INTRODUCCIÓN A LA SEGURIDAD EN APLICACIONES MÓVILES

android

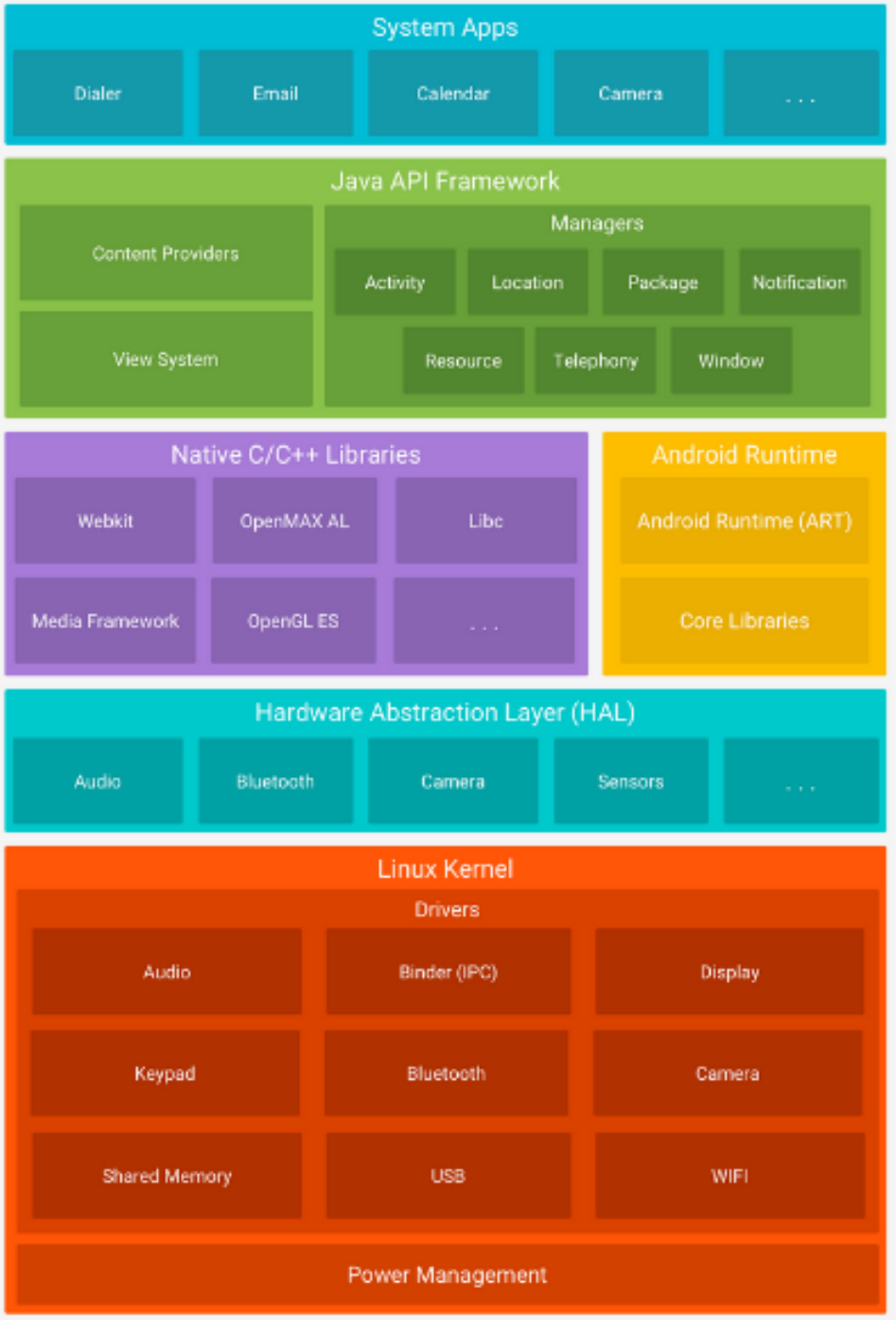
AGENDA

1. Conceptos básicos
   - SO Android
   - Estructura de archivos & sandboxing

2. Reverseando una app
   - emulador & adb
   - estructura apk & decompilación
   - manifest | activities | broadcasts
     providers | services

3. Atacando vulnerabilidades
   - Explotando componentes de Android

ANDROID SO



https://developer.android.com/guide/platform

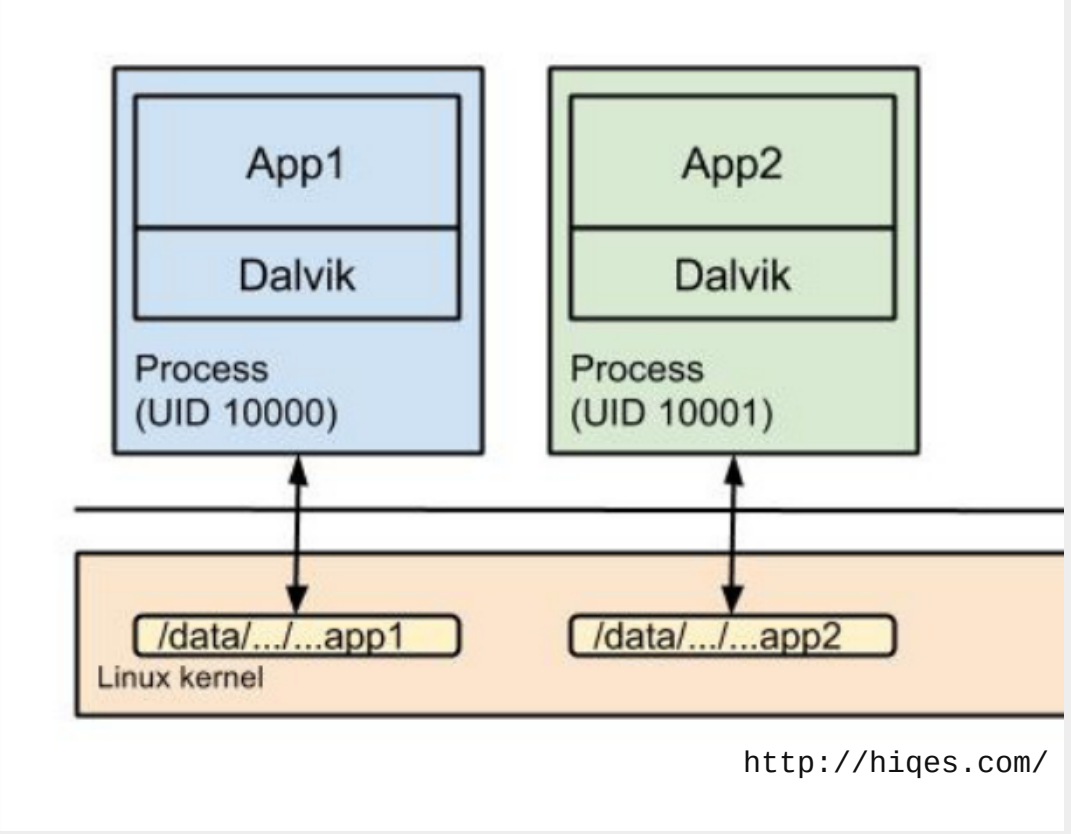
CONCEPTOS BÁSICOS

SANDBOXING

- separacion de UID x app

- directorio de datos privado

- separacion en la ejecucion:
cada app cuenta con su propia Dalvik VM



1. Instalo app -> 2. SO asigna userID unico & setea perms
3. Ejecuto app -> 4. Cada proceso su propia Dalvik VM

CONCEPTOS BÁSICOS

ESTRUCTURA DE ARCHIVOS

```
/data/data ~ data de apps instaladas por usr

/data/app ~ apps instaladas por usuario
```

CONCEPTOS BÁSICOS

SANDBOXING

APP 1

```
vbox86p:/ # ps -A | grep insecure
u0_a106       3287   285 1021568 116888 ep_poll      f39e3bb9 S com.dns.insecurepass
```

```
vbox86p:/ # ls -la /data/data/com.dns.insecurepass/
total 72
drwx------   7 u0_a106 u0_a106        4096 2020-09-17 22:28 .
drwxrwx--x 162 system  system        12288 2020-09-17 22:03 ..
drwxrws--x   2 u0_a106 u0_a106_cache  4096 2020-07-13 01:06 cache
drwxrws--x   2 u0_a106 u0_a106_cache  4096 2020-07-13 01:06 code_cache
drwxrwx--x   2 u0_a106 u0_a106        4096 2020-07-13 01:38 databases
drwxrwx--x   2 u0_a106 u0_a106        4096 2020-07-13 01:07 files
```

