

# 闲鱼Flutter实践

闲鱼技术团队  
国有

# Table of Contents

- Flutter简介
- 闲鱼项目已解决的问题
- Flutter上线效果，性能对比，成熟度
- 分模块技术简介
  - 混合栈解决方案
  - 混合工程方案
  - 视频、图片
  - 原有生态适配
- 使用体感 - 工具链、问题排查
- 小结

# Flutter, 一种新的可能性



Google Flutter

or



Android

+



iOS

or

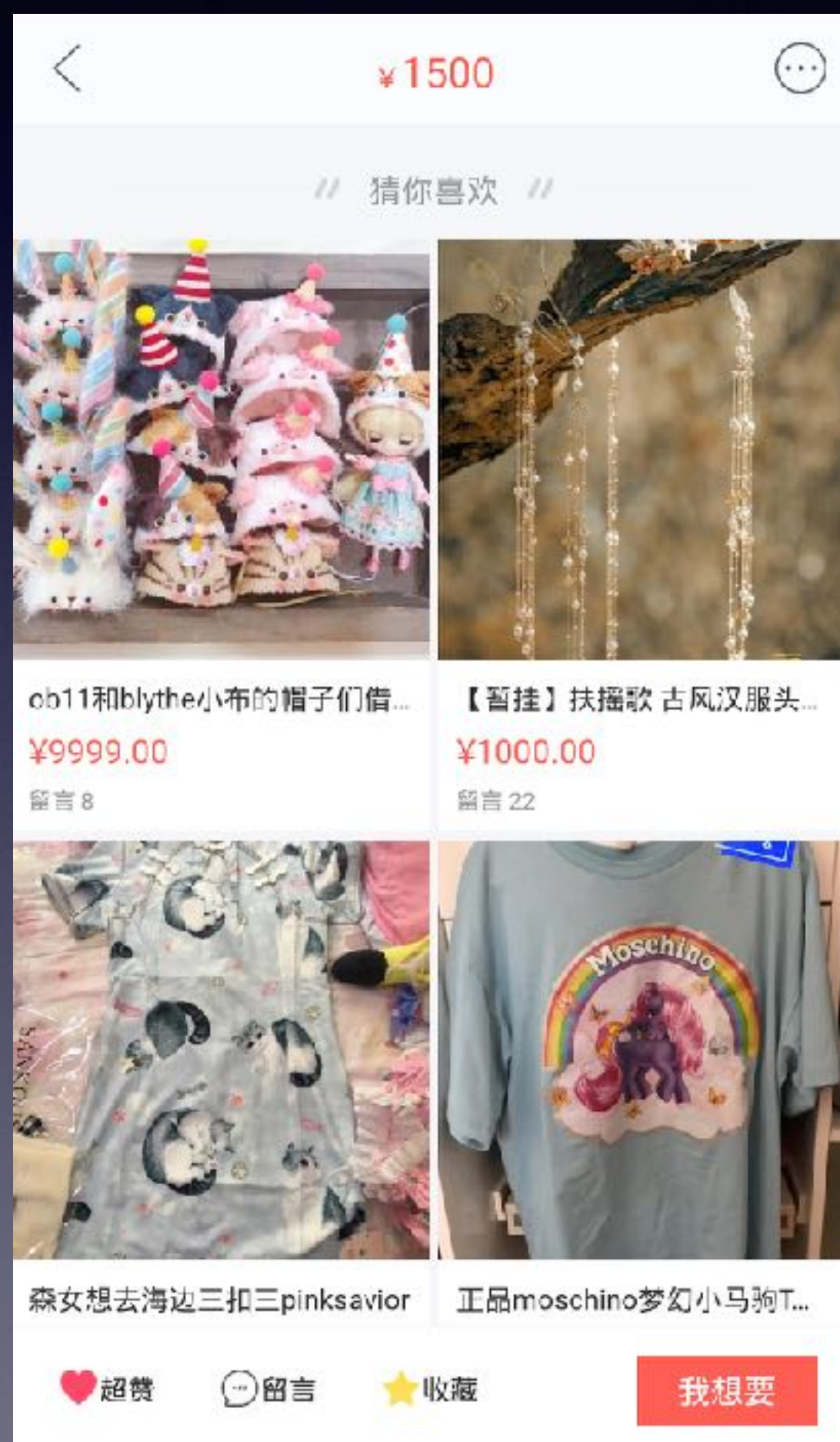


or

RN, Mono, Kotlin, Cocos2d...

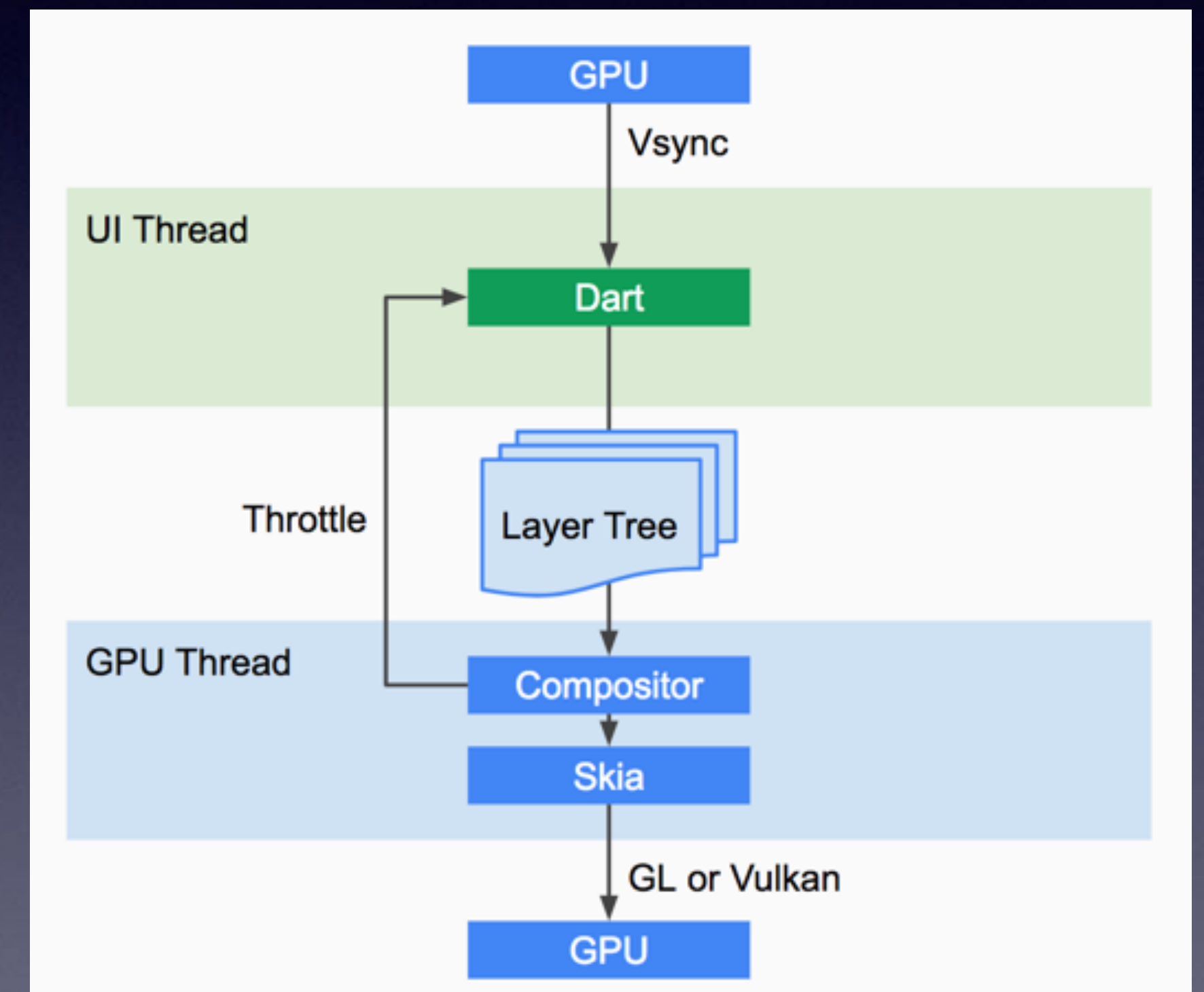
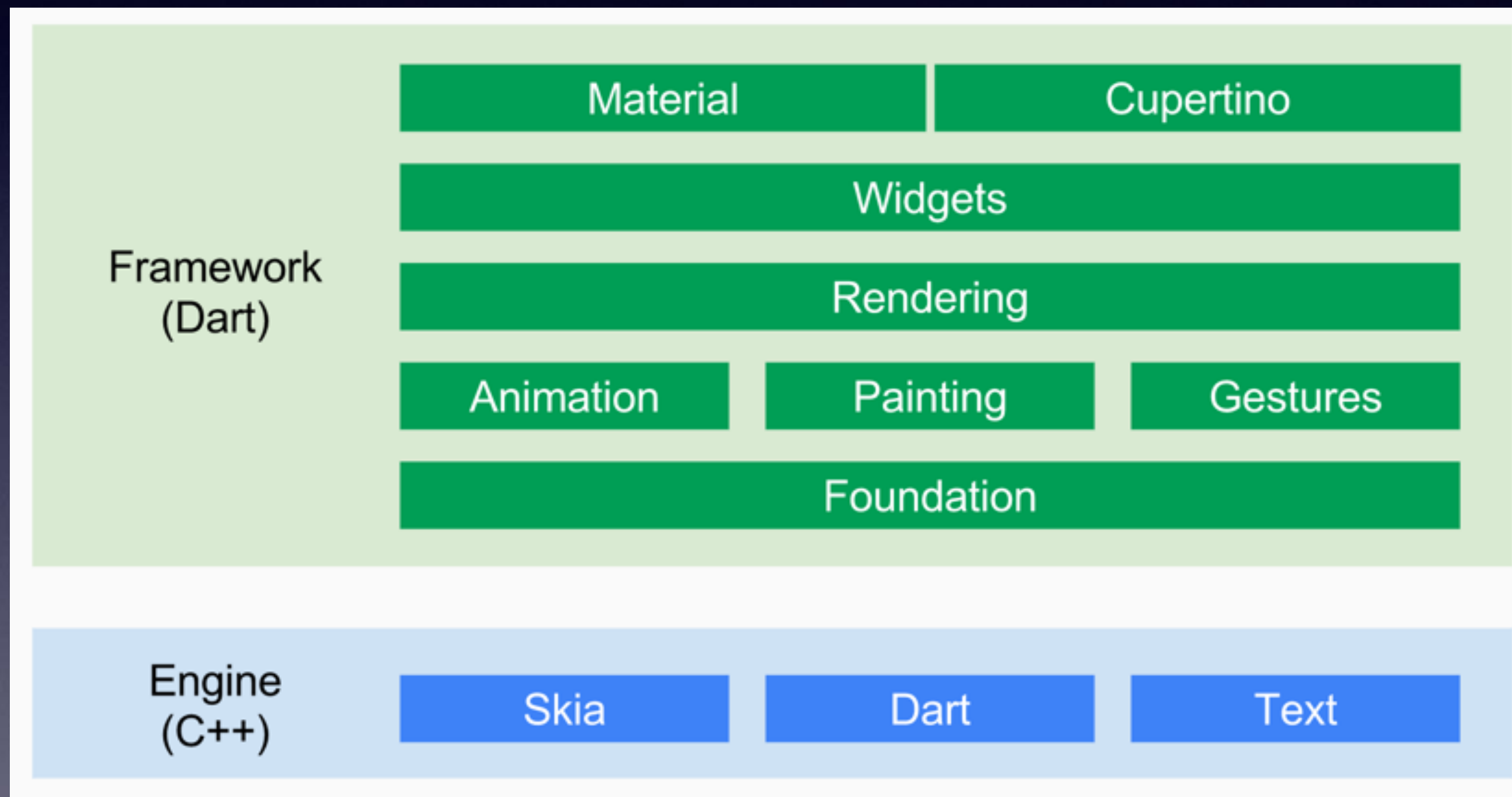


