

Flutter在百度贴吧的落地实践

高飞

百度资深研发工程师

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

高飞

11年加入百度贴吧，资深研发工程师，百度贴吧客户端技术负责人



目录

- 为何引入Flutter
- 实践之旅
- 总结

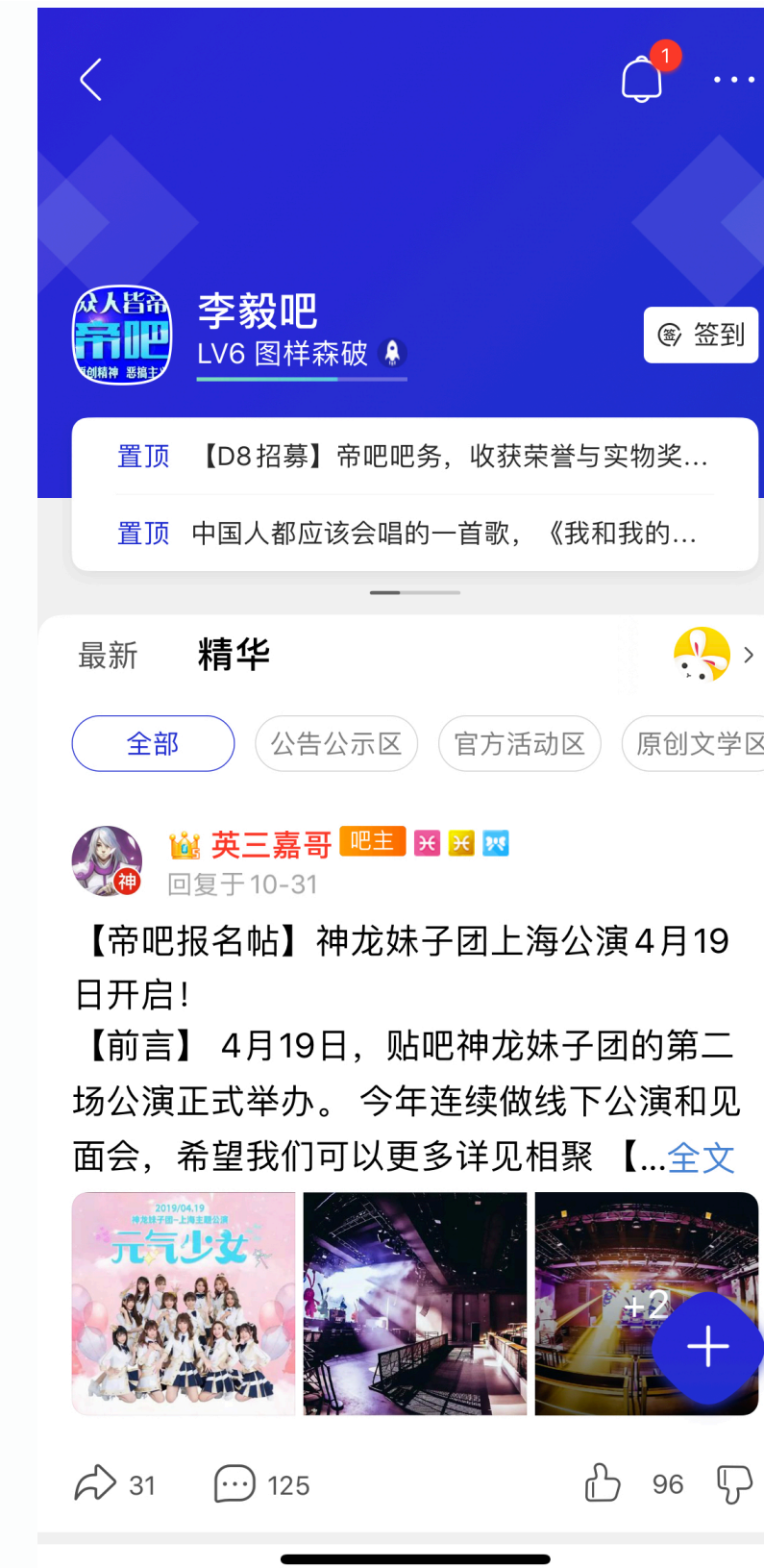
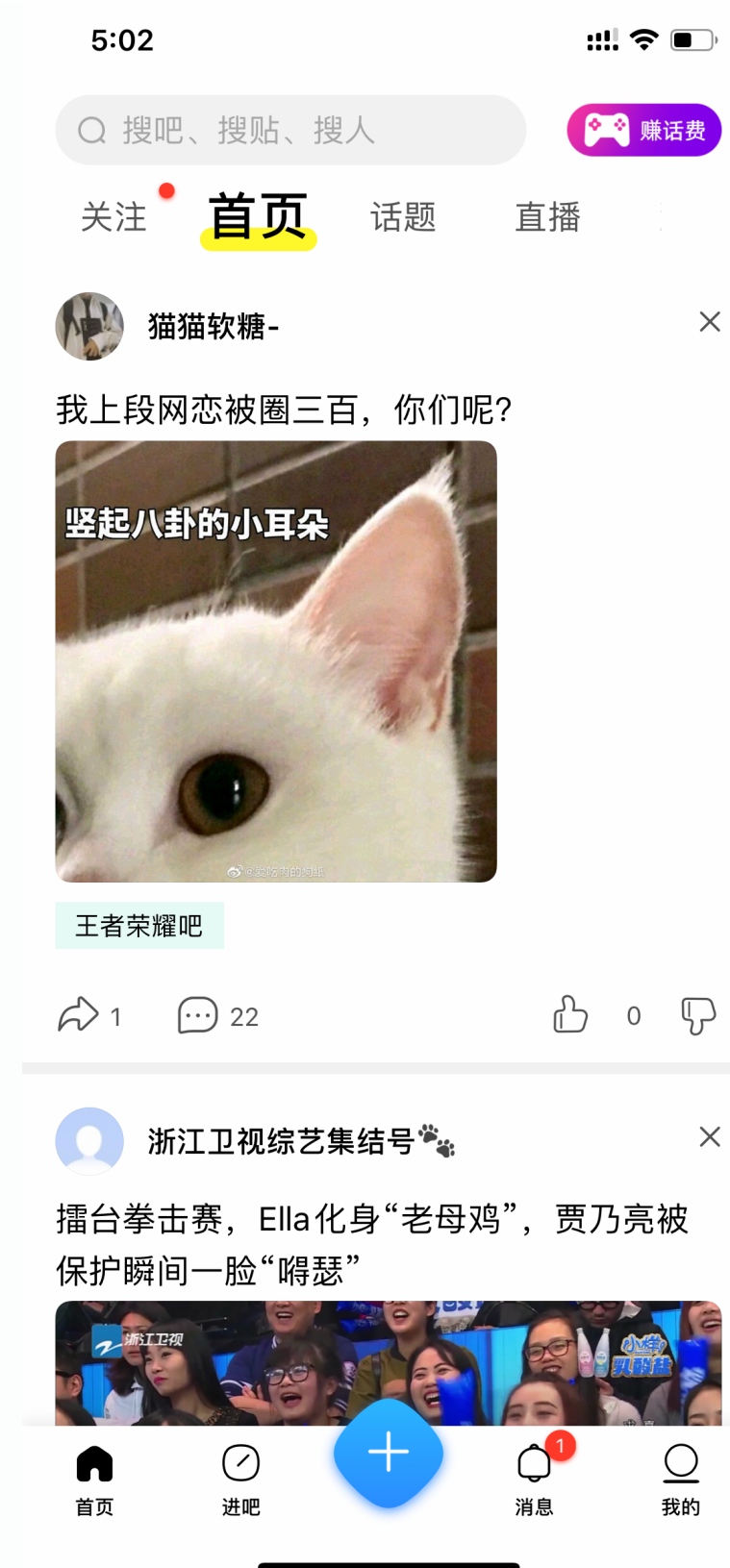
目录

