DEXCALIBUR

Hook it yourself !

Geoges-B. MICHEL - SSTIC 2020

GEORGES-B. MICHEL aka @FrenchYeti

- georges@dexcalibur.org
- Working as Reverse Engineer / Software Security evaluator
- Author of Dexcalibur
- Passionate by reverse & development
- Frida addict

(Copyright GTO, feat. Onizuka)

WHAT IS DEXCALIBUR ?

- Dynamic Application Security Testing (DAST) tool
- Free and Open-Source Android RE platform
- Extensible, comprehensive, multi-user, GUI
- Personal project developed on spare time started 2018
- Few contributors (4)

HOW IS DEXCALIBUR DIFFERENT ?

Be aware of a maximum of information about device and application at rest or at runtime. Be autonomous.

- Deep integration of various reverse engineering technics
- New built-in bytecode static analyzer (no smalisca, no androguard)
- Custom Smali VM closer to DVM behavior
- Update static analysis (corpus) with instrumentation result
- Generate instrumentation thanks to static analysis of bytecode
- Designed to be able to analyze obfuscated APK with dynamic loading
- Usable with vendor applications depending of undocumented OS feature
- Multi-user
- More ...

STATIC OBFUSCATION, REAL LIFE

- Is "java.lang.StringSplitter"
  (for example) an internal class ?

STATIC OBFUSCATION, REAL LIFE

- Is "java.lang.StringSplitter"
  (for example) an internal class ?

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/[redacted]/crackme/act

const/4 v0, 0x1

move-result-object v0

goto/32 :goto_3

:goto_2
const-string v0, "3780b8133459f5a028d742efbcfc7d2d"

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/[redacted]/crackme/activities/Crac
```

WHAT HAPPENS ?

STATIC OBFUSCATION, REAL LIFE

- Is "java.lang.StringSplitter"
  (for example) an internal class ?

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/

const/4 v0, 0x1

move-result-object v0

goto/32 :goto_3

:goto_2
const-string v0, "3780b8133459f5a028

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/[redacted]/crac
```

WH

static	QSLrjaCtdwZMXxl(<java.lang.String><int>)<char>
static	RugleAJCMcOYiED(<java.lang.String>)<int>
static	UAozYOTEbCdtGyS(<java.lang.String><int>)<char>
static	VqxuLoJHbjvgTRW(<java.lang.String>)<javax.crypto.SecretKeyFactory>
static	XjYcgdUCozPhGQw(<java.lang.StringBuilder><char>)<java.lang.StringBuilder>
static	XvyipIDZgVArked(<java.lang.String>)<byte>[]
static	YIXqJEOvWdtLHcU(<java.lang.String>)<int>
private static	a03674e6c(<java.lang.String>)<java.lang.String>
static	bVjSqgAehlQJpnT(<java.lang.String>)<int>
static	bXEVATfiSCtzpkh(<java.lang.String>)<javax.crypto.Cipher>
static	bwXLNAdkapcUmDB(<java.lang.StringBuilder>)<java.lang.String>
static	convertToString(<java.lang.String>)<java.lang.String>
static	dTxzoBgvOZreERy(<java.lang.StringBuilder><char>)<java.lang.StringBuilder>
static	deDnvrZgyKlqtmM(<java.lang.String><int>)<char>
private static	decode(<java.lang.String>)<byte>[]
private static	getStorageEncryption(<int><java.lang.String>)<javax.crypto.Cipher>
static	igDHotavmxKEbdh(<java.lang.String>)<int>
static	ihxCrtGAOVjZsgR(<java.lang.String><java.lang.String>)<byte>[]
static	klbUpHVXTuWmenL(<java.lang.StringBuilder>)<java.lang.String>
static	ldKjpFnmfkBtvxi(<java.lang.String>)<int>
static	lwLuoCTSFzaYIVf(<java.lang.String><int>)<char>
static	mlhOSnrEbkoFsXa(<java.lang.StringBuilder><char>)<java.lang.StringBuilder>
static	nKPaBlvidtuJCyq(<java.lang.StringBuilder><char>)<java.lang.StringBuilder>
static	orteEvxHPWFaXNB(<java.lang.String><int>)<char>
static	qUjcZulrtsfMTIH(<java.lang.String><int>)<char>
static	qoYcZGfwieahVQd(<java.lang.Object>)<java.lang.String>
static	rxmCKugynlUZGEh(<java.lang.String><int>)<char>
static	sJKaZhFGlCTUuwz(<java.lang.String>)<int>

STATIC OBFUSCATION, REAL LIFE

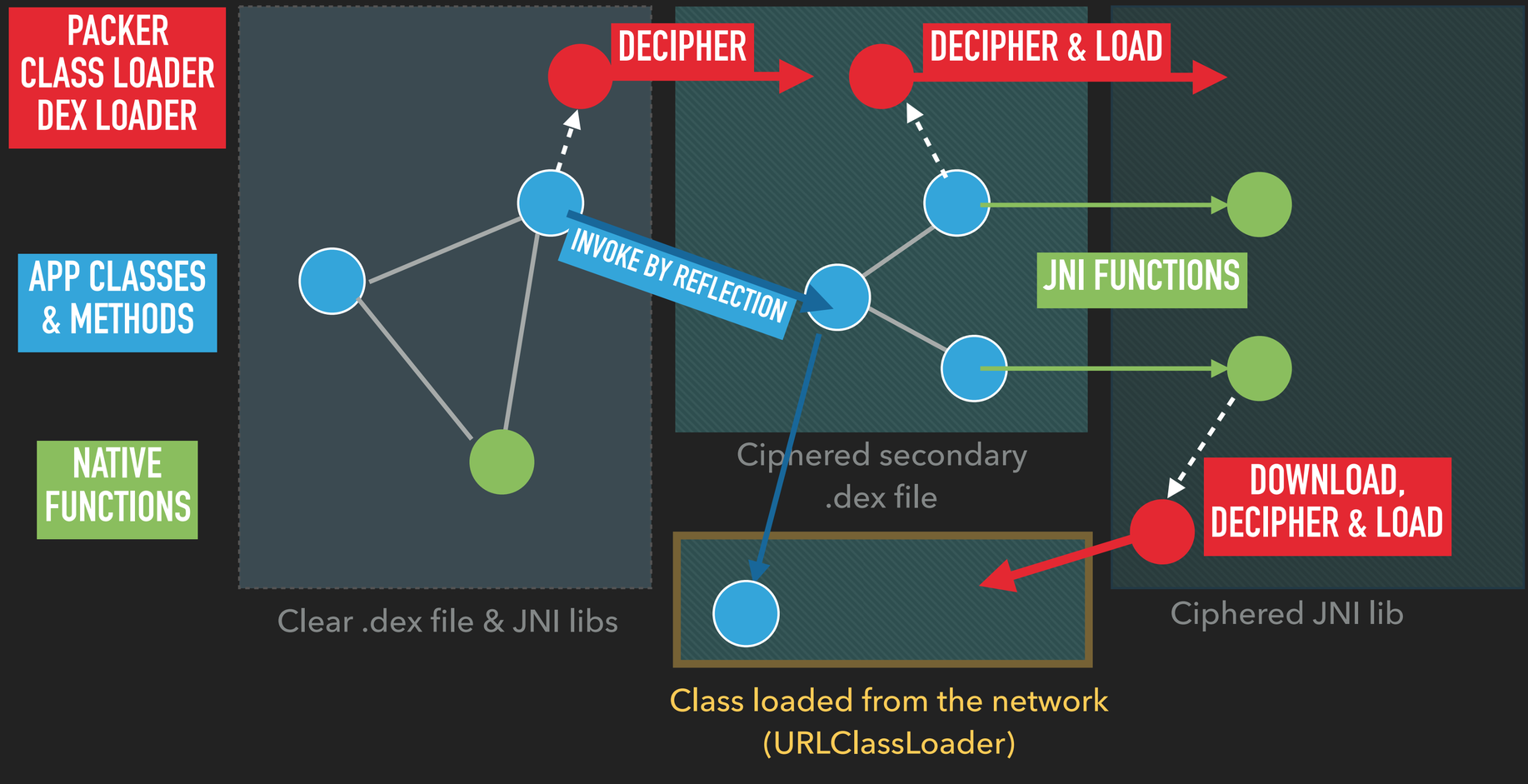
```
goto/32 :goto_7b


:goto_0
goto/32 :goto_75


:goto_1
goto/16 :goto_27


:sswitch_0
.line 112
goto/32 :goto_79


:goto_2
invoke-static {v4, v0},
goto/32 :goto_33


:goto_3
if-ge v3, v8, :cond_0

goto/32 :goto_26


:cond_0
goto/32 :goto_5c


:goto_4
invoke-static {p0, v6},
move-result v6
goto/32 :goto_77
```

```
:goto_5
const/16 v0, 0xd
.line 101
goto/32 :goto_23


:goto_6
if-lt v2, v6, :cond_1

goto/32 :goto_38


:cond_1
goto/32 :goto_3f


:goto_7
invoke-static {v6, v3}, Landroid/content/res/abltMZGC;->XjYcgdUCozPhGQw(Lj
    move-result-object v6
goto/32 :goto_53


:goto_8
if-lt v2, v6, :cond_2

goto/32 :goto_38


:cond_2
goto/32 :goto_20


:goto_9
add-int/lit8 v6, v2, 0x1
goto/32 :goto_5b
```

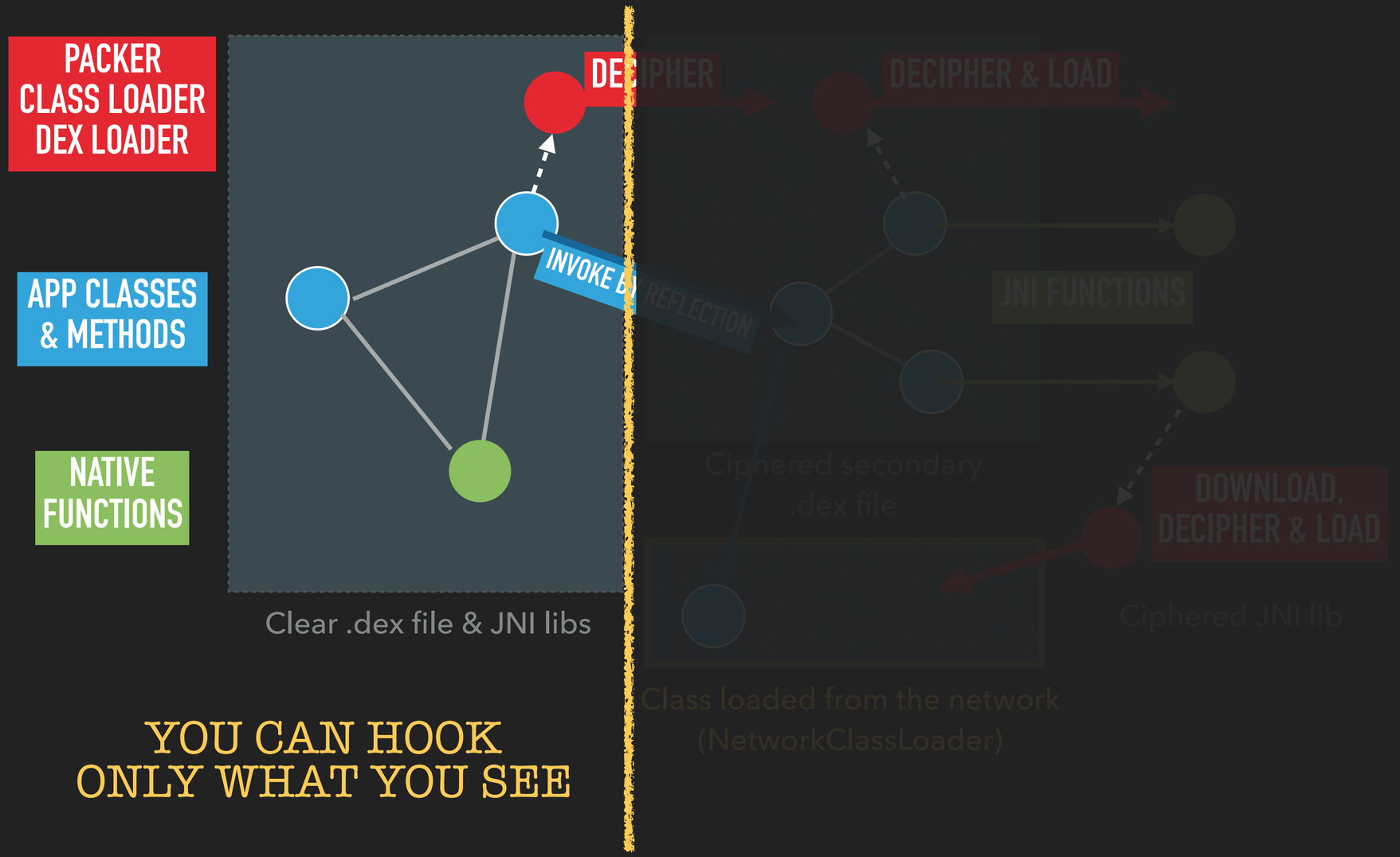
ETC ...

... ~ 124 goto(s)

DYNAMIC (MULTI-LEVELS) OBFUSCATION, WHAT CAN I HOOK ?

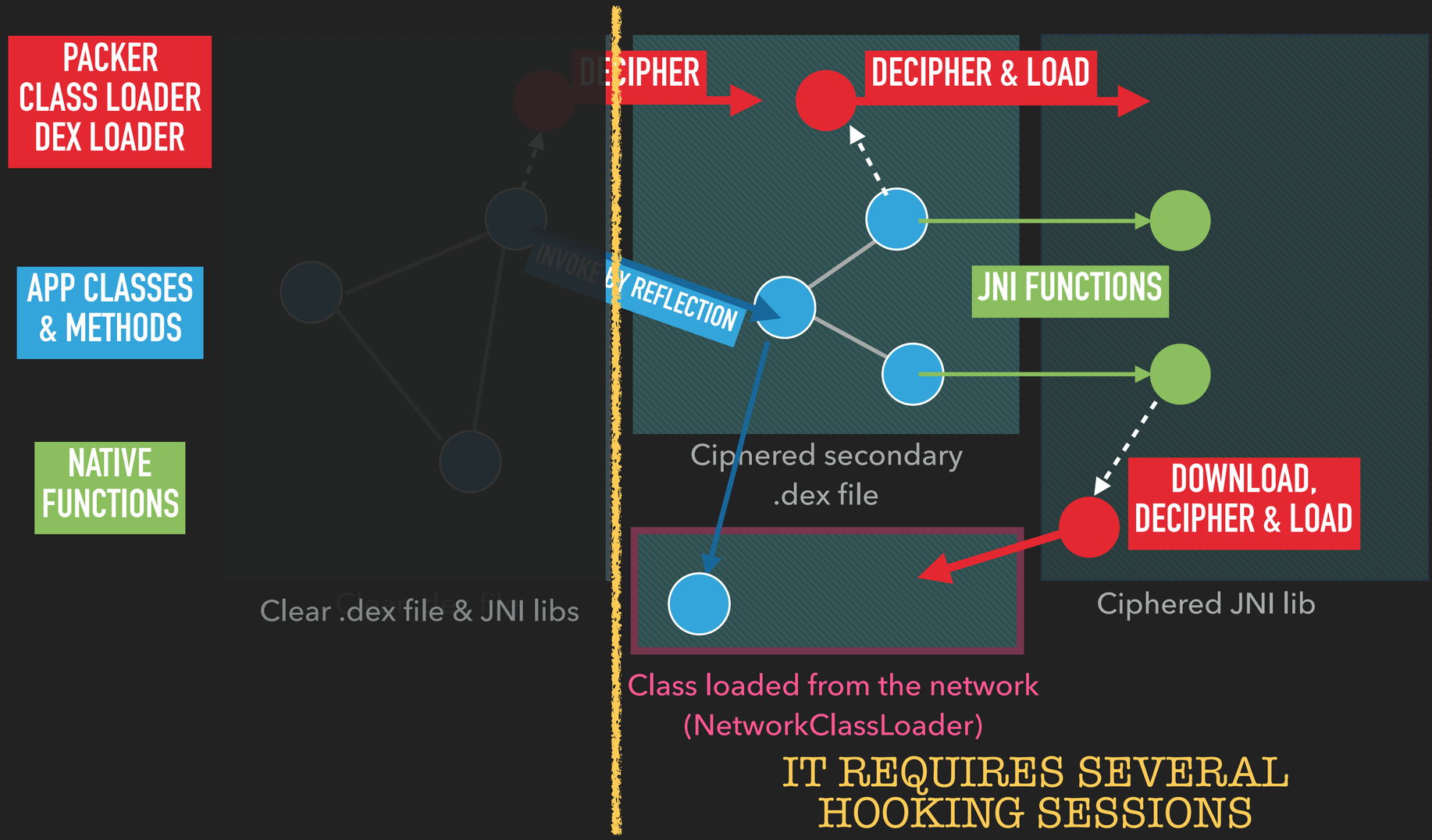


DYNAMIC (MULTI-LEVELS) OBFUSCATION, WHAT CAN I HOOK ?