# 闲鱼商品详情页面

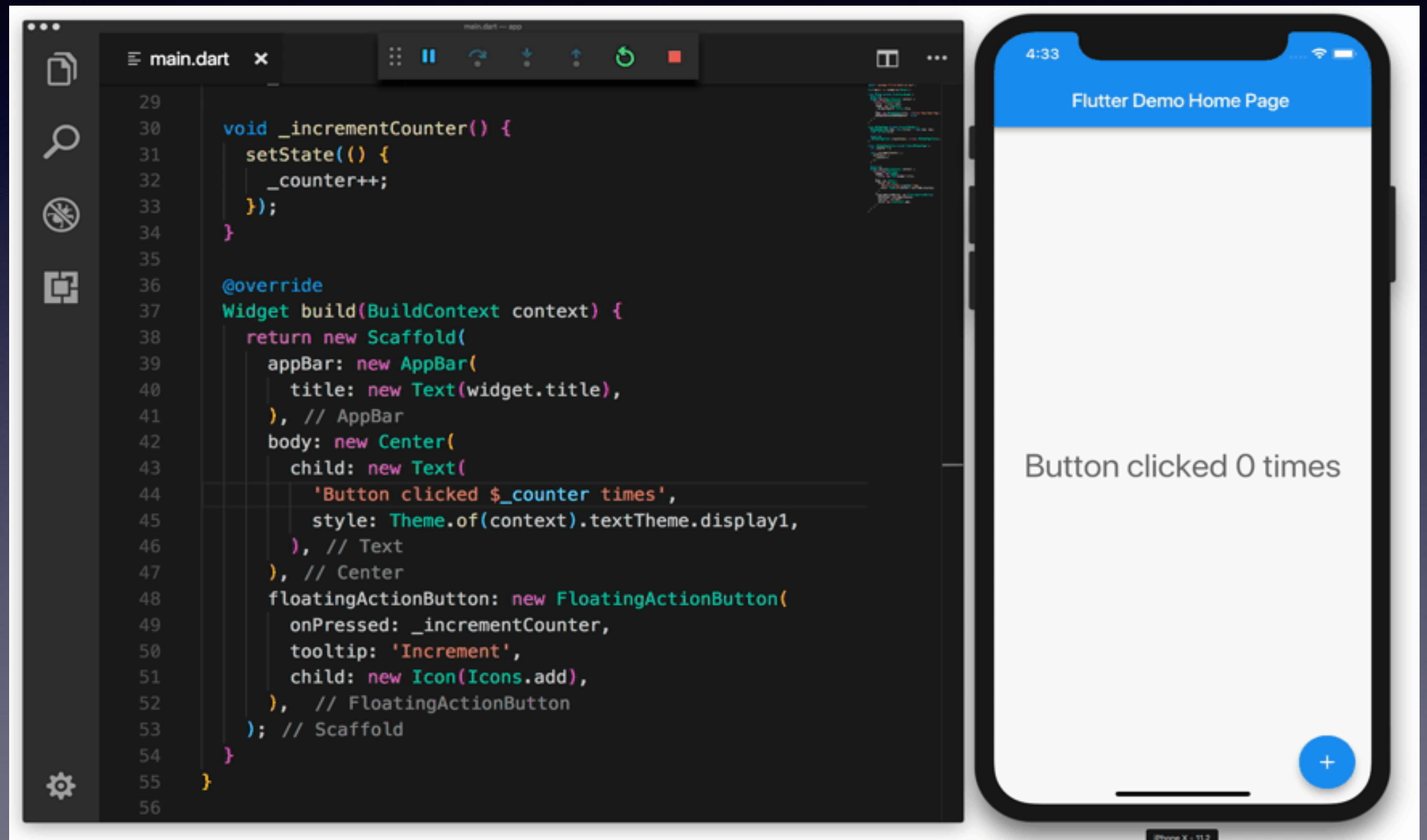
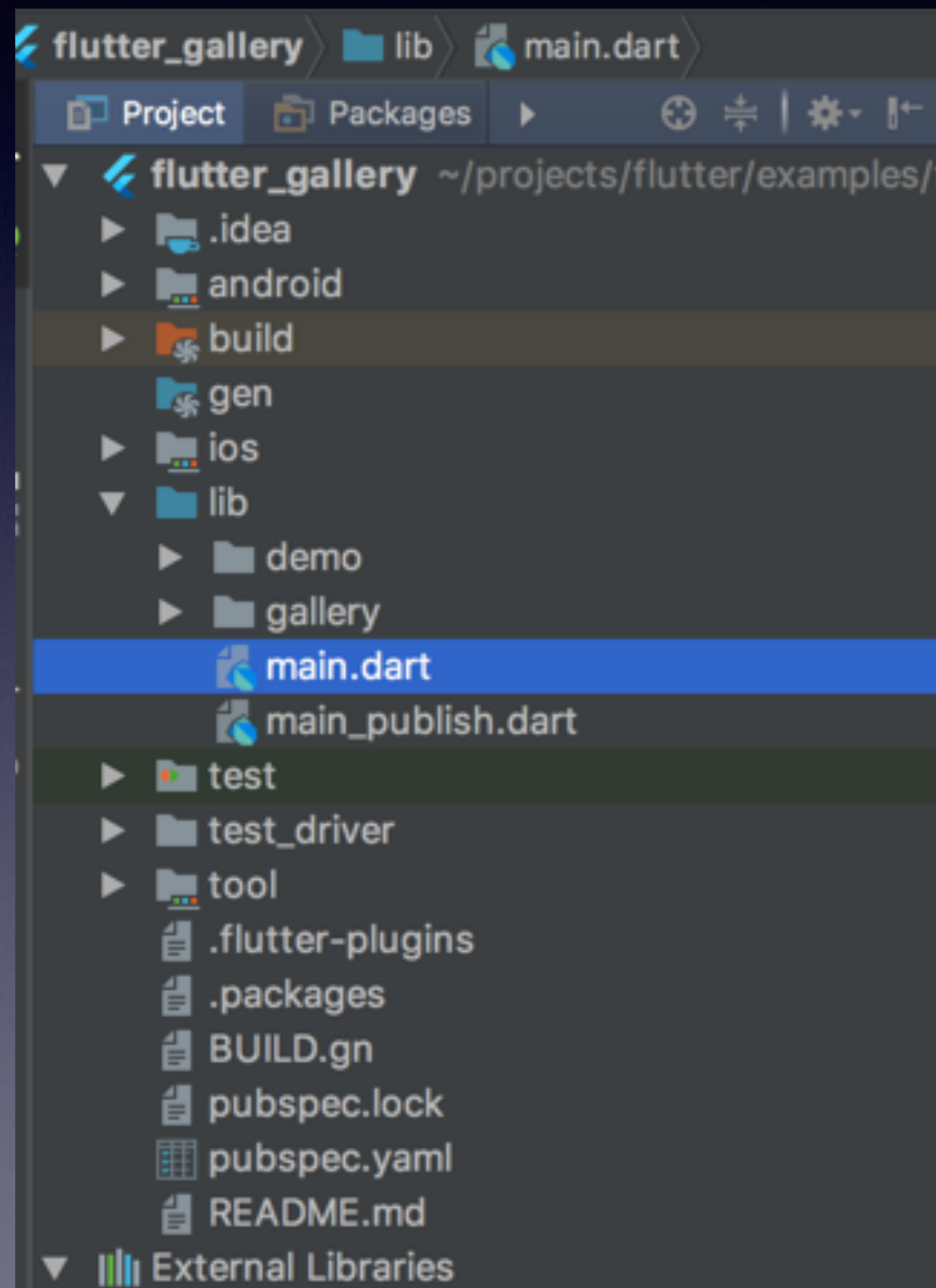




# Flutter系统结构



# Demo



# Table of Contents

- Flutter简介
- 闲鱼项目已解决的问题
- Flutter上线效果，性能对比，成熟度
- 分模块技术简介
  - 混合栈解决方案
  - 混合工程方案
  - 视频、图片
  - 原有生态适配
- 使用体感 - 工具链、问题排查
- 小结



