Take a picture of your app code - Android MRI interpreter

SungHyoun Song
(@decash, decash@fsec.or.kr)

Principal Researcher,
FSI (Financial Security Institute)

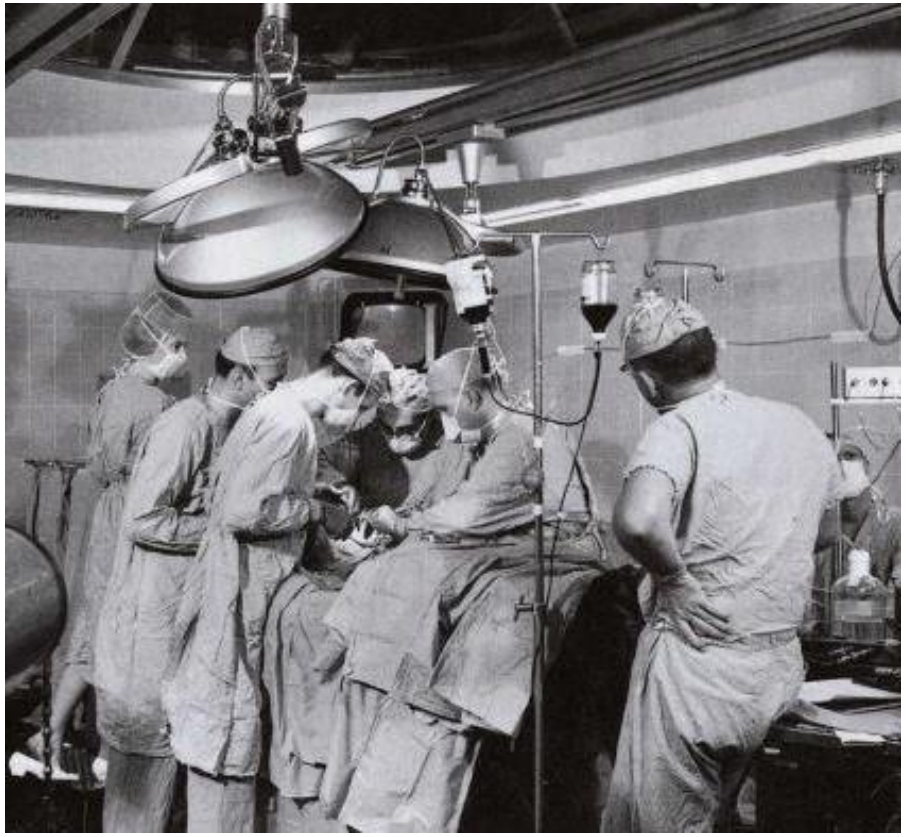
WHO AM I

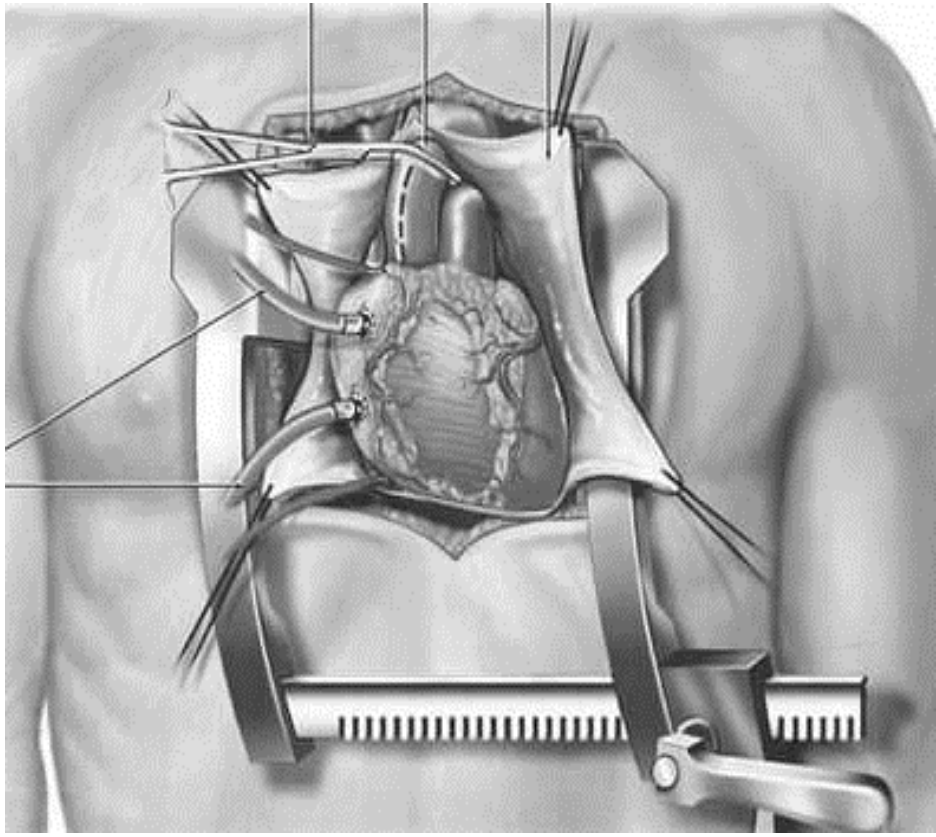
- **SungHyoun Song** (@decashx)
- **Principal Researcher at FSI**(Financial Security Institute) of South Korea
  - Mobile security researcher
  - Penetration tester for the Korean financial industry
- **Speaker of** { ITU-T / BlackHat USA / BlackHat ASIA / Ekoparty / Nullcon / HITB / CanSecWest / SEC-T / PacSec / HITCON / beVX / BlackAlps }
- **Interested in Android OS and Linux Kernel**

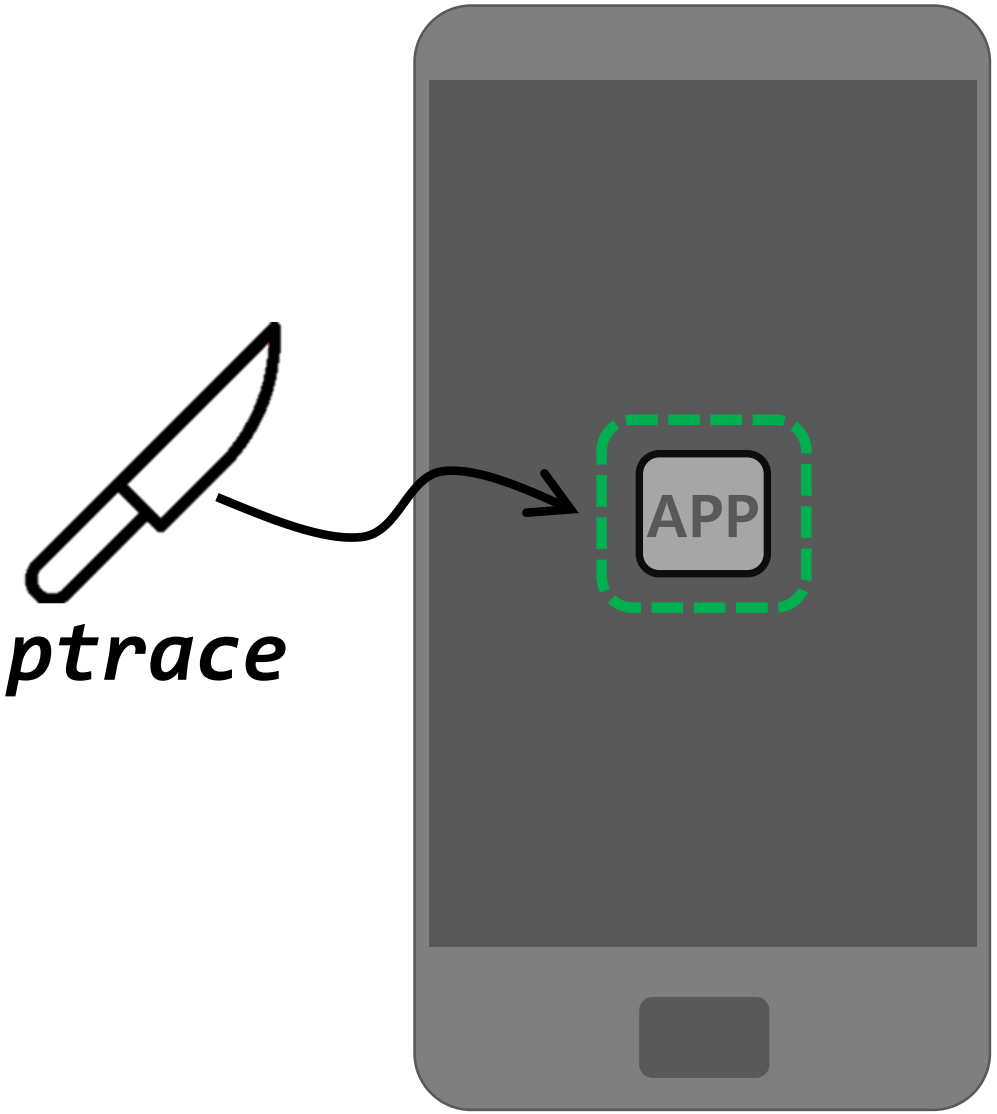
Surgery in the 1600's

Pieter van Miereveld, 1617, https://en.wikipedia.org/wiki/Pieter_van_Mierevelt

X-Rays





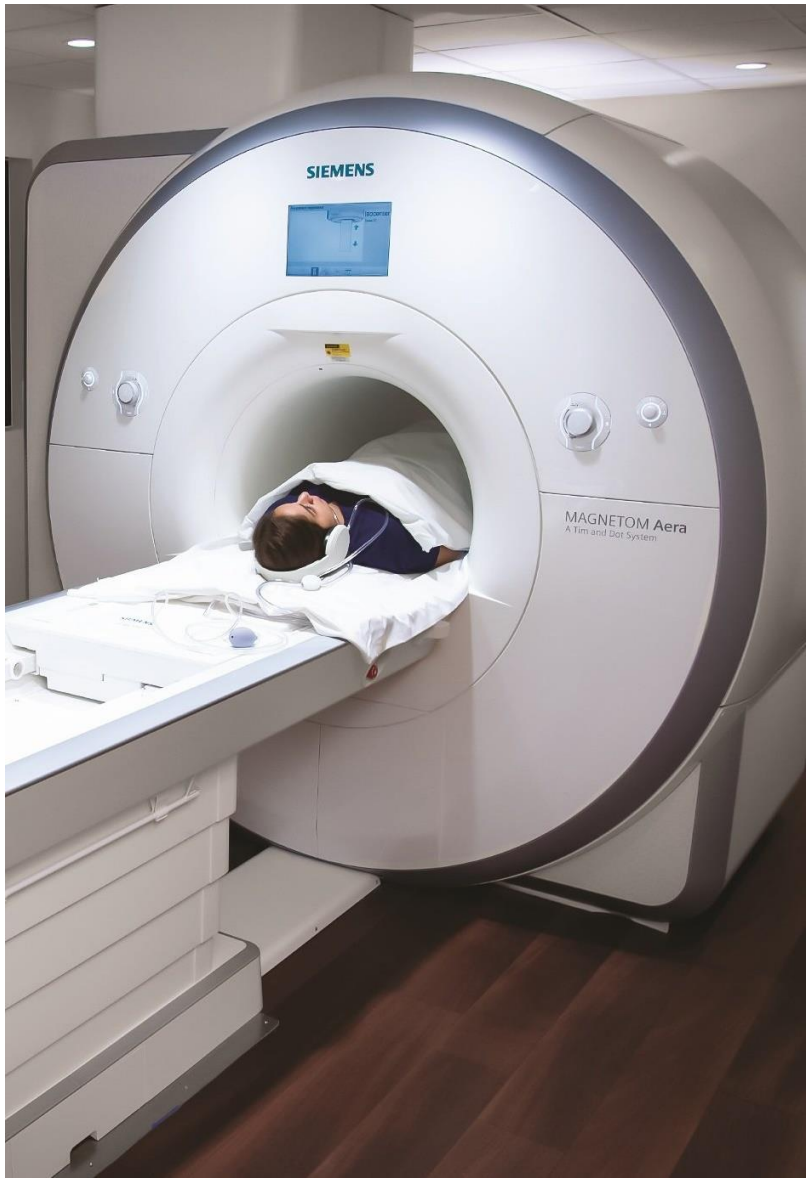


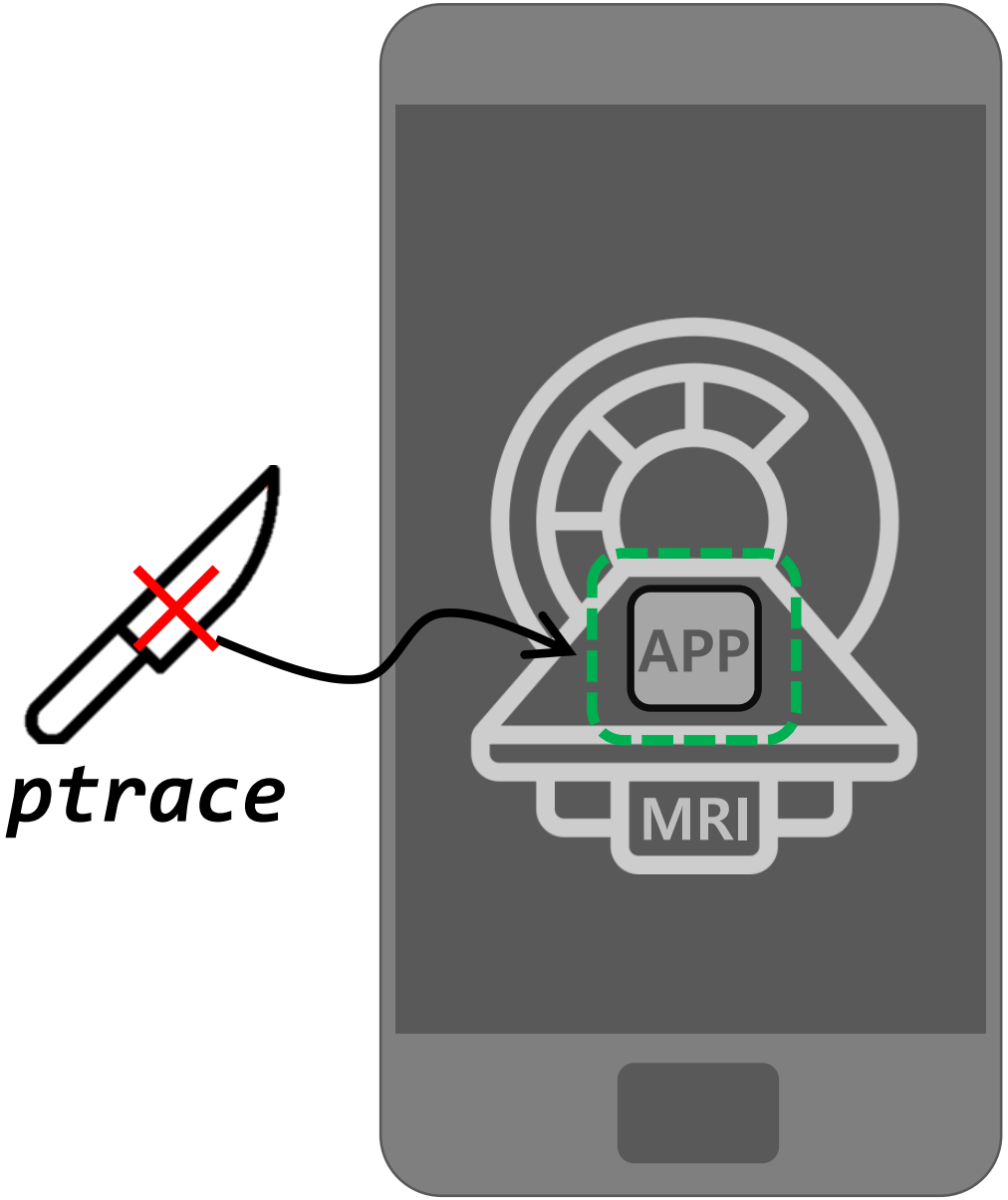
Runtime Application Self-Protection (RASP)

OS Integrity Check
Code Obfuscation
Device Binding
Anti-Emulator
Anti-Debugging
Data Encryption
Anti-Tampering
Secure Communication
Anti-Keylogger
Anti-Hooking

