iDecompile:
Writing an iOS Decompiler

Laurie Kirk

whoami

- Laurie Kirk
- Reverse Engineer
- Specialize in cross-platform malware with a focus on mobile threats
- Run YouTube channel @lauriewired

@lauriewired

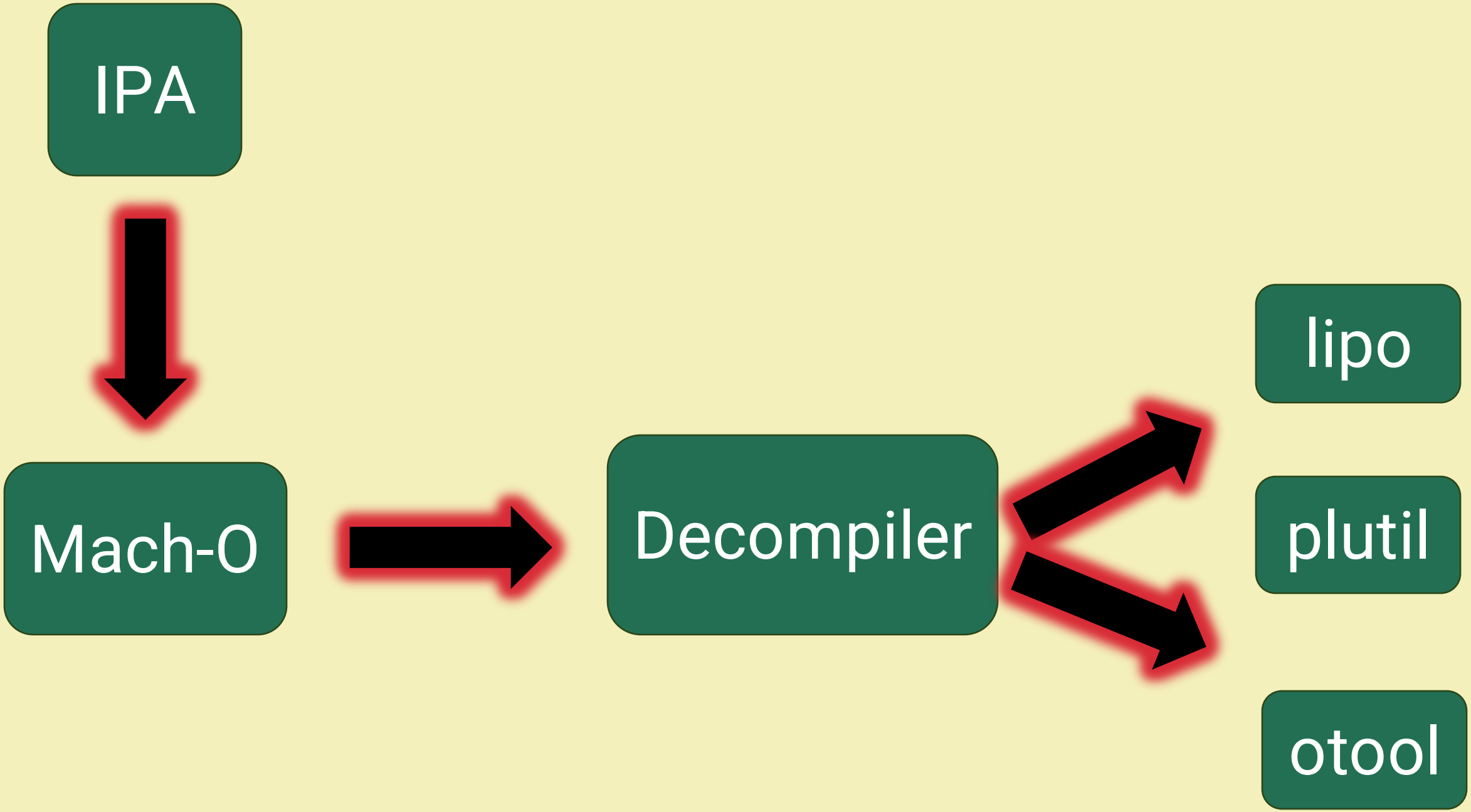
DECOMPILERS

THOUSANDS OF APPS

LITTLE RESEARCH

It’s Hard.


Melimite - Analysis
Classes
Files
Payload/ControlFlowFlattening.app/
Resources
ControlFlowFlattening
PkgInfo
_CodeSignature
CodeResources
Info.plist
Info.plist
ControlFlowFlattening
"CFBundleDevelopmentRegion": "en",
"MinimumOSVersion": "16.4",
"CFBundleIdentifier": "com.testing.ControlFlowFlattening",
"DTXcodeBuild": "14E300c",
"CFBundleExecutable": "ControlFlowFlattening",
"LSRequiresIPhoneOS": true,
"BuildMachineOSBuild": "22G68",
"UIApplicationSupportsIndirectInputEvents": true,
"CFBundlePackageType": "APPL",
"UISupportedInterfaceOrientations~iphone": [
"UIInterfaceOrientationPortrait",
"UIInterfaceOrientationLandscapeLeft",
"UIInterfaceOrientationLandscapeRight"
],
"DTCompiler": "com.apple.compilers.llvm.clang.1_0",
"CFBundleName": "ControlFlowFlattening",
"CFBundleSupportedPlatforms": [
"iPhoneSimulator"
],
"UISupportedInterfaceOrientations~ipad": [
"UIInterfaceOrientationPortrait",
"UIInterfaceOrientationPortraitUpsideDown",
"UIInterfaceOrientationLandscapeLeft",
"UIInterfaceOrientationLandscapeRight"
],
"DTPlatformBuild": "20E238",
"UIApplicationSceneManifest": {
"UISceneConfigurations": {},
"UIApplicationSupportsMultipleScenes": true
},
"CFBundleInfoDictionaryVersion": "6.0",
"DTSDKBuild": "20E238",
"DTXcode": "1431",
"CFBundleVersion": "1",
"DTSDKName": "iphonesimulator16.4",
"UIDeviceFamily": [
1,
2
],
"DTPlatformVersion": "16.4",
"UILaunchScreen": {
"UILaunchScreen": {}
},
"CFBundleShortVersionString": "1.0",
"DTPlatformName": "iphonesimulator"
File Analysis Report
File: ControlFlowFlattening.ipa
Size: 18 KB
Type: Single Architecture Mach-O
Mach-O Analysis
Language: Swift
Value Segment Label
Resource Strings
Value File Type
Function Assist
Select All
Model: OpenAI GPT-4 Turbo
Auto Fix
esc
tab
caps lock
shift
fn
control
option
command
delete
return
Settings
Display & Brightness
Home Screen & App Library
Search
StandBy
Wallpaper
Notifications
Sounds & Haptics
Focus
Screen Time
Face ID & Passcode
Emergency SOS
Privacy & Security
App Store
Game Center
iCloud
Wallet & Apple Pay


IPA
Mach-O
Decompiler
lipo
plutil
otool

Goal:
Simplify iOS Analysis

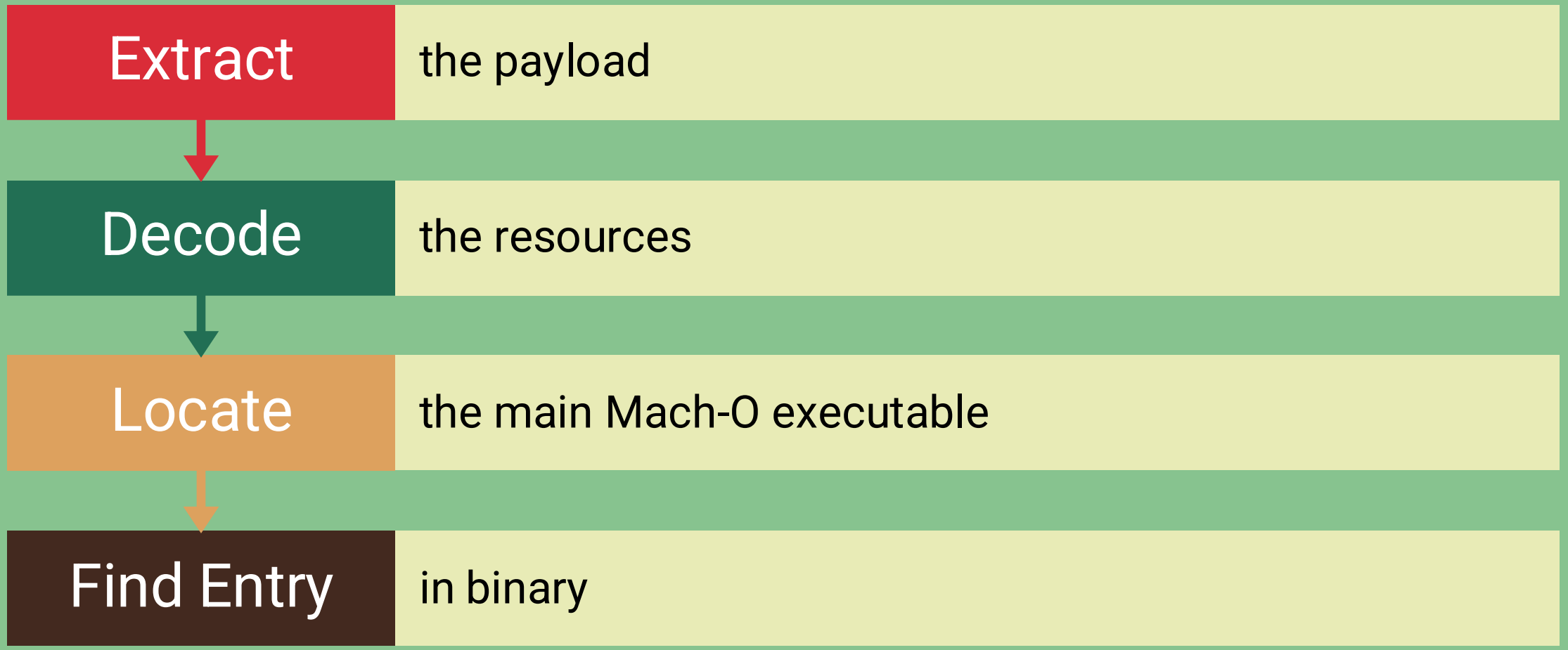
OUTCOME

- Platform-Agnostic iOS RE
- Reduce manual analysis
- Increase iOS app research

MALIMITE

IPA FILES

- Main application file for iOS
- Contains code, libs, permissions, and resources


Extract
the payload
Decode
the resources
Locate
the main Mach-O executable
Find Entry
in binary

Important
resources are neglected.