# 体验链接



✓ **GooglePlay**, 全量开放 v6.1.2.3

<https://play.google.com/store/apps/details?id=com.taobao.idlefish>

国内正式渠道 v6.1.3

- 灰度量: 5%

AppStore v6.1.3

- 灰度量: 5%

数据时间: 2018.5.24





# 闲鱼项目已解决的问题概览

## 1, Flutter与Native混合编程

- 混合页面栈
- 混合工程、工具

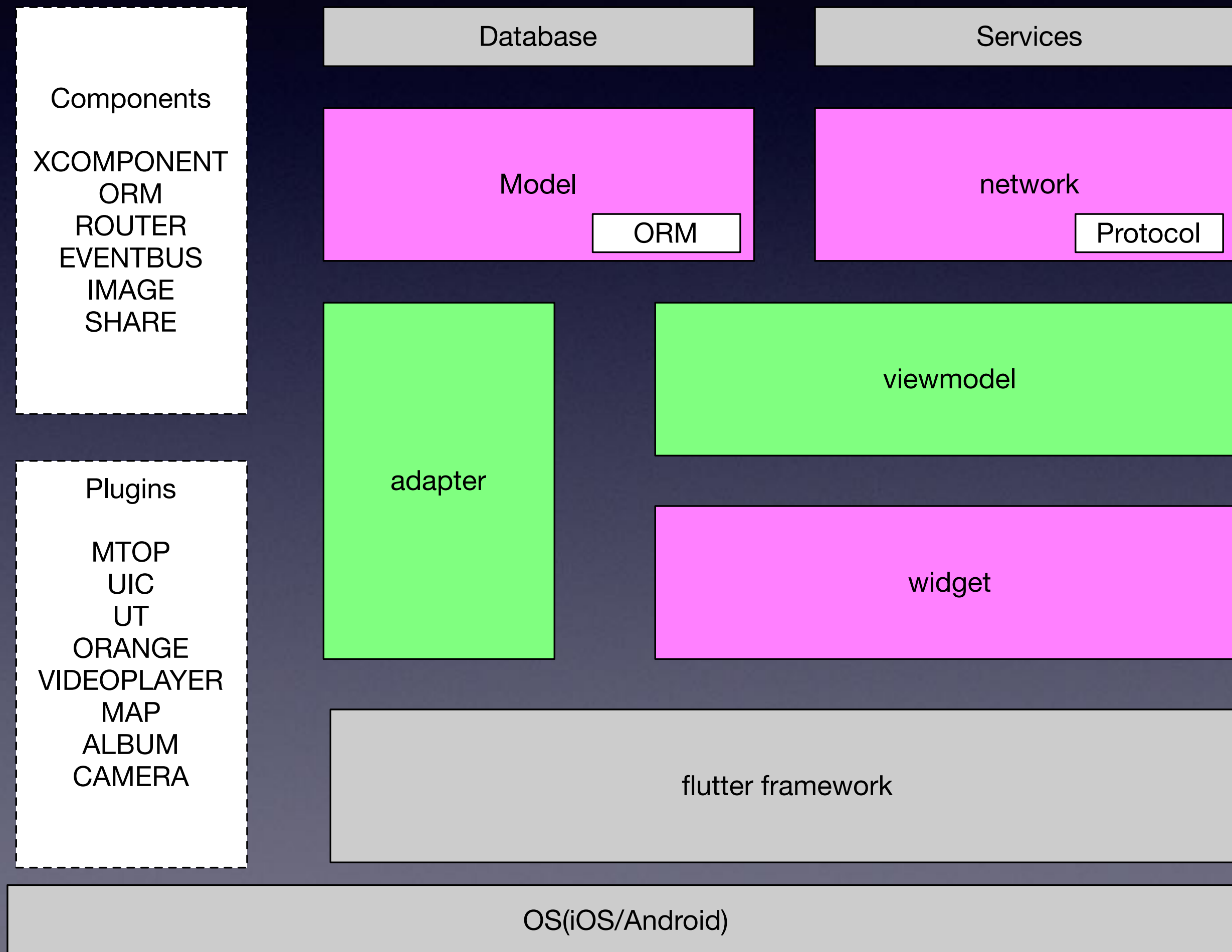
## 2, 从Beta到生产的gap问题

- armv7编译
- 生产机型兼容性

## 3, 阿里生态问题

- 私有仓库
- Mtop、UT、监控等适配

# Flutter应用架构概览





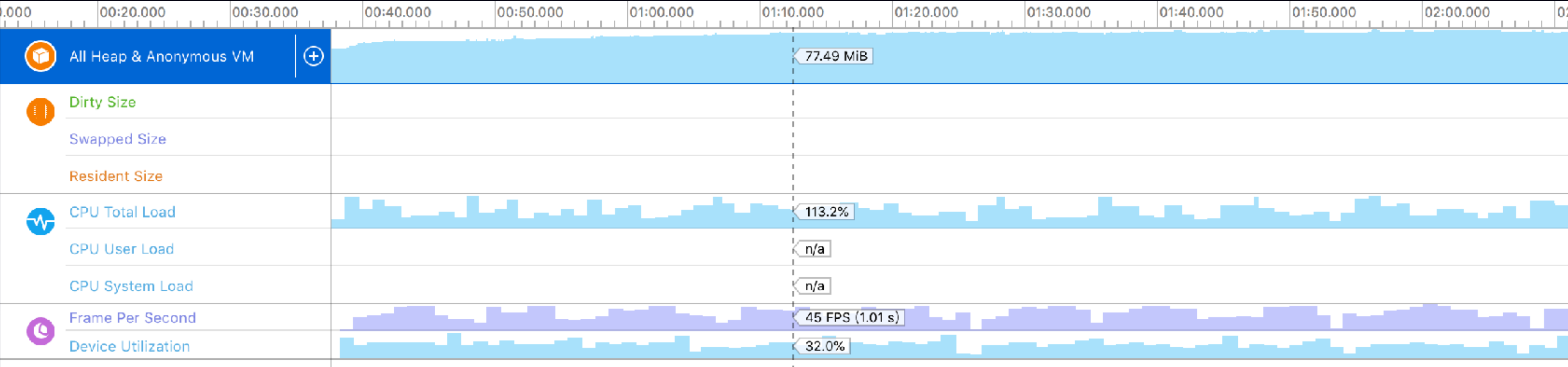
# Table of Contents

- Flutter简介
- 闲鱼项目已解决的问题
- Flutter上线效果，性能对比，成熟度
- 分模块技术简介
  - 混合栈解决方案
  - 混合工程方案
  - 视频、图片
  - 原有生态适配
- 使用体感 - 工具链、问题排查
- 小结

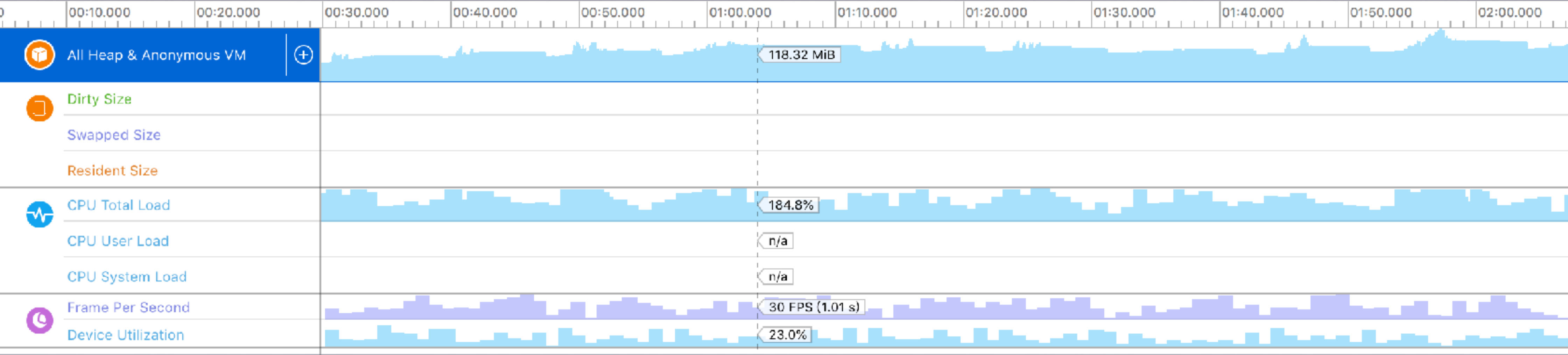
测试机型：iphone6  
iOS版本：11.3  
闲鱼版本：6.1.3  
测试方法：flutter版本和native版本的  
详情页面执行相同操作，即从我发布的  
的页面进入10个不同的详情页面

# Native性能对比

Flutter表现



Native表现





# Flutter的成熟度

Android篇					
灰度次数	灰度版本	Crash率(flutter相关比例)	Crash原因	Crash解法	备注
1	6.0.7.1	245%(72%)	flutter引擎内部getBitmap存在内存泄漏导致libflutter.so OOM Abort	移除getBitmap/screenshot逻辑	flutter v0.2.3
2	6.0.7.4	62.64%(42%)	flutter引擎内部embedder层拷贝icudtl.dat异常导致资源损坏	重构此段一次性拷贝逻辑到主线程	
3	6.0.8.5	3.3%(72%)	flutter视频实现依赖的libaliplayer.so异常导致	切换player内核修复	
4	6.1.1.2	0.4%(4%)	趋于稳定，Flutter相关Crash正常	无	flutter v0.3.1

iOS篇					
灰度次数	灰度版本	Crash率(flutter相关比例)	Crash原因	Crash解法	备注
1	6.0.7.2	48.75%(35%)	引擎内部有Bug，导致AcquireFrame异常	engine升级	
2	6.1.1.1	14.7%(80%)	flutter引擎中shell中的io/gpu逻辑可能会在后台操作GPU	修改shell层，确保io/gpu task runner在后台不执行	
3	6.1.1.3	0.4%(20%)	趋于稳定		

其他问题:		
端测	问题	解法
Android	带视频的宝贝页截图Crash	Flutter修改engine,getBitmap时注册Texture.
iOS	iOS9之前文字空白	Flutter修改engine，9之前使用Helvetica
iOS	中文无法加粗	Flutter修改engine,txt处理笔画时读取加粗信息
iOS	修复iOS9之前文字空白问题时，因为是cherry-pick，因此ninja报错	自行修改ninja配置
iOS	页面内容无法全屏展示	自行修改Flutter中Embedder逻辑，暴露可覆盖的接口给应用侧。
iOS	armv7不支持	自行修改ninja，flutter_tools等提供armv7支持
iOS	iPhone4S指令异常Crash	通过指定gensnapshot flag,将最终的armassemble中对于整数除法的实现从SDIV调整为VFP
Flutter	Observatory never became ready	自行修改Flutter IntelliJ插件并重编译安装插件
Android	部分机型视频不能播放	暂时屏蔽了有问题的设备，稍后的版本将通过Skia的Option来确保正常播放(master已验证)
Android	FlutterActivity跳转半透明Activity并跳走的时候，再返回页面不正常	Flutter修改引擎，此种Case下考虑生命周期中添加操作保证OK

经过详情页项目的几个灰度迭代，坑踩差不多了，目前趋于稳定。

# Table of Contents

- Flutter简介
- 闲鱼项目已解决的问题
- Flutter上线效果，性能对比，成熟度
- 分模块技术简介
  - 混合栈解决方案
  - 混合工程方案
  - 视频、图片
  - 原有生态适配
- 使用体感 - 工具链、问题排查
- 小结