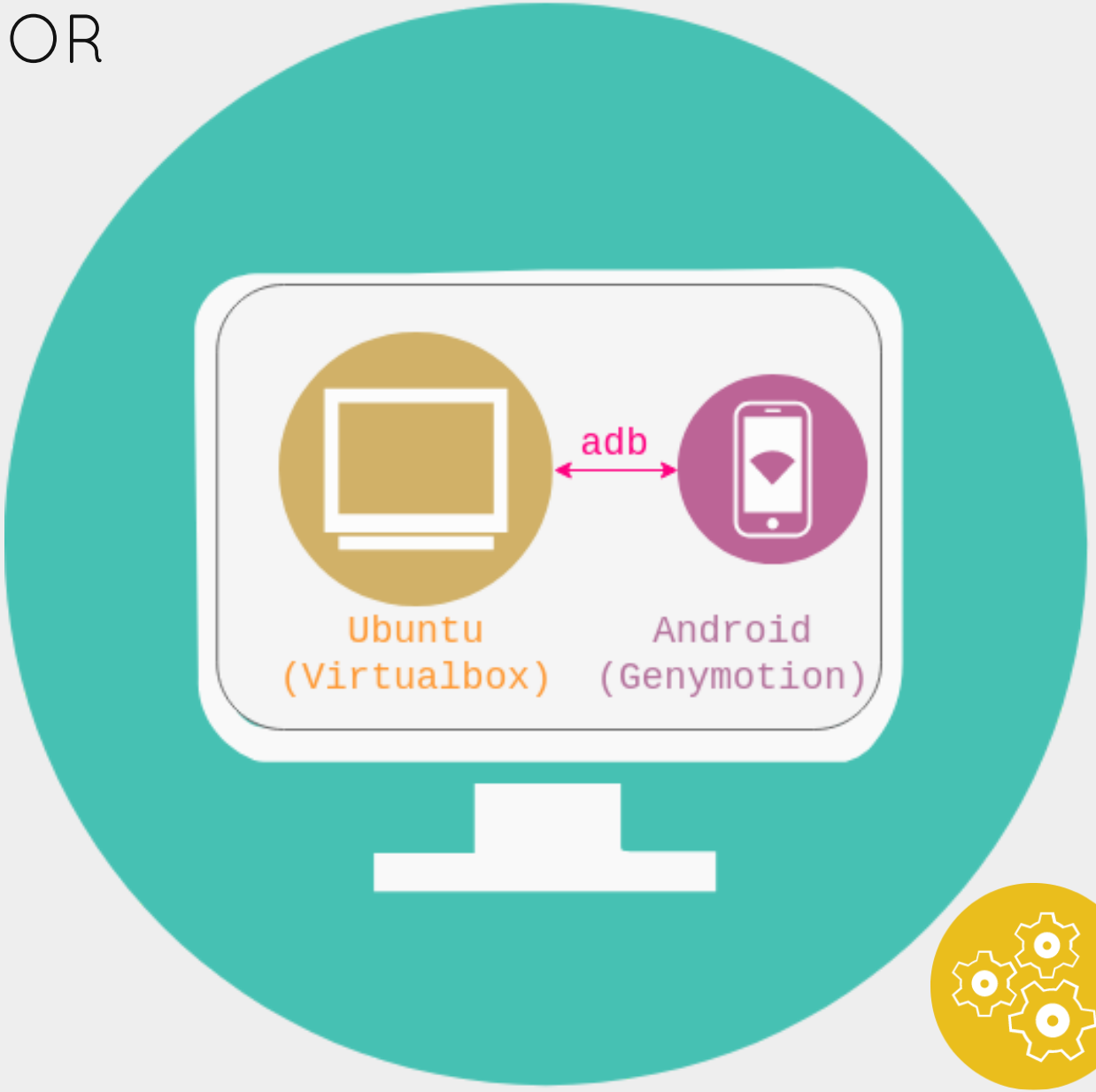
APP 2

```
127|vbox86p:/ # ps -A | grep diva
u0_a144       3446   285  996480 104536 ep_poll      f39e3bb9 S jakhar.aseem.diva
```

```
vbox86p:/ # ls -la /data/data/jakhar.aseem.diva/
total 80
drwxr-x--x   8 u0_a144 u0_a144        4096 2020-09-17 22:28 .
drwxrwx--x 162 system  system        12288 2020-09-17 22:03 ..
drwxrwx--x   2 u0_a144 u0_a144        4096 2020-08-22 19:57 app_textures
drwx------   3 u0_a144 u0_a144        4096 2020-08-22 19:58 app_webview
drwxrws--x   4 u0_a144 u0_a144_cache  4096 2020-08-22 19:57 cache
drwxrws--x   2 u0_a144 u0_a144_cache  4096 2020-07-24 23:00 code_cache
drwxrwx--x   2 u0_a144 u0_a144        4096 2020-07-24 23:00 databases
```

REVERSEANDO UNA APP

EMULADOR



REVERSEANDO UNA APP

QUÉ ES UN APK?

-> Google play apps

-> zip con código y resources

-> com.package.name.app

REVERSEANDO UNA APP
CONSIGUIENDO LOS APKS

- Método 1: de Internet -> apk downloader para chrome (apkcombo, apkpure)

- Método 2: del teléfono -> ADB

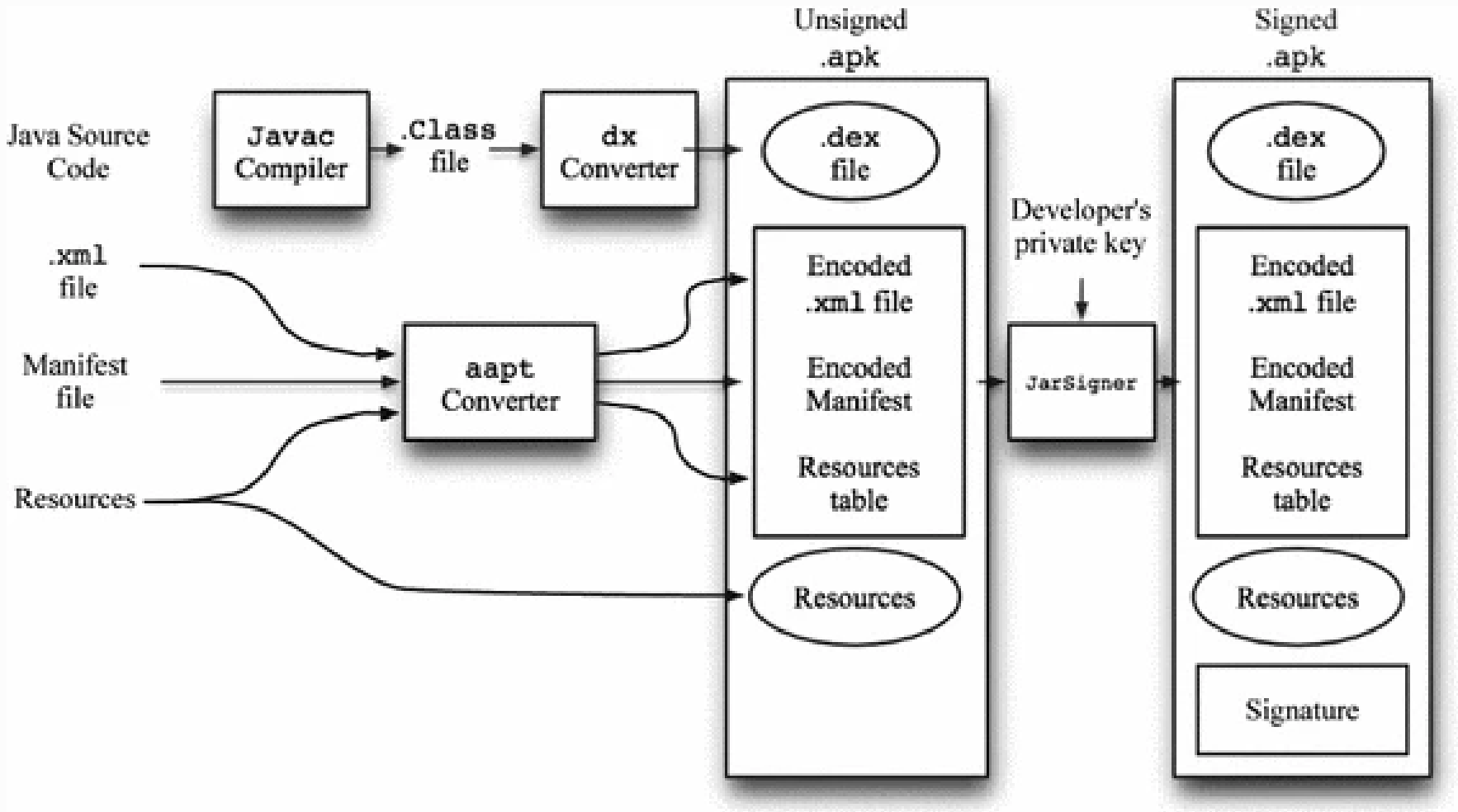
  -> command line tool that lets you communicate with an emulator/device

```
[VM]$ adb shell
[VM]$ adb logcat

[VM]$ adb install app.apk
[VM]$ adb push <src> <dest>
[VM]$ adb pull <src> <dest>
```