FOUR PILLARS

Permissions

Resources

Executable

Libraries / Frameworks

PLAN:

Analyze each pillar on multiple platforms

Pillar 1

Permissions

Understand

application capabilities

PRIMARY PERMISSION LOCATIONS

Info.plist

embedded.mobileprovision

INFO.PLIST

- Permissions granted by the user at runtime
- Example: camera, location

EMBEDDED.MOBILEPROVISION

- Entitlements granted by the system
- Example: iCloud, push notifications

PROPERTY LISTS

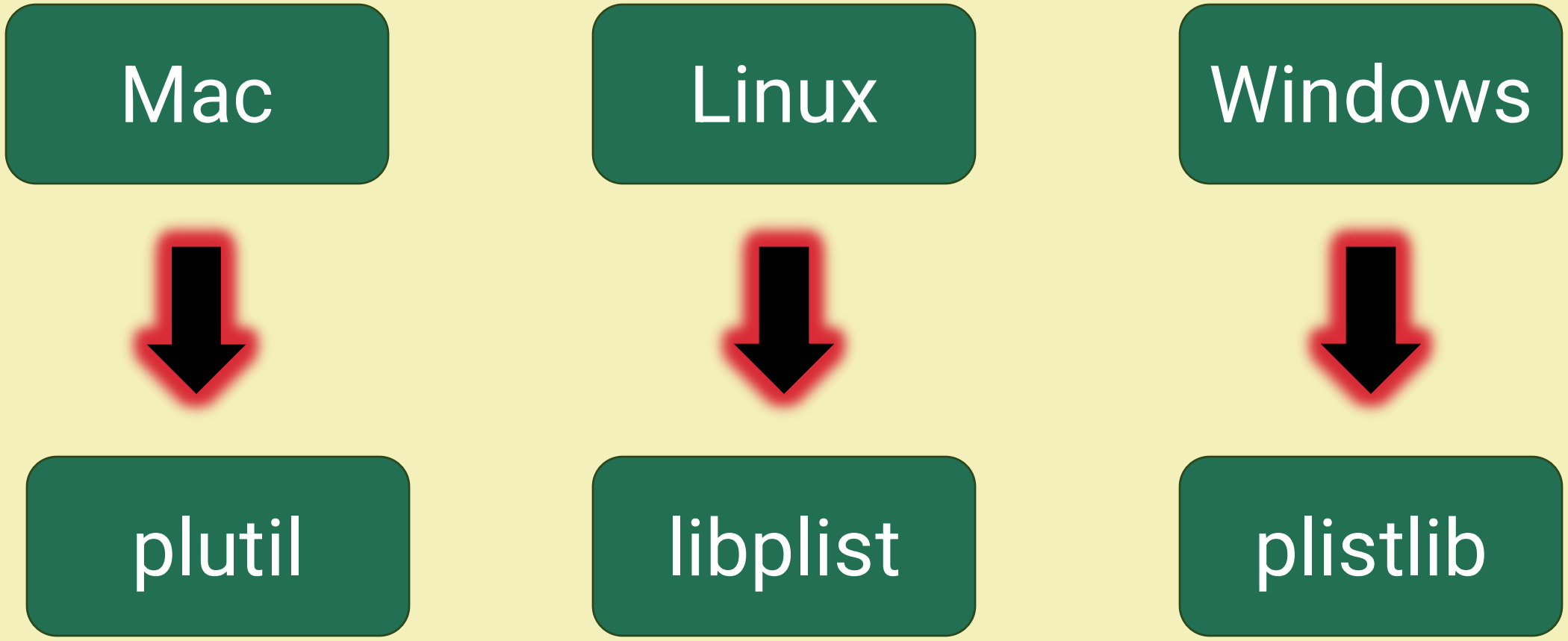
- Binary or XML formats
- More than just the Info.plist!

XML PLISTS ARE SIMPLE

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>CFBundleName</key>
    <string>MyApp</string>
    <key>CFBundleIdentifier</key>
    <string>com.example.myapp</string>
    <key>CFBundleVersion</key>
    <string>1.0</string>
    <key>CFBundleShortVersionString</key>
    <string>1.0.0</string>
    <key>NSCameraUsageDescription</key>
    <string>This app requires access to the camera to take photos.</string>
    <key>NSLocationWhenInUseUsageDescription</key>
    <string>This app uses your location to provide recommendations.</string>
    <key>UIBackgroundModes</key>
    <array>
        <string>location</string>
        <string>audio</string>
    </array>
</dict>
</plist>
```

Binary plists are a custom Apple format.

DECODING LIBRARIES



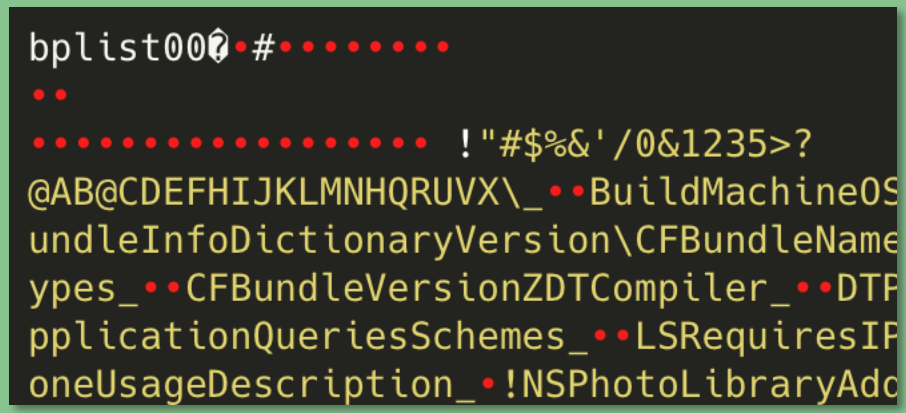
Unique

tool per platform?

PROPRIETARY PLIST REVERSED

```
private static int determineType(InputStream is, int offset) throws IOException {
    int index = offset;
    int readLimit = index + 1024;
    if (is.markSupported()) {
        is.mark(readLimit);
    }

    is.skip(offset);
    int b;
    ByteOrderMarkReader bomReader = new ByteOrderMarkReader();
    boolean bom = true;
    //Skip any possible whitespace at the beginning of the file
    do {
        if (++index > readLimit) {
            is.reset();
            return determineType(is, readLimit);
        }
        b = is.read();
        //Check if we are reading the Unicode byte order mark (BOM) and skip it
        bom &= bomReader.readByte(b);
    } while (b != -1 && (b == ' ' || b == '\t' || b == '\r' || b == '\n' || b == '\f' || bom));
```