YOU CAN HOOK
ONLY WHAT YOU SEE

DYNAMIC (MULTI-LEVELS) OBFUSCATION, WHAT CAN I HOOK ?



IT REQUIRES SEVERAL
HOOKING SESSIONS

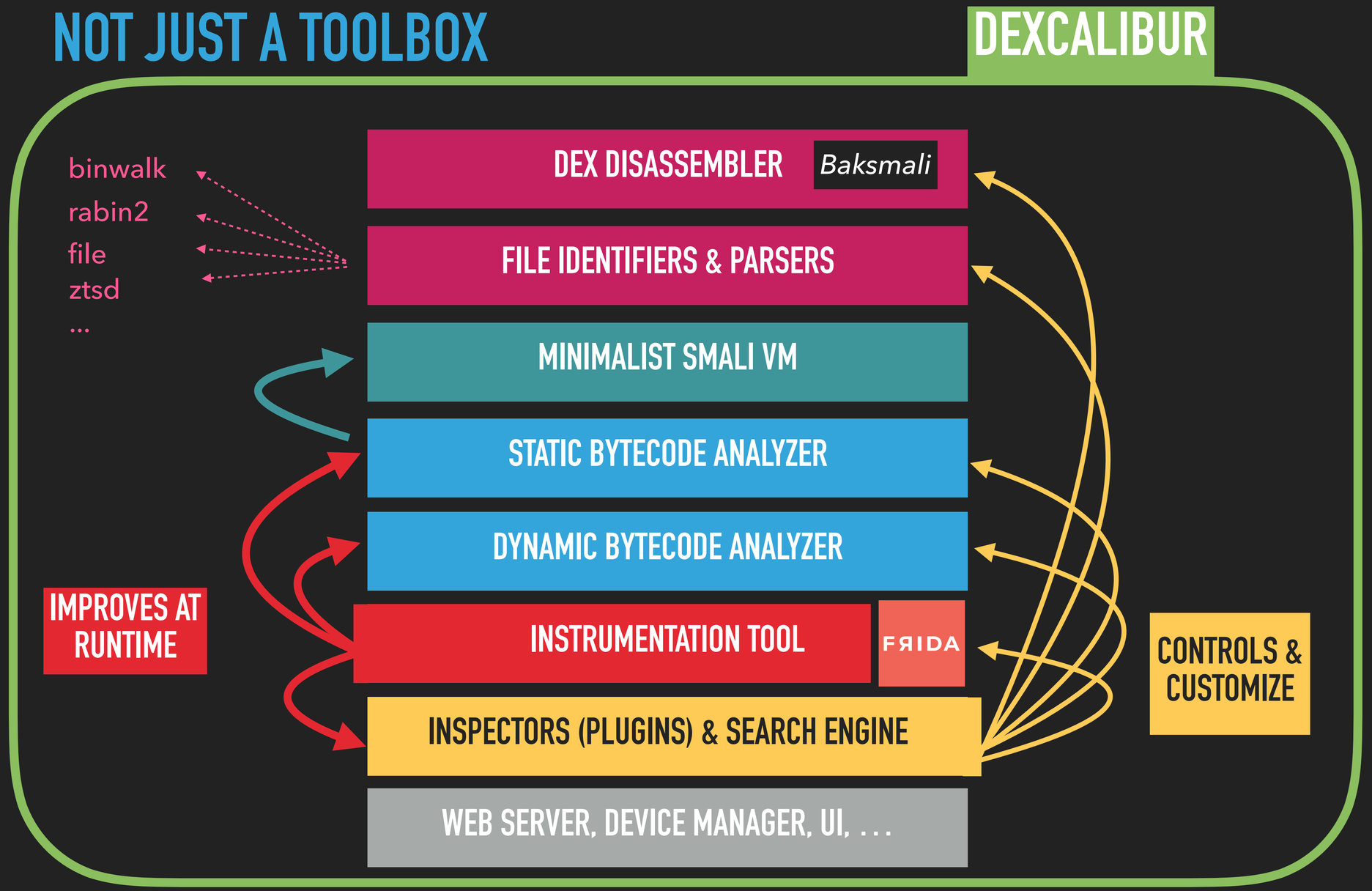
FINALLY, THE "PERFORMANCE PROBLEM"

- Common app
  - + 42 000 classes
  - + 250 000 methods
  - + 650 000 calls
  - + 3M instructions

(Copyright GTO, feat. Onizuka)

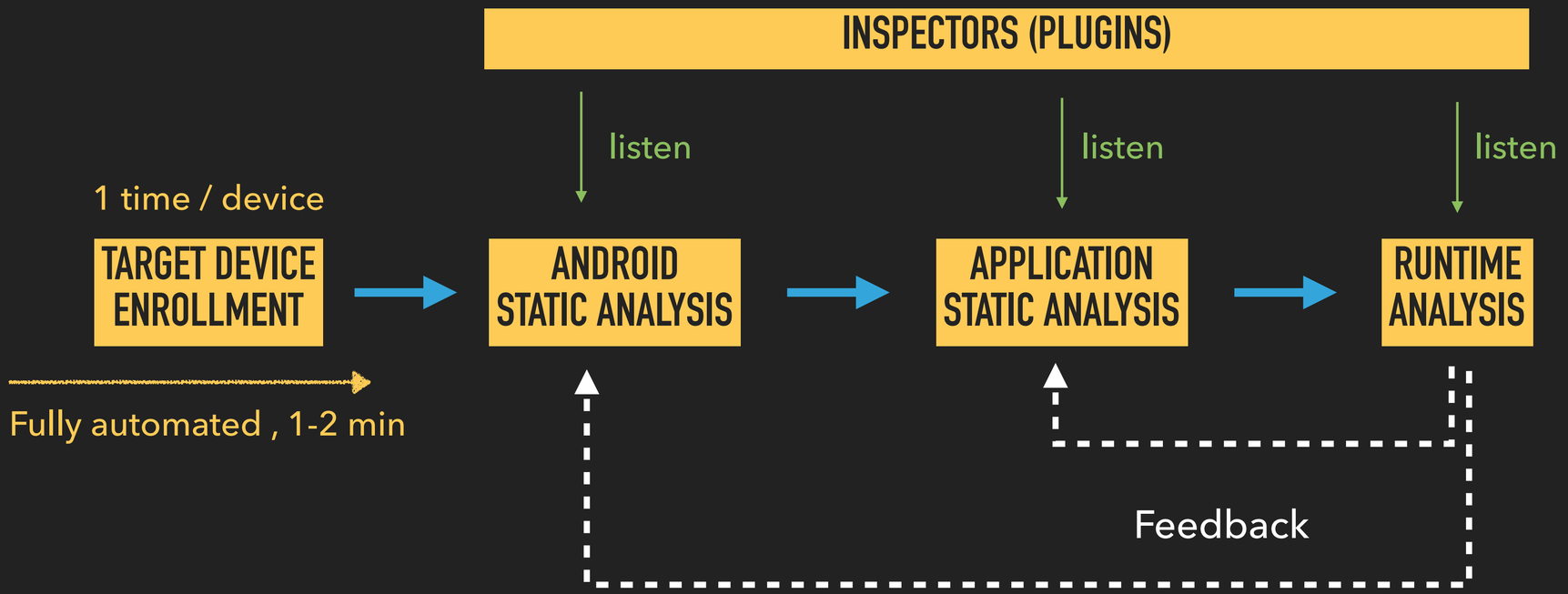
WHAT IS DEXCALIBUR ?

NOT JUST A TOOLBOX

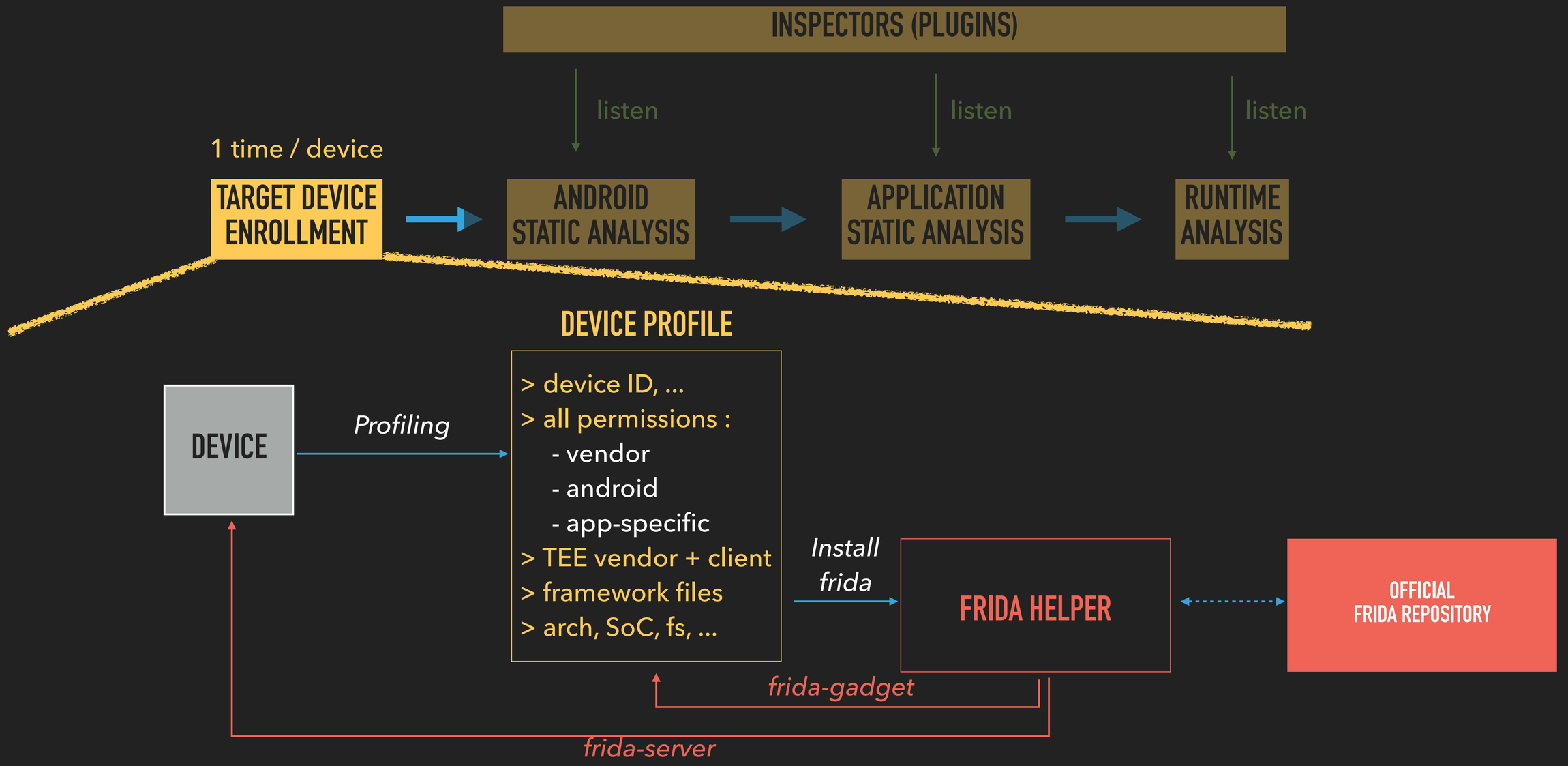


HOW IT WORKS ?

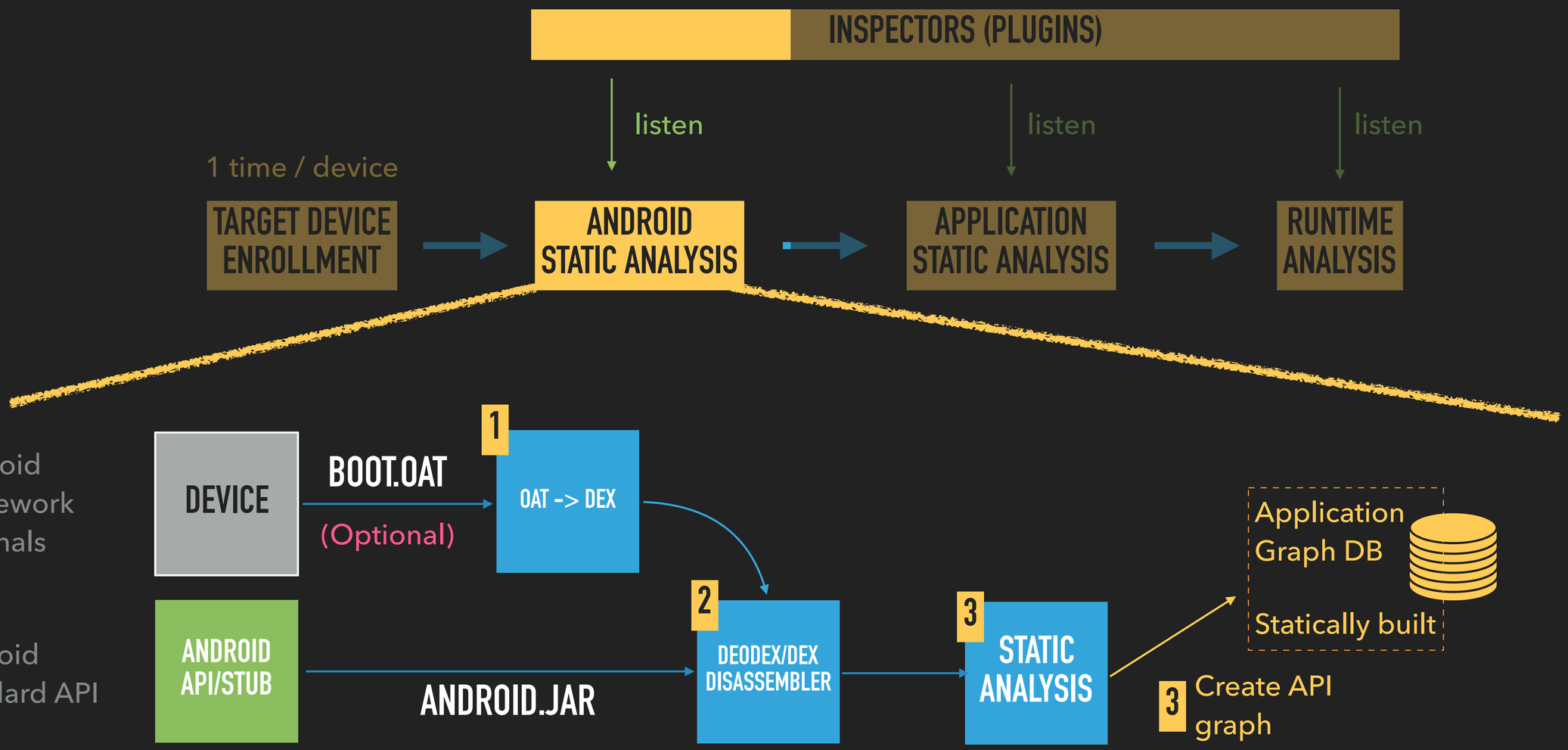
HOW IT WORKS ?


