

MTFlutter: 美团外卖 Flutter 容器 化生态建设实践

陈航

美团点评高级技术专家

InfoQ官网 全新改版上线

促进软件开发领域知识与创新的传播



关注InfoQ网站
第一时间浏览原创IT新闻资讯



免费下载迷你书
阅读一线开发者的技术干货

自我介绍



美团外卖商家终端团队负责人，极客时间《Flutter 核心技术与实战》作者。

曾就职于雅虎北研和新浪微博，2015年加入美团，经历了外卖用户端和商家端发展的多个阶段，推动了外卖移动端架构演进、线上运维及终端技术栈融合等相关工作。

目录

- 业务背景
- Flutter 的优势和挑战
- Flutter 容器化生态建设
- 展望和总结

美团外卖 Flutter 实践历史

2018 Q2

美团外卖用户端调研

美团技术博客: *Flutter* 的原理及美团的实践

美团技术博客2018年阅读量前三

2018 Q4

美团外卖商家端接入

确定技术选型方向, 铺开业务实践, 沉淀基建

业务需求占比>30%, 页面数>15, 指标与 *Native*持平

2019 Q2

美团点评多个事业部中后台业务线上大规模应用

MTFlutter 容器化基础设施及周边工具链建设

截止至12月15日, 共10个 *BU*, 12个 *App* 接入

中后台业务特点

- 面向非用户

标准化程度低（1个 C 端 App & 页面对应 N 个 B 端 App & 页面），逻辑重，交互复杂，主要解决数据生产、运营管理的问题，**多端一致性要求高**