```
"MinimumOSVersion": "11.1",
"UIStatusBarStyle": "UIStatusBarStyleDefault",
"CFBundleIdentifier": "fun.Animoji",
"DTXcodeBuild": "9C40b",
"NSMicrophoneUsageDescription": "I need access to the microphone to record audio",
"CFBundleExecutable": "AnimojiStudio",
"LSRequiresIPhoneOS": true,
"NSPhotoLibraryAddUsageDescription": "I need access to your photo library to save your video",
"BuildMachineOSBuild": "17C88",
"CFBundlePackageType": "APPL",
"LSApplicationQueriesSchemes": [
  "spotify-action"
```

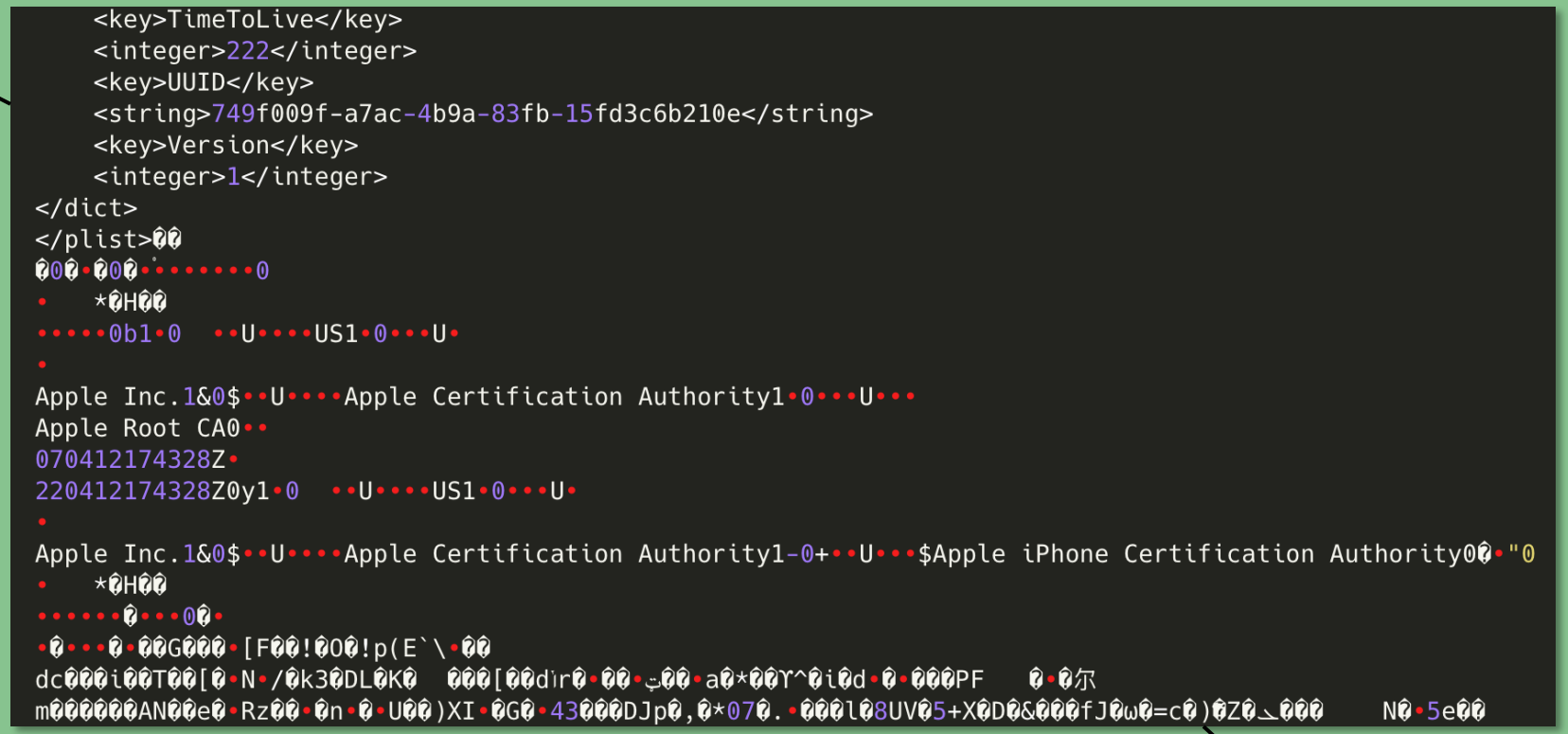
EMBEDDED.MOBILPROVISION

- Part property list
- Part PKCS #7 signature
  - Signed by Apple

Less popular,

less research

Part plist



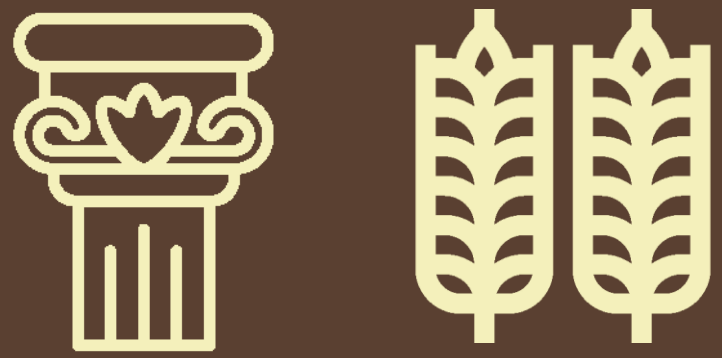
Part certificate

Extract and decode plist

embedded.mobileprovision

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
<plist version="1.0">
<dict>
  <key>AppIDName</key>
  <string>AnimojiStudio</string>
  <key>ApplicationIdentifierPrefix</key>
  <array>
    <string>8C7439RJLG</string>
  </array>
  <key>CreationDate</key>
  <date>2017-11-14T14:43:41Z</date>
  <key>Platform</key>
  <array>
    <string>iOS</string>
  </array>
  <key>DeveloperCertificates</key>
  <array>
    <data>
      MIIFoTCCBImgAwIBAgIILokx/b0wKe0wDQYJKoZIhvcNAQE
    </data>
  </array>
  <key>Entitlements</key>
  <dict>
    <key>keychain-access-groups</key>
    <array>
      <string>8C7439RJLG.*</string>
    </array>
    <key>get-task-allow</key>
```

Pillar 2



Resources

Supporting **resources** are spread throughout the IPA

Various proprietary formats.

PLISTS

- Sometimes encoded
- Metadata, identifiers, URL schemes

Plists

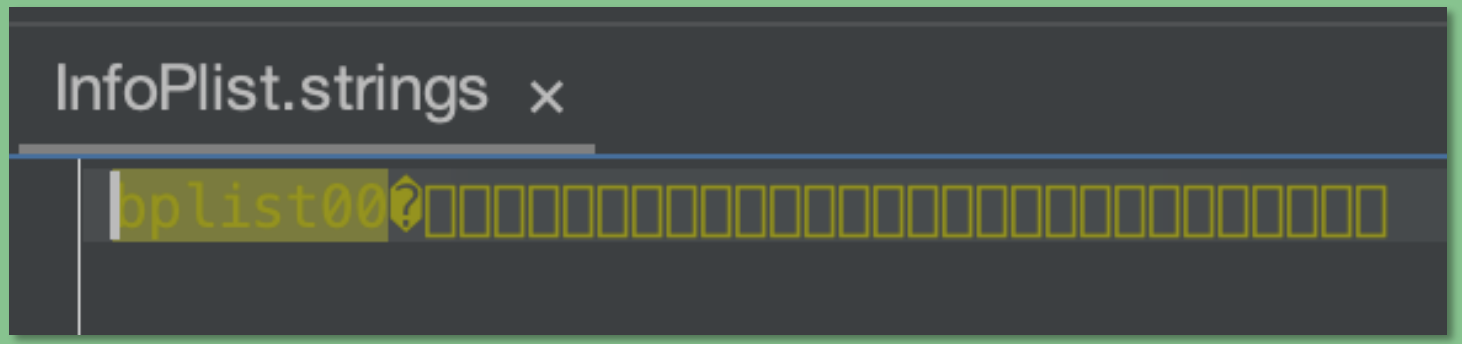
LOCALIZED STRINGS

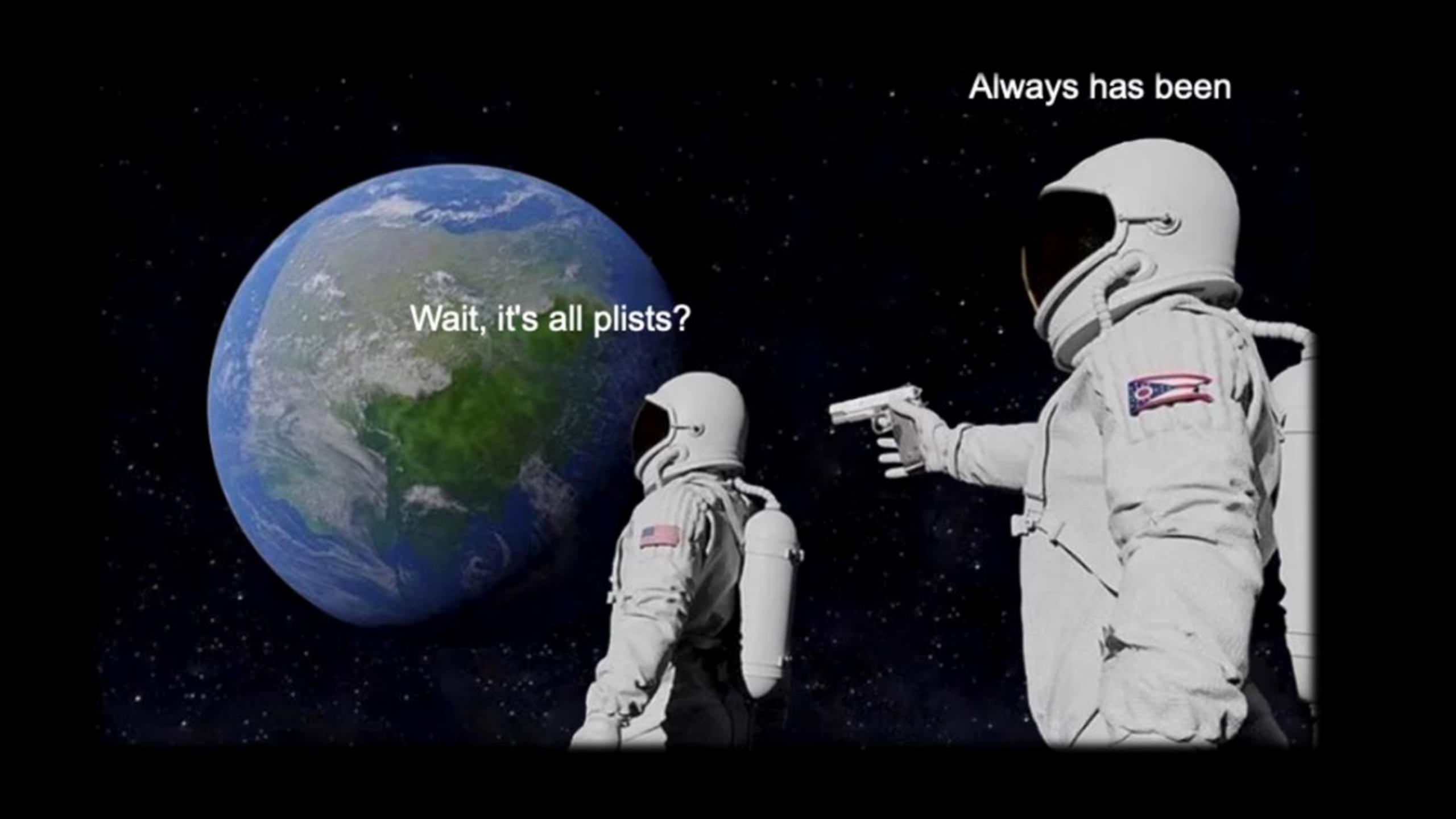
- User-facing text
- Labels, error messages, dev strings

LOCALIZED STRING FILES

- Stored in .strings
- Multiple languages bundled into IPA

This looks ***familiar....***




Always has been
Wait, it's all plists?

Localized Strings

THE RESOURCES NEVER END

- Nib
- Xib
- Storyboard

Parse *strings* from encoded files.

