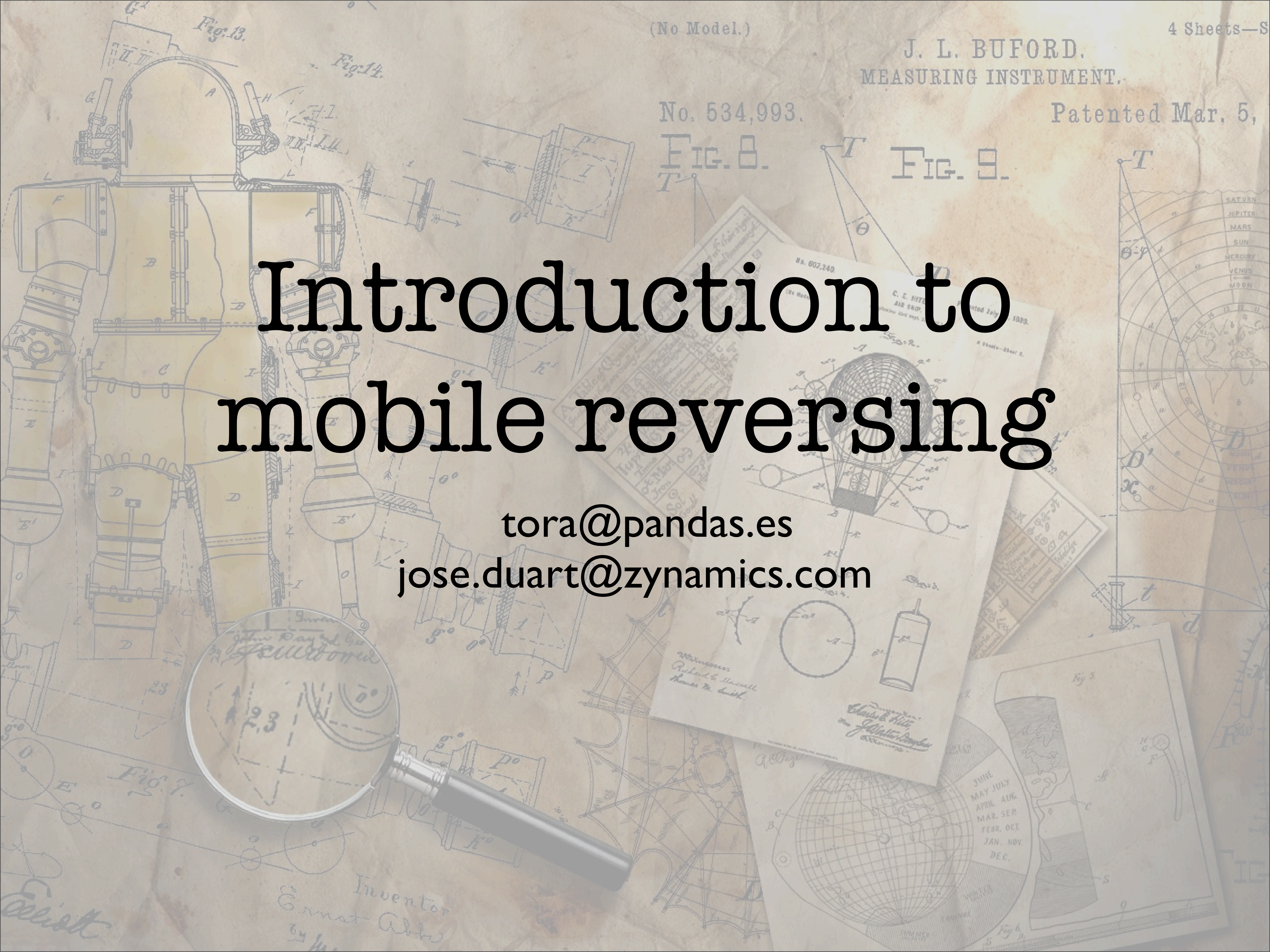


Introduction to mobile reversing

tora@pandas.es
jose.duart@zynamics.com

Who are you?

- Reverse engineer since late 90s
  - Malware analysis, binary auditing and behaviour analysis
- Mobile and embedded systems reversing as a hobby
- Working at Zynamics GmbH

