面试突击班——动态库&静态库

面试与答疑

Cat 04.22

今天课程的主要内容

1. 动态库和静态库的区别是什么?
2. 静态库链接到主程序，静态库存放在什么位置？动态库呢?
3. 静态库、与动态库与framework的关系?
4. 什么是xcframework，使用有什么优势?
5. 什么是dead strip与-ObjC参数与-force_load之间有联系吗?
6. 什么是tbd文件，在日常开发中那些应用场景?
7. 要减小App的体积，应该使用静态库还是动态库，为什么?

常用库文件格式

.a .dylib .framework

.xcframework

什么是静态库?

静态库即静态链接库：可以简单的看成一组目标文件的集合。即很多目标文件经过压缩打包后形成的文件。Windows 下的 .lib，Linux 和 Mac 下的 .a。Mac独有的.framework。

缺点：

浪费内存和磁盘空间，模块更新困难

什么是动态库?

与静态库相反，动态库在编译时并不会被拷贝到目标程序中，目标程序中只会存储指向动态库的引用。等到程序运行时，动态库才会被真正加载进来。格式有：.framework、.dylib、.tdb。

缺点：

会导致一些性能损失。但是可以优化，比如延迟绑定(Lazy Binding)技术

Framework

Mac OS/iOS 平台还可以使用 Framework。Framework 实际上是一种打包方式，将库的二进制文件，头文件和有关的资源文件打包到一起，方便管理和分发。

Framework 和系统的 UIKit.Framework 还是有很大区别。系统的 Framework 不需要拷贝到目标程序中，我们自己做出来的 Framework 哪怕是动态的，最后也还是要拷贝到 App 中（App 和 Extension 的 Bundle 是共享的），因此苹果又把这种 Framework 称为 Embedded Framework。

Embedded Framework

开发中使用的动态库会被放入到ipa下的framework目录下，基于沙盒运行。

不同的App使用相同的动态库，并不会只在系统中存在一份。而是会在多个App中各自打包、签名、加载一份。

XCFramework

XCFramework：是苹果官方推荐的、支持的，可以更方便的表示一个多个平台和架构的分发二进制库的格式。

需要Xcode11以上支持。

是为更好的支持Mac Catalyst和ARM芯片的macOS。

专门在2019年提出的framework的另一种先进格式。

XCFramework

iOS/iPad：arm64

iOS/iPad Simulator：x86_64 arm64

Mac Catalyst: x86_64 arm64

Mac: x86_64 arm64

XCFramework

和传统的framework相比：

1. 可以用单个.xcframework文件提供多个平台的分发二进制文件；
2. 与Fat Header相比，可以按照平台划分，可以包含相同架构的不同平台的文件；
3. 在使用时，不需要再通过脚本去剥离不需要的架构体系。

