

DEXCALIBUR



Hook it yourself !

WHO AM I ?

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/gethervent/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/gethervent/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, "3780b8133459f5a0280"

goto/32 :goto_1

:goto_3
invoke-static {v0}, Lcom/.../crack...
```

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	bVjSqqAehlQJpnT(<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	rxmCKugynIUZGEh(<java.lang.String><int>)<char>
static	sJKaZhFGICTUuwz(<java.lang.String>)<int>

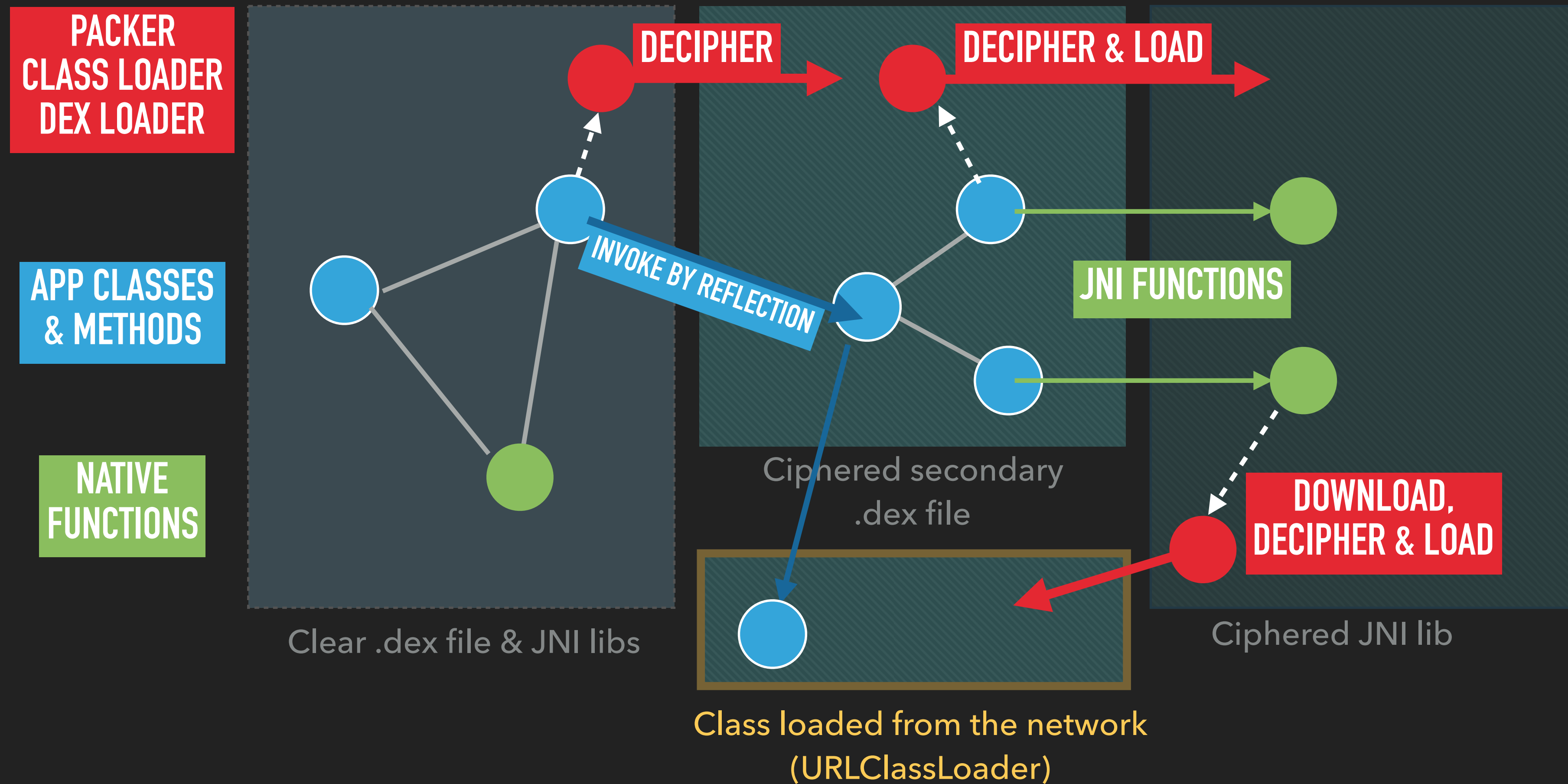
STATIC OBFUSCATION, REAL LIFE