INSPECTORS (PLUGINS)
listen
listen
listen
1 time / device
TARGET DEVICE ENROLLMENT
ANDROID STATIC ANALYSIS
APPLICATION STATIC ANALYSIS
RUNTIME ANALYSIS
Fully automated , 1-2 min
Feedback

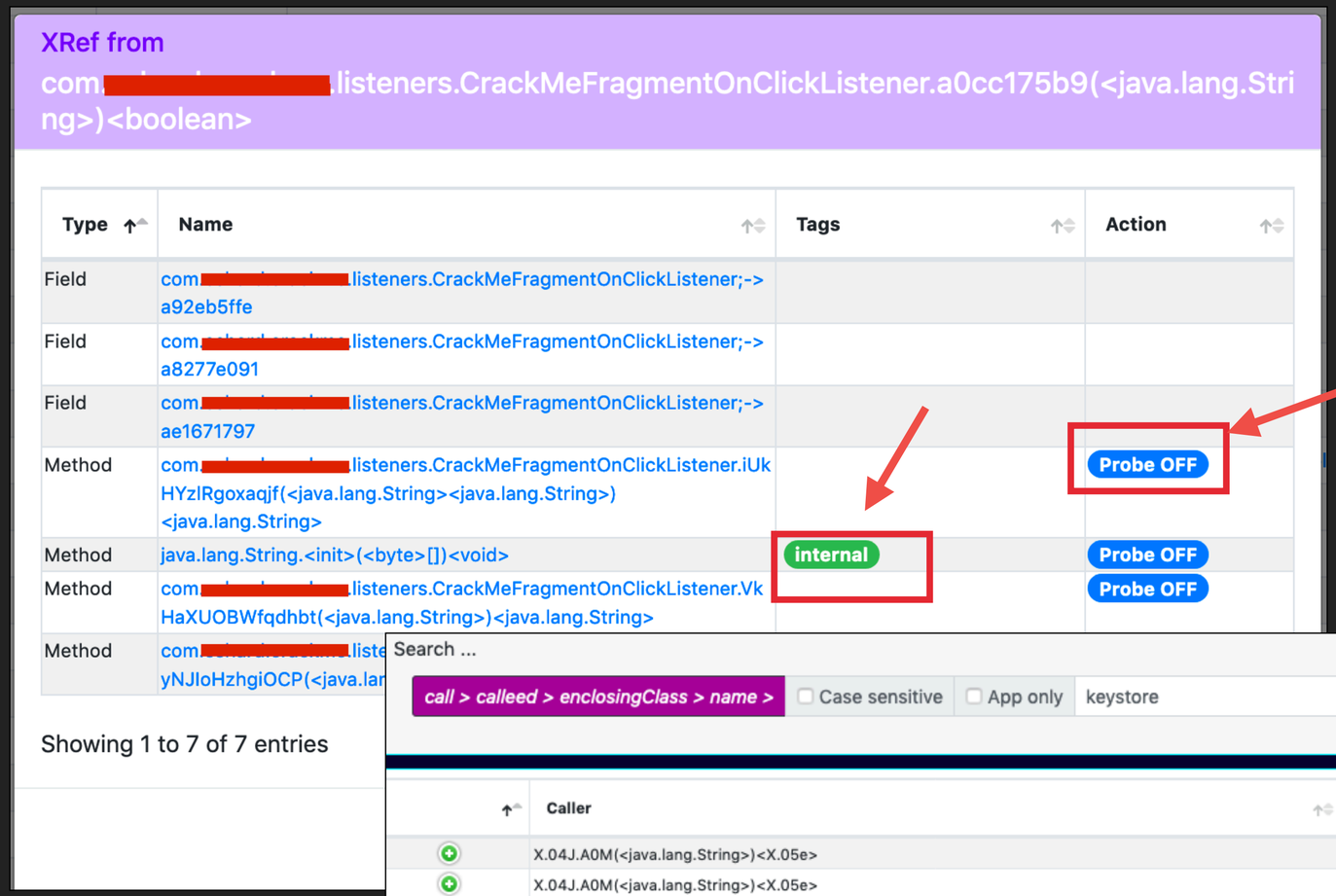
HOW IT WORKS ? - ENROLLMENT PHASE



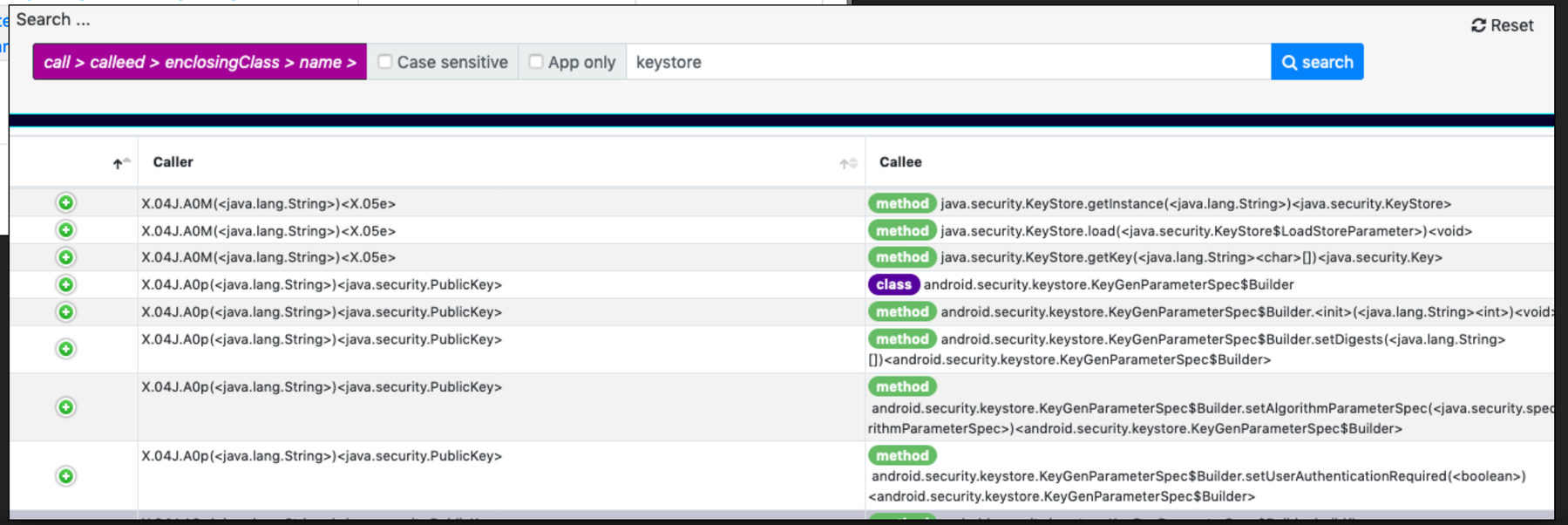
HOW IT WORKS ? - ANDROID FRAMEWORK ANALYSIS PHASE



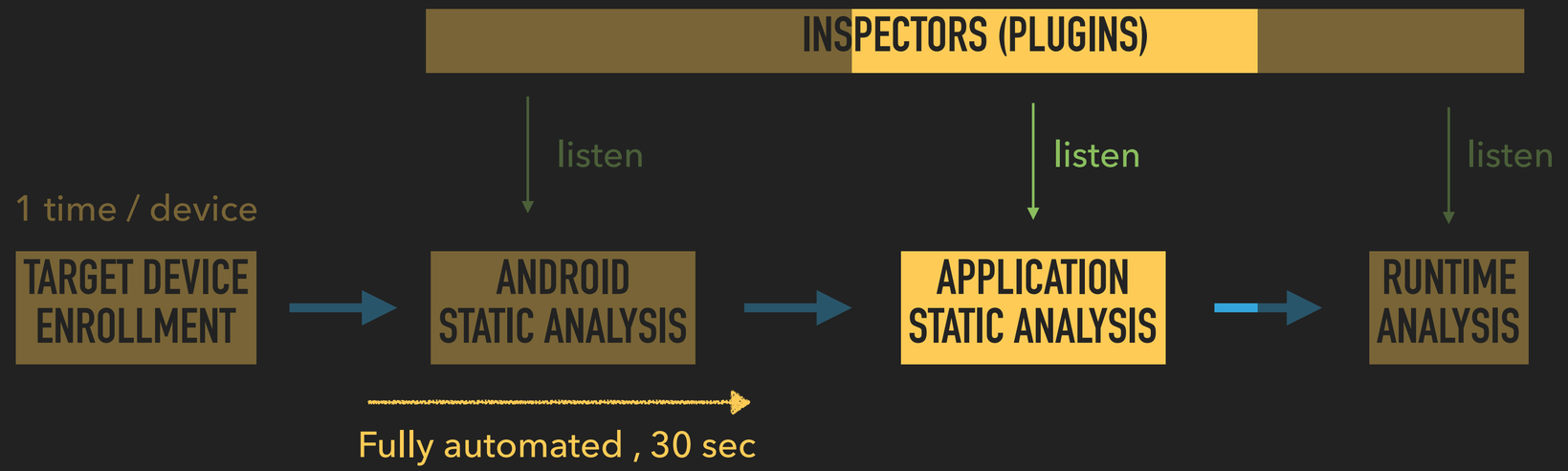
HOW IT WORKS ? - ANDROID FRAMEWORK ANALYSIS PHASE



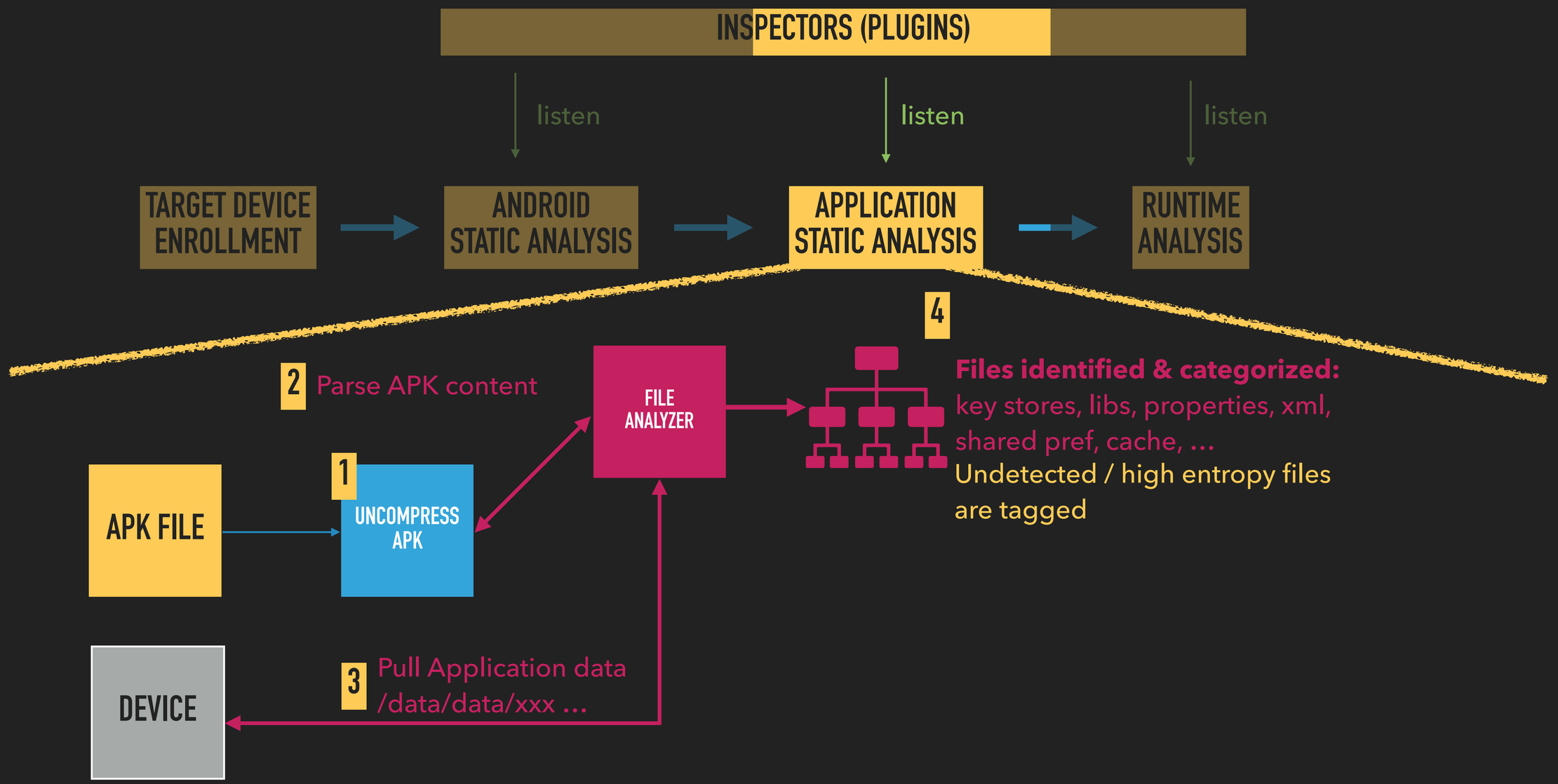
click to generate + deploy a hook



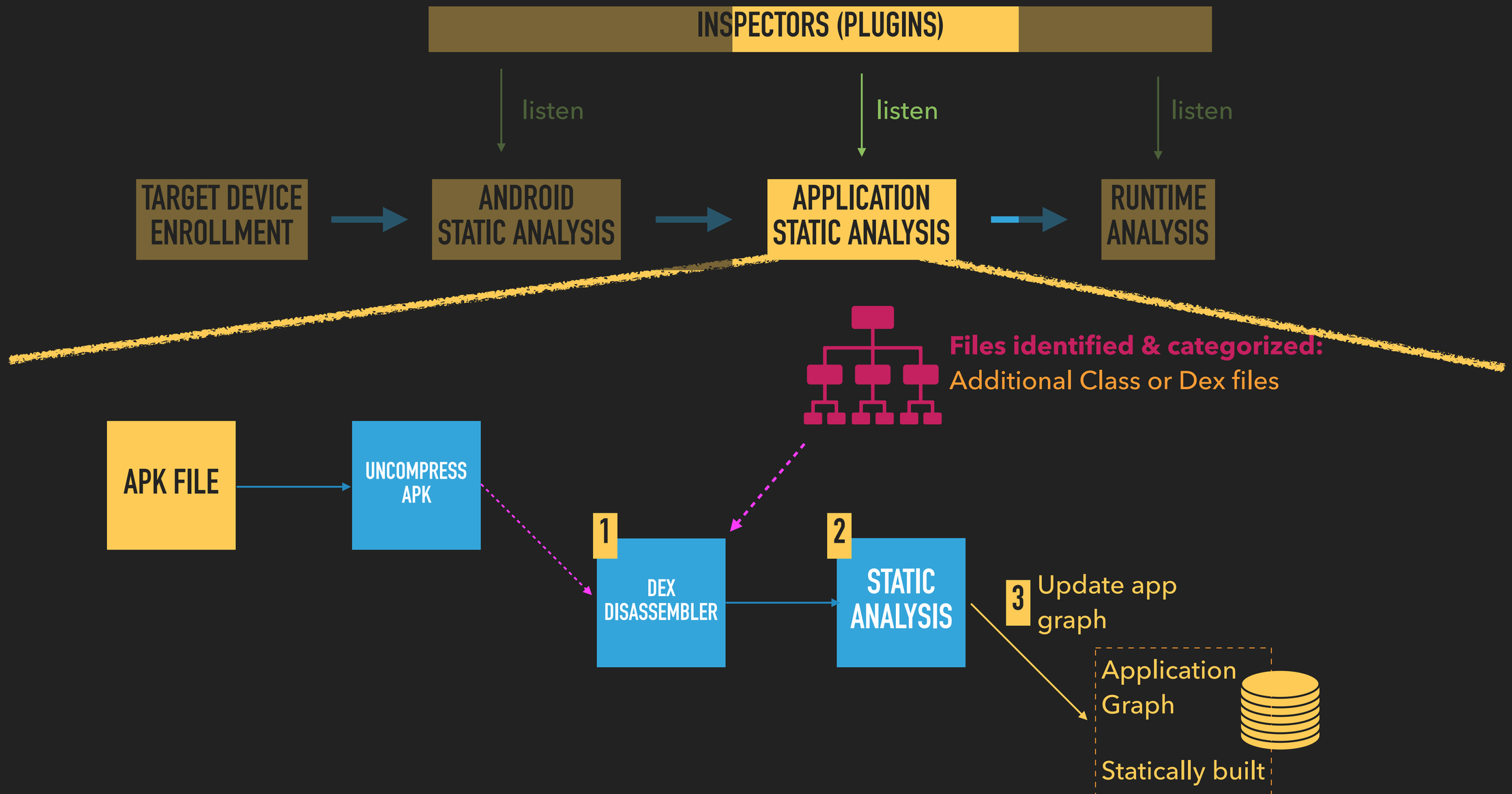
HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE