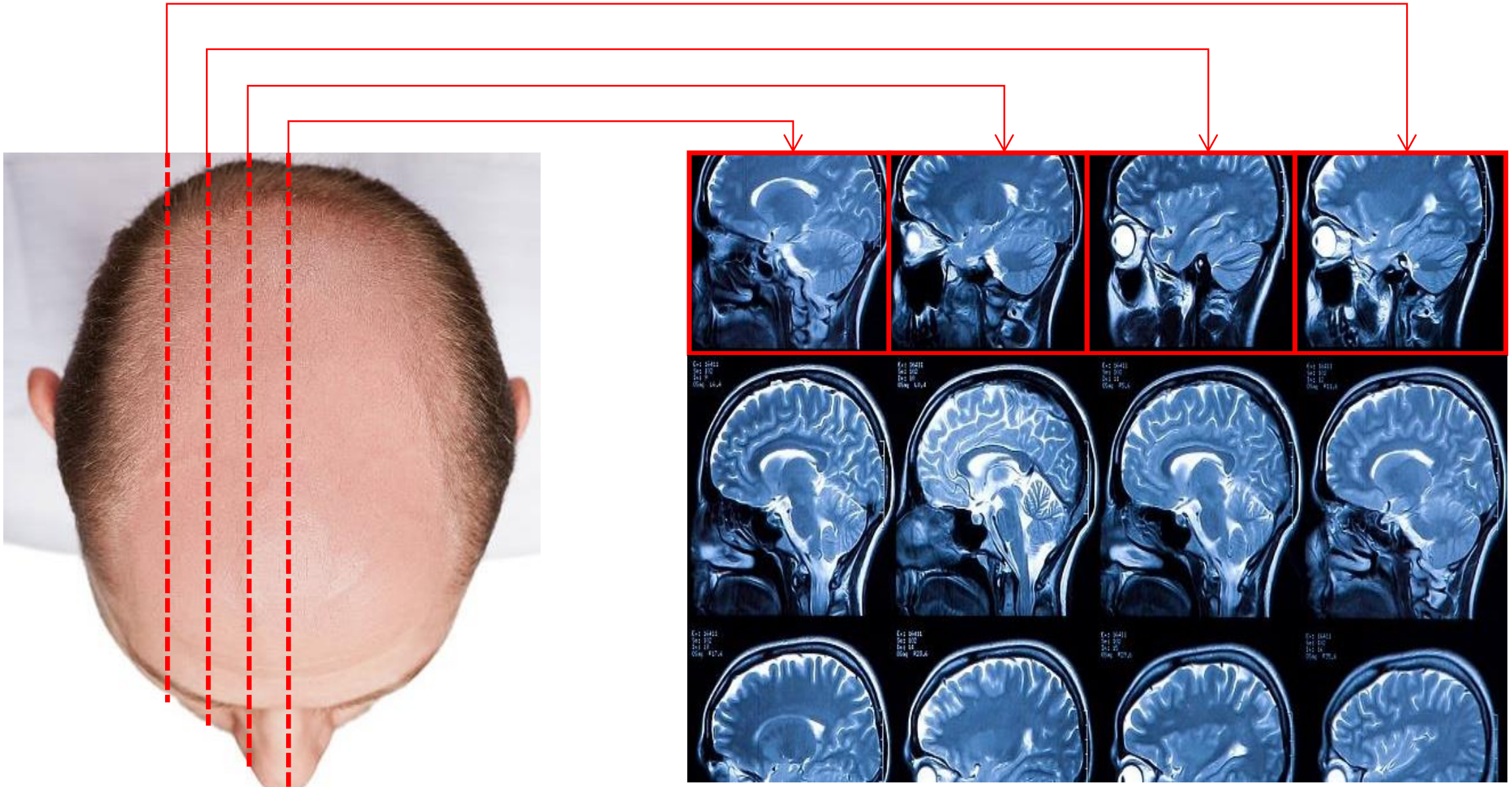
MRI

Android MRI in 2023

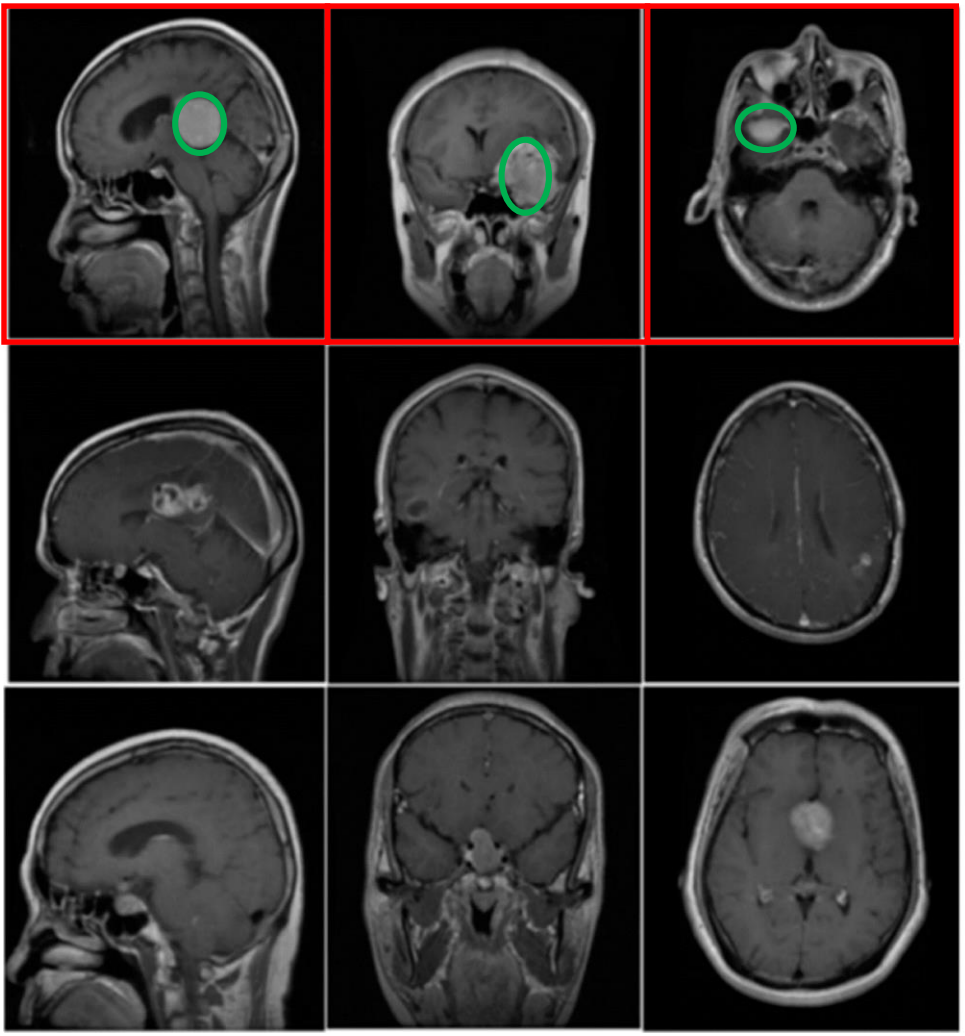




Android MRI in 2023



Android MRI in 2023



boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1a: ***invoke-direct {v0}, void java.lang.StringBuilder.<init>()***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x00000000 vreg2=0x00000000 vreg3=0x133492A0/
com.ai.obf.am vreg4=0x1334F6C0/string "1qw3e4r!@" vreg5=0x13347F30/string "570436"

boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1d: ***const-string v1, ".X G,E(g$Ym\nm"***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x00000000 vreg2=0x00000000 vreg3=0x133492A0/
com.ai.obf.am vreg4=0x13347F30/string "570436" vreg5=0x1334F6C0/string "1qw3e4r!@"

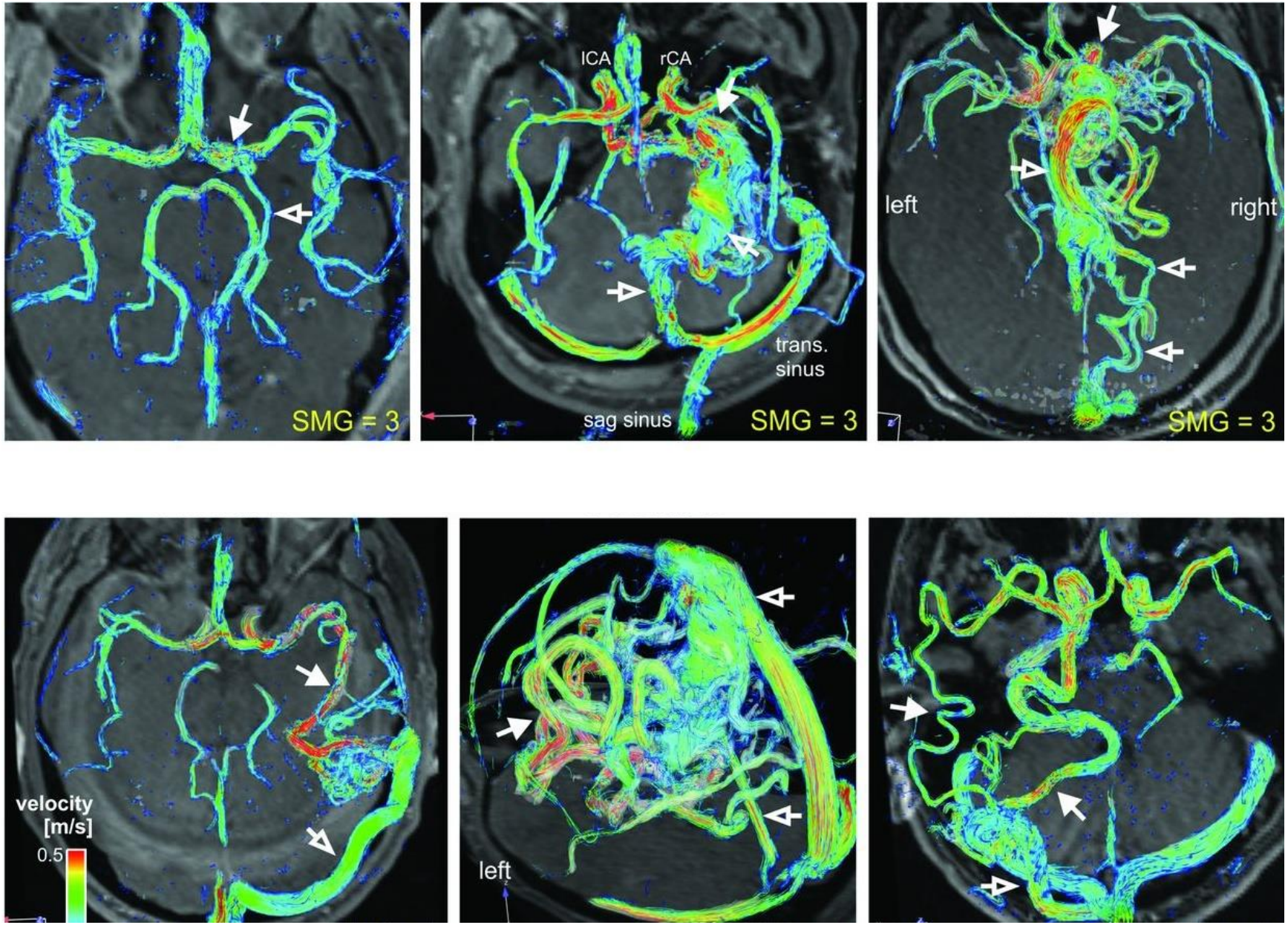
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1f: ***invoke-static {v1}, java.lang.String mip.mia.data.model.verify.DataWrap.G(java.lang.Object)***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1= 0x1334F6C0/string "1qw3e4r!@" vreg2=0x000000
vreg3=0x133492A0/com.ai.obf.am vreg4=0x1334F718/string ".X G,E(g$Ymm" vreg5=0x13347F30/strin

boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x22: ***move-result-object v1***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x1334F718/string ".X G,E(g$Ymm" vreg2=0x0000
vreg3=0x133492A0/com.ai.obf.am vreg4=0x1334F6C0/string "1qw3e4r!@" vreg5=0x13347F30/string "

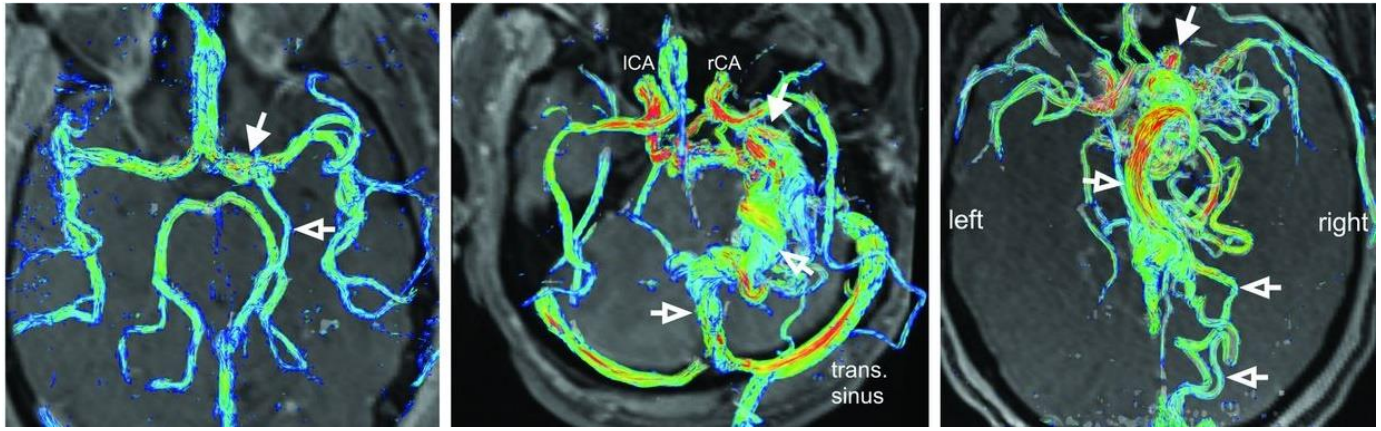
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x23: ***const/4 v2, #+0***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x1334F770/string "comparePin = " vreg2=0x000
vreg3=0x133492A0/com.ai.obf.am vreg4=0x1334F6C0/string "1qw3e4r!@" vreg5=0x13347F30/string "

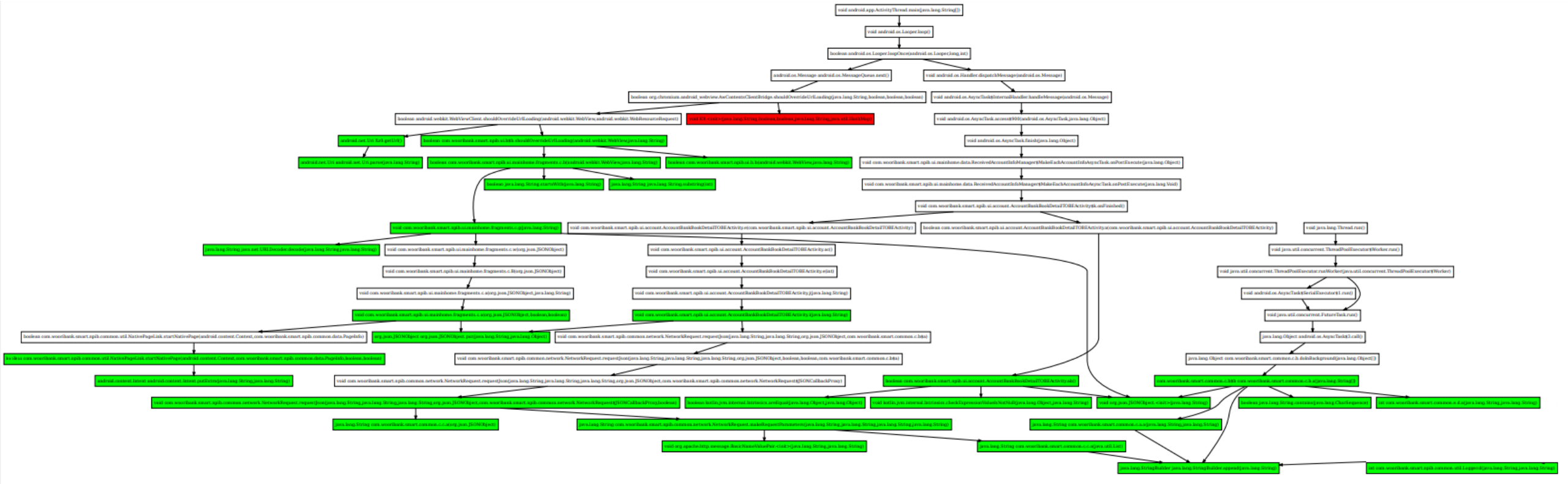
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x24: ***invoke-virtual {v0, v2, v1}, java.lang.StringBuilder java.lang.StringBuilder.insert(int,java.lang.String)***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x1334F770/string "comparePin = " vreg2=0x000
vreg3=0x133492A0/com.ai.obf.am vreg4=0x1334F6C0/string "1qw3e4r!@" vreg5=0x13347F30/string "

