THESE AREN'T AN...DROIDS YOU'RE LOOKING FOR

ABSTRACT

Android is being pushed inside almost everything these days, people say to make things "smart". We tend to do more and more things on our phones, something used only to make calls (and play snake) some time ago, now is used to make bank transfers, store personal files and who knows what more. These small devices tend to get more and more power, but with great power comes great responsibility. The aim of this talk is to look on Android through security lens and check what's there to see.

TABLE OF CONTENTS

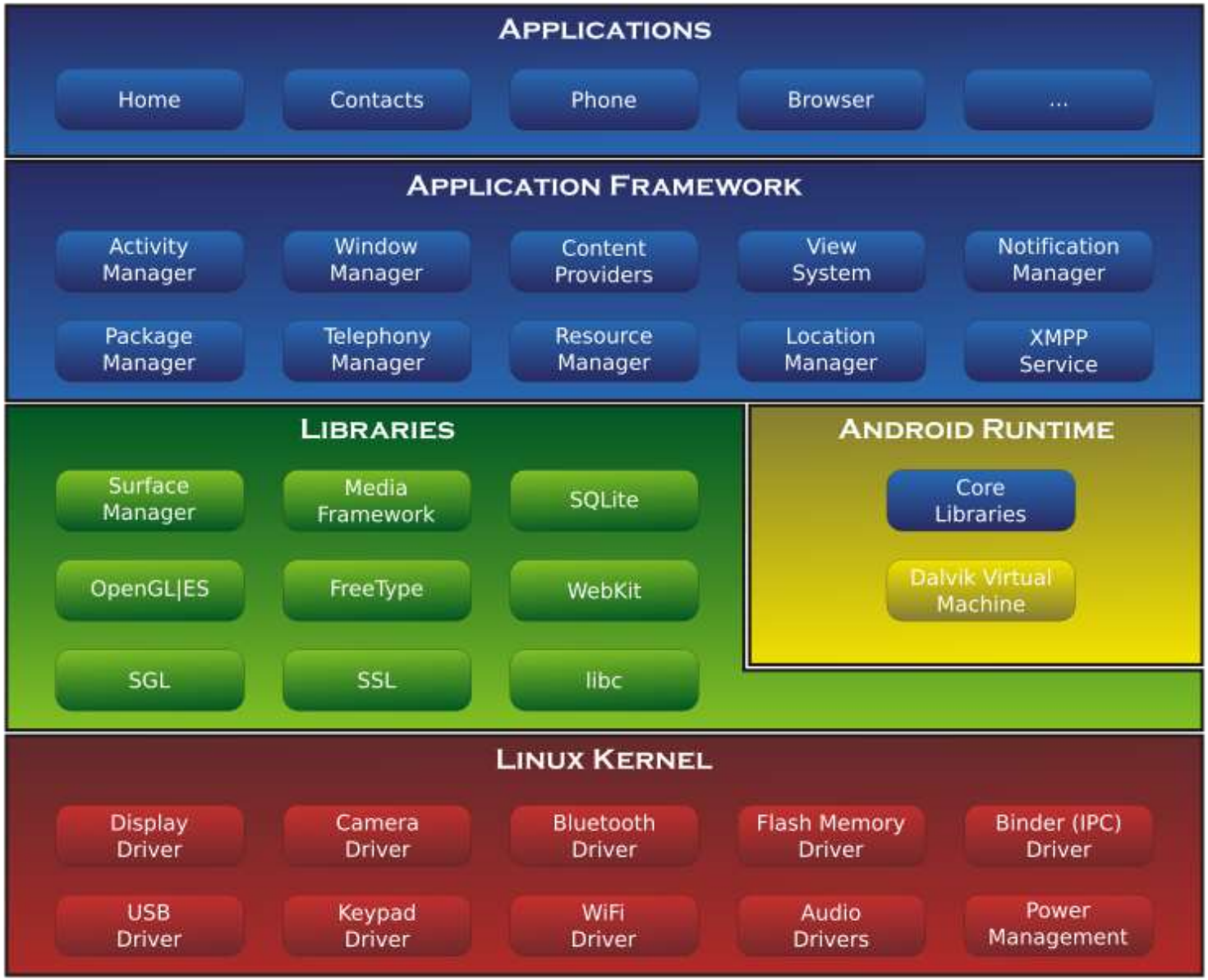
AND I AM...

- JAVA DEV :)
- SECURTY RESEARCHER
- http://blog.pentests.pl/
- @PentestsPL

ANDROID

- LINUX
- „MOBILE” OS
- OPEN SOURCE
- UPDATES – NOT FOR EVERYONE

WHAT'S INSIDE?