- 生产工具类应用

使用时间长（持续十几个小时），**性能和稳定性要求高**

- 产品迭代周期短

产品两周一迭代，每个迭代需求众多，**开发效率要求高**

目录

- 业务背景
- Flutter 的优势和挑战
- Flutter 容器化生态建设
- 展望和总结

MFlutter

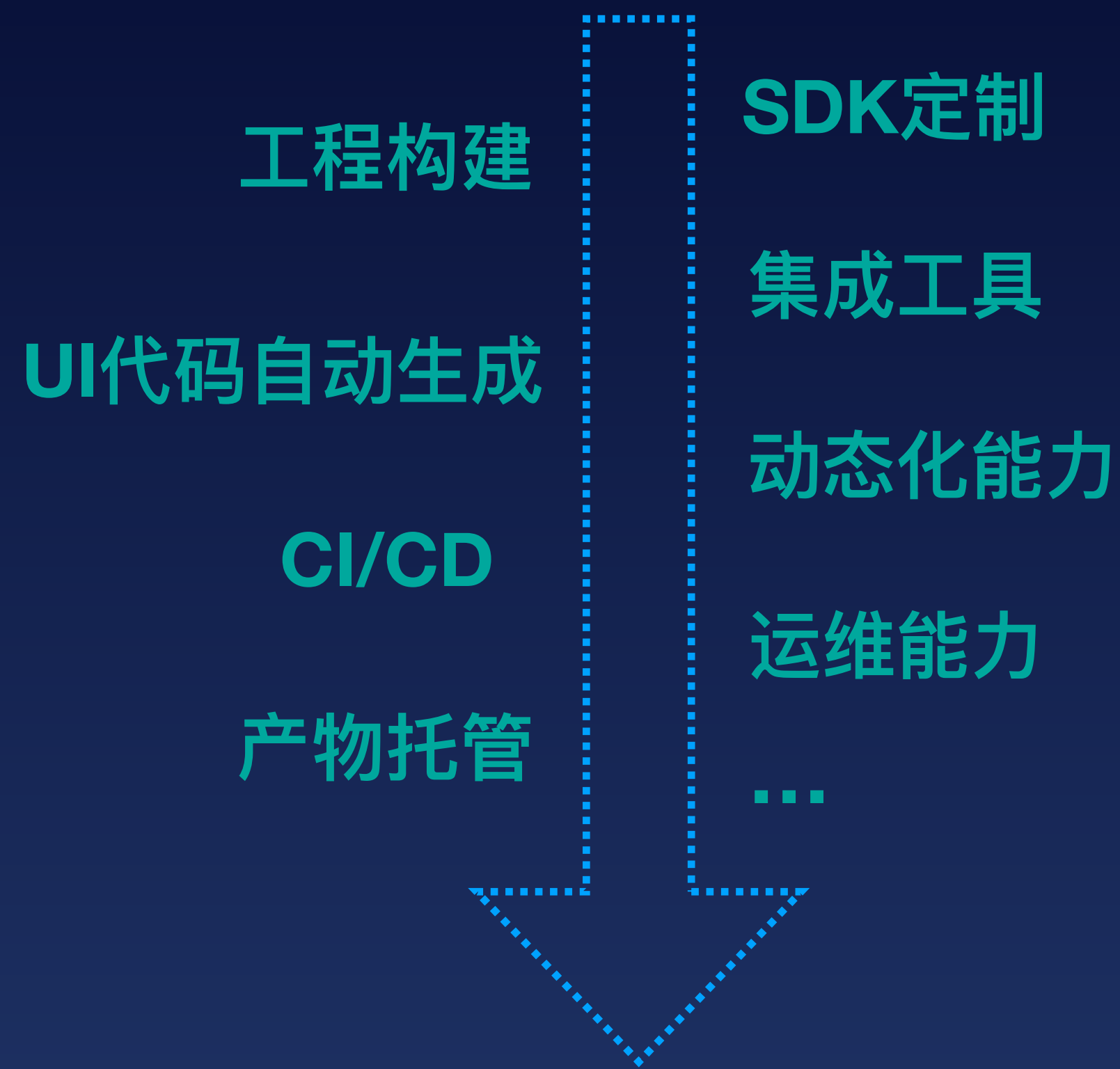
适用场景

效率、性能、稳定性、多端体验一致

动态性、包大小无特殊诉求



Flutter

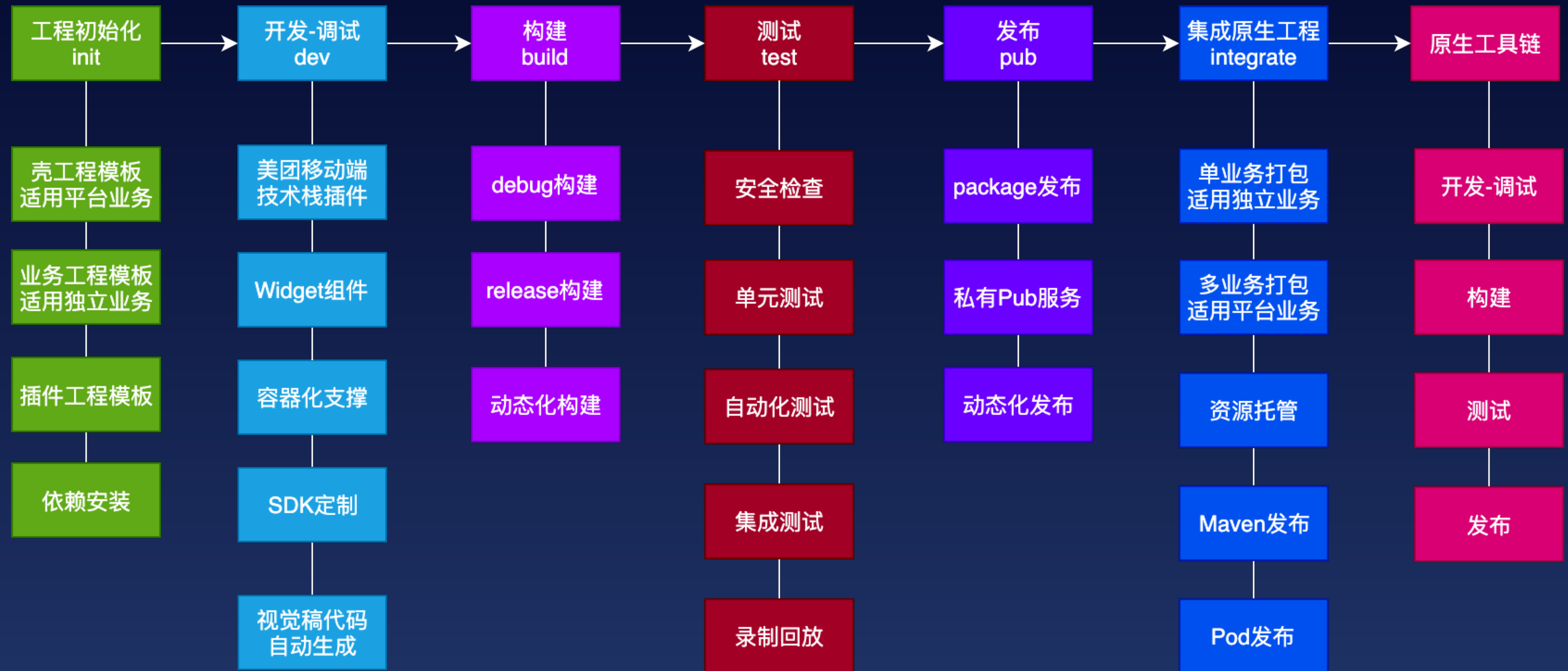


MFlutter

目录

- 业务背景
- Flutter 的优势和挑战
- Flutter 容器化生态建设
 - MTFlutter 工具链
 - 业务支撑
- 展望和总结

MTFlutter 标准 workflow



将作为原生工程上游的 Flutter 模块开发，抽象为原生依赖产物的工程管理，并提炼出对应的工作流

MFlutter – 脚手架

解决环境配置、团队/项目环境统一

打包 workflow 命令行工具

宿主/CI环境



SAKGems

```
gem install flutter_cli -v "0.3.3"
```

美团云

```
flutter_linux_v1.2.1-hotfix.1-stable.tar.xz  
flutter_macos_v1.2.1-hotfix.1-stable.tar.xz
```

- install [version] —autodotfiles
- uninstall
- upgrade [version]
- downgrade [version]
- help
- config —project=path —sdk-version
- config —project=path —consistent-sdk