Summary

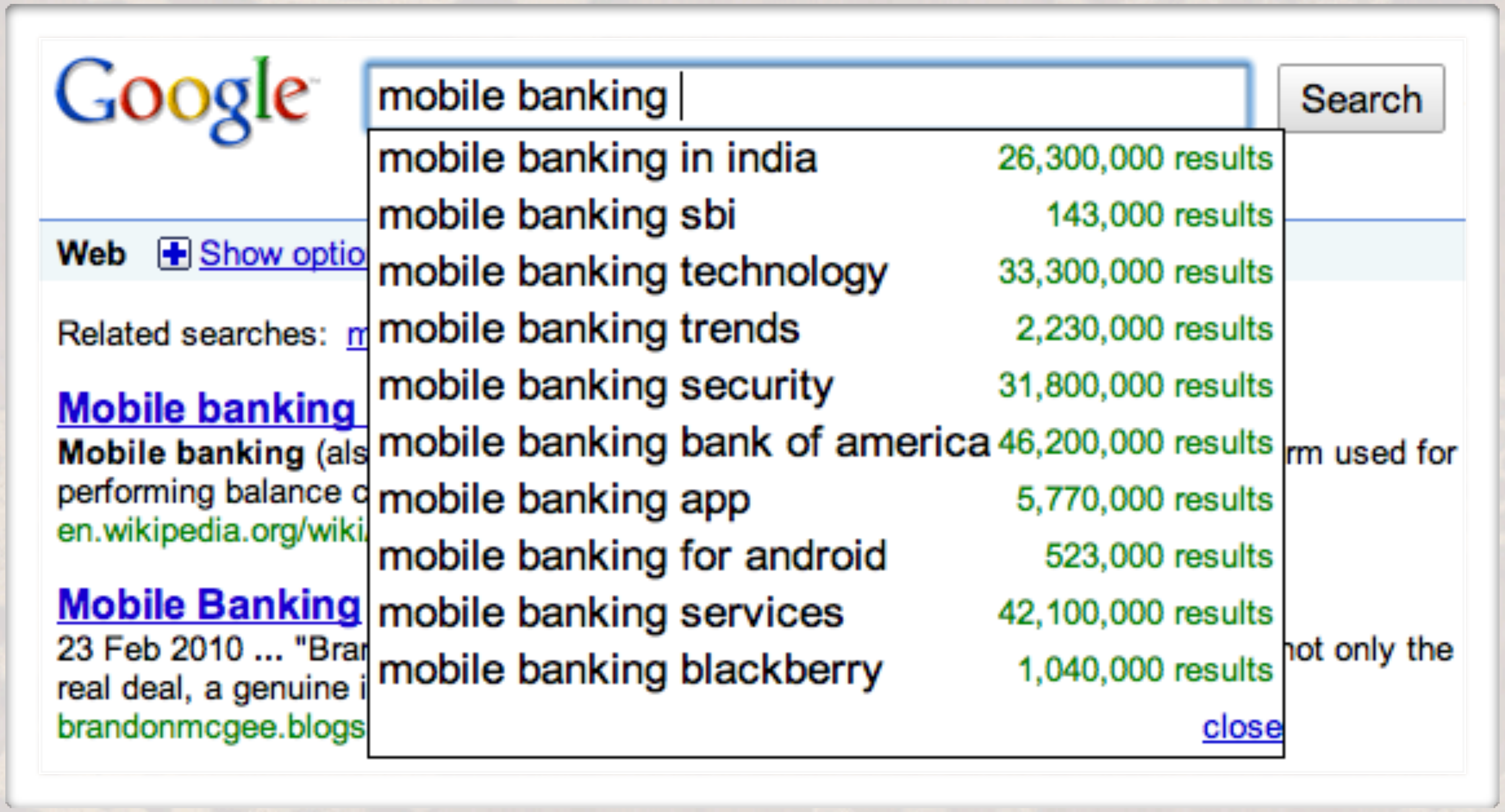
- Windows Mobile
- Android
  - Dalvik VM
  - Decompilation
- iPhone
  - AppStore DRM
  - Reversing Objective-C code

Why mobile reversing?

- Few years ago:
  - Lots of java-based mobileOS
  - Only Symbian and PalmOS were “interesting”
    - But not much development community
    - First? mobile malware developed for Symbian, codename Caribe/Cabir, author Vallez/29A
- Now:
  - Android, iPhoneOS, Windows Mobile, Bada...

Why mobile reversing?

- More SDK's, more developers... more interesting stuff



mobile banking

mobile banking in india	26,300,000 results
mobile banking sbi	143,000 results
mobile banking technology	33,300,000 results
mobile banking trends	2,230,000 results
mobile banking security	31,800,000 results
mobile banking bank of america	46,200,000 results
mobile banking app	5,770,000 results
mobile banking for android	523,000 results
mobile banking services	42,100,000 results
mobile banking blackberry	1,040,000 results

Related searches: m

Mobile banking

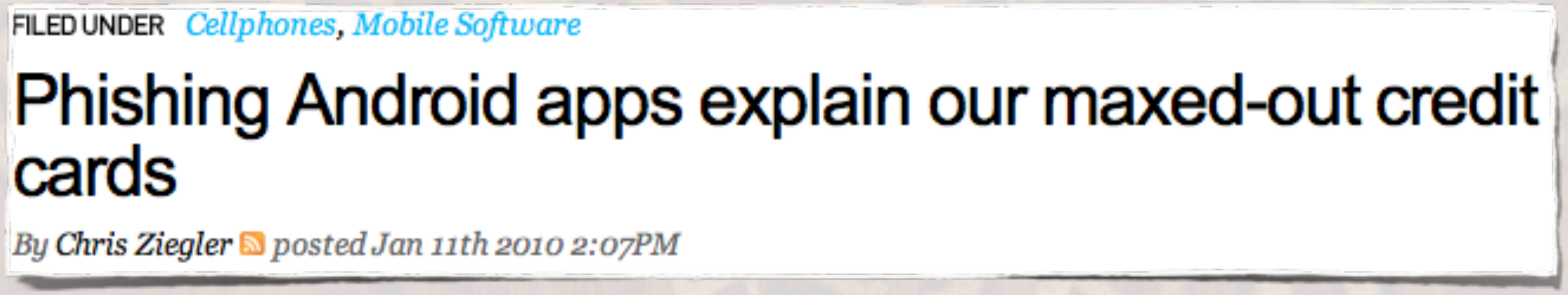
Mobile banking (als rm used for
performing balance c
en.wikipedia.org/wiki/

Mobile Banking

23 Feb 2010 ... "Bran not only the
real deal, a genuine i
brandonmcgee.blogs

Why mobile reversing?