# Flutter、Native页面混合调用

Native



Flutter



Native浮层



Flutter浮层



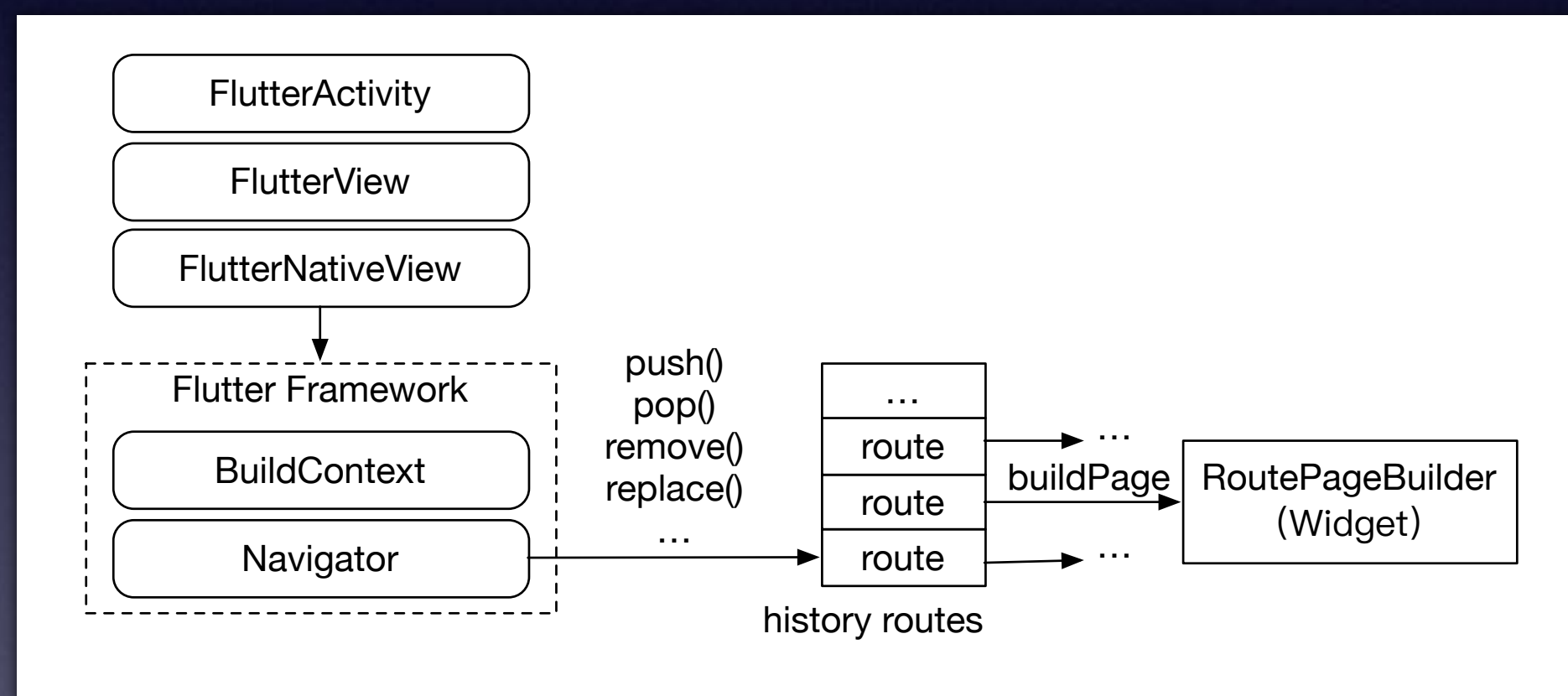
Native





# 混合栈管理

## Flutter默认栈原理



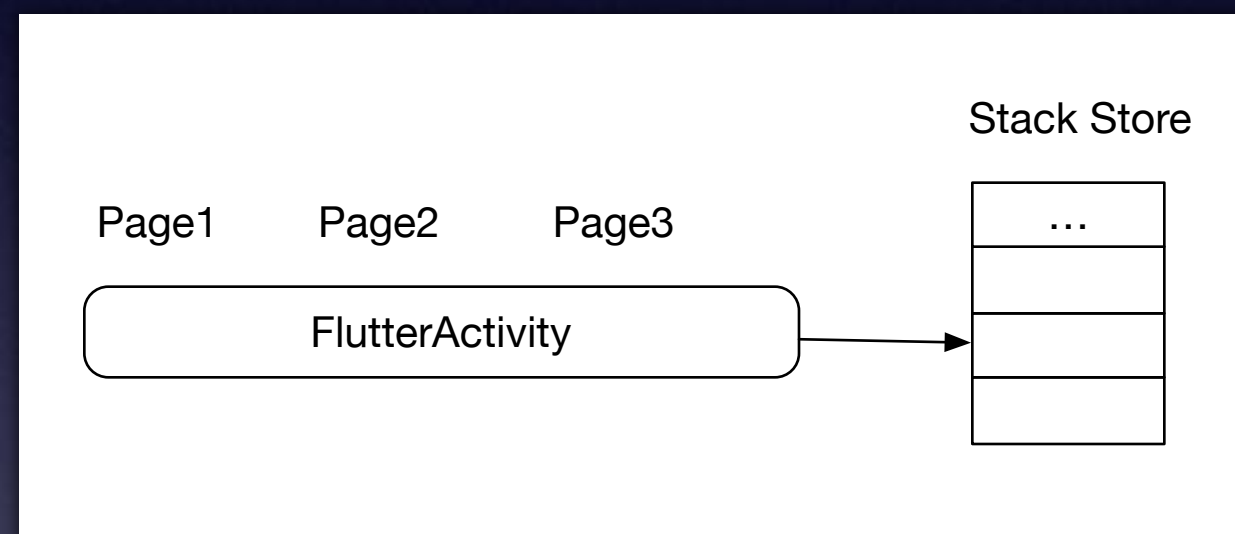
沉淀： 提供package

用法：

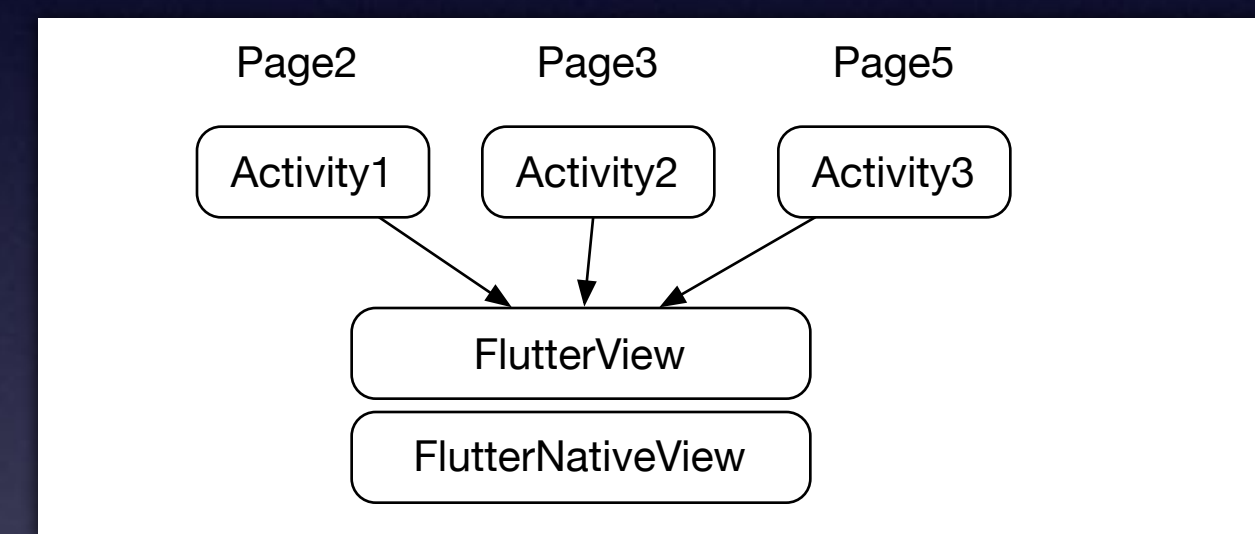
创建Flutter页面时，用package中的FlutterWrapperActivity替换默认的FlutterActivity (以Android为例)