```
2 goto/32 :goto_7b
3
4
5 :goto_0
6 goto/32 :goto_75
7
8
9 :goto_1
10 goto/16 :goto_27
11
12
13 :sswitch_0
14 .line 112
15 goto/32 :goto_79
16
17
18 :goto_2
19 invoke-static {v4, v0},
20 goto/32 :goto_33
21
22
23 :goto_3
24 if-ge v3, v8, :cond_0
25
26 goto/32 :goto_26
27
28
29 :cond_0
30 goto/32 :goto_5c
31
32
33 :goto_4
34 invoke-static {p0, v6},
35 move-result v6
36 goto/32 :goto_77
37
38
39 :goto_5
40 const/16 v0, 0xd
41 .line 101
42 goto/32 :goto_23
43
44
45 :goto_6
46 if-lt v2, v6, :cond_1
47
48 goto/32 :goto_38
49
50
51 :cond_1
52 goto/32 :goto_3f
53
54
55 :goto_7
56 invoke-static {v6, v3}, Landroid/content/res/abltMZGC; ->XjYcgdUCozPhGQw(Lj
57     move-result-object v6
58 goto/32 :goto_53
59
60
61 :goto_8
62 if-lt v2, v6, :cond_2
63
64 goto/32 :goto_38
65
66
67 :cond_2
68 goto/32 :goto_20
69
70
71 :goto_9
72 add-int/lit8 v6, v2, 0x1
73 goto/32 :goto_5b
74
```

ETC ...

... ~ 124 goto(s)

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

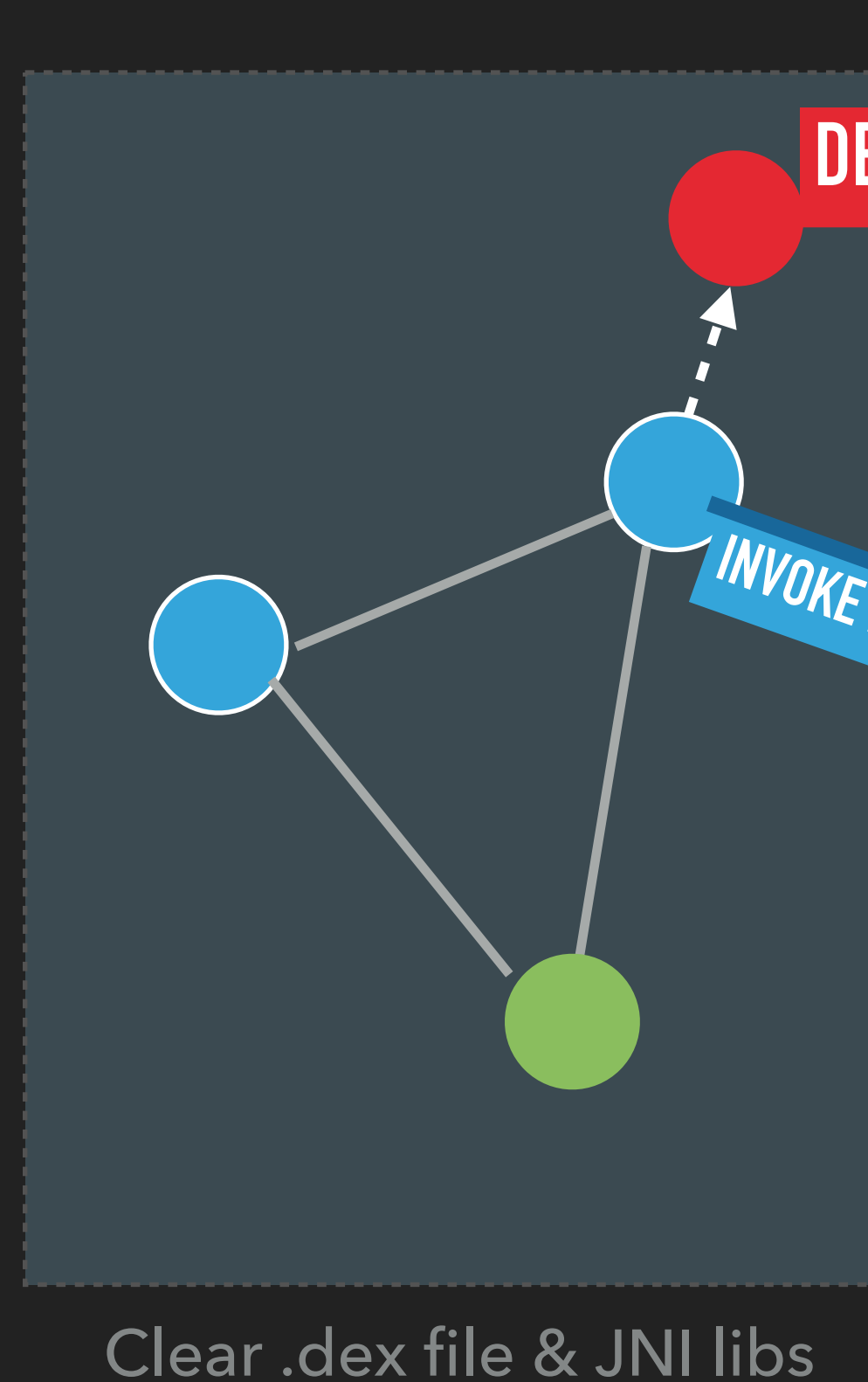


DYNAMIC (MULTI-LEVELS) OBFUSCATION, WHAT CAN I HOOK ?

PACKER
CLASS LOADER
DEX LOADER

APP CLASSES
& METHODS

NATIVE
FUNCTIONS



DECIPHER

DECIPHER & LOAD

INVOKE BY REFLECTION

JNI FUNCTIONS

Ciphered secondary
.dex file

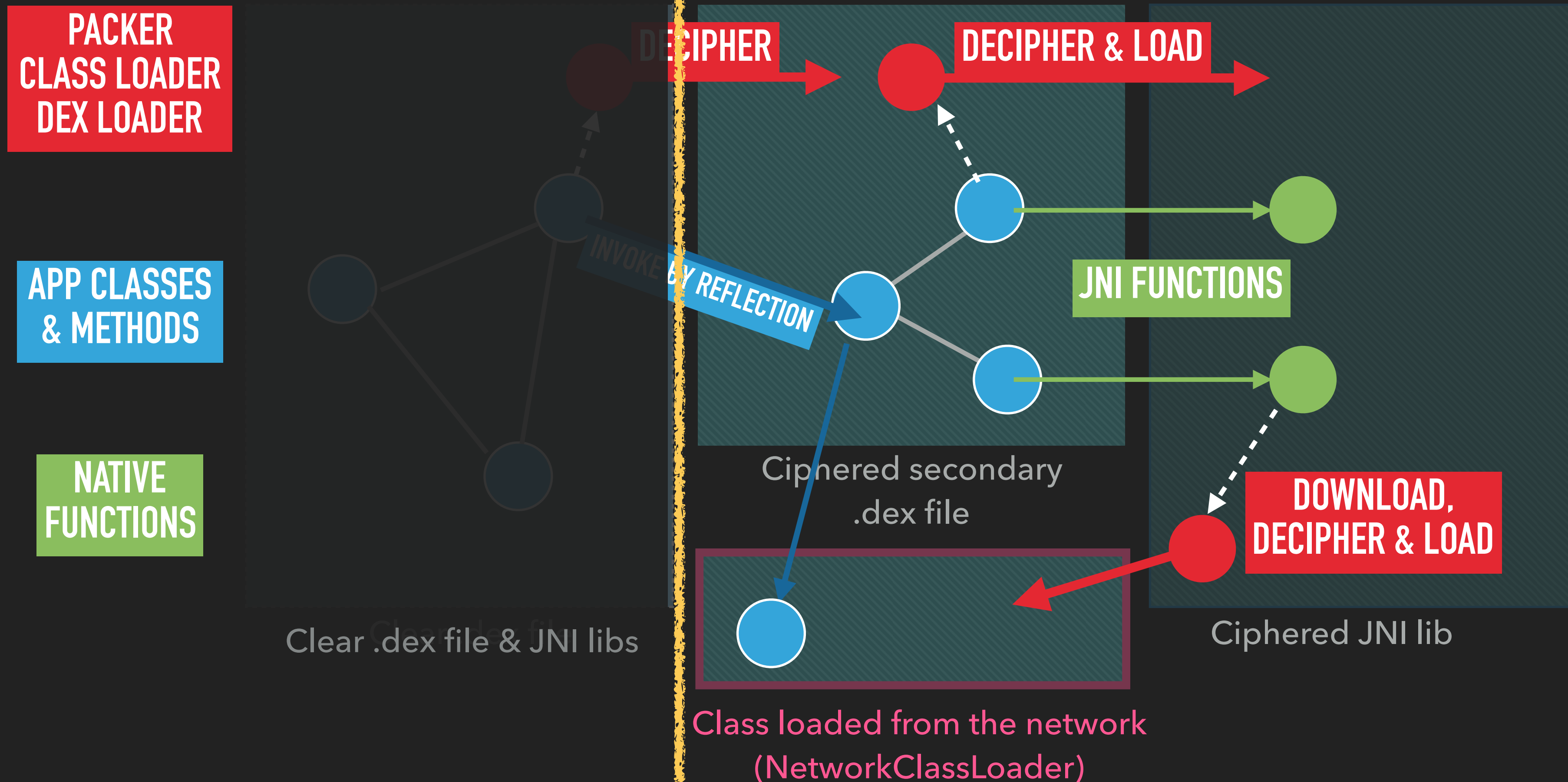
DOWNLOAD,
DECIPHER & LOAD

