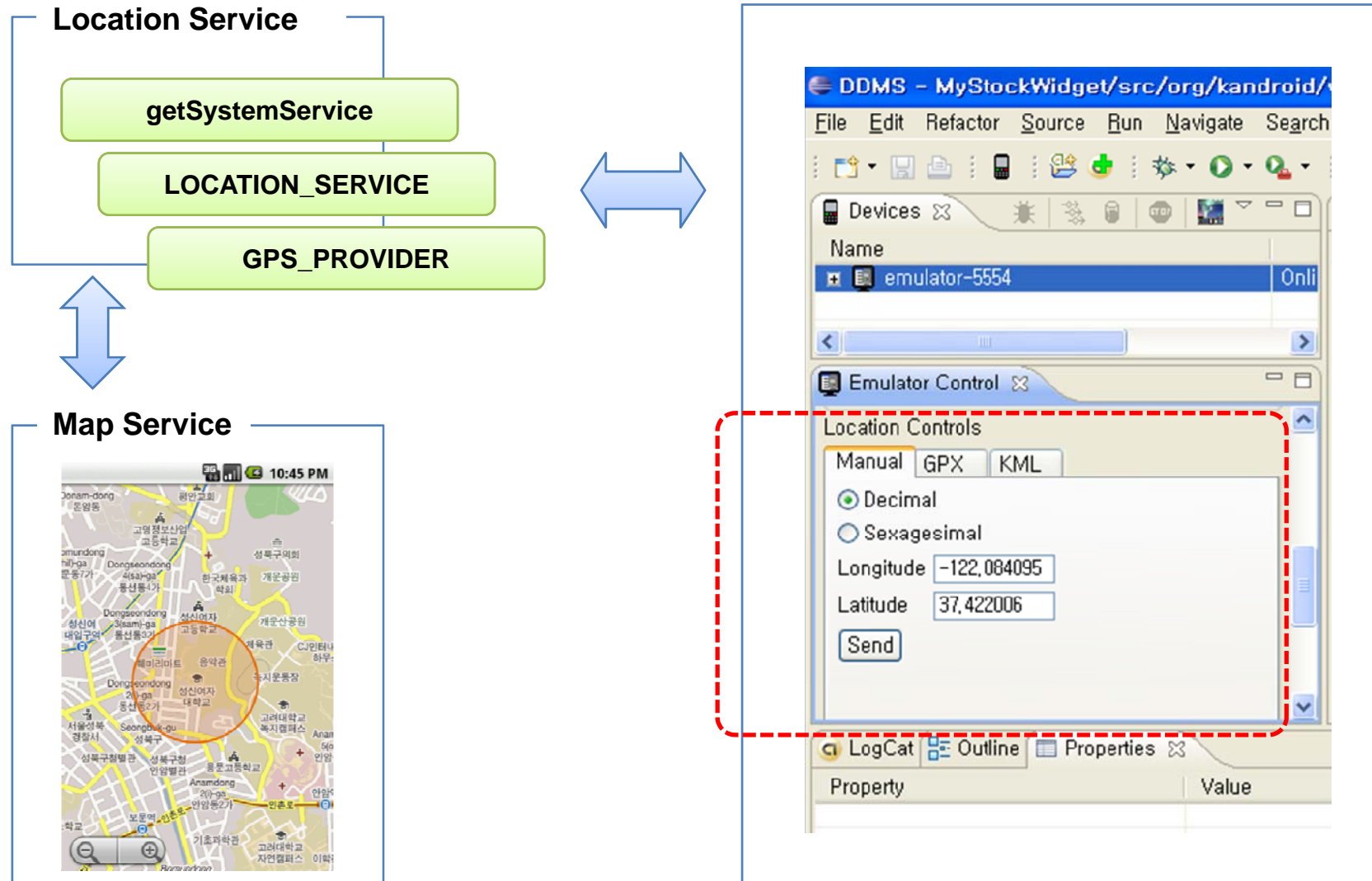




## Android Supported Media Formats (Core Media Formats)

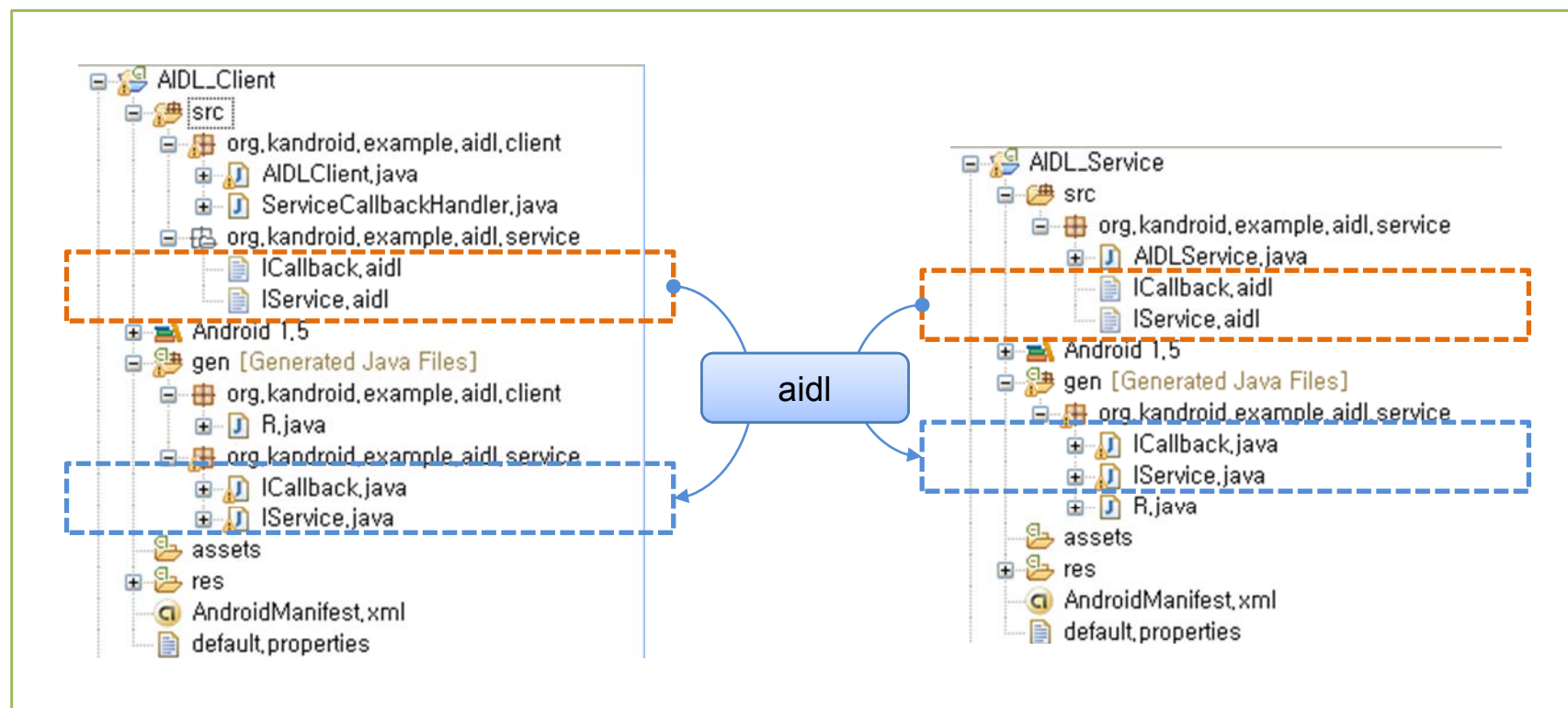
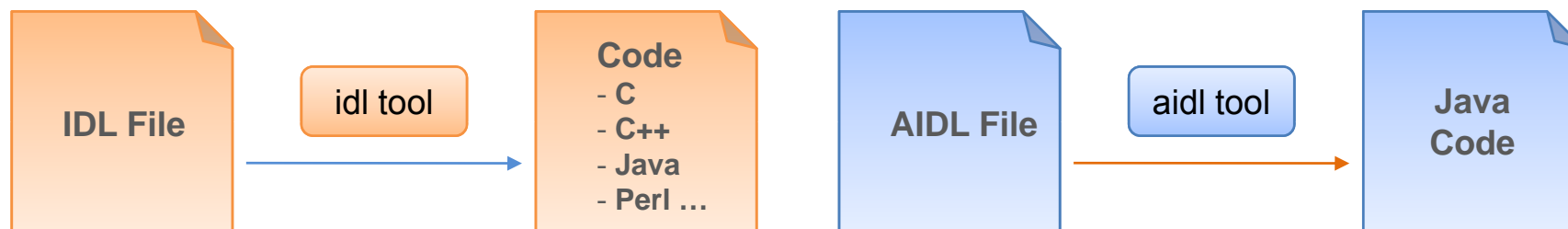
Type	Format	E	D	Details	File Type(s) Supported
Audio	AAC LC/LTP		X	Mono/Stereo content in any combination of standard bit rates up to 160 kbps and sampling rates from 8 to 48kHz	3GPP (.3gp) and MPEG-4 (.mp4, .m4a). No support for raw AAC (.aac)
	HE-AACv1(AAC+)		X		
	HE-AACv2 (enhanced AAC+)		X		
	AMR-NB	X	X	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)
	AMR-WB		X	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)
	MP3		X	Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR)	MP3 (.mp3)
	MIDI		X	MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody	Type 0 and 1 (.mid, .xmf, .mxmf). Also RTTTL/RTX (.rtttl, .rtx), OTA (.ota), and iMelody (.imy)
	Ogg Vorbis		X		Ogg (.ogg)
	PCM/WAVE		X	8- and 16-bit linear PCM (rates up to limit of hardware)	WAVE (.wav)
Image	JPEG	X	X	base+progressive	JPEG (.jpg)
	GIF		X		GIF (.gif)
	PNG		X		PNG (.png)
	BMP		X		BMP (.bmp)
Video	H.263	X	X		3GPP (.3gp)
	H.264	X	X		3GPP (.3gp) and MPEG-4 (.mp4)
	<b>MPEG4 SP</b>		<b>X</b>		<b>3GPP (.3gp)</b>