## 混合栈的选型

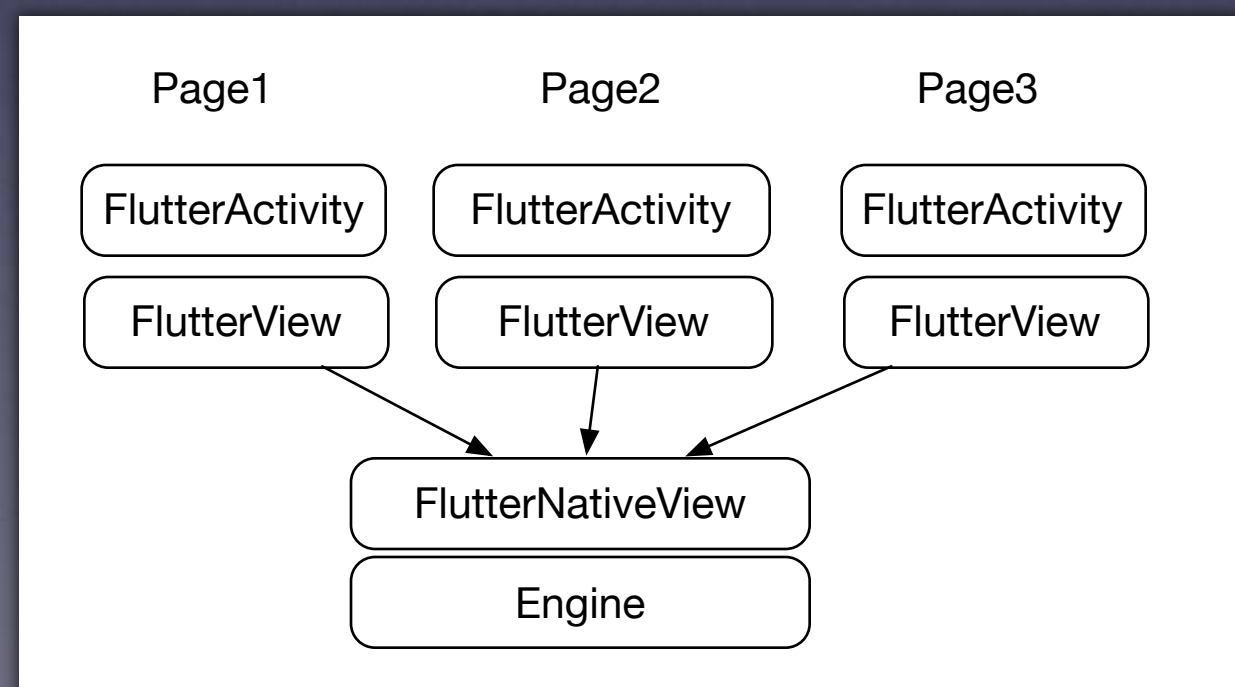
方案1



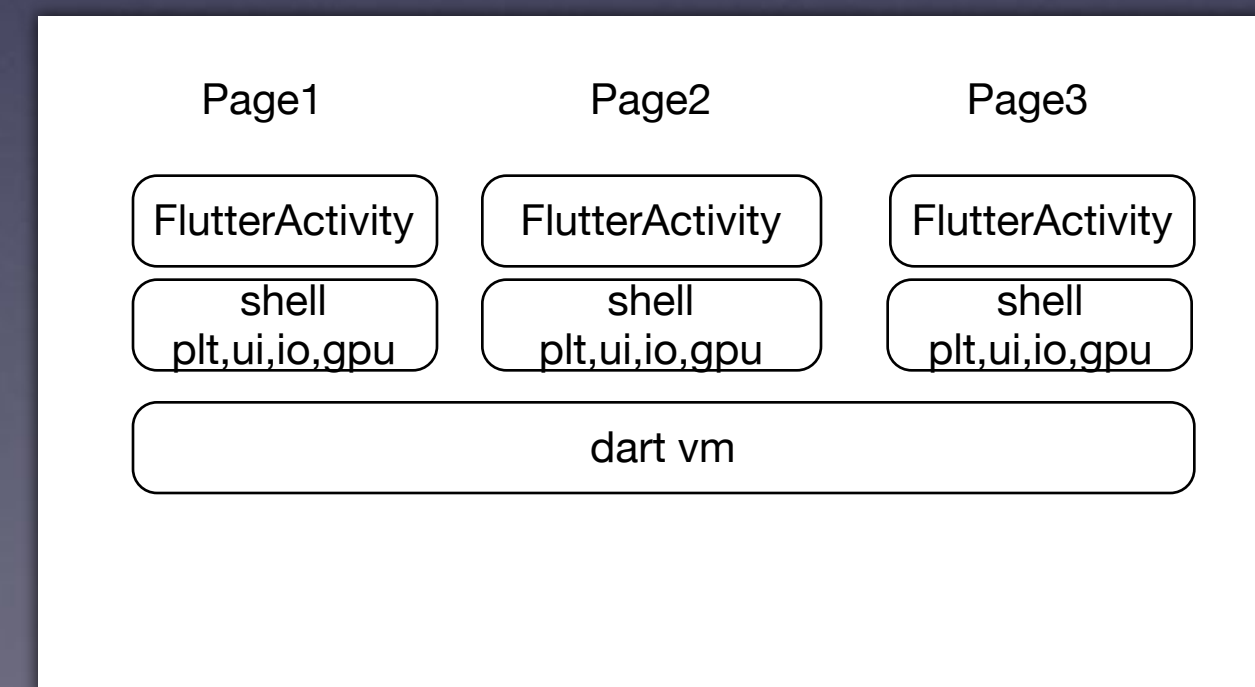
方案2 ✓



方案3



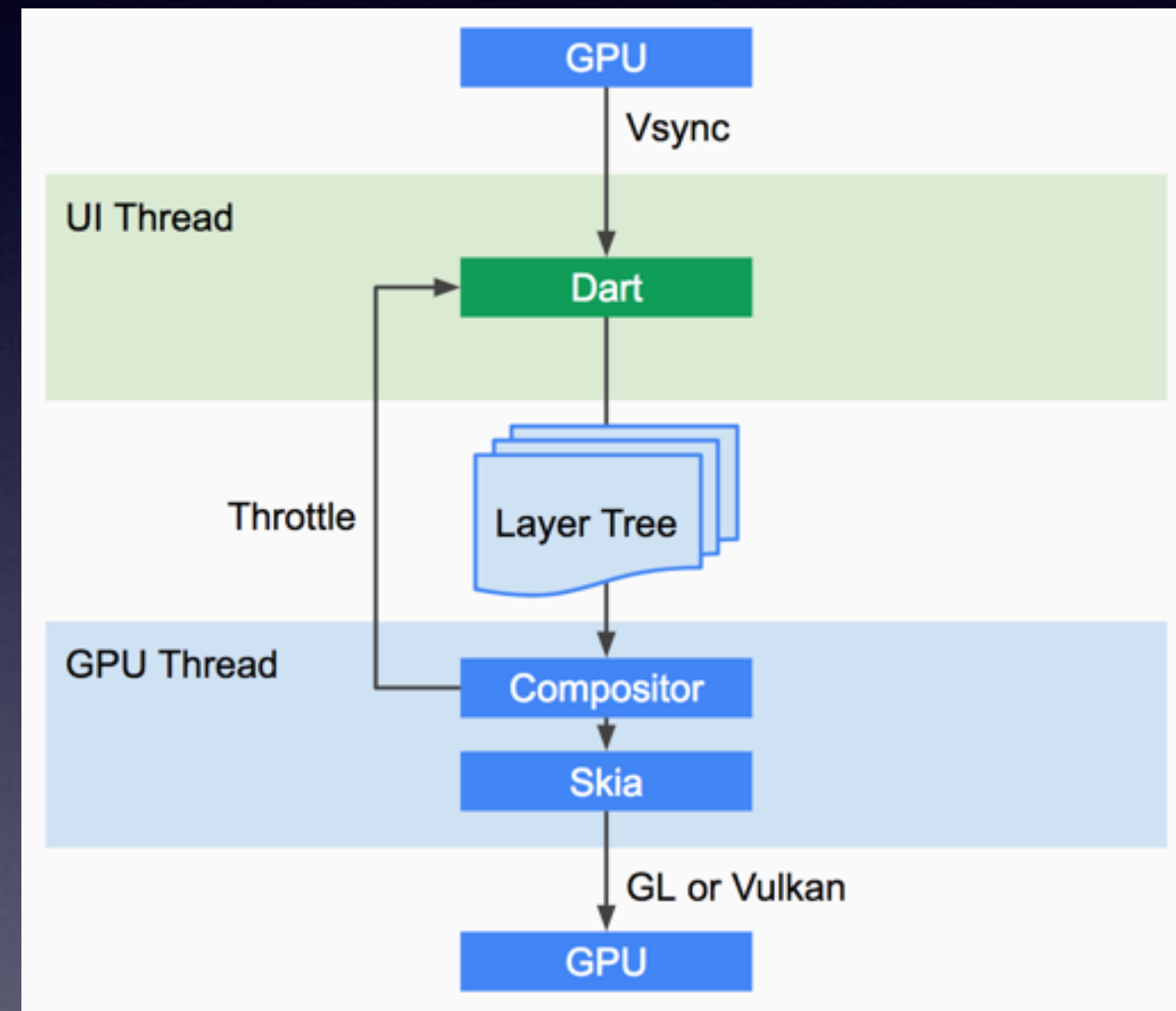
方案4



参考 <https://github.com/flutter/flutter/issues/15559>  
<https://github.com/flutter/engine/pull/4932>



# Flutter页面中唤起Native View控件



注意事项:

Layer冲突问题 (iOS)

解决方案:

用Window或者独立的View唤起Native

# 混编工程

Flutter默认：0->1 新工程

混编工程：(Android | iOS) 2 -> mix

问题：

- 现有Android和iOS是两个仓库
- 现有工程中不包含Flutter引用，需要手工处理引入、构建逻辑
- mtl还不支持flutter打包（Android环境是Linux）
- 纯Native开发同学不需要开发Flutter
- 3个工程，需求来时怎么拉分支
- iOS默认无armv7包



# 混编工程-项目结构

以前

iOS(git&cocoapods)

Android(git&gradle)



现状(Flutter&Native/Weex)

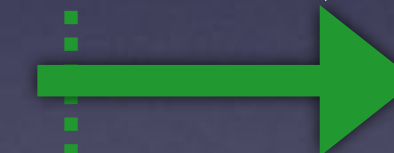
Flutter App Repo(pub)

iOS(git subm&cocoapods)

Android(git subm&gradle)

Flutter(git submodule)

Flutter产物



iOS(git&cocoapods)

Android(git&gradle)

Flutter Team

Native Team



# 混编工程 - Flutter产物

## Android

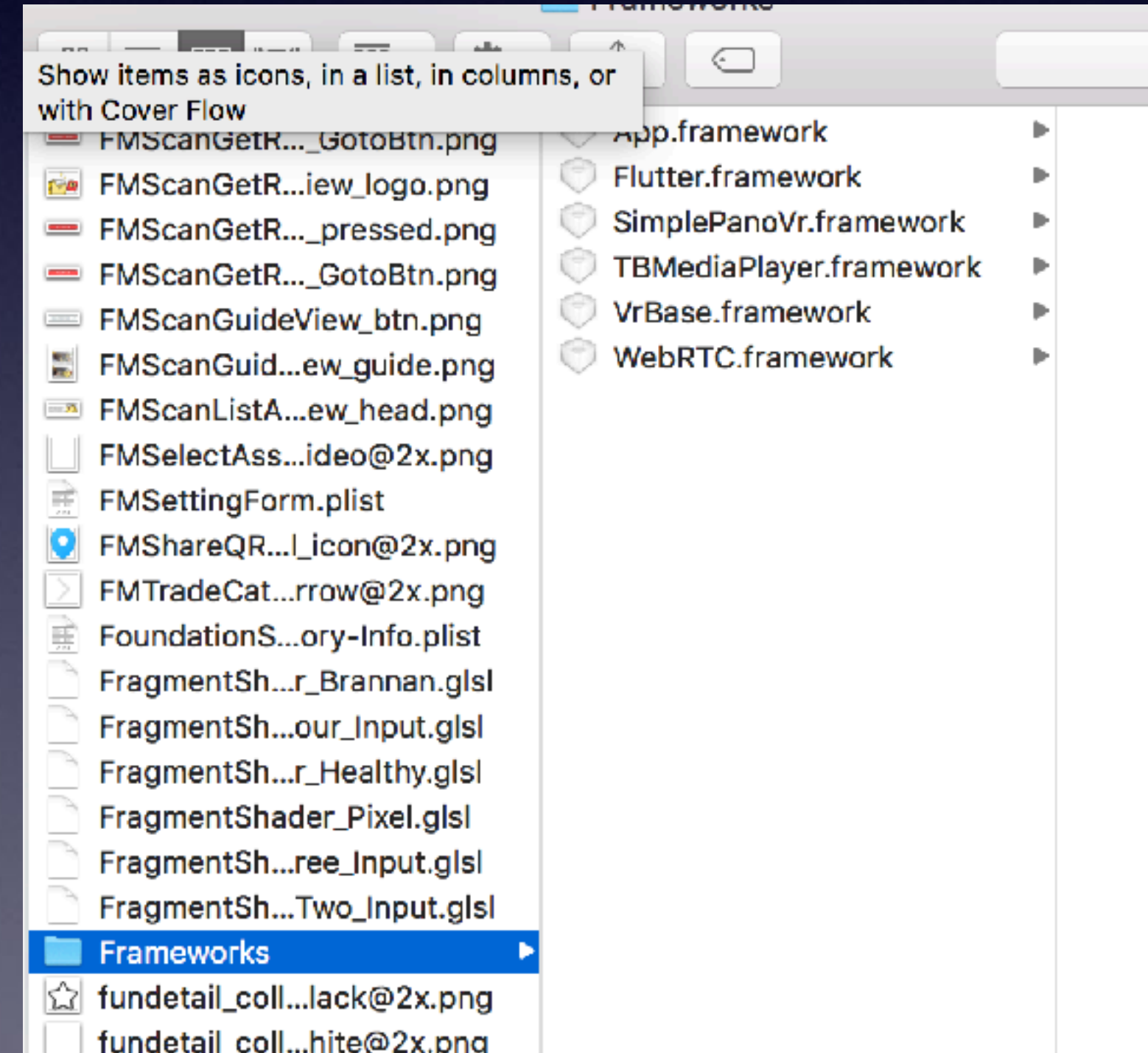
assets	4.4 MB	4 MB	50.2%	
icudtl.dat	2.2 MB	2.2 MB	26.9%	
isolate_snapshot_data	1.1 MB	993.5 KB	12.2%	
isolate_snapshot_instr	946.5 KB	705.9 KB	8.6%	
flutter_assets	222 KB	183.6 KB	2.2%	
vm_snapshot_data	12.7 KB	11.3 KB	0.1%	
vm_snapshot_instr	3.2 KB	2.9 KB	0%	
lib	4.2 MB	3.9 MB	49.1%	
arm64-v8a	4.2 MB	3.9 MB	49.1%	
libflutter.so	4.2 MB	3.9 MB	49.1%	
classes.dex	58.7 KB	54.7 KB	0.7%	
res	4.3 KB	4 KB	0%	
META-INF	2.2 KB	2.2 KB	0%	
AndroidManifest.xml	892 B	892 B	0%	
resources.arsc	1.5 KB	371 B	0%	