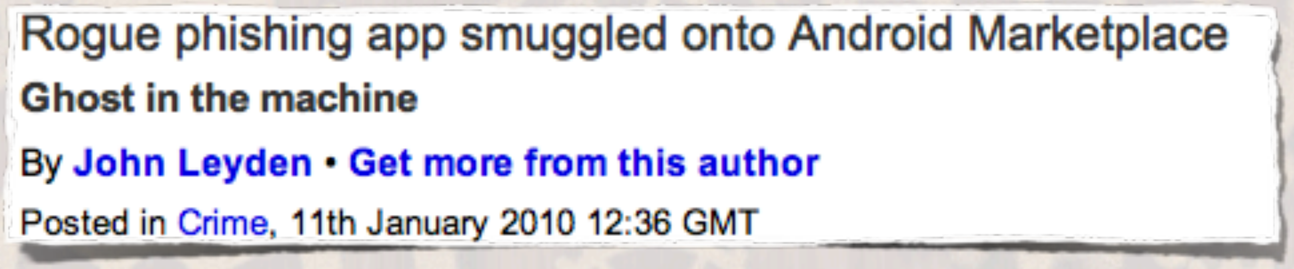
- ... and evil stuff



FILED UNDER *Cellphones*, *Mobile Software*

Phishing Android apps explain our maxed-out credit cards

By Chris Ziegler posted Jan 11th 2010 2:07PM

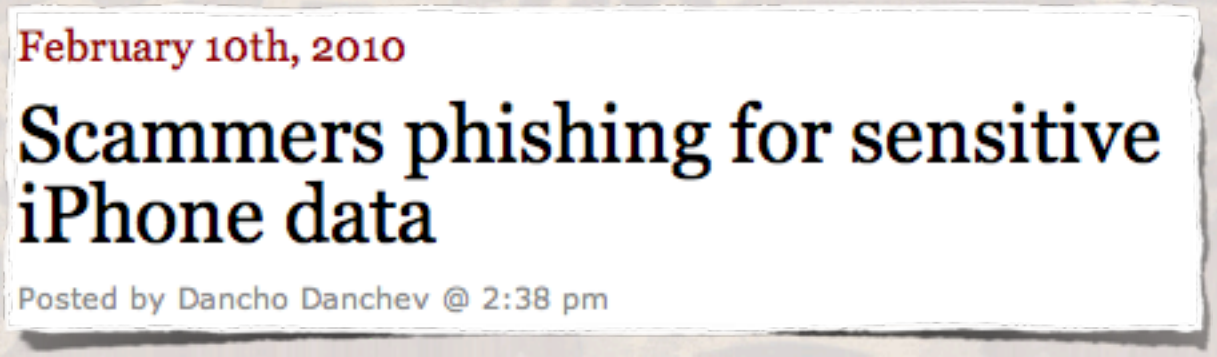


Rogue phishing app smuggled onto Android Marketplace

Ghost in the machine

By **John Leyden** • **Get more from this author**

Posted in Crime, 11th January 2010 12:36 GMT



February 10th, 2010

Scammers phishing for sensitive iPhone data

Posted by Dancho Danchev @ 2:38 pm

Windows Mobile

Reversing a banking app: BBVA

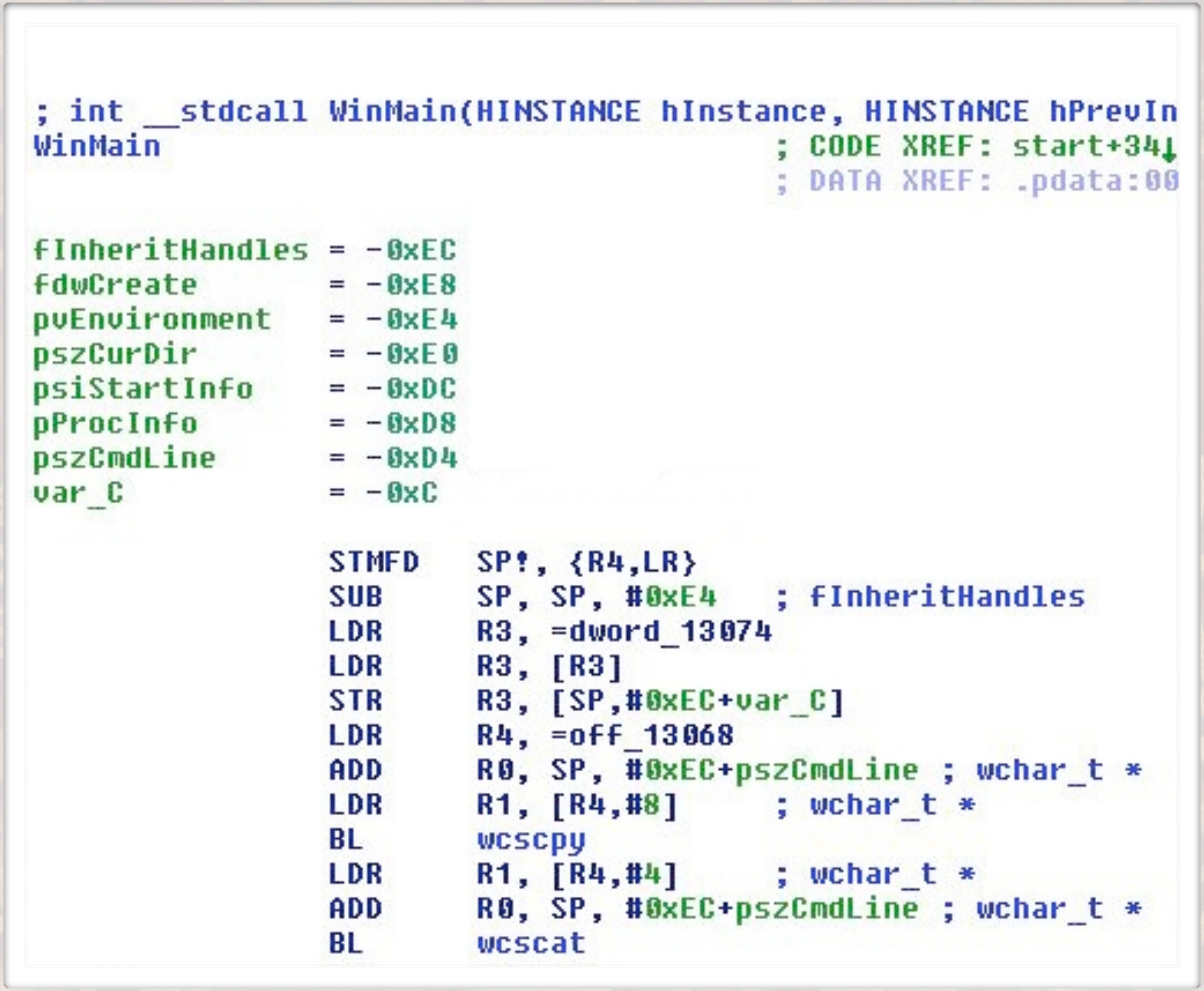
WM: First Steps

- Executable file format: PE
- Architecture: ARM
- Language: C++
- API: Windows-like

WM: Target

- Spanish “banking app”: BBVA
- https://www.bbva.es/TLBS/BBVA_MobiDesc.htm
- Versions for Java, RIM, Windows Mobile, iPhone and Android

WM: Analysis



```
; int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevIn
WinMain                                      ; CODE XREF: start+34↓
                                             ; DATA XREF: .pdata:00

fInheritHandles = -0xEC
fdwCreate       = -0xE8
pvEnvironment   = -0xE4
pszCurDir       = -0xE0
psiStartInfo    = -0xDC
pProcInfo       = -0xD8
pszCmdLine      = -0xD4
var_C           = -0xC

                STMFD   SP!, {R4,LR}
                SUB     SP, SP, #0xE4   ; fInheritHandles
                LDR     R3, =dword_13074
                LDR     R3, [R3]
                STR     R3, [SP,#0xEC+var_C]
                LDR     R4, =off_13068
                ADD     R0, SP, #0xEC+pszCmdLine ; wchar_t *
                LDR     R1, [R4,#8]     ; wchar_t *
                BL      wcscpy
                LDR     R1, [R4,#4]     ; wchar_t *
                ADD     R0, SP, #0xEC+pszCmdLine ; wchar_t *
                BL      wcscat
```

WM: Thoughts

- ARM can look a bit hard at first, but is much easier than x86!!
- The API is a piece of cake for people with experience in Windows XP/Vista development/reversing.
- Easy to debug the targets, emulate the OS and patch apps.