## 안드로이드 App. 교육 : 강의 개요 (13강. 안드로이드 인터페이스 정의 언어)





## RPC Mechanism

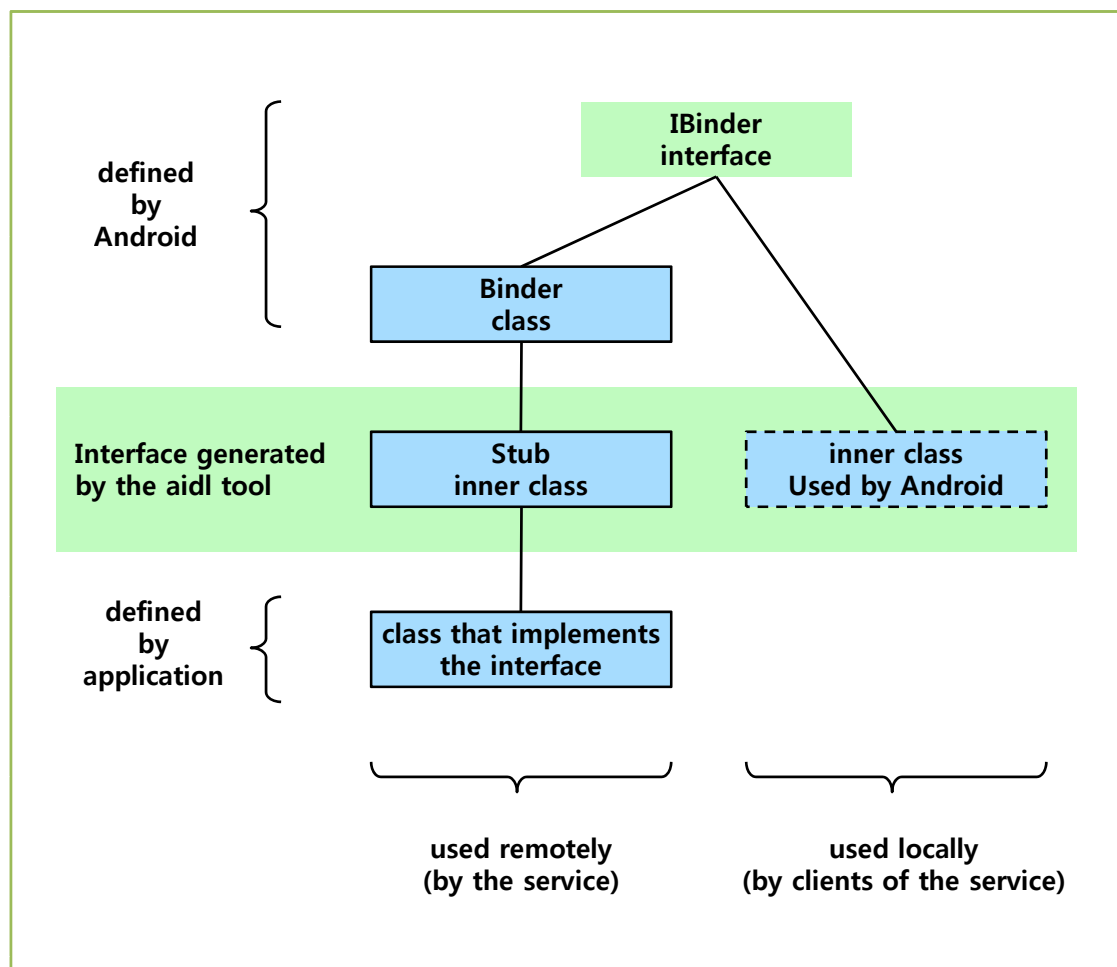
- case : called locally
- case : called remotely