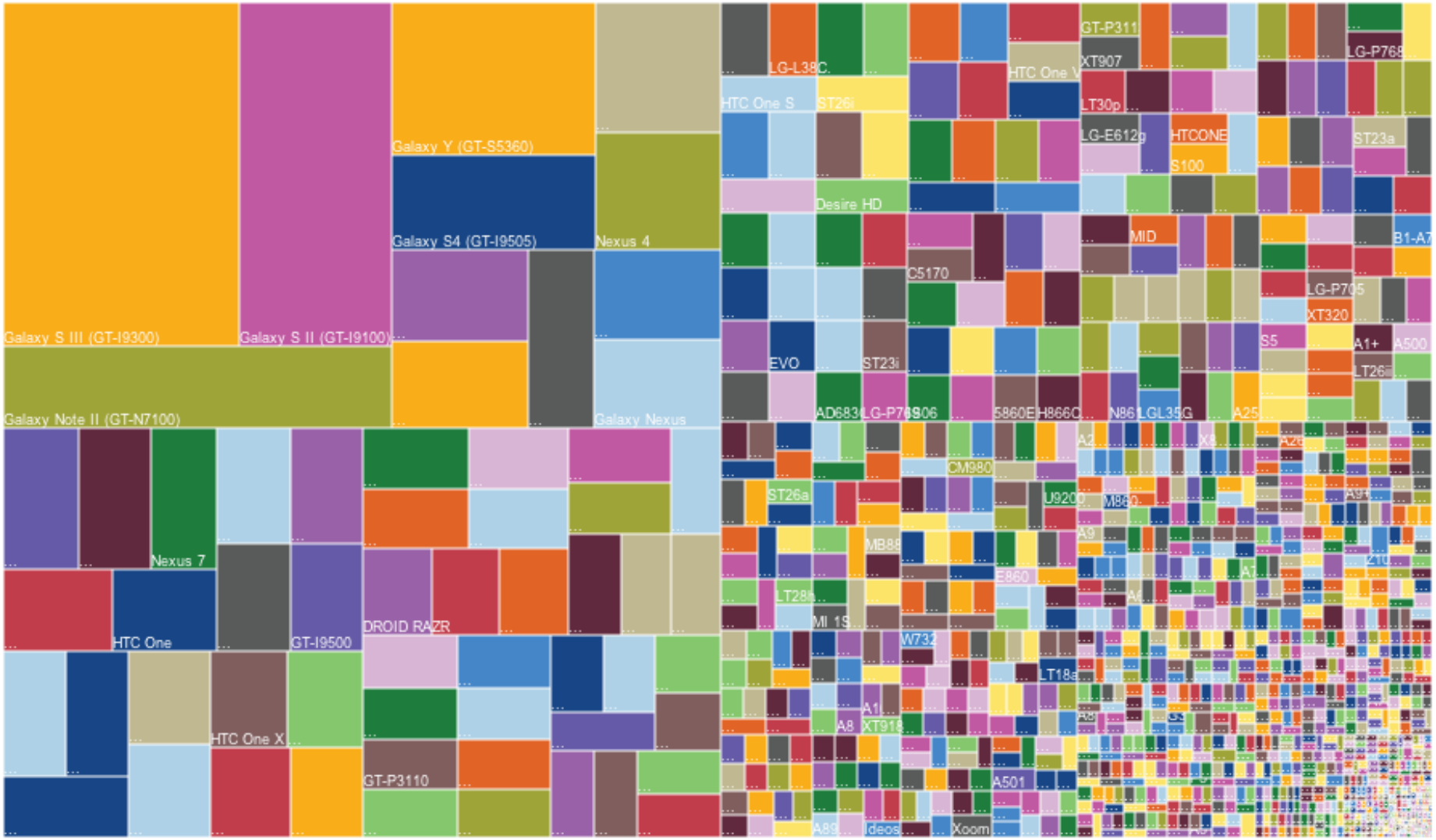
ANDROID APPS

- JAVA -> „FRONT END”
- DALVIK VIRTUAL MACHINE
- SUPPLIED TO DEVICE AS *.APK PACKAGE

ANDROID SECURITY

REALITY

ANDROID FRAGMENTATION



APPS AND SECURITY

- EVERYONE CAN CREATE APP AND PUSH IT TO GOOGLE PLAY STORE
- TWO SIDES:

CLIENT

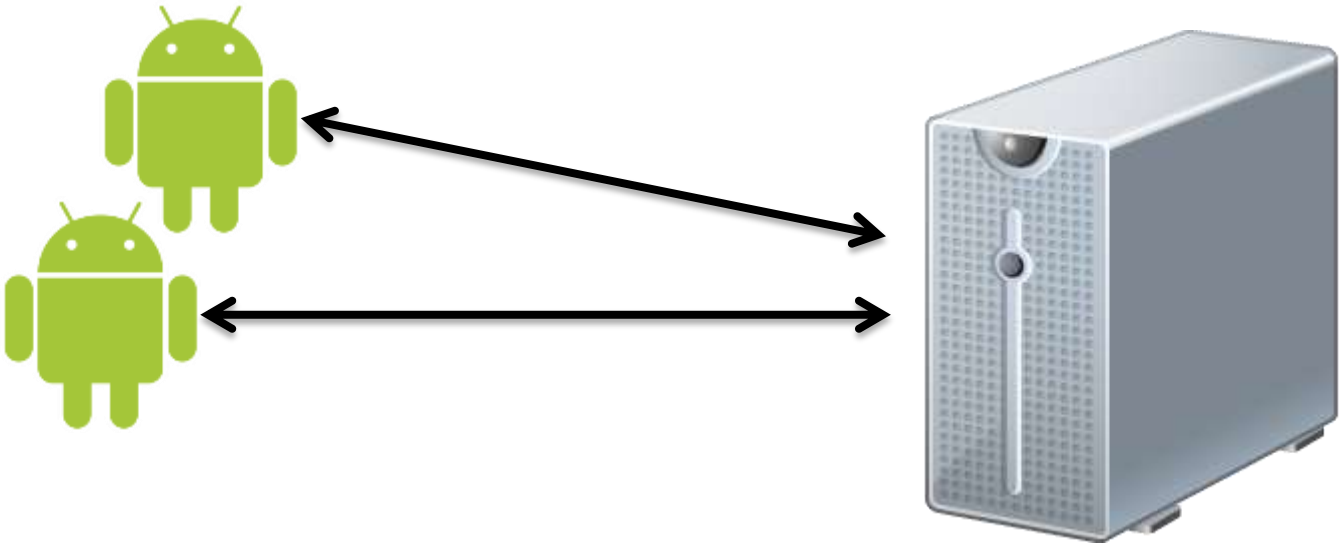
APP PROVIDER

CLIENT (PHONE OWNER)