Resource Strings

Value	File	Type
"CFBundleShortVersionString": "1.0",	ObjectAL-Info.plist	plist
"NSPrincipalClass": "",	ObjectAL-Info.plist	plist
"CFBundleIdentifier": "org.stenerud.${PRODUCT_NAME:rfc1034...	ObjectAL-Info.plist	plist
"CFBundleName": "${PRODUCT_NAME}",	ObjectAL-Info.plist	plist
"CFBundleInfoDictionaryVersion": "6.0",	ObjectAL-Info.plist	plist
"CFBundleExecutable": "${EXECUTABLE_NAME}",	ObjectAL-Info.plist	plist
"CFBundleSignature": "????"	ObjectAL-Info.plist	plist
"rules": {	ResourceRules.plist	plist
".*": true,	ResourceRules.plist	plist
"ResourceRules.plist": {	ResourceRules.plist	plist
"weight": 100.0,	ResourceRules.plist	plist
"omit": true	ResourceRules.plist	plist
"Info.plist": {	ResourceRules.plist	plist
"weight": 10.0,	ResourceRules.plist	plist
"omit": true	ResourceRules.plist	plist

Pillar 3

Executable

INFO.PLIST

- Defines the main executable binary

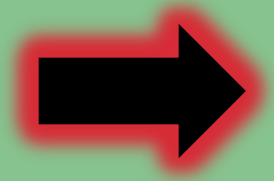
```
<key>CFBundleDisplayName</key>
<string>Flappy Bird</string>
<key>CFBundleExecutable</key>
<string>flap</string>
```

ACCOUNT FOR UNIVERSAL BINARIES

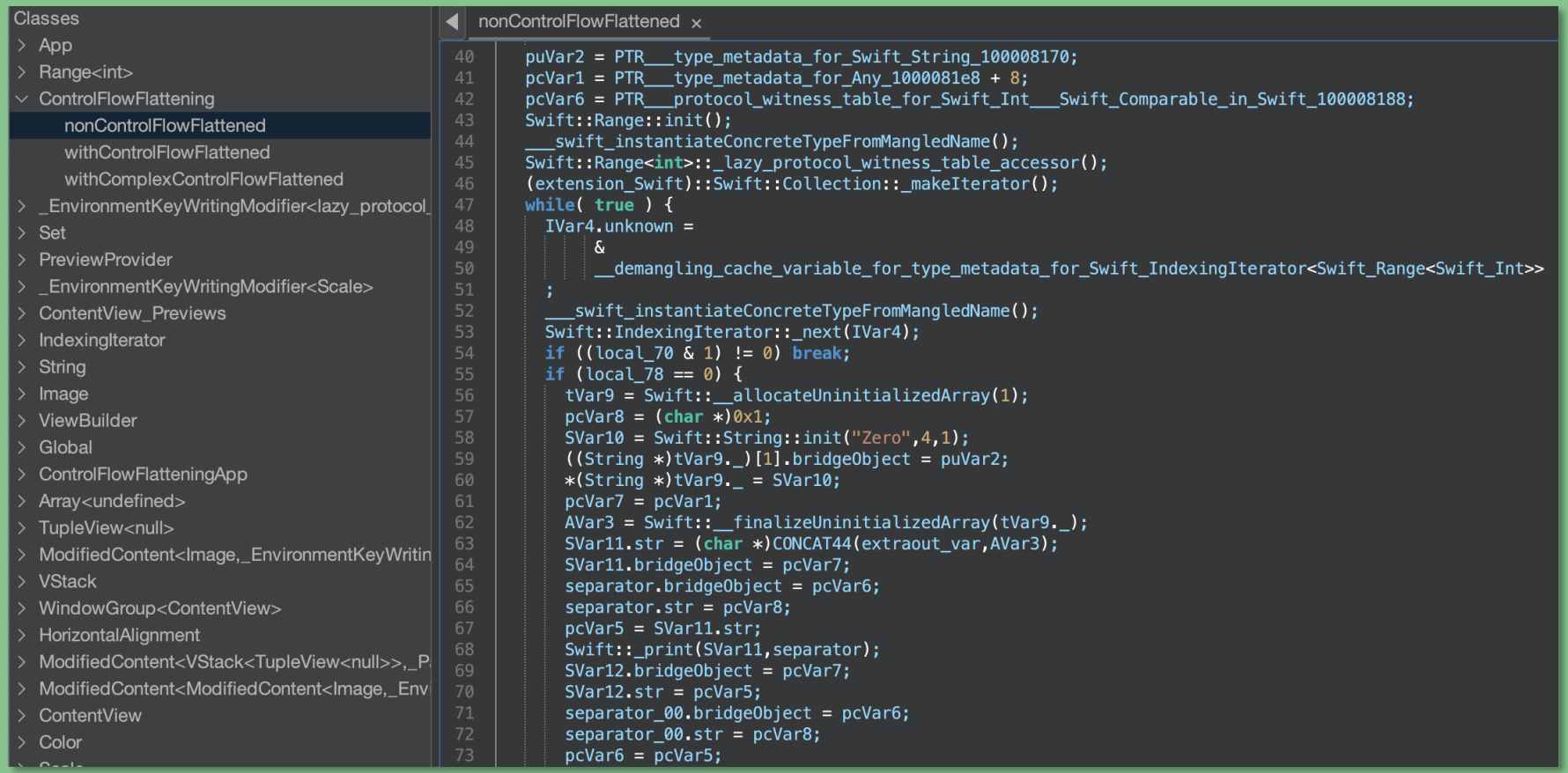
Field	Size	Description
magic	4 bytes	Magic number (indicates fat binary type).
nfat_arch	4 bytes	Number of architectures in the binary.
cputype	4 bytes	CPU type (e.g., arm64, x86_64).
cpusubtype	4 bytes	CPU subtype (e.g., armv7, arm64e).
offset	4 bytes	Offset to the Mach-O binary for this arch.
size	4 bytes	Size of the Mach-O binary for this arch.
align	4 bytes	Alignment of the binary (power of 2).

EXTRACT ARCH SLICE FOR DECOMP

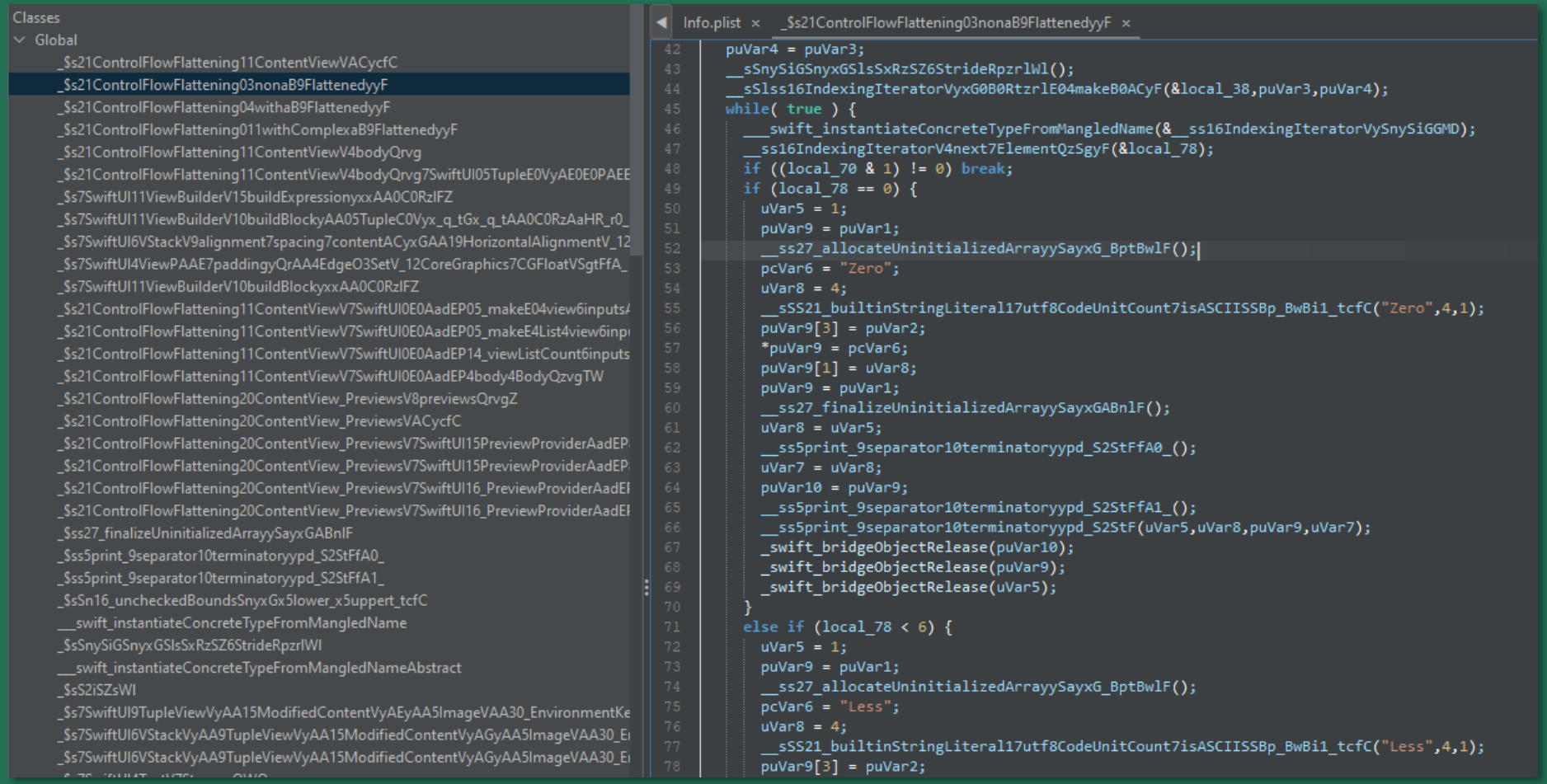
```
laurie@Mac AirRec.app % otool -hv AirRec
AirRec (architecture armv7):
Mach header
      magic  cputype cpusubtype  caps    filetype ncmds sizeofcmds
      flags
   MH_MAGIC      ARM         V7  0x00     EXECUTE    31       3520
  NOUNDEFS DYLDLINK TWOLEVEL WEAK_DEFINES BINDS_TO_WEAK PIE
AirRec (architecture arm64):
Mach header
      magic  cputype cpusubtype  caps    filetype ncmds sizeofcmds
      flags
MH_MAGIC_64    ARM64        ALL  0x00     EXECUTE    31       4048
  NOUNDEFS DYLDLINK TWOLEVEL WEAK_DEFINES BINDS_TO_WEAK PIE
```



LOOKS GREAT ON MAC!