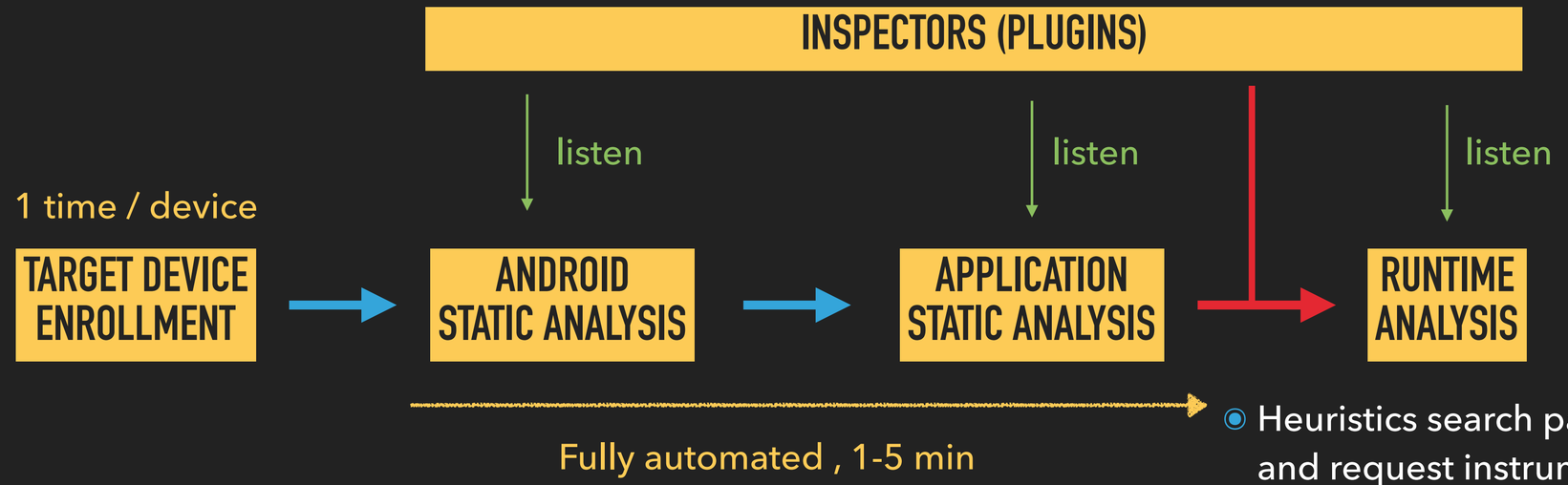
HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE



HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE

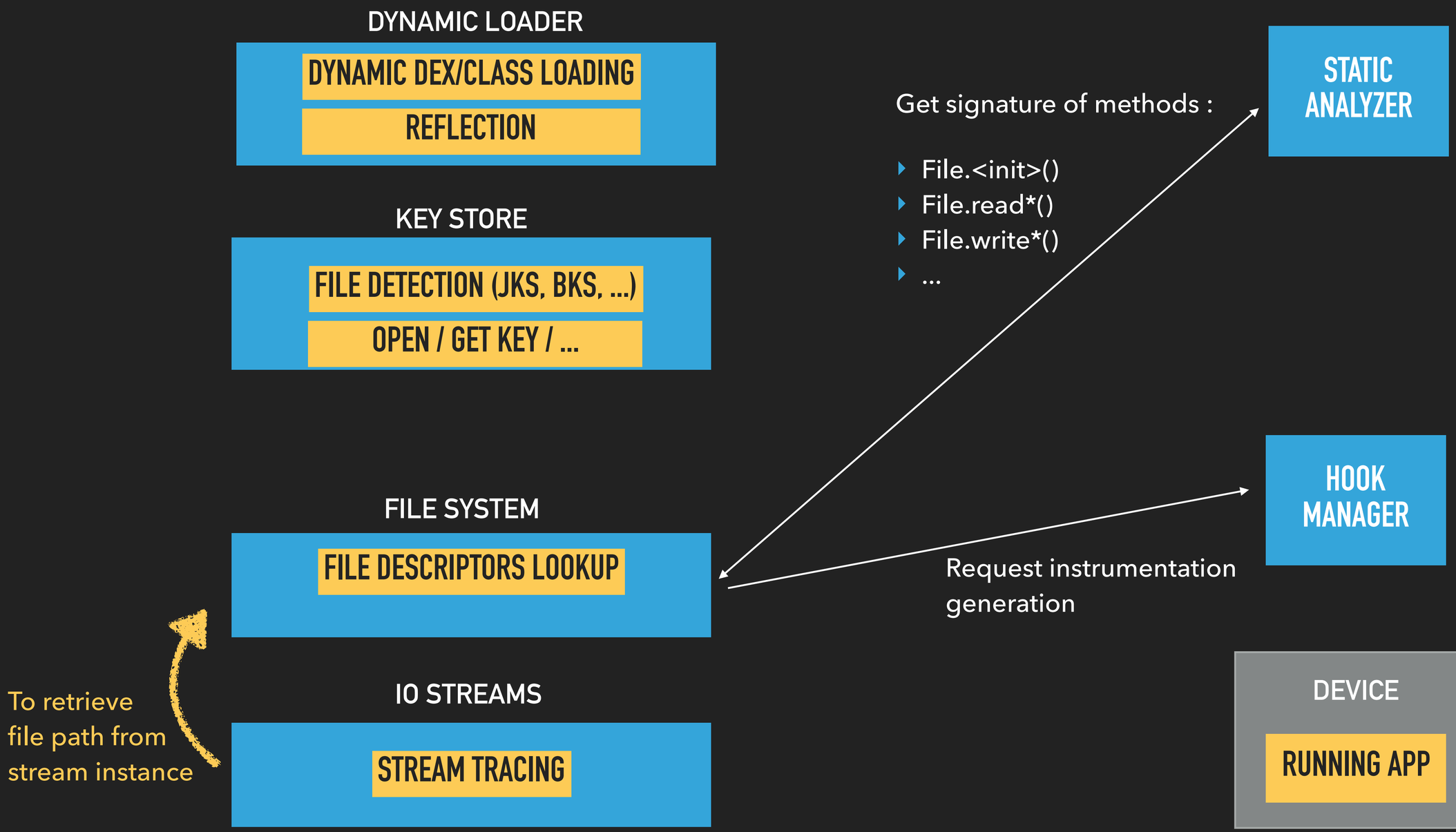


HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

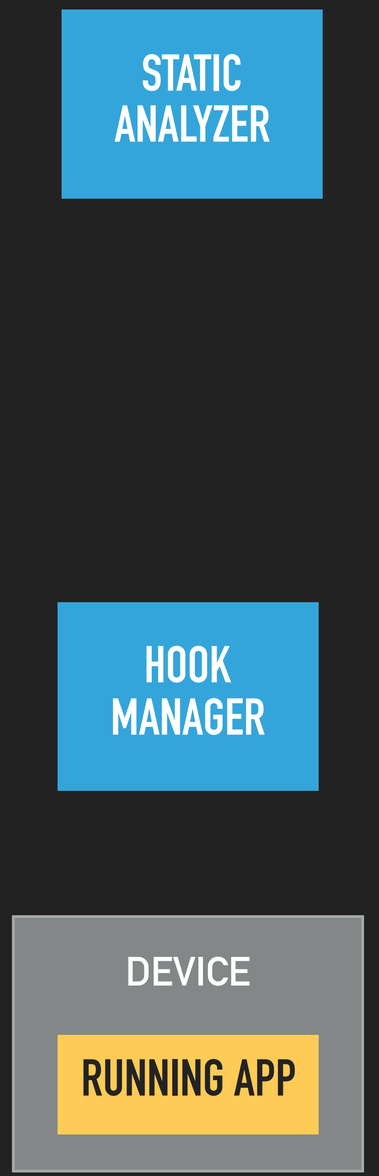


- Heuristics search patterns and request instrumentation
- HookManager generate script source code of the hook ...
- .. and spawn/attach to application
- Intercepted data send back from device to Dexcalibur
- Data gathered broadcasted to plugins

HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

INSPECTOR "DYNAMIC LOADER"

Hook Listeners :
- "InMemoryDexClassLoader.load()"
- "defineClass" from classes implementing "ClassLoader"

STATIC ANALYZER

HOOK MANAGER

DEVICE

RUNNING APP

HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

INSPECTOR "DYNAMIC LOADER"

Hook Listeners :
- "InMemoryDexClassLoader.load()"
- "defineClass" from classes implementing "ClassLoader"