工程
初始化

开发
调试

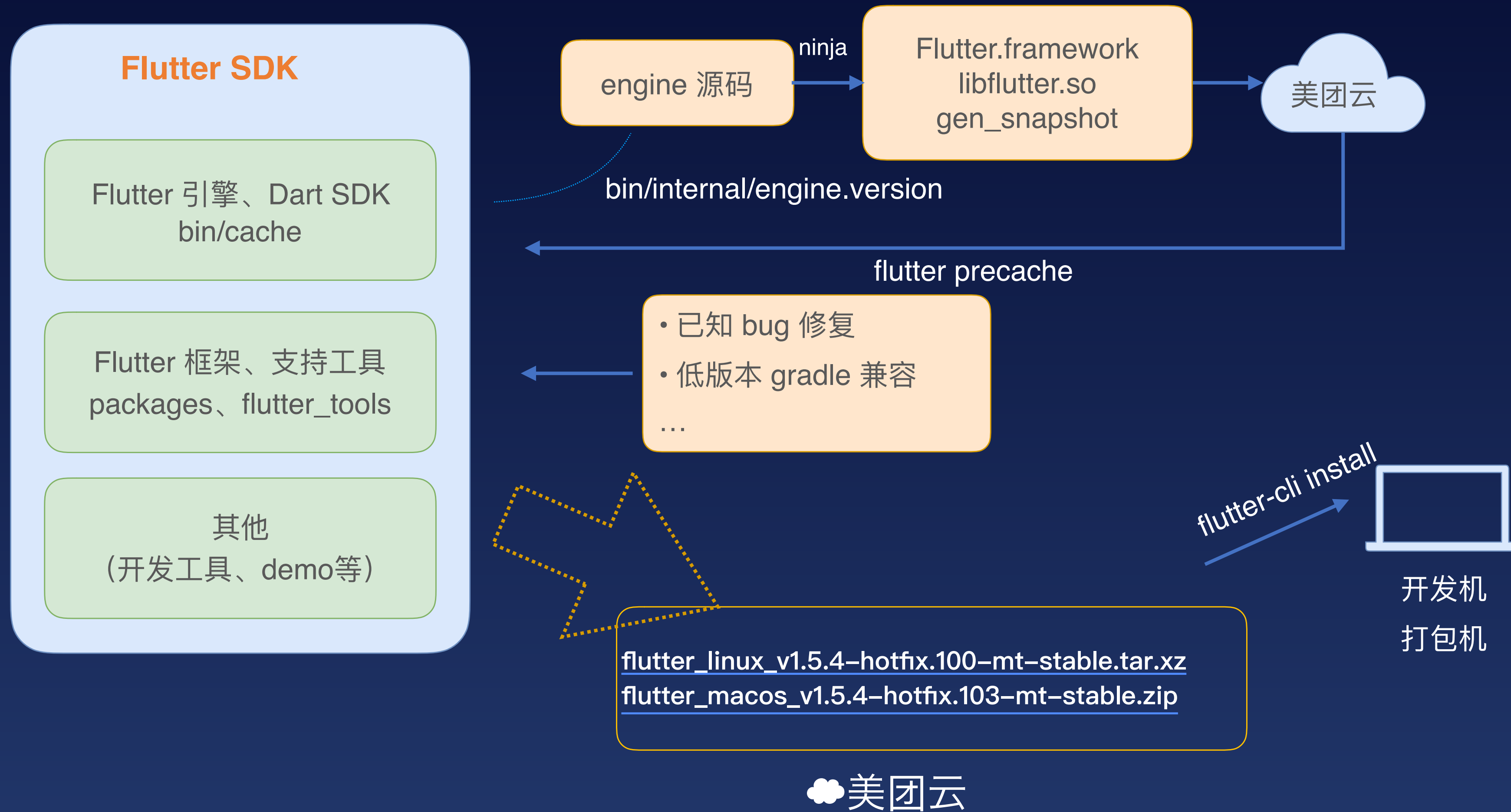
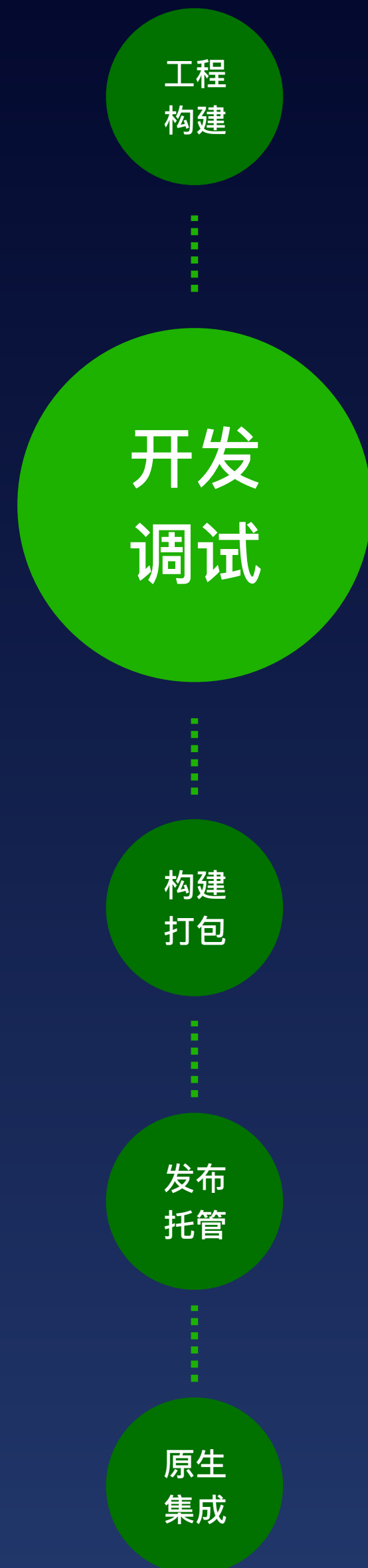
构建
打包

发布
托管

原生
集成

MFlutter – SDK定制

定制模板工程，适配公司基建生态，修复已知问题等，持续提供免坑版SDK



MFlutter – CI/CD

解决 Flutter 项目打包、依赖关联问题

1. Dart 产物编译
2. 业务资源复制
3. Flutter 引擎集成
4. 原生侧编译流程对接
5. 三方插件处理
6. 版本号管理

工程构建

开发调试

打包构建

发布托管

原生集成

新建项目

没有本地工程？试试 Talos 模板市场(基于 Atoms Store)

* 项目名 ?

a

仓库地址

ssh://git@git.sankuai.com/~lihuichao/a.git

* 发布地址 ?

内网 外网

* 归属团队 ?

到家事业群/iOS研发

* 项目级别 ?

C 级

* 业务类型 ?

商户型

* 项目类型 ?

MFlutter

* 项目负责人 ?

lihuichao @meituan.com / @dianping.com / @maoyan.com

删除负责人

增加负责人

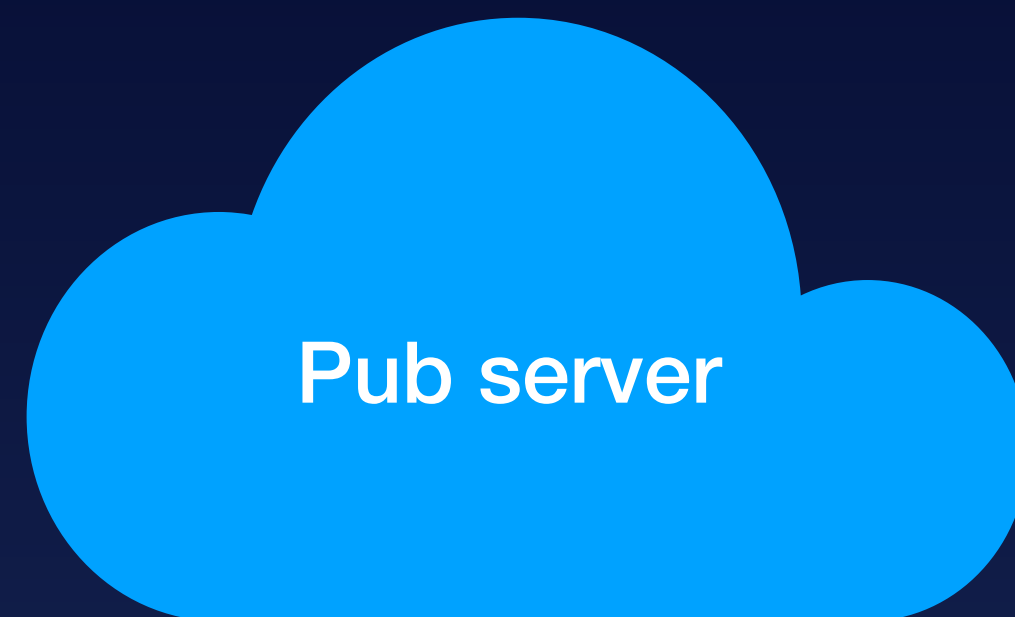
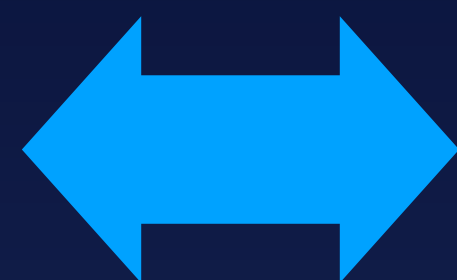