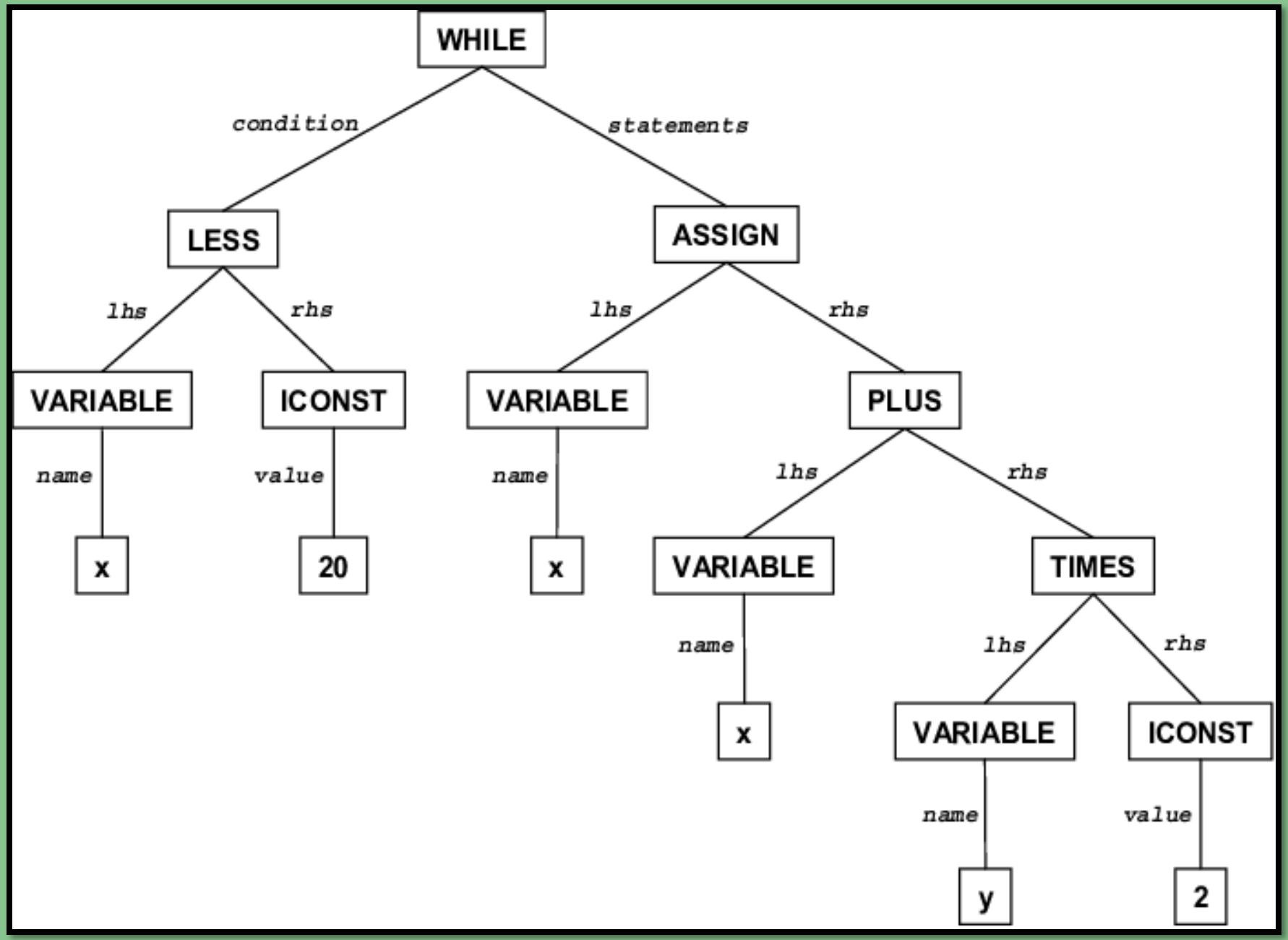
NOT SO GREAT ELSEWHERE...



Parse code to

reconstruct

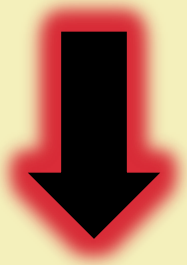
classes


WHILE
condition
statements
LESS
ASSIGN
lhs
rhs
lhs
rhs
VARIABLE
ICONST
VARIABLE
PLUS
name
value
name
lhs
rhs
x
20
x
VARIABLE
TIMES
name
lhs
rhs
x
VARIABLE
ICONST
name
value
y
2