Ciphered JNI lib

Class loaded from the network
(NetworkClassLoader)

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

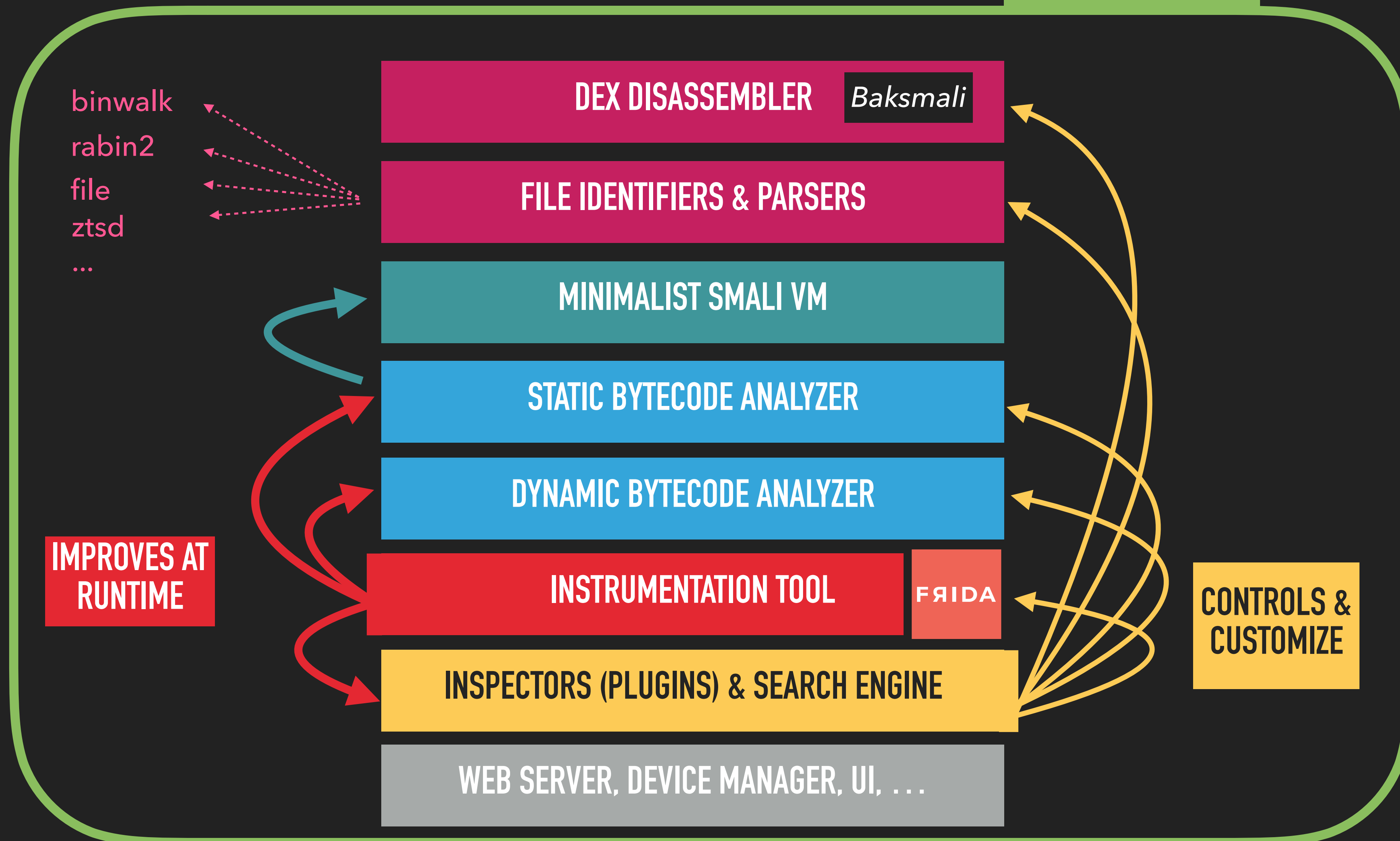


(Copyright GTO, feat. Onizuka)

**WHAT IS
DEXCALIBUR ?**

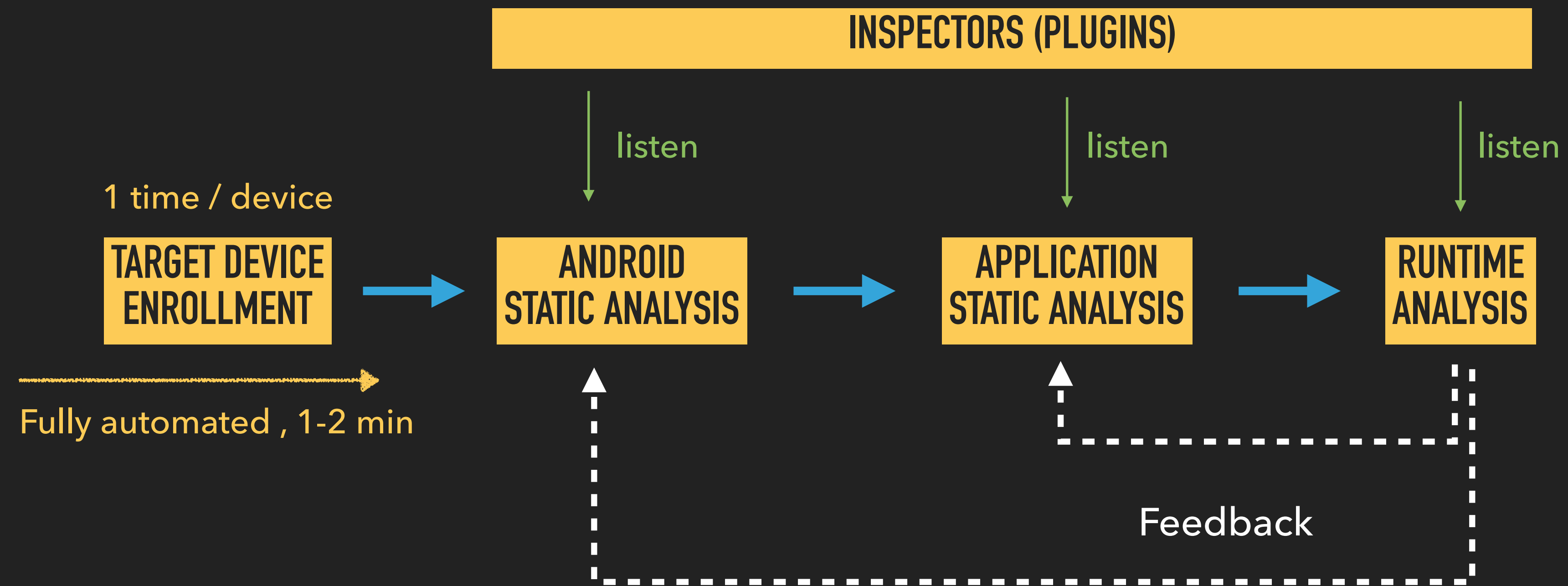
NOT JUST A TOOLBOX

DEXCALIBUR

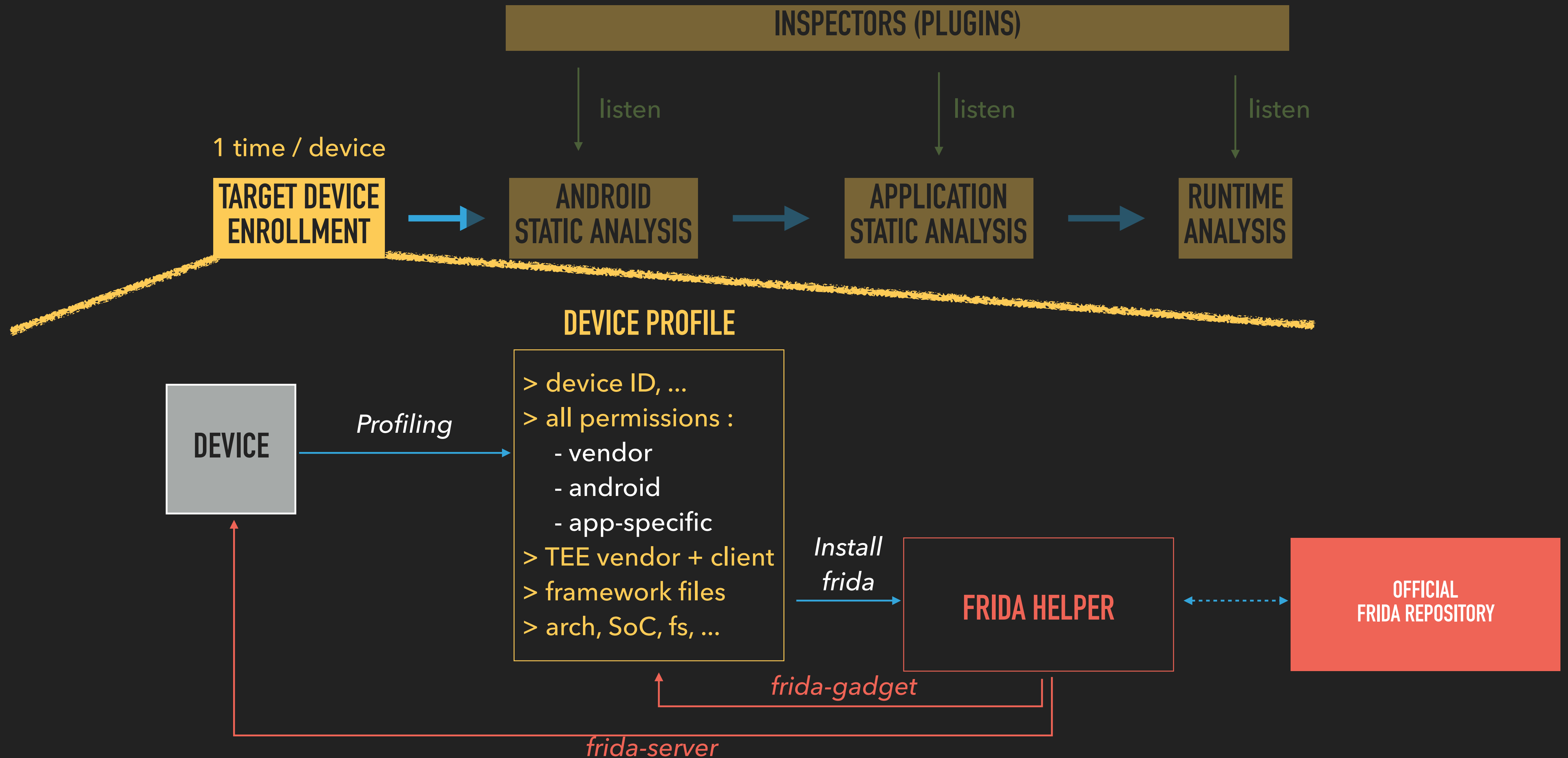


HOW IT WORKS ?

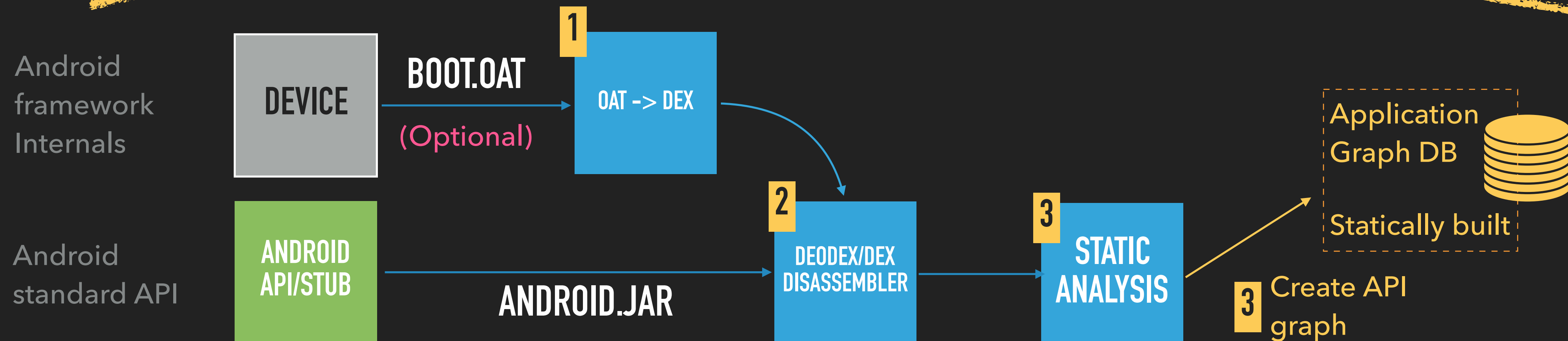
HOW IT WORKS ?



HOW IT WORKS ? - ENROLLMENT PHASE



HOW IT WORKS ? - ANDROID FRAMEWORK ANALYSIS PHASE



HOW IT WORKS ? - ANDROID FRAMEWORK ANALYSIS PHASE

XRef from
com.██████████.listeners.CrackMeFragmentOnClickListener.a0cc175b9(<java.lang.String>)<boolean>

Type ↑	Name	Tags	Action
Field	com.██████████.listeners.CrackMeFragmentOnClickListener;->a92eb5ffe		
Field	com.██████████.listeners.CrackMeFragmentOnClickListener;->a8277e091		
Field	com.██████████.listeners.CrackMeFragmentOnClickListener;->ae1671797		
Method	com.██████████.listeners.CrackMeFragmentOnClickListener.iUkHYzIRgoxaqjf(<java.lang.String><java.lang.String>)<java.lang.String>		Probe OFF
Method	java.lang.String.<init>(<byte>[])<void>	internal	Probe OFF
Method	com.██████████.listeners.CrackMeFragmentOnClickListener.VKHaXUOBWfqdhbt(<java.lang.String>)<java.lang.String>		Probe OFF