- „Virus Shield” CASE
- FROM GOOGLE PLAY != 100% SAFE
- FAKE/PIRATED APPS

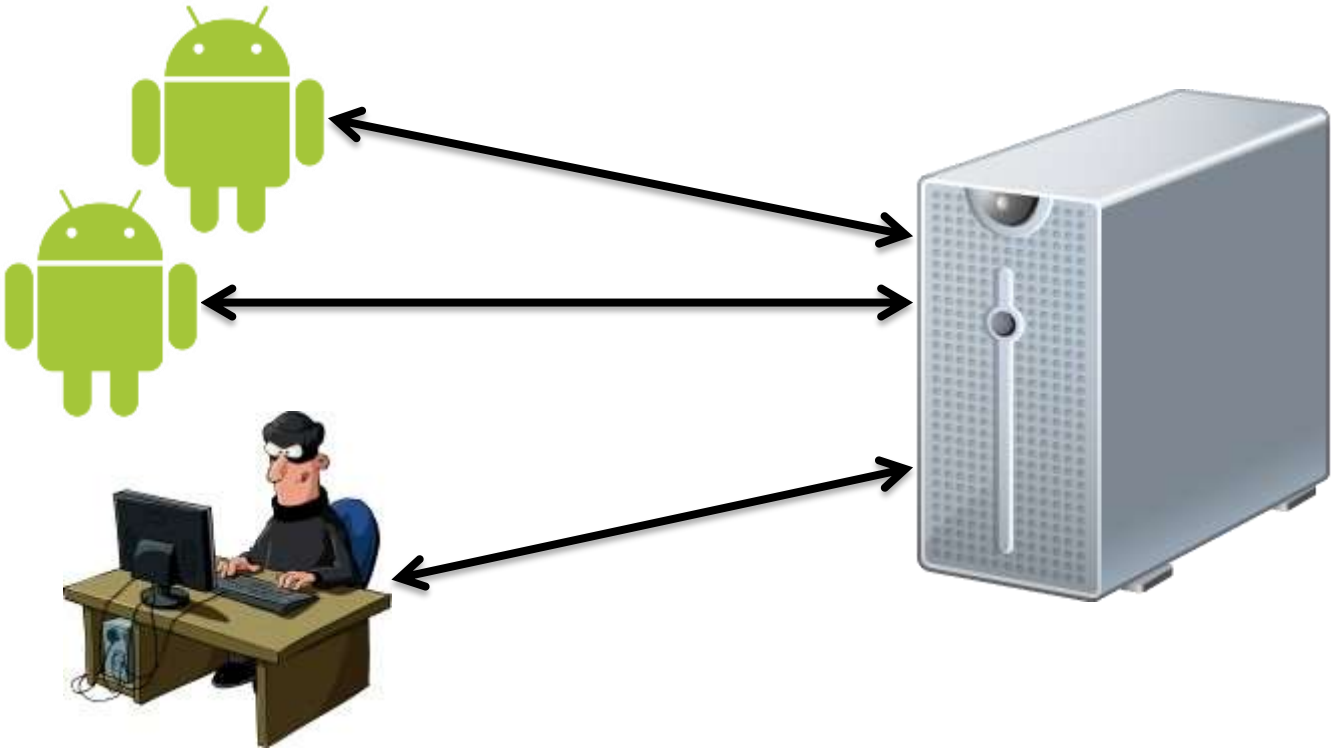
APP CREATOR/OWNER

- MOBILE APPS == CLIENT
- SERVER IS A PART OF AN APP
- APP CODE CAN BE REVERSED/CHANGED



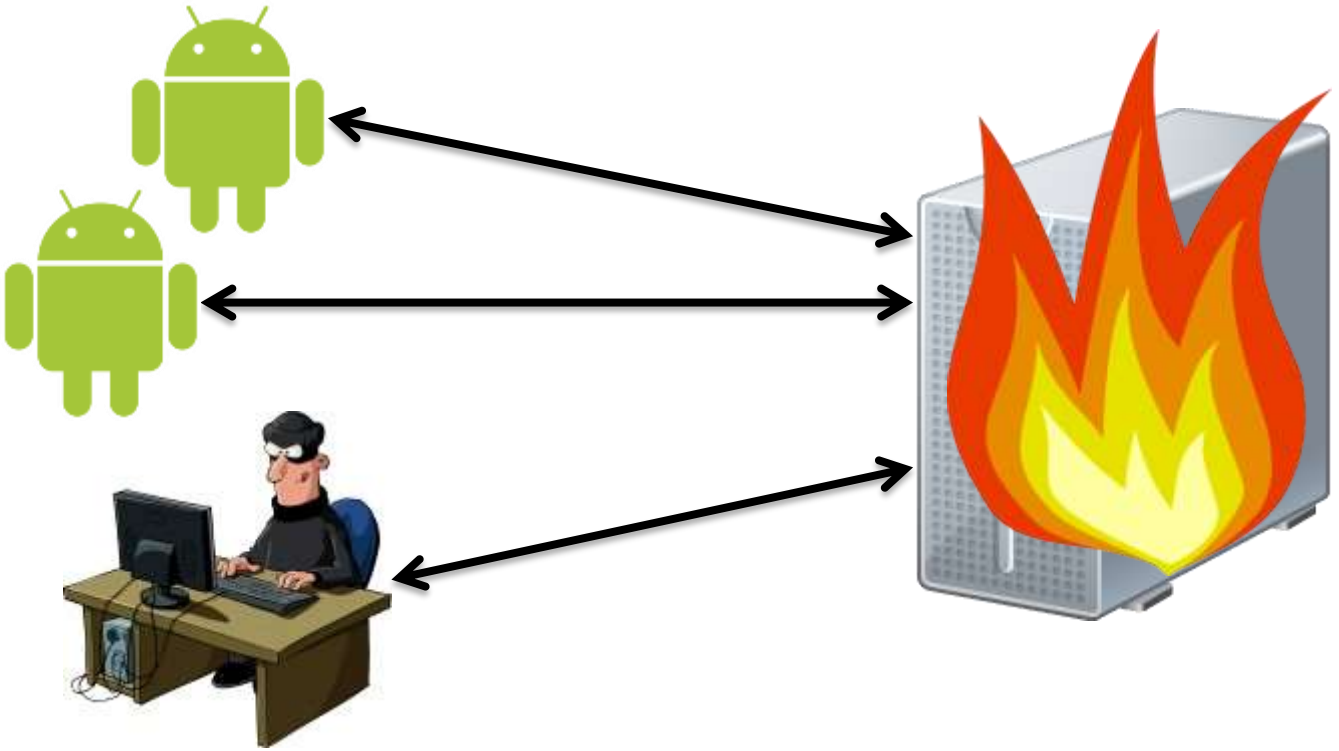
APP CREATOR/OWNER

- MOBILE APPS == CLIENT