STATIC ANALYZER

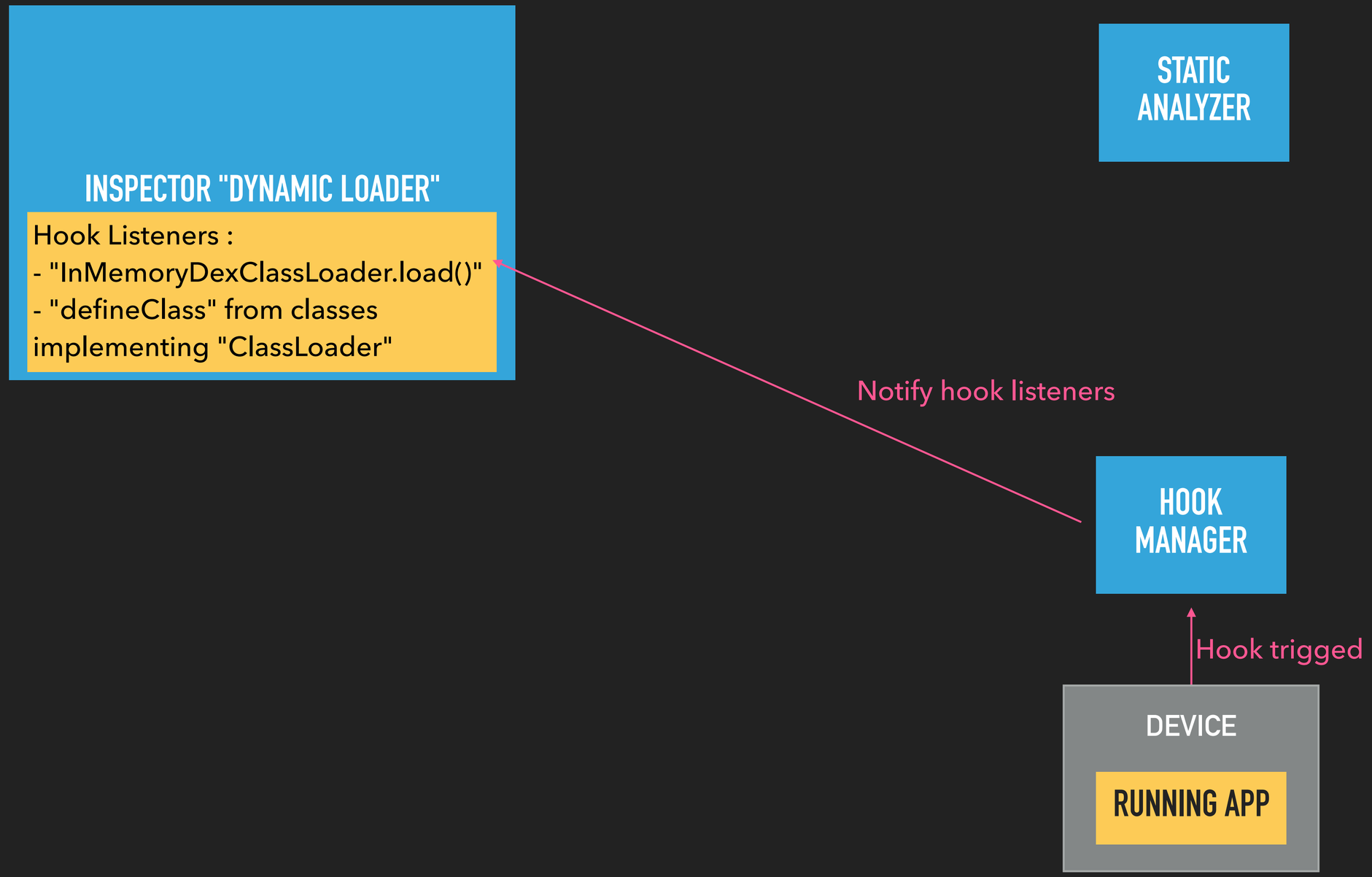
HOOK MANAGER

Hook trigged

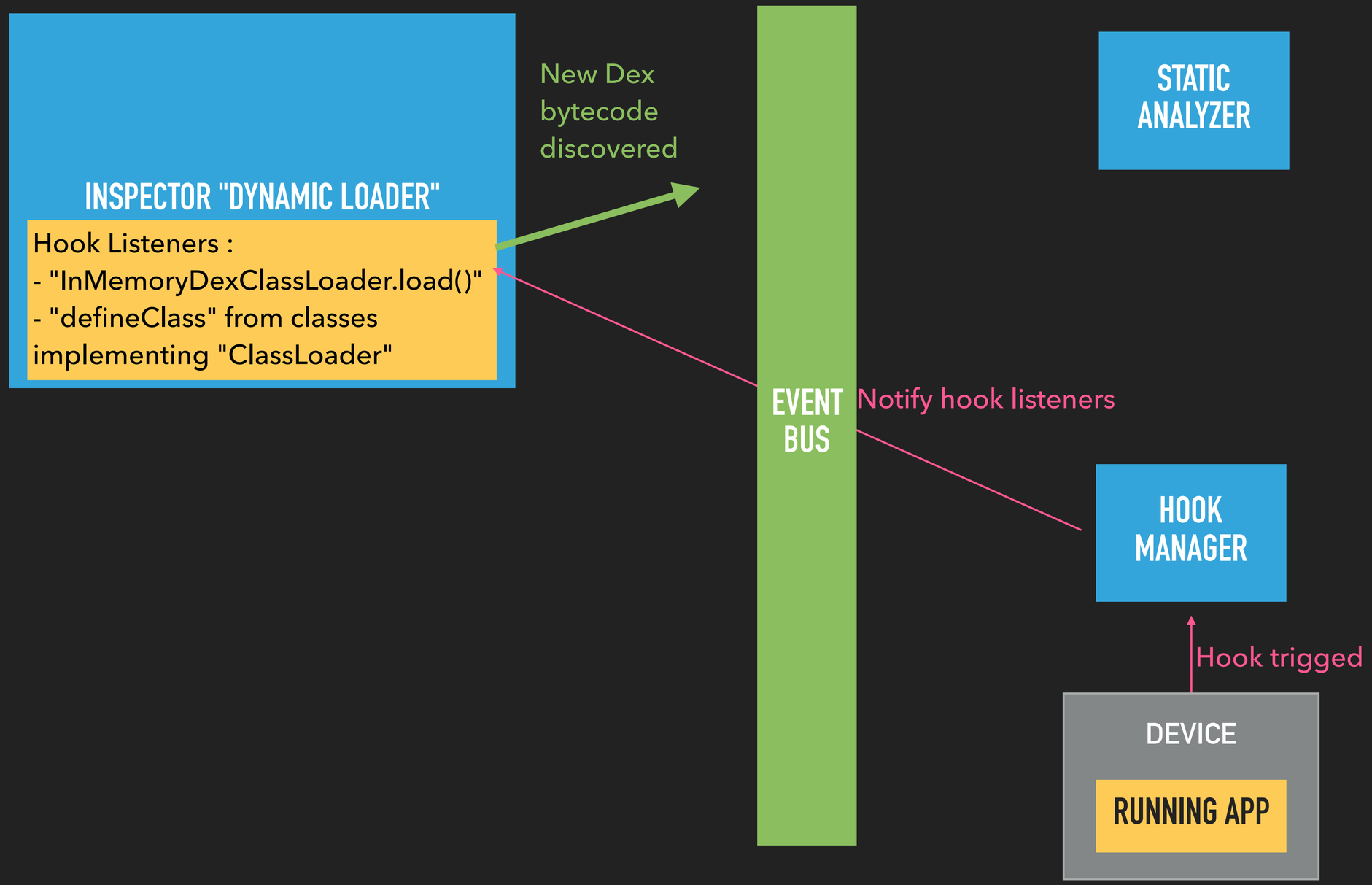
DEVICE

RUNNING APP

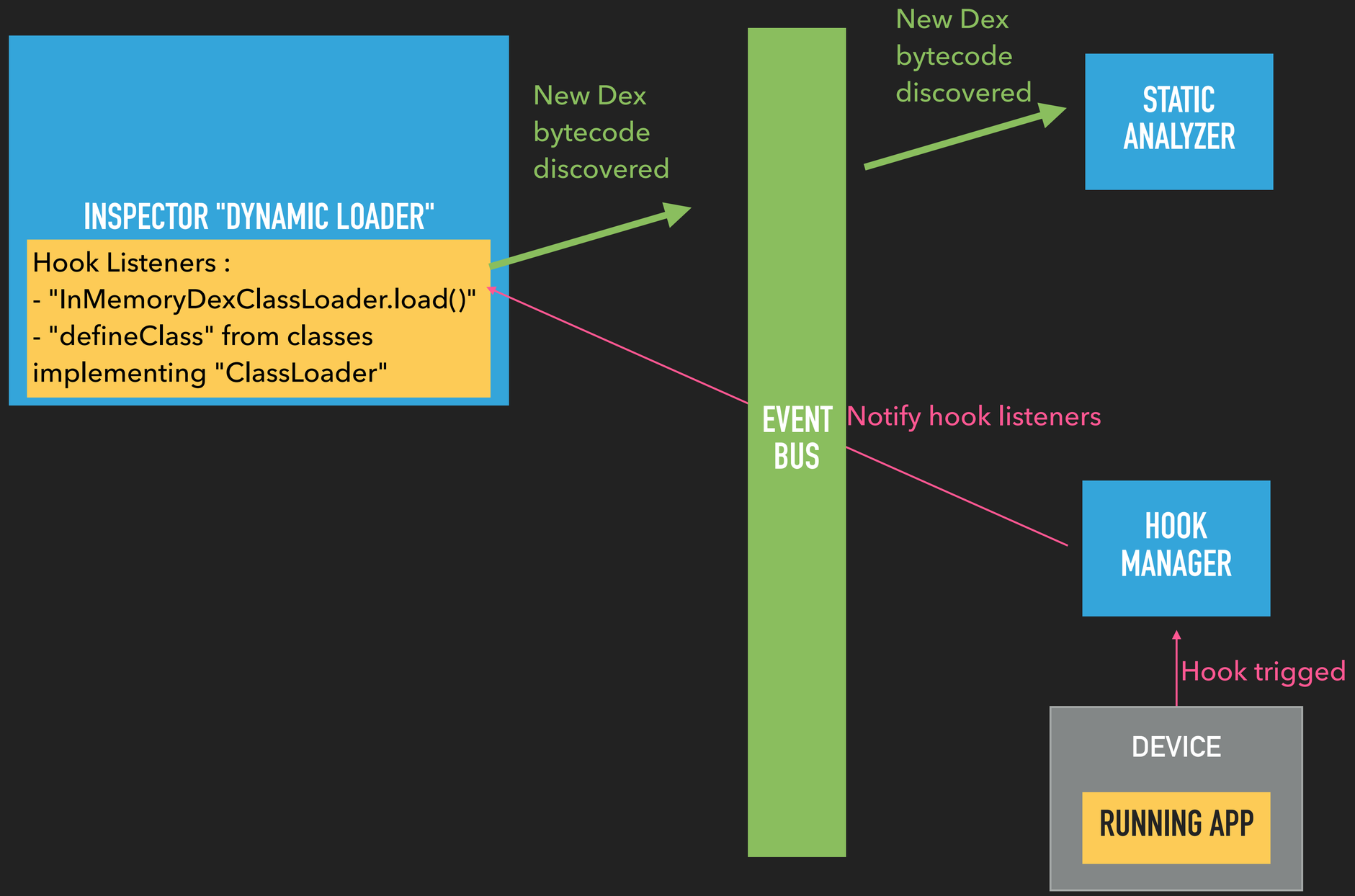
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



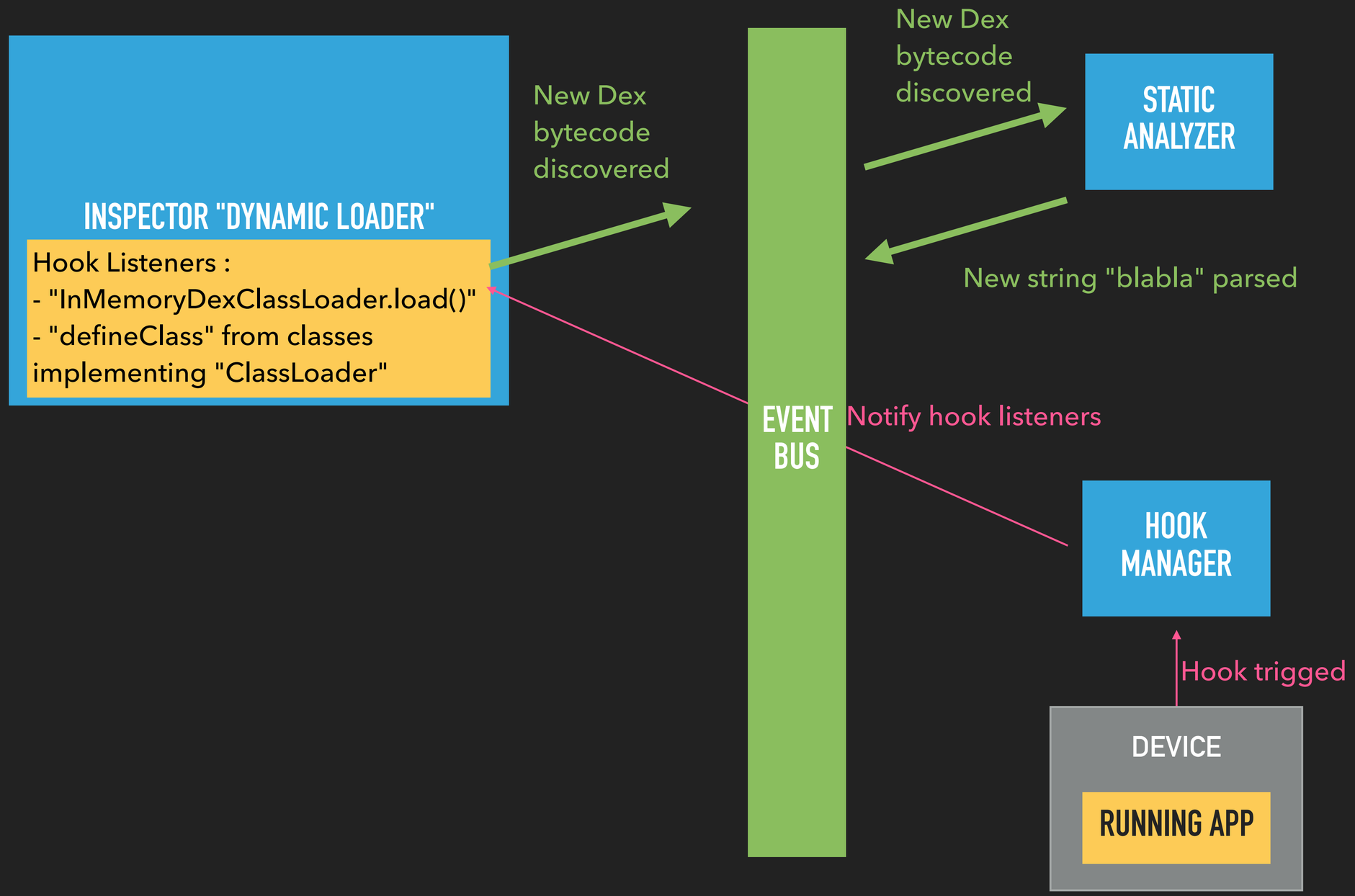
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



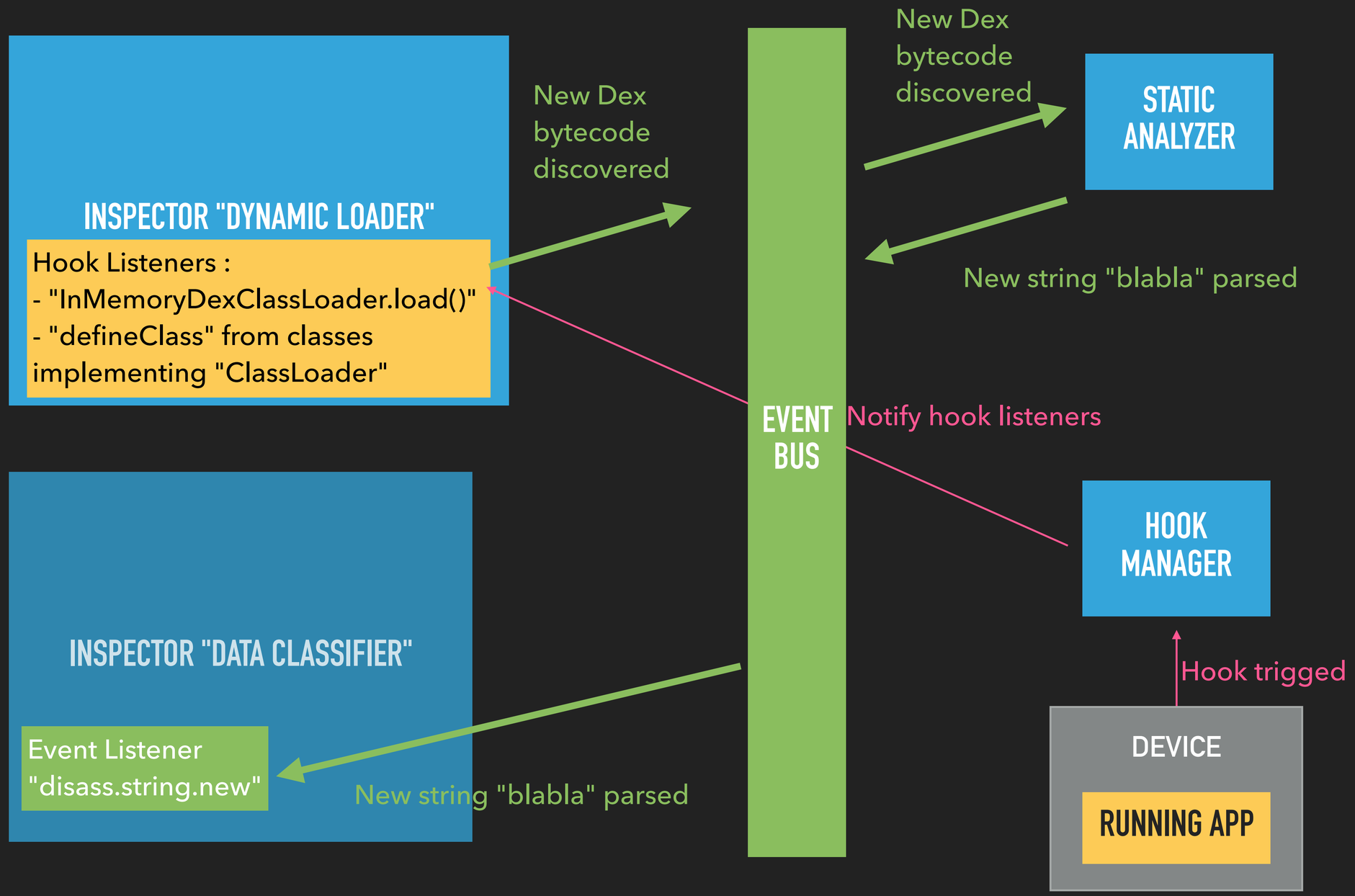
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