- 为何引入Flutter
- 实践之旅
- 总结

贴吧



- 年轻人兴趣社区
- 2003年至今
- 拥有10亿注册用户
- 2000多万贴吧



业务方向

社区生态

- 用户增长
- 吧主
- 垂类
- 风控

内容运营

- Push
- 话题
- 神回复
- 个人动态

商业化

- 品牌大使
- 直播
- Banner
- 原生广告



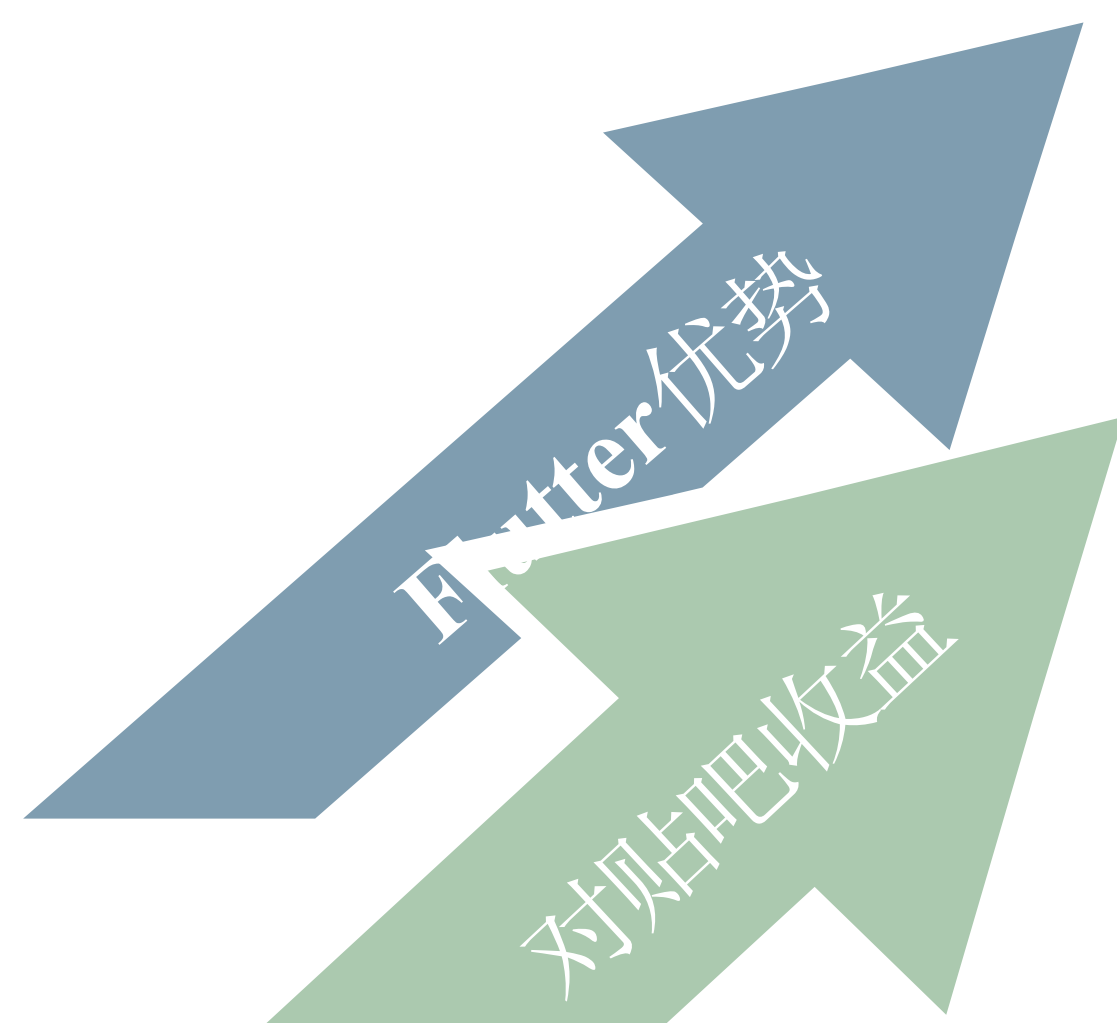
研发效率



用户体验

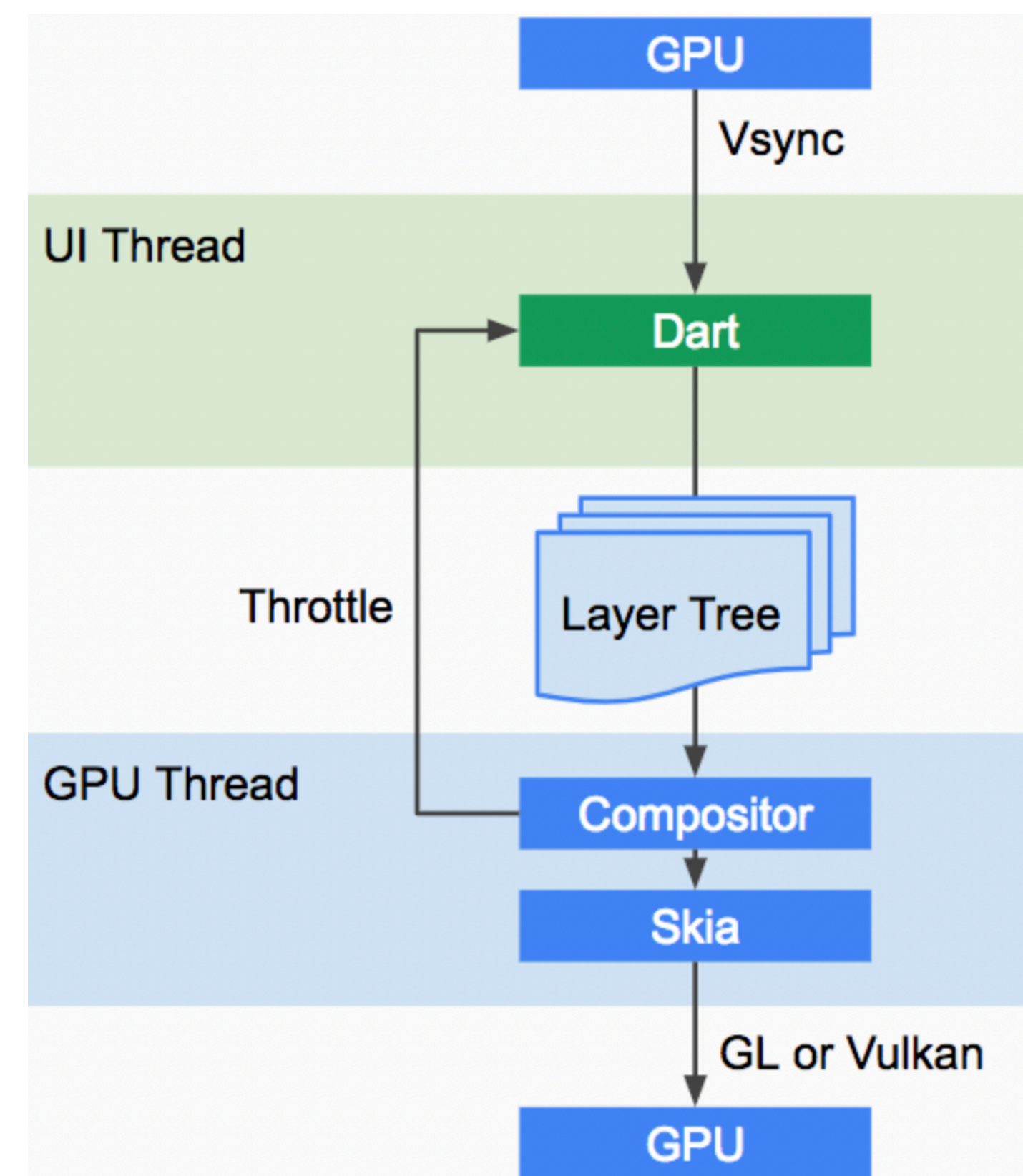