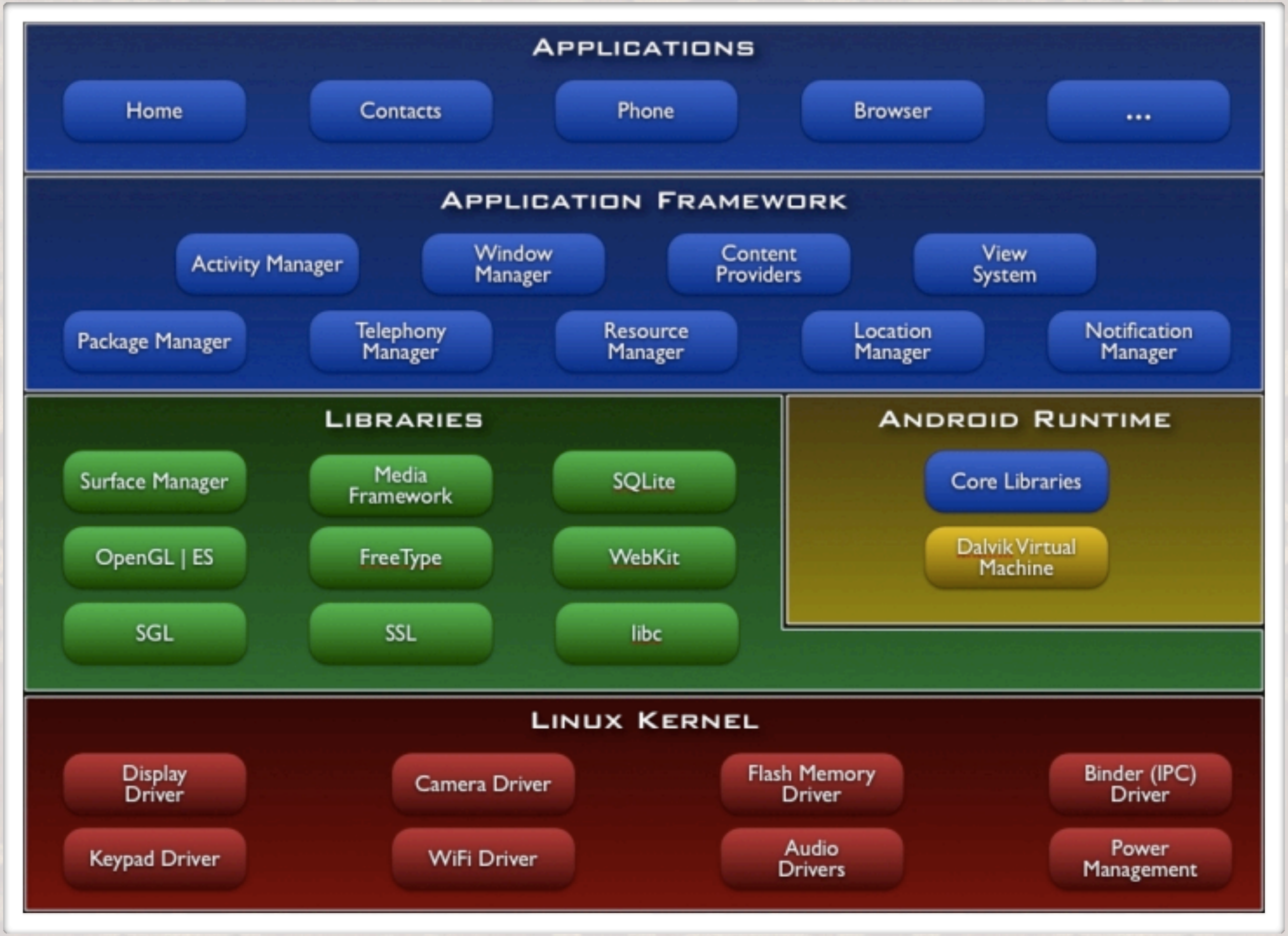
Google Android

Reversing a banking app: Wells Fargo

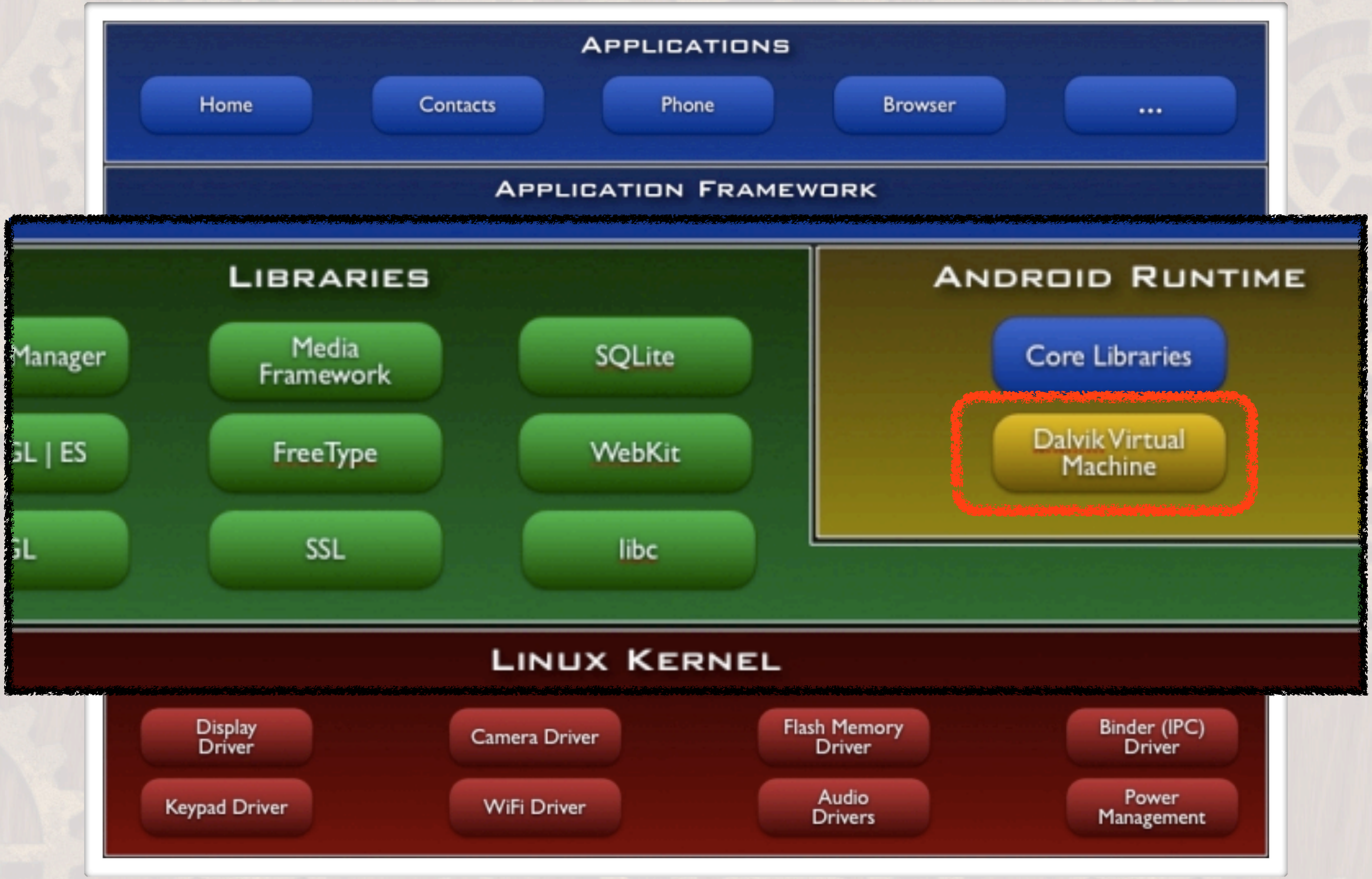
Android: First Steps

- Architecture: ARM
- API: Dalvik VM based
- Two ways of delevolment (SDK/NDK)
  - Executable file format: Dex/ELF
  - Language: Java/C++