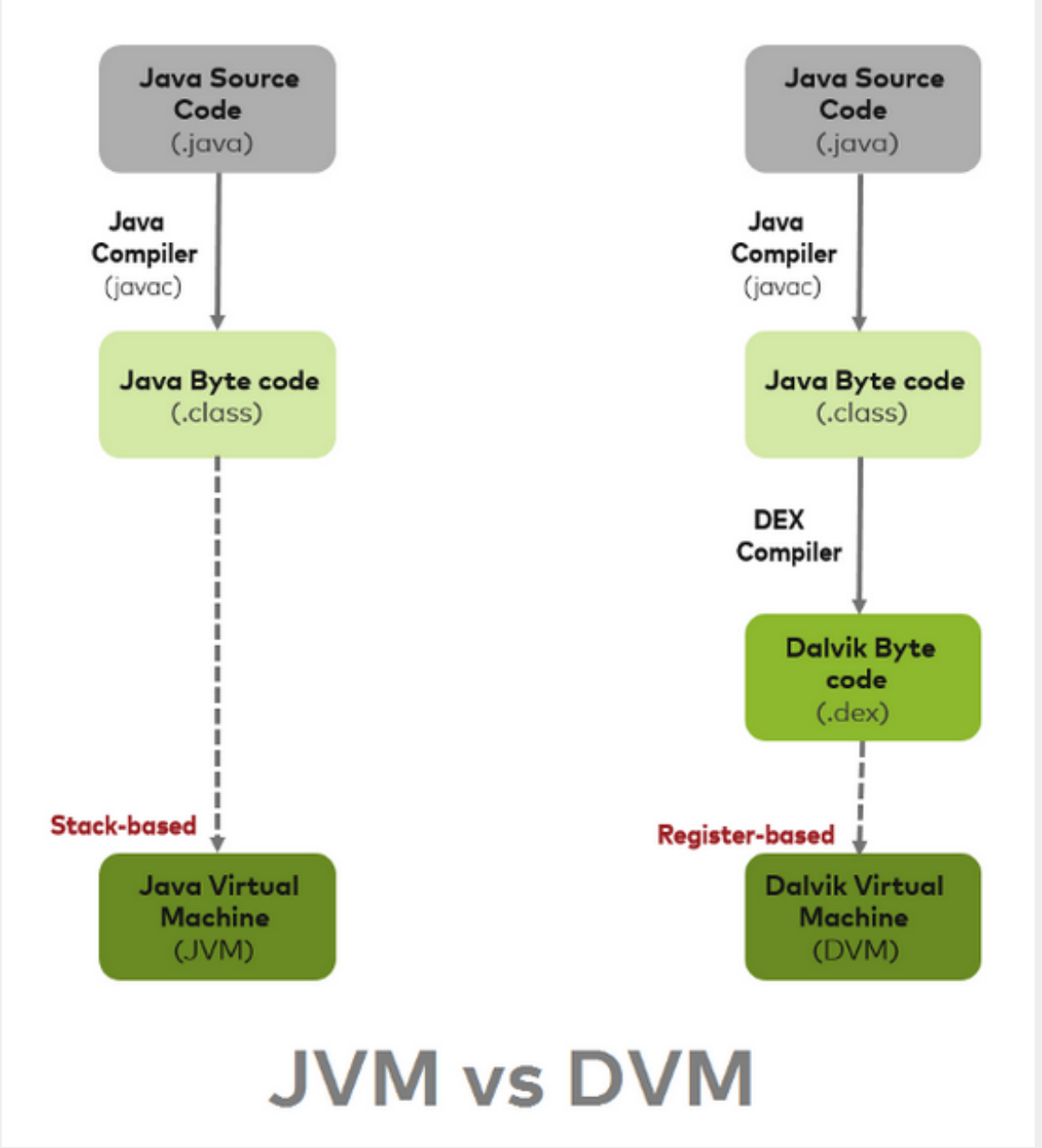
REVERSEANDO UNA APP

ESTRUCTURA DE UN APK



(RE)VERSEANDO UNA APP

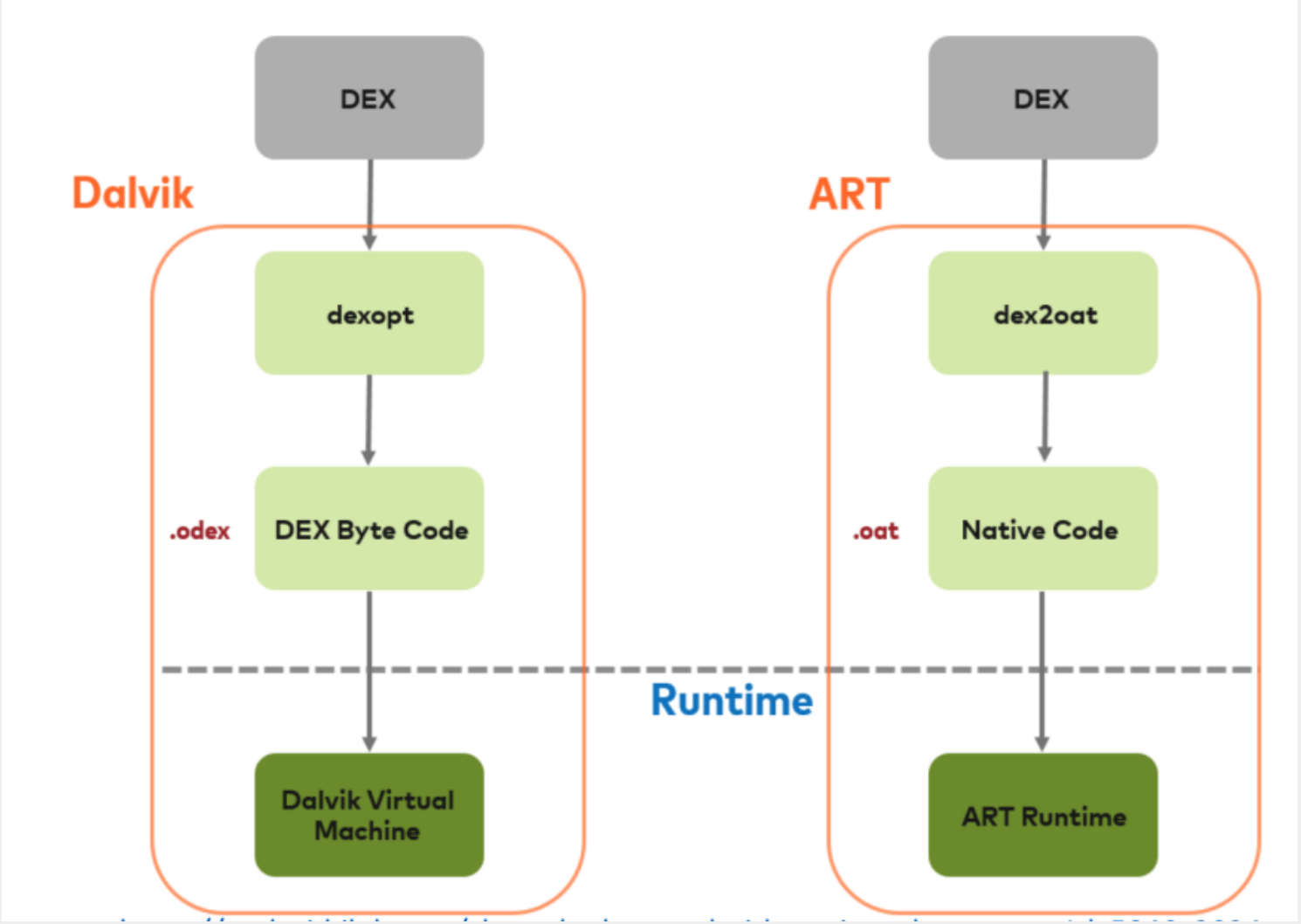
CONSTRUYENDO UN APK



REVERSING DE UNA APP

DALVIK VM VS ART VM

JUST IN TIME VS AHEAD OF TIME



REVERSEANDO UNA APP

DECOMPILANDO UN APK

DECOMPILANDO APP: DIVA

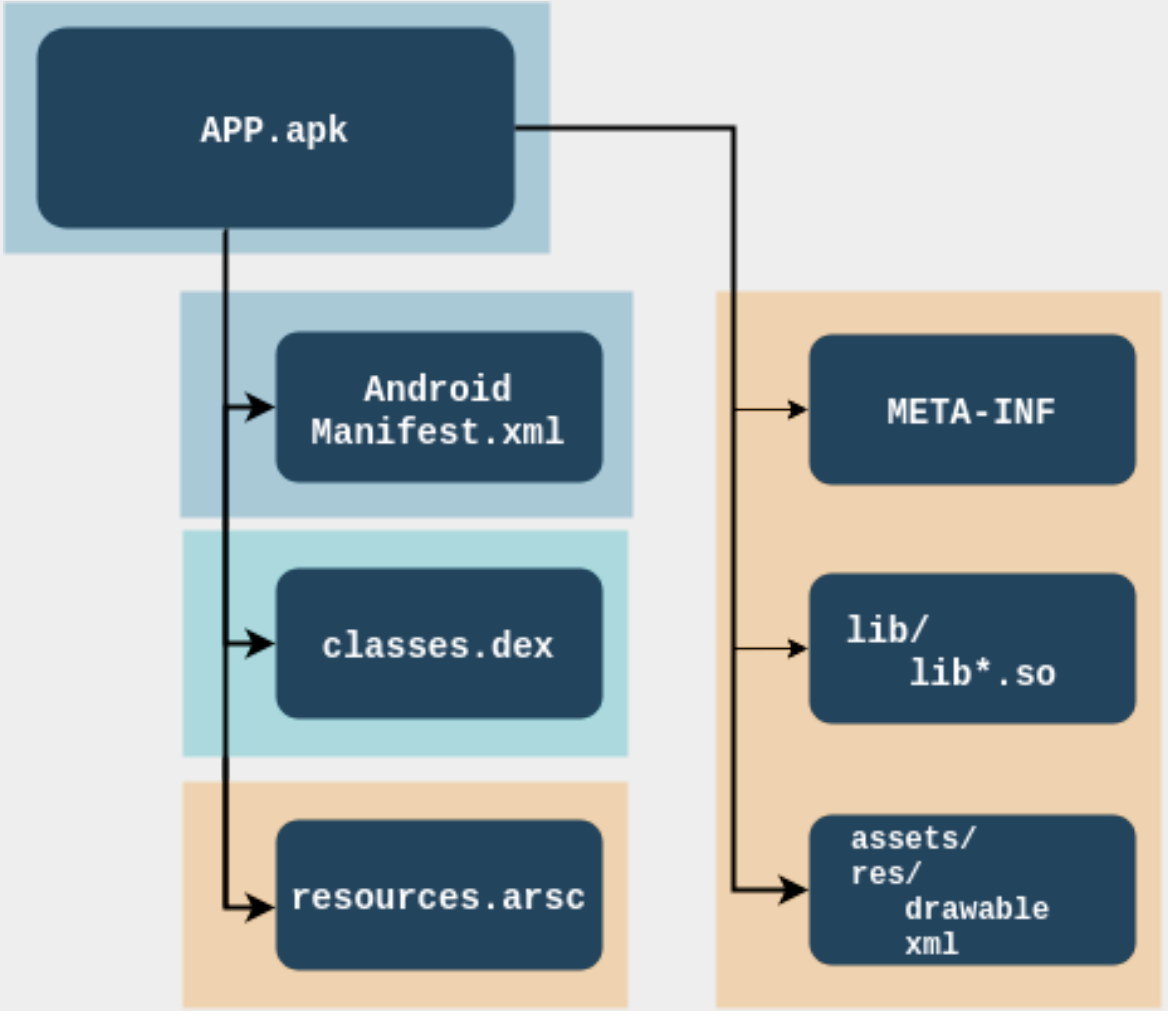
```
phone]$ pm list packages -f | grep appName
phone]$ pm path appName


VM]$ adb pull /data/app/com.app.name.apk

VM]$ unzip -e app.apk -d contents   dex == binary dalvik bytecode
VM]$ apktool d app.apk              dex ~> smali
VM]$ jadx app.apk  -d dir           dex ~> java code
VM]$ jadx-gui app.apk
```