最快触达用户

Flutter的优势



- 1 跨平台开发
- 2 高效的开发体验
- 3 性能接近Native

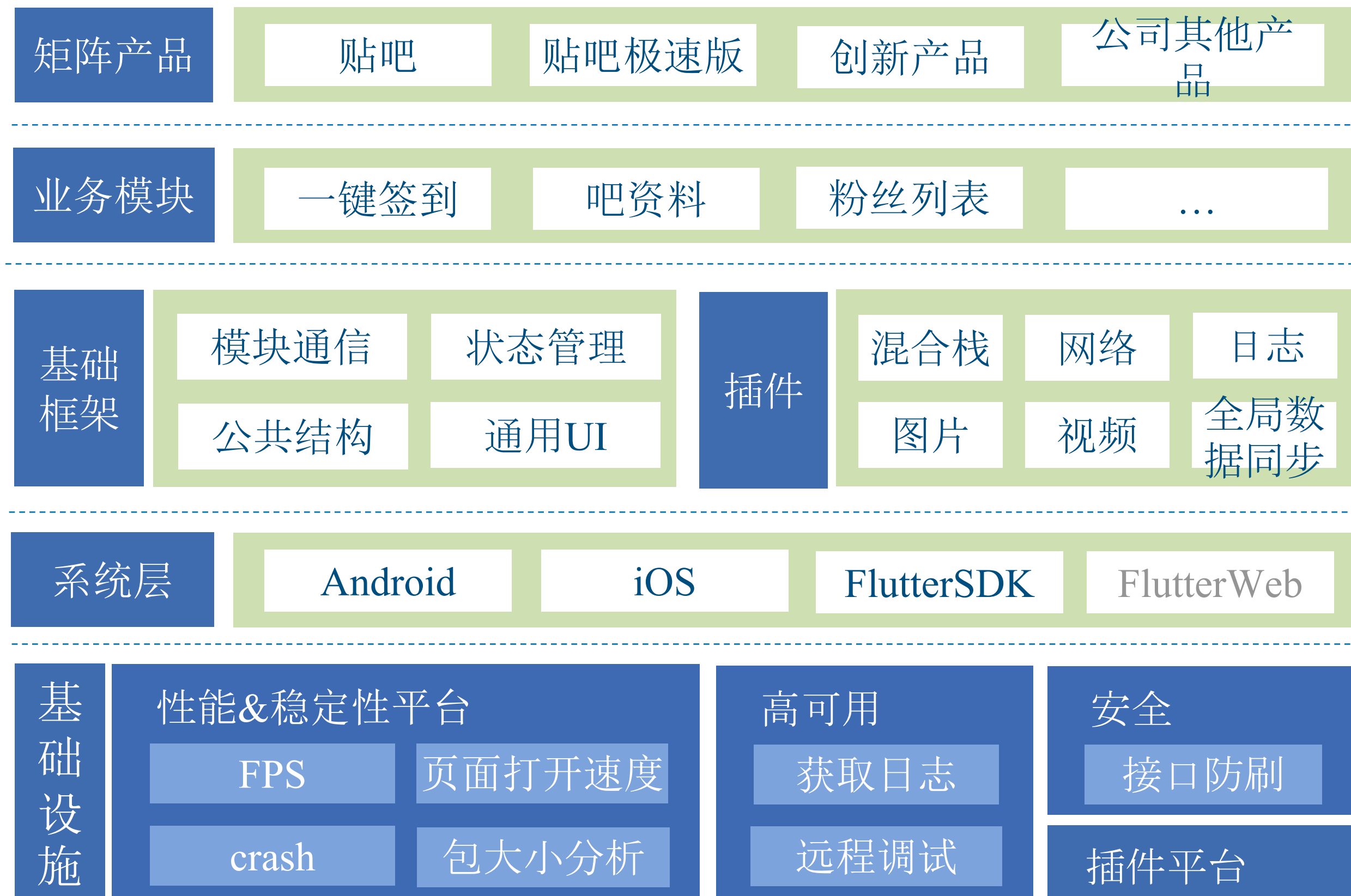
- N倍平台产出
 - 多端逻辑一致
- } 效率
质量



目录

- 为何引入Flutter
- 实践之旅
 - 技术全景
 - 工程体系接入
 - 业务框架升级
 - Android插件化
- 总结

先看一下全景



目录

- 为何引入Flutter
- 实践之旅
 - 技术全景
 - 工程体系接入
 - 业务框架升级
 - Android插件化