- SERVER IS A PART OF AN APP
- APP CODE CAN BE REVERSED/CHANGED



APP CREATOR/OWNER

- MOBILE APPS == CLIENT
- SERVER IS A PART OF AN APP
- APP CODE CAN BE REVERSED/CHANGED



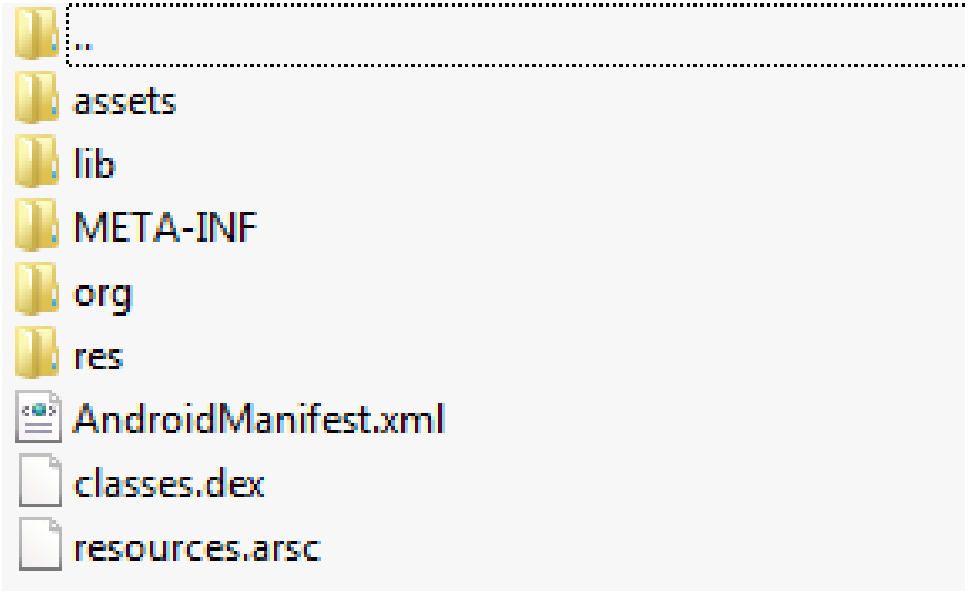
OWASP TOP 10 MOBILE

DiGGING INTO ANDROID APPS

- IT'S TRIVIAL TO EXTRACT APK FROM PHONE