REVERSING DE UNA APP

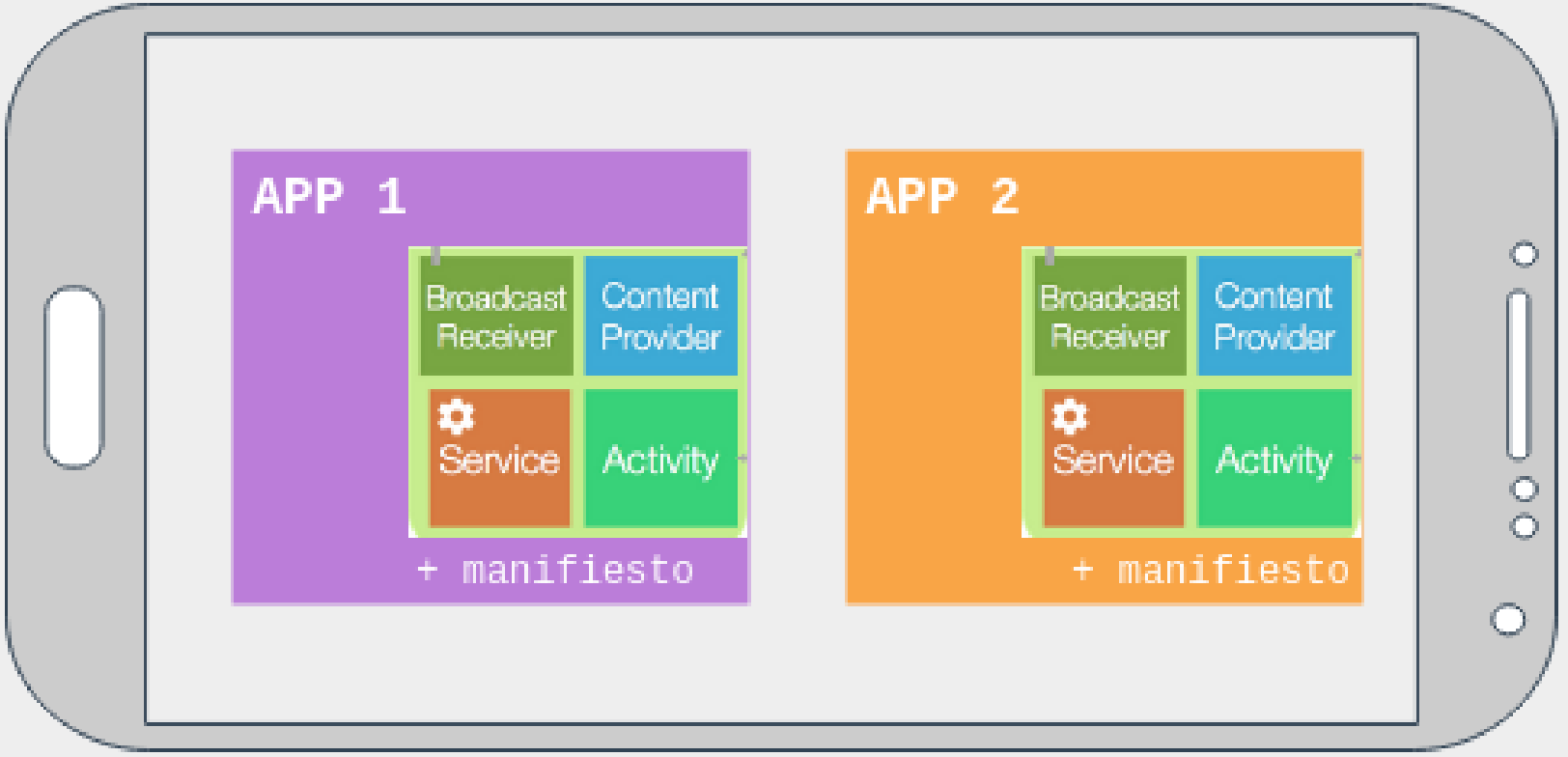
DECOMPILANDO UN APK



VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

ACTIVITIES | CONTENT PROVIDERS

SERVICES | BROADCAST RECEIVERS



VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

MANIFEST -> archivo de configuración
-> permisos ~ entry point
-> componentes ~ android:name

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.htbridge.pivaa">
    <uses-sdk android:minSdkVersion="19" android:targetSdkVersion="26"/>

    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>

    <application android:theme="@style/AppTheme" android:debuggable="true" android:allowBackup="true">
        <activity android:label="@string/app_name" android:name="com.htbridge.pivaa.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name="com.htbridge.pivaa.EncryptionActivity"/>
        <activity android:name="com.htbridge.pivaa.WebviewActivity"/>
        <activity android:name="com.htbridge.pivaa.DatabaseActivity"/>

        <service android:name="com.htbridge.pivaa.handlers.VulnerableService" android:exported="true"/>

        <receiver android:name="com.htbridge.pivaa.handlers.VulnerableReceiver" android:exported="true">