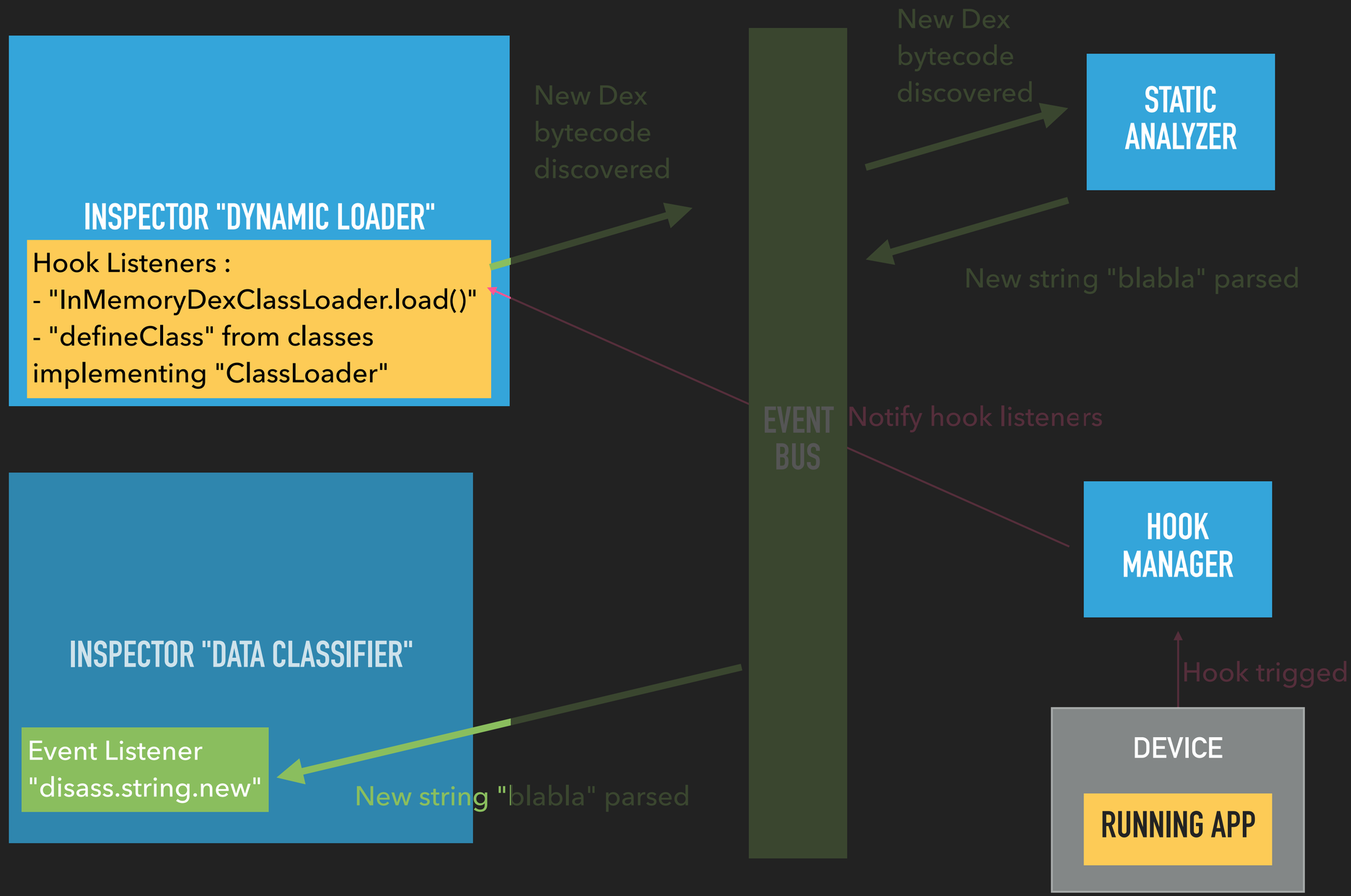
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



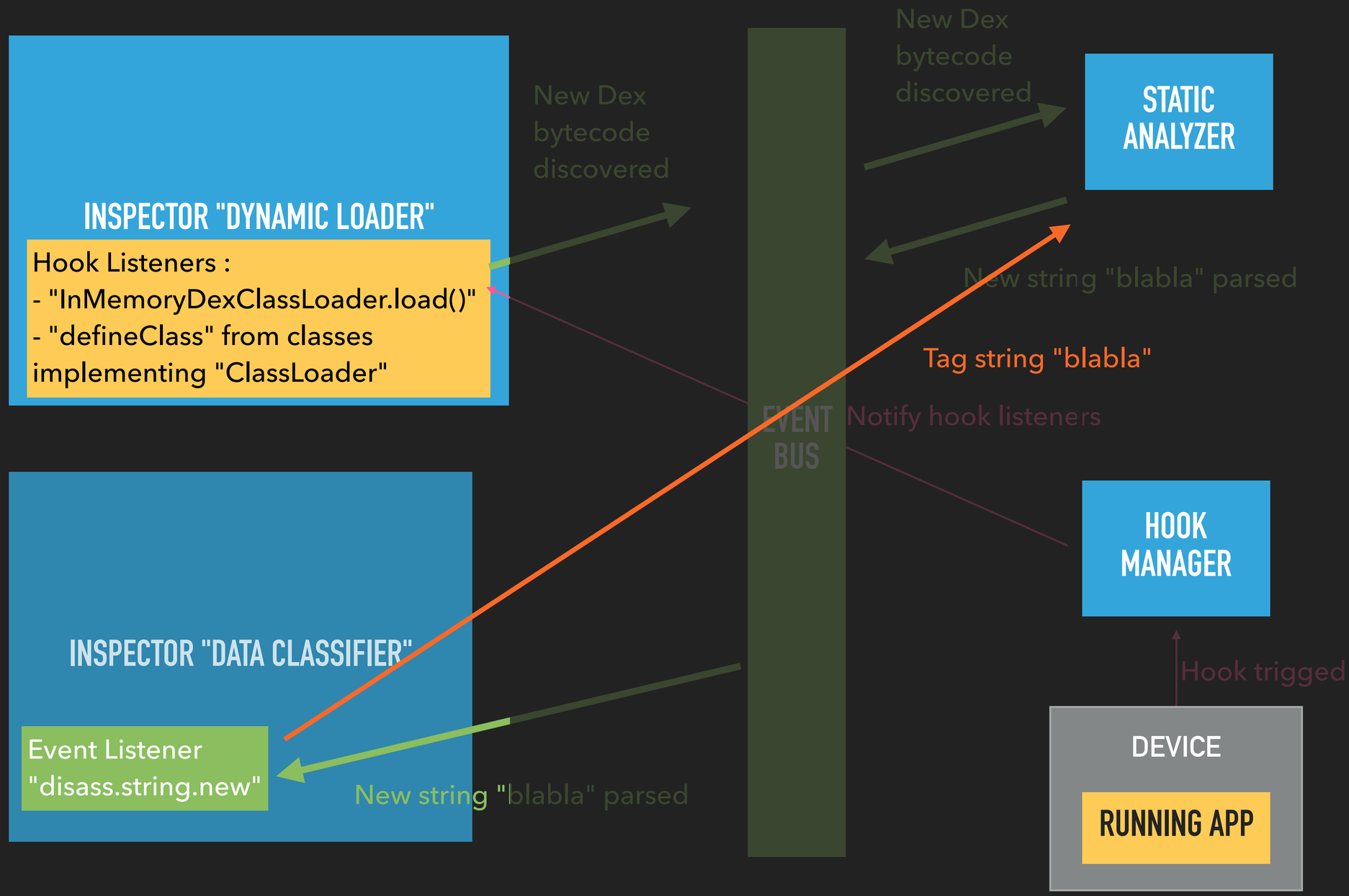
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



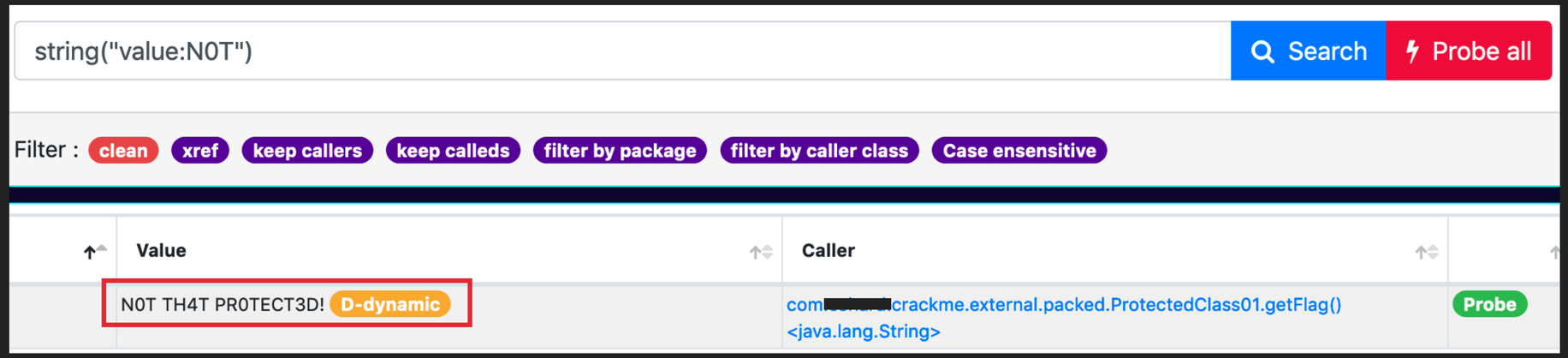
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



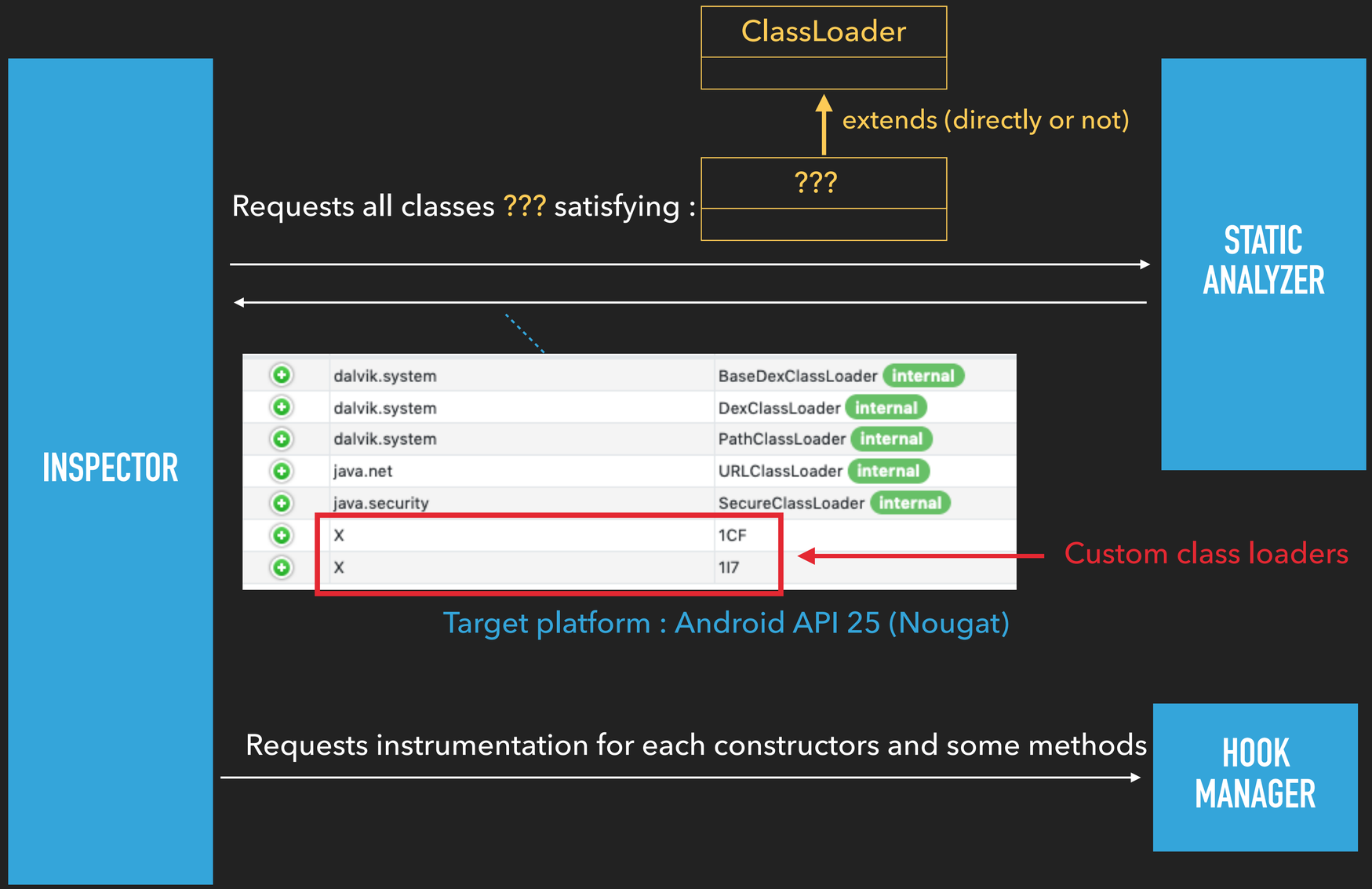
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



CASE #1

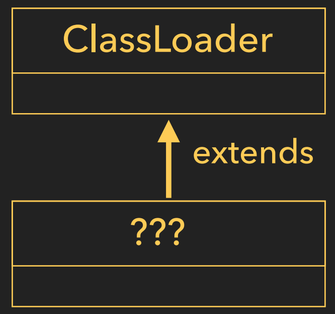
AUTOMATING ANALYSIS OF DEX/ODEX/CLASS FILE LOADED DYNAMICALLY

HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

Request



Result :

	Package	Name
⊕	dalvik.system	BaseDexClassLoader internal
⊕	dalvik.system	DexClassLoader internal
⊕	dalvik.system	PathClassLoader internal
⊕	java.net	URLClassLoader internal
⊕	java.security	SecureClassLoader internal
⊖	X	1CF

Package : X (size: 13092)
Extends : dalvik.system.PathClassLoader < dalvik.system.BaseDexClassLoader < java.lang.ClassLoader < java.lang.Object
Implements : None

Fields

Action	Modifiers	Type	Name

Methods

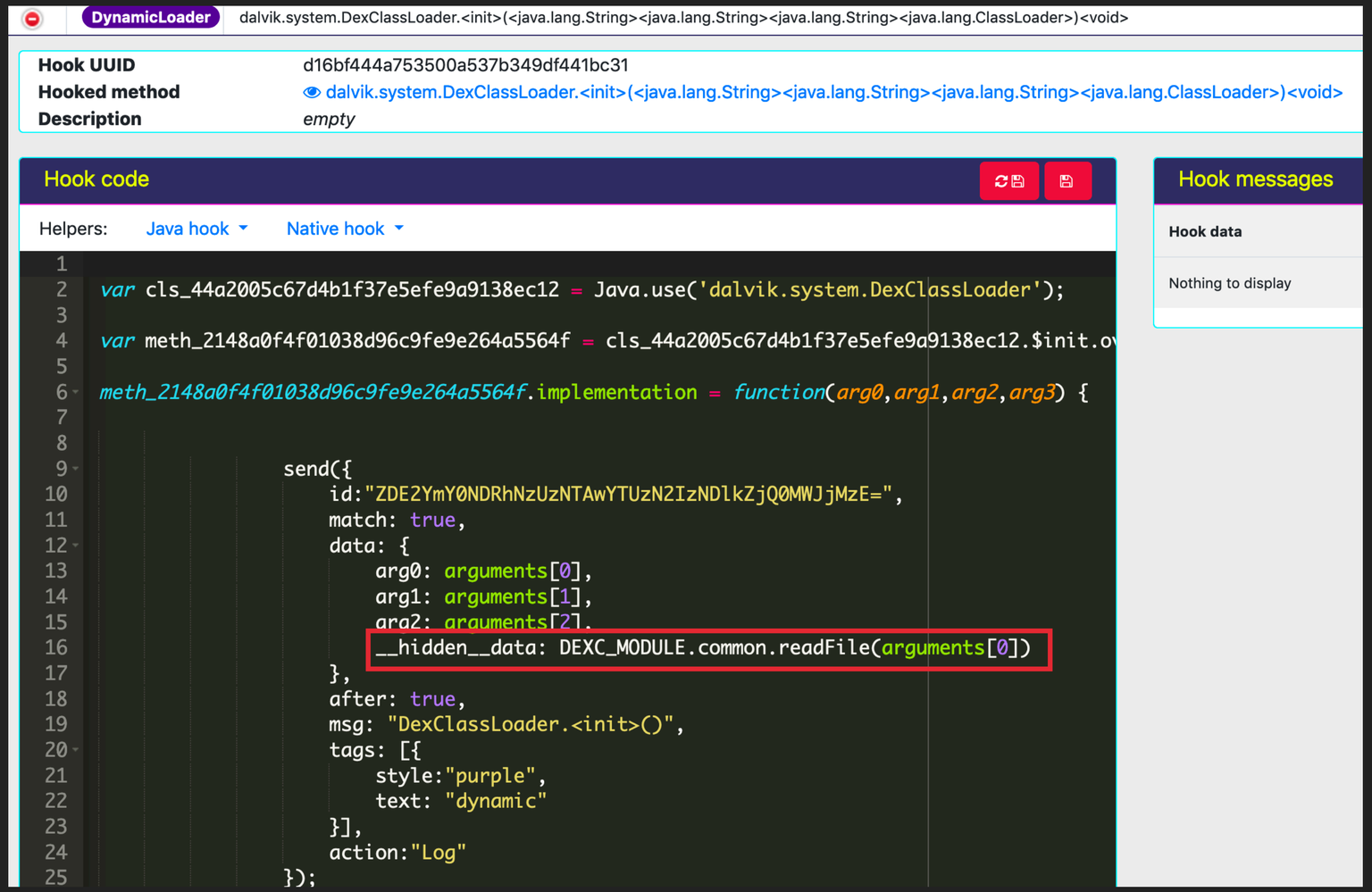
Action	Type	Name
Probe OFF xref to xref from		<init>(<java.lang.String><java.lang.ClassLoader>)<void>
Probe OFF xref to xref from		loadClass(<java.lang.String><boolean>)<java.lang.Class>