- APK CAN BE ACCESSED, MODIFIED AND PUT BACK INTO PHONE

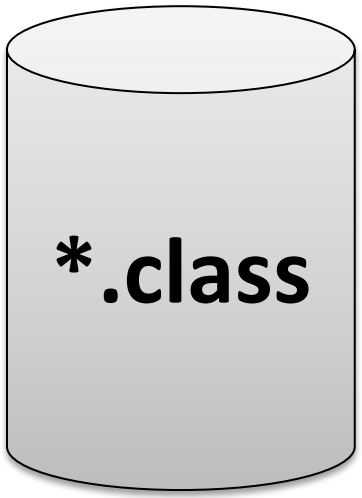
APK PACKAGE

- ZIP ARCHIVE
- AndroidManifest.xml
- classes.dex
- lib/
- res/
- other goodies



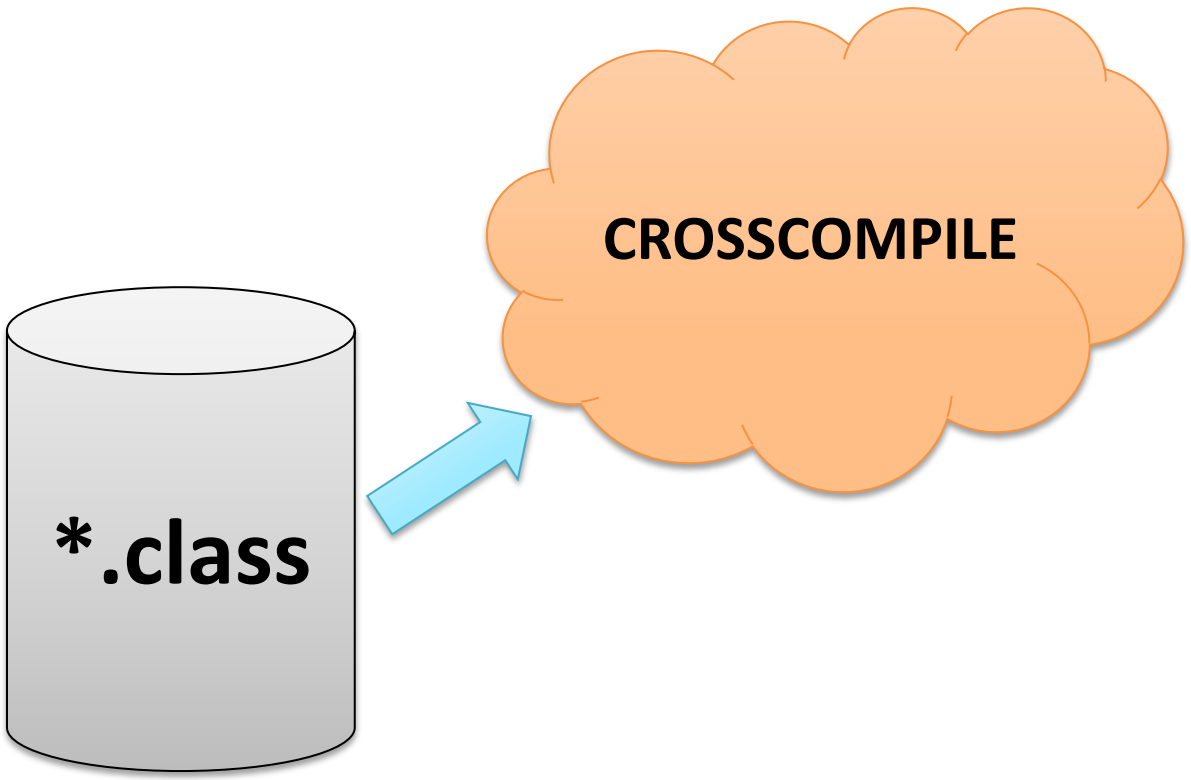
CLASSES.DEX

- APP SOURCE CODE IS COMPILED TO CLASS FILES