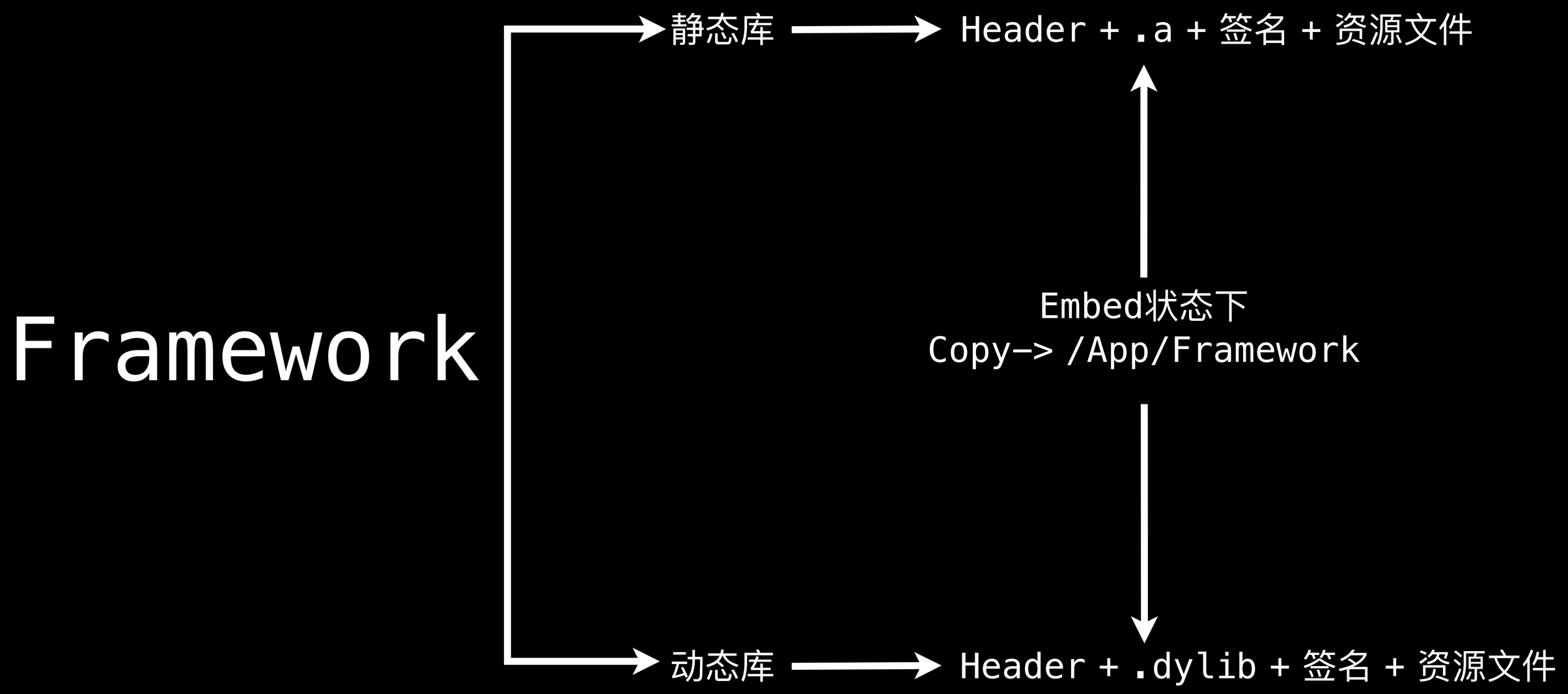
什么是tbd格式?

tbd全称是text-based stub libraries，本质上就是一个YAML描述的文本文件。

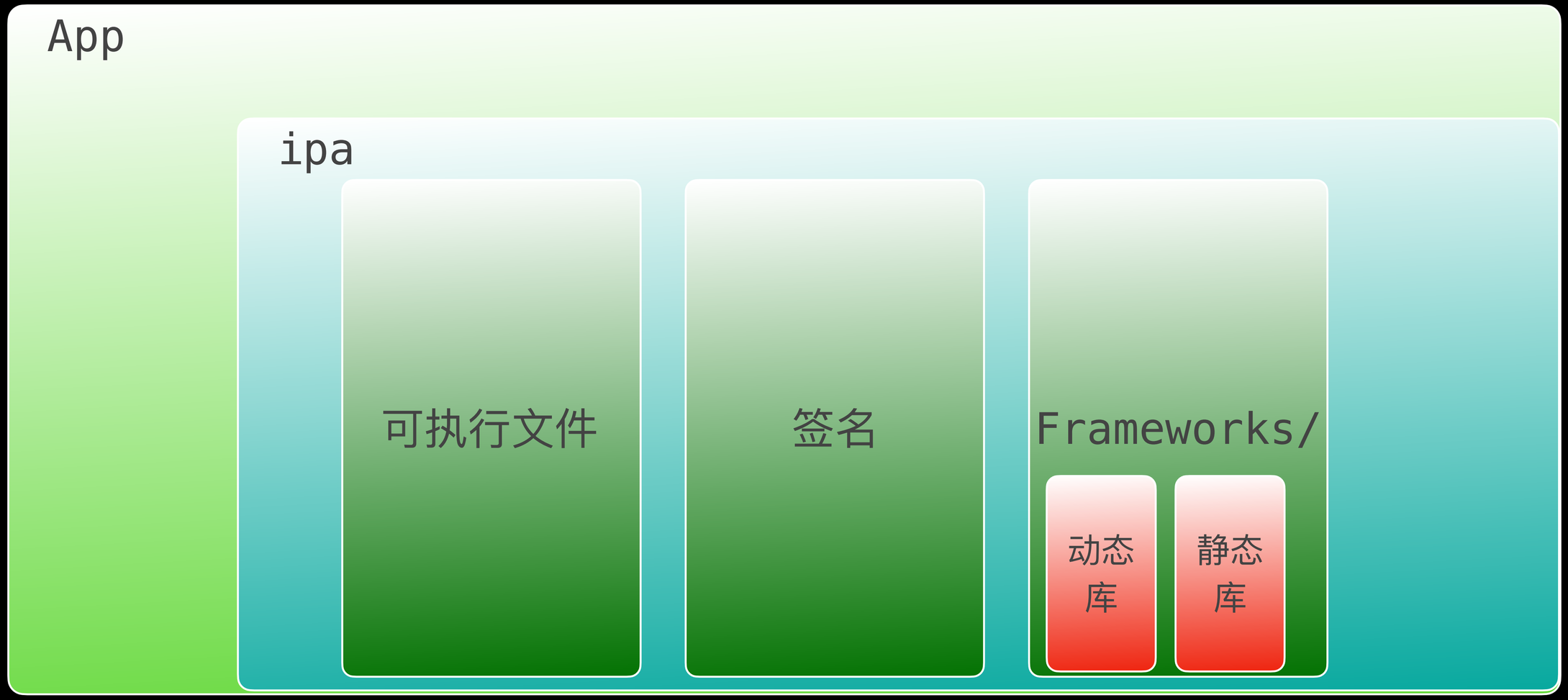
他的作用是用于记录动态库的一些信息，包括导出的符号、动态库的架构信息、动态库的依赖信息