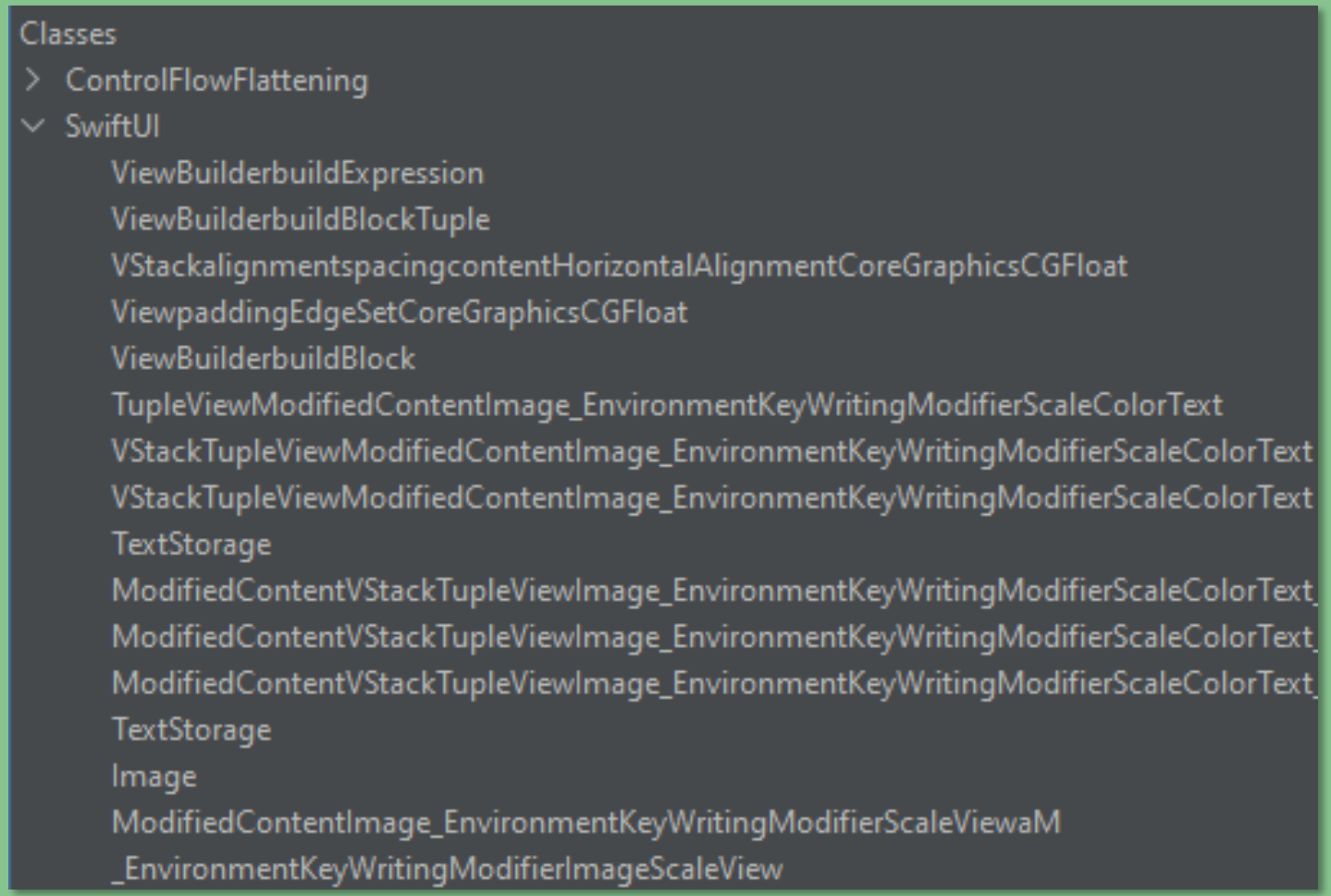
MODIFIED C++ GRAMMAR

- Handle Swift namespace declarations
- Tolerate incorrectly formatted code

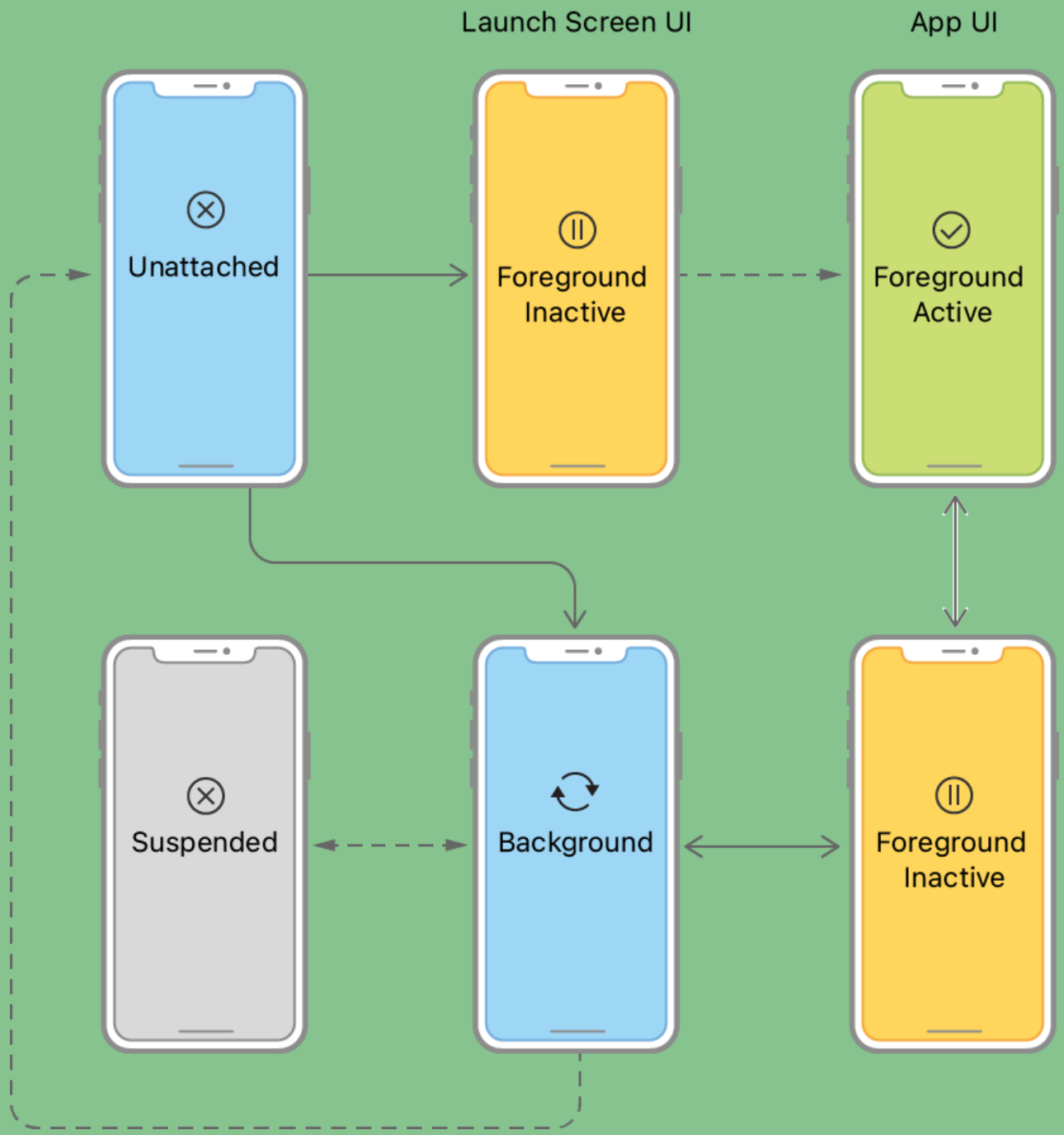
s21ControlFlowFlattening03nonaB9FlattenedyyF



void ControlFlowFlattening::nonControlFlowFlattened (void)


Classes
ControlFlowFlattening
SwiftUI
ViewBuilderbuildExpression
ViewBuilderbuildBlockTuple
VStackalignmentspacingcontentHorizontalAlignmentCoreGraphicsCGFloat
ViewpaddingEdgeSetCoreGraphicsCGFloat
ViewBuilderbuildBlock
TupleViewModifiedContentImage_EnvironmentKeyWritingModifierScaleColorText
VStackTupleViewModifiedContentImage_EnvironmentKeyWritingModifierScaleColorText
VStackTupleViewModifiedContentImage_EnvironmentKeyWritingModifierScaleColorText
TextStorage
ModifiedContentVStackTupleViewImage_EnvironmentKeyWritingModifierScaleColorText_
ModifiedContentVStackTupleViewImage_EnvironmentKeyWritingModifierScaleColorText_
ModifiedContentVStackTupleViewImage_EnvironmentKeyWritingModifierScaleColorText_
TextStorage
Image
ModifiedContentImage_EnvironmentKeyWritingModifierScaleViewaM
_EnvironmentKeyWritingModifierImageScaleView

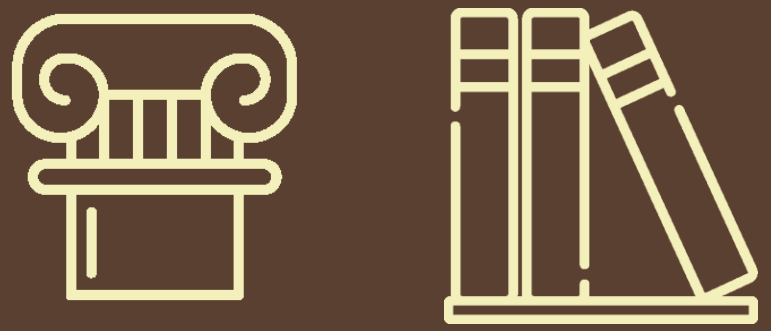
Where do we start analysis?


Launch Screen UI
App UI
Unattached
Foreground Inactive
Foreground Active
Suspended
Background
Foreground Inactive

COMMON ENTRYPOINTS

- applicationDidFinishLaunchingWithOptions
- applicationDidBecomeActive
- applicationDidEnterBackground
- applicationWillTerminate

Pillar 4

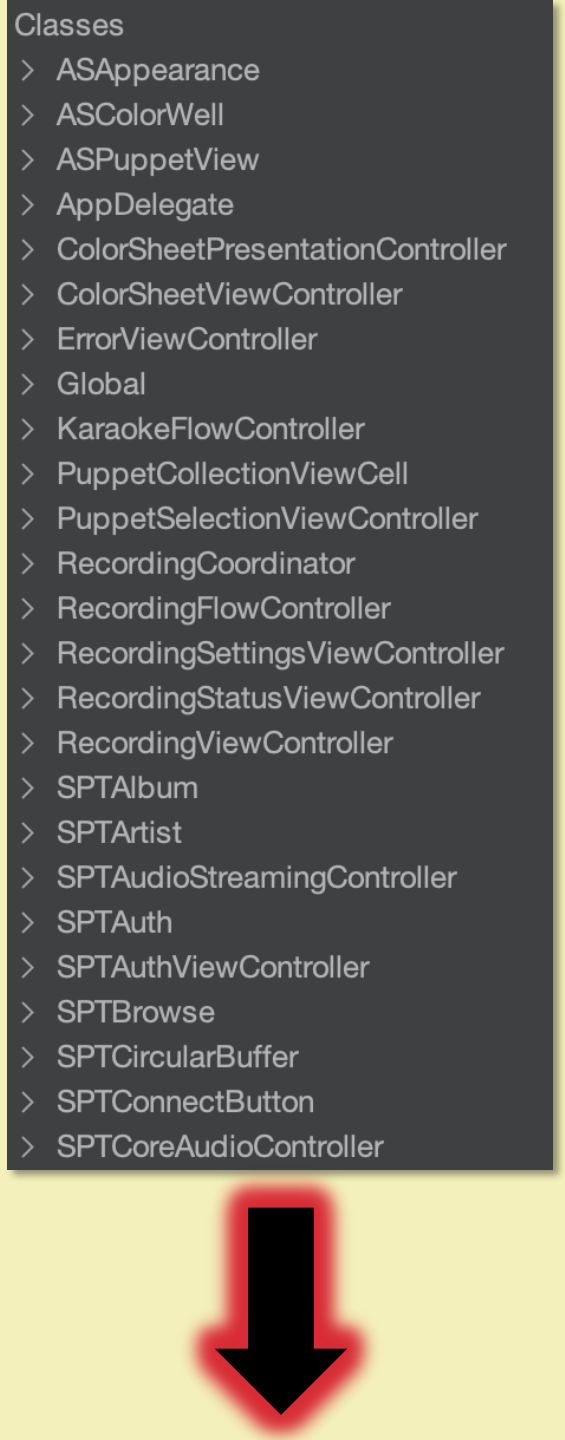


Libraries and Frameworks

Why are iOS apps so

LARGE ?

Libraries and frameworks are often bundled.



HANDS ON:

Wait for this app
to decompile

How can we

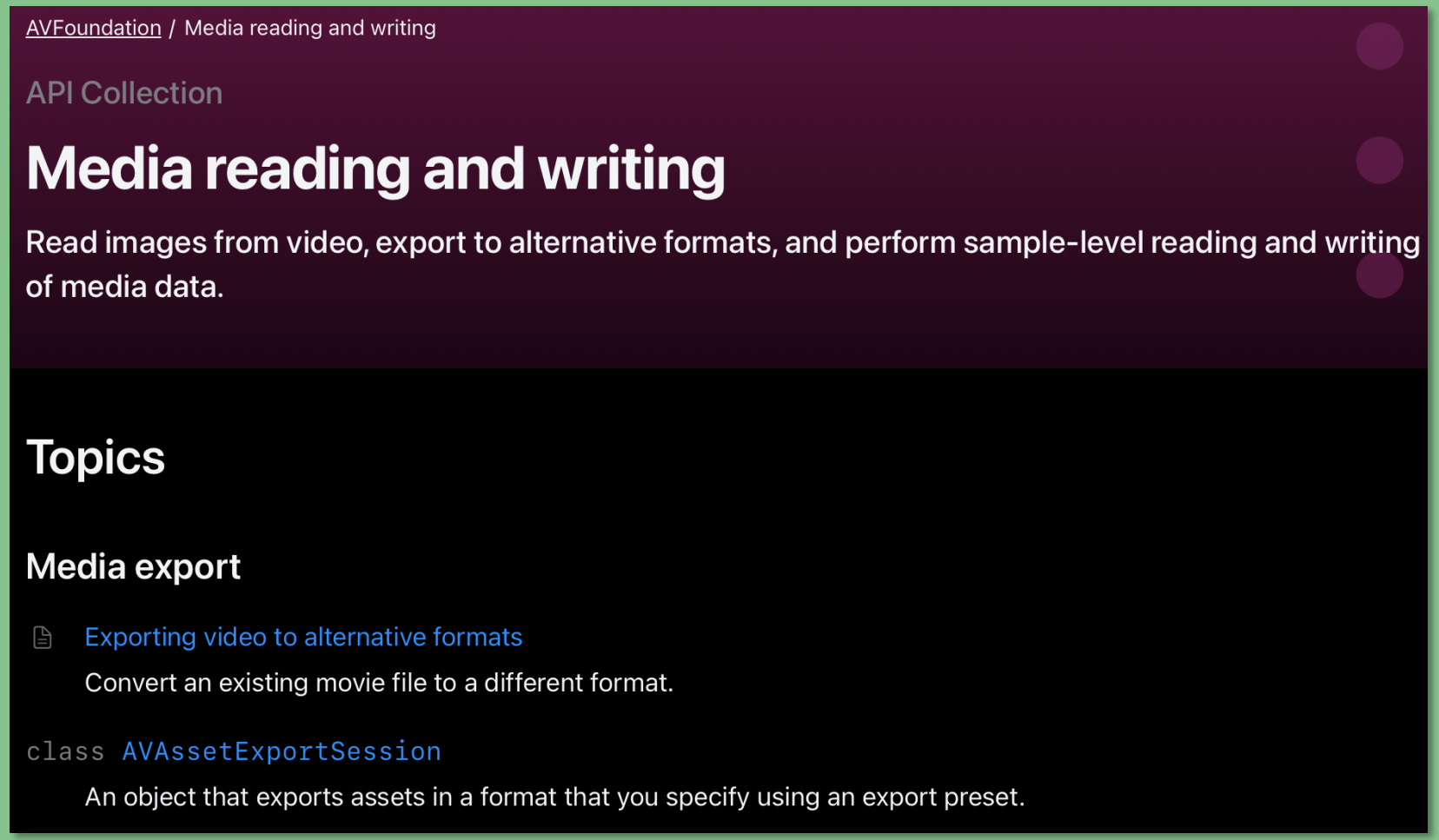
distinguish

libraries?

AVFoundation

AVFoundation is the full featured framework for working with time-based audiovisual media on iOS, macOS, watchOS and tvOS. Using AVFoundation, you can easily play, create, and edit QuickTime movies and MPEG-4 files, play HLS streams, and build powerful media functionality into your apps.

READ DOCUMENTATION, NOT DECOMPILATION



RUNTIME METHODS

```
local_38 = PTR_PTR_1000af8f0;
local_40 = param_1;
_objc_msgSendSuper2(&local_40,PTR_s_viewDidLoad_1000ad278);
puVar1 = PTR_s_view_1000ad2a8;
_objc_msgSend(param_1,PTR_s_view_1000ad2a8);
uVar2 = _objc_retainAutoreleasedReturnValue();
_objc_msgSend(PTR__OBJC_CLASS___UIColor_1000af460,PTR_s_whiteColor_1000ad5d8);
uVar3 = _objc_retainAutoreleasedReturnValue();
_objc_msgSend(uVar2,PTR_s_setBackgroundColor__1000ad5e0,uVar3);
_objc_release(uVar3);
_objc_release(uVar2);
_objc_msgSend(param_1,puVar1);
uVar2 = _objc_retainAutoreleasedReturnValue();
_objc_msgSend(uVar2,PTR_s_setOpaque__1000ad5e8,1);
_objc_release(uVar2);
_objc_msgSend(param_1,PTR_s_navigationItem_1000ade38);
uVar2 = _objc_retainAutoreleasedReturnValue();
_objc_msgSend(uVar2,PTR_s_setLargeTitleDisplayMode__1000ae4b8,2);
_objc_release(uVar2);
_objc_msgSend(param_1,PTR_s_setTitle__1000ad5f0,&cf_Record);
return;
```

RUNTIME METHODS

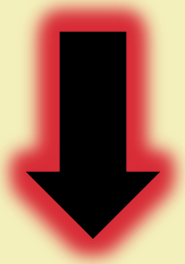
Objective-C

- objc_msgSend
- objc_release

Swift

- swift_demangle
- swift_allocObject

```
Class cls = swift_getClassByName("MyApp.MyClass");
void *instance = swift_allocObject(cls, sizeof(MyClass), alignof(MyClass));
```



```
let instance = MyClass()
```

Pillar 1

Permissions ✓

Pillar 2

Resources ✓

Pillar 3

Executable ✓

Pillar 4

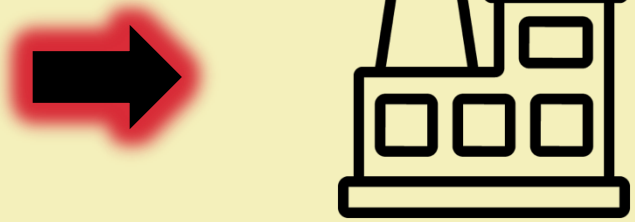
Libraries and Frameworks ✓

One more thing…

```
Swift::Range<int>::_lazy_protocol_witness_table_accessor();
(extension_Swift)::Swift::Collection::_makeIterator();
while( true ) {
  IVar4.unknown =
       &
       __demangling_cache_variable_for_type_metadata_for_Swift_IndexingIterator<Swift_Range<Swift_Int>>
  ;
  ___swift_instantiateConcreteTypeFromMangledName();
  Swift::IndexingIterator::_next(IVar4);
  if ((local_70 & 1) != 0) break;
  if (local_78 == 0) {
    tVar9 = Swift::__allocateUninitializedArray(1);
    pcVar8 = (char *)0x1;
    SVar10 = Swift::String::init("Zero",4,1);
    ((String *)tVar9._)[1].bridgeObject = puVar2;
    *(String *)tVar9._ = SVar10;
    pcVar7 = pcVar1;
    AVar3 = Swift::__finalizeUninitializedArray(tVar9._);
    SVar11.str = (char *)CONCAT44(extraout_var,AVar3);
    SVar11.bridgeObject = pcVar7;
    separator.bridgeObject = pcVar6;
    separator.str = pcVar8;
    pcVar5 = SVar11.str;
    Swift::_print(SVar11,separator);
    SVar12.bridgeObject = pcVar7;
    SVar12.str = pcVar5;
    separator_00.bridgeObject = pcVar6;
    separator_00.str = pcVar8;
    pcVar6 = pcVar5;
    pcVar8 = pcVar7;
    Swift::_print(SVar12,separator_00);
    SVar13.bridgeObject = pcVar5;
    SVar13.str = SVar11.str;
    separator_01.bridgeObject = pcVar6;
    separator_01.str = pcVar7;
    Swift::_print(SVar13,separator_01);
    _swift_bridgeObjectRelease(pcVar8);
    _swift_bridgeObjectRelease(pcVar7);
    _swift_bridgeObjectRelease(SVar11.str);
  }
```

APPROXIMATE CODE IN TARGET LANGUAGES

C-Pseudocode



LLM Processing

Swift Approximation

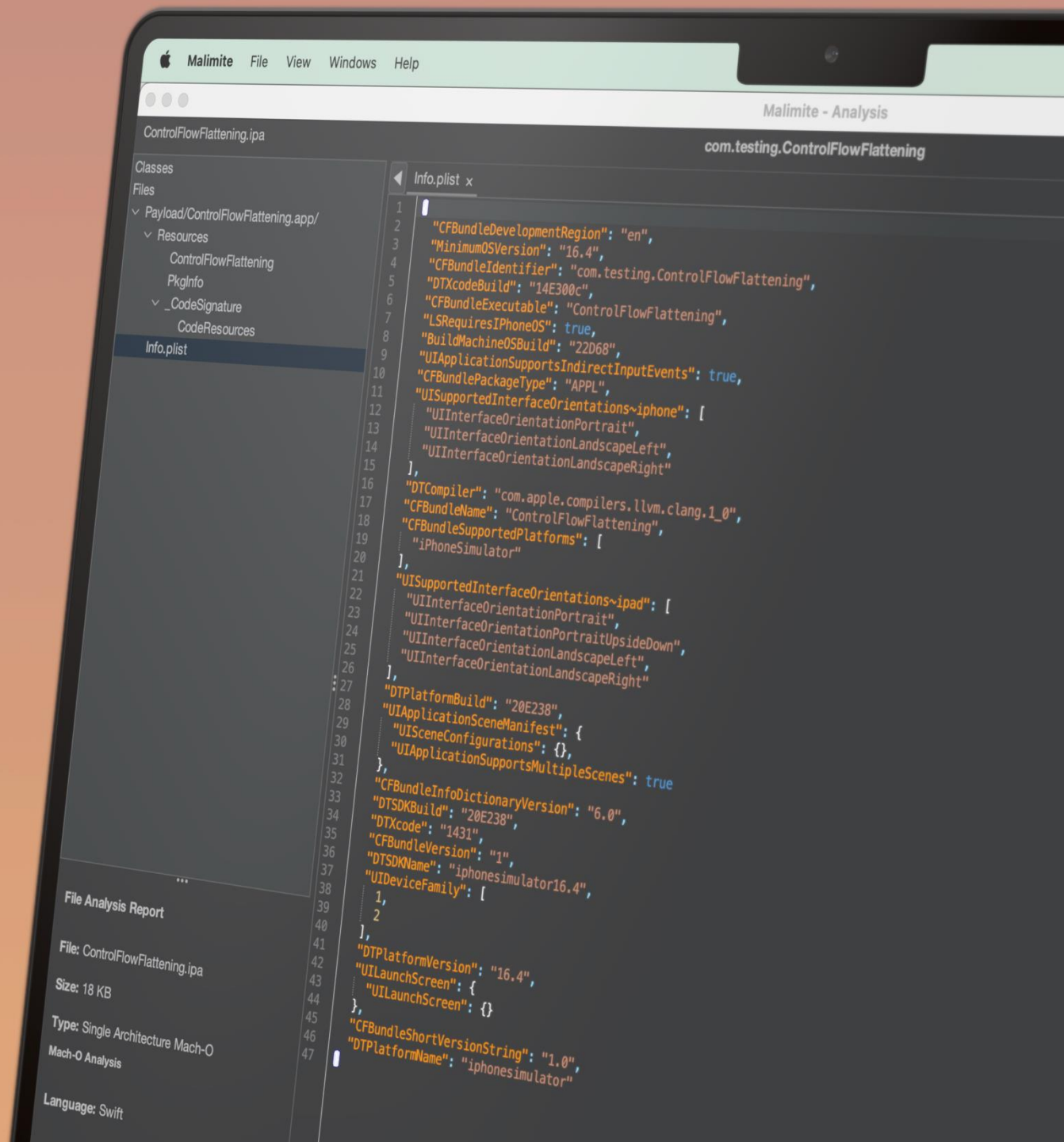
```swift
// Class: ControlFlowFlattening
// Function: nonControlFlowFlattened

func nonControlFlowFlattened() {
    var local78: Int = 0 // Placeholder for local_78
    var local70: UInt8 = 0 // Placeholder for local_70
    let puVar2 = typeMetadataForSwiftString() // Placeholder for puVar2
    let pcVar1 = typeMetadataForAny() + 8 // Placeholder for pcVar1
    let pcVar6 = protocolWitnessTableForSwiftIntComparable() // Placeholder for pcVar6
    // Assuming the necessary Swift constructs replacing the low-level operations in the C-like code:
    while true {
        // This is a placeholder next.
        // The .next() method should be properly adjusted according to the actual iterable used in Swift
        local70 &= 1
        if local70 != 0 {
            break
        }

        if local78 == 0 {
            let tVar9 = allocateUninitializedArray(1) // Placeholder for tuple allocation
            let sVar10 = "Zero"
            print(sVar10)
            // Releasing memory if needed or using ARC in Swift, this can be omitted.
        } else if local78 < 6 {
            let tVar9 = allocateUninitializedArray(1) // Placeholder for tuple allocation
            let sVar10 = "Less"
            print(sVar10)
            // Releasing memory if needed or using ARC in Swift, this can be omitted.
        } else {
            let tVar9 = allocateUninitializedArray(1) // Placeholder for tuple allocation
            let sVar10 = "Greater"
            print(sVar10)
            // Releasing memory if needed or using ARC in Swift, this can be omitted.
        }
    }
}
```