Method	com.██████████.listeners.CrackMeFragmentOnClickListener.yNJloHzhgiOCP(<java.lang.String>)<boolean>		

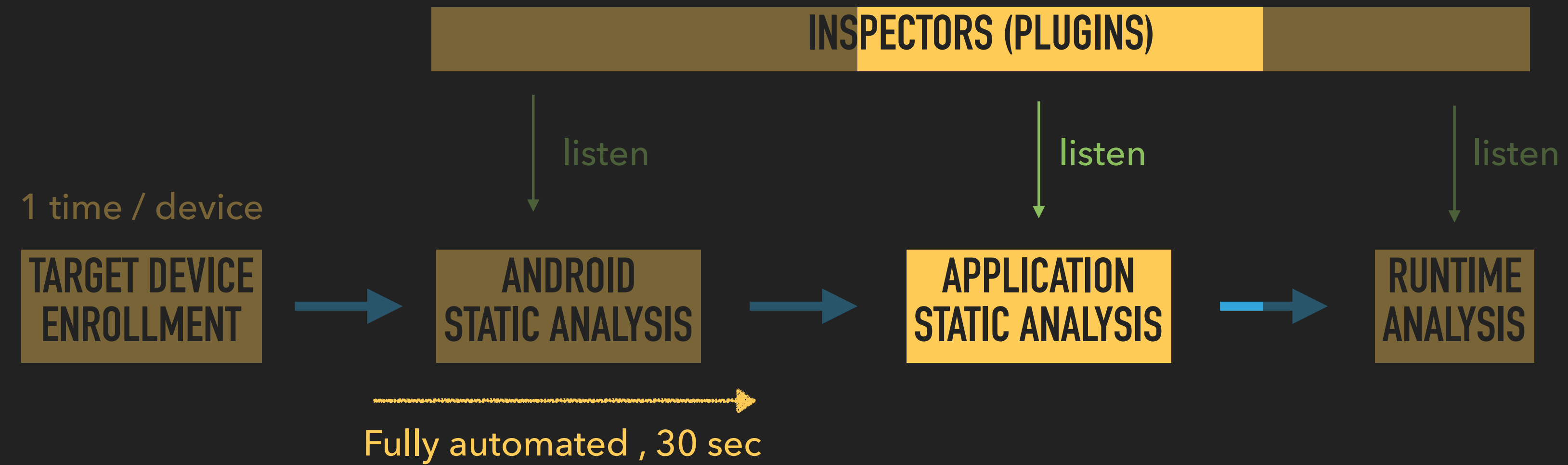
Showing 1 to 7 of 7 entries

Search ...
call > calleed > enclosingClass > name > ☐ Case sensitive ☐ App only

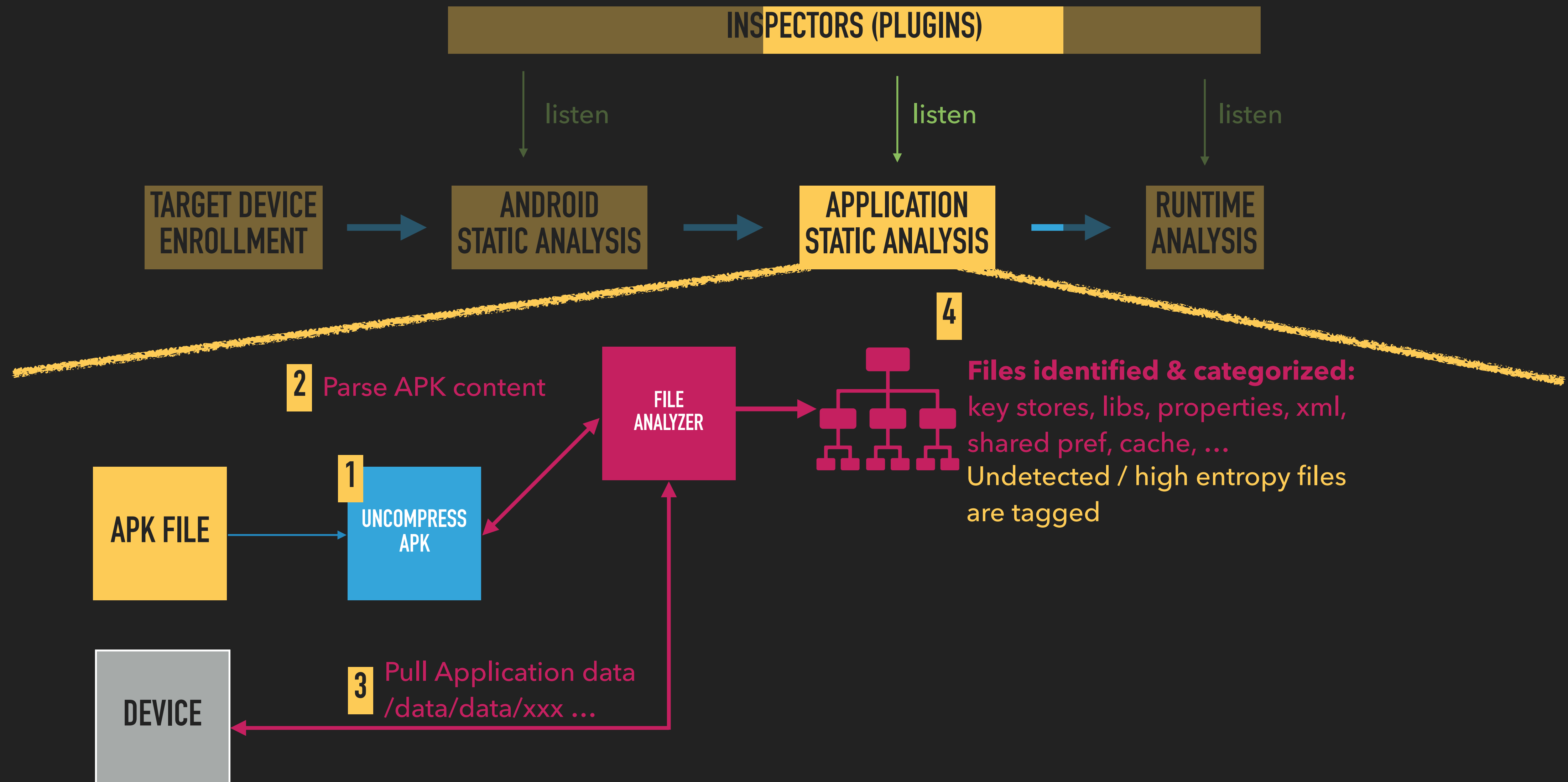
↑	Caller	↓	Callee
+	X.04J.A0M(<java.lang.String>)<X.05e>	method	java.security.KeyStore.getInstance(<java.lang.String>)<java.security.KeyStore>
+	X.04J.A0M(<java.lang.String>)<X.05e>	method	java.security.KeyStore.load(<java.security.KeyStore\$LoadStoreParameter>)<void>
+	X.04J.A0M(<java.lang.String>)<X.05e>	method	java.security.KeyStore.getKey(<java.lang.String><char>[])<java.security.Key>
+	X.04J.A0p(<java.lang.String>)<java.security.PublicKey>	class	android.security.keystore.KeyGenParameterSpec\$Builder
+	X.04J.A0p(<java.lang.String>)<java.security.PublicKey>	method	android.security.keystore.KeyGenParameterSpec\$Builder.<init>(<java.lang.String><int>)<void>
+	X.04J.A0p(<java.lang.String>)<java.security.PublicKey>	method	android.security.keystore.KeyGenParameterSpec\$Builder.setDigests(<java.lang.String>[])<android.security.keystore.KeyGenParameterSpec\$Builder>
+	X.04J.A0p(<java.lang.String>)<java.security.PublicKey>	method	android.security.keystore.KeyGenParameterSpec\$Builder.setAlgorithmParameterSpec(<java.security.spec.AlgorithmParameterSpec>)<android.security.keystore.KeyGenParameterSpec\$Builder>