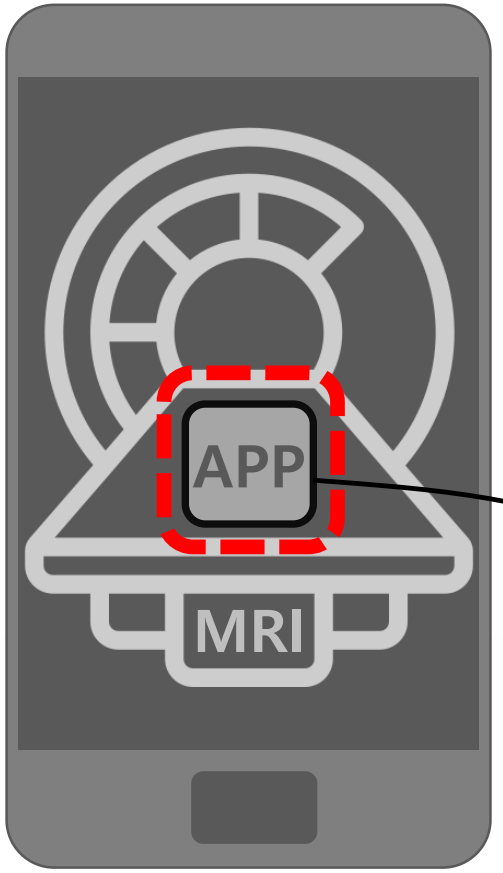
Android MRI in 2023



Android MRI in 2023







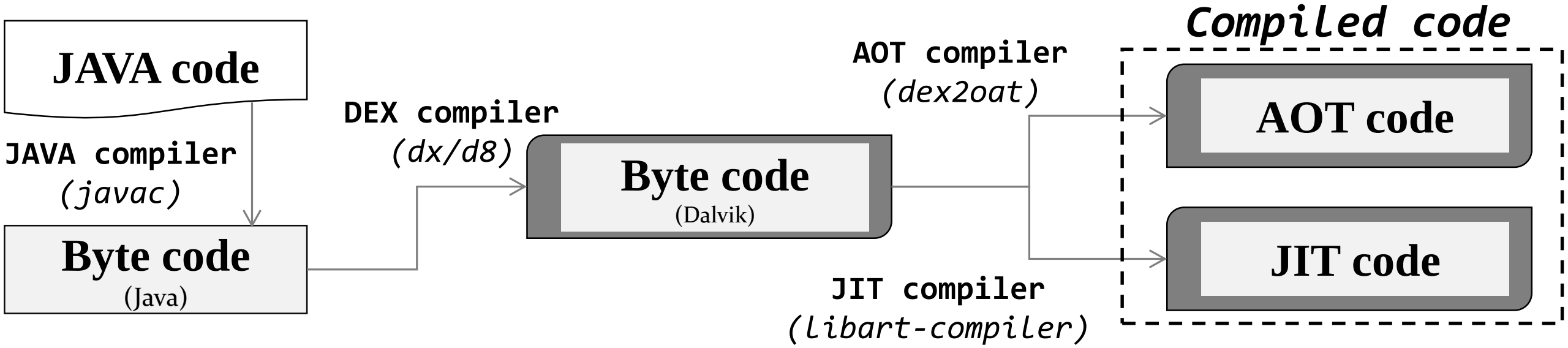
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1a: ***invoke-direct {v0}, void java.lang.StringBuilder.<init>()***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x00000000 vreg2=0x00000000 vreg3=0x133492A0/
com.ai.obf.am vreg4=0x1334F6C0/string "1qw3e4r!@" vreg5=0x13347F30/string "570436"

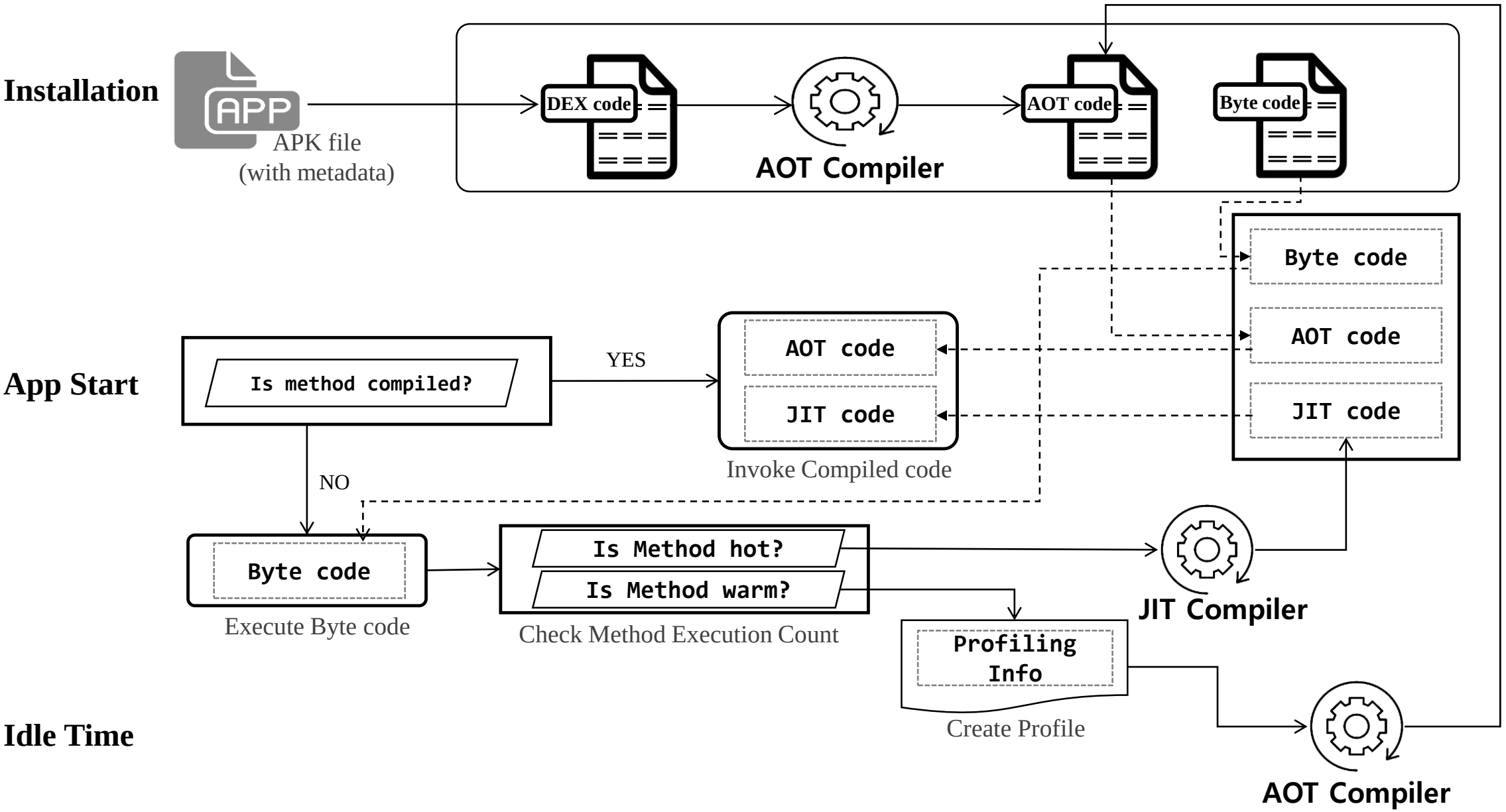
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1d: ***const-string v1, ".X G,E(g$Ym\nm"***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1=0x00000000 vreg2=0x00000000 vreg3=0x133492A0/
com.ai.obf.am vreg4=0x13347F30/string "570436" vreg5=0x1334F6C0/string "1qw3e4r!@"

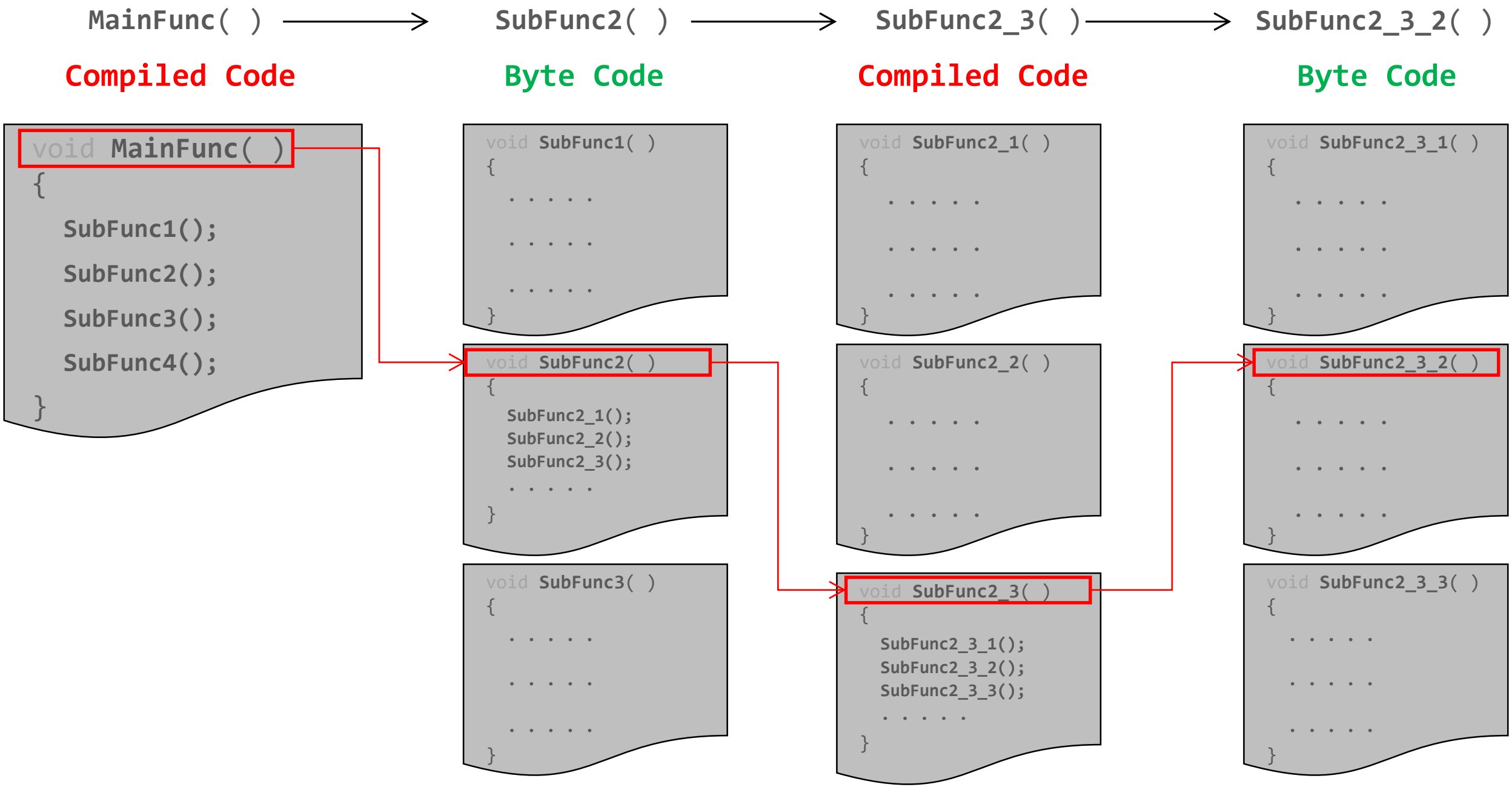
boolean com.ai.obf.am.G(android.content.Context,java.lang.String)
0x1f: ***invoke-static {v1}, java.lang.String mip.mia.data.model.verify.DataWrap.G(java.lang.Object)***
vreg0=0x1334F6D8/java.lang.StringBuilder vreg1= 0x1334F6C0/string "1qw3e4r!@" vreg2=0x00000000
vreg3=0x133492A0/com.ai.obf.am vreg4=0x1334F718/string ".X G,E(g$Ymm" vreg5=0x13347F30/string "570436"

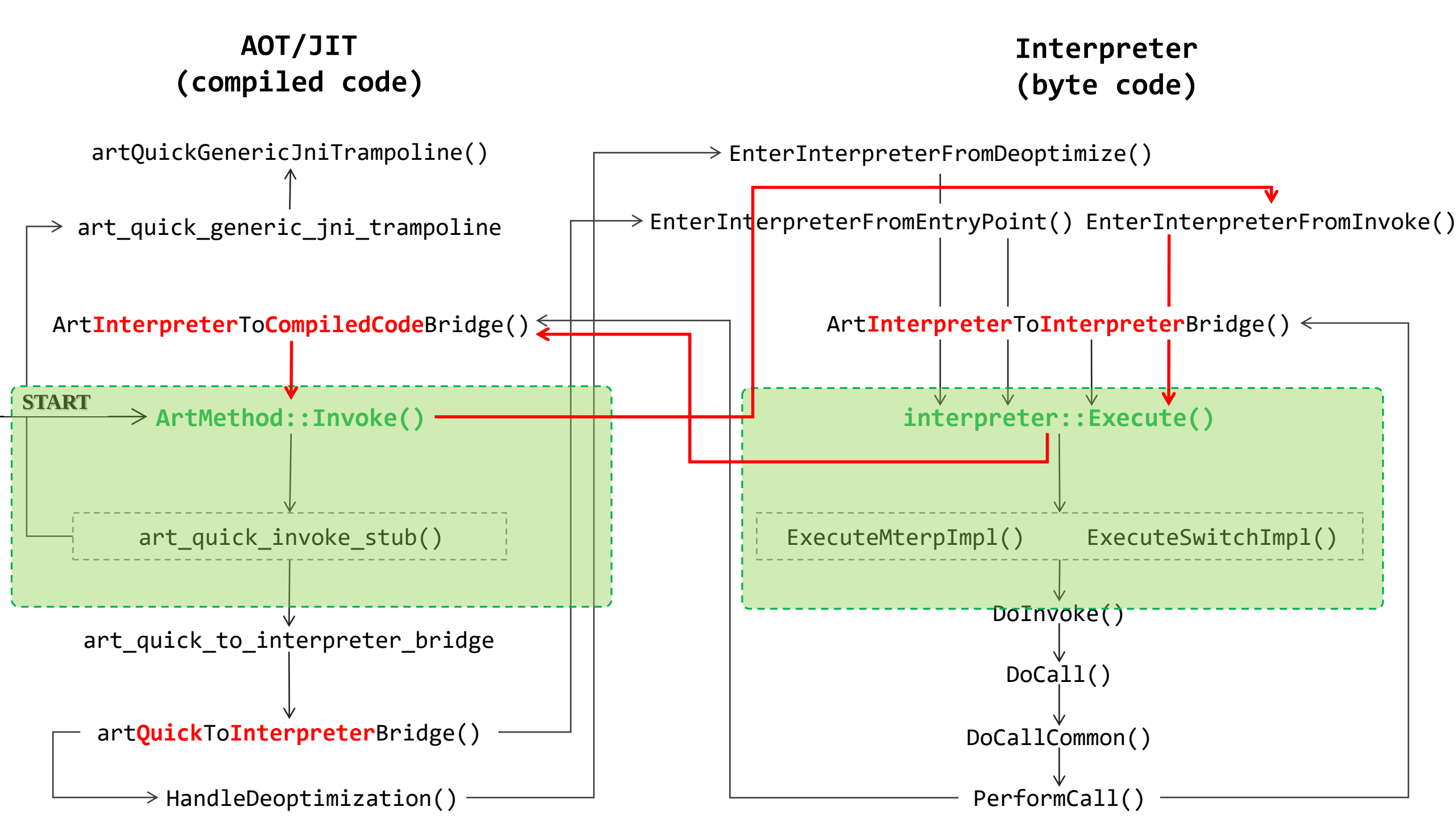
Design of Android MRI interpreter

File type of ART




Installation
APP
APK file
(with metadata)
DEX code
AOT Compiler
AOT code
Byte code
Byte code
AOT code
JIT code
App Start
Is method compiled?
YES
AOT code
JIT code
Invoke Compiled code
NO
Byte code
Execute Byte code
Is Method hot?
Is Method warm?
Check Method Execution Count
JIT Compiler
Profiling Info
Create Profile
Idle Time
AOT Compiler