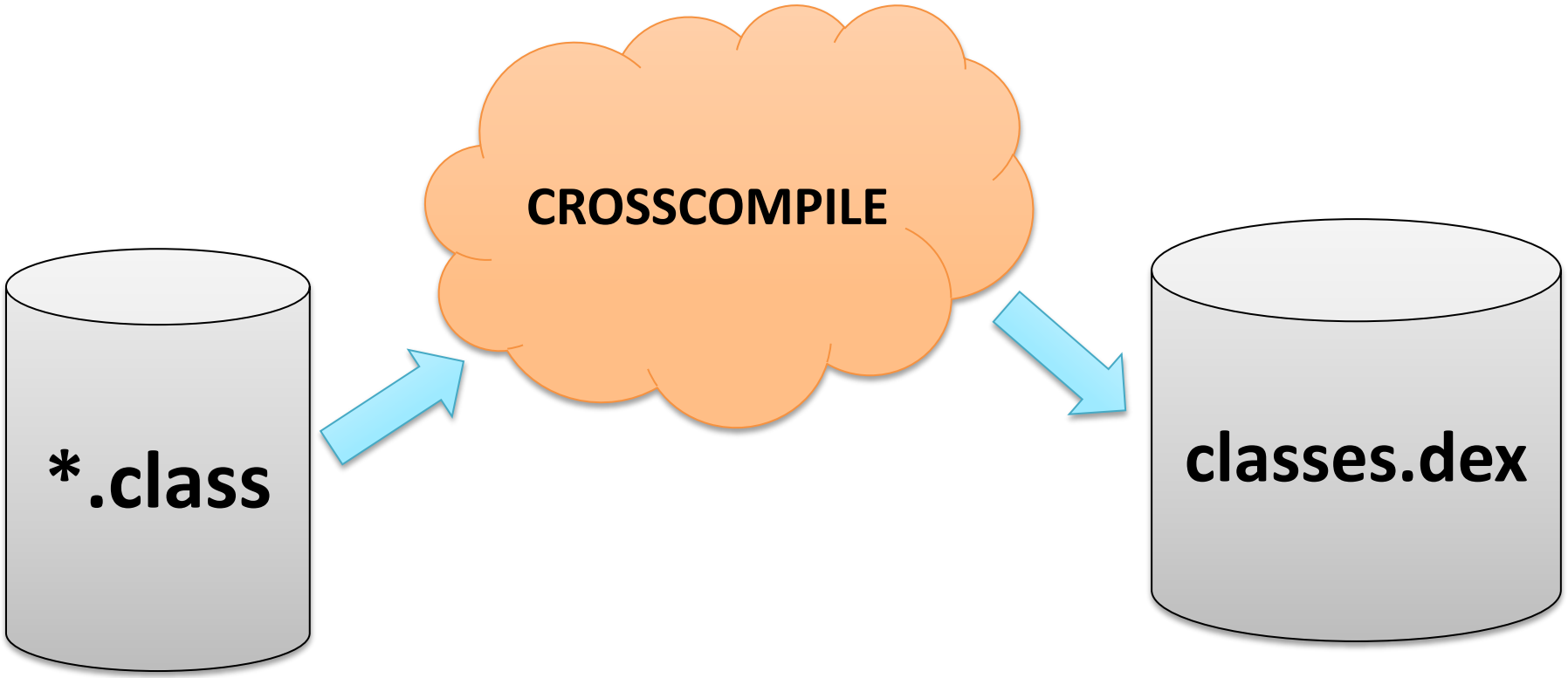
CLASSES.DEX

- CLASS FILES ARE CROSSCOMPILED

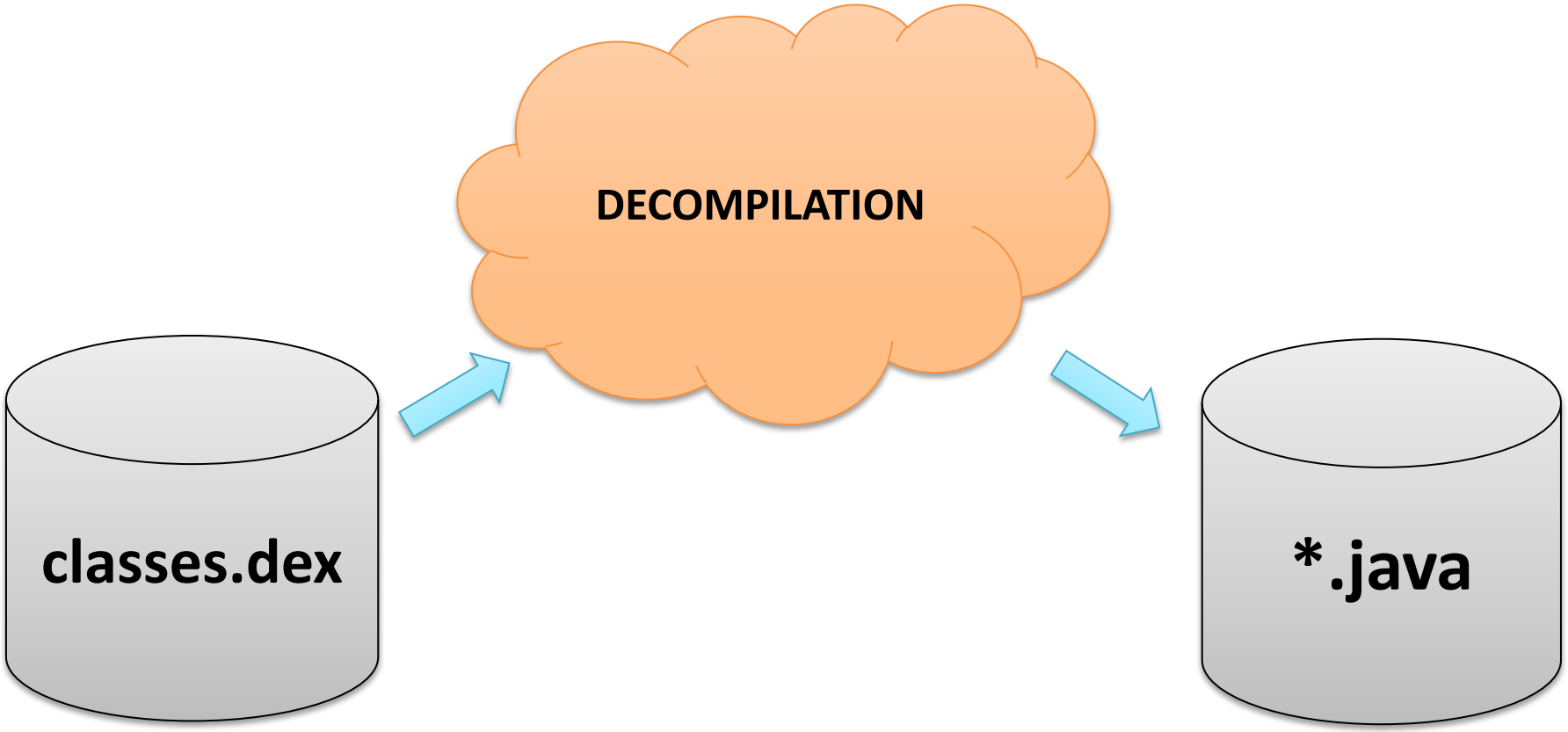


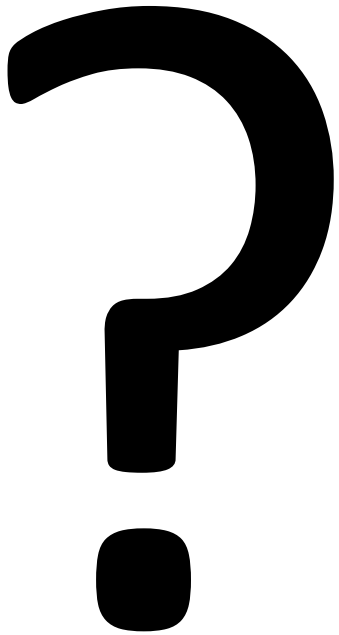
CLASSES.DEX

- CLASS FILES ARE CROSSCOMPILED INTO CLASSES.DEX



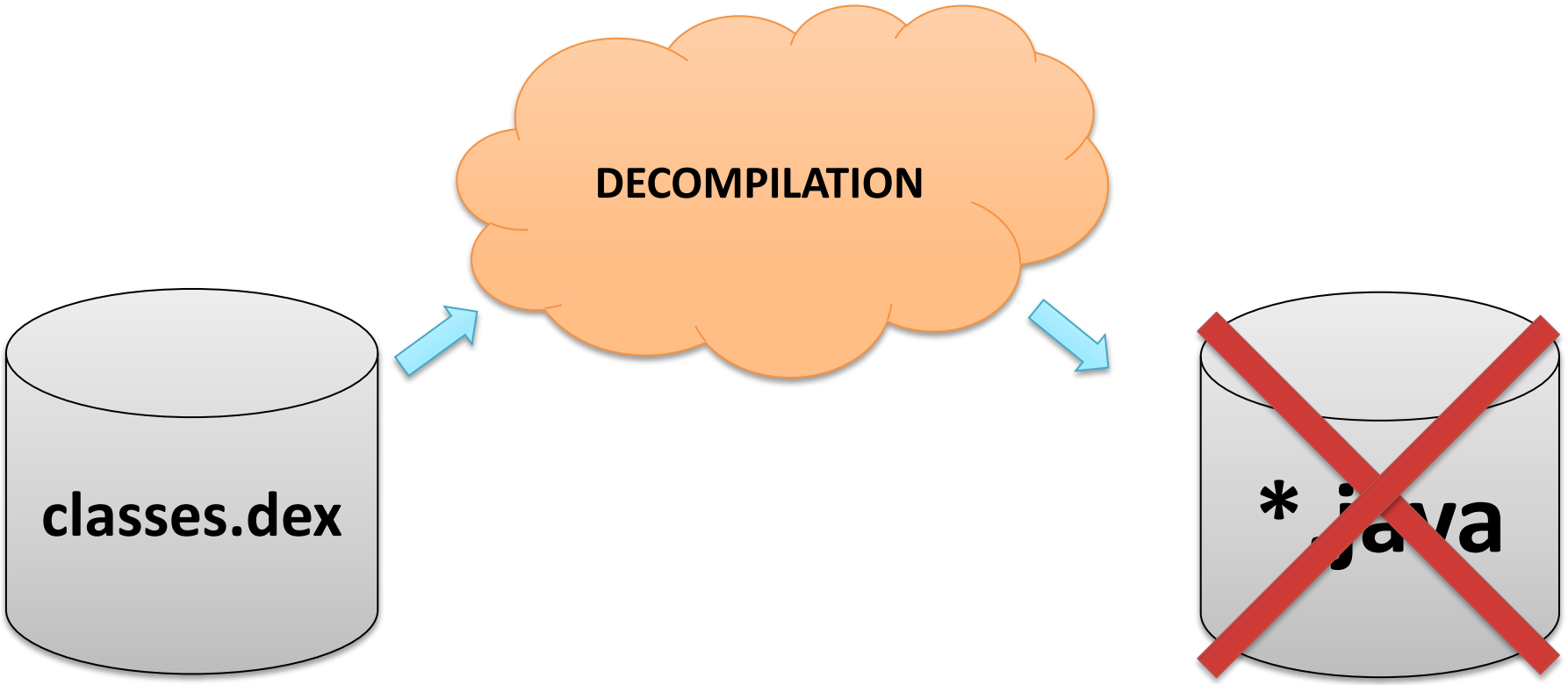
BACK TO *.JAVA?