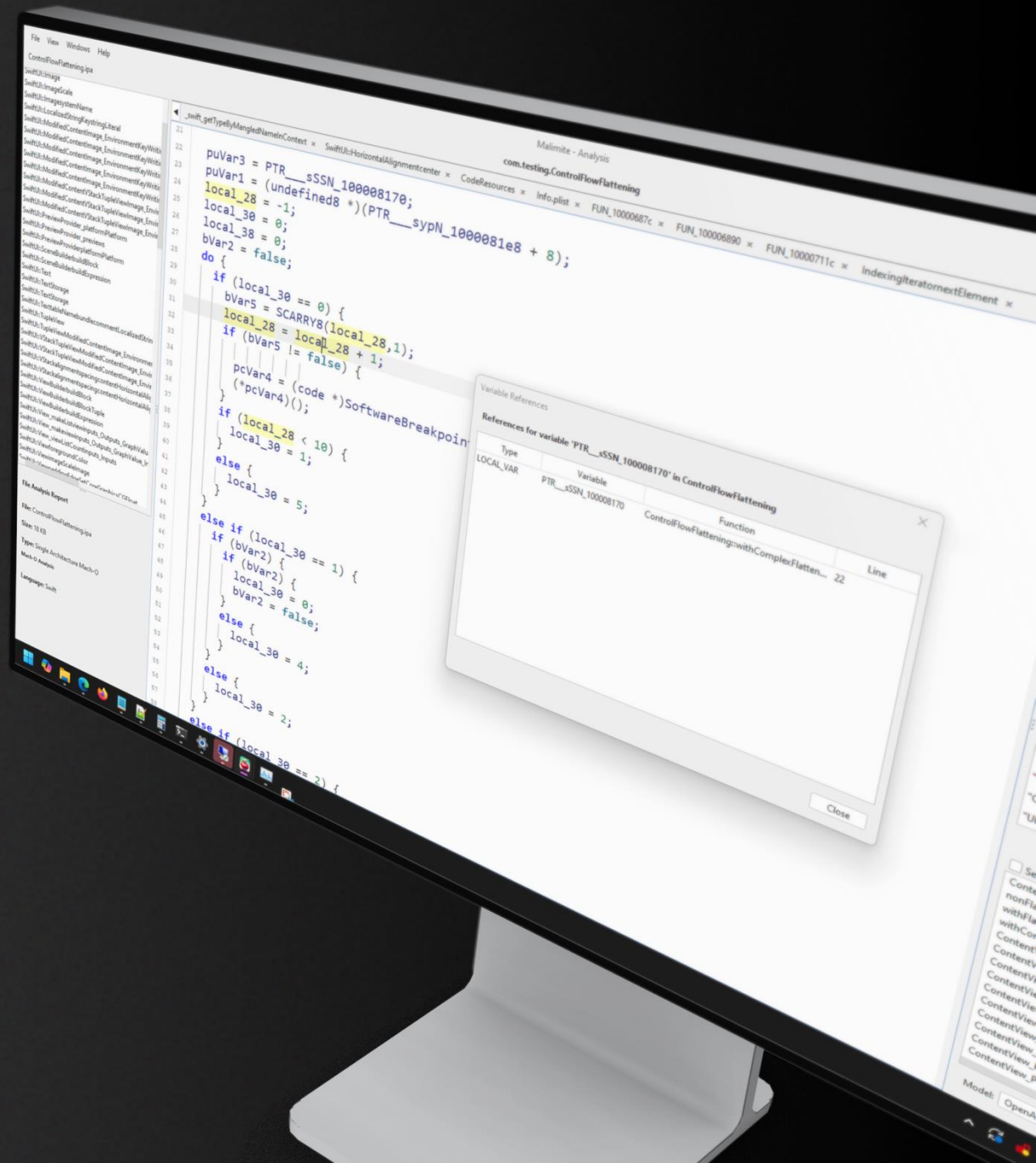
Introducing

Malimite



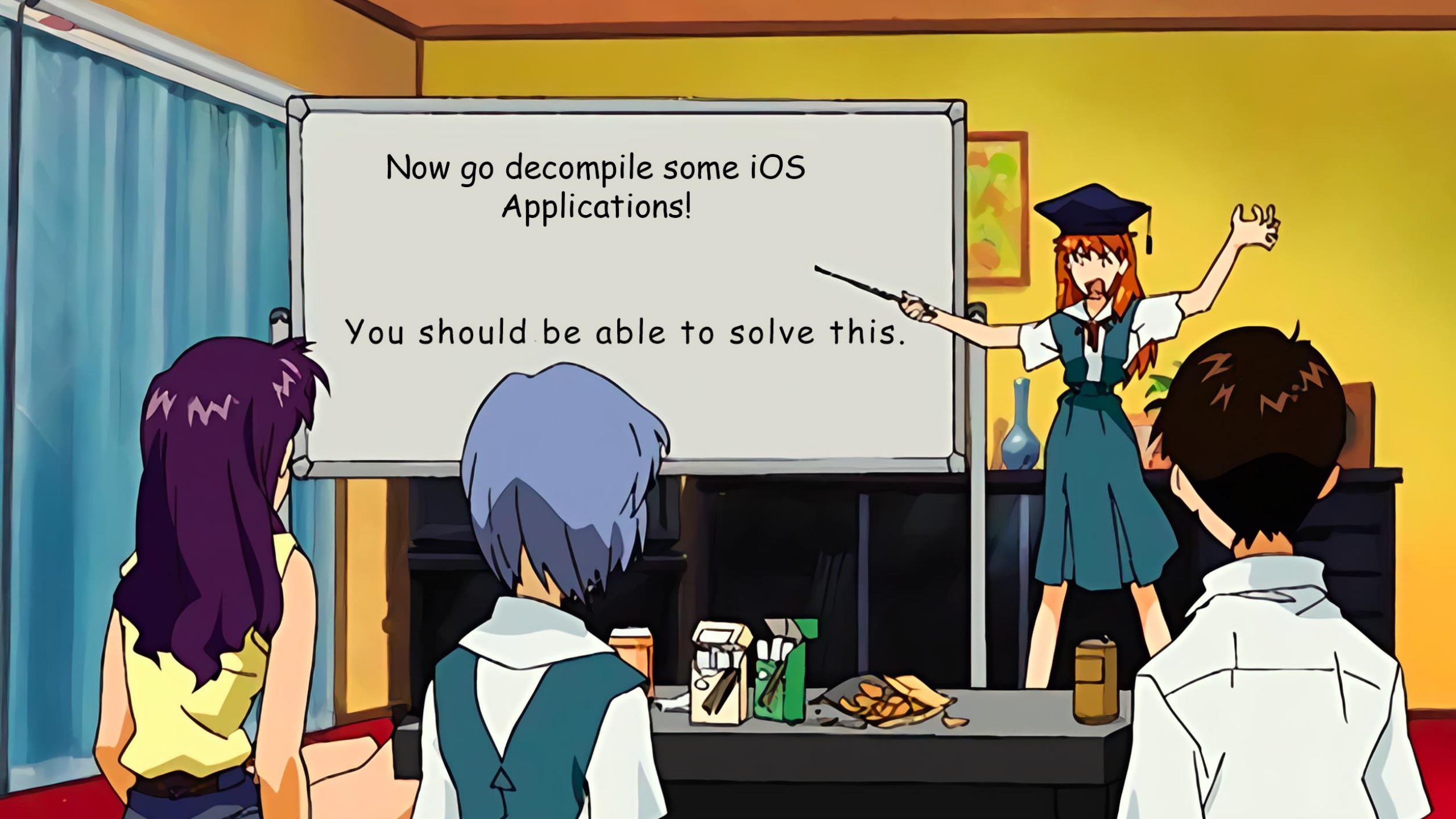
FEATURES

- Multi-Platform
  - Mac, Windows, Linux
- Direct support for IPA files
- Auto decodes iOS resources
- Avoids lib code decompilation
- Reconstructs Swift classes
- Built-in LLM method translation



DEMO:

Analyzing iOS Apps with Malimite


Now go decompile some iOS Applications!
You should be able to solve this.

THANK YOU!

https://github.com/LaurieWired/Malimite