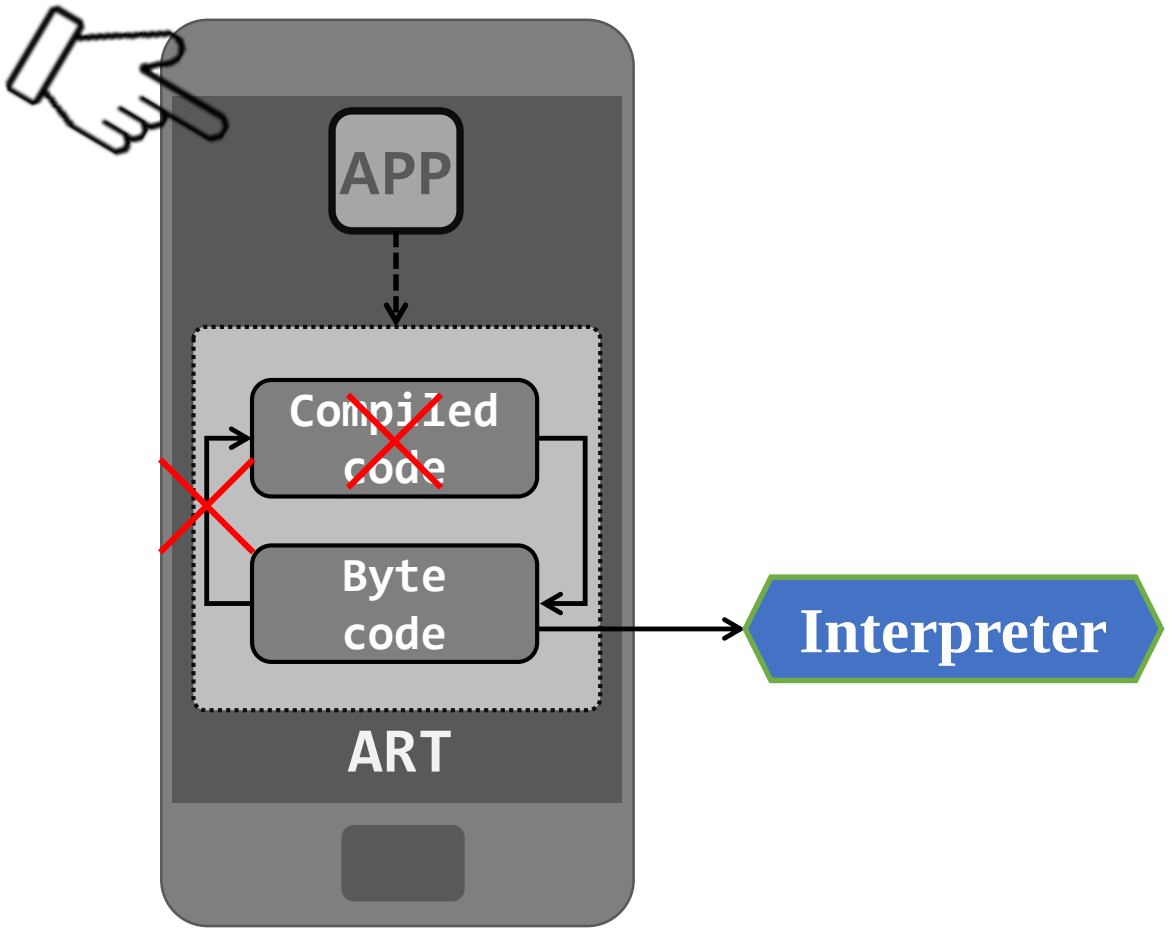

MainFunc( )
SubFunc2( )
SubFunc2_3( )
SubFunc2_3_2( )
Compiled Code
Byte Code
Compiled Code
Byte Code
void MainFunc( )
{
SubFunc1();
SubFunc2();
SubFunc3();
SubFunc4();
}
void SubFunc1( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2( )
{
SubFunc2_1();
SubFunc2_2();
SubFunc2_3();
. . . . .
}
void SubFunc3( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2_1( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2_2( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2_3( )
{
SubFunc2_3_1();
SubFunc2_3_2();
SubFunc2_3_3();
. . . . .
}
void SubFunc2_3_1( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2_3_2( )
{
. . . . .
. . . . .
. . . . .
}
void SubFunc2_3_3( )
{
. . . . .
. . . . .
. . . . .
}


AOT/JIT
(compiled code)
Interpreter
(byte code)
artQuickGenericJniTrampoline()
art_quick_generic_jni_trampoline
EnterInterpreterFromDeoptimize()
EnterInterpreterFromEntryPoint()
EnterInterpreterFromInvoke()
ArtInterpreterToCompiledCodeBridge()
ArtInterpreterToInterpreterBridge()
START
ArtMethod::Invoke()
interpreter::Execute()
art_quick_invoke_stub()
ExecuteMterpImpl()
ExecuteSwitchImpl()
art_quick_to_interpreter_bridge
DoInvoke()
artQuickToInterpreterBridge()
DoCall()
DoCallCommon()
HandleDeoptimization()
PerformCall()

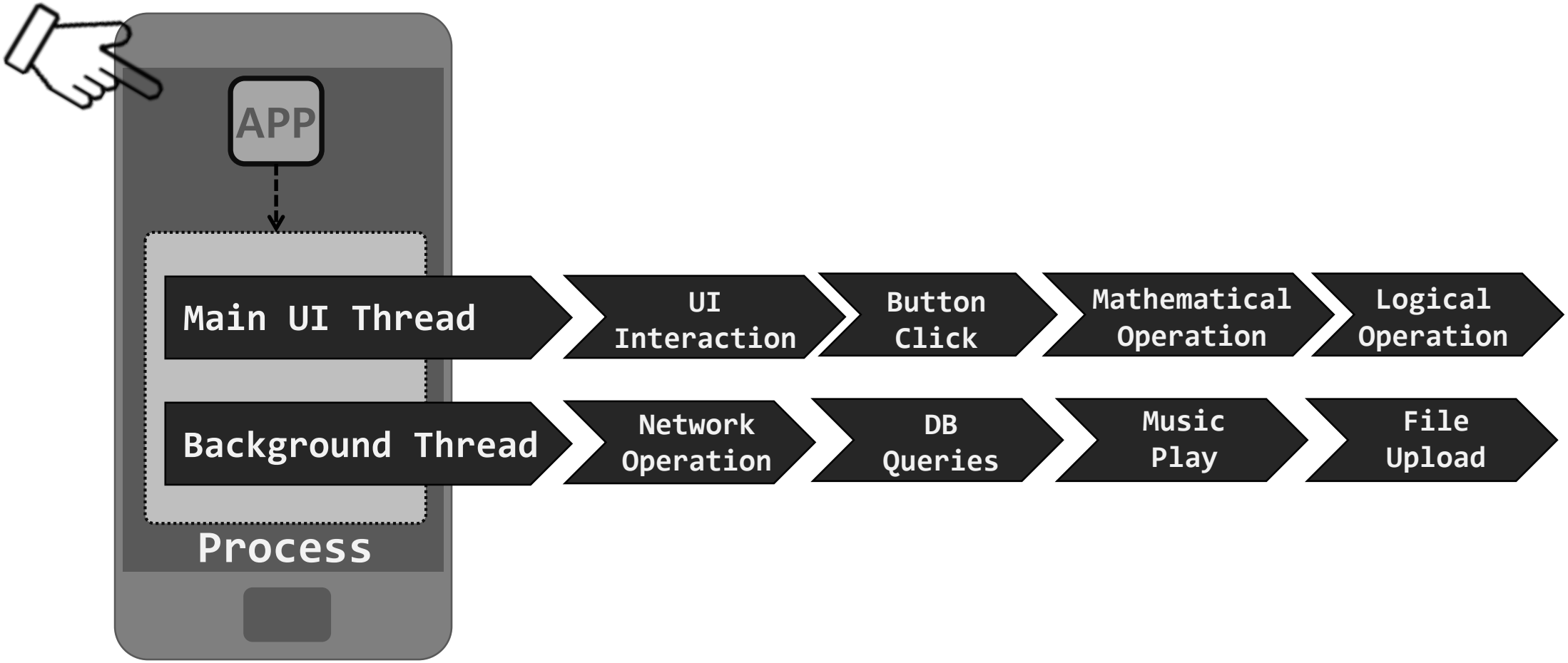
adb shell oatdump --oat-file=/data/app/com.myandroid.app/oat/arm4/base.odex --output=/tmp/dump.txt

```
297: Lcom/appsflyer/internal/ac; (offset=0x0006ff38) (type_idx=1233) (Initialized) (OatClassSomeCompiled)
  0: void com.appsflyer.internal.ac.<init>() (dex_method_idx=7531)
    DEX CODE:
     0x0000: 7010 f95b 0100 | invoke-direct {v1}, void java.lang.Object.<init>() // method@23545
     0x0003: 1200           | const/4 v0, #+0
     0x0004: 5c10 a314      | iput-boolean v0, v1, Z com.appsflyer.internal.ac.ˏ // field@5283
     0x0006: 1210           | const/4 v0, #+1
     0x0007: 5c10 a414      | iput-boolean v0, v1, Z com.appsflyer.internal.ac.ˑ // field@5284
     0x0009: 1200           | const/4 v0, #+0
     0x000a: 5b10 a214      | iput-object v0, v1, Lcom/appsflyer/internal/ac$b; com.appsflyer.internal.ac.ˋ
     0x000c: 7300           | return-void-no-barrier
    ....
    CODE: (code_offset=0x0041bcd0 size_offset=0x0041bccc size=20)...
     0x0041bcd0: 52800020 mov w0, #0x1
     0x0041bcd4: 3900303f strb wzr, [x1, #12]
     0x0041bcd8: 39003420 strb w0, [x1, #13]
     0x0041bcdc: b900083f str wzr, [x1, #8]
     0x0041bce0: d65f03c0 ret
    ....
326: Lcom/appsflyer/internal/t$1; (offset=0x00070148) (type_idx=1265) (Initialized) (OatClassNoneCompiled)
  0: void com.appsflyer.internal.t$1.<init>(com.appsflyer.internal.t) (dex_method_idx=7673)
    DEX CODE:
     0x0000: 5b01 2815      | iput-object v1, v0, Lcom/appsflyer/internal/t; com.appsflyer.internal.t$1.ˊ
     0x0002: 7010 f95b 0000 | invoke-direct {v0}, void java.lang.Object.<init>() // method@23545
     0x0005: 7300           | return-void-no-barrier
    ....
    CODE: (code_offset=0x003447a0 size_offset=0x0034479c size=24)...
     NO CODE!
```

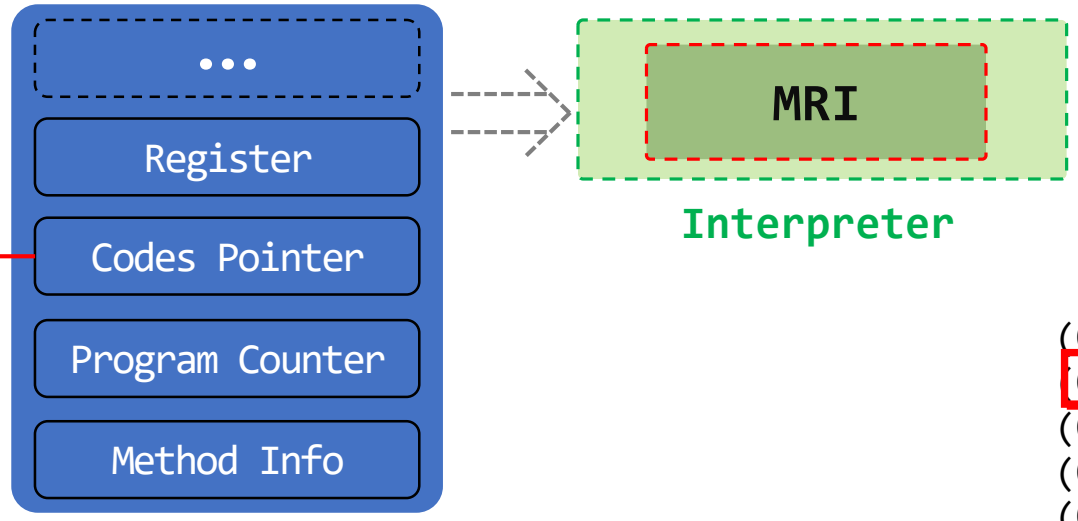
Trace dalvik instruction and register value (in Interpreter)



Trace dalvik instruction and register value (in Interpreter)