- 总结

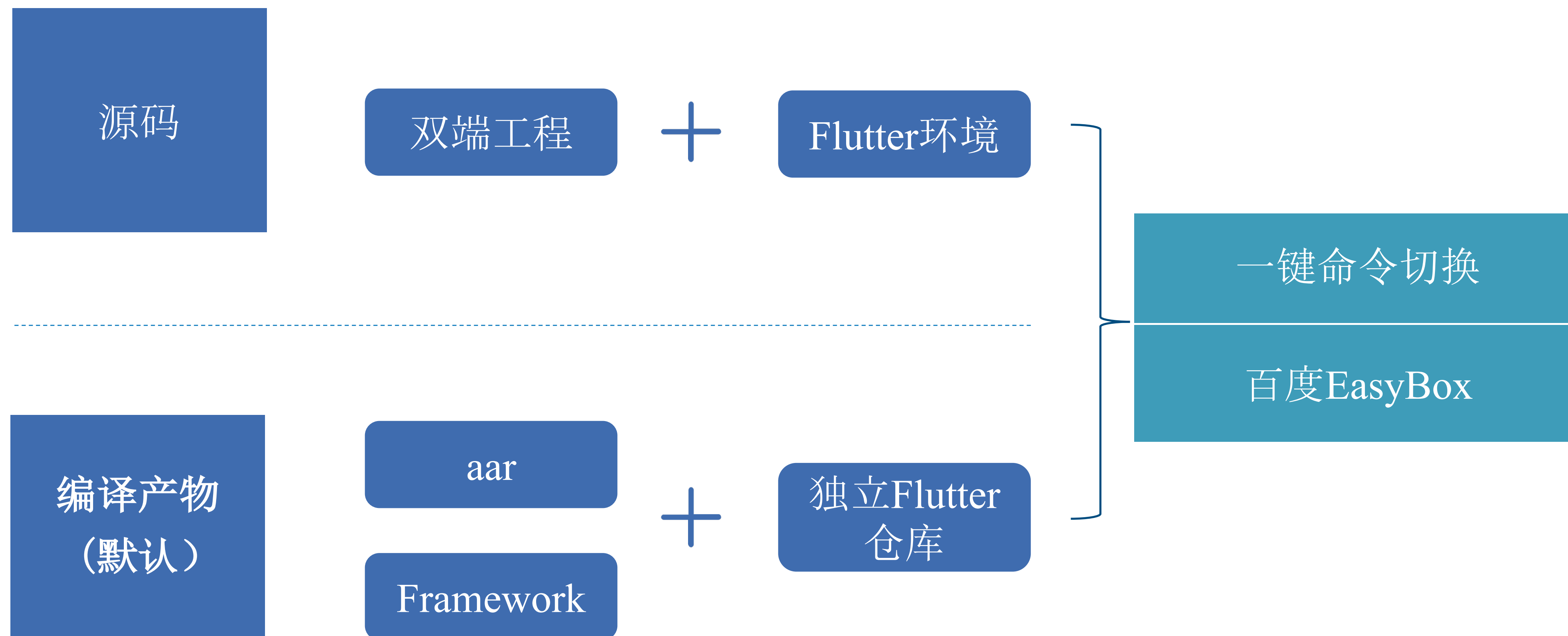
第一步



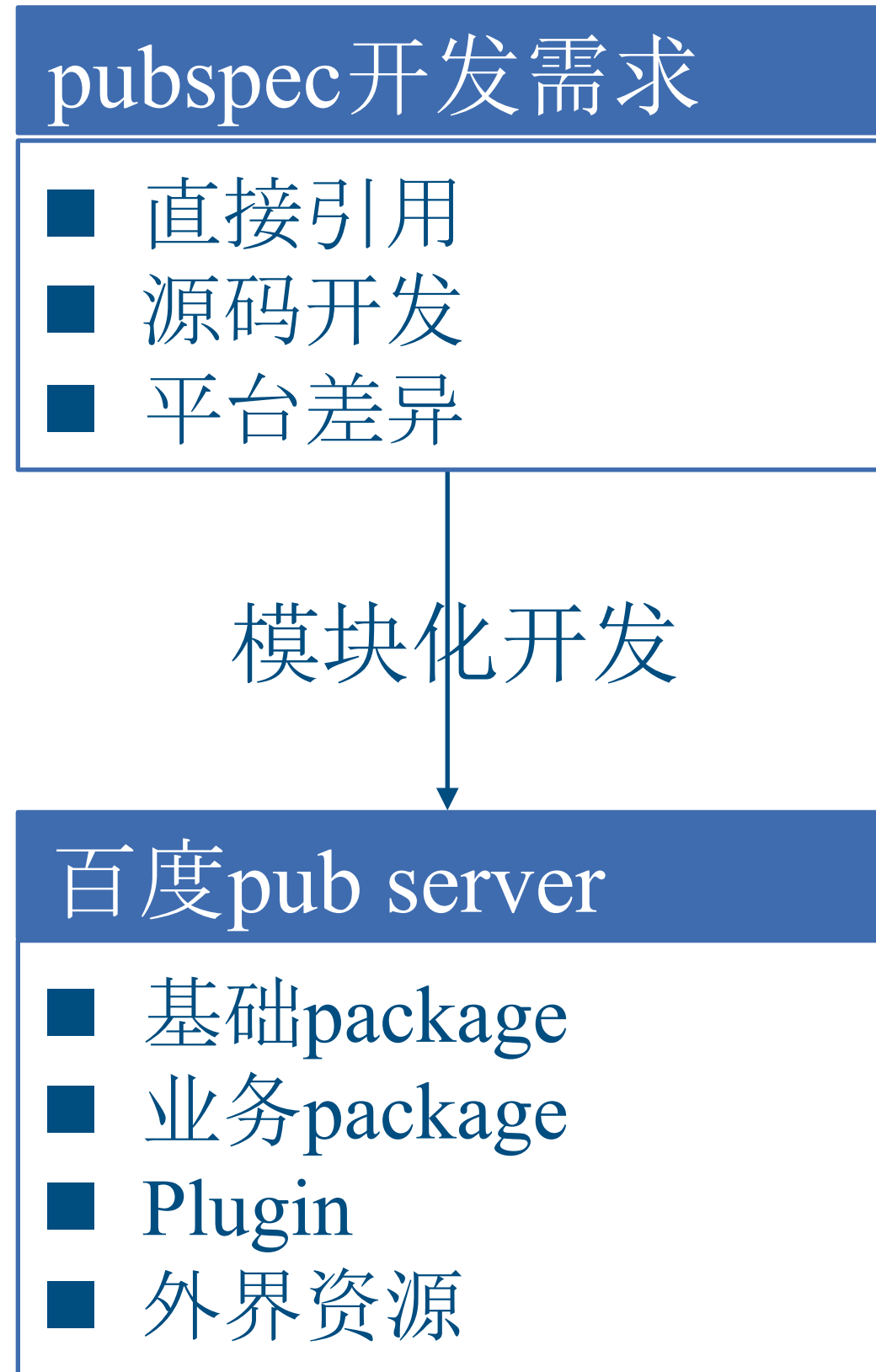
工程体系接入

- Flutter接入方式
- FlutterBox

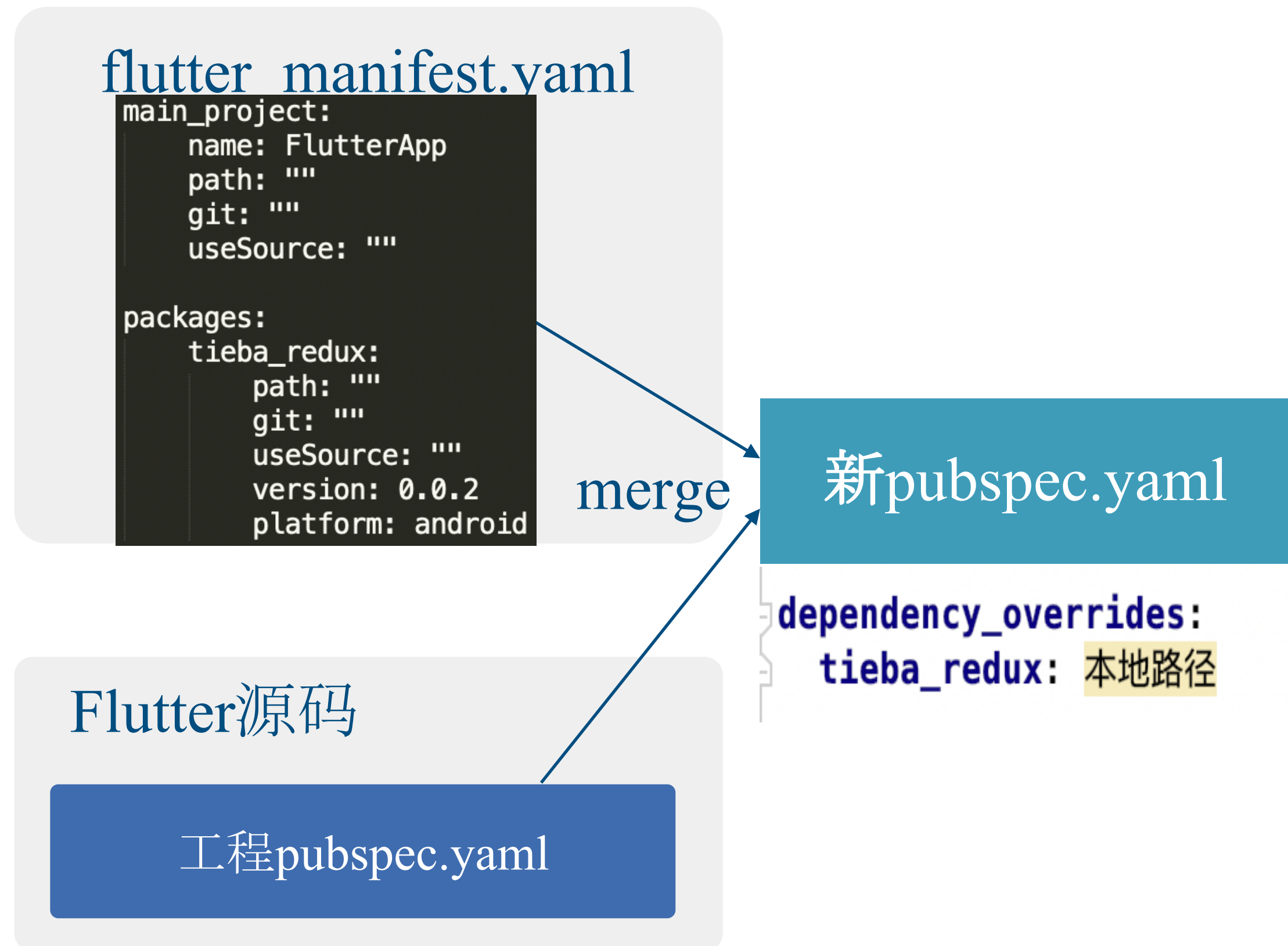
Flutter接入方式



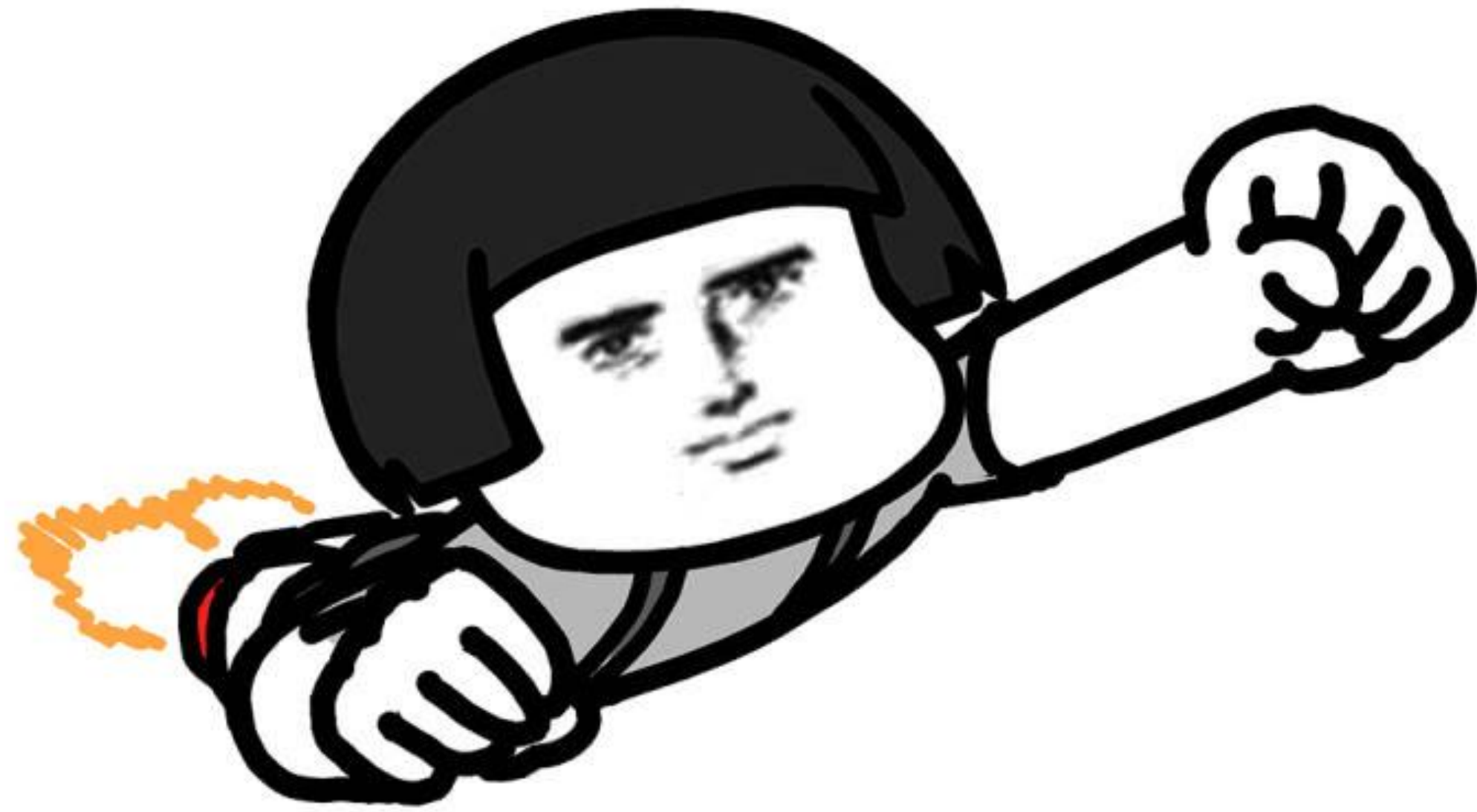
FlutterBox



源码接入
一键命令



业务框架升级之旅



接下来.....