MFlutter – Dart代码托管

解决 Flutter Package 发布托管问题

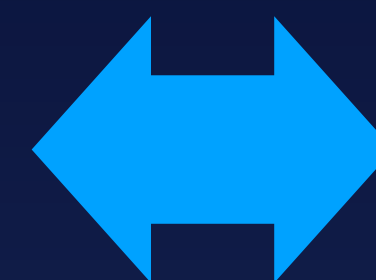
<https://pub.sankuai.com>

pub client



Pub server

nodeJS + sso

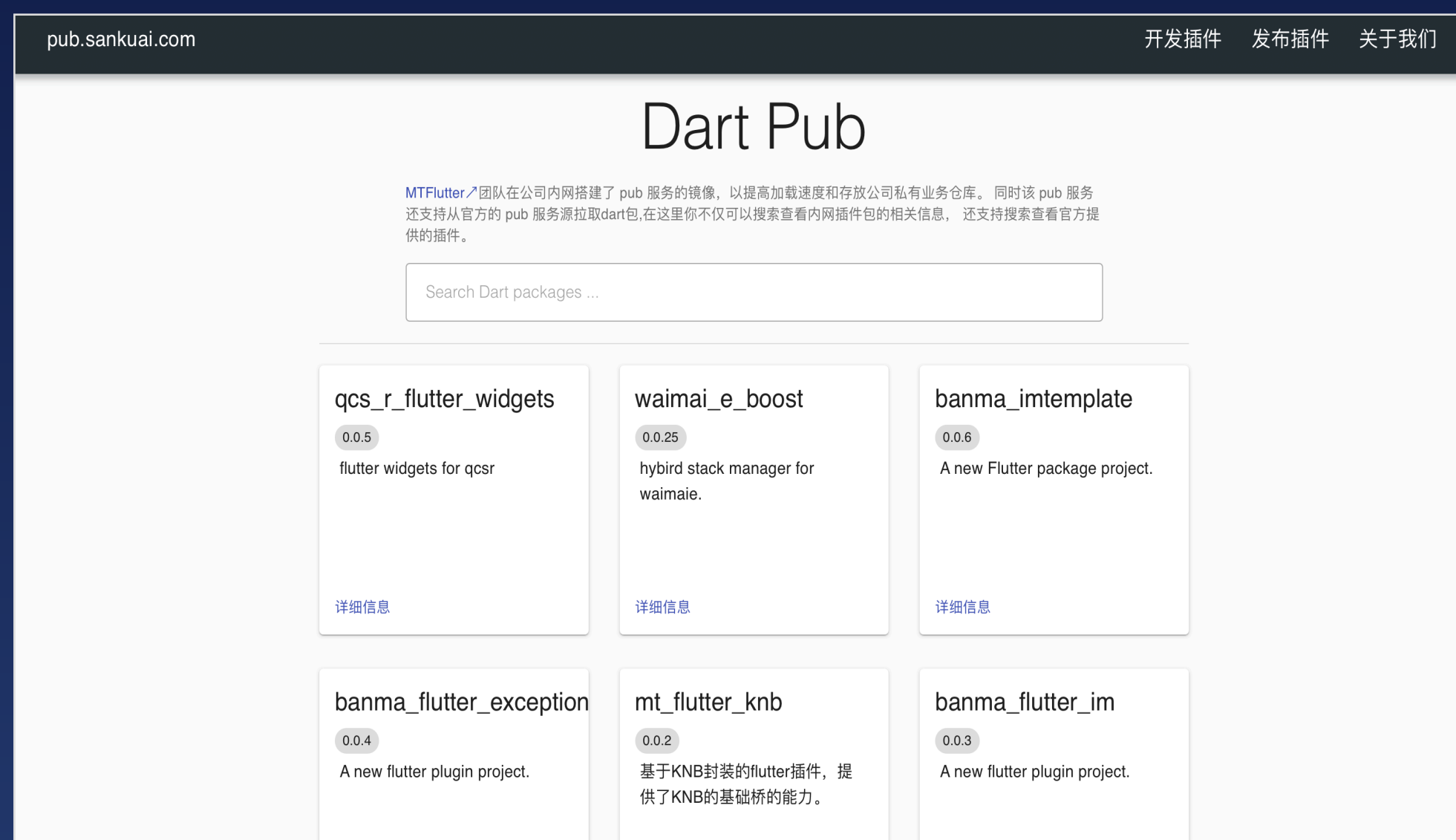


美团云

packages

Flutter Package

```
├─ lib
│   └─ xxx.dart
│   └─ main.dart
├─ pubspec.lock
├─ pubspec.yaml
└─ test
    └─ counter_test.dart
├─ flutter_hello_world.iml
```



工程构建

开发调试

打包构建

发布托管

原生集成

MFlutter – 二进制产物托管

解决Flutter项目原生构建产物托管问题

业务部分 (代码与资源)

```
android: snapshot*...  
iOS: App.framework  
资源: flutter_assets
```

框架部分

```
android: flutter.jar  
        libflutter.so  
iOS: flutter.framework
```

flutter plugins

```
android: plugin1 proj  
        plugin2 proj  
  
iOS:   plugin1 local pod  
      plugina local pod
```

Flutter Project

```
├─ lib  
  └─ main.dart  
├─ assets
```

```
├─ .ios  
├─ .android
```

```
├─ pubspec.yaml  
├─ .flutter-plugins
```