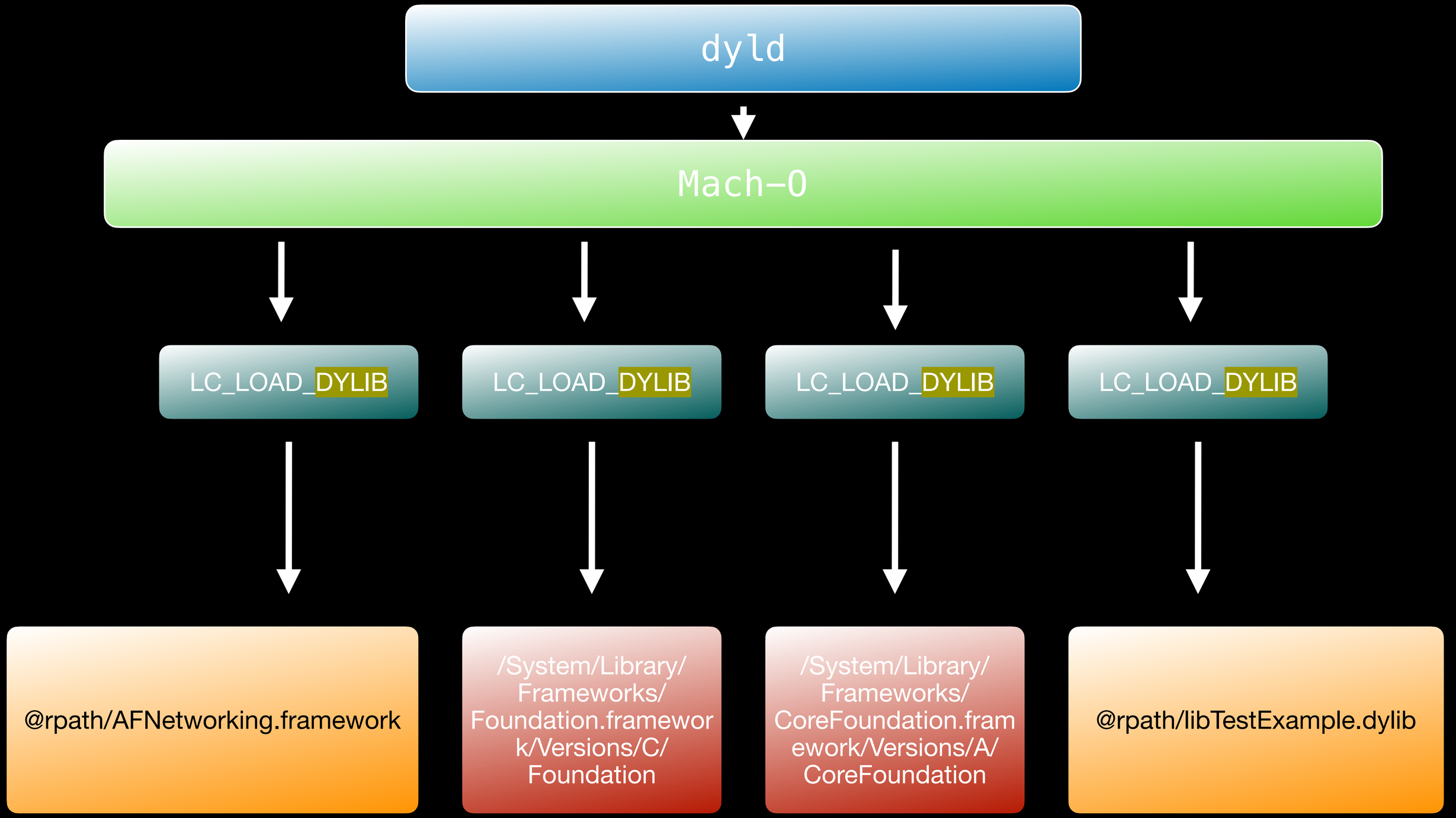
用于避免在真机开发过程中直接使用传统的dylib。