            <intent-filter>
                <action android:name="service.vulnerable.vulnerableservice.LOG"/>
            </intent-filter>
        </receiver>

        <provider android:name="com.htbridge.pivaa.handlers.VulnerableContentProvider" android:exported="t

    </application>
</manifest>
```

VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

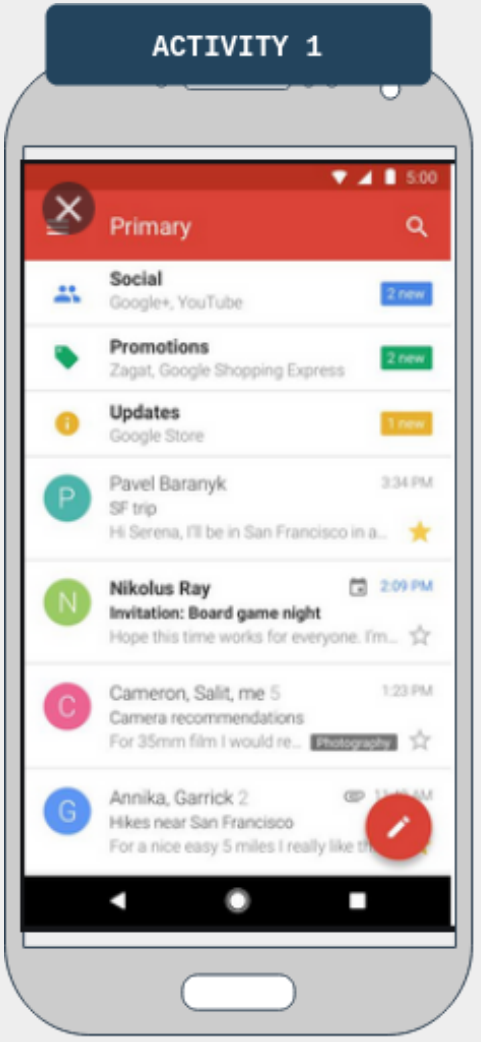
ANALISIS DE APP: INSECUREPASS

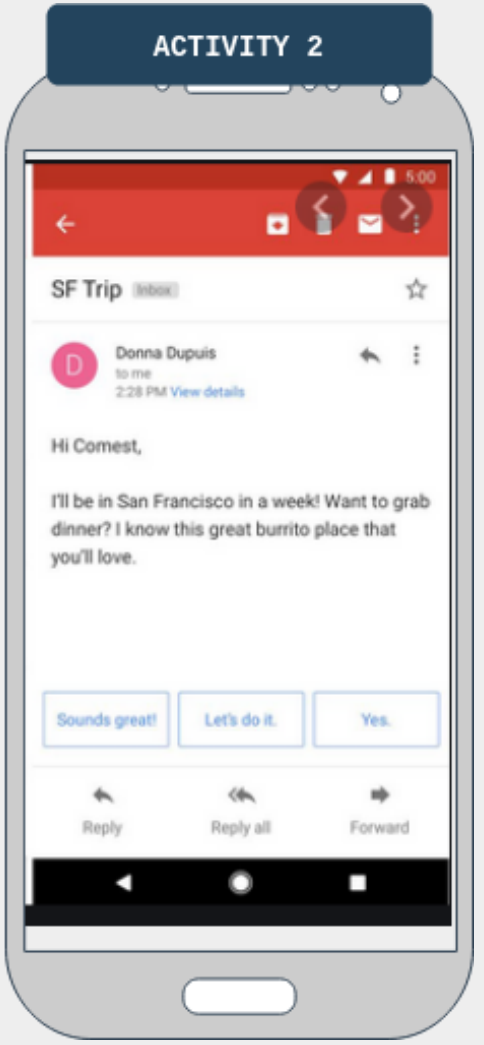
```
-> extract
-> decompile
-> manifest analysis
```

VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

ACTIVITIES

- Representa una pantalla con una interfaz de usr
- Se corresponde con una actividad de la usr: mandar un mail, sacar una foto

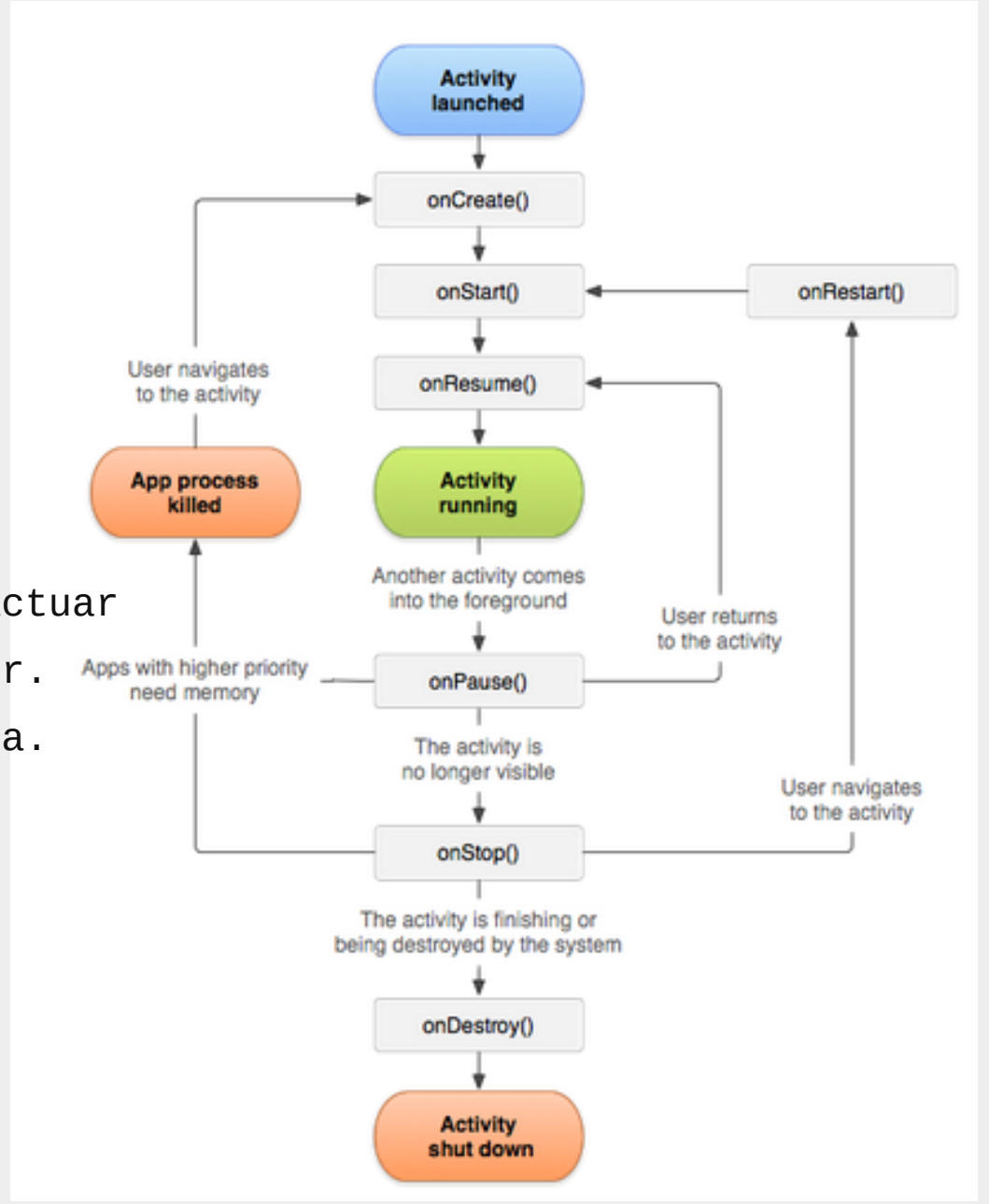




ACTIVITIES

LIFECICLE

onCreate()-> se crea por primera vez.
onStart() -> es visible al usuario.
onResume()-> le usr comienza a interactuar
onStop() -> no es más mostrada al usr.
onDestroy()-> es destruida del sistema.



- Entry points
- Control flow

https://developer.android.com/guide/components/activities/activity-lifecycle

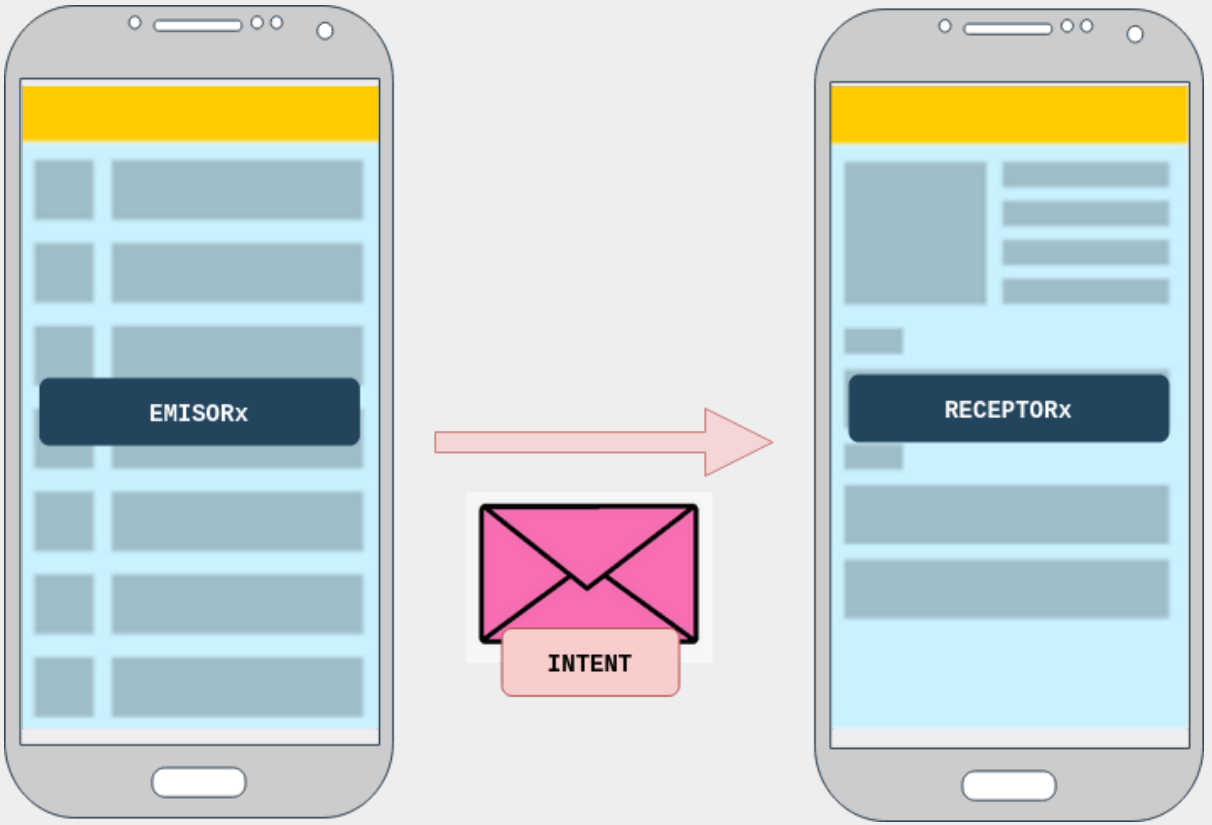
VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

ANALISIS DE APP: INSECUREPASS