Android: First Steps



Android: First Steps

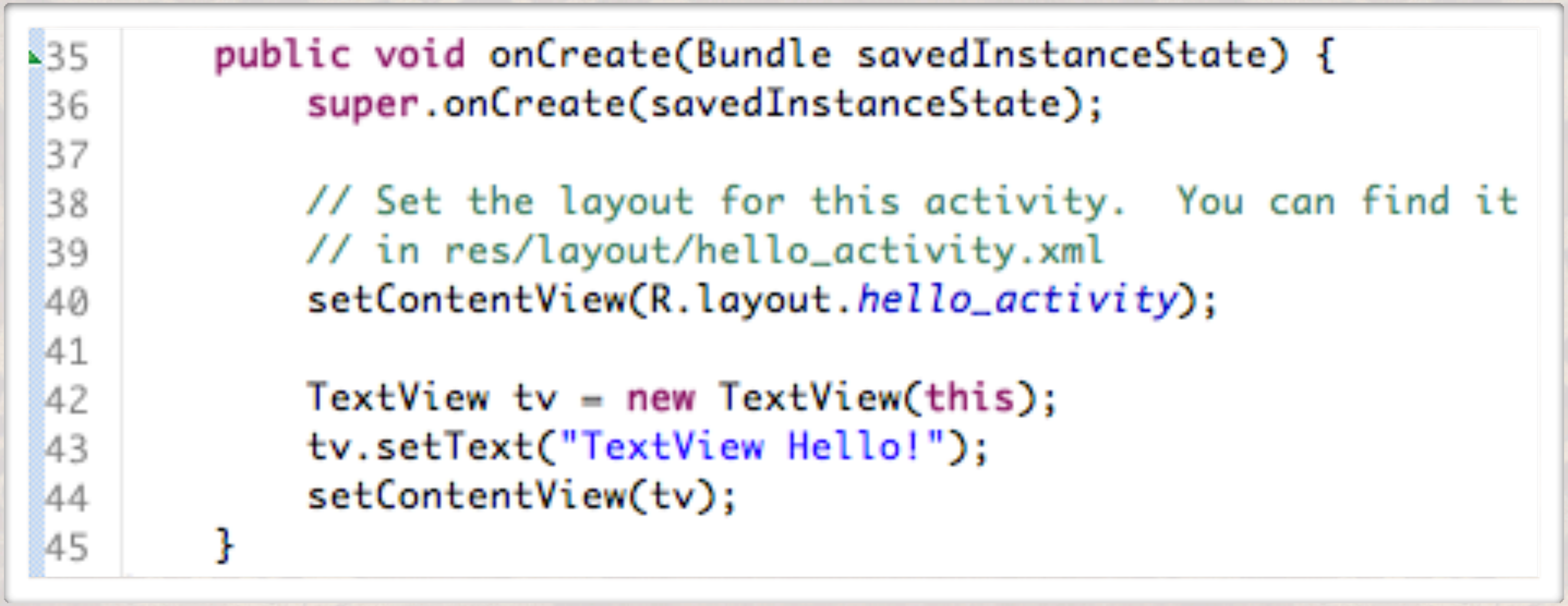


Android: First Steps

	Android	Java
Source	.java	.java
Binary	.dex	.class
Binary Optimized	.odex	N/A
Packages	.apk	.jar
Reversing Tools	DeDexer, Baksmali, DeoDexerant	JAD, DJ Java Decompiler, JD-GUI...