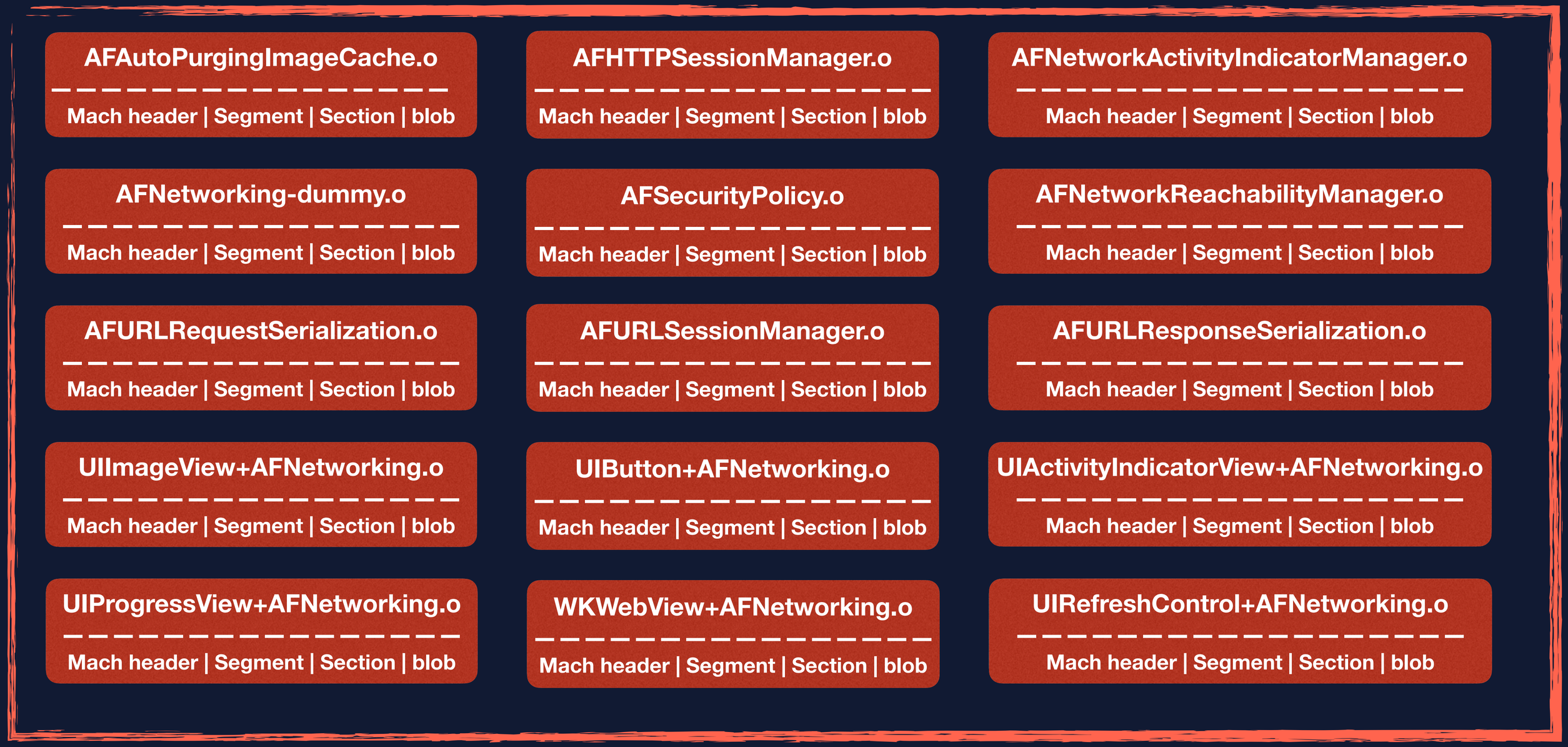
对于真机来说，由于动态库都是在设备上，在Xcode上使用基于tbd格式的伪framework可以大大减少Xcode的大小。