libflutter.so - flutter framework & engine

\*\_snapshot\_data - 应用程序数据段

\*\_snapshot\_instr - 应用程序指令段

## iOS



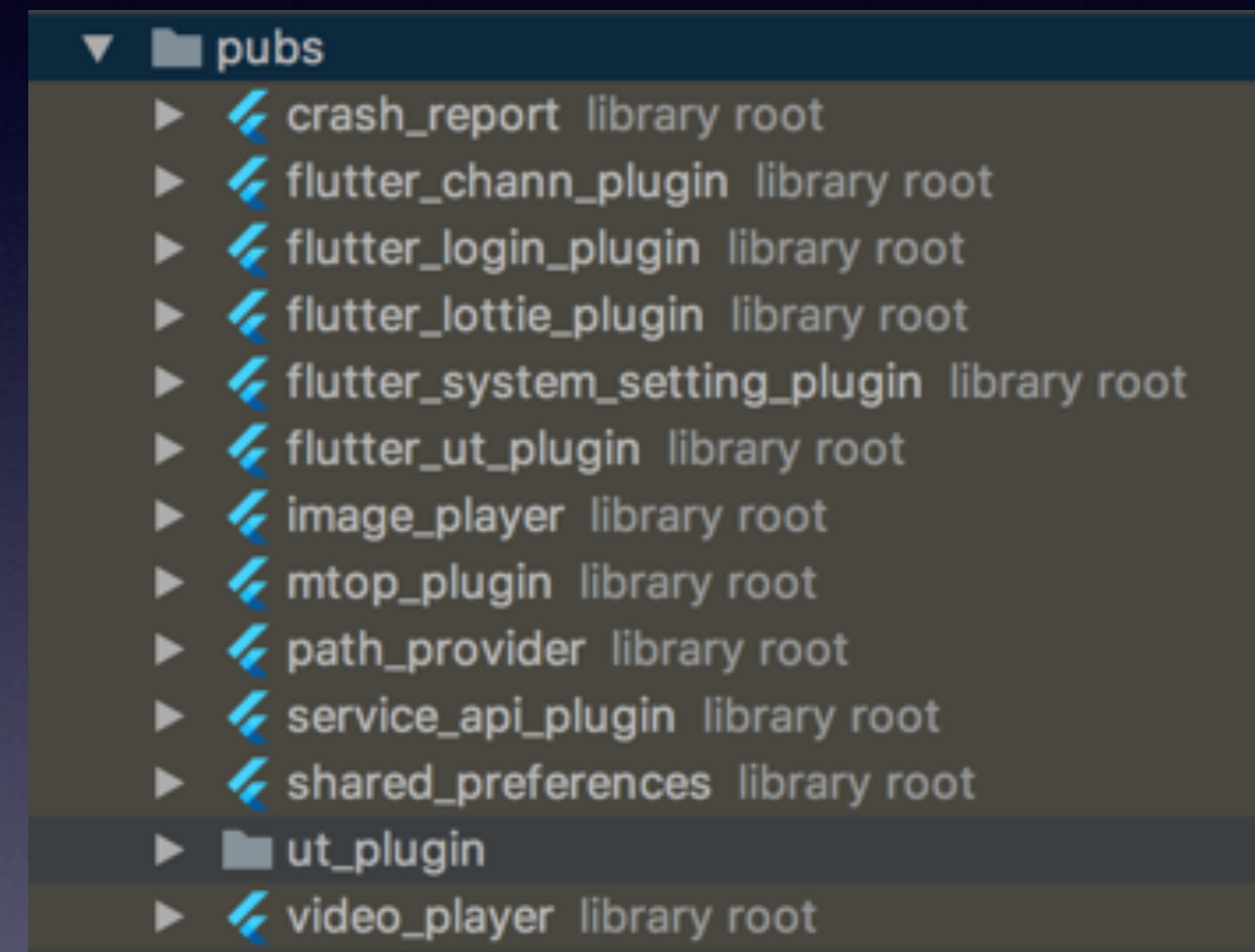
# 混编工程沉淀小结

- Submodule
- Pod/Gradle处理
- 中间产物
- engine, flutter\_tools



# 阿里生态适配

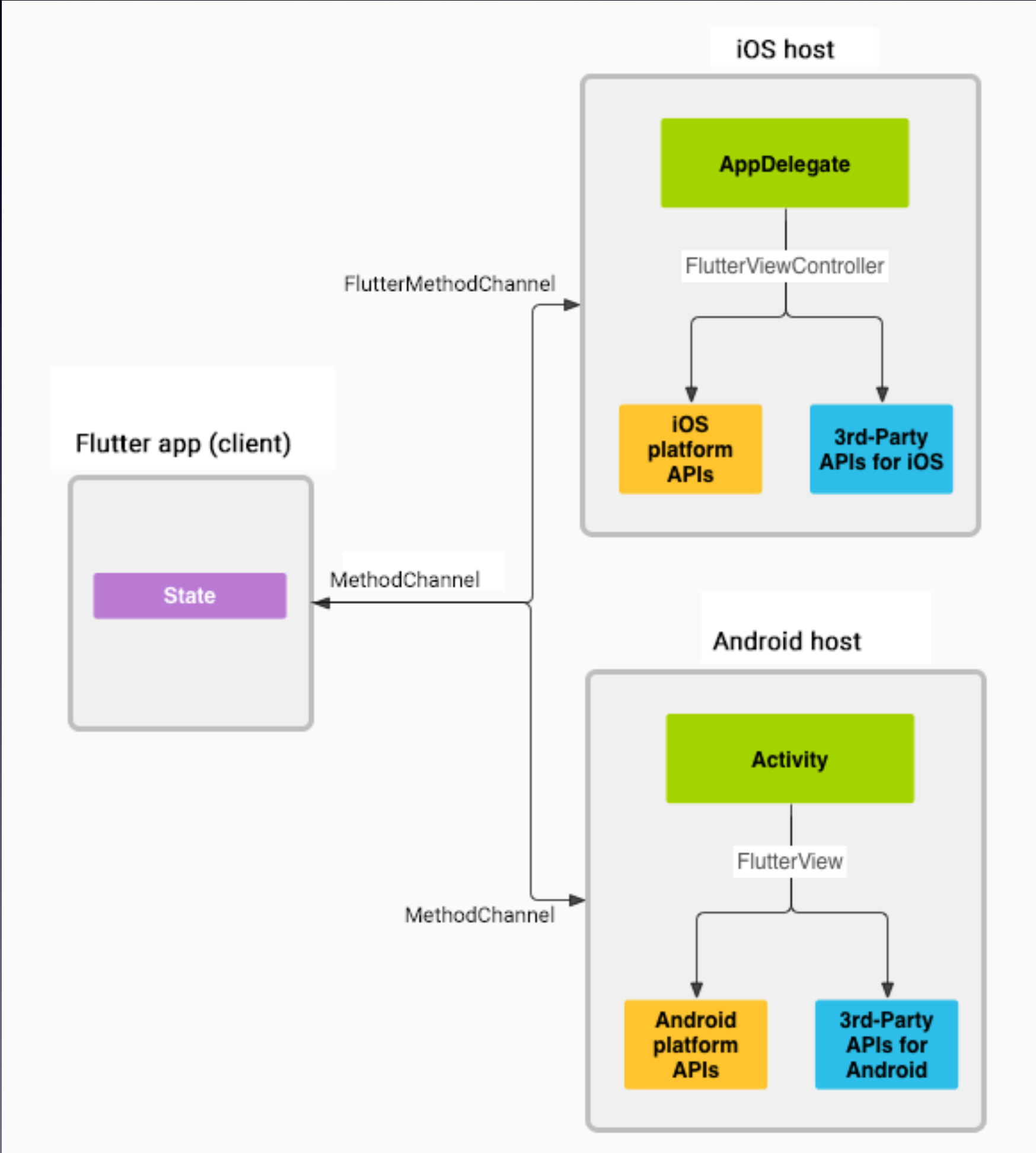
- UT埋点
- 登录、账号
- Mtop
- TBVideoPlayer
- Crash 魔兔
- Orange



沉淀：

可以直接Git引用，后续可建私有仓库

# 生态适配原理及性能



## MethodChannel性能

- iPhone7&iOS11
- 连续发送1000个包，计算平均吞吐量

包大小(B)	平均单个包耗时 (ms)
100	0.28
500	0.323
1K	0.386
10K	1.478
100K	11.665



# Texture对接自定义视频播放器



Flutter->Render(layer\_tree)