- 技术需求（背景）
- 现有业务框架
- 现有业务框架的问题与挑战
- `redux`到`tieba_redux`

技术需求（背景）



定制版本、
快速开发新app



细粒度模块化

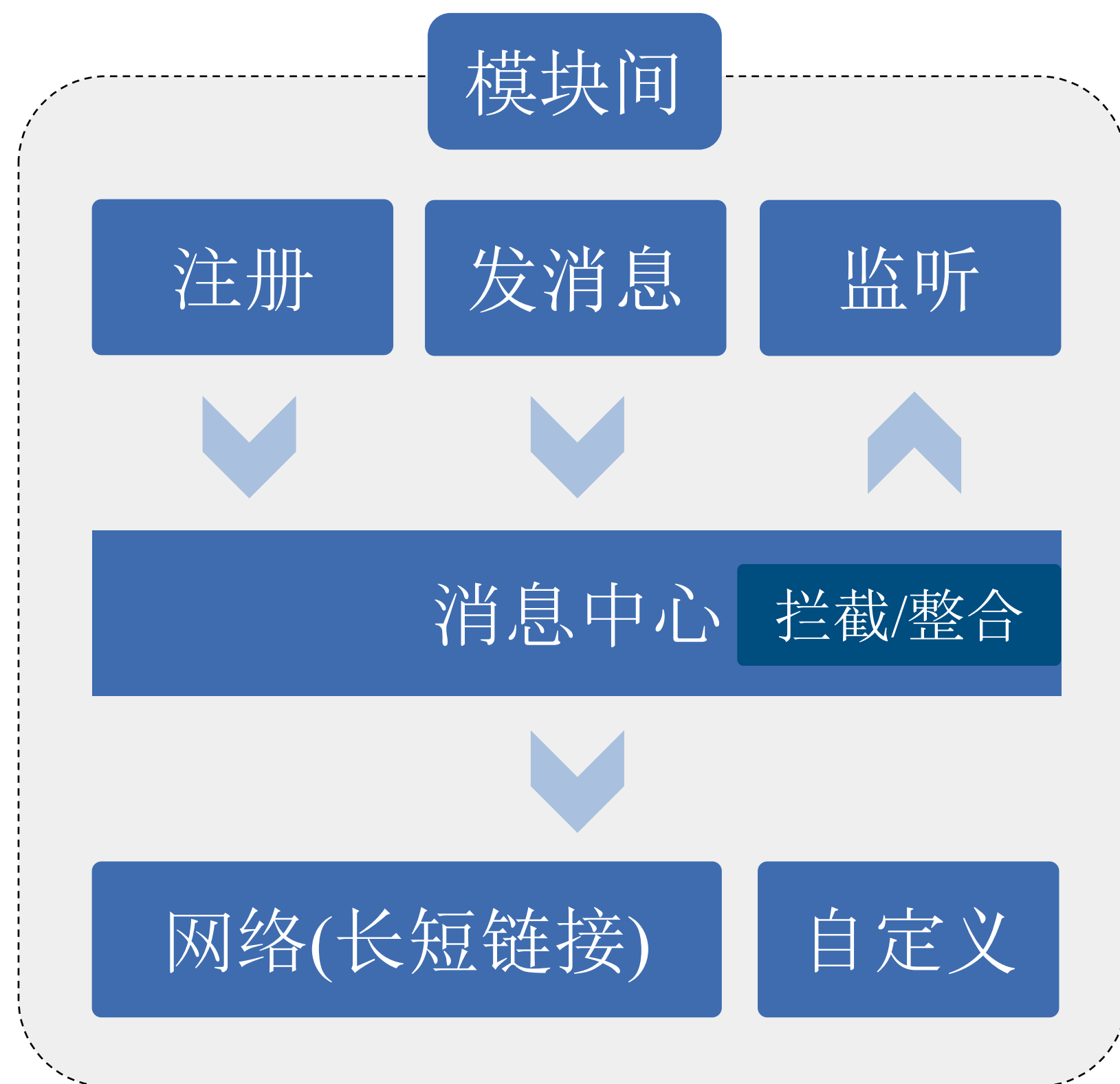


中台能力

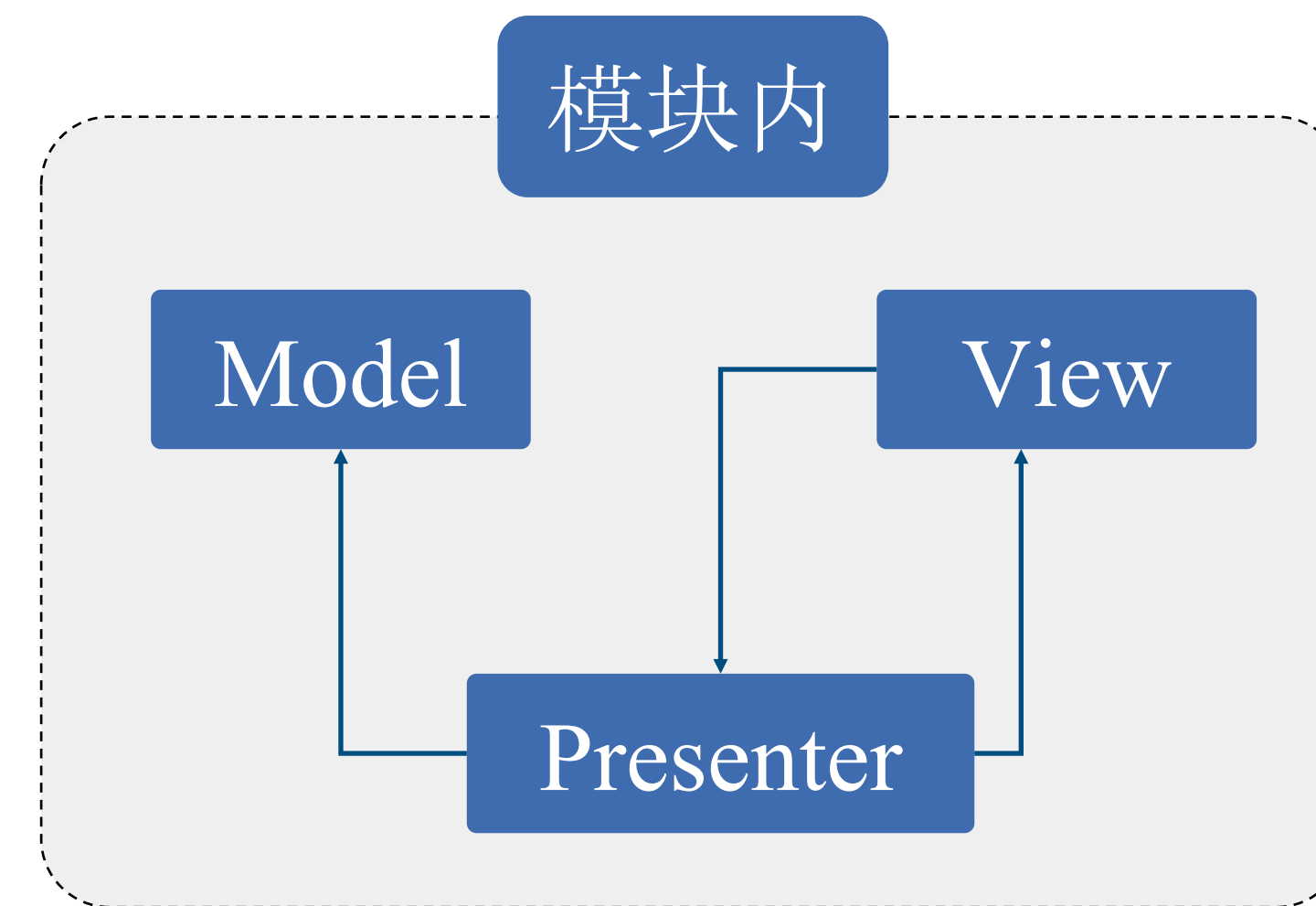


一键配置生成

现有业务框架（传统做法）



+



现有业务框架的问题与挑战



数据和UI一致性较差

数据本身的变化可控性较差

模块内和模块间代码风格不一致

现代

响应式

+

redux

tieba_redux

Redux的优势与问题

- 生态成熟，适用复杂场景
- 带有编码规范的框架
- 满足拦截类开发需求
 - 网络访问添加通用参数
 - 打印action序列和State，加速问题定位
 - 改变跳转，热修复

组织代码

- 组件:函数式, class
- 数据:props, state
- 纯函数

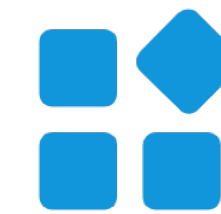
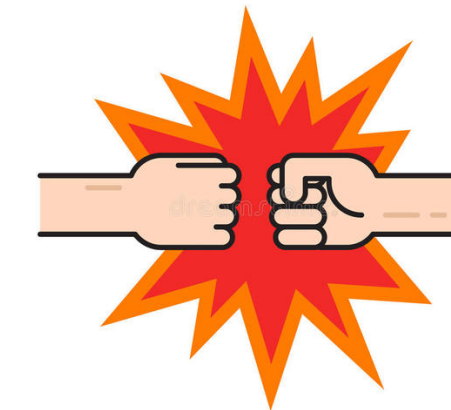


组织思想

- State变化可预期
- 面向action序列编程
- 松耦合调用方式



Redux

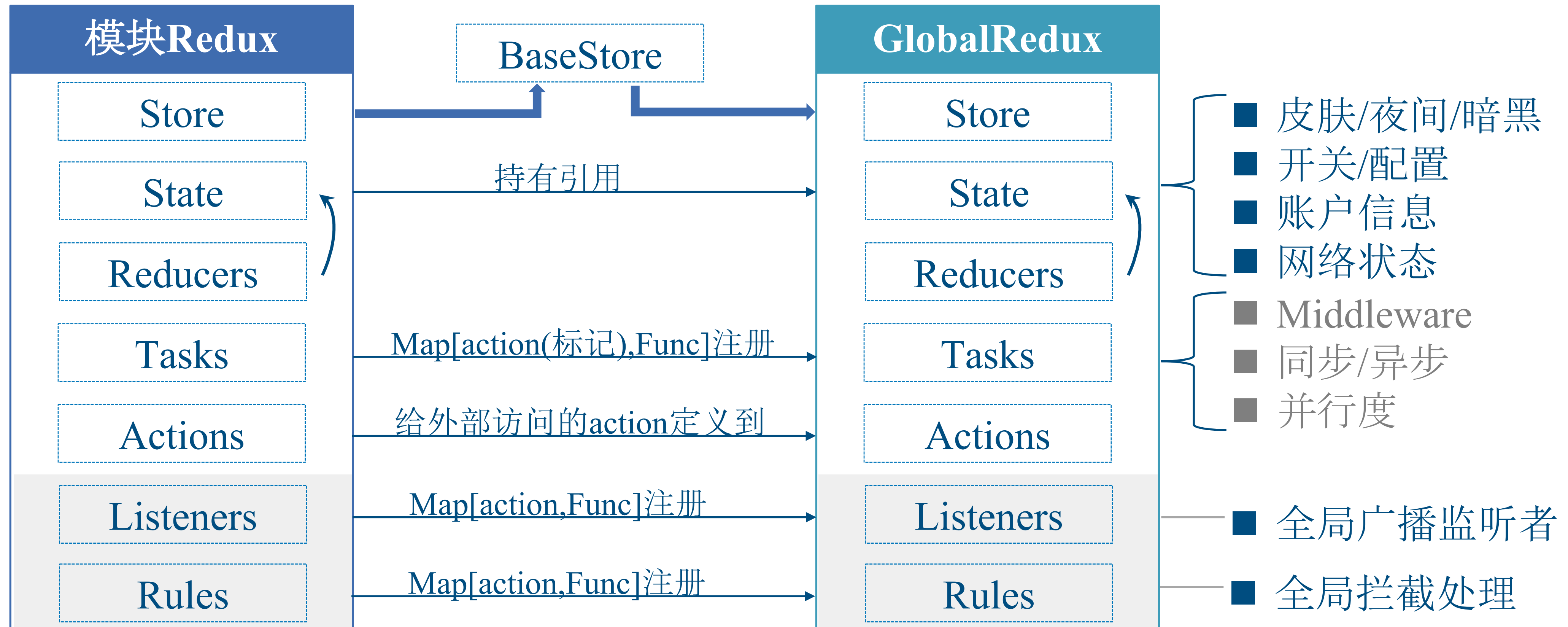


模块化

- 概念上的『和』与『分』
- 分之后的代码组织与运行
- 细粒度模块化的编码需求

tieba_redux

tieba_redux



Task、Listener、Rule

	1个Action对应1个/多个	模块内/全局	静态/动态注册	使用场景
Task	1个	模块内	动态	Redux标准用法如发起网络
		全局	静态	打开页面；获取定位
Listener	多个	全局	动态	未读消息数变化
			静态	App启动；网络变化
Rule	多个	全局	静态	拦截页面跳转； 添加通用参数；打日志

SHOW CODE

```
forum_detail
├── build.yaml
├── lib
│   ├── widgets
│   └── redux
│       ├── state.dart
│       ├── listener.dart
│       ├── action.dart
│       ├── store.dart
│       ├── rule.dart
│       ├── task.dart
│       └── reducer.dart
│   ├── pages
│   ├── assets
│   ├── data
│   │   └── forum_detail.dart
│   ├── json
│   └── config
└──
```

```
import 'package:base/base.dart';
import 'package:forum_detail/redux/listener.dart';
import 'package:forum_detail/redux/rule.dart';
import 'package:redux/redux.dart';
import 'package:common/common.dart';

import 'state.dart';
import 'reducer.dart';
import 'task.dart';

class ForumDetailStore extends BaseStore<ForumDetailState> {
  ForumDetailStore(
    ForumDetailTask task,
    ForumDetailListener listener,
    ForumDetailRule rule,
    Reducer<ForumDetailState> reducer,
    {ForumDetailState initialState, bool syncStream, bool distinct, Map initParams})
    : assert(reducer != null),
      super(BdUniqueId.sharedInstance.getId(), task, listener, rule, reducer,
        initialState: initialState ?? ForumDetailState(),
        syncStream: syncStream ?? false,
        distinct: distinct ?? false,
      ) {
    state.forumId = initParams['_forumId'];
  }

  factory ForumDetailStore.withParams([Map initParams]) => ForumDetailStore(
    ForumDetailTask(forumDetailStorePrivateTaskMap),
    ForumDetailListener(forumDetailStoreListenerList),
    ForumDetailRule(),
    forumDetailReducers,
    initParams: initParams);
}
```