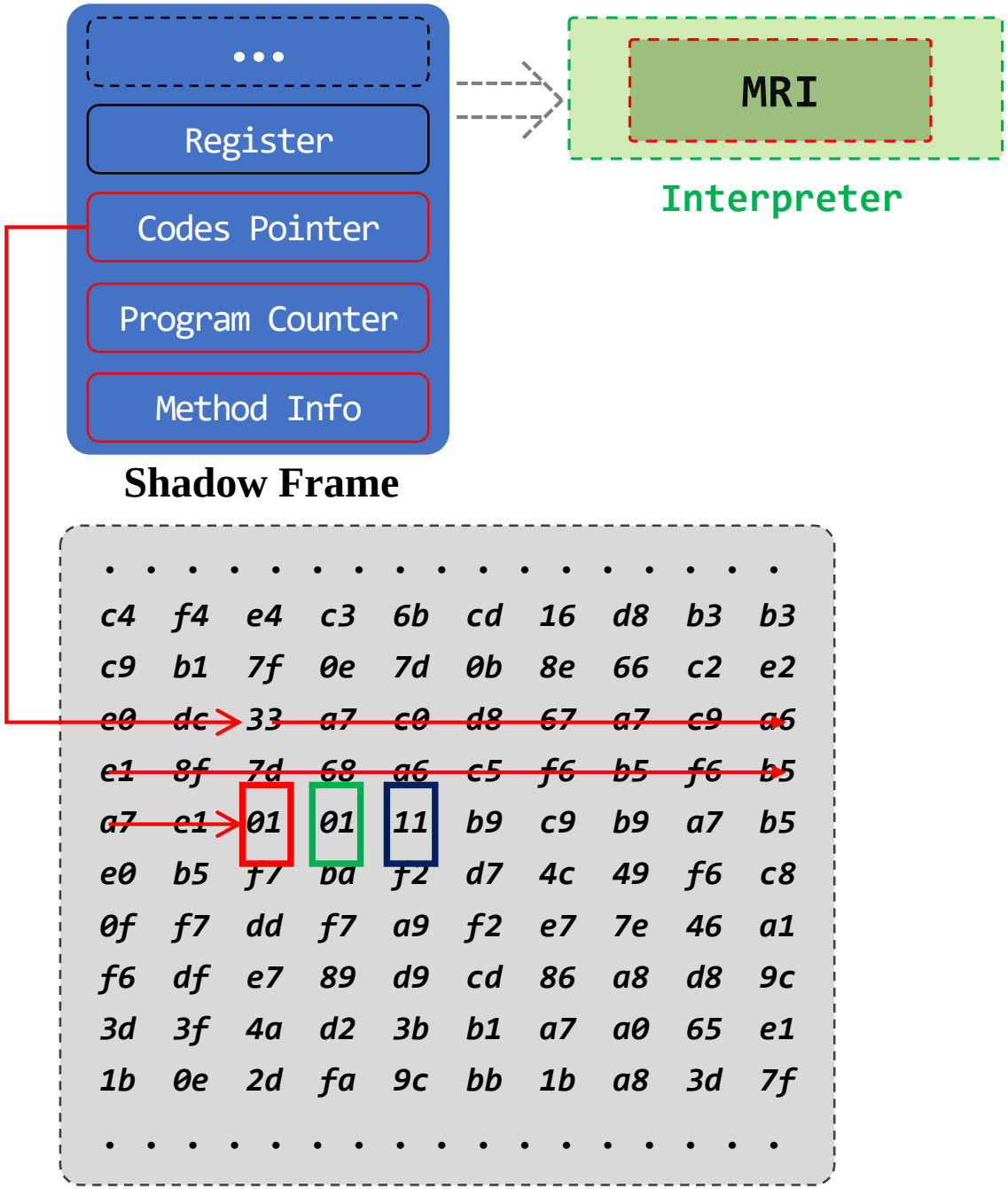


https://www.compuhoy.com/what-is-main-thread-and-background-thread-in-android/



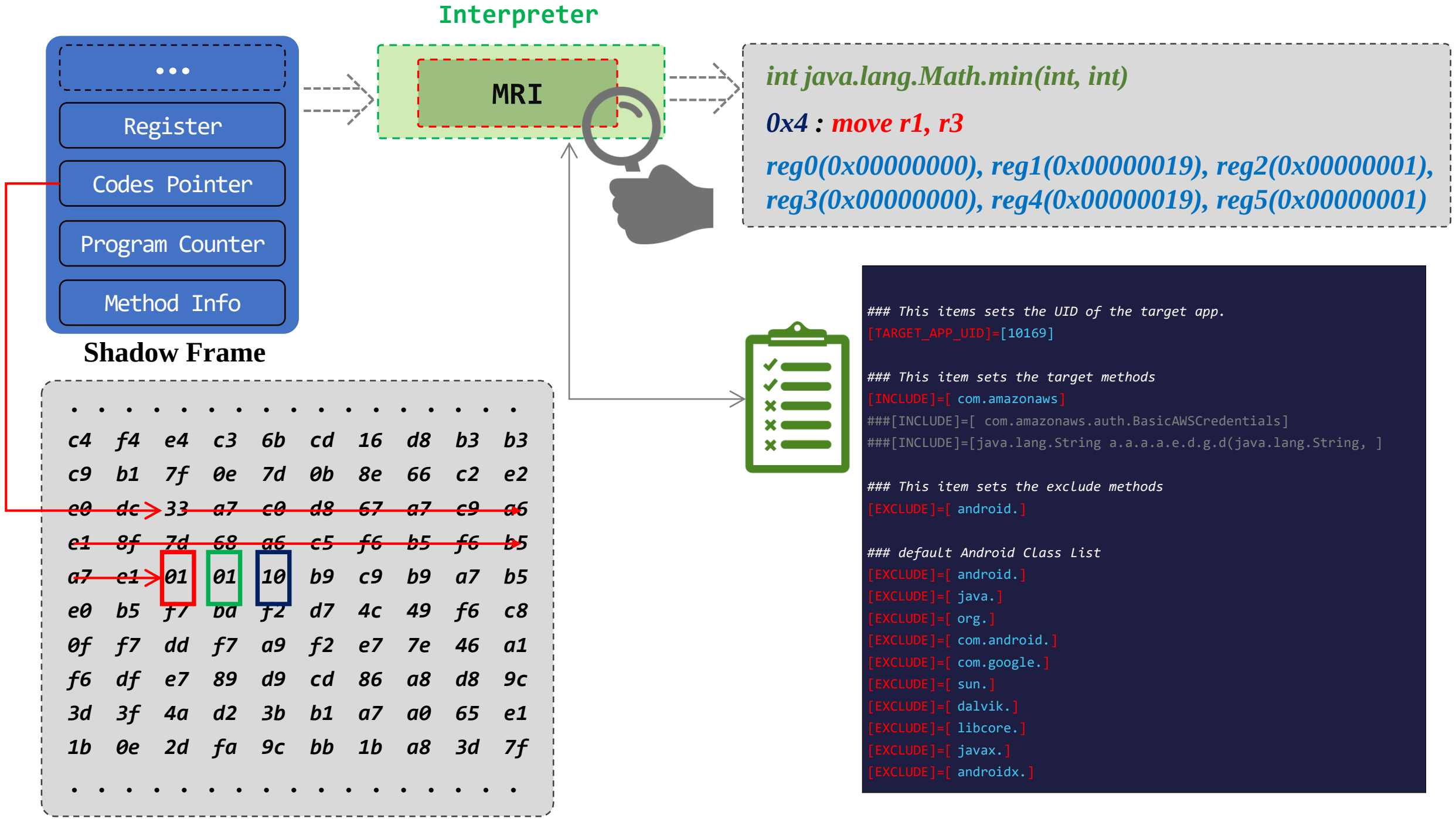
DEX INSTRUCTION LIST

```
(0x00, NOP,                  "nop",                 k10x, ...)
(0x01, MOVE,                 "move",                k12x, ...)
(0x02, MOVE_FROM16,          "move/from16",         k22x, ...)
(0x03, MOVE_16,              "move/16",             k32x, ...)
(0x04, MOVE_WIDE,            "move-wide",           k12x, ...)
(0x05, MOVE_WIDE_FROM16,     "move-wide/from16",    k22x, ...)
(0x06, MOVE_WIDE_16,         "move-wide/16",        k32x, ...)
(0x07, MOVE_OBJECT,          "move-object",         k12x, ...)
(0x08, MOVE_OBJECT_FROM16,   "move-object/from16",  k22x, ...)
(0x09, MOVE_OBJECT_16,       "move-object/16",      k32x, ...)
(0x0A, MOVE_RESULT,          "move-result",         k11x, ...)
(0x0B, MOVE_RESULT_WIDE,     "move-result-wide",    k11x, ...)
(0x0C, MOVE_RESULT_OBJECT,   "move-result-object",  k11x, ...)
(0x0D, MOVE_EXCEPTION,       "move-exception",      k11x, ...)
(0x0E, RETURN_VOID,          "return-void",         k10x, ...)
(0x0F, RETURN,               "return",              k11x, ...)
(0x10, RETURN_WIDE,          "return-wide",         k11x, ...)
(0x11, RETURN_OBJECT,        "return-object",       k11x, ...)
(0x12, CONST_4,              "const/4",             k11n, ...)
(0x13, CONST_16,             "const/16",            k21s, ...)
(0x14, CONST,                "const",               k31i, ...)
(0x15, CONST_HIGH16,         "const/high16",        k21h, ...)
(0x16, CONST_WIDE_16,        "const-wide/16",       k21s, ...)
```



DEX INSTRUCTION FORMAT

```
( k10x,   // op                                    )
( k12x,   // op  vA, vB                            )
( k11n,   // op  vA, #+B                           )
( k11x,   // op  vAA                               )
( k10t,   // op  +AA                               )
( k20t,   // op  +AAAA                             )
( k22x,   // op  vAA, vBBBB                        )
( k21t,   // op  vAA, +BBBB                        )
( k21s,   // op  vAA, #+BBBB                       )
( k21h,   // op  vAA, #+BBBB00000[00000000]        )
( k21c,   // op  vAA, thing@BBBB                   )
( k23x,   // op  vAA, vBB, vCC                     )
( k22b,   // op  vAA, vBB, #+CC                    )
( k22t,   // op  vA, vB, +CCCC                     )
( k22s,   // op  vA, vB, #+CCCC                    )
( k22c,   // op  vA, vB, thing@CCCC                )
( k32x,   // op  vAAAA, vBBBB                      )
( k30t,   // op  +AAAAAAAA                         )
( k31t,   // op  vAA, +BBBBBBBB                    )
( k31i,   // op  vAA, #+BBBBBBBB                   )
( k31c,   // op  vAA, thing@BBBBBBBB               )
( k35c,   // op  {vC, vD, vE, vF, vG}, thing@BBBB  )
( k3rc,   // op  {vCCCC .. v(CCCC+AA-1)}, meth@BBBB )
```


Interpreter
...
Register
Codes Pointer
Program Counter
Method Info
MRI
int java.lang.Math.min(int, int)
0x4 : move r1, r3
reg0(0x00000000), reg1(0x00000019), reg2(0x00000001),
reg3(0x00000000), reg4(0x00000019), reg5(0x00000001)
Shadow Frame
c4 f4 e4 c3 6b cd 16 d8 b3 b3
c9 b1 7f 0e 7d 0b 8e 66 c2 e2
e0 dc 33 a7 c0 d8 67 a7 c9 a6
e1 8f 7d 68 a6 c5 f6 b5 f6 b5
a7 e1 01 01 10 b9 c9 b9 a7 b5
e0 b5 f7 ba f2 d7 4c 49 f6 c8
0f f7 dd f7 a9 f2 e7 7e 46 a1
f6 df e7 89 d9 cd 86 a8 d8 9c
3d 3f 4a d2 3b b1 a7 a0 65 e1
1b 0e 2d fa 9c bb 1b a8 3d 7f
This items sets the UID of the target app.
[TARGET_APP_UID]=[10169]
This item sets the target methods
[INCLUDE]=[com.amazonaws]
###[INCLUDE]=[com.amazonaws.auth.BasicAWSCredentials]
###[INCLUDE]=[java.lang.String a.a.a.a.e.d.g.d(java.lang.String,]
This item sets the exclude methods
[EXCLUDE]=[android.]
default Android Class List
[EXCLUDE]=[android.]
[EXCLUDE]=[java.]
[EXCLUDE]=[org.]
[EXCLUDE]=[com.android.]
[EXCLUDE]=[com.google.]
[EXCLUDE]=[sun.]
[EXCLUDE]=[dalvik.]
[EXCLUDE]=[libcore.]
[EXCLUDE]=[javax.]
[EXCLUDE]=[androidx.]

java.lang.String com.android.server.pm.SettingsXml$ReadSectionImpl.getName()
0x0: *iget-object v0, v1, Landroid/util/TypedXmlPullParser; com.android.server.pm.SettingsXml$ReadSectionImpl.mParser*
vreg0=0x00000000 vreg1=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

java.lang.String com.android.server.pm.SettingsXml$ReadSectionImpl.getName()
0x2: *invoke-interface {v0}, java.lang.String android.util.TypedXmlPullParser.getName()*
vreg0=0x13D16DF0/com.android.internal.util.BinaryXmlPullParser vreg1=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

java.lang.String com.android.server.pm.SettingsXml$ReadSectionImpl.getName()
0x5: *move-result-object v0*
vreg0=0x13D16DF0/com.android.internal.util.BinaryXmlPullParser vreg1=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

java.lang.String com.android.server.pm.SettingsXml$ReadSectionImpl.getName()
0x6: *return-object v0*
vreg0=0x1400BC38/java.lang.String "user-state" vreg1=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

void com.android.server.pm.verify.domain.DomainVerificationLegacySettings.readUserStates(com.android.server.pm.SettingsXml$ReadSection)
0x1c: *move-result-object v5*
vreg0=0x14016978/java.lang.String "com.google.vr.apps.ornament" vreg1=0x13B16588/java.lang.Object
vreg2=0x140169A8/com.android.server.pm.verify.domain.DomainVerificationLegacySettings$LegacyState
vreg3=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl vreg4=0x13B16538/java.lang.String "user-state" vreg5=0x00000000
vreg6=0x13B16578/com.android.server.pm.verify.domain.DomainVerificationLegacySettings
vreg7=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

void com.android.server.pm.verify.domain.DomainVerificationLegacySettings.readUserStates(com.android.server.pm.SettingsXml$ReadSection)
0x1d: *invoke-virtual {v4, v5}, boolean java.lang.String.equals(java.lang.Object)*
vreg0=0x14016978/java.lang.String "com.google.vr.apps.ornament" vreg1=0x13B16588/java.lang.Object
vreg2=0x140169A8/com.android.server.pm.verify.domain.DomainVerificationLegacySettings$LegacyState
vreg3=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl vreg4=0x13B16538/java.lang.String "user-state"
vreg5=0x1400BC38/java.lang.String "user-state" vreg6=0x13B16578/com.android.server.pm.verify.domain.DomainVerificationLegacySettings
vreg7=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

void com.android.server.pm.verify.domain.DomainVerificationLegacySettings.readUserStates(com.android.server.pm.SettingsXml$ReadSection)
0x20: *move-result v4*
vreg0=0x14016978/java.lang.String "com.google.vr.apps.ornament" vreg1=0x13B16588/java.lang.Object
vreg2=0x140169A8/com.android.server.pm.verify.domain.DomainVerificationLegacySettings$LegacyState
vreg3=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl vreg4=0x13B16538/java.lang.String "user-state"
vreg5=0x1400BC38/java.lang.String "user-state" vreg6=0x13B16578/com.android.server.pm.verify.domain.DomainVerificationLegacySettings
vreg7=0x1400B920/com.android.server.pm.SettingsXml$ReadSectionImpl

Interpreter

int java.lang.Math.min(int, int) *// method info*

0x4 : ***move r0, r3*** *// pc and instruction*

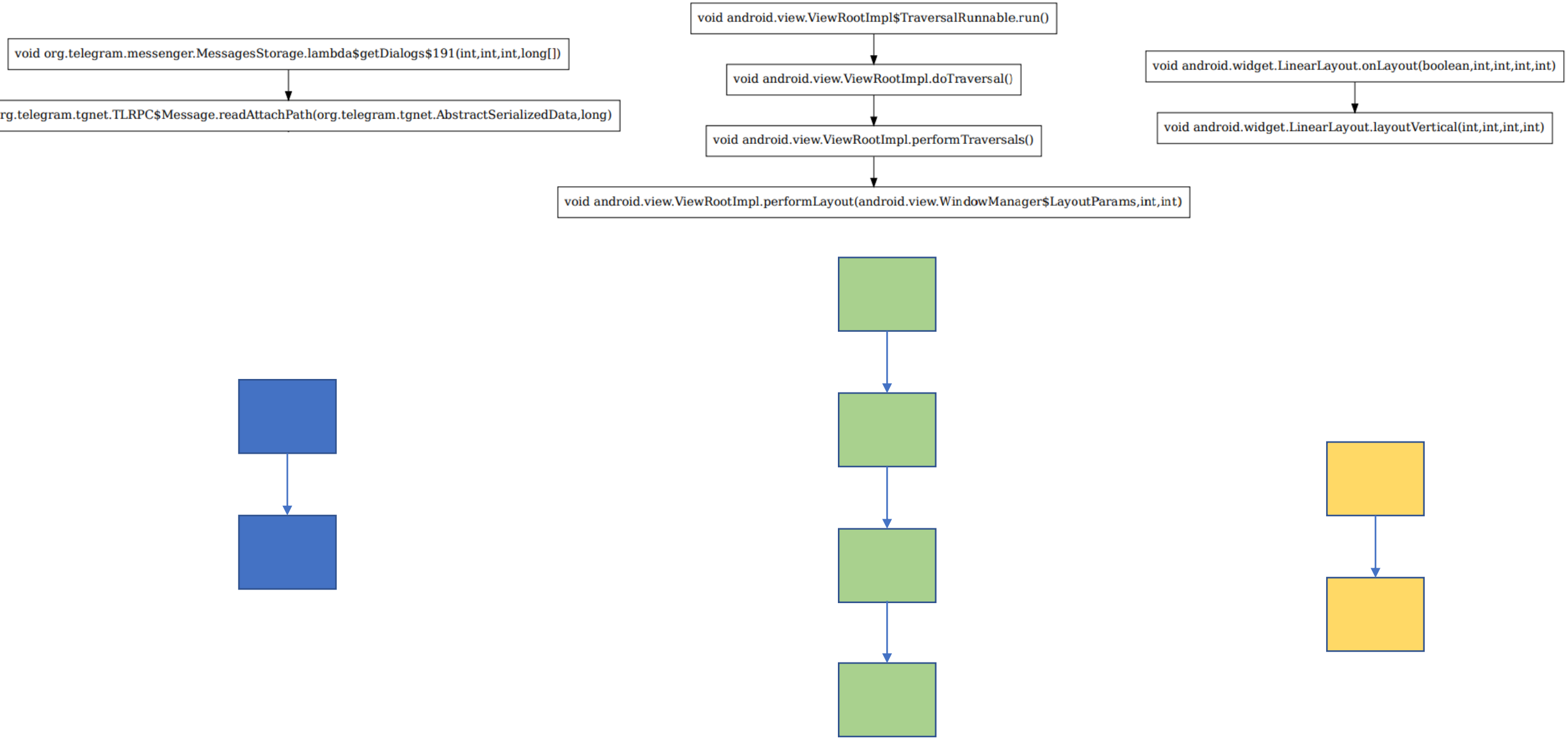
reg0(0x00000000), reg1(0x00000019), reg2(0x00000001), reg3(0xffec30008), reg4(0x003000ff), … *// register*

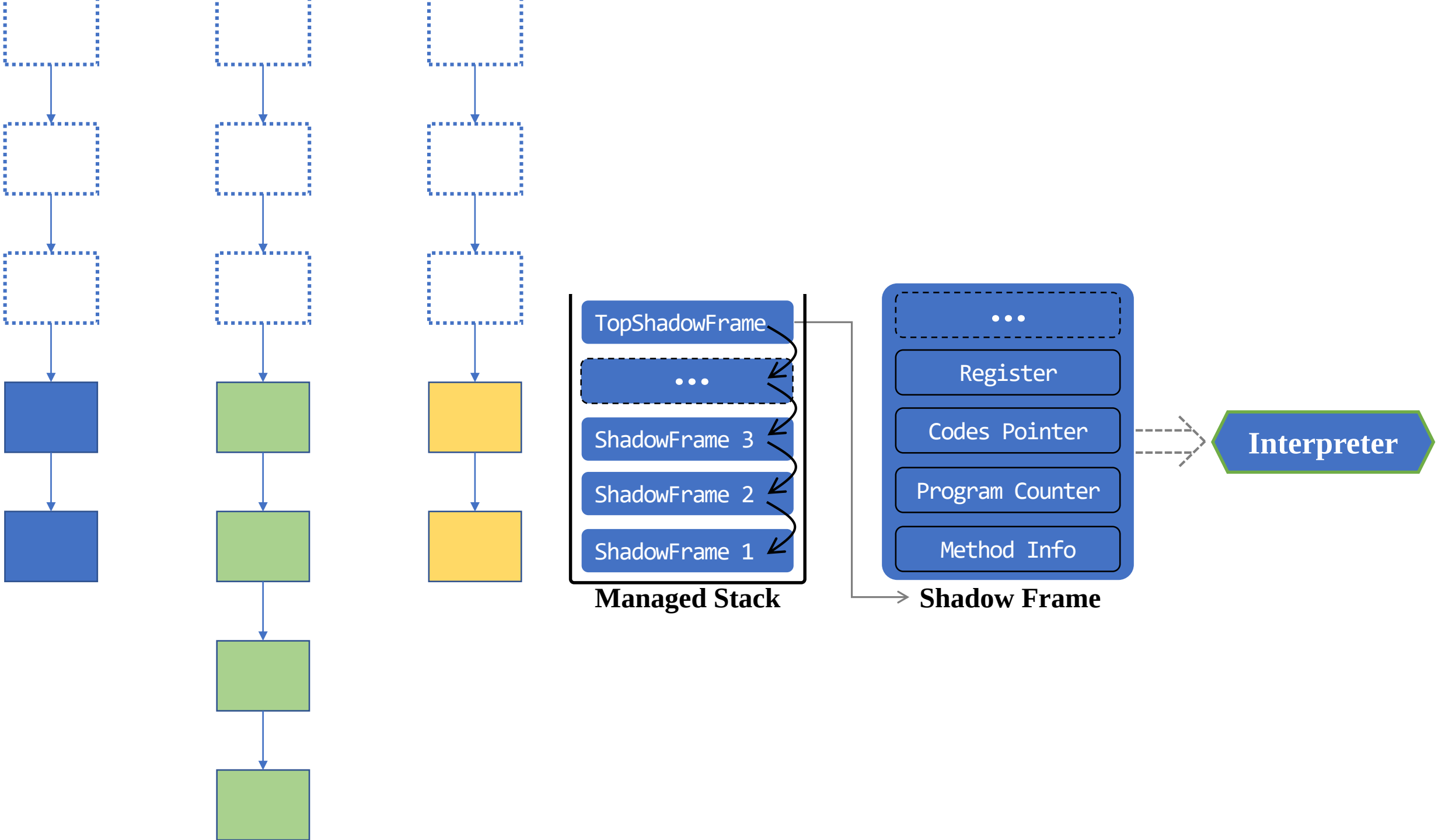
Action	Dalvik Instruction List
Create	*const-string, const-string-jumbo*
Move	*move-object, move-object_from16, move-object-16, move-result-object, return-object*
Copy	*aget-object, aput-object, iget-object, iput-object, sget-object, sput-object*
Call	*invoke-virtual, invoke-virtual-range, invoke-super, invoke-super-range, invoke-direct, invoke-direct-range, invoke-interface, invoke-interface-range, invoke-static, invoke-static-range*