美团云

工程
构建

开发
调试

打包
构建

发布
托管

原生
集成

MFlutter – 集成插件

以原生方式解决中大型团队集成 Flutter Module 的侵入问题

工程
构建

开发
调试

打包
构建

发布
托管

原生
集成

build.gradle

```
gradle.ext.flutterModuleAAR="com.example.business  
:your_flutter_module_name:1.2.0"  
apply plugin: 'flutter_integration'
```

Native
Project

AAR

Pod

集成插件

Gradle

Cocoapods

美团云

Gemfile

```
source 'http://sakgems.sankuai.com' do  
  gem 'cocoapods-flutter-plugin', '0.0.27'  
end
```

Podfile

```
target :your_target_name do  
  flutter 'your_flutter_module_name', '1.2.0'  
end
```

业务部分 (代码与资源)

```
android: snapshot*...  
iOS: App.framework  
资源: flutter_assets
```

框架部分

```
android: flutter.jar  
        libflutter.so  
iOS: flutter.framework
```

flutter plugins

```
android: plugin1 proj  
        plugin2 proj  
iOS:   plugin1 local pod  
        plugina local pod
```

目录

- 业务背景
- Flutter 的优势和挑战
- Flutter 容器化生态建设
 - MTFlutter 工具链
- 业务支撑
- 展望和总结

MFlutter – UI 代码自动生成