+	X.04J.A0p(<java.lang.String>)<java.security.PublicKey>	method	android.security.keystore.KeyGenParameterSpec\$Builder.setUserAuthenticationRequired(<boolean>)<android.security.keystore.KeyGenParameterSpec\$Builder>

click to generate + deploy a hook

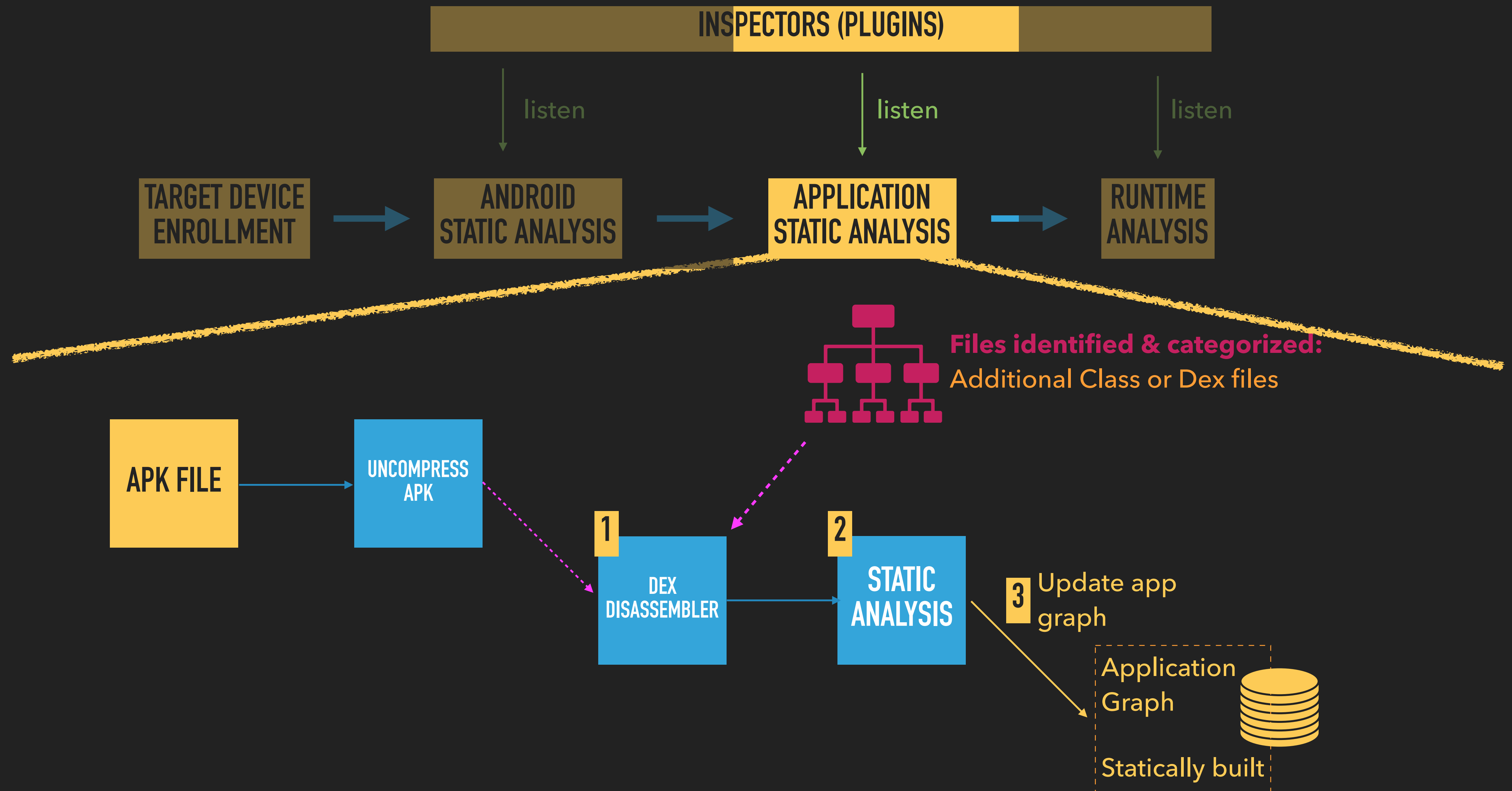
HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE



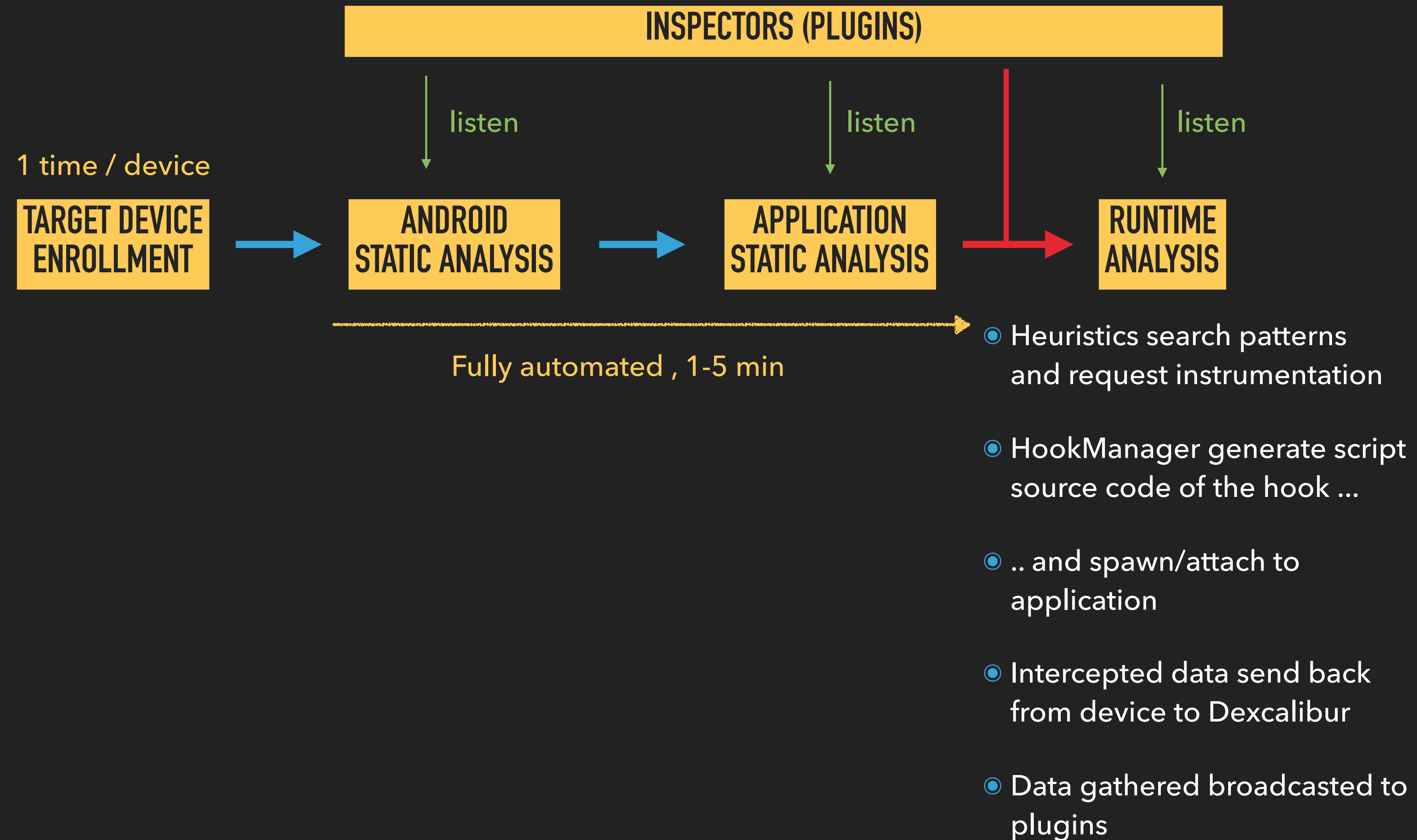
HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE



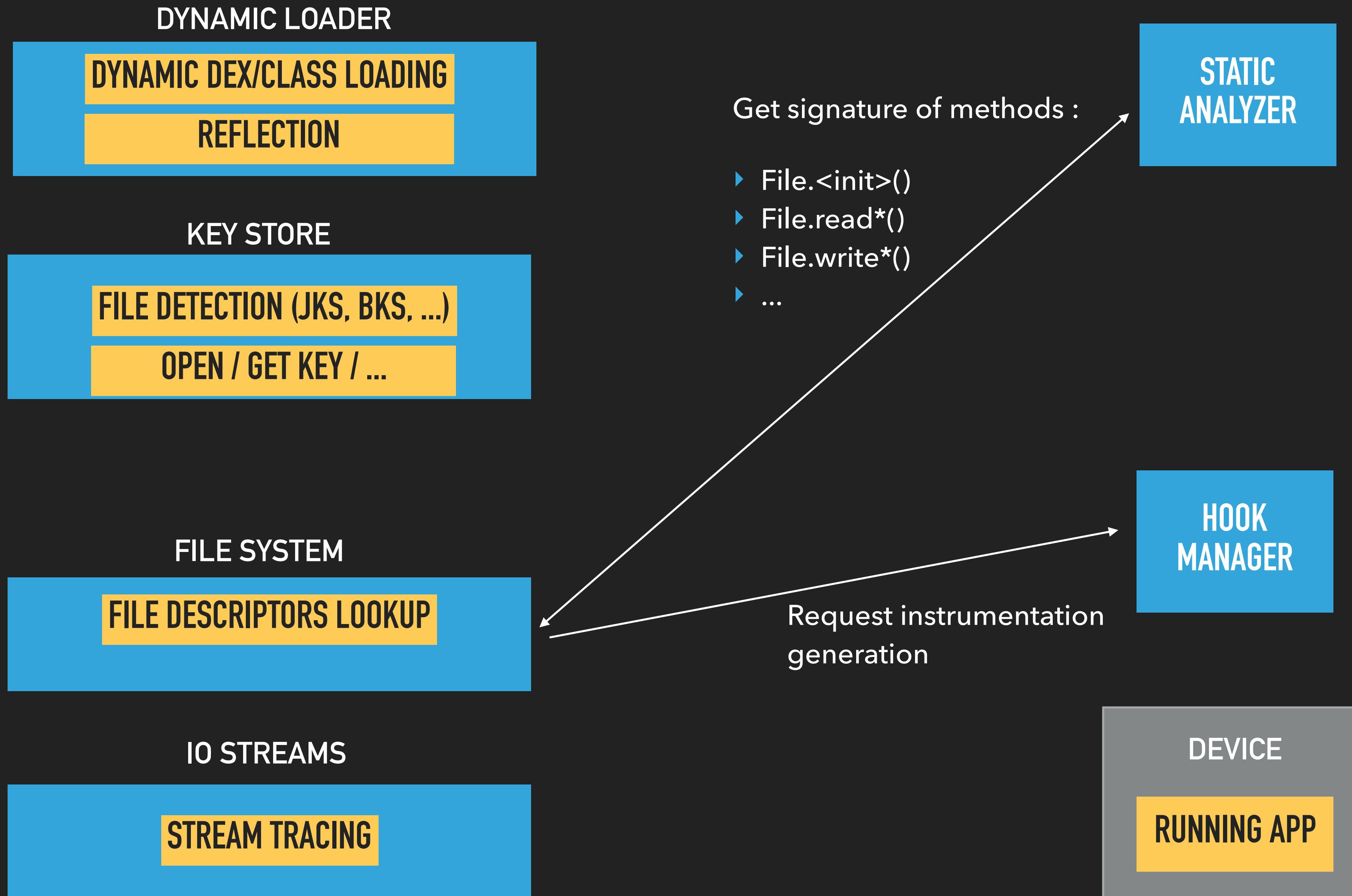
HOW IT WORKS ? - APPLICATION STATIC ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

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

DEVICE

RUNNING APP

HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

INSPECTOR "DYNAMIC LOADER"

Hook Listeners :

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

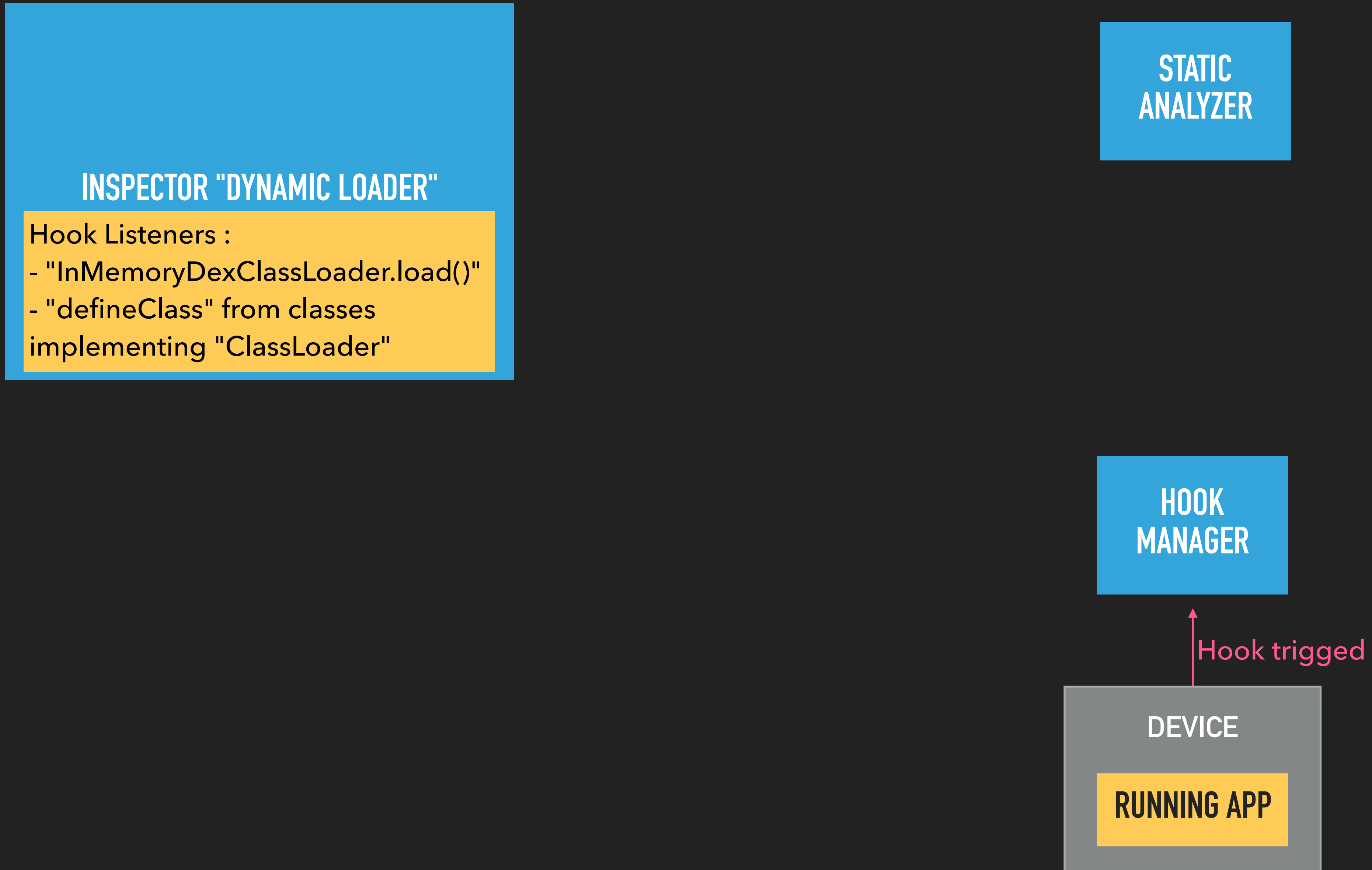
STATIC
ANALYZER

HOOK
MANAGER

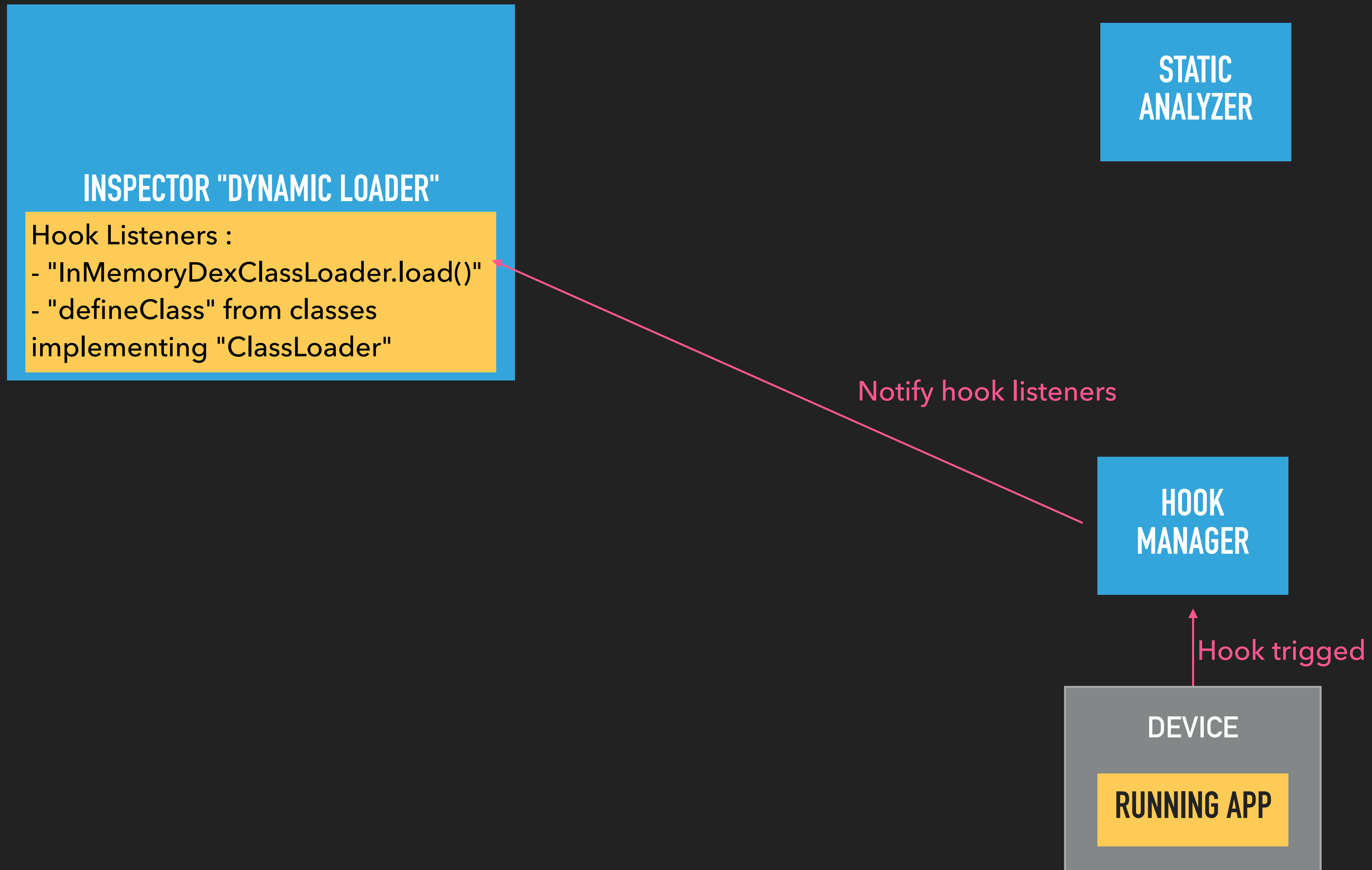
↑ Hook triggered

DEVICE

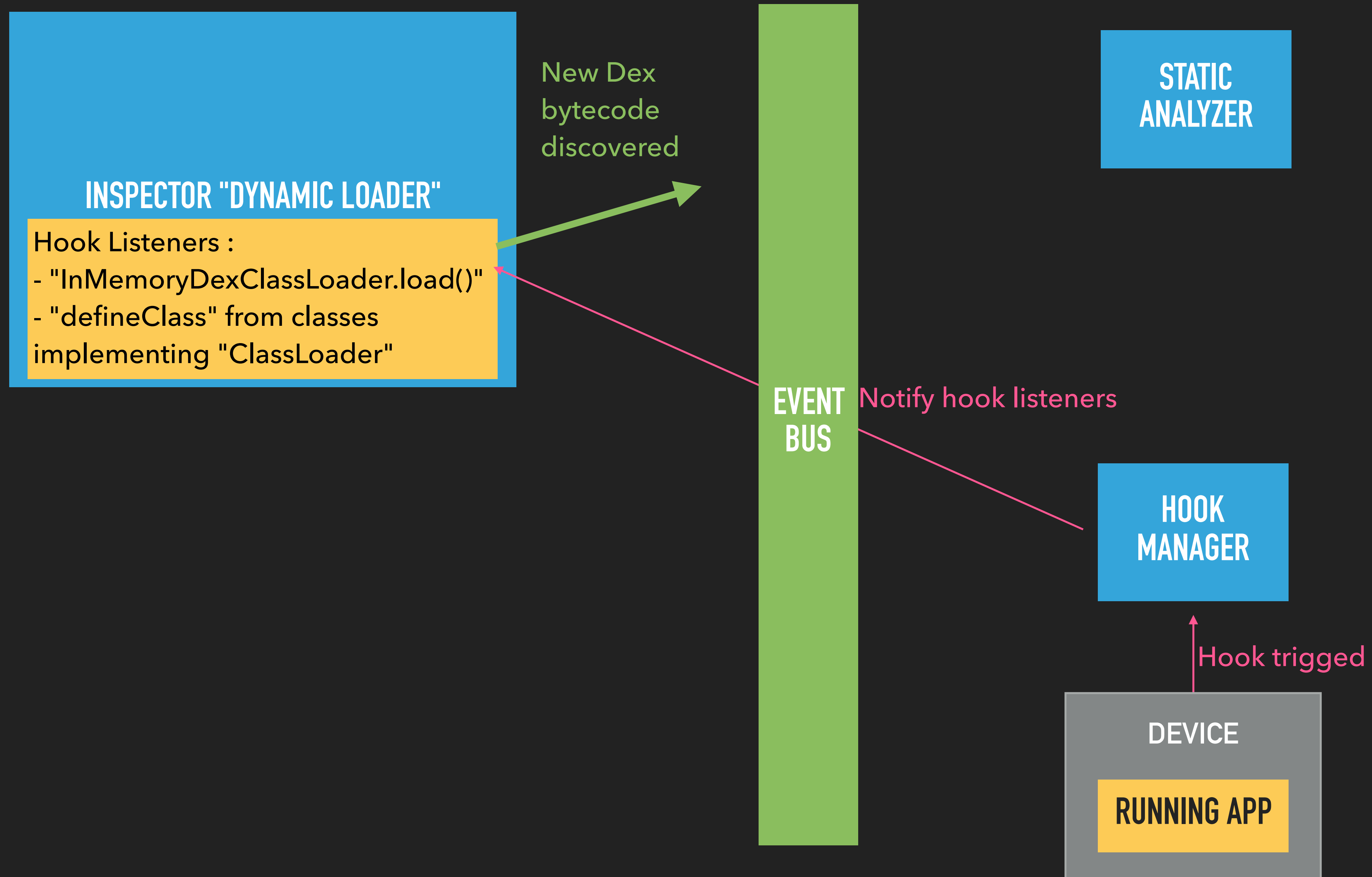
RUNNING APP



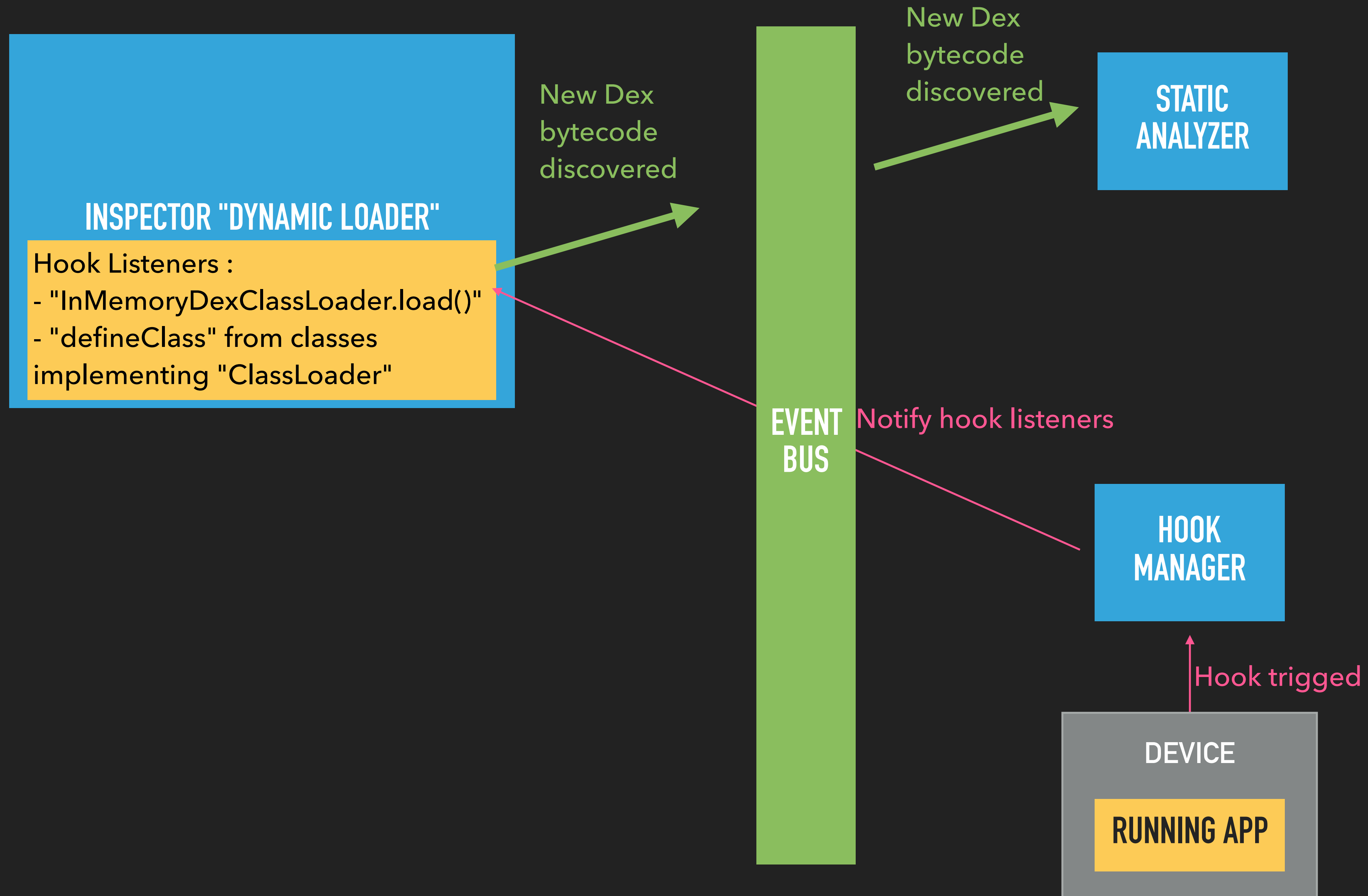
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



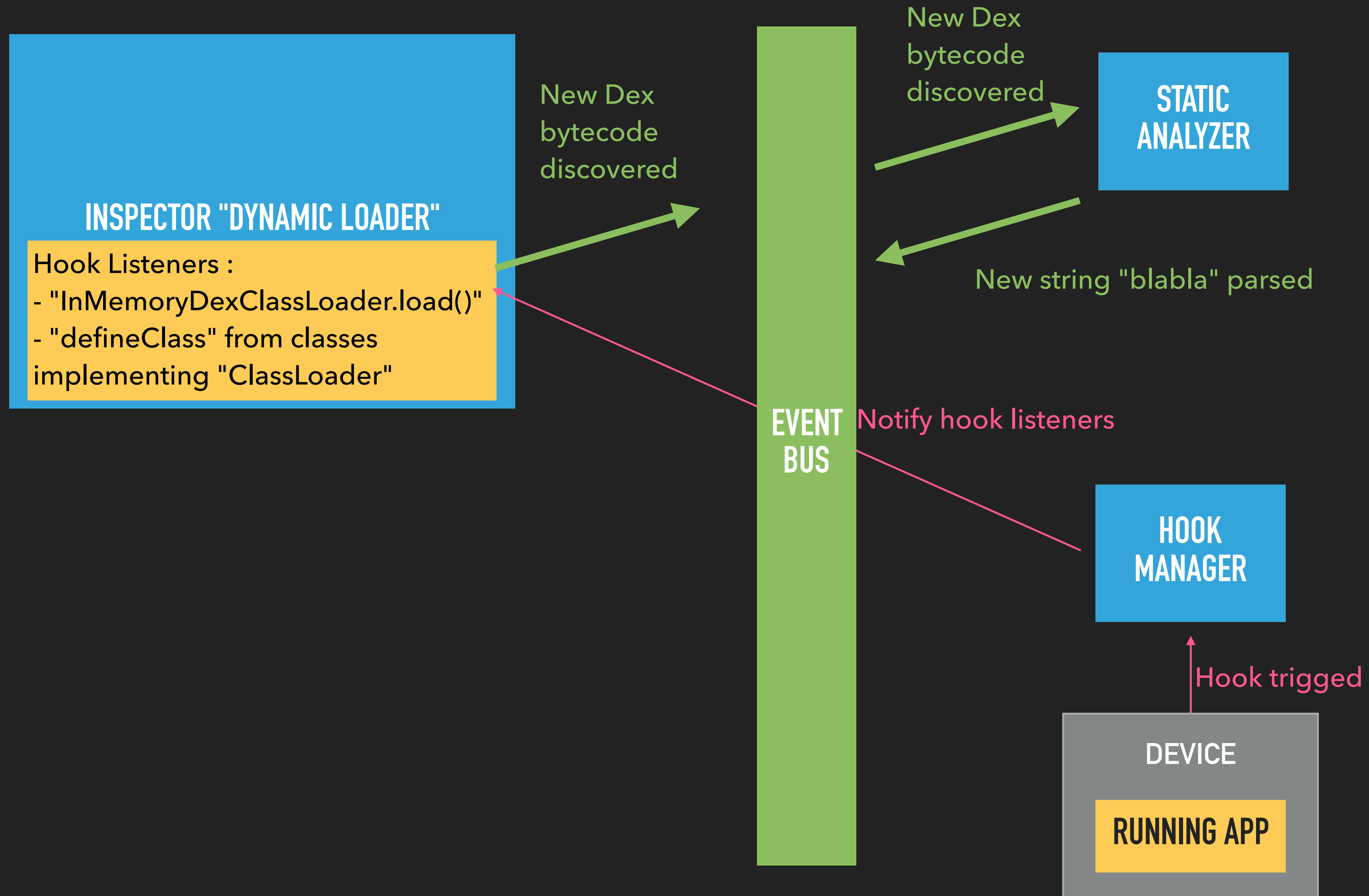
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