⊕	X	1I7

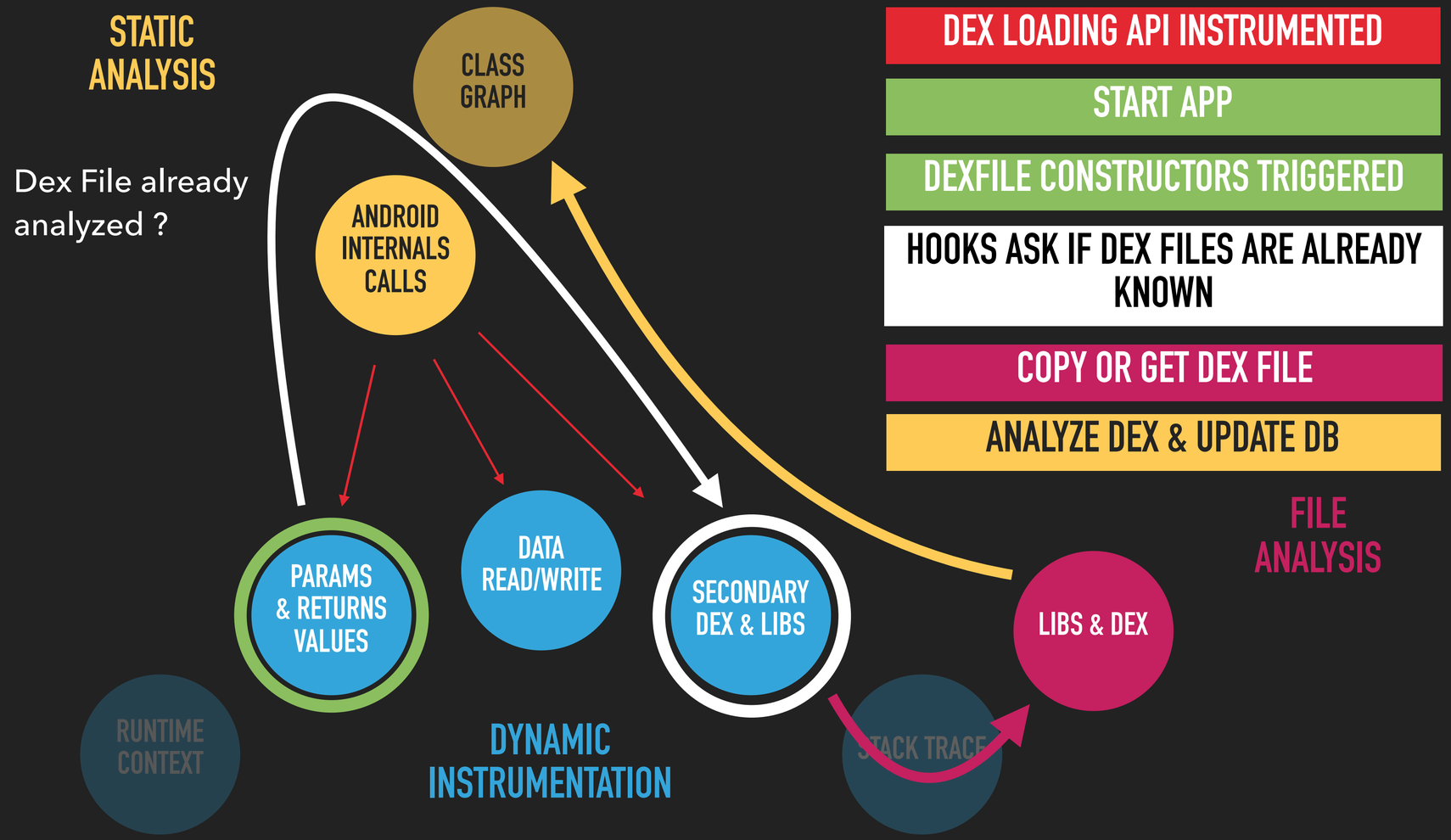
GENERATE HOOK CLASS LOADERS

⊕	DynamicLoader	java.lang.Class.getMethod(<java.lang.String><java.lang.Class>[])<java.lang.reflect.Method>	ON
⊕	DynamicLoader	java.lang.Class.forName(<java.lang.String><boolean><java.lang.ClassLoader>)<java.lang.Class>	ON
⊕	DynamicLoader	dalvik.system.BaseDexClassLoader.findClass(<java.lang.String>)<java.lang.Class>	ON
⊕	DynamicLoader	dalvik.system.DexClassLoader.<init>(<java.lang.String><java.lang.String><java.lang.String><java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	dalvik.system.DexFile.loadDex(<java.lang.String><java.lang.String><int>)<dalvik.system.DexFile>	ON
⊕	DynamicLoader	dalvik.system.DexFile.<init>(<java.io.File>)<void>	ON
⊕	DynamicLoader	dalvik.system.DexFile.<init>(<java.lang.String>)<void>	ON
⊕	DynamicLoader	dalvik.system.BaseDexClassLoader.<init>(<java.lang.String><java.io.File><java.lang.String><java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	dalvik.system.DexClassLoader.<init>(<java.lang.String><java.lang.String><java.lang.String><java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	dalvik.system.PathClassLoader.<init>(<java.lang.String><java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	dalvik.system.PathClassLoader.<init>(<java.lang.String><java.lang.String><java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	java.net.URLClassLoader.<init>(<java.net.URL>[])<void>	ON
⊕	DynamicLoader	java.net.URLClassLoader.<init>(<java.net.URL>[]<java.lang.ClassLoader>)<void>	ON
⊕	DynamicLoader	java.net.URLClassLoader.<init>(<java.net.URL>[]<java.lang.ClassLoader><java.net.URLStreamHandlerFactory>)<void>	ON
⊕	DynamicLoader	java.security.SecureClassLoader.<init>()<void>	ON
⊕	DynamicLoader	java.security.SecureClassLoader.<init>(<java.lang.ClassLoader>)<void>	ON

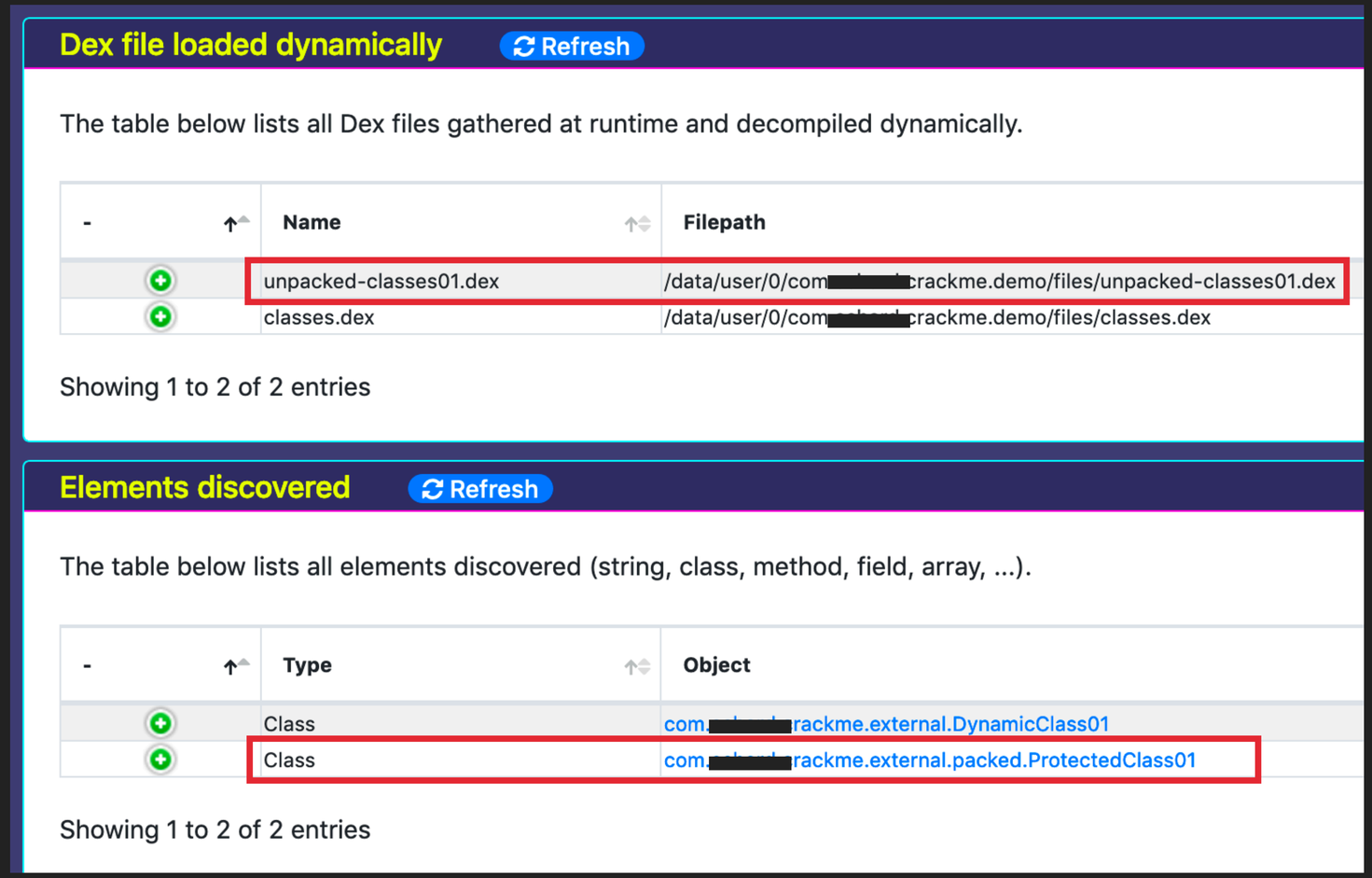
GENERATE HOOK CLASS LOADERS



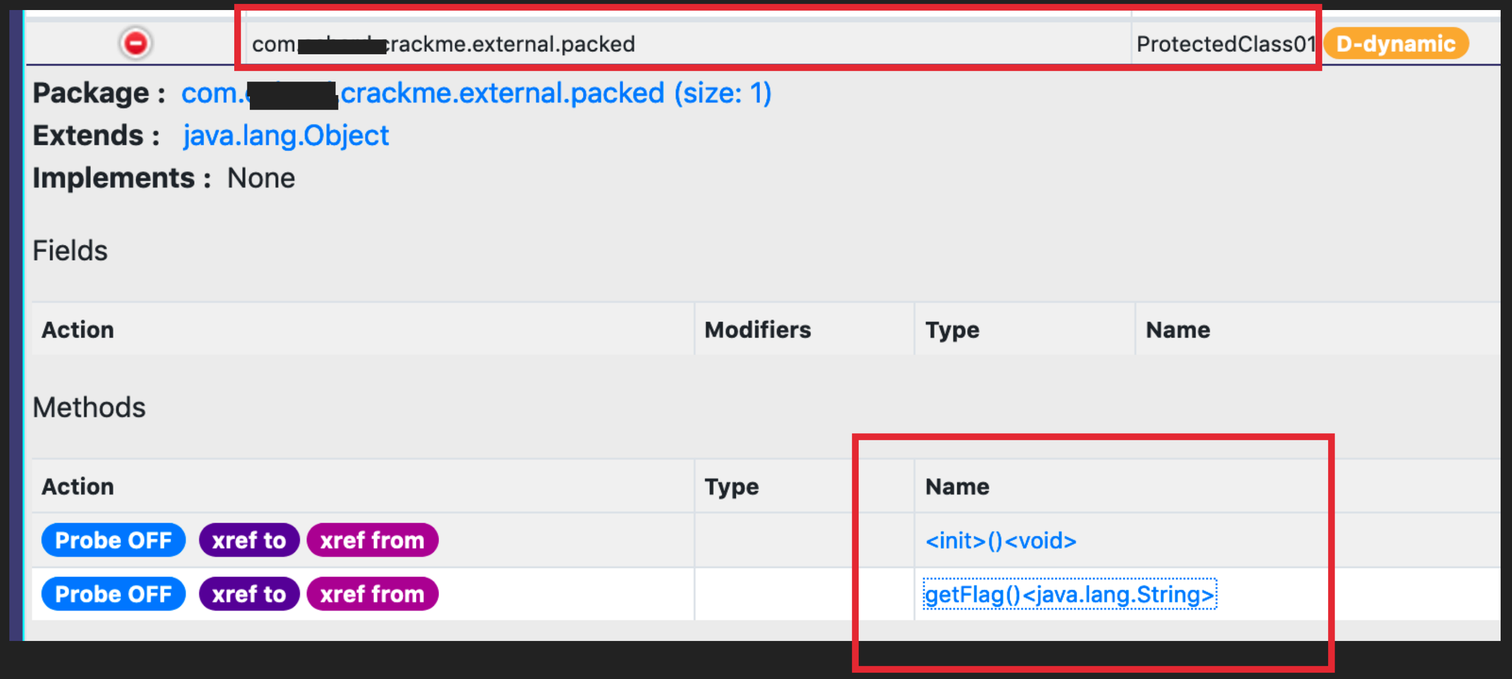
ANALYZING DEX FILE LOADED DYNAMICALLY



AFTER 1ST RUN



AFTER 1ST RUN



AFTER 1ST RUN

com[redacted]crackme.external.packed.ProtectedClass01 | getFlag()<java.lang.String> | D-dynamic

Modifiers	public
Class	com[redacted]crackme.external.packed.ProtectedClass01
Fullname	com[redacted]crackme.external.packed.ProtectedClass01.getFlag
Return	java.lang.String

Smali | Run smali (VM) new | Hook history new

```
1
2  .line 7
3  const-string v0, "N0T TH4T PR0TECT3D!"
4  return-object v0
5
```

(Copyright GTO, feat. Onizuka)