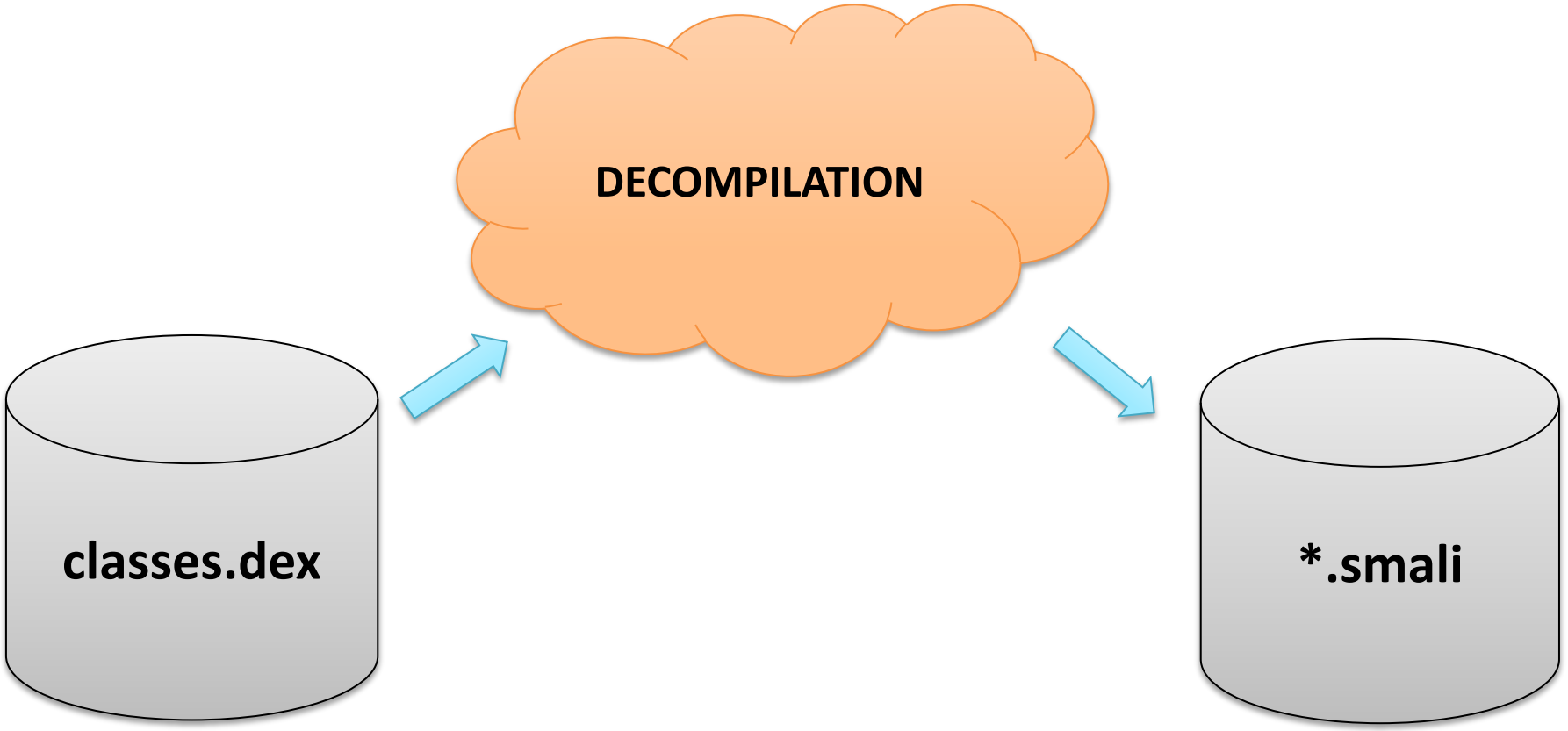


PROFIT

NOT SO FAST...



SMALI



SMALI?

```
private boolean isPassCorrect(String password) {
        //generate random password
        final String correctPass = getRandomString(8);
        //check if user guessed correct
        return correctPass.equals(password);
}
```

```
.method private isPassCorrect(Ljava/lang/String;)Z
    .registers 4
    .param p1, "password"    # Ljava/lang/String;

    .prologue
    .line 39
    const/16 v1, 0x8

    invoke-direct {p0, v1}, Lpl/com/marcing/android/testreverse/MainActivity;-
>getRandomString(I)Ljava/lang/String;

    move-result-object v0

    .line 40
    .local v0, "correctPass":Ljava/lang/String;
    invoke-virtual {v0, p1}, Ljava/lang/String;->equals(Ljava/lang/Object;)Z

    move-result v1

    return v1
.end method
```

```
.method private
isPassCorrect(Ljava/lang/String;)Z
    .registers 4
    .param p1, "password"    # Ljava/lang/String;
```

```
.prologue
    .line 39
    const/16 v1, 0x8

    invoke-direct {p0, v1},
Lpl/com/marcing/android/testreverse/MainActivi
ty;->getRandomString(I)Ljava/lang/String;

    move-result-object v0

    .line 40
    .local v0, "correctPass":Ljava/lang/String;
```

```
invoke-virtual {v0, p1}, Ljava/lang/String;-
>equals(Ljava/lang/Object;)Z

    move-result v1

    return v1
.end method
```

SMALI TYPES

V	void - can only be used for return types
Z	boolean
B	byte
S	short
C	char
I	int
J	long (64 bits)
F	float
D	double (64 bits)

ARRAYS

[[I = int[][]

[Ljava/lang/String; = String[]

EXAMPLE

method(I[[IILjava/lang/String;[Ljava/lang/Object;
)Ljava/lang/String;

EXAMPLE

method(I[[IILjava/lang/String;[Ljava/lang/Object;
)Ljava/lang/String;

String method(int, int[][], int, String, Object[])

PLAYING WITH APK

- MULTIPLE „TO SMALI” DECOMPILERS
  - BAKSMALI
  - APKTOOL
- SOME „TO JAVA” (MAYBE EVEN GOOD)
- SMALI TO CLASSES.DEX
  - SMALI
  - APKTOOL
- APK -> DEX -> SMALI -> DEX -> APK

„PATCHING” APKS

1. GET APK

- DOWNLOAD FROM INTERNET
- GET FROM PHONE
  - MANUAL
  - BACKUP APP

2. DECOMPRESS APK

- USING SOFTWARE LIKE 7ZIP, WINRAR
- APKTOOL WORKS FROM HERE (SKIP 3rd AND 5th STEP)

3. DECOMPILE CLASSES.DEX

- BAKSMALI

java -jar baksmali-*.jar -o classes classes.dex

(*.SMALI FILES WILL APPEAR IN „classes” FOLDER)

4. EDIT *.SMALI FILES

- NOTEPAD, NOTEPAD++

5. PACK CLASSES.DEX

- SMALI

java -jar smali-*.jar classes -o classes.dex

6. CREATE *.APK

- ZIP IT
- APKTOOL

7. SIGNING

keytool -genkey -v -keystore [KEYSTORE NAME]
-alias [ALIAS] -keyalg RSA -keysize 2048 -validity 20000

7. SIGNING

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore [KEYSTORE] [APK FILE] [ALIAS]

7. SIGNING

zipalign -v 4 [PATH TO APK] [OUTPUT FILENAME]

8. PUT BACK TO PHONE/EMULATOR

- UINSTALL „UNPATCHED” APP 1st
- FROM PC: adb install [APK FILE]