设计痛点：前端还原度低

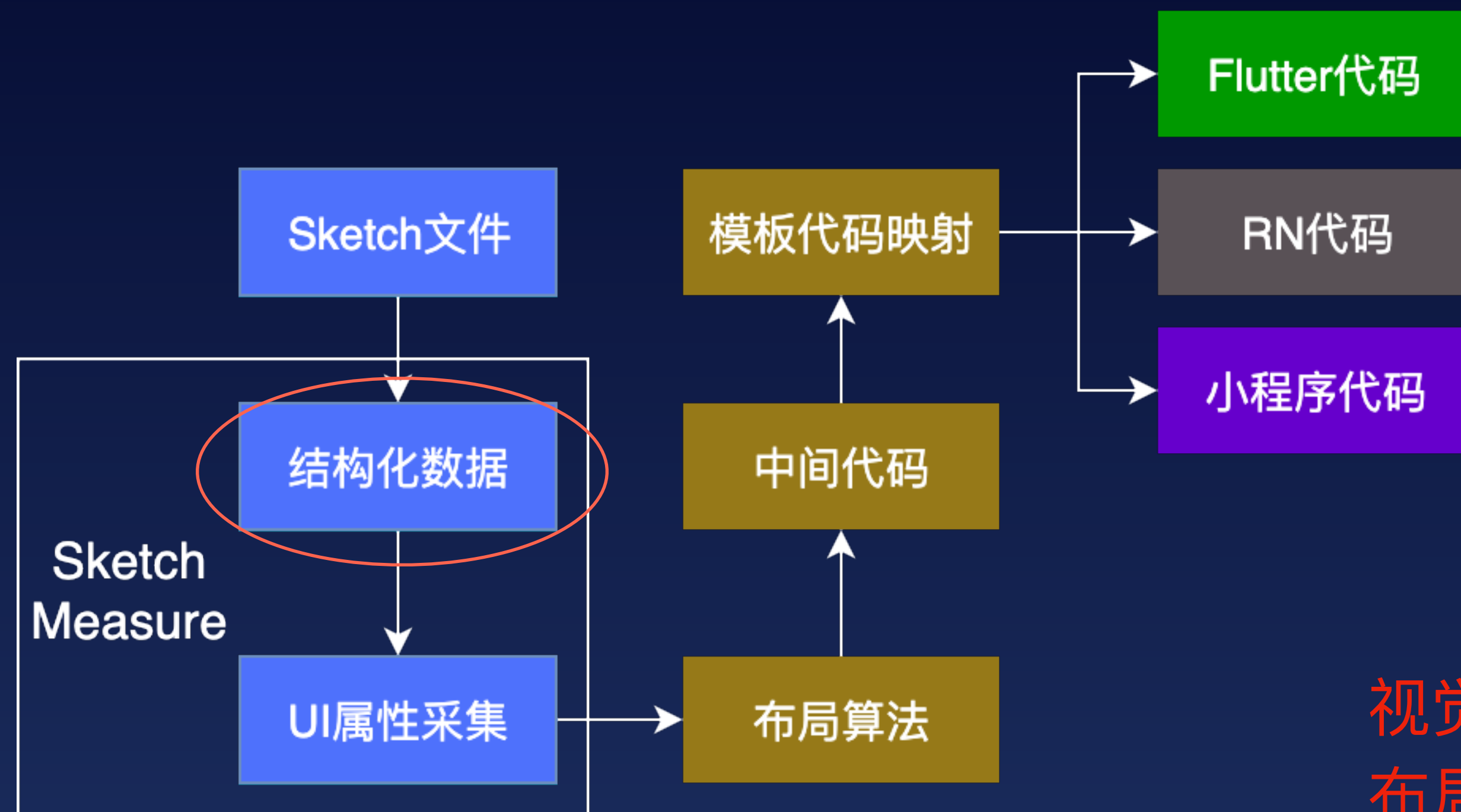
- 开发说实现不了
- 卖家秀 VS 买家秀

研发痛点：设计风格不统一、大量界面开发调试工作

PM 痛点：改个样式动则5+天

目标：打通视觉稿转换 UI 代码链路，减少信息/资源损耗

MFlutter – UI 代码自动生成



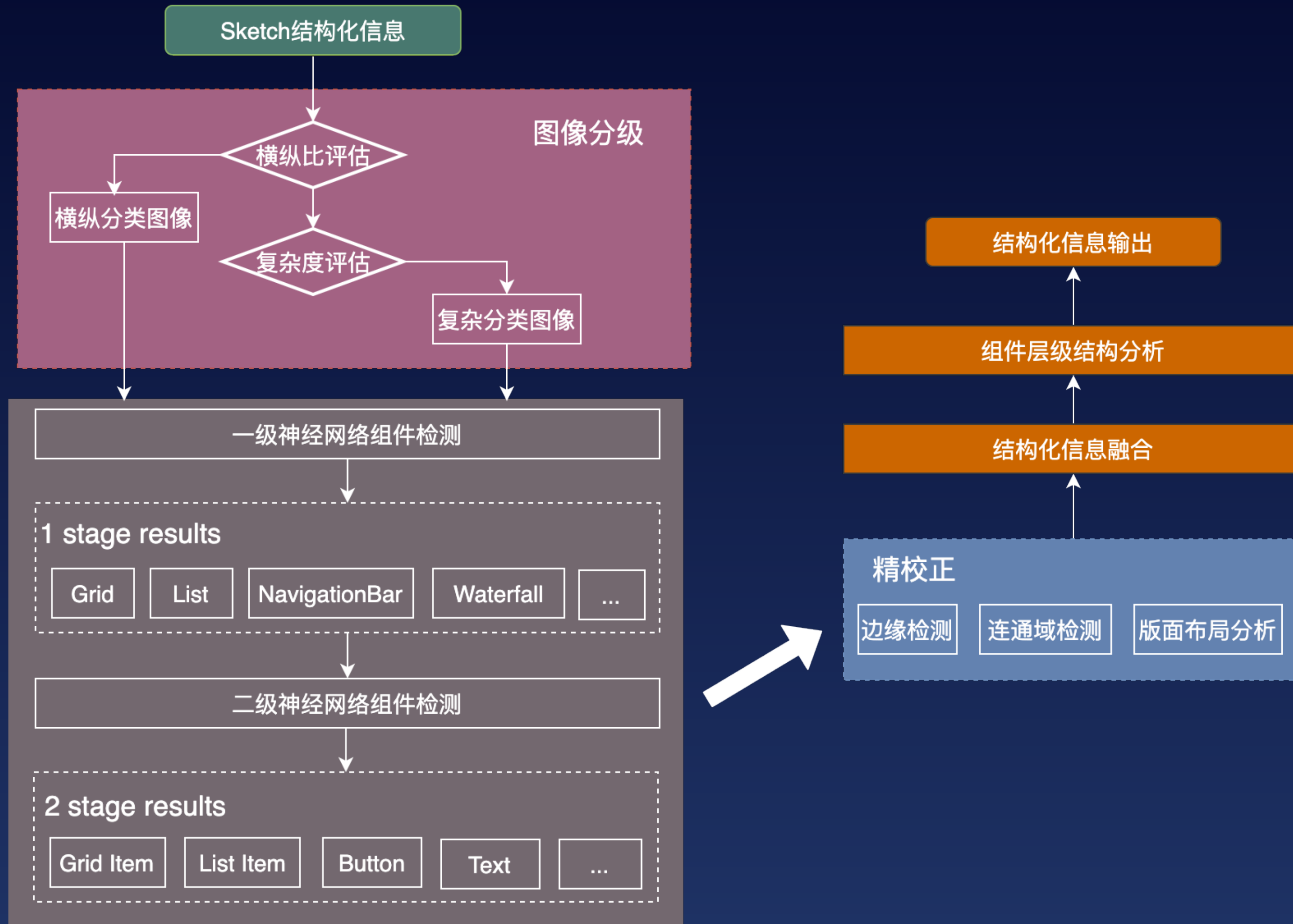
普通方案

1. Sketch 插件导出视觉结构化信息
2. 切割布局算法转成中间数据结构
3. 语义模板库转成目标代码

视觉稿结构与代码一一对应，组件识别粒度低，布局层级冗余，可维护性低

如何优化结构化数据？

MFlutter – UI 代码自动生成

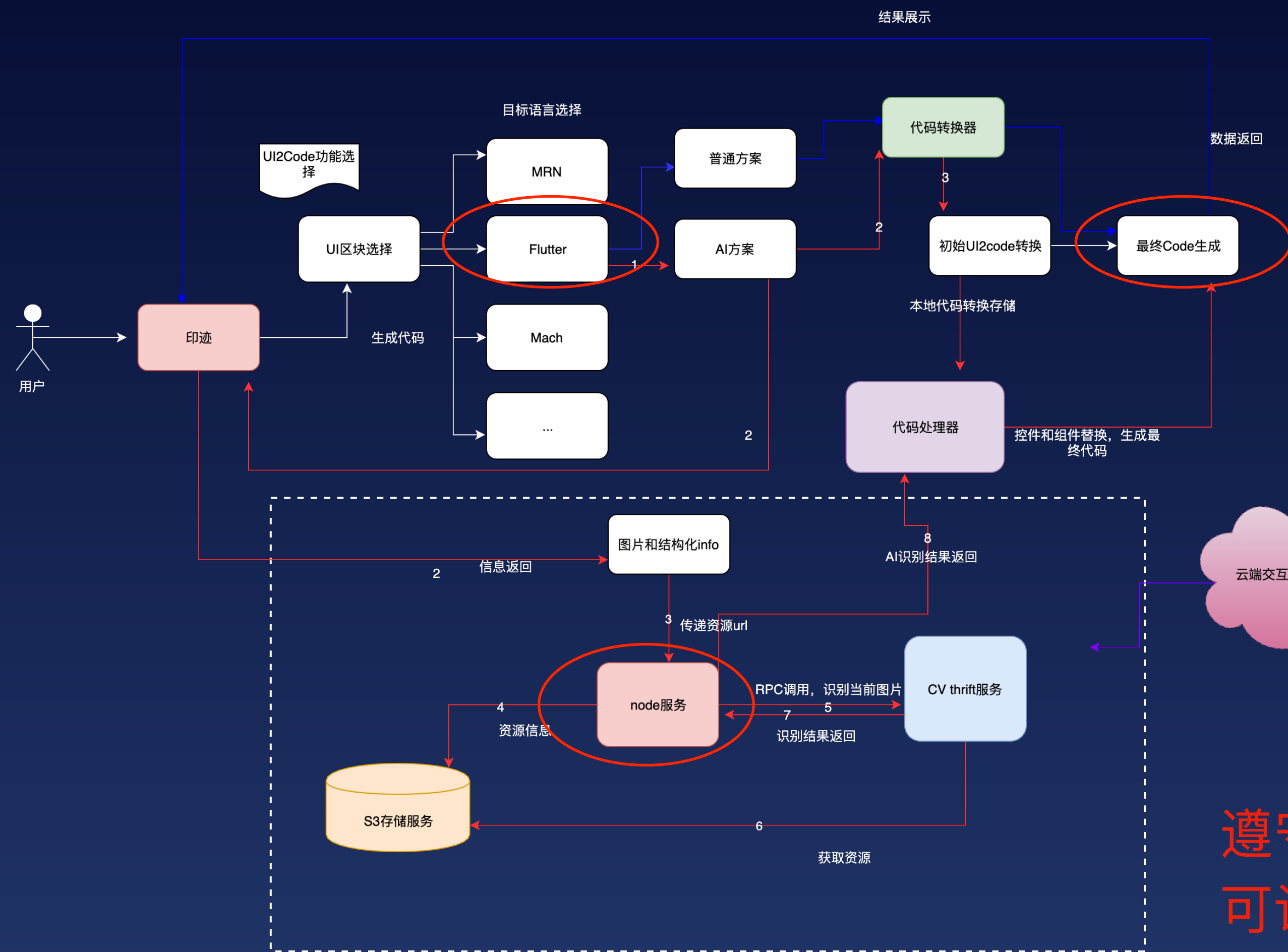


视觉结构优化

1. 图像分级预处理, 优化识别度
2. 多级卷积神经网络结构实现组件检测
3. 校正识别结果
4. 相似度匹配算法融合原始视觉结构

by 美团无人配送与视觉部/视觉图像中心

MFlutter – UI 代码自动生成



AI 方案

1. Sketch 插件导出视觉结构化信息
2. AI 识别服务进行版面分析、组件识别、纠错及替换
3. 切割布局算法转成中间数据结构
4. 语义模板库转成目标代码

遵守设计规范，贴近开发人员的书写风格和习惯，可识别高阶组件，无冗余层级，可维护性高

MFlutter – UI 代码自动生成



视觉稿直转 Flutter、RN、小程序代码，提升研发效率

视觉规范集成至云协作平台，遵守设计与研发习惯

复用高阶组件，布局层级清晰，可维护性好

视觉稿云协作平台-印迹

MFlutter – 动态化

动态化技术是运营类业务必备的基础能力，也是缩减包大小的重要手段

但出于应用市场合规、性能和安全考虑，Flutter 官方并无支持计划

业界两大方向：

产物替换

- vm_snapshot_data、vm_snapshot_instr、isolate_snapshot_data、isolate_snapshot_instr