```
-> activities
-> code flow analysis
```

VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

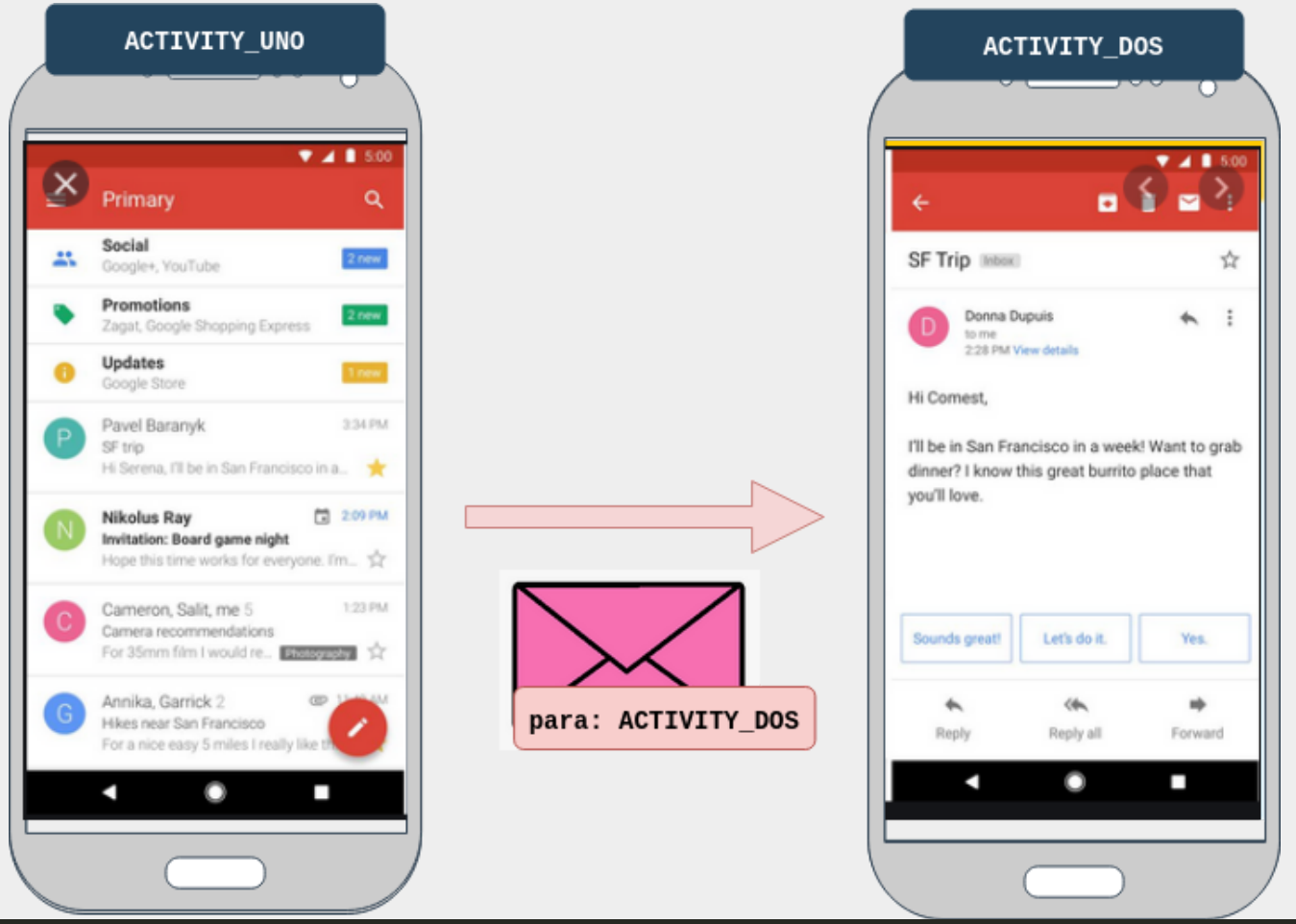
INTENTS



son mensajes | comunicación entre componentes (== app o != app)

VULNERABILIDADES ANDROID

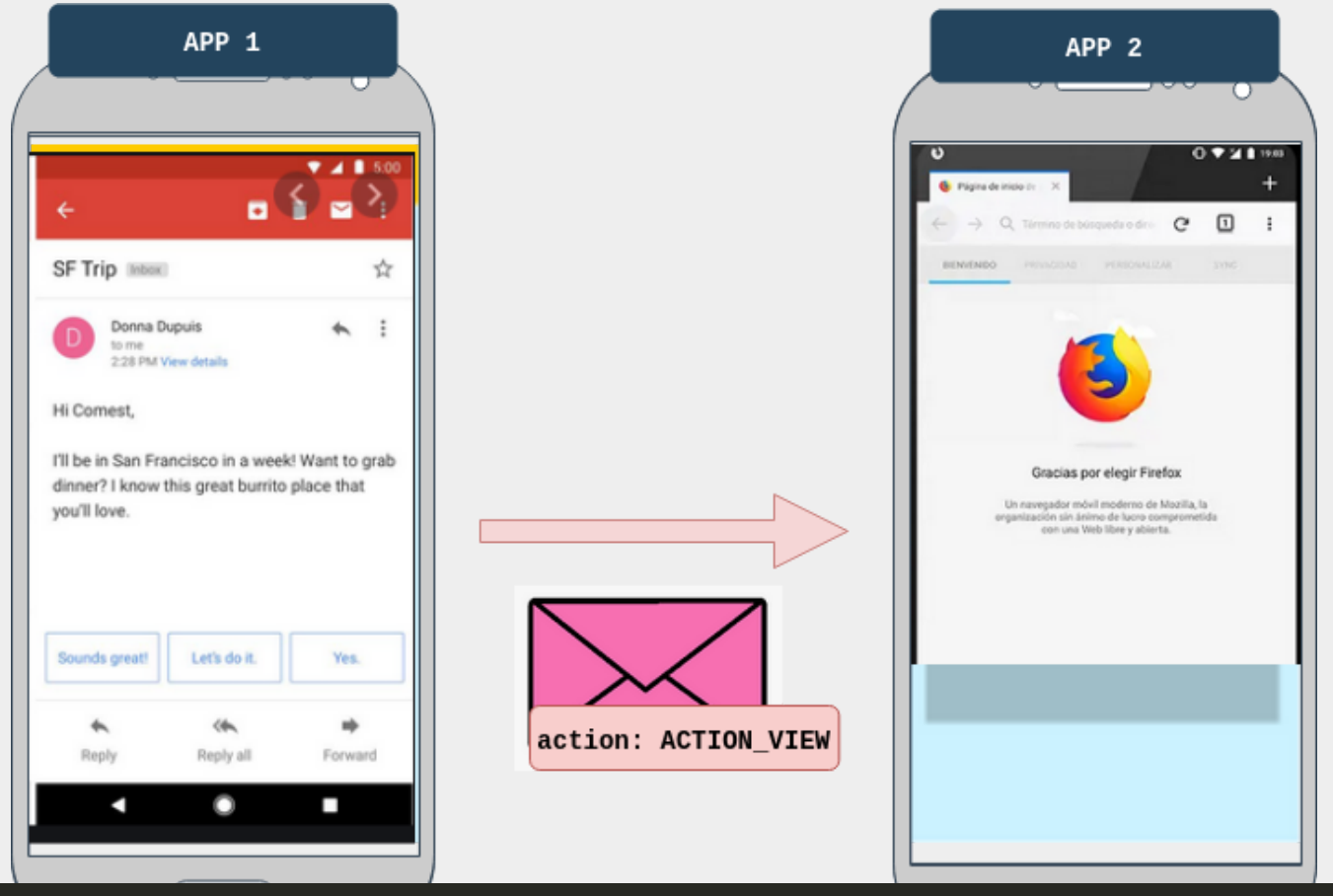
EXPLOITING ANDROID COMPONENTS