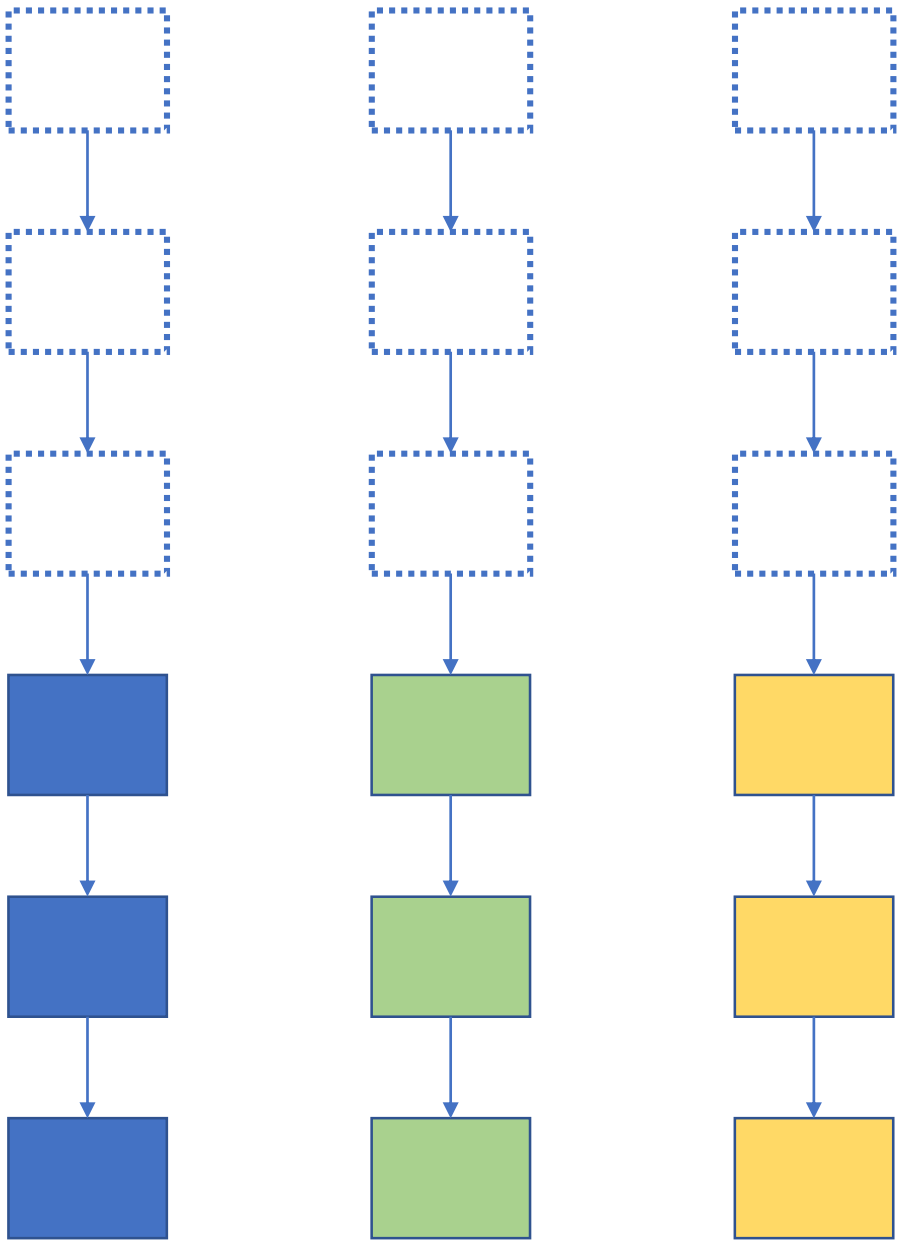
reg3(0xffec30008)

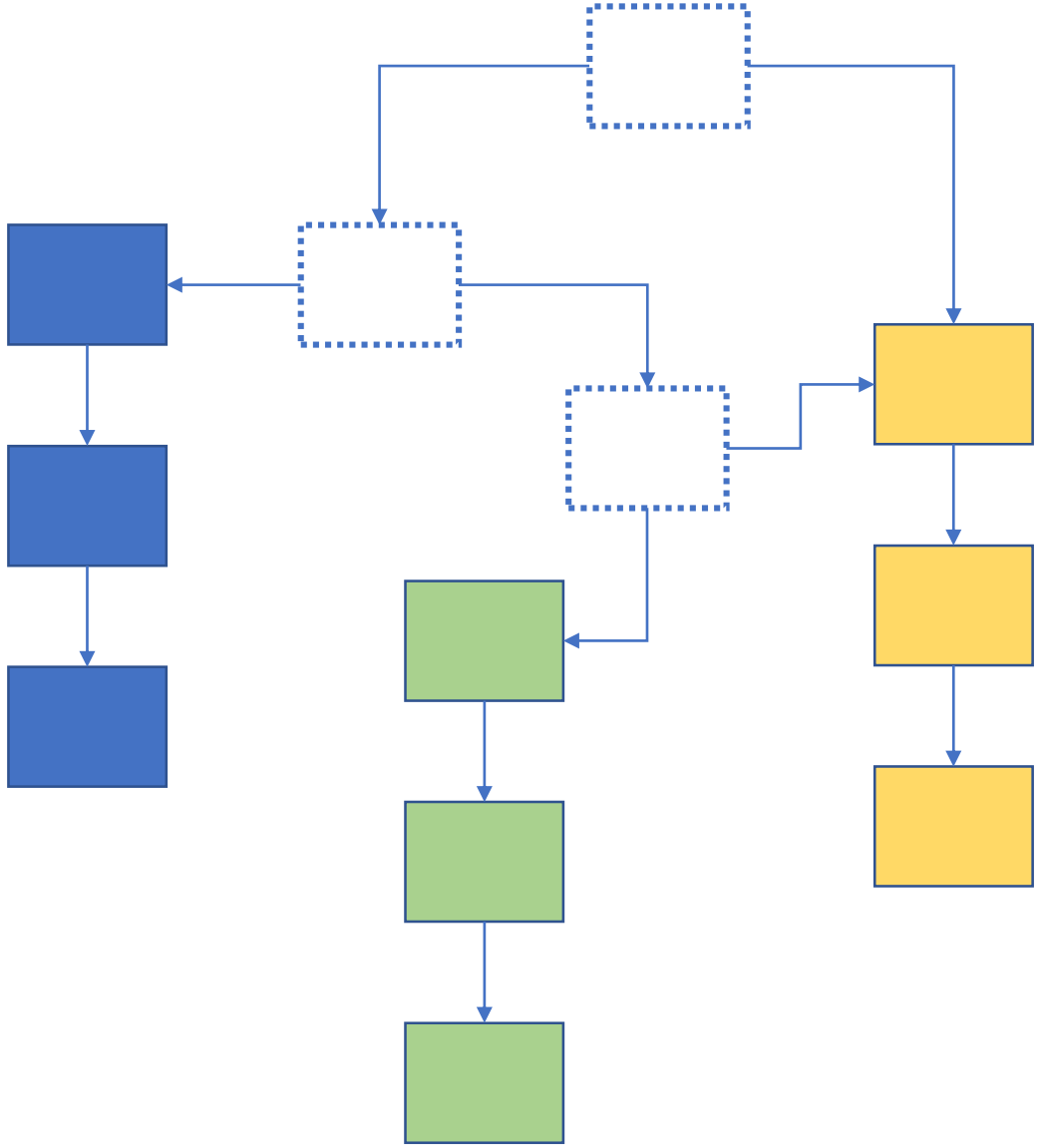
⇒ CHECK TYPE : byte[], char[], java.string.String, java.lang.charsequence, …

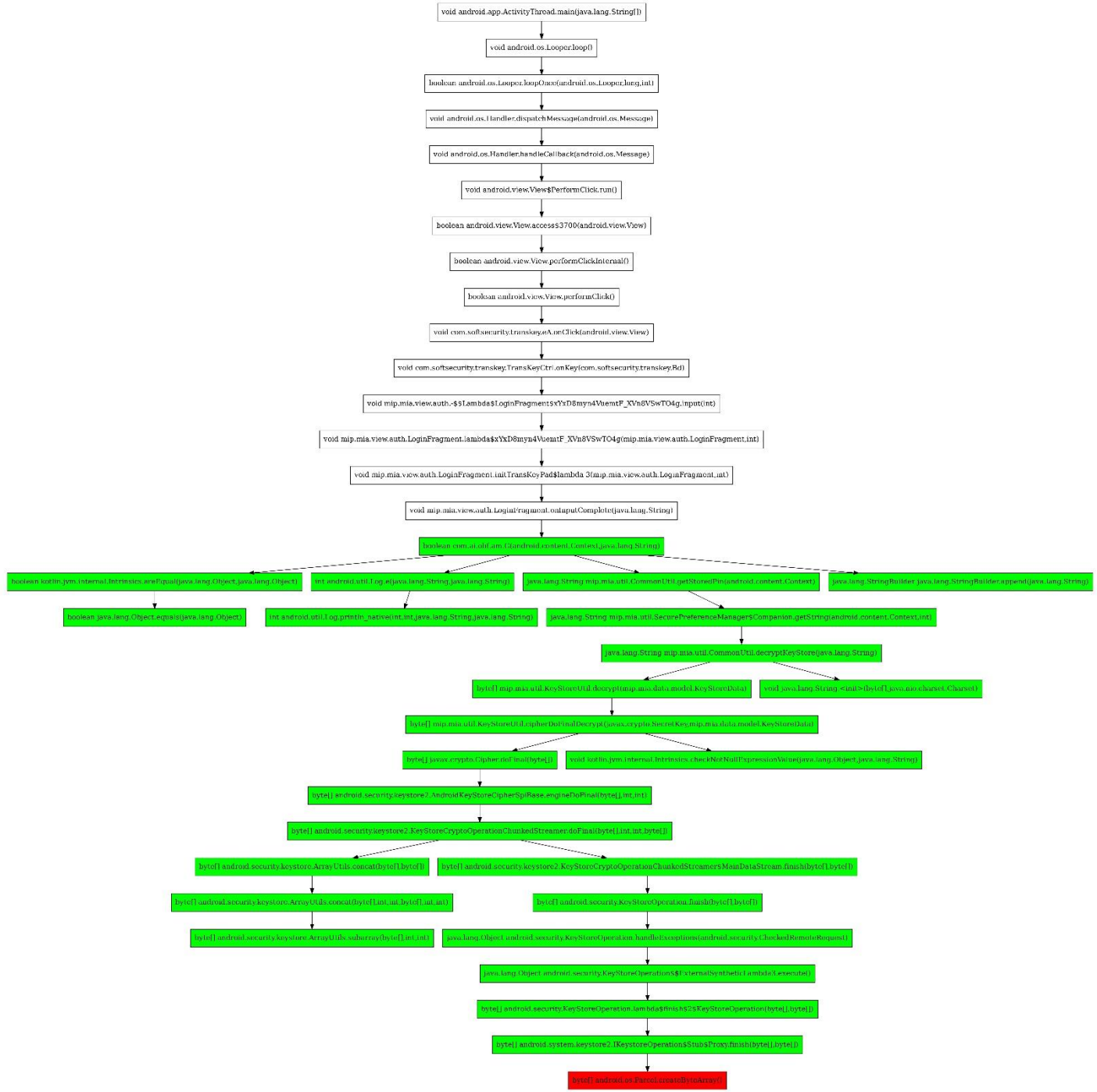
⇒ CHECK TARGET : reg3(0xffec30008) == "1q2w3e4r!"

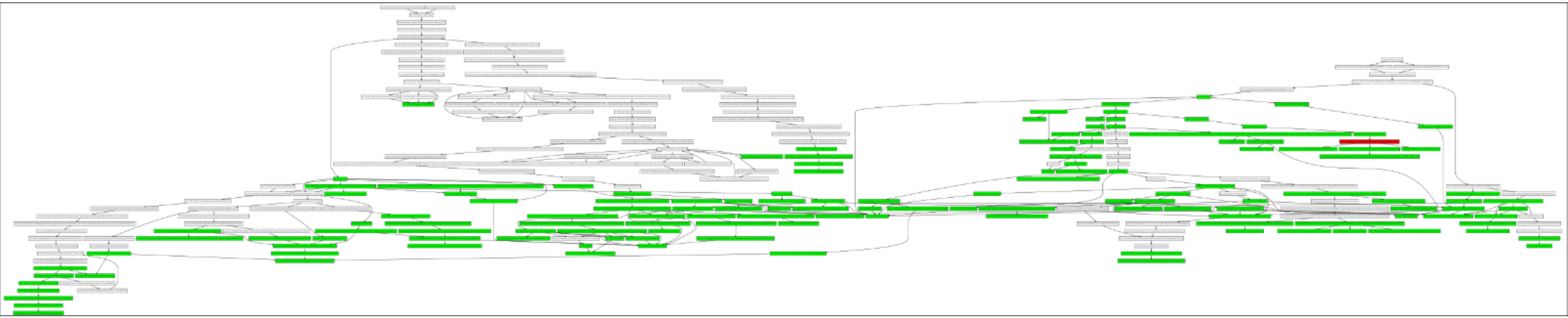

void org.telegram.messenger.MessagesStorage.lambda$getDialogs$191(int,int,int,long[])
rg.telegram.tgnet.TLRPC$Message.readAttachPath(org.telegram.tgnet.AbstractSerializedData,long)
void android.view.ViewRootImpl$TraversalRunnable.run()
void android.view.ViewRootImpl.doTraversal()
void android.view.ViewRootImpl.performTraversals()
void android.view.ViewRootImpl.performLayout(android.view.WindowManager$LayoutParams,int,int)
void android.widget.LinearLayout.onLayout(boolean,int,int,int,int)
void android.widget.LinearLayout.layoutVertical(int,int,int,int)

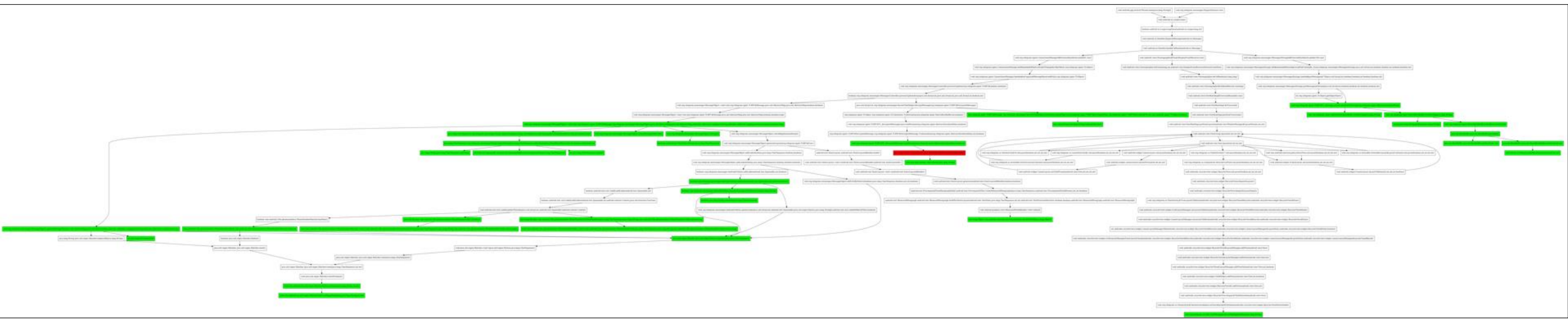

TopShadowFrame
...
ShadowFrame 3
ShadowFrame 2
ShadowFrame 1
Managed Stack
...
Register
Codes Pointer
Program Counter
Method Info
Shadow Frame
Interpreter