Android: Analysis

- Sample code



```
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Set the layout for this activity.  You can find it
        // in res/layout/hello_activity.xml
        setContentView(R.layout.hello_activity);

        TextView tv = new TextView(this);
        tv.setText("TextView Hello!");
        setContentView(tv);
    }
```

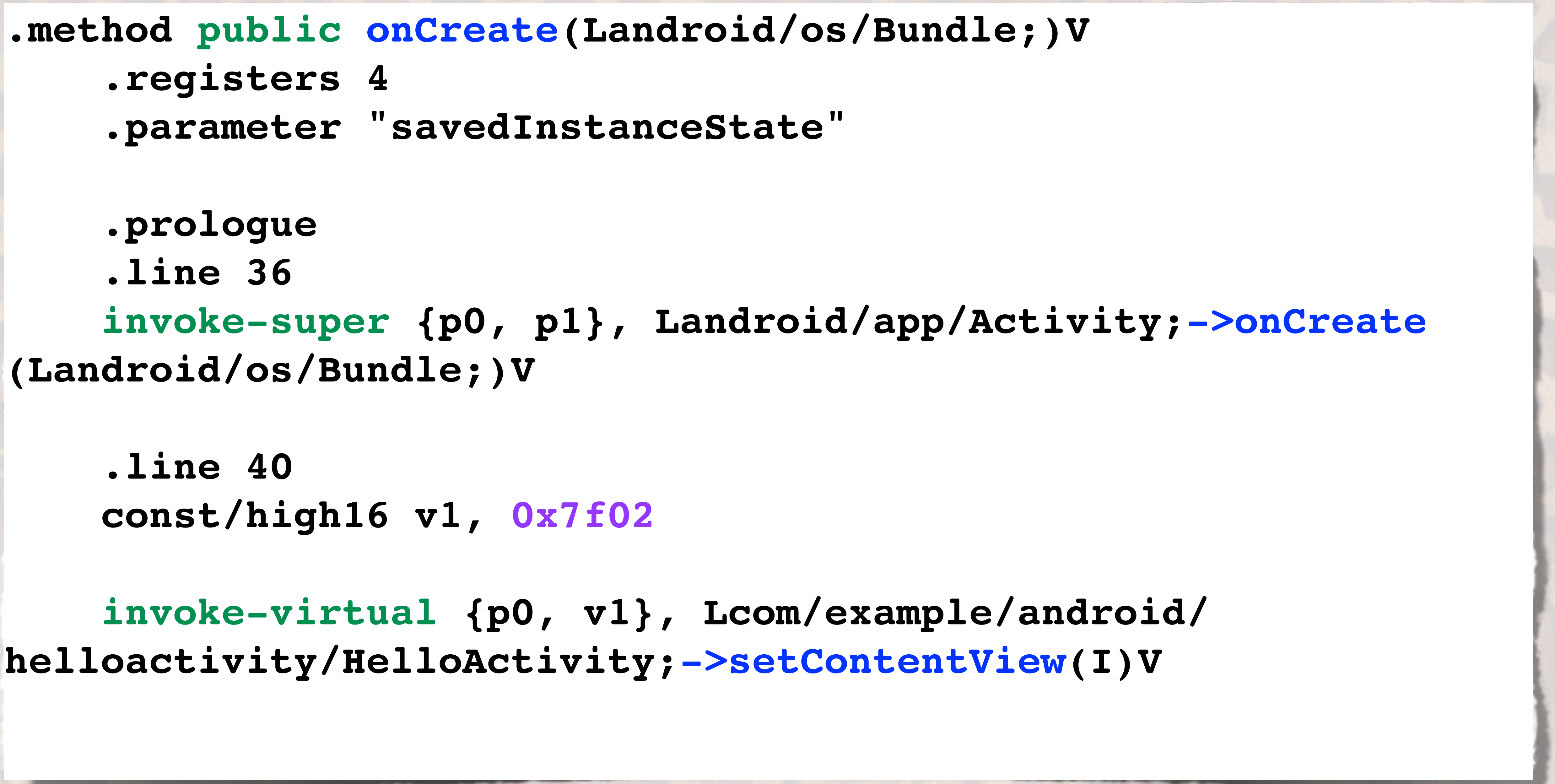
Android: Analysis

- DeDexer way

```
.method public onCreate(Landroid/os/Bundle;)V
.limit registers 4
; this: v2 (Lcom/example/android/helloactivity/HelloActivity;)
; parameter[0] : v3 (Landroid/os/Bundle;)
.line 36
        invoke-super    {v2,v3},android/app/Activity/
onCreate   ; onCreate(Landroid/os/Bundle;)V
.line 40
        const/high16    v1,32514
        invoke-virtual  {v2,v1},com/example/android/
helloactivity/HelloActivity/setContentView  ; setContentView
(I)V
```