```
Intent msj = new Intent(this, Activity_dos.class);
msj.putExtra("Info", "utilizada por Activity_dos");
startActivity(msj);
```

VULNERABILIDADES ANDROID

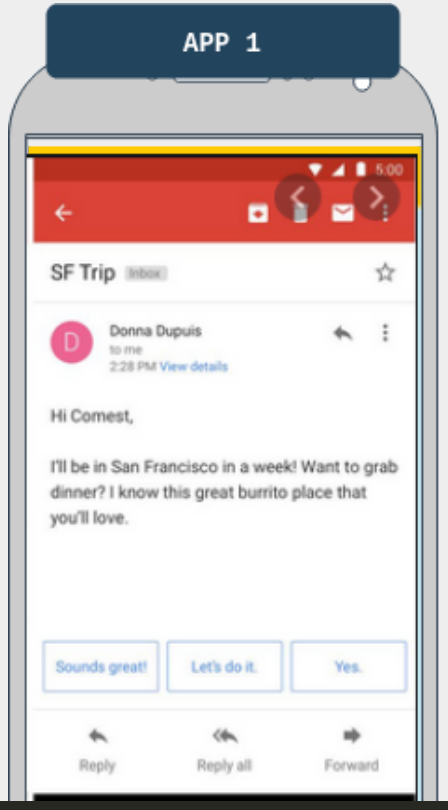
EXPLOITING ANDROID COMPONENTS

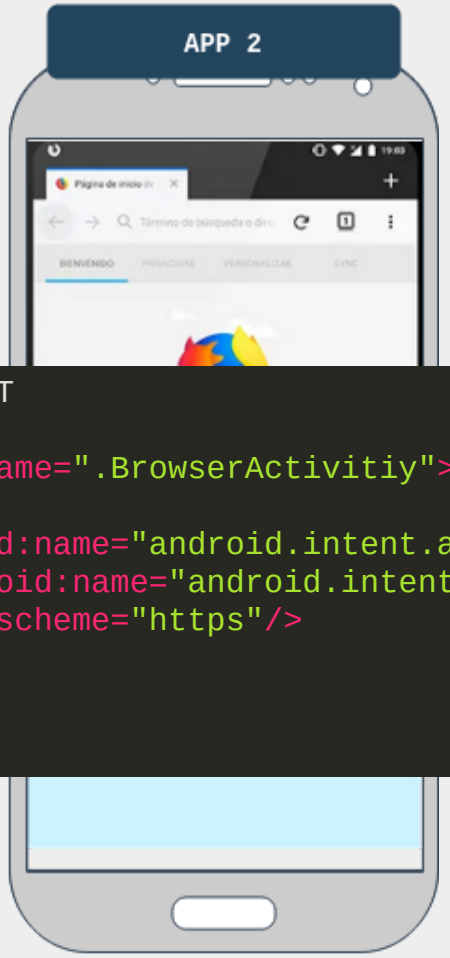



```
String url = "https://www.link.com/";
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(Uri.parse(url));
startActivity(i);
```

VULNERABILIDADES ANDROID

EXPLOITING ANDROID COMPONENTS