void android.app.ActivityThread.main(java.lang.String[])
void android.os.Looper.loop()
boolean android.os.Looper.loopOnce(android.os.Looper,long,int)
void android.os.Handler.dispatchMessage(android.os.Message)
void android.os.Handler.handleCallback(android.os.Message)
void android.view.View$PerformClick.run()
boolean android.view.View.access$3700(android.view.View)
boolean android.view.View.performClickInternal()
boolean android.view.View.performClick()
void com.softsecurity.transkey.eA.onClick(android.view.View)
void com.softsecurity.transkey.TransKeyCtrl.onKey(com.softsecurity.transkey.Rd)
void mip.mia.view.auth.-$$Lambda$LoginFragmentsXYxD8myn4VuemtF_XVn8VSwTO4g.input(int)
void mip.mia.view.auth.LoginFragment.lambda$xYxD8myn4VuemtF_XVn8VSwTO4g(mip.mia.view.auth.LoginFragment,int)
void mip.mia.view.auth.LoginFragment.initTransKeyPad$lambda 3(mip.mia.view.auth.LoginFragment,int)
void mip.mia.view.auth.LoginFragment.onInputComplete(java.lang.String)
boolean com.ai.ahf.ars (android.content.Context,java.lang.String)
boolean kotlin.jvm.internal.Intrinsics.areEqual(java.lang.Object,java.lang.Object)
int android.util.Log.e(java.lang.String,java.lang.String)
java.lang.String mip.mia.util.CommonUtil.getStoredPin(android.content.Context)
java.lang.StringBuilder java.lang.StringBuilder.append(java.lang.String)
boolean java.lang.Object.equals(java.lang.Object)
int android.util.Log.println_native(int,int,java.lang.String,java.lang.String)
java.lang.String mip.mia.util.SecurePreferenceManager$Companion.getString(android.content.Context,int)
java.lang.String mip.mia.util.CommonUtil.decryptKeyStore(java.lang.String)
byte[] mip.mia.util.KeyStoreUtil.decrypt(mip.mia.data.model.KeyStoreData)
void java.lang.String.<init>(byte[],java.nio.charset.Charset)
byte[] mip.mia.util.KeyStoreUtil.cipherDoFinalDecrypt(javax.crypto.SecretKey,mip.mia.data.model.KeyStoreData)
byte[] javax.crypto.Cipher.doFinal(byte[])
void kotlin.jvm.internal.Intrinsics.checkNotNullExpressionValue(java.lang.Object,java.lang.String)
byte[] android.security.keystore2.AndroidKeyStoreCipherSpiBase.engineDoFinal(byte[],int,int)
byte[] android.security.keystore2.KeyStoreCryptoOperationChunkedStreamer.doFinal(byte[],int,int,byte[])
byte[] android.security.keystore.ArrayUtils.concat(byte[],byte[])
byte[] android.security.keystore2.KeyStoreCryptoOperationChunkedStreamer$MainDataStream.finish(byte[],byte[])
byte[] android.security.keystore.ArrayUtils.concat(byte[],int,int,byte[],int,int)
byte[] android.security.KeyStoreOperation.finish(byte[],byte[])
byte[] android.security.keystore.ArrayUtils.subarray(byte[],int,int)
java.lang.Object android.security.KeyStoreOperation.handleExceptions(android.security.CheckedRemoteRequest)
java.lang.Object android.security.KeyStoreOperation$$ExternalSyntheticLambda3.execute()
byte[] android.security.KeyStoreOperation.lambda$finish$2$KeyStoreOperation(byte[],byte[])
byte[] android.system.keystore2.IKeystoreOperation$Stub$Proxy.finish(byte[],byte[])
byte[] android.os.Parcel.createByteArray()

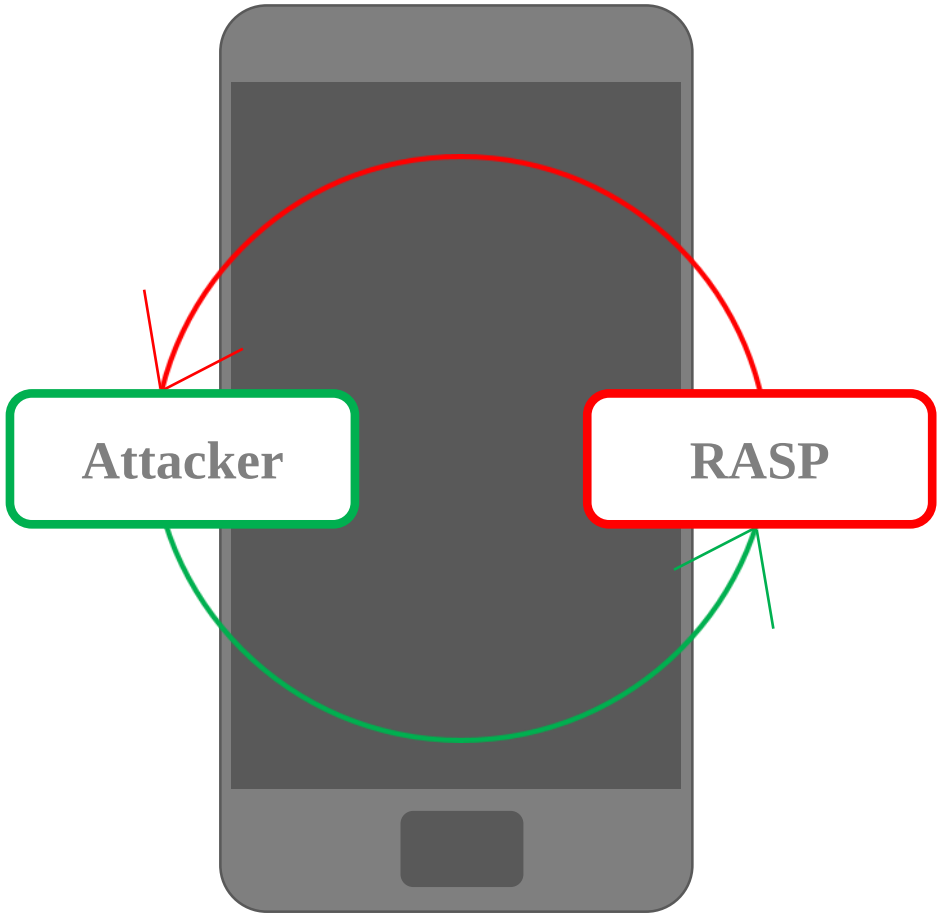




RASP (Runtime Application Self-Protection)

RASP is a security technology that uses runtime instrumentation to detect and block computer attacks by taking advantage of information from inside the running software.

https://en.wikipedia.org/wiki/Runtime_application_self-protection



OS Integrity Check
Code Obfuscation
Device Binding
Anti-Emulator
Anti-Debugging
Data Encryption
Anti-Tampering
Secure Communication
Anti-Keylogger
Anti-Hooking

Root Detections

/system/xbin/su
/system/bin/.ext/su
/odm/bin/su ...

Hooking Framework Detections

frida-agent-64.so
libxposed_art.so
frida-agent-32.so ...

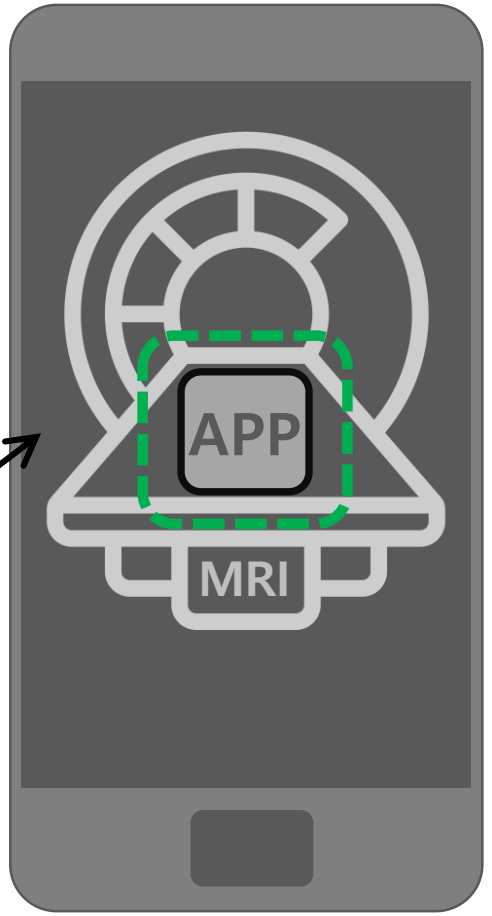
VS

Emulator Checks

init.nox.rc
com.google.android.launcher.layouts.genymotion
com.bluestacks ...

Custom ROM Checks

/system/build.prop
ro.build.tags, test-keys
ro.boot.vbmeta.device_state ...




EvolutionX
Pixel UI, Android 13, Customization and more.
We are Evolution X!


Bliss Roms


ArrowOS


AOSiP


Havoc-OS


Corvus OS


LINEAGE


paranoidandroid


pixelexperience

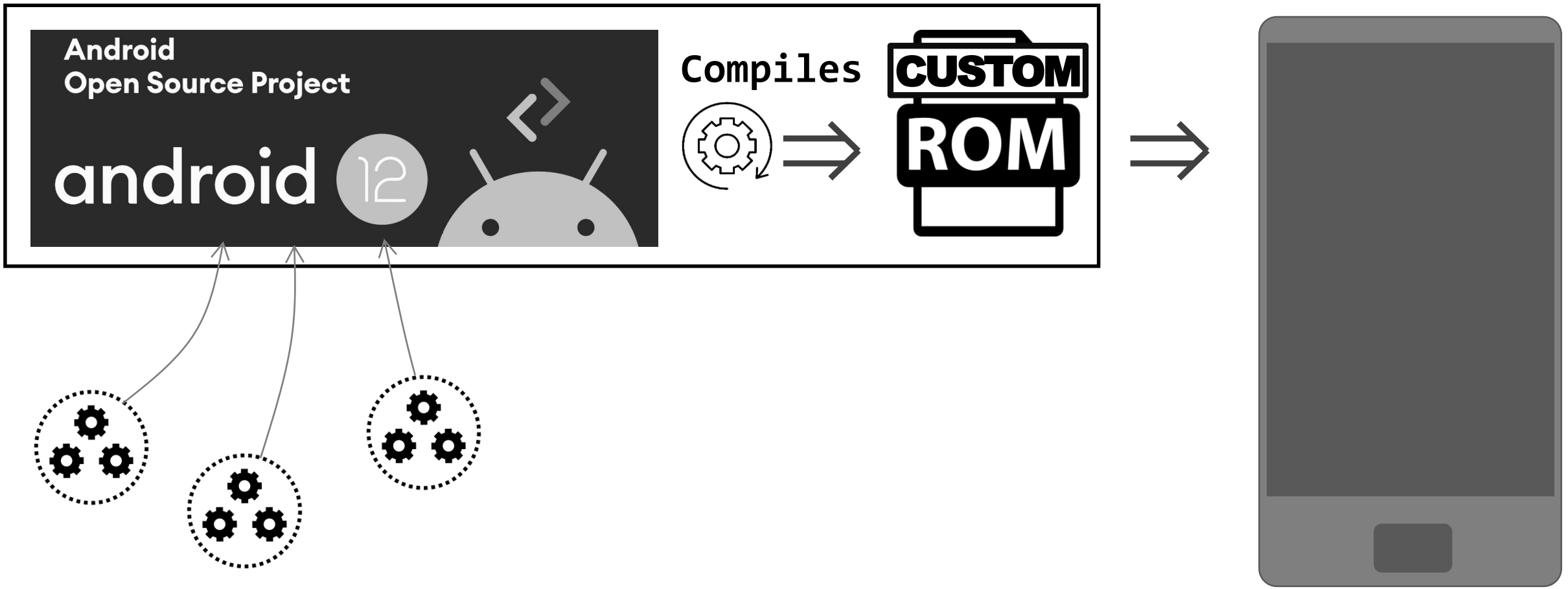

crDroid

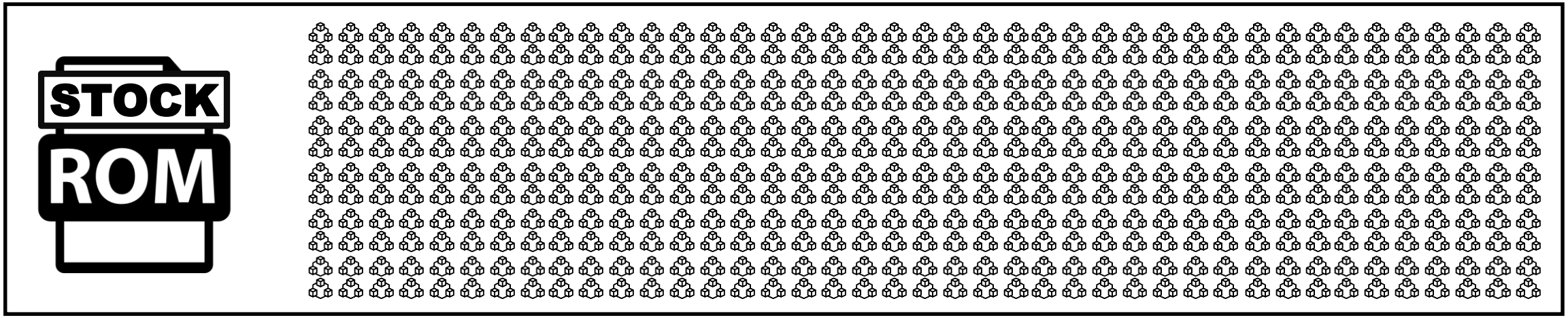

RevengeOS


PROJECT SAKURA
feels like spring.


SYBERIA


MSMXTENDED
CUSTOM ROM REDEFINED


Android
Open Source Project
android 12
Compiles
CUSTOM
ROM



Application Layer

Application 1

Application 2

....