Framework
静态库
Header + .a + 签名 + 资源文件
Embed状态下
Copy-> /App/Framework
动态库
Header + .dylib + 签名 + 资源文件

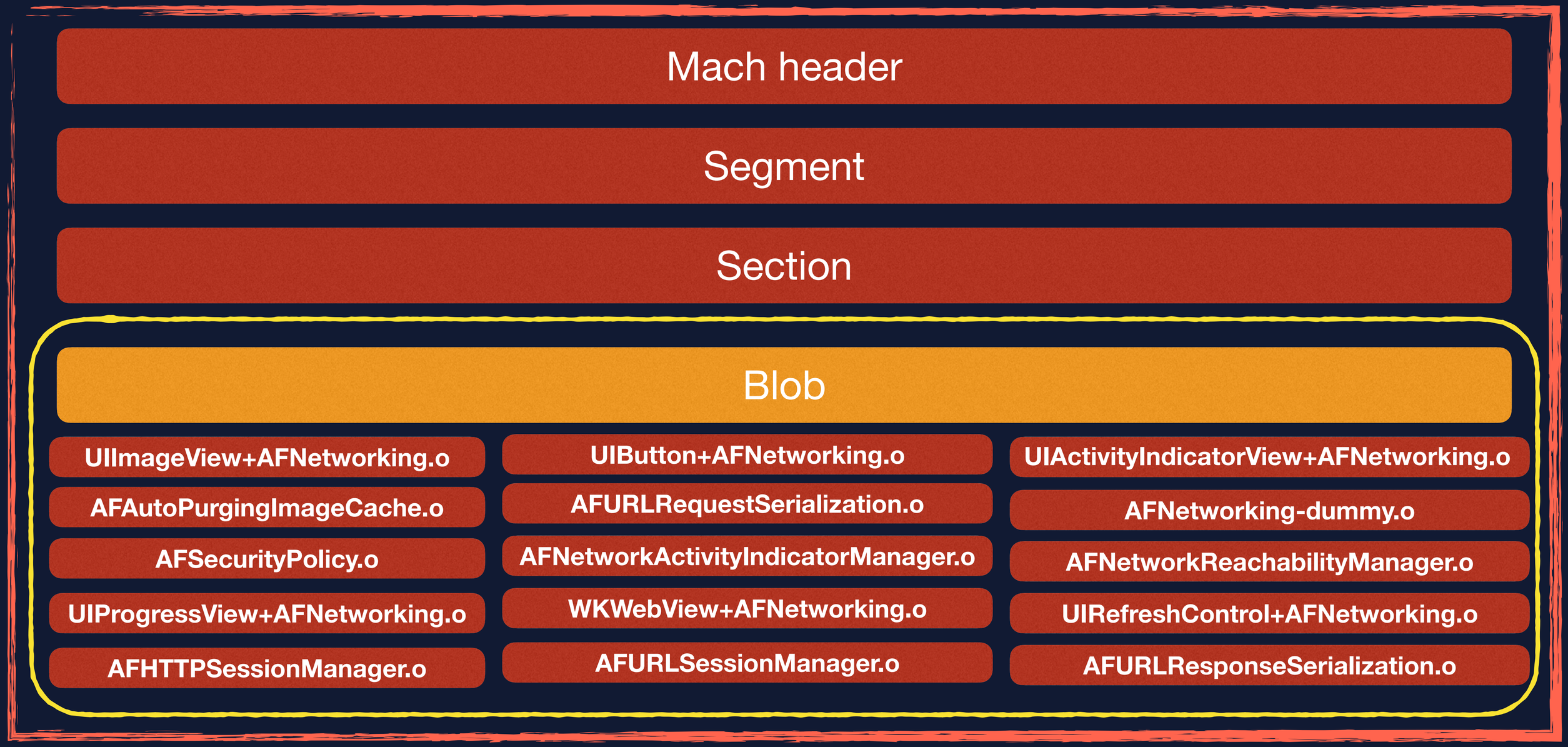
App framework存放位置




dyld
Mach-O
LC_LOAD_DYLIB
LC_LOAD_DYLIB
LC_LOAD_DYLIB
LC_LOAD_DYLIB
@rpath/AFNetworking.framework
/System/Library/Frameworks/Foundation.framework/Versions/C/Foundation
/System/Library/Frameworks/CoreFoundation.framework/Versions/A/CoreFoundation
@rpath/libTestExample.dylib



libAFNetworking.a



libAFNetworking.dylib

-noall_load

-all_load

-ObjC

-force_load <file>

-dead_strip