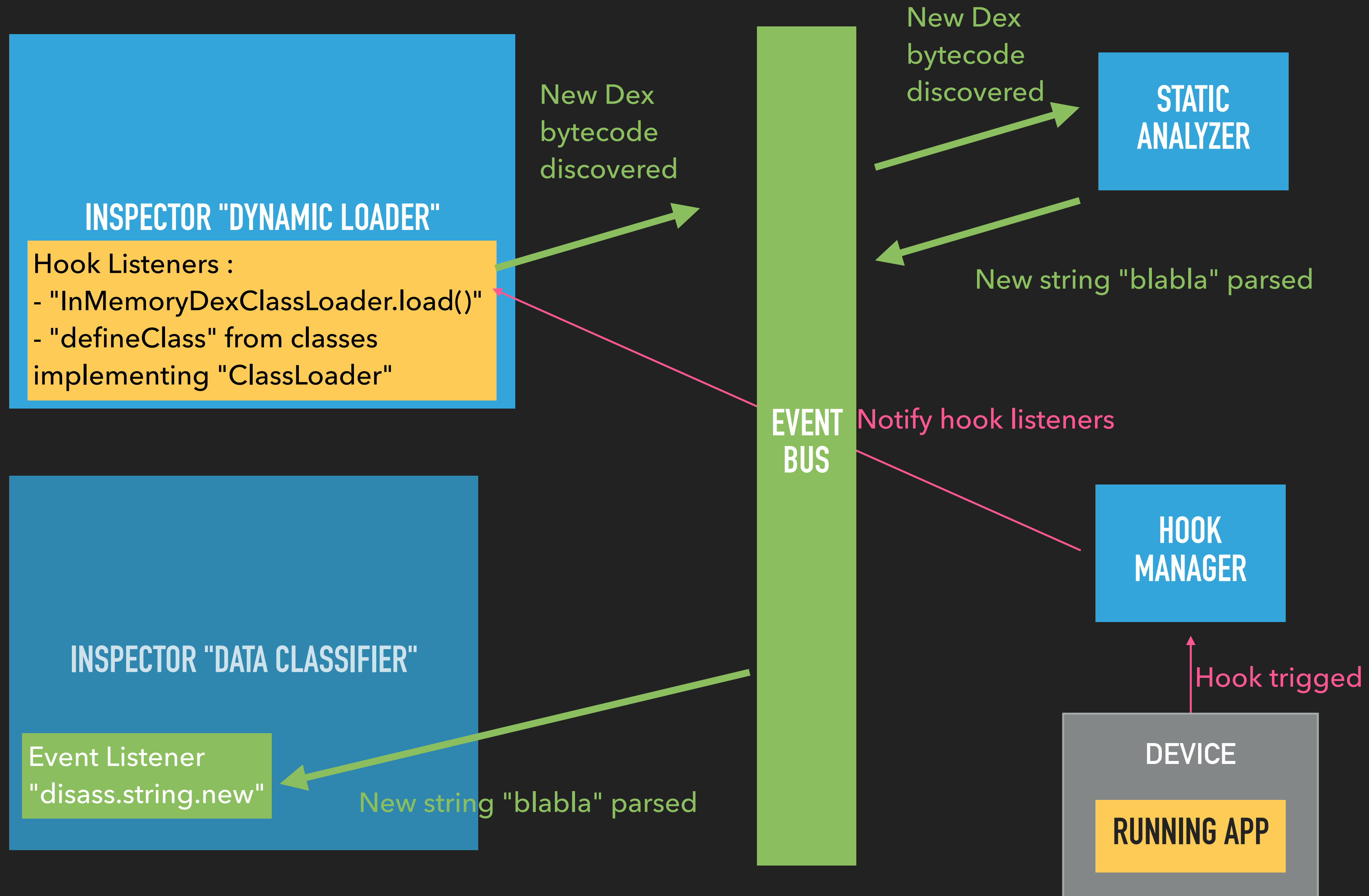
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



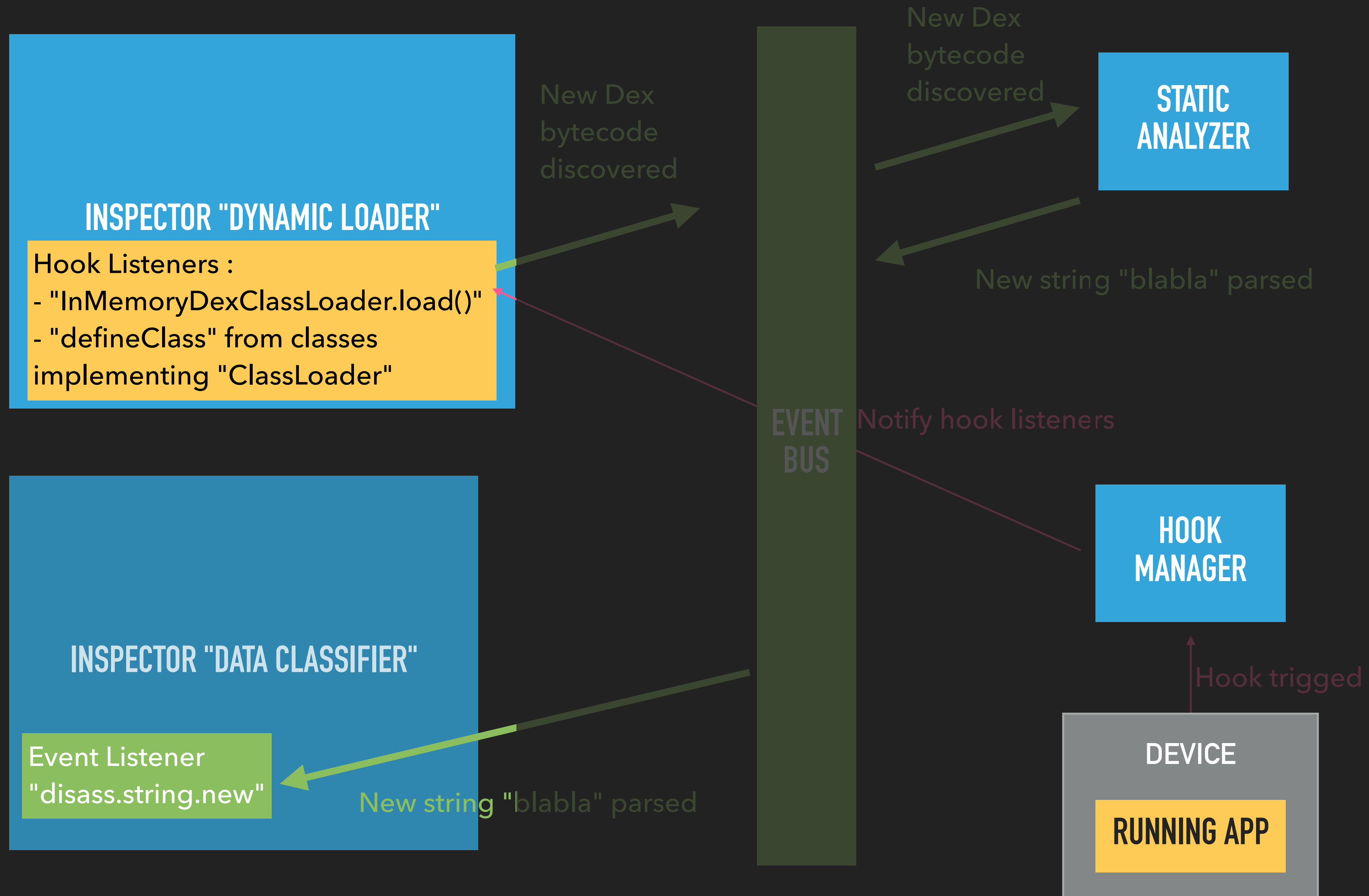
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



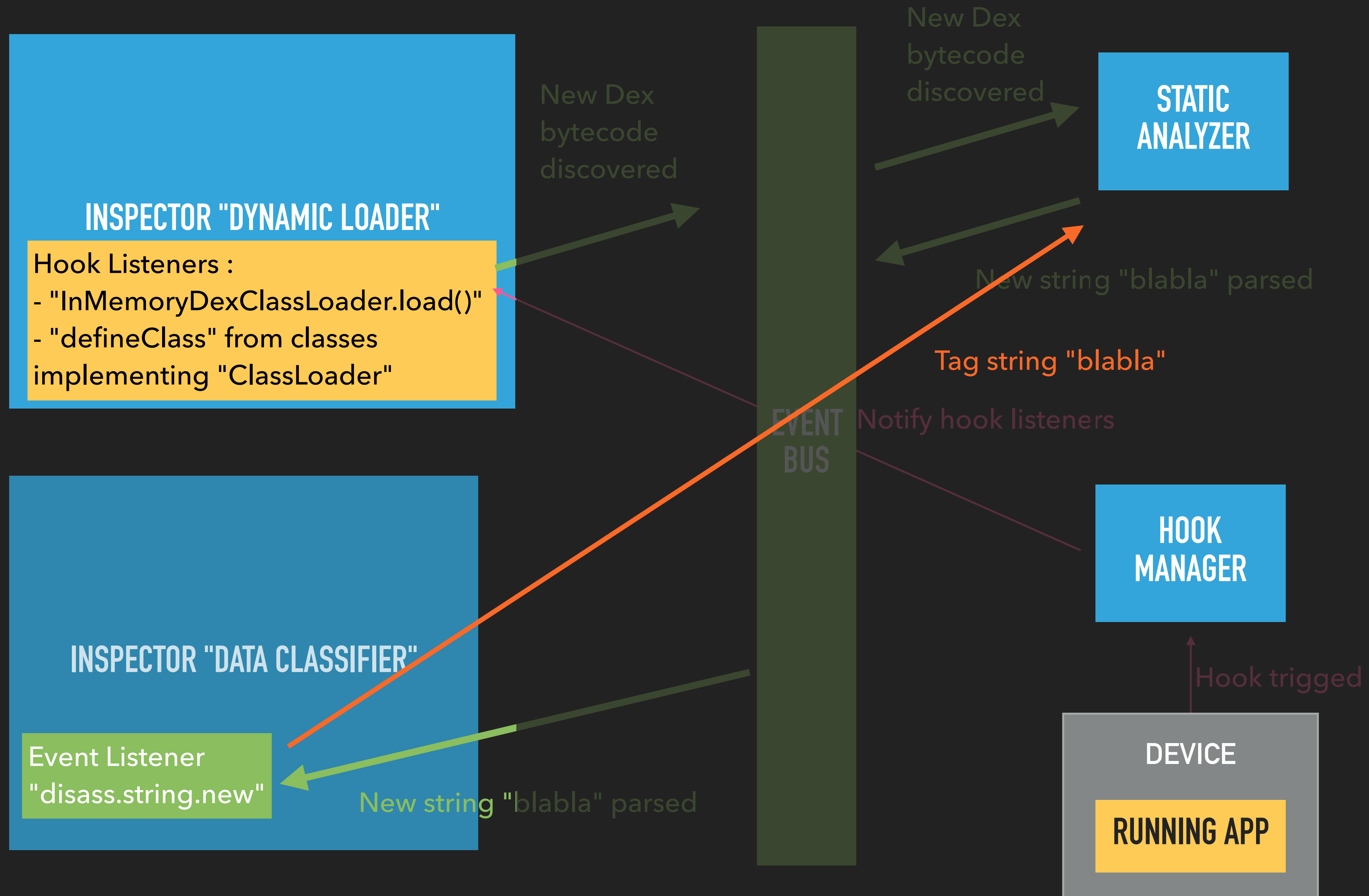
HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE



HOW IT WORKS ? - RUNTIME ANALYSIS PHASE

Search Probe all

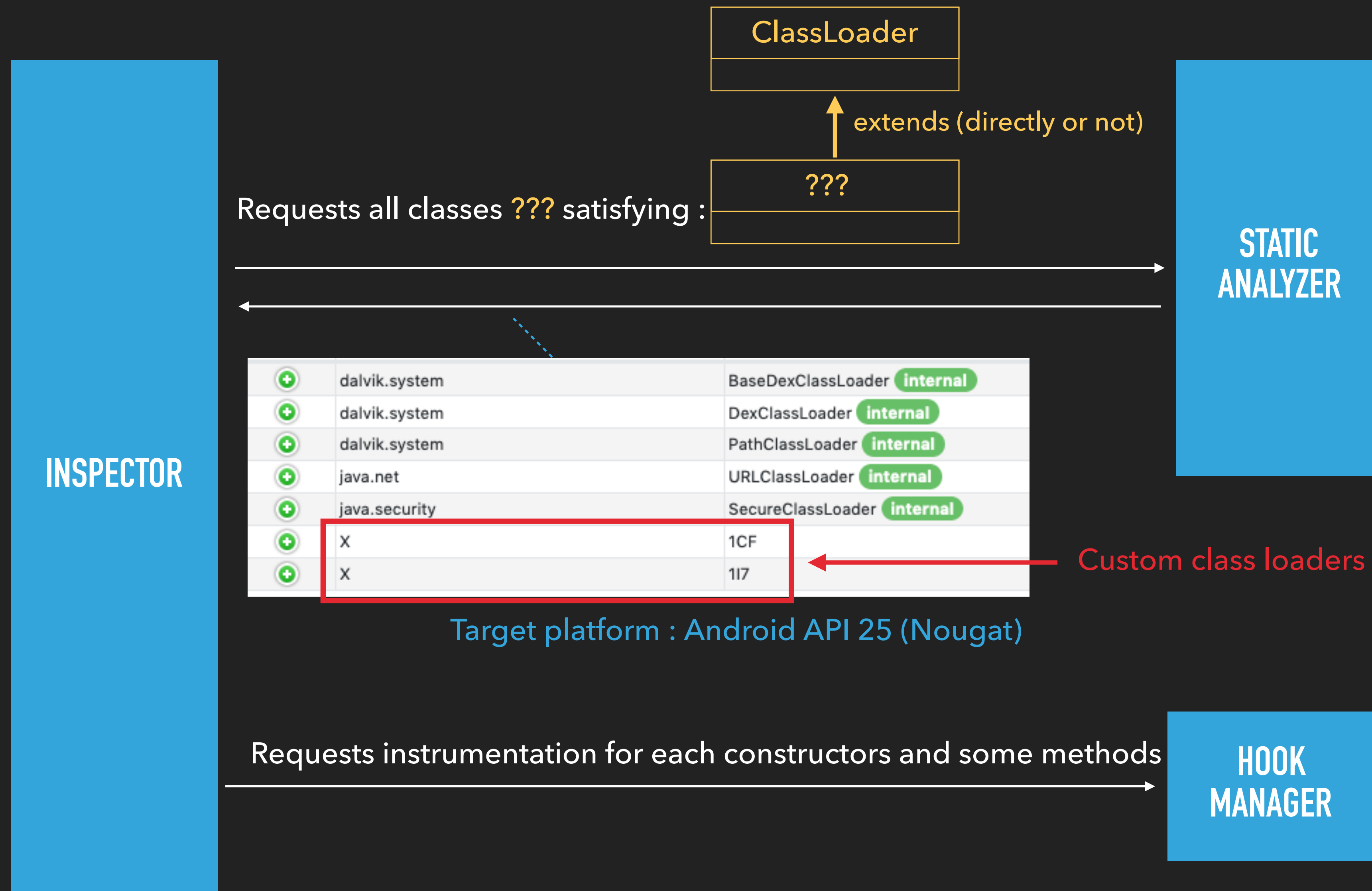
Filter : clean xref keep callers keep calleds filter by package filter by caller class Case ensensitive

↑	Value	↑	Caller	↑
	NOT TH4T PROTECT3D! D-dynamic		com.crackme.external.packed.ProtectedClass01.getFlag() <java.lang.String>	Probe

CASE #1

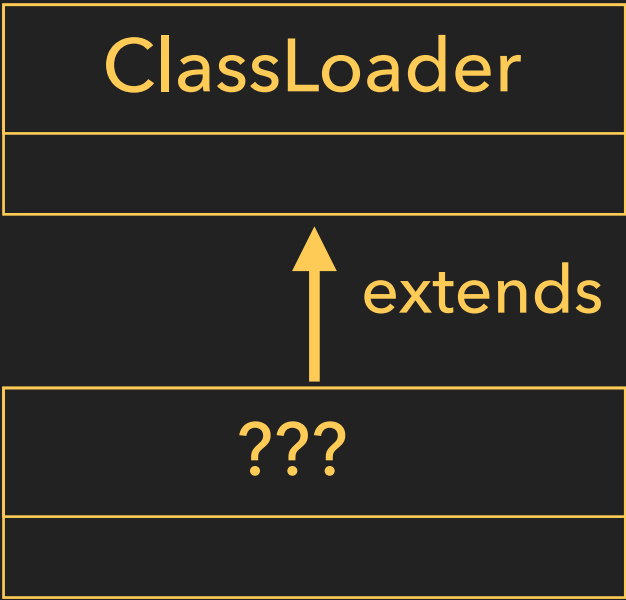
AUTOMATING ANALYSIS OF DEX/ODEX/CLASS FILE
LOADED DYNAMICALLY

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		117

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

DynamicLoader

dalvik.system.DexClassLoader.<init>(<java.lang.String><java.lang.String><java.lang.String><java.lang.ClassLoader>)<void>

Hook UUID

d16bf444a753500a537b349df441bc31

Hooked method

[dalvik.system.DexClassLoader.<init>\(<java.lang.String><java.lang.String><java.lang.String><java.lang.ClassLoader>\)<void>](#)

Description

empty

Hook code

Helpers:

Java hook

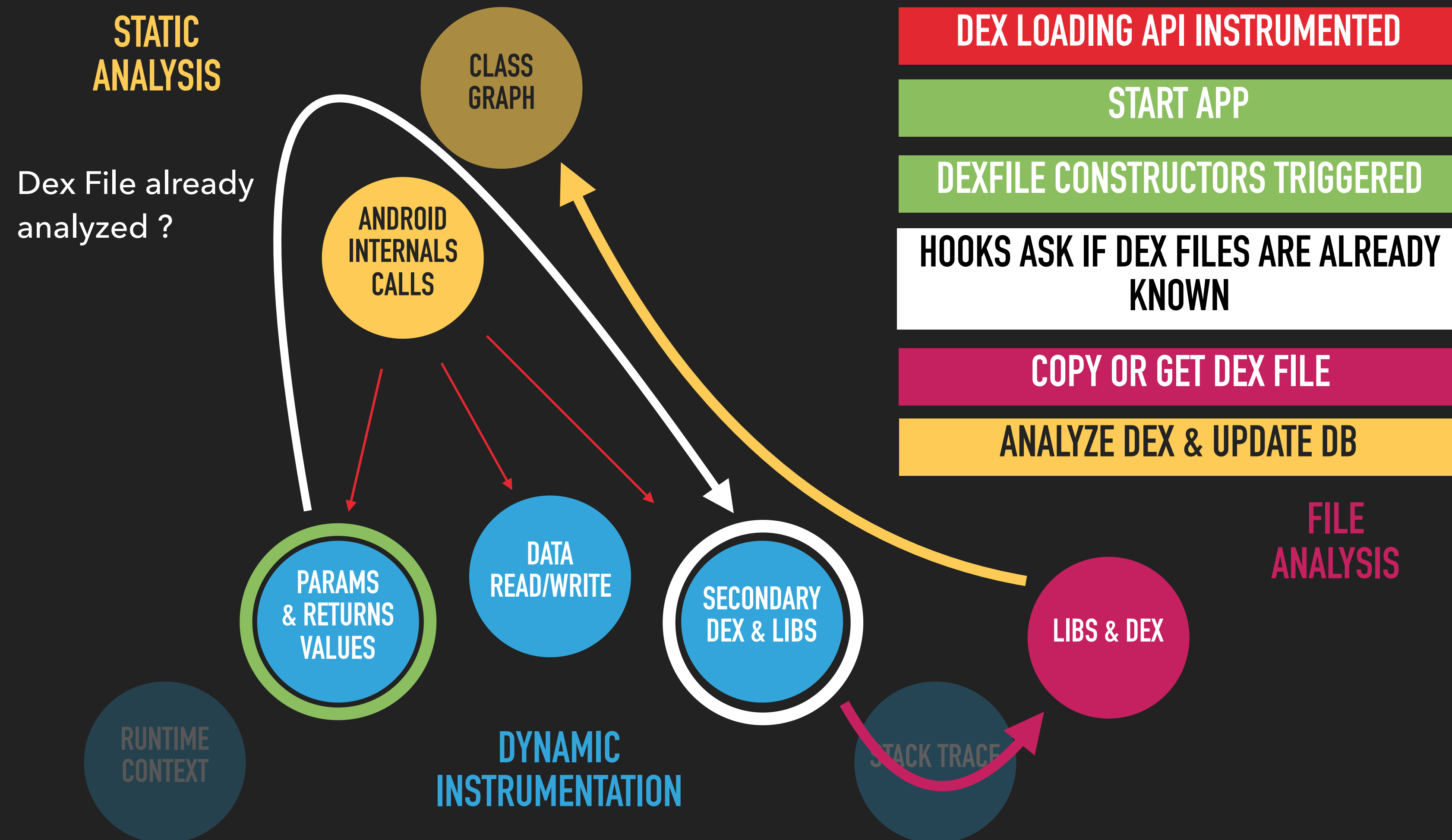
Native hook