Application N

Platform Layer

Runtime

MRI
MODULE

Framework

MEMORY MODIFY
DRIVER INTERFACE

SYSCALL HOOK
DRIVER INTERFACE

Kernel Layer

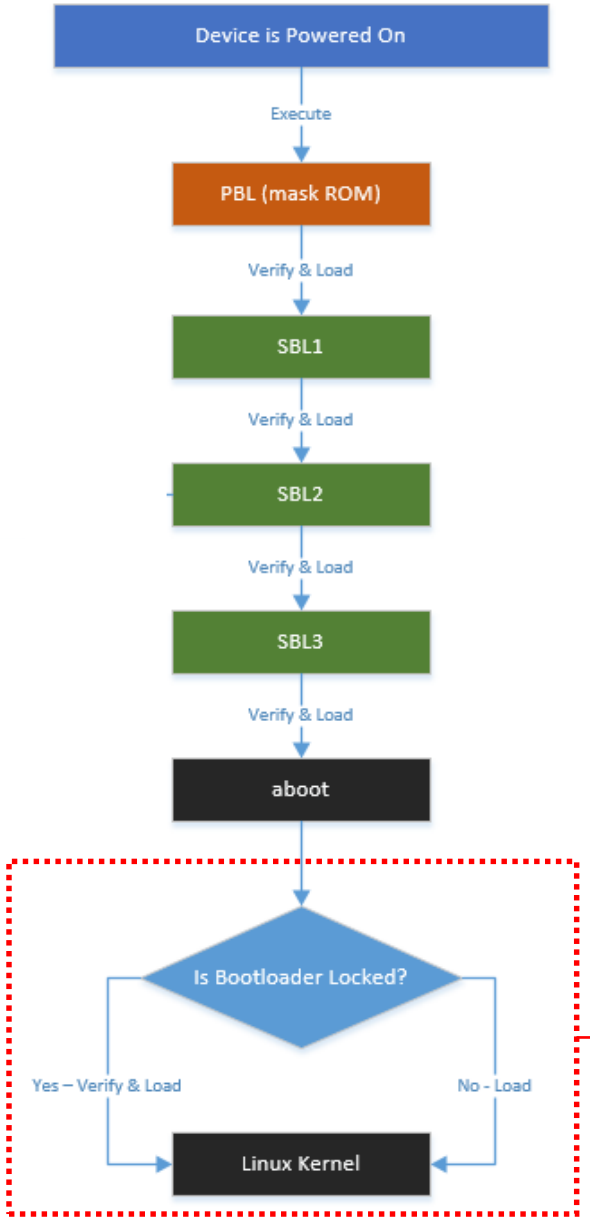
KERNEL

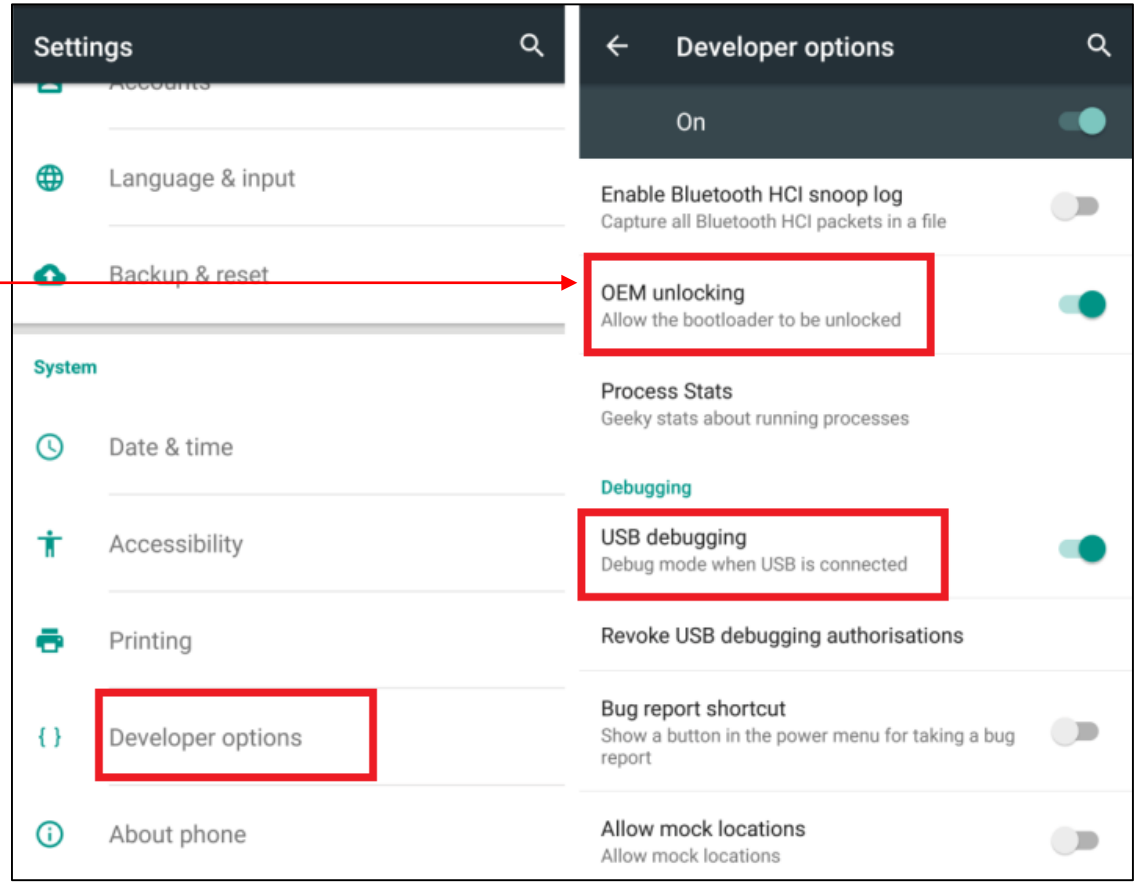
PRIVILEGE
MODULE

SELINUX
BYPASS

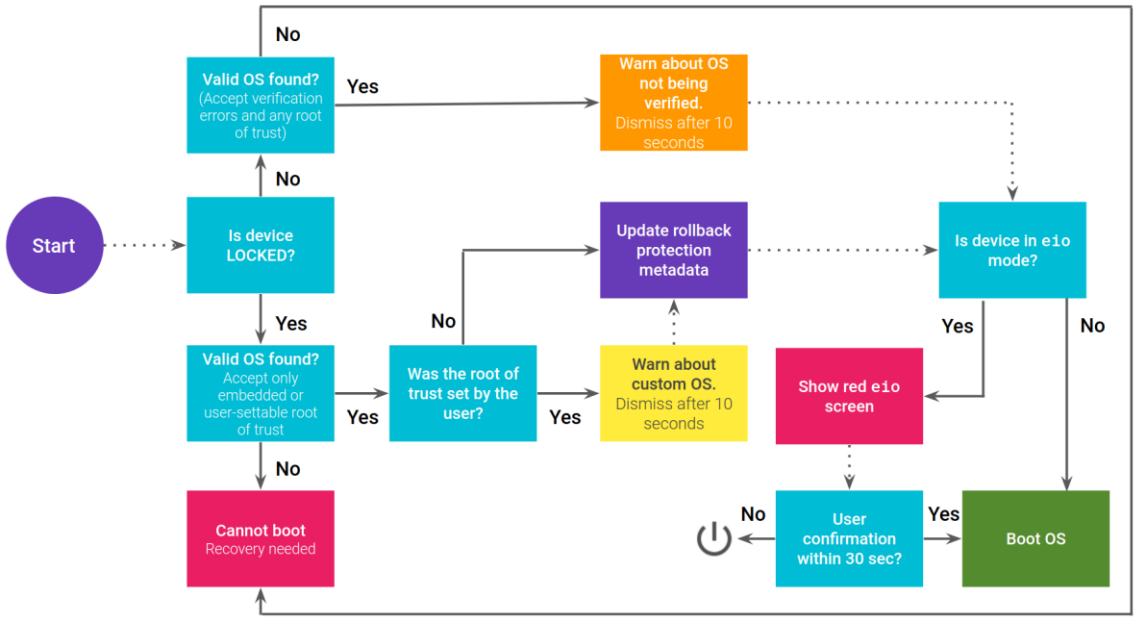
AVB
BYPASS

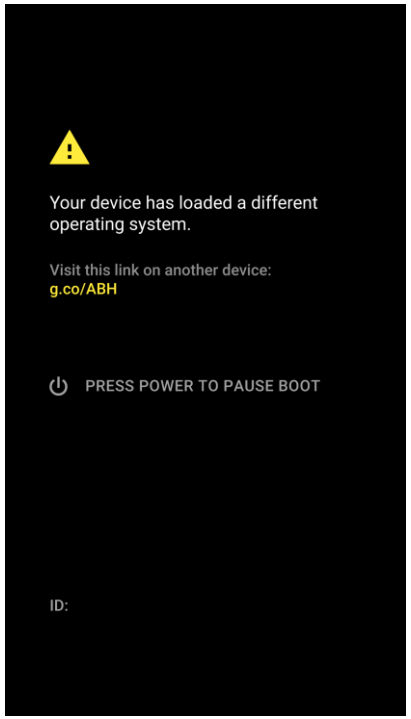
DRIVER
MODULE

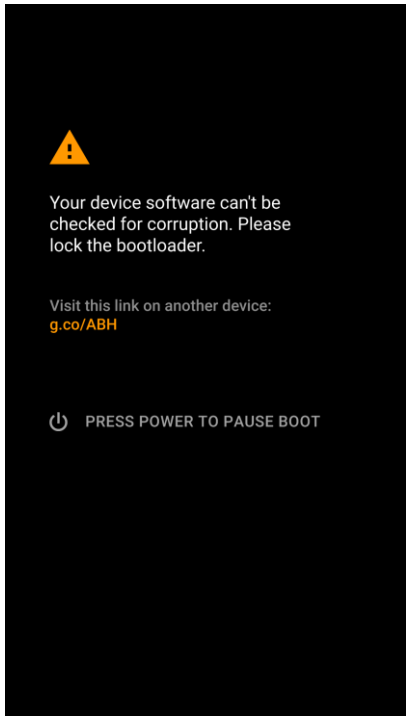


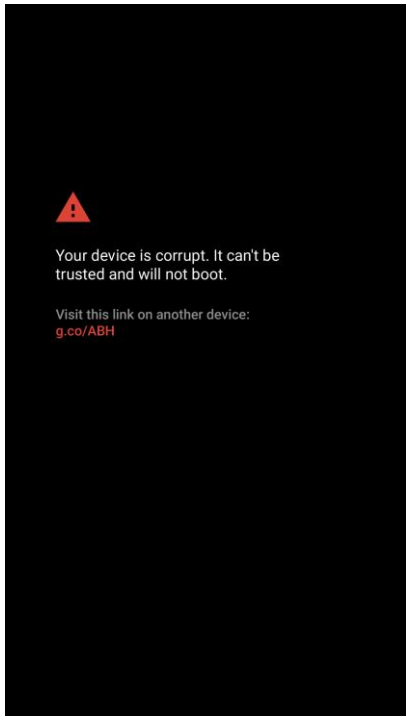


http://bits-please.blogspot.kr/2016/02/unlocking-motorola-bootloader.html









https://source.android.com/docs/security/features/verifiedboot/boot-flow

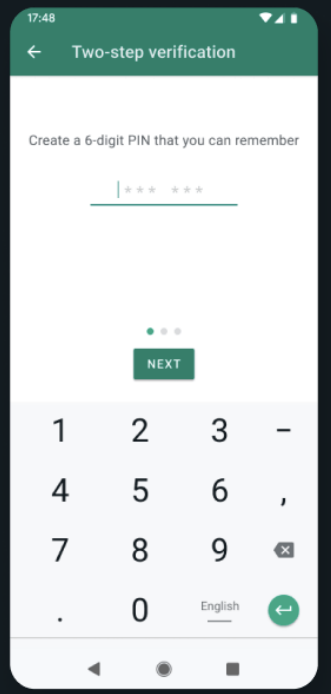
```
static int cmdline_proc_show(struct seq_file *m, void *v)
{
  seq_printf(m, "%s\n", saved_command_line);
  return 0;
}
```

```
static int cmdline_proc_show(struct seq_file *m, void *v)
{
  seq_printf(m, "%s\n", "rcupdate.rcu_expedited=1 rootwait ro init=/init androidboot.bootdevice=1d84000.ufshc androidboot.baseband=sdm androidboot.keymaster=1 msm_drm.dsi_display0=dsi_s6e3ha8_cmd_display::timing0 androidboot.force_normal_boot=1 androidboot.serialno=8B9Y0VSZ3 androidboot.slot_suffix=_a androidboot.slot_retry_count=2 androidboot.slot_successful=no androidboot.hardware.platform=sdm845 androidboot.hardware=crosshatch androidboot.revision=MP1.0 androidboot.bootloader=b1c1-0.4-7617406 androidboot.hardware.sku=G013C androidboot.hardware.radio.subtype=0 androidboot.hardware.dsds=0 androidboot.secure_boot=PRODUCTION androidboot.cdt_hwid=0x05012800 androidboot.hardware.majorid=0x01 androidboot.dtb_idx=0 androidboot.dtbo_idx=13 androidboot.bootreason=reboot androidboot.hardware.ddr=4GB,Samsung,LPDDR4X androidboot.ddr_info=Samsung androidboot.ddr_size=4GB androidboot.hardware.ufs=128GB,Micron androidboot.cid=00000000 androidboot.boottime=0BLE:88,1BLL:141,1BLE:1042,2BLL:121,2BLE:498,SW:10028,KL:0,KD:72,ODT:106,AVB:245,AFTL:0 androidboot.ramdump=disabled androidboot.blockchain=disabled usbcfg.suzyq=disabled androidboot.theme=1 androidboot.hardware.pcbcfg=G650-01995-06 androidboot.hardware.devcfg=G950-00762-03 androidboot.vbmeta.device=PARTUUID=b7fc981c-7b25-4f78-848a-e5015e7b3cf5 androidboot.vbmeta.avb_version=1.1 androidboot.vbmeta.device_state=locked androidboot.vbmeta.hash_alg=sha256 androidboot.vbmeta.size=4224 androidboot.vbmeta.digest=68cd5529bb07077fee41e72d9e4bf35e89cb61be1488e08f37681a94e87f2c26 androidboot.veritymode=enforcing androidboot.verifiedbootstate=green androidboot.aftlstate=8 printk.devkmsg=on msm_rtb.filter=0x237 ehci-hcd.park=3 service_locator.enable=1 cgroup.memory=nokmem lpm_levels.sleep_disabled=1 usbcore.autosuspend=7 loop.max_part=7 androidboot.boot_devices=soc/1d84000.ufshc androidboot.super_partition=system buildvariant=user console=null");
  //seq_printf(m, "%s\n", saved_command_line);
  return 0;
}
```

ls /system/bin/

DEMO

WhatsApp

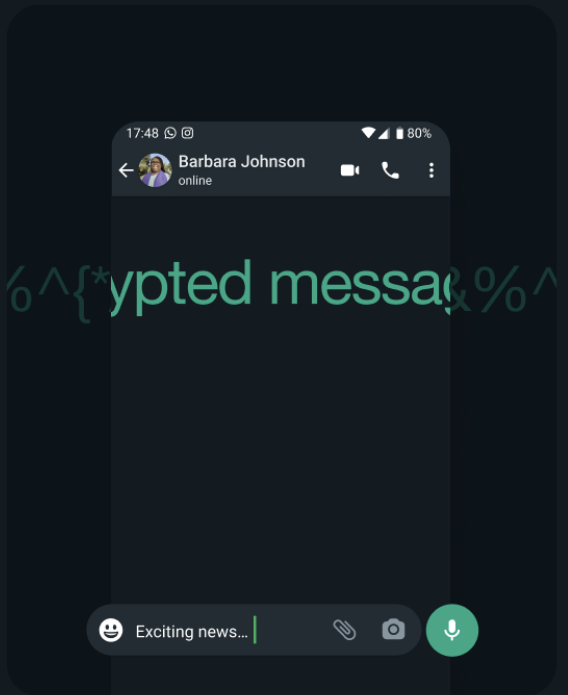


Two-step verification

Stay two steps ahead of intruders. Protect your account from hackers and scammers trying to use your phone number.

Message privately

Your privacy is our priority. With end-to-end encryption, you can be sure that your personal messages stay between you and who you send them to.



WhatsApp

faq.whatsapp.com/649203676836357/?helpref=search&cms_platform=android&query=last&search_session_id=3b3bd40d8314d0c1c775921d22cbcb9c&sr=2

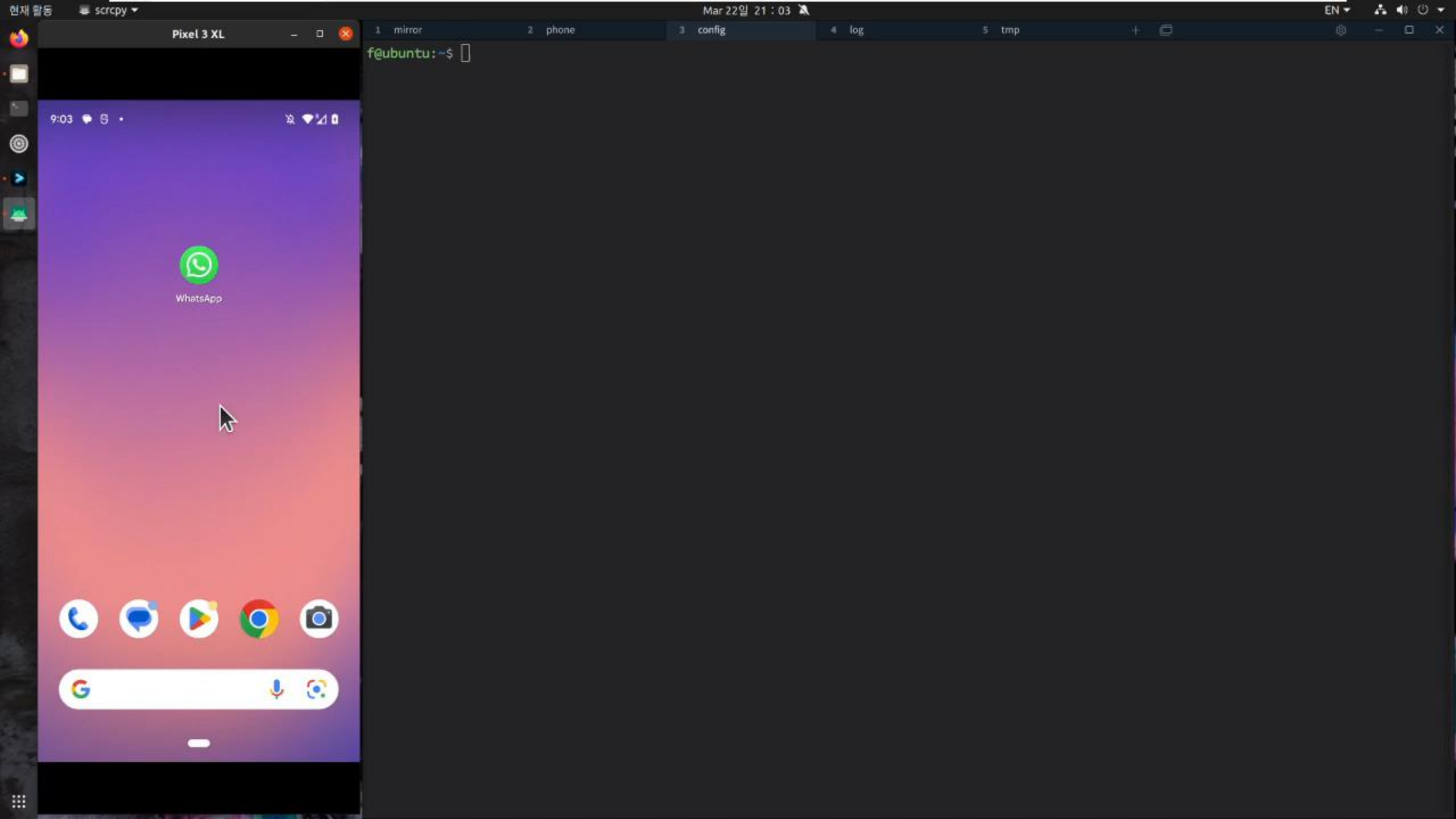
About rooted phones and custom ROMs

Copy link

Android

Custom ROMs and rooted phones aren't supported by WhatsApp. There are too many variations in these customizations for us to maintain a working product. Furthermore, custom ROMs and rooting don't allow the WhatsApp security model to function as intended. If you're using a custom ROM or rooted phone, other apps might be able to read your messages despite the end-to-end-encryption.

For the best WhatsApp experience, please use a stock ROM and remove root. Contact your phone's manufacturer for specific instructions on how to unroot.


현재 활동
scrcpy
Mar 22일 21 : 03
EN
Pixel 3 XL
9:03
WhatsApp
1 mirror
2 phone
3 config
4 log
5 tmp
f@ubuntu:~$

END

SungHyoun Song
decash@fsec.or.kr (@decashx)