根据pubspec文件修改
触发pubget动态生成

Main函数调用注册函数

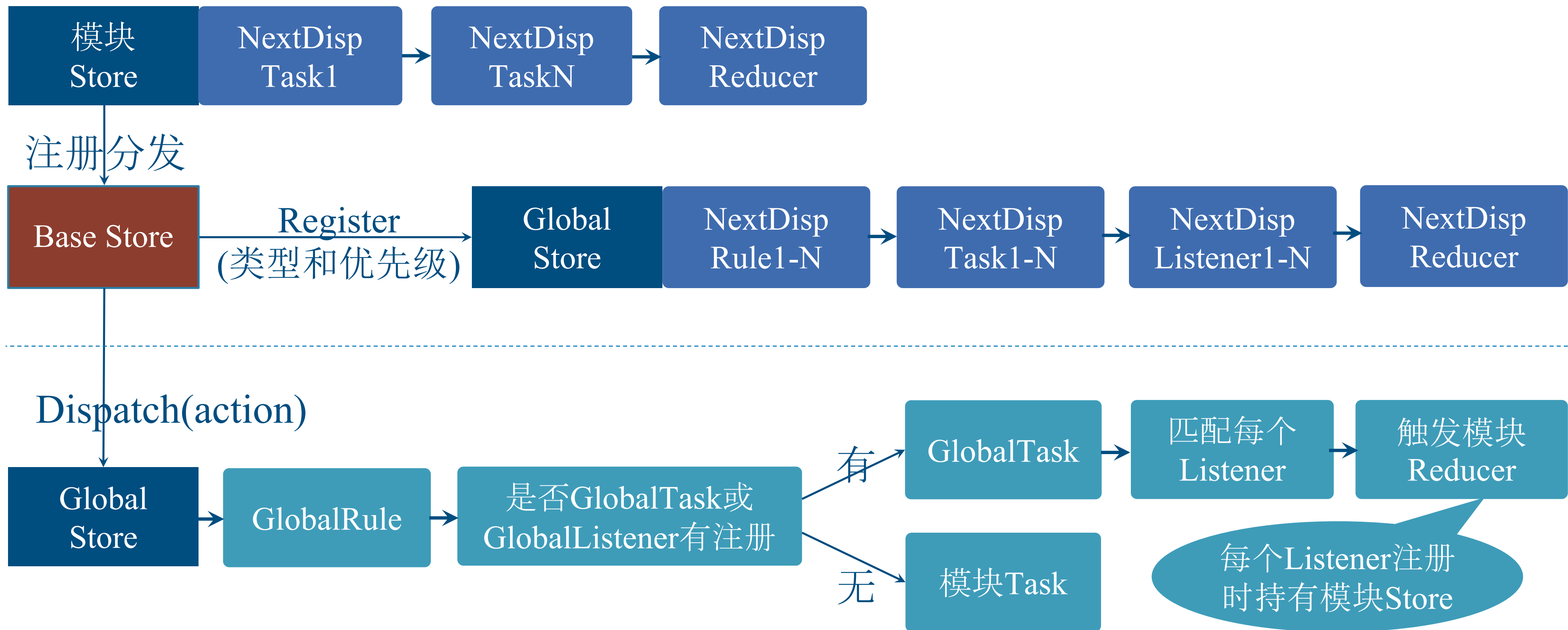
```
// GENERATED CODE - DO NOT MODIFY BY HAND

import 'package:common/common.dart';
import 'package:demo_module/demo_module.dart';
// import 各个有需要提前初始化注册的模块

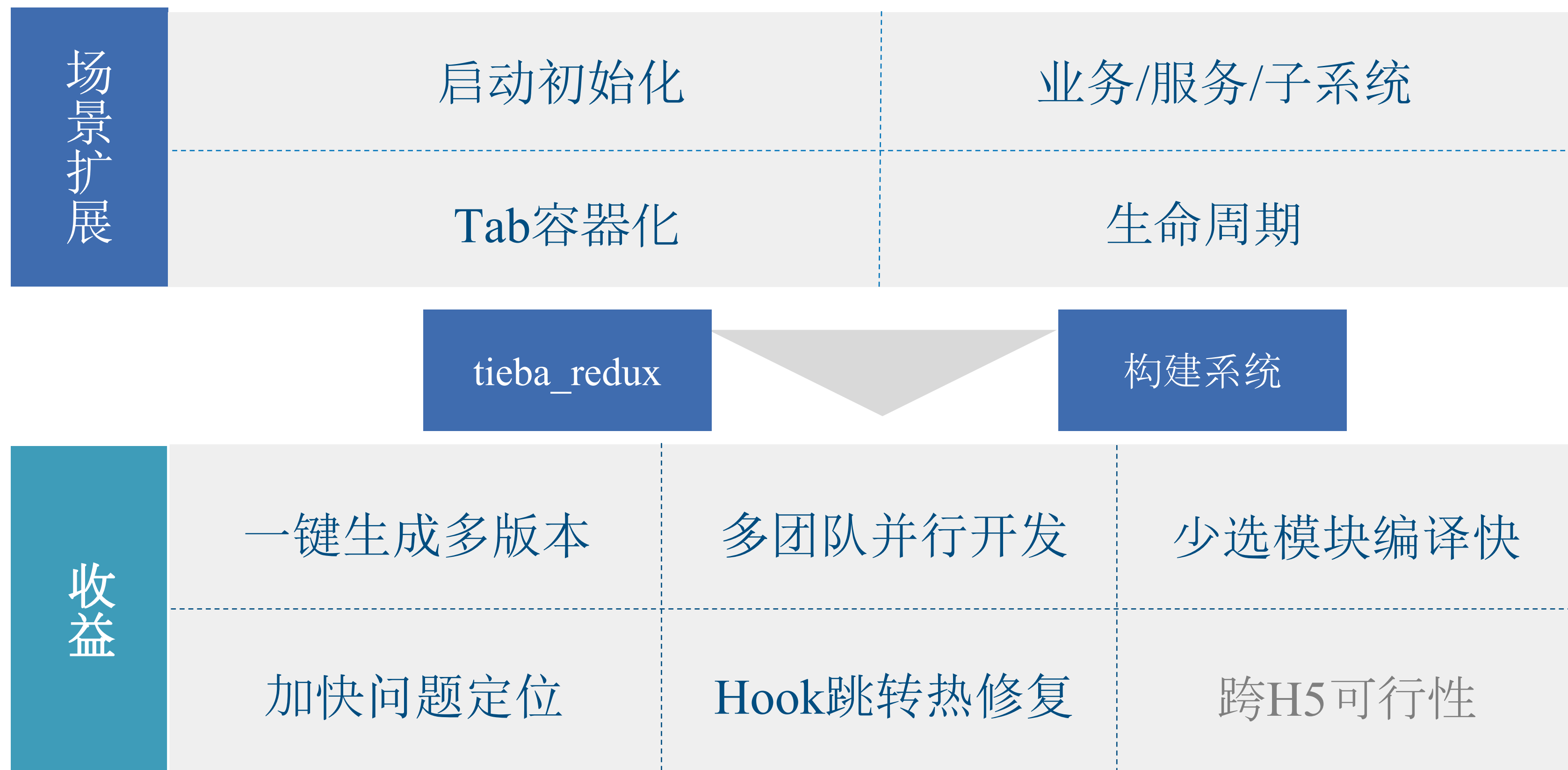
class Register {
  static void register() {
    GlobalStore store = GlobalStore.sharedInstance;

    store.registerTasks(testModuleStaticTaskMap);
    store.registerListeners(testModuleStaticListenerList);
    store.registerRules(testModuleRuleList);
  }
}
```

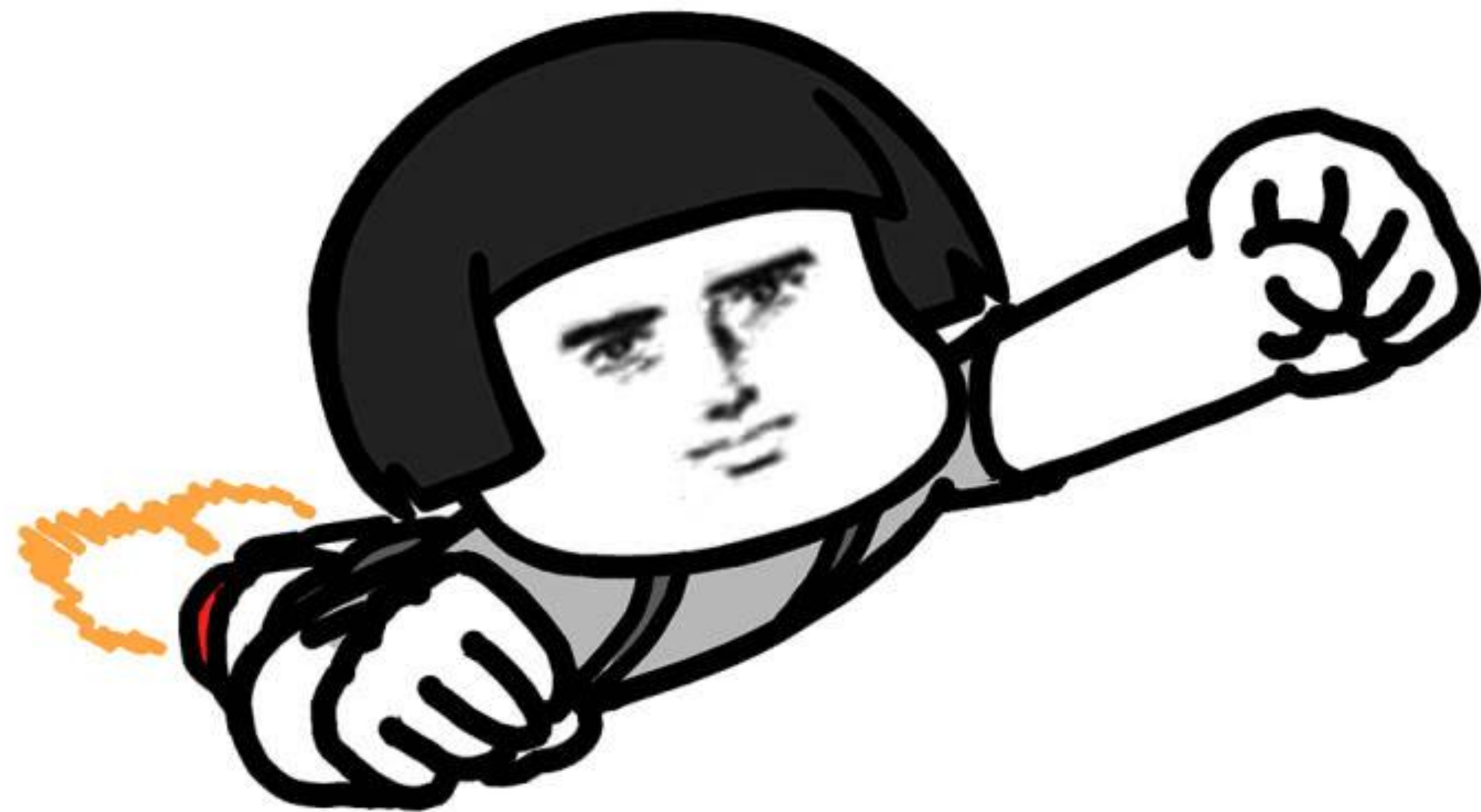
注册和分发执行



业务框架小结



Flutter Android插件之旅



接下来.....