```
1
2 var cls_44a2005c67d4b1f37e5efe9a9138ec12 = Java.use('dalvik.system.DexClassLoader');
3
4 var meth_2148a0f4f01038d96c9fe9e264a5564f = cls_44a2005c67d4b1f37e5efe9a9138ec12.$init.ov
5
6 meth_2148a0f4f01038d96c9fe9e264a5564f.implementation = function(arg0,arg1,arg2,arg3) {
7
8
9
10     send({
11         id:"ZDE2YmY0NDRhNzUzNTAwYTUzN2IzNDlkZjQ0MWJjMzE=",
12         match: true,
13         data: {
14             arg0: arguments[0],
15             arg1: arguments[1],
16             arg2: arguments[2],
17             __hidden__data: DEXC_MODULE.common.readFile(arguments[0])
18         },
19         after: true,
20         msg: "DexClassLoader.<init>()",
21         tags: [{
22             style:"purple",
23             text: "dynamic"
24         }],
25         action:"Log"
26     });
27 }
```

Hook messages

Hook data

Nothing to display



AFTER 1ST RUN

Dex file loaded dynamicallyRefresh

The table below lists all Dex files gathered at runtime and decompiled dynamically.

-	↑	Name	↕	Filepath
+		unpacked-classes01.dex		/data/user/0/com.████████.crackme.demo/files/unpacked-classes01.dex
+		classes.dex		/data/user/0/com.████████.crackme.demo/files/classes.dex

Showing 1 to 2 of 2 entries

Elements discoveredRefresh

The table below lists all elements discovered (string, class, method, field, array, ...).

-	↑	Type	↕	Object
+		Class		com.████████.crackme.external.DynamicClass01
+		Class		com.████████.crackme.external.packed.ProtectedClass01

Showing 1 to 2 of 2 entries

AFTER 1ST RUN

com.████████crackme.external.packedProtectedClass01D-dynamic

Package : com.████████crackme.external.packed (size: 1)
Extends : java.lang.Object
Implements : None

Fields

Action	Modifiers	Type	Name
--------	-----------	------	------

Methods

Action	Type	Name
Probe OFF xref to xref from		<init>()<void>
Probe OFF xref to xref from		getFlag()<java.lang.String>

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

3

4

5

.line 7

const-string v0, "N0T TH4T PR0TECT3D!"

return-object v0



(Copyright GTO, feat. Onizuka)

CASE #2

DYNAMIC UPDATE OF XREF WITH INVOKED METHODS

METHOD INVOKED DYNAMICALLY

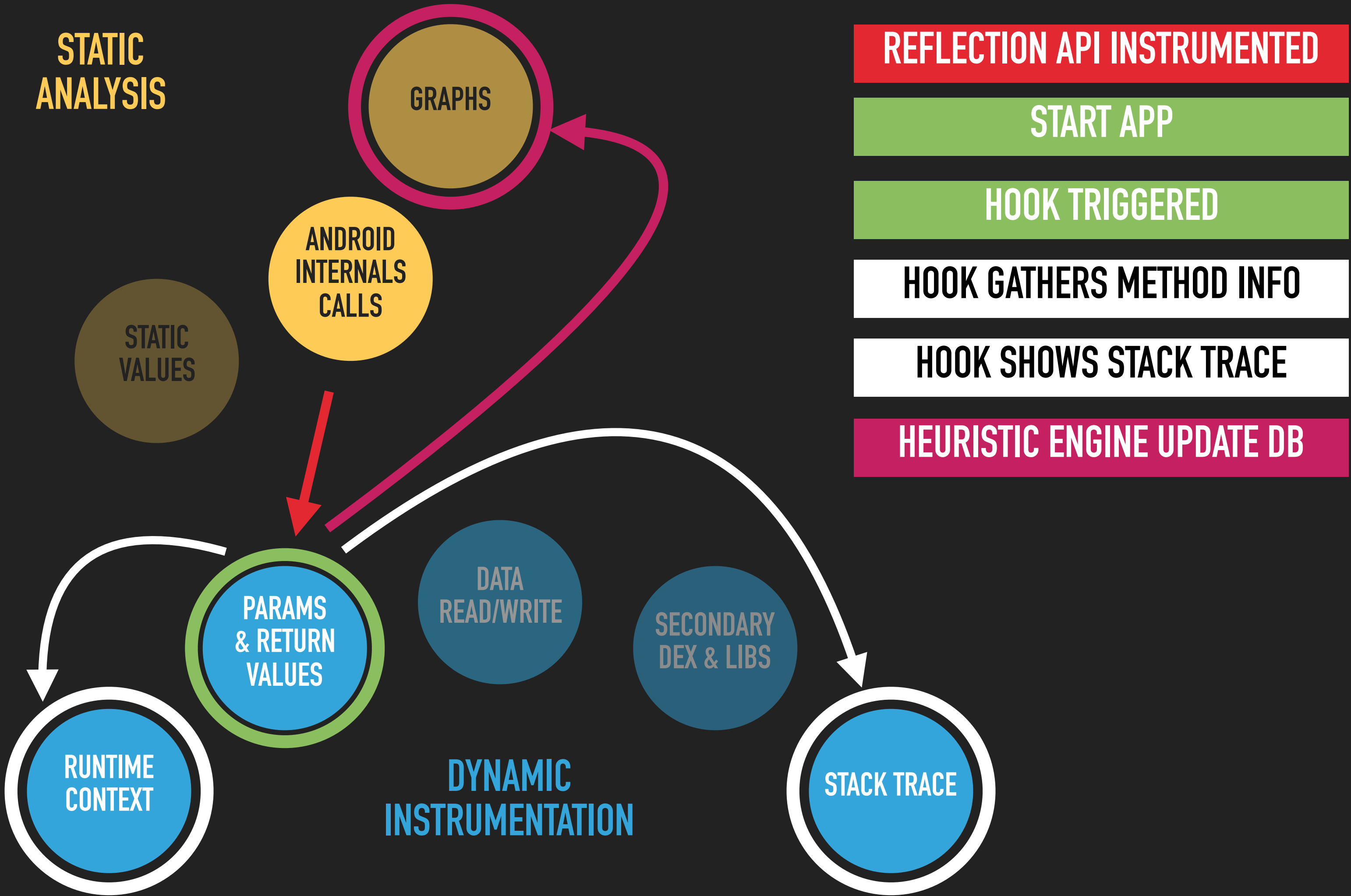
```
2  const v0, 0x1
3  new-array v1, v0, [Ljava/lang/Class;
4  new-array v2, v0, [Ljava/lang/Object;
5  const v0, 0x0
6  const-class v3, Ljava/lang/String;
7  aput-object v3, v1, v0
8  aput-object p0, v2, v0
9  const-string v0, "convertToString"
10 const-class v3, Landroid/content/res/abltMZGC;
11 invoke-virtual {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