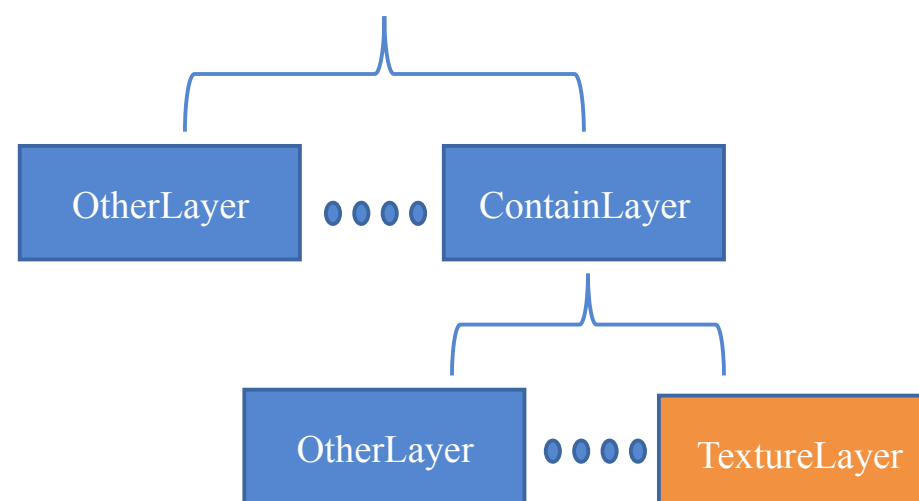
Flutter侧创建当前屏幕状态下的layer\_tree，并调用Render渲染

Rasterizer->Draw(layer\_tree)

Rasterizer 和传统GPU光栅化概念不一样，这里其实就是一个layer\_tree的渲染(Paint)并呈现(Present)在屏幕上的过程

LayerTree->Paint()

LayerTree是一个渲染树，Paint方法将会递归的遍历每一个节点并执行节点的Paint方法

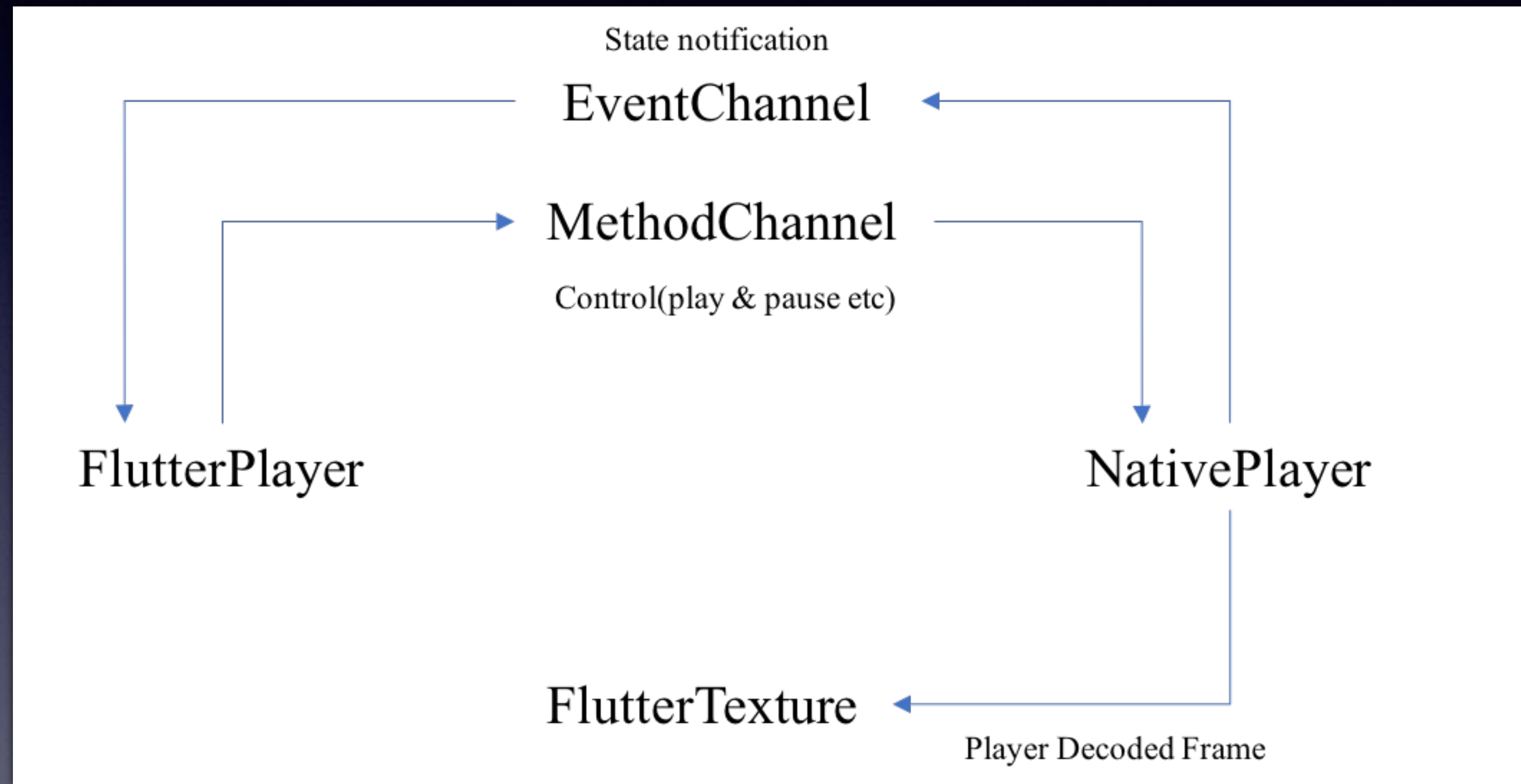


TextureLayer是一个特殊的Layer叶子节点，调用通过不同平台的实现可以实现External\_Texture的渲染

SKCanvas->Draw

所有的Layer最终都是调用SKIA的Canvas来实现渲染，我们的External\_texture，会封装成一个SKImage，交给SKCanvas渲染。

# Texture对接自定义视频播放器



- 1: 设置一个EventChannel, 用于向Flutter通知视频状态变化
- 2: 设置一个MethodChannel, 用于控制video player
- 3: 设置一个FlutterTexture, 用于显示视频帧
- 4: 从Native播放器中提取出video frame, 贴到FlutterTexture.



# 图片性能优化

漂亮。8成新，原价RMB2000。  
本交易支持自提、当面交易、邮寄



担保交易

浏览46



默认的图片缓存策略：

“

class ImageCache

Implements a **least-recently-used** cache of up to **1000** images. The maximum size can be adjusted using [maximumSize].

”

问题：

内存占用



# 图片性能优化

优化策略1，对于单页面场景，简单就是美：

1，设置最大缓存图片的数量

```
class MyWidgetsFlutterBinding extends
WidgetsFlutterBinding {

    @override
    ImageCache createImageCache() {
        ImageCache cache = new ImageCache();
        cache.maximumSize = 100; // the maximum num
of cached images.
        return cache;
    }
}
```