Flutter Android插件

- 插件设计与目标
- 问题与挑战

插件设计与目标

特性

- 模块配置
- 透明开发

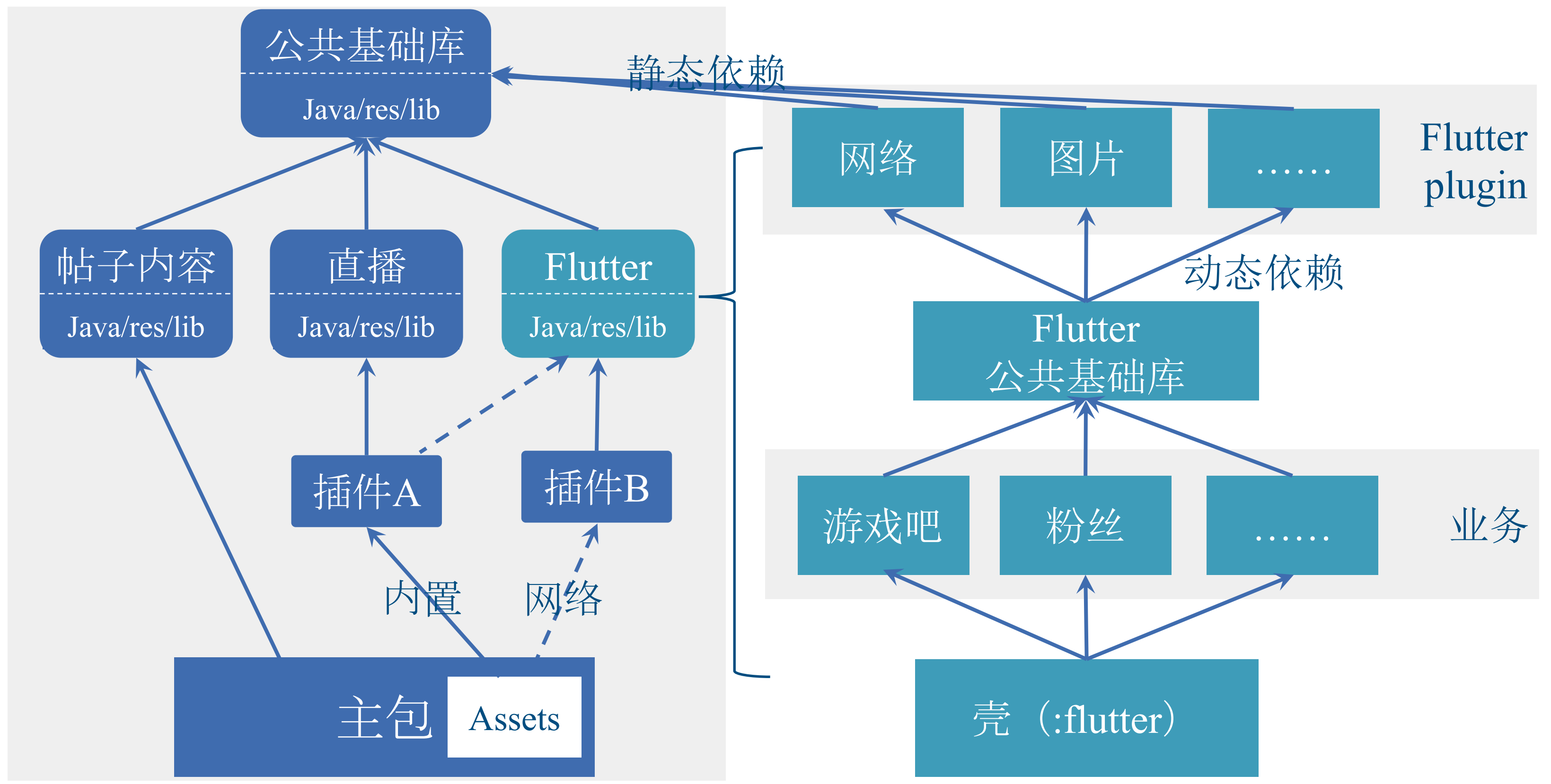
形态

- 内置
- 网络

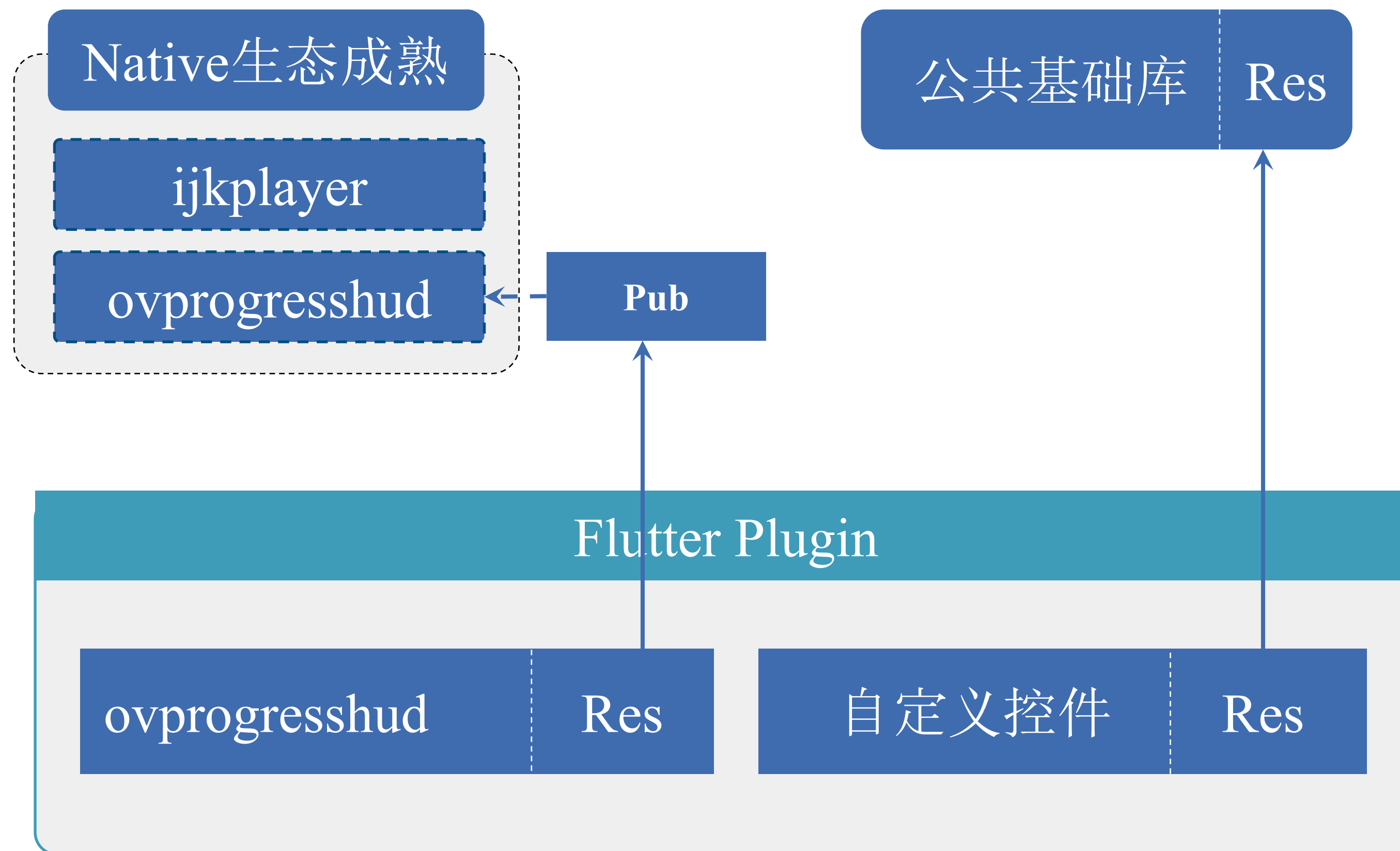
Runtime

- 运行容器
- 部署管理

构建平台



插件问题与挑战 - plugin资源打包



透明灵活

方案1：
代理模式
独立容器

方案2：✓
定制AAPT
资源分段
查找容器
运行注入

插件问题与挑战 – AndroidView资源使用

Build#PlatformViewsChannel::create

FlutterView::createSurfaceTexture

获取textureId, SurfaceTexture

#FlutterJNI.registerTexture

#PlatformViewAndroid::Register

通过GpuRunner到mapping[textureId, AndroidExternalTextureGL]

CompositorContext::Raster#Paint

TextureLayer::Paint(textureId)

#Texture::Paint

Attach, UpdateTexImage,
drawImage

VirtualDisplayController

#SingleViewPresentation(context:FlutterActivity)