CASE #2

DYNAMIC UPDATE OF XREF WITH INVOKED METHODS

METHOD INVOKED DYNAMICALLY

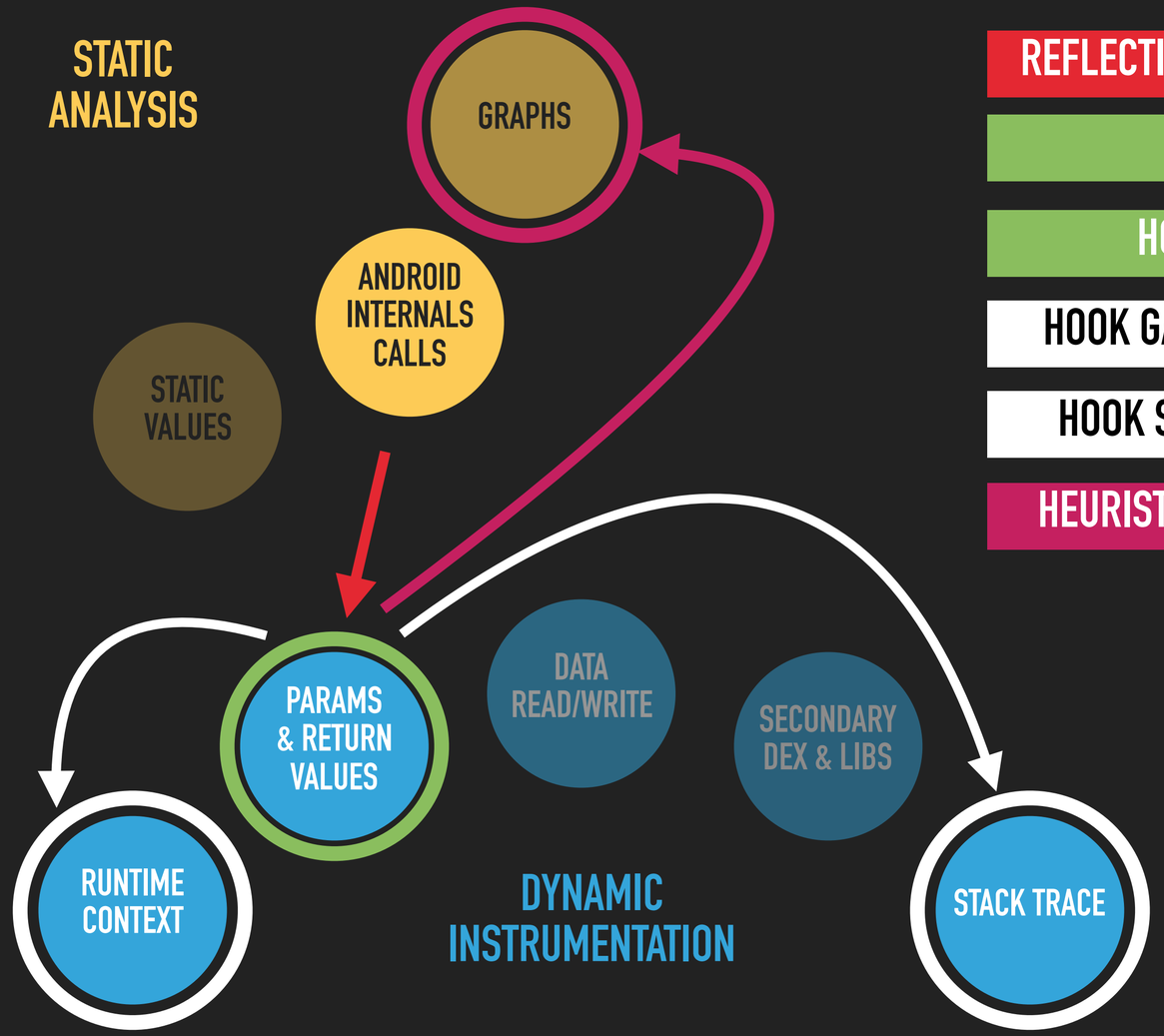
```
2  const v0, 0x1
3  new-array v1, v0, [Ljava/lang/Class;
4  new-array v2, v0, [Ljava/lang/Object;
5  const v0, 0x0
6  const-class v3, Ljava/lang/String;
7  aput-object v3, v1, v0
8  aput-object p0, v2, v0
9  const-string v0, "convertToString"
10 const-class v3, Landroid/content/res/abltMZGC;
11 invoke-virtual {v3, v0, v1}, Ljava/lang/Class;->getMethod(Ljava/lang/String;[Ljava/lang/Class;)Ljava/lang/reflect/Method;
12     move-result-object v0
13 invoke-virtual {v0, v3, v2}, Ljava/lang/reflect/Method;->invoke(Ljava/lang/Object;[Ljava/lang/Object;)Ljava/lang/Object;
14     move-result-object v0
15 check-cast v0, Ljava/lang/String;
16 return-object v0
```

Smali code

From a static point-of-view, only two methods are called :

- Class.getMethod()
- Method.invoke()

DYNAMIC UPDATE OF XREF WITH INVOKED METHODS



REFLECTION API INSTRUMENTED

START APP

HOOK TRIGGERED

HOOK GATHERS METHOD INFO

HOOK SHOWS STACK TRACE

HEURISTIC ENGINE UPDATE DB

METHOD INVOKED DYNAMICALLY

BEFORE RUNTIME

XRef from

com.eshard.a.a.CGSuroKqvFzdyAH(<java.lang.String>)<java.lang.String>

Method	Tags	Action
java.lang.Class.getMethod(<java.lang.String><java.lang.Class>[])<java.lang.reflect.Method>	internal	Probe OFF
java.lang.reflect.Method.invoke(<java.lang.Object><java.lang.Object>[])<java.lang.Object>	internal	Probe OFF

Showing 1 to 2 of 2 entries

AFTER RUNTIME

XRef from

com.eshard.a.a.CGSuroKqvFzdyAH(<java.lang.String>)<java.lang.String>

Method	Tags	Action
android.content.res.abltMZGC.convertToString(<java.lang.String>)<java.lang.String>	invoked	Probe OFF
java.lang.Class.getMethod(<java.lang.String><java.lang.Class>[])<java.lang.reflect.Method>	internal	Probe OFF
java.lang.reflect.Method.invoke(<java.lang.Object><java.lang.Object>[])<java.lang.Object>	internal	Probe OFF

AND MORE ...

SOME COMMON PROBLEMS

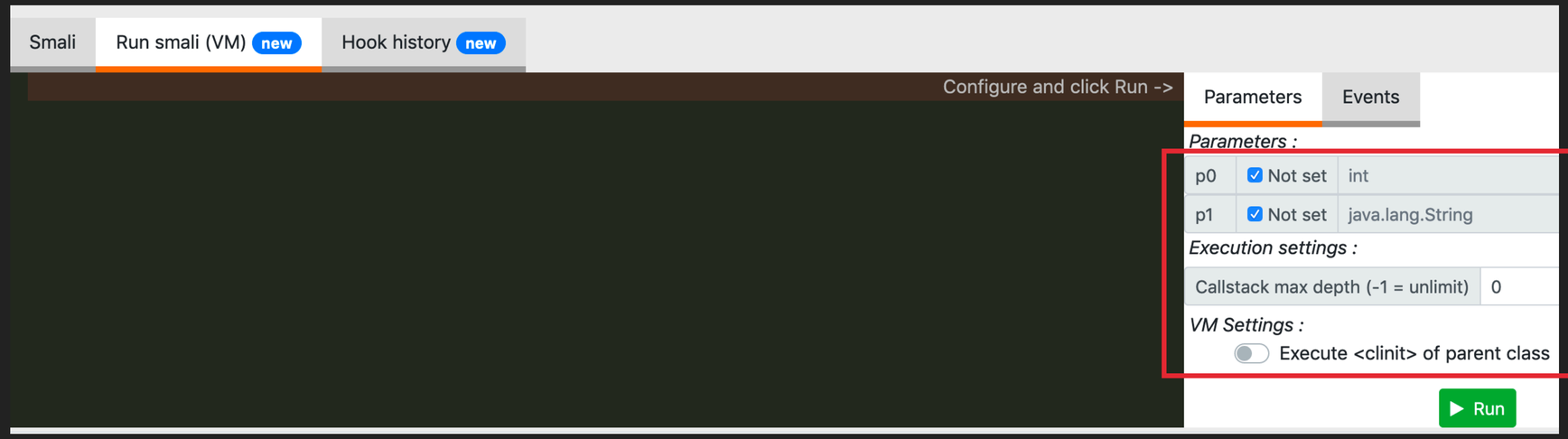
- Always TRUE / FALSE predicate
- Useless Goto(s)
- Implicit exceptions thrown (NPE, IOB, ..)
- Wrapped function
- ...

A- REMOVING USELESS GOTO(S) - BEFORE

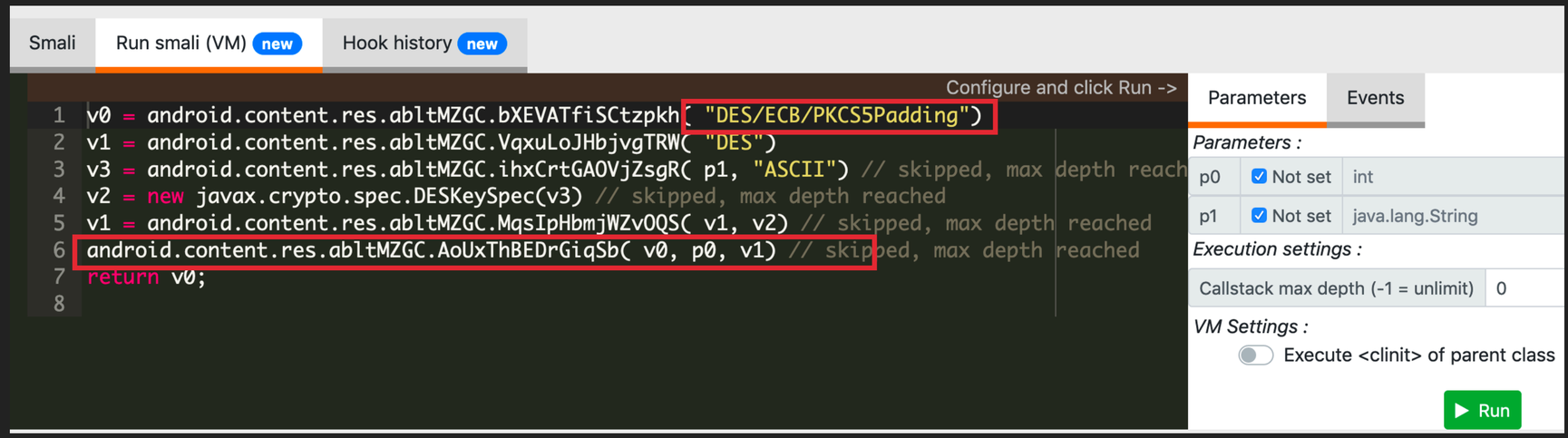
Smali | Run smali (VM) new | Hook history new

```
1
2  goto/32 :goto_c
3
4  nop
5  nop
6
7  :goto_0
8  .line 47
9  goto/32 :goto_3
10
11 nop
12 nop
13 nop
14
15 :goto_1
16 invoke-static {v0, p0, v1}, Landroid/content/res/abltMZGC;->AoUxThBEDrGiqSb(Ljavax/crypto/Cipher;ILjava/security/Key;)V
17 .line 58
18 goto/32 :goto_2
19
20 nop
21 nop
22
23 :goto_2
24 return-object v0
25 nop
26 nop
27
28 :goto_3
29 const-string v1, "DES/ECB/PKCS5Padding"
30 nop
31 goto/32 :goto_7
```

A- REMOVING USELESS GOTO(S) - BEFORE



A- REMOVING USELESS GOTO(S) - AFTER



B- HELP TO DETECT IMPLICIT EXCEPTION

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->HVRQbvhaFlYdeGD(Ljava/lang/String;)Ljava/lang/String;

const/4 v0, 0x1

move-result-object v0

goto/32 :goto_3

:goto_2
const-string v0, "3780b8133459f5a028d742efbcfc7d2d"

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->cVEqFjJdYlzfiuy(Ljava/lang/String;)V

goto/32 :goto_4
```

JADX

```
static {
    HVRQbvhaFlYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfiuy(1);
}
```

B- HELP TO DETECT IMPLICIT EXCEPTION

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->HVRQbvhaFlYdeGD(Ljava/lang/String;)Ljava/lang/String;

const/4 v0, 0x1

move-result-object v0

goto/32 :goto_3

:goto_2
const-string v0, "3780b8133459f5a028d742efbcfc7d2d"

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->cVEqFjJdYlzfiuy(Ljava/lang/String;)V

goto/32 :goto_4
```

Always throws AndroidVerifier exception at runtime

JADX

```
static {
    HVRQbvhaFlYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfiuy(1);
}
```

WRONG

B- HELP TO DETECT IMPLICIT EXCEPTION

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->HVRQbvhaFlYdeGD(Ljava/lang/String;)Ljava/lang/String;

const/4 v0, 0x1

move-result-object v0

goto/32 :goto_3

:goto_2
const-string v0, "3780b8133459f5a028d742efbcfc7d2d"

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->cVEqFjJdYlzfiuy(Ljava/lang/String;)V

goto/32 :goto_4
```

Always throws AndroidVerifier exception at runtime

JADX

```
static {
    HVRQbvhaFlYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfiuy(1);
}
```

WRONG

TRUE PATH

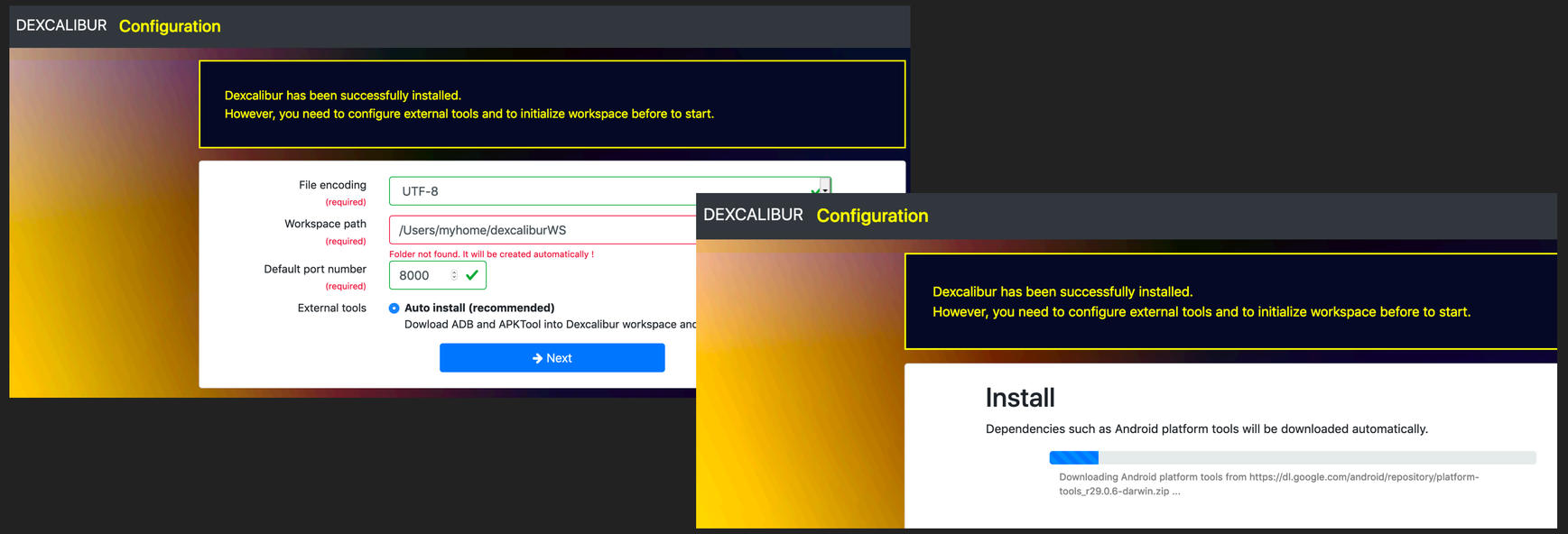
```
static {
    HVRQbvhaFlYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    throw new AndroidVerifier();
}
```

AND FINALLY ... HOW TO INSTALL DEXCALIBUR ?

FROM NPM ...

```
$ npm install  dexcalibur

$ dexcalibur
```

WITH USER-FRIENDLY UI AND AUTOMATIC INSTALL OF DEPENDENCIES:



Thanks

(Copyright GTO, feat. Onizuka)

Q&A