## AIDL Client

- `bindService(intent, ..)`

## AIDL Service

- return `IBinder`

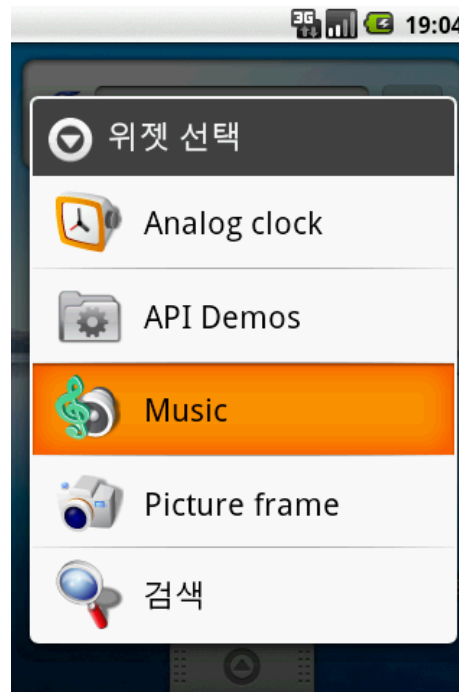
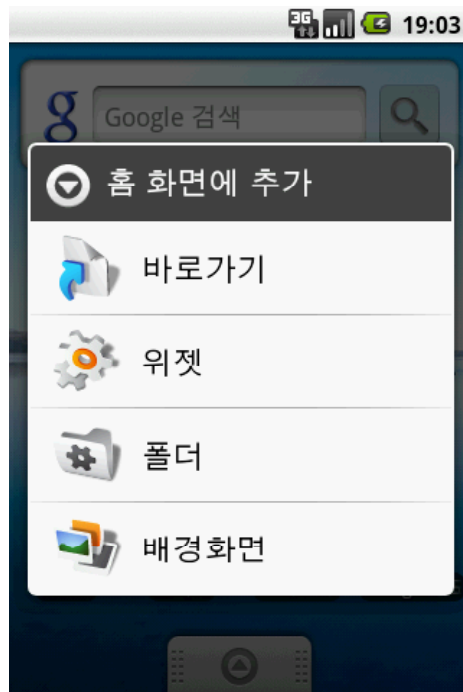






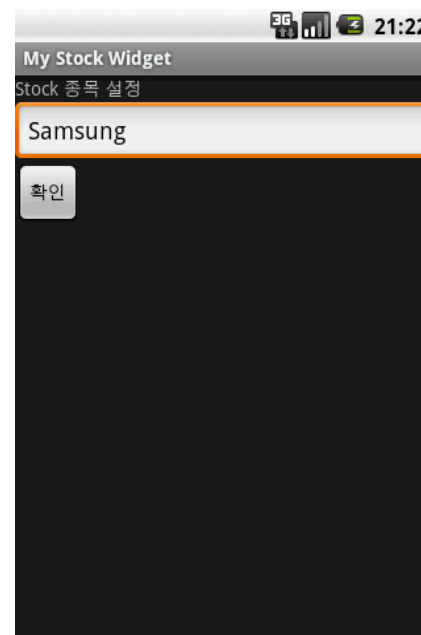
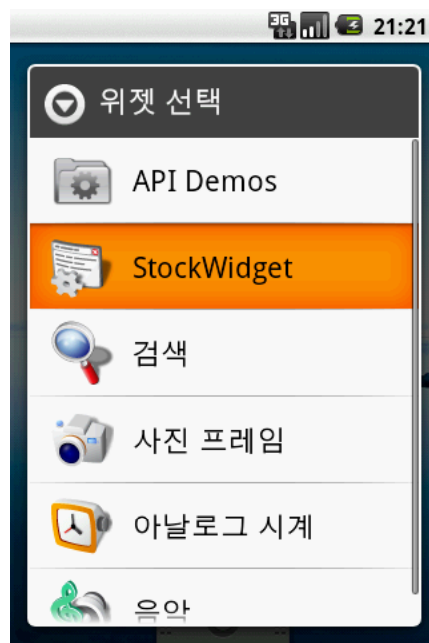
### Adding App Widget