2，图片尺寸自适应剪裁

- 图片宽度统一泛化为一系列标准尺寸，如  
60, 128, 234 ... 640, 720, 960, ...
- 与CDN策略一致

优化策略2，对于多页面场景，考虑深入优化：

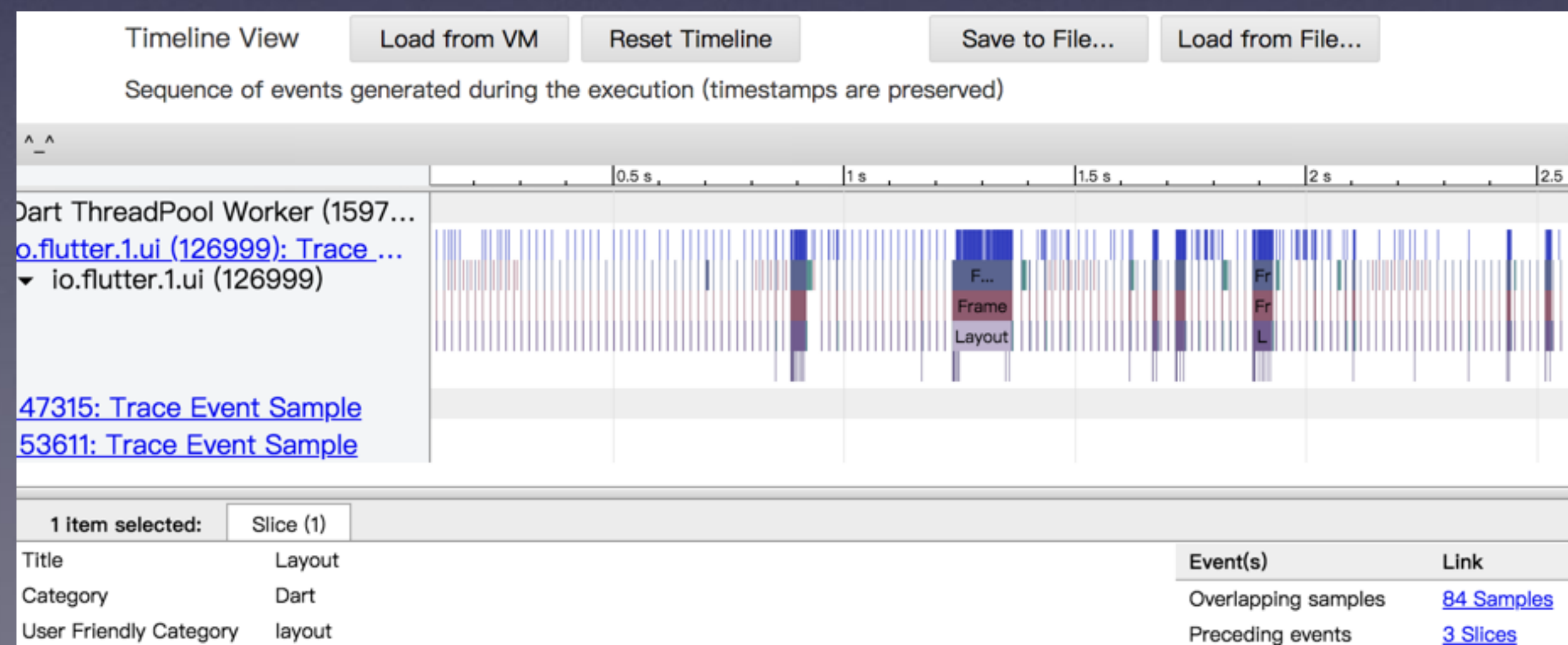
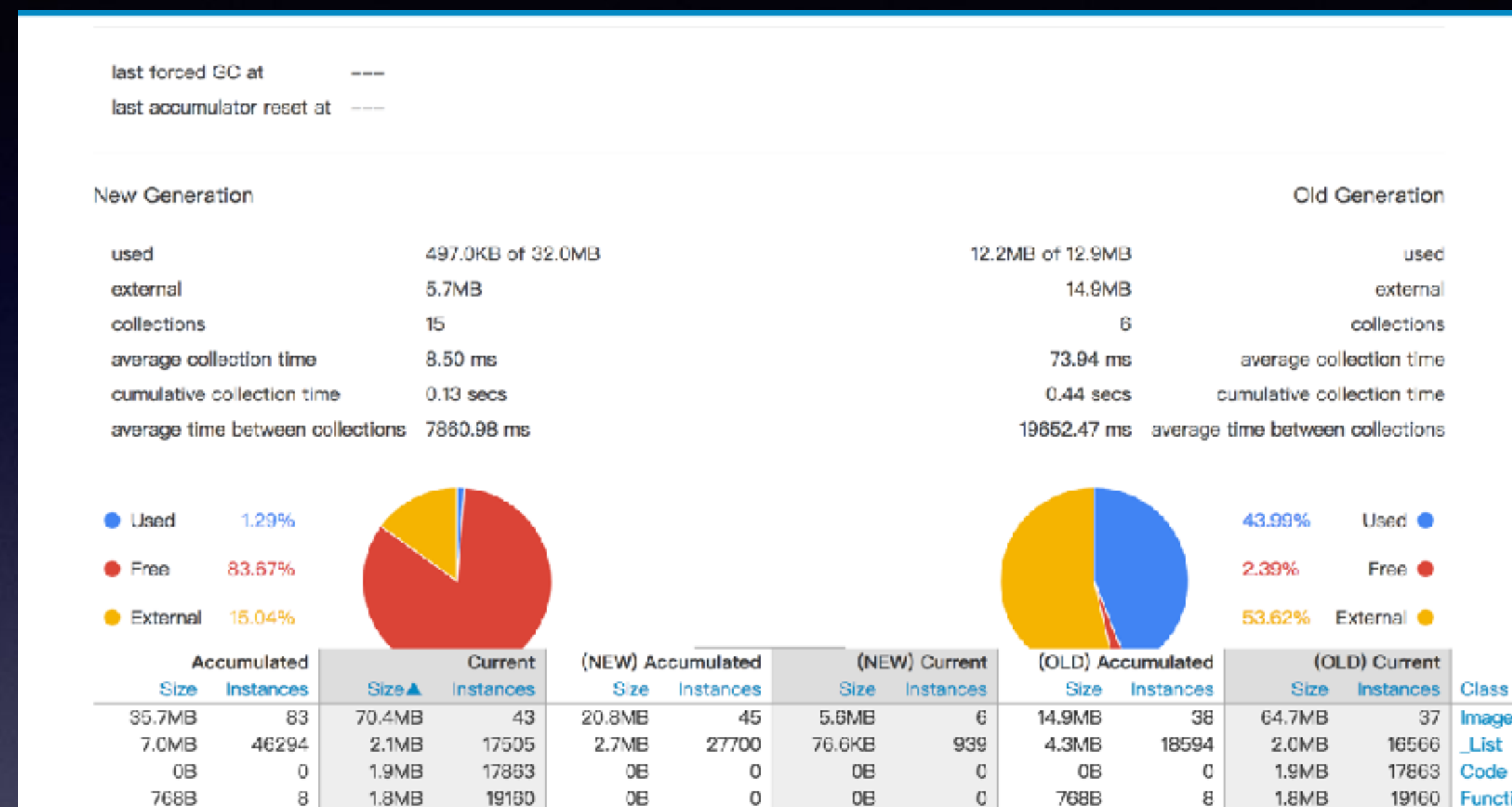
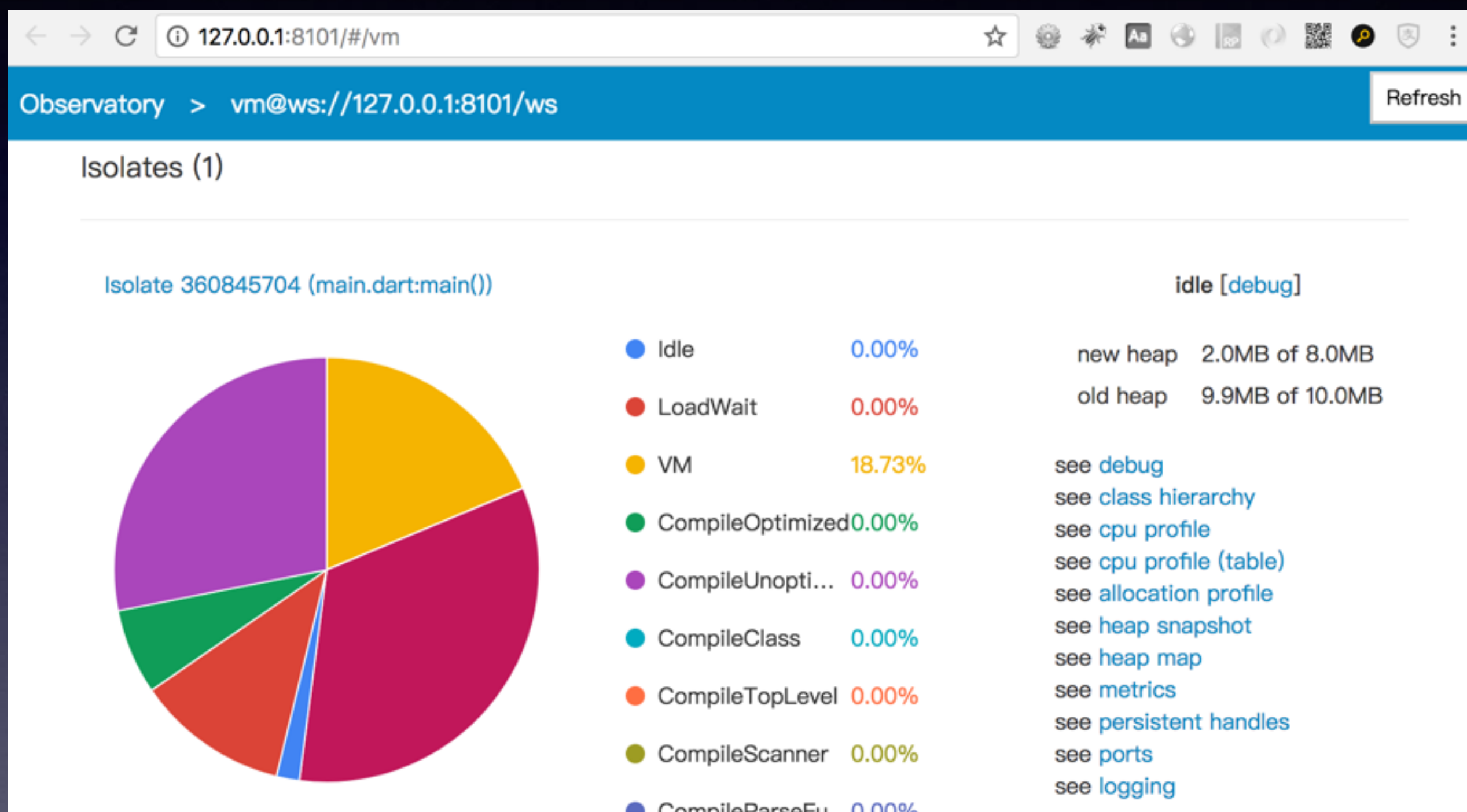
- 增加磁盘多级缓存
- 混编下，考虑与Native缓存复用



# Table of Contents

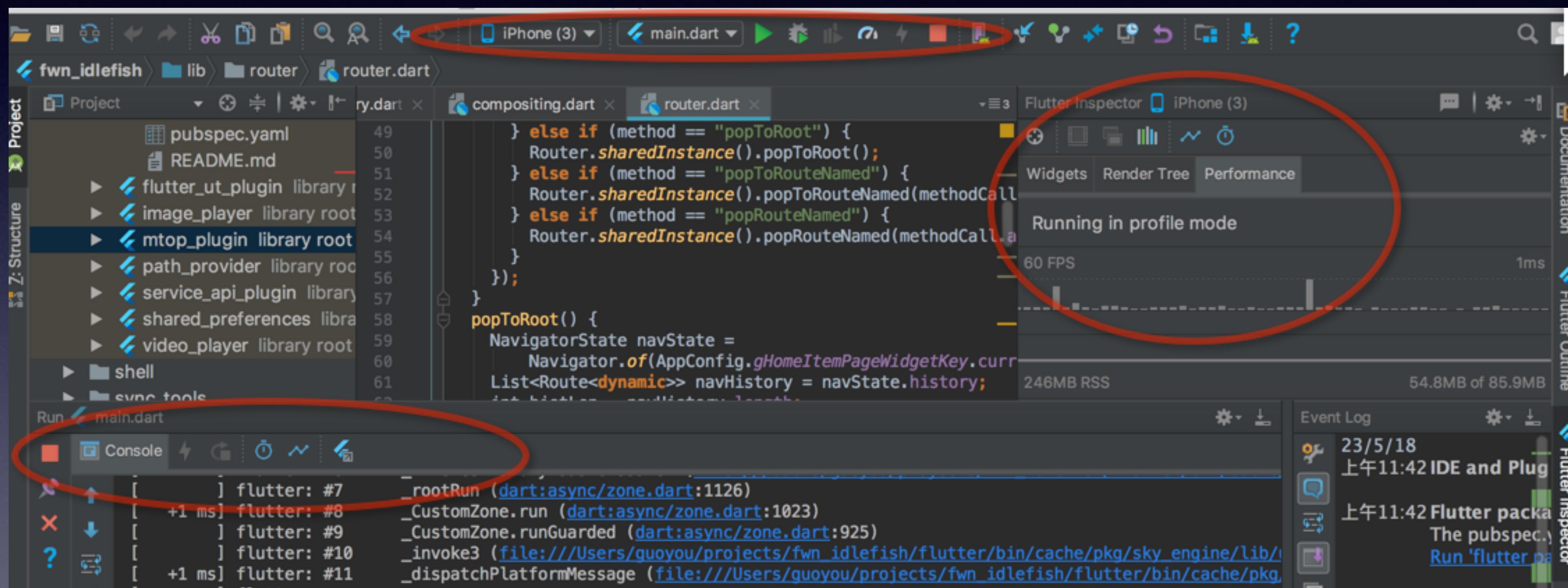
- Flutter简介
- 闲鱼项目已解决的问题
- Flutter上线效果，性能对比，成熟度
- 分模块技术简介
  - 混合栈解决方案
  - 混合工程方案
  - 视频、图片
  - 原有生态适配
- 使用体感 - 工具链、问题排查
- 小结

# 工具链-Obsevatory





# 工具链 - Flutter Inspector



# 小结 & FAQ

闲鱼：

- 混合栈管理
- 混编工程化
- 阿里私有仓库
- 编译产物优化

Google：

- 兼容性
- 中文字体

后续交流： 敬请关注6.25左右的活动，Flutter官方计划，底层原理等。



Thanks