Android: Analysis

- Baksmali way

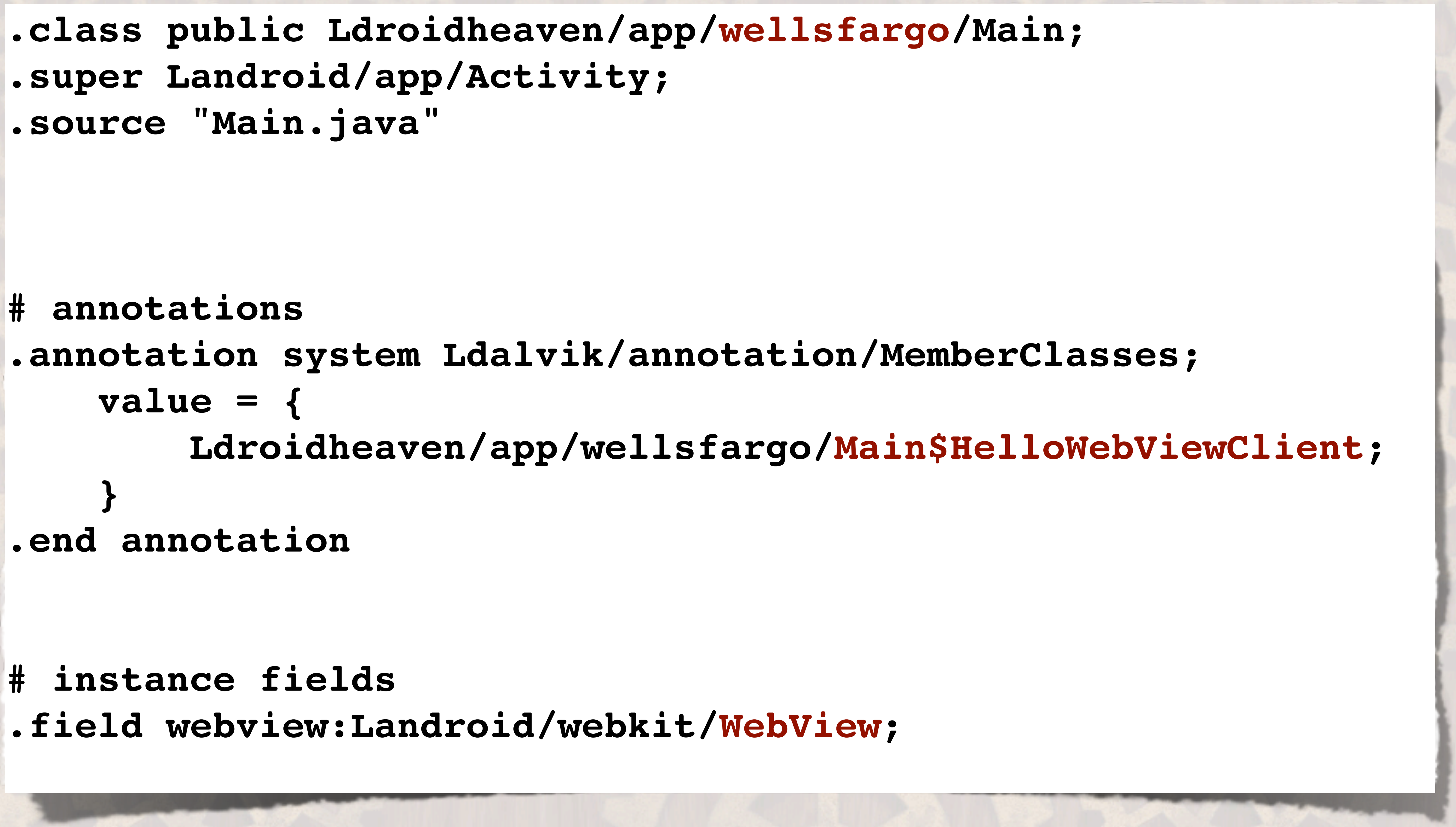


```
.method public onCreate(Landroid/os/Bundle;)V
    .registers 4
    .parameter "savedInstanceState"

    .prologue
    .line 36
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate
(Landroid/os/Bundle;)V

    .line 40
    const/high16 v1, 0x7f02

    invoke-virtual {p0, v1}, Lcom/example/android/
helloactivity/HelloActivity;->setContentView(I)V
```

Android: Target

- US banking app: Wells Fargo
- Downloaded from androlib.com
- Also available for iPhone

Android: Analysis II



```
.class public Ldroidheaven/app/wellsfargo/Main;
.super Landroid/app/Activity;
.source "Main.java"




# annotations
.annotation system Ldalvik/annotation/MemberClasses;
    value = {
        Ldroidheaven/app/wellsfargo/Main$HelloWebViewClient;
    }
.end annotation


# instance fields
.field webview:Landroid/webkit/WebView;
```

Android: Thoughts

- Decompilers are not ready yet
    - In the future, it will be similar to Java-reversing
- Free emulator:
    - It’s possible to install apk’s
    - No access to Android Market :(

Apple iPhone

Reversing a banking app: HanaBank

iPhone: First Steps

- Executable file format: Mach-O
- Architecture: ARM
- Language: Objective-C
- API: iPhone SDK Framework and libc