#Presentation构造函数super(createPresentationContext)

```
private static Context createPresentationContext(
    Context outerContext, Display display, int theme) {
    if (outerContext == null) {
        throw new IllegalArgumentException("outerContext must not be null");
    }
    if (display == null) {
        throw new IllegalArgumentException("display must not be null");
    }
}
```

```
Context displayContext = outerContext.createDisplayContext(display);
```

```
@Override
public Context createDisplayContext(Display display) {
    if (display == null) {
        throw new IllegalArgumentException("display must not be null");
    }

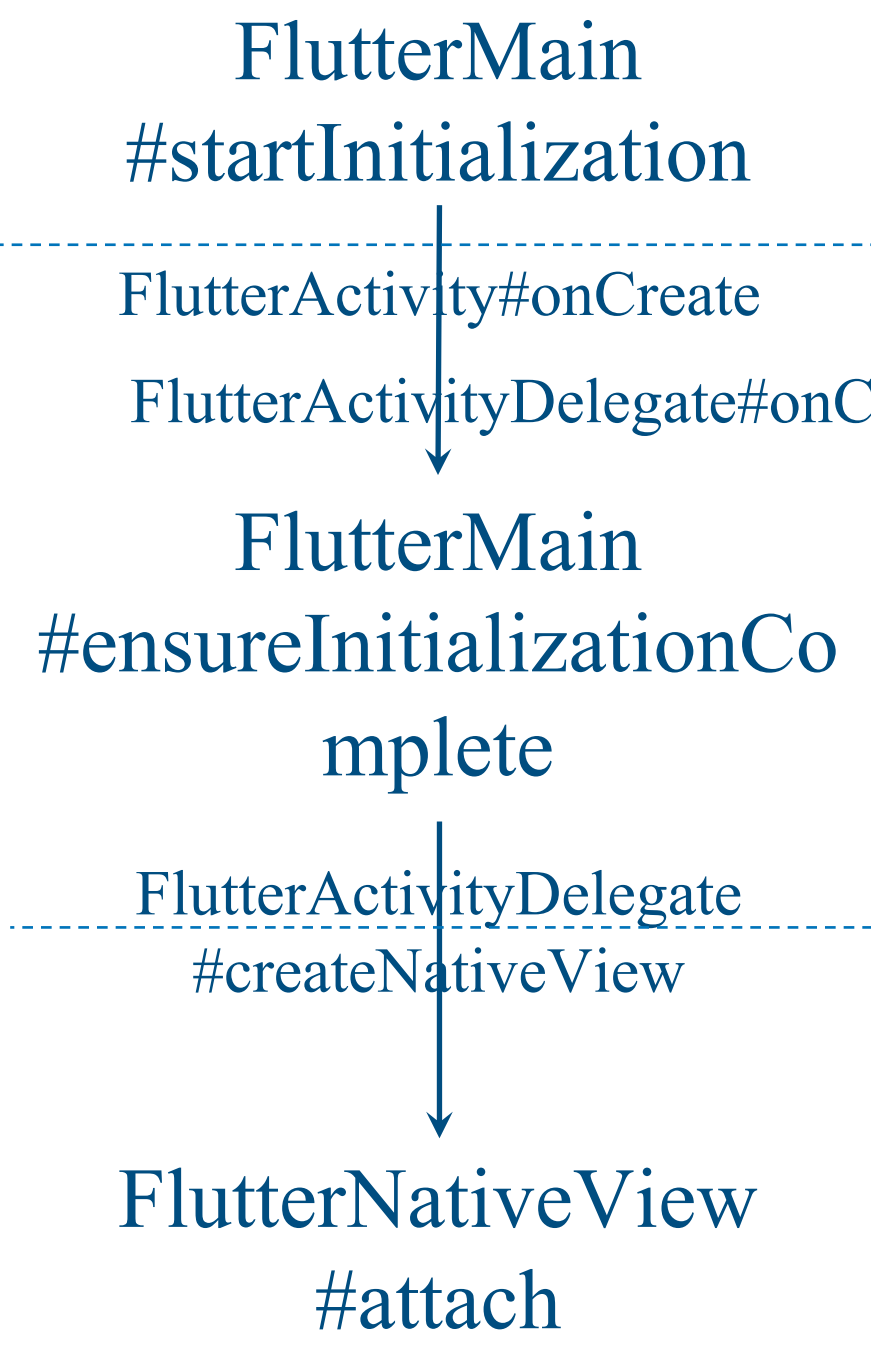
    ContextImpl context = new ContextImpl(this, mMainThread, mPackageInfo, mSplitName,
        mActivityToken, mUser, mFlags, mClassLoader);

    final int displayId = display.getDisplayId();
    context.setResources(createResources(mActivityToken, mPackageInfo, mSplitName, displayId,
        null, getDisplayAdjustments(displayId).getCompatibilityInfo()));
    context.mDisplay = display;
    return context;
}
```

反射调用setResource

插件问题与挑战 – 加载libapp.so

初始化



initConfig

```
#sAotSharedLibraryName = metadata.getString("aot-shared-library-name", "libapp.so");
```

反射修改路径

```
if (BuildConfig.DEBUG) {  
  String snapshotAssetPath = PathUtils.getDataDirectory(applicationContext) + File.separator + "snapshot";  
  kernelPath = snapshotAssetPath + File.separator + DEFAULT_KERNEL_BLOB;  
  shellArgs.add("--" + SNAPSHOT_ASSET_PATH_KEY + "=" + snapshotAssetPath);  
  shellArgs.add("--" + VM_SNAPSHOT_DATA_KEY + "=" + sVmSnapshotData);  
  shellArgs.add("--" + ISOLATE_SNAPSHOT_DATA_KEY + "=" + sIsolateSnapshotData);  
} else {  
  shellArgs.add("--" + AOT_SHARED_LIBRARY_NAME + "=" + sAotSharedLibraryName);  
}
```

FlutterJNI.nativeInit #SettingsFromCommandLine

```
if (aot_shared_library_name.size() > 0) {  
  settings.application_library_path = aot_shared_library_name;  
} else if (snapshot_asset_path.size() > 0) {  
  settings.vm_snapshot_data_path =  
    fml::paths::JoinPaths({snapshot_asset_path, vm_snapshot_data_filename});  
  settings.vm_snapshot_instr_path = fml::paths::JoinPaths(  
    {snapshot_asset_path, vm_snapshot_instr_filename});  
  settings.isolate_snapshot_data_path = fml::paths::JoinPaths(  
    {snapshot_asset_path, isolate_snapshot_data_filename});  
  settings.isolate_snapshot_instr_path = fml::paths::JoinPaths(  
    {snapshot_asset_path, isolate_snapshot_instr_filename});  
}
```

创建shell#创建engine之前, 创建DartVM#创建DartVMData
#DartSnapshot::VMSnapshotFromSettings和
DartSnapshot::IsolateSnapshotFromSettings
#NativeLibrary::Create(native_library_path.c_str());
#handle_ = ::dlopen(path, RTLD_NOW);

目录

- 为何引入Flutter
- 实践之旅
- 总结

主要分享了

工程体系接入

- 通过丰富脚本工具
满足复杂工程需求

技术全景

- 分层和工具链

总结

业务框架升级

- 传统到现代
- tieba_redux

Flutter Android插件

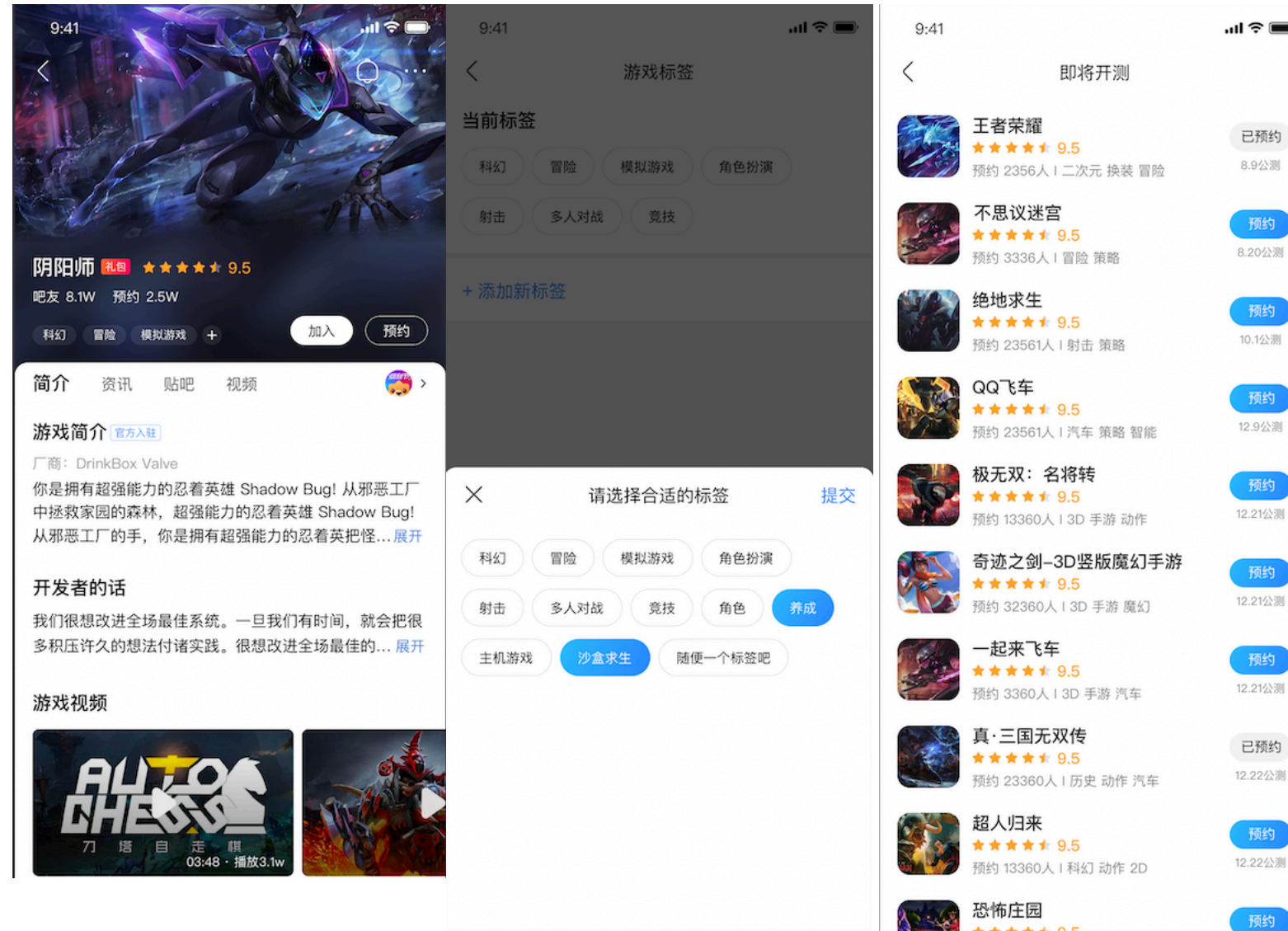
- 资源和libapp加载

落地情况

- 主版本：一键签到，吧详情，粉丝列表等
- 内部版（纯flutter版）：
Android, iOS, iPad



- 正在进行：个人中心、垂类吧



展望一下



有Flutter要上；
没有Flutter，创造Flutter也要上



扫码查看

60 天学透算法与数据结构

600+ 名企内推通道向你打开!

✓ 大厂内推 ✓ 实战演练 ✓ 闭环体系 ✓ 社群连接



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