EXAMPLE

ANDROID NON INSTALLED ACTIVITY LOADING

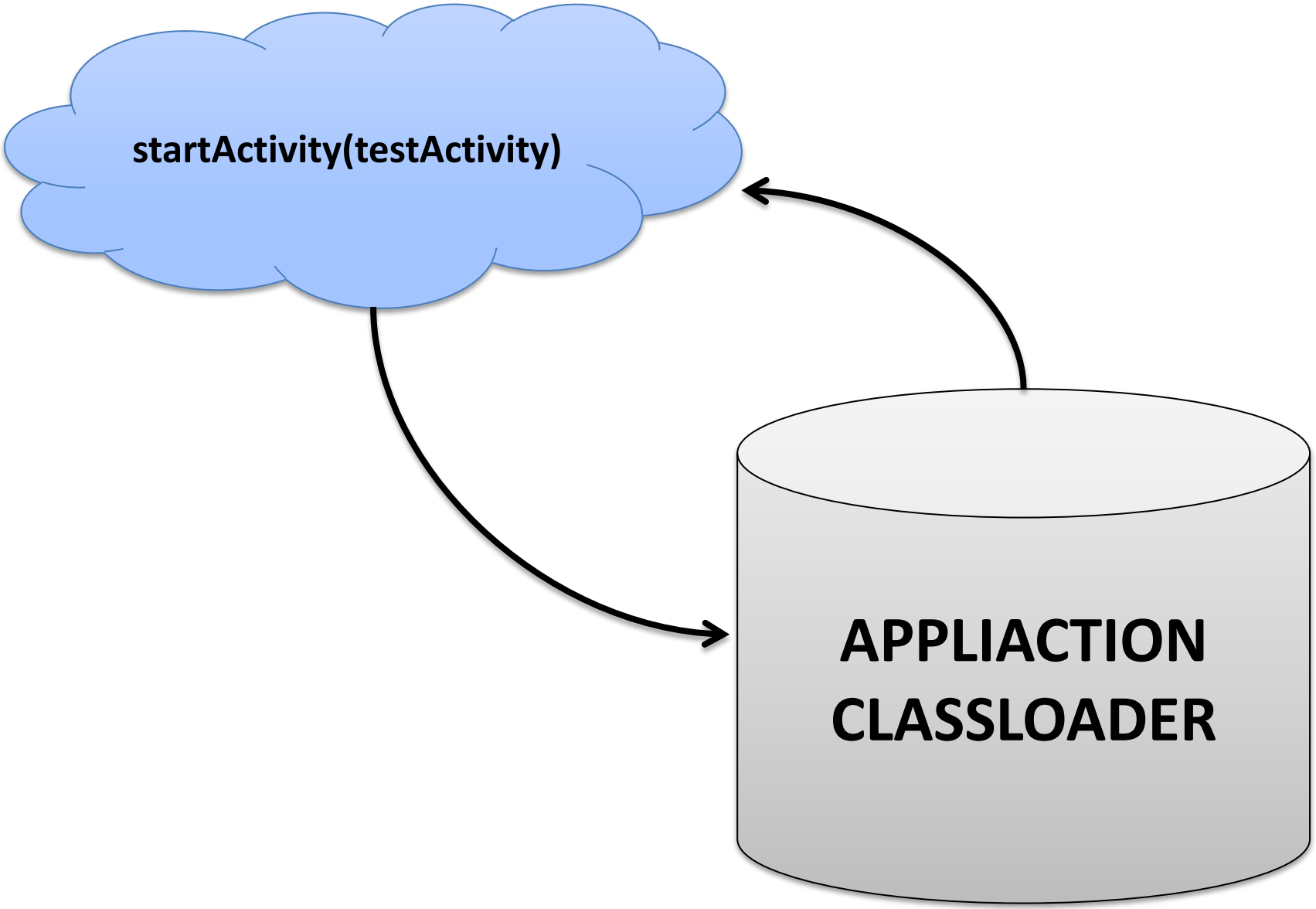
ACTIVITY

- PIECE OF CODE DESIGNED TO DO SOMETHING IN APP
- USUALLY HAS UI
- HAS TO BE DEFINED IN MANIFEST
- MUST BE INSIDE INSTALLED APK IN ORDER TO START IT

ACTIVITY

- PIECE OF CODE DESIGNED TO DO SOMETHING IN APP
- USUALLY HAS UI
- HAS TO BE DEFINED IN MANIFEST
- ~~MUST BE INSIDE INSTALLED APK IN ORDER TO START IT~~

STARTING ACITIVITIES



NONEXISTING ACTIVITY

android.content.ActivityNotFoundException: Unable to find explicit activity class {pl.com.marcing.android.dynamicactivityloader/pl.com.marcing.android.customdex.TestActivity}; **have you declared this activity in your AndroidManifest.xml?**

XML FIXED

FATAL EXCEPTION: main
Process: pl.com.marcing.android.dynamicactivityloader, PID: 14987
java.lang.RuntimeException: Unable to instantiate activity ComponentInfo{pl.com.marcing.android.dynamicactivityloader/pl.com.marcing.android.customdex.TestActivity}: java.lang.ClassNotFoundException: Didn't find class "pl.com.marcing.android.customdex.TestActivity" on path: DexPathList[[zip file "/data/app/pl.com.marcing.android.dynamicactivityloader-2.apk"], nativeLibraryDirectories=[/data/app-lib/pl.com.marcing.android.dynamicactivityloader-2, /system/lib]]

APPLICATION CLASSLOADER

```
27. public class BaseDexClassLoader extends ClassLoader {
28.     /** originally specified path (just used for {@code toString()}) */
29.     private final String originalPath;
30.
31.     /** structured lists of path elements */
32.     private final DexPathList pathList;
33.
34.     /**
35.      * Constructs an instance.
36.      *
37.      * @param dexPath the list of jar/apk files containing classes and
38.      * resources, delimited by {@code File.pathSeparator}, which
39.      * defaults to {@code ":"} on Android
40.      * @param optimizedDirectory directory where optimized dex files
41.      * should be written; may be {@code null}
42.      * @param libraryPath the list of directories containing native
43.      * libraries, delimited by {@code File.pathSeparator}; may be
44.      * {@code null}
45.      * @param parent the parent class loader
46.      */
47.     public BaseDexClassLoader(String dexPath, File optimizedDirectory,
48.             String libraryPath, ClassLoader parent) {
```

MESSING AROUND

- pathList
- private final -> NO ACCESS AND CONSTANT
- Java Reflection API
- CUSTOM CLASSLOADER WITH NEW APK
- TAKE **pathList** FROM CUSTOM CLASSLOADER AND PUT INSIDE APPLICATION CLASSLOADER

CLASSLOADER FIXED

02-20 11:51:46.742 15359-15359/pl.com.marcing.android.dynamicactivityloader I/DynamicActivityLoadActivity： Test: dalvik.system.DexClassLoader[DexPathList[[zip file "/data/data/pl.com.marcing.android.dynamicactivityloader/app_dex/test.apk", zip file "/data/app/pl.com.marcing.android.dynamicactivityloader-1.apk"],nativeLibraryDirectories=[/data/app-lib/pl.com.marcing.android.dynamicactivityloader-1, /system/lib, /system/lib]]]
02-20 11:51:46.750 15359-15359/pl.com.marcing.android.dynamicactivityloader I/TestActivity：::onStart
02-20 11:51:46.750 15359-15359/pl.com.marcing.android.dynamicactivityloader I/TestActivity：action inside TestActivity

FACTS

- NEW ACTIVITY RUNS INSIDE OLD CONTEXT
  - PERMISSIONS
  - RESOURCES
- NEW ACTIVITY MUST BE DEFINED INSIDE MANIFEST
- „PATCHING CODE” CAN BE INSIDE NEW APK (INSTALLED APK NEEDS LOADER).

POSSIBILITIES

- HIDDEN FEATURES…
- WITH FEW MODS OF SMALI AND MANIFEST ONE CAN INJECT NEW CODE TO APP (ACCESS TO RUNNING APP MEMORY WITHOUT ROOTING DEVICE)

QUESTIONS?