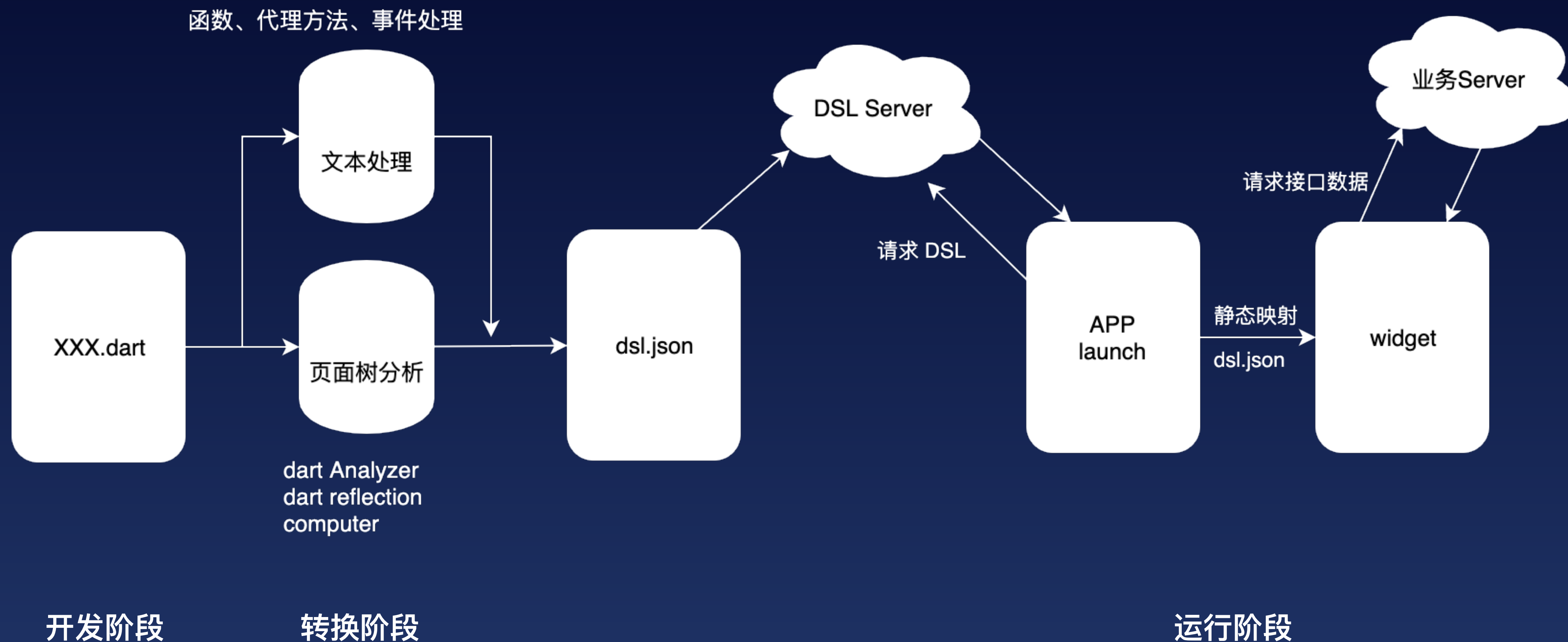
仅支持 Android，且必须是纯 Flutter 应用 ❌

DSL

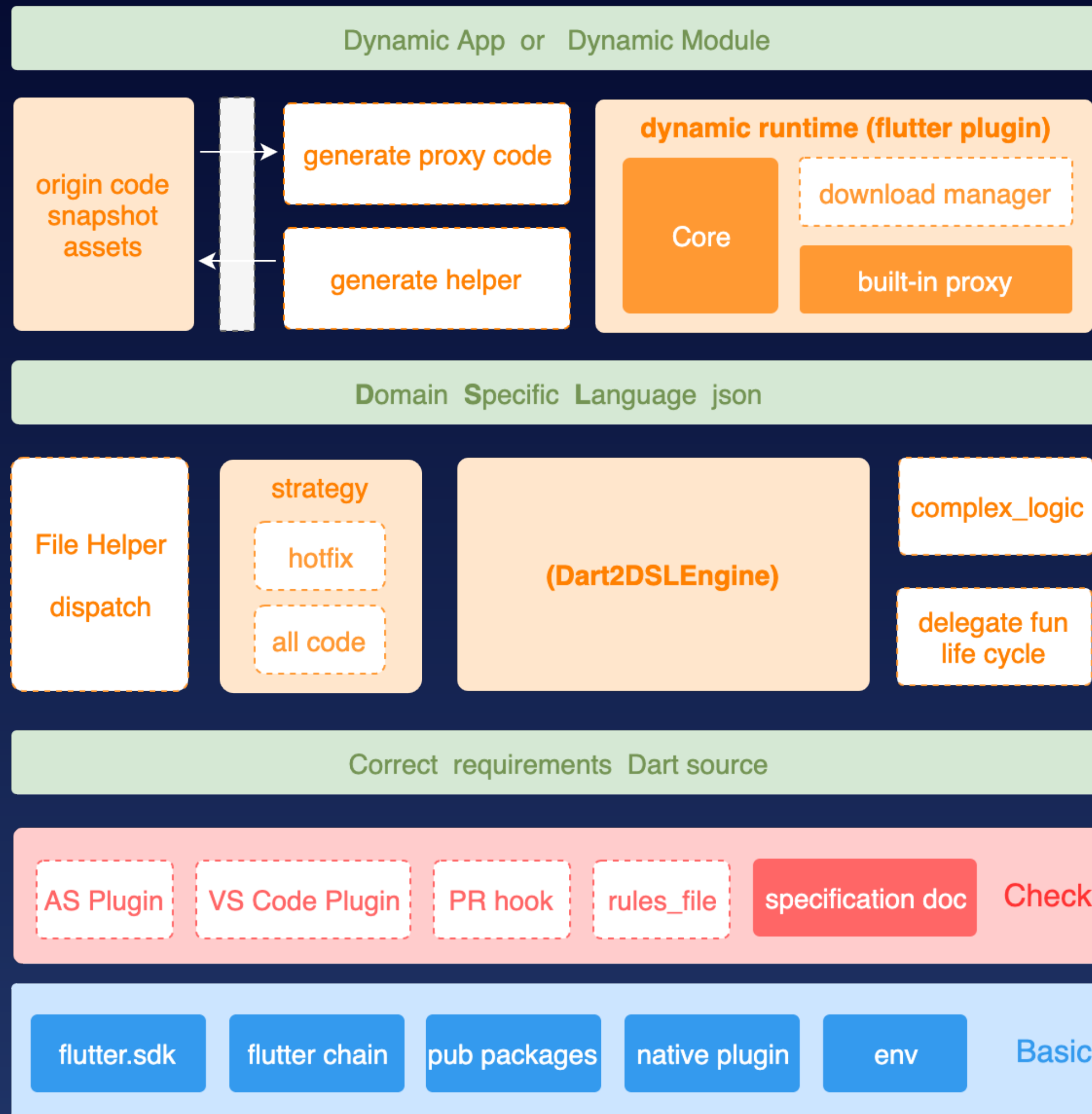
- 动态解析 (JS 运行时解析)
- 静态解析 ?

引入 JS 虚拟机，包大小增加，且技术栈需要切换为 JS ❌

MFlutter – 动态化



MFlutter – 动态化



运行阶段

以静态映射的方式执行 JSON DSL
使用 proxy 处理 DSL 与源码的互相调用

转换阶段

通过 Dart Analyzer 得到根节点
解析 AST 转化成完整 DSL 数据结构

开发阶段

支持通用编程范式，不支持写法插件会报错

MFlutter – 动态化



沿用 Flutter 生态，无技术栈切换成本
线下解析，支持常规开发范式
静态渲染，无运行时依赖，执行效率接近 AOT
复用原生基建，支持随时发版、灰度能力

MTFlutter/Dymanic各阶段产物流转

目录

- 业务背景
- Flutter的优势和挑战
- Flutter容器化生态建设
- 展望和总结

MFlutter – 总结

1. 中后台业务的特点和 Flutter 的优势

- 标准化程度低、使用时间长、迭代周期短
- 效率、性能、稳定性、多端体验一致

2. 美团外卖 Flutter 生态建设

- 标准 workflow 及 MFlutter 工具链
- UI 代码自动生成：引入 CV 算法，复用高阶组件，视觉规范云化、易用化