```
1 FIREFOX APP MANIFEST
2
3 <activity android:name=".BrowserActivitiy">
4   <intent-filter>
5      <action android:name="android.intent.action.VIEW" />
6      <category android:name="android.intent.category.DEFAUL
7      <data android:scheme="https"/>
8   </intent-filter>
9 </activity>
```

```
GMAIL APP

String url = "https://www.link.com/";
Intent i = new Intent(android.intent.action.VIEW);
i.setData(Uri.parse(url));
startActivity(i);
```

VULNERABILIDADES ANDROID

EXPLOITING ANDROID COMPONENTS

INSECUREPASS
BYPASSING AUTHENTICATION

-> análisis del manifest

-> code flow

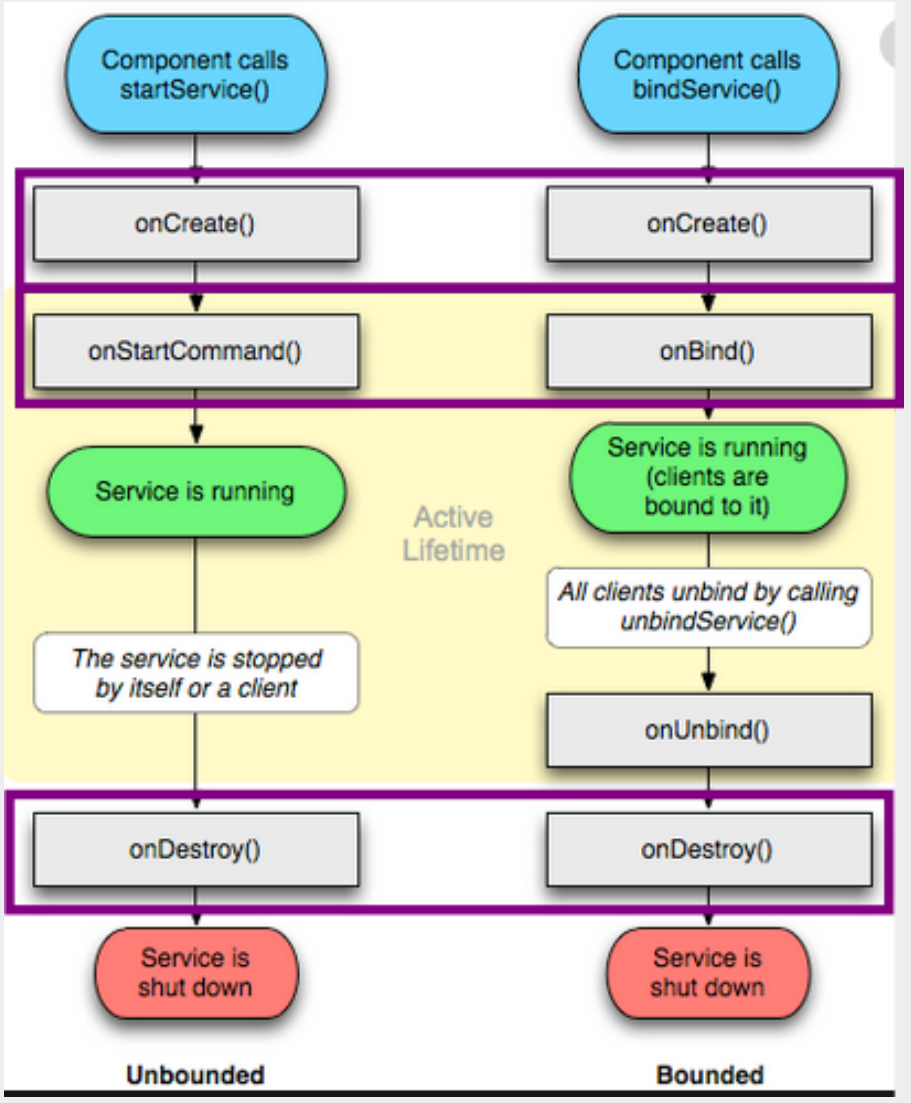
-> activities exposed

-> am start -n <package-name>/<activity-name>

SERVICIOS

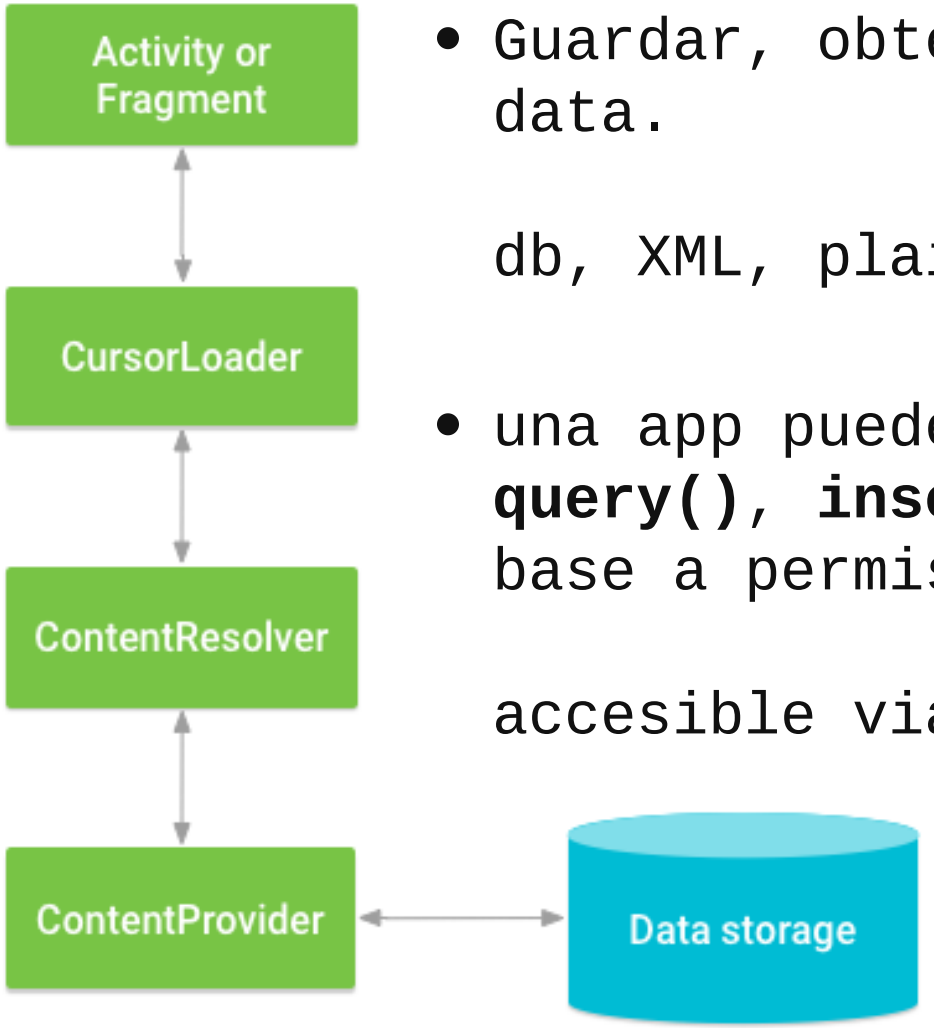
LIFECICLE

- corren en background
- operaciones de largo término
- sin interfaz de usuarix
- ej. descargar un archivo, sincronizar mails, reproducir música



https://developer.android.com/guide/components/activities/activity-lifecycle

CONTENT PROVIDER



- Guardar, obtener y compartir app data.

  db, XML, plaintext

- una app puede acceder con métodos **query()**, **insert()**, **delete()** y en base a permisos

  accesible via uri content://

VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

CATCH APP
EXPOSED PII

```
-> grep -iRn "content://"
-> content query --uri content://<provider-uri>

-> provider & data exposed
```

BROADCAST RECEIVERS

- broadcast: es un msg que recibe una app (eventos del sistema)
- al recibirlo via Receiver: realiza una acción
- ej. llamada entrante -> se detiene la música
- Se envia un broadcast a otra app pasando un `Intent` to `sendBroadcast()`

VULNERABILIDADES ANDROID
EXPLOITING ANDROID COMPONENTS

INSECUREPASS
ENVIO DE SMS PREMIUM

```
-> análisis del manifest

-> code flow

-> broadcast exposed

-> am broadcast -a <receiverAction>
   -n <packageName>/<receiverName>
   --es <additionalData>
```