METHOD INVOKED DYNAMICALLY

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

BEFORE
RUNTIME

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
32

```

A- REMOVING USELESS GOTO(S) – BEFORE

Smali

Run smali (VM) new

Hook history new

Configure and click Run ->

Parameters

Events

Parameters :

p0	<input checked="" type="checkbox"/> Not set	int
p1	<input checked="" type="checkbox"/> Not set	java.lang.String

Execution settings :

Callstack max depth (-1 = unlimit)	0
------------------------------------	---

VM Settings :

☐ Execute <clinit> of parent class

▶ Run

A- REMOVING USELESS GOTO(S) – AFTER

Smali

Run smali (VM) new

Hook history new

Configure and click Run ->

Parameters

Events

Parameters :

p0	<input checked="" type="checkbox"/> Not set	int
p1	<input checked="" type="checkbox"/> Not set	java.lang.String

Execution settings :

Callstack max depth (-1 = unlimit)0

VM Settings :

☐ Execute <clinit> of parent class

▶ Run

1

v0 = android.content.res.abltMZGC.bXEVAfISCtzpkh("DES/ECB/PKCS5Padding")

2

v1 = android.content.res.abltMZGC.VqxuLoJHbjvgTRW("DES")

3

v3 = android.content.res.abltMZGC.iHxCrtGAOVjZsGR(p1, "ASCII") // skipped, max depth reached

4

v2 = new javax.crypto.spec.DESKeySpec(v3) // skipped, max depth reached

5

v1 = android.content.res.abltMZGC.MqsIpHbmjWZv0QS(v1, v2) // skipped, max depth reached

6

android.content.res.abltMZGC.AoUxThBEDrGiqSb(v0, p0, v1) // skipped, max depth reached

7

return v0;

8

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

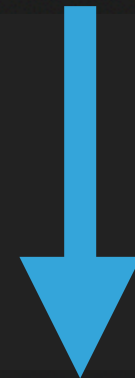
:goto_2
const-string v0, "3780b8133459f5a028d742efbcfc7d2d"

goto/32 :goto_1

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

goto/32 :goto_4
```

JADX



```
static {
    HVRQbvhaFLYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfuiuy(1);
}
```


B- HELP TO DETECT IMPLICIT EXCEPTION

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->HVRQbvhaF1YdeGD(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;->cVEqFjJdYlzfuiuy(Ljava/lang/String;)V

goto/32 :goto_4
```

Always throws AndroidVerifier exception
at runtime

JADX

```
static {
    HVRQbvhaF1YdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfuiuy(1);
}
```

WRONG

B- HELP TO DETECT IMPLICIT EXCEPTION

```
:goto_1
invoke-static/range {v0 .. v0}, Lcom/eshard/crackme/activities/CrackMeChallenge;->HVRQbvhaFIYdeGD(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;->cVEqFjJdYlzfuiy(Ljava/lang/String;)V
goto/32 :goto_4
```

Always throws AndroidVerifier exception
at runtime

JADX

```
static {
    HVRQbvhaFIYdeGD("3780b8133459f5a028d742efbcfc7d2d");
    cVEqFjJdYlzfuiy(1);
}
```

WRONG

TRUE PATH

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

END

AND FINALLY ... HOW TO INSTALL DEXCALIBUR ?

FROM **NPM** ...

```
$ npm install dexcalibur
```

```
$ dexcalibur
```

WITH USER-FRIENDLY UI AND AUTOMATIC INSTALL OF DEPENDENCIES:

The image displays two overlapping screenshots of the DEXCALIBUR application interface. The background screenshot shows the 'DEXCALIBUR Configuration' window with a dark header and a yellow sidebar. A dark blue message box at the top states: 'Dexcalibur has been successfully installed. However, you need to configure external tools and to initialize workspace before to start.' Below this, the configuration form includes: 'File encoding (required)' set to 'UTF-8'; 'Workspace path (required)' set to '/Users/myhome/dexcaliburWS' with a red error message 'Folder not found. It will be created automatically !'; 'Default port number (required)' set to '8000' with a green checkmark; and 'External tools' with the 'Auto install (recommended)' option selected. A blue 'Next' button is at the bottom. The foreground screenshot shows the 'DEXCALIBUR Configuration' window with the same header, but the configuration form is replaced by an 'Install' section. It contains the same success message and a progress bar for downloading Android platform tools from 'https://dl.google.com/android/repository/platform-tools_r29.0.6-darwin.zip ...'.

Thanks



(Copyright GTO, feat. Onizuka)

Q&A