- Long Click at Launcher



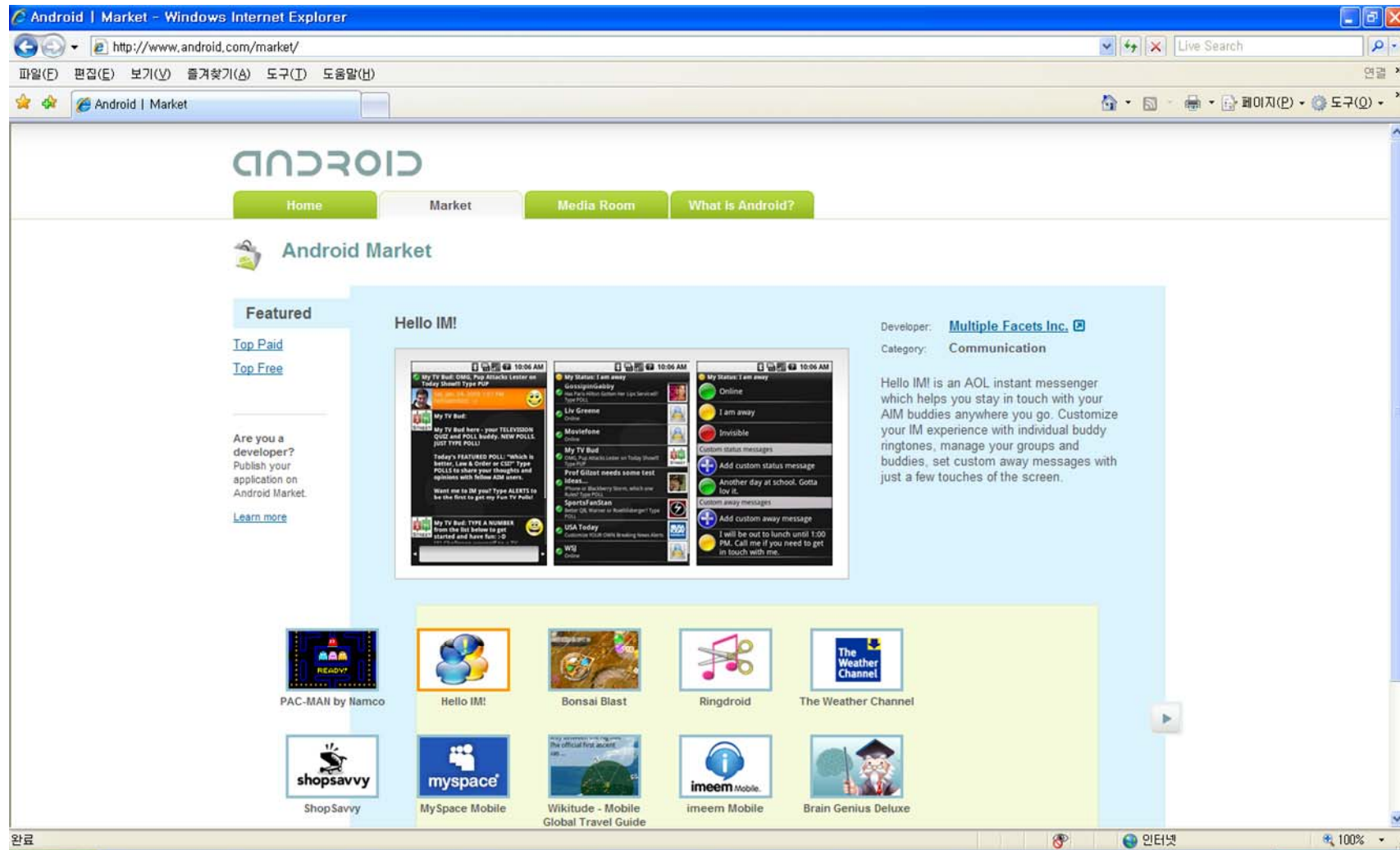


## App Widget Examples





Android Market Service : [www.android.com/market/](http://www.android.com/market/)





## Android Market Client

<http://www.kandroid.org/board/board.php?board=androidmarket&command=body&no=17>