- 动态化：线下解析，静态渲染，支持通用编程范式，沿用 Flutter 技术栈

MFlutter – 规划

1. 业务复用

- 业务模板/标准框架/开发一体化
- 简化需求交付链路（视觉稿->动态 DSL->灰度->全量）
- SDK 瘦身

2. 终端技术融合

- 统一上层 DSL，按需选择底层渲染技术

TGO 鲲鹏會

汇聚全球科技领导者的高端社群

🏠 全球12大城市

👤 900+ 高端科技领导者

使命

Mission

为社会输送更多优秀的
科技领导者

愿景

Vision

构建全球领先的有技术背景
优秀人才的学习成长平台



扫描二维码，了解更多内容



批量购课
扫码联系顾问

5G 时代大前端进阶之路

重学前端 | 每天 10 分钟，重构你的前端知识体系

Android 开发高手课 | 由表及里，给你的 App“把把脉”

iOS 开发高手课 | 从原理到实战，带你解决 80% 的开发难题

React 实战进阶 45 讲 | 掌握当下热门的前端利器

Vue 开发实战 | 从 0 开始搭建大型 Vue 项目

Node.js 开发实战 | 带你开发一个完整的 Node.js 项目

Flutter 核心技术与实战 | 来自 Google 的高性能跨平台开发框架

DevOps 实战笔记 | 精要 30 计，让 DevOps 快速落地

批量购课特惠

购买本系列课程总价满 ¥1000，享 **8** 折优惠。

获取优惠，请联系客服「豆包」



13167596032

THANKS

GMTC
全球大前端技术大会