

Flutter 技术分享

刘丰恺（好好学习）

个人简介

刘丰恺

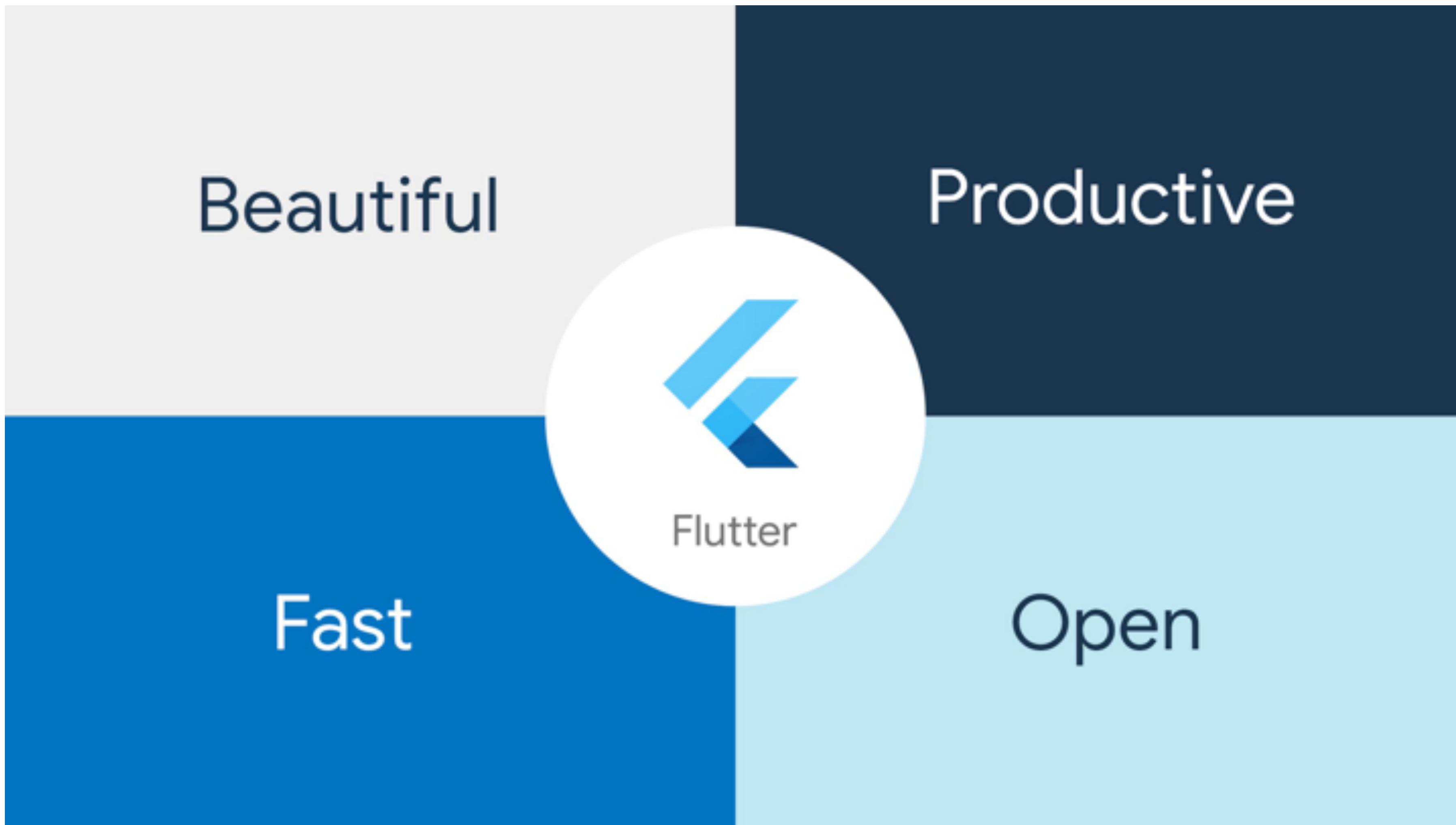
- 头条研发 好好学习客户端 致力于 Flutter 相关探索、场景化开发
- 曾就职阿里、dashbase，从事后端 infra 相关框架开发
- GitHub: <https://github.com/lfkdsk>

目录