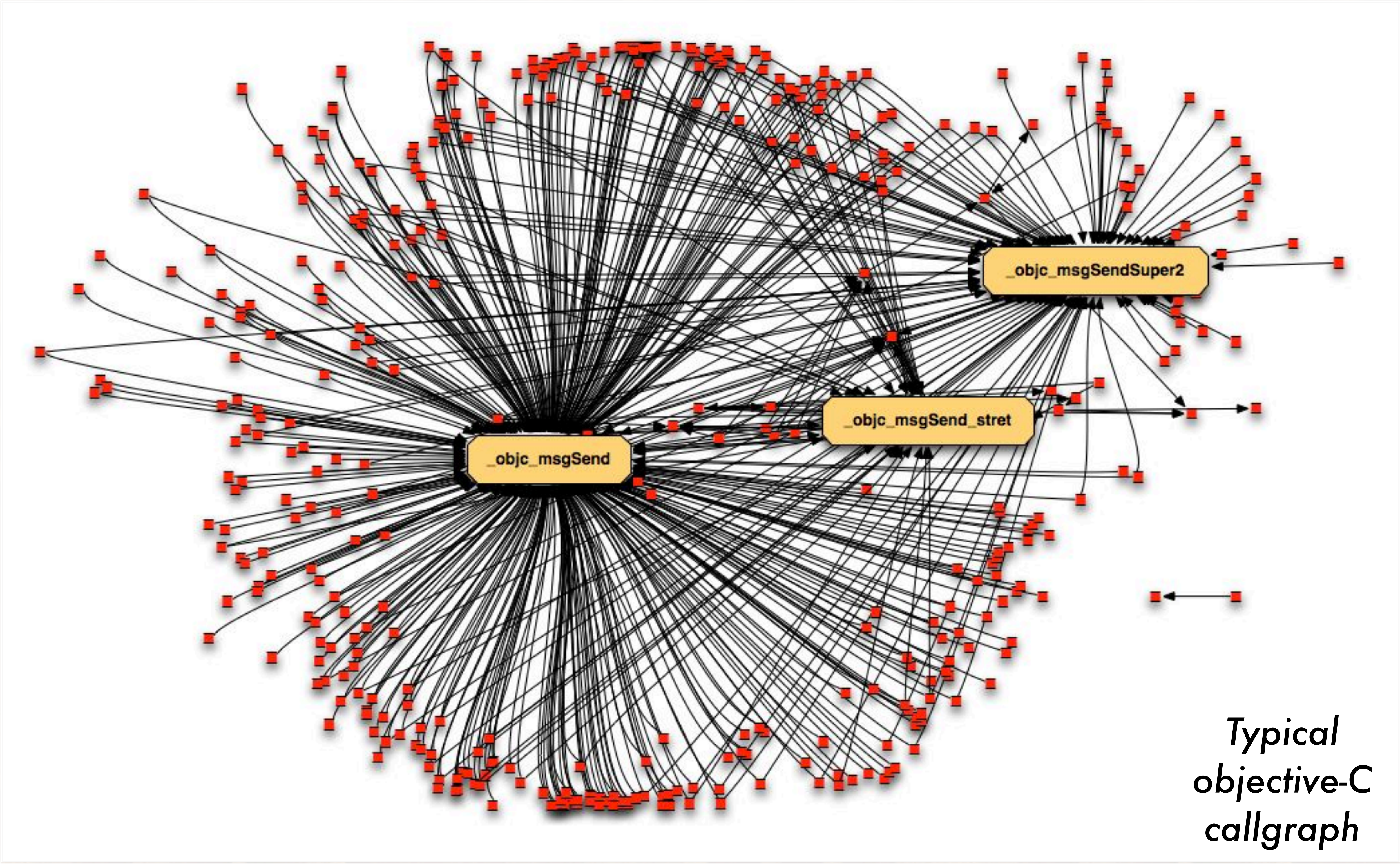
iPhone: First Steps

- Apps come in IPA packages
  - ZIP files with executables and resources (images, package info, config files...)
- iPhone Simulator works with i386
- Tricky to setup a debug environment
  - enable ssh access
  - iphonedbg/gdb

iPhone: First Steps

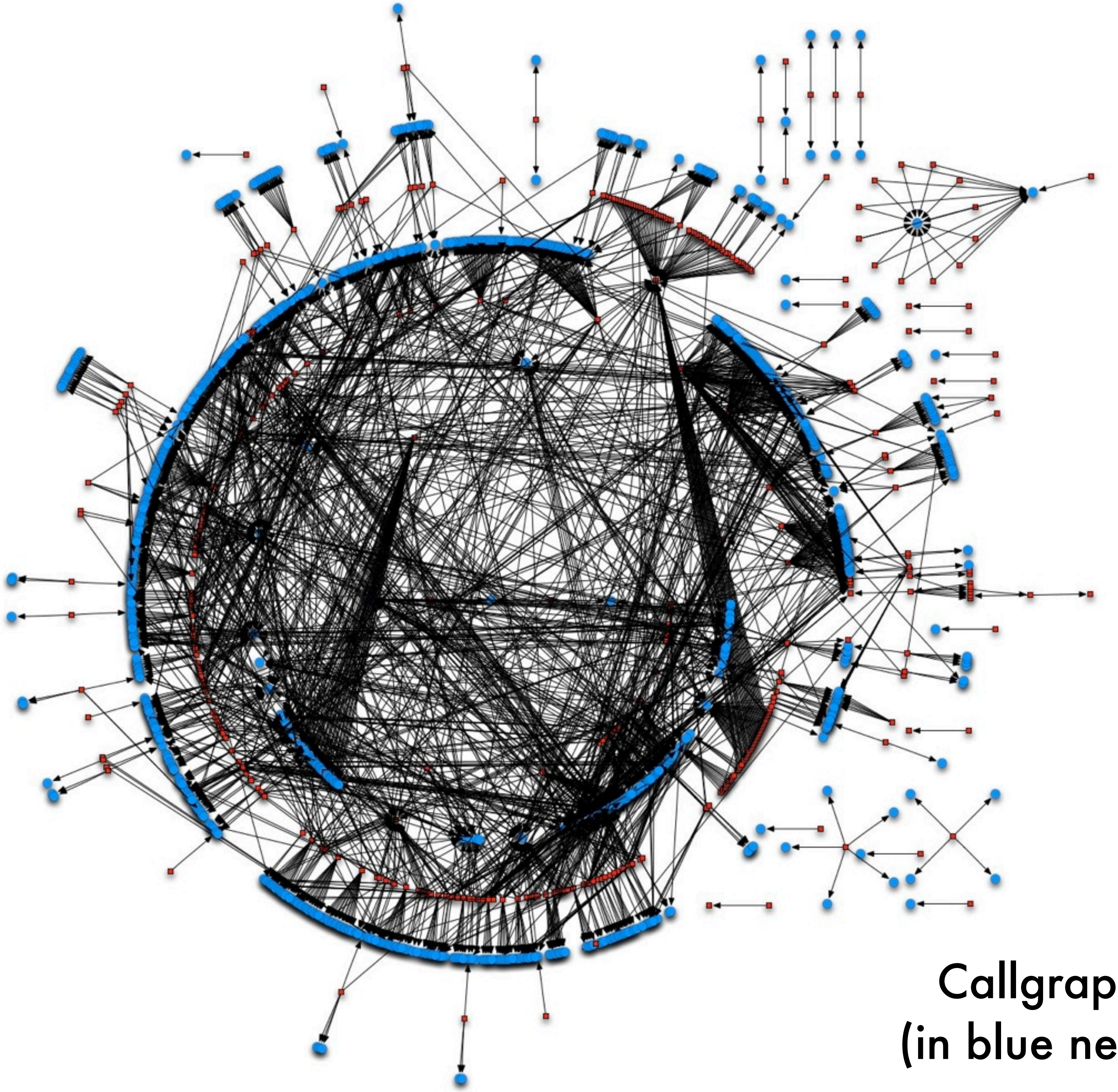
- AppStore
  - Apps are encrypted, iPhone Mach-O loader is in charge of decryption
  - GDB and a little understanding of Mach-O headers (otool) can help
- Objective-C
  - 80% calls to msgSend() == lots of fun :)

iPhone: First Steps



Typical objective-C callgraph

iPhone: First Steps



Callgraph after script
(in blue new “msg” subs)

iPhone: Target

- Korean banking app: HanaBank
- Downloaded from AppStore

iPhone: Analysis

- Removing AppStore encryption
- Playing with FLIRT signatures
- Reversing Objective-C code

iPhone: Thoughts

- AppStore DRM is not a big problem
- Objective-C can be quite painful
- Debug using GDB:
  - Well known tool, but you need delevoper account or a jailbroken device.
- Simulator is not useful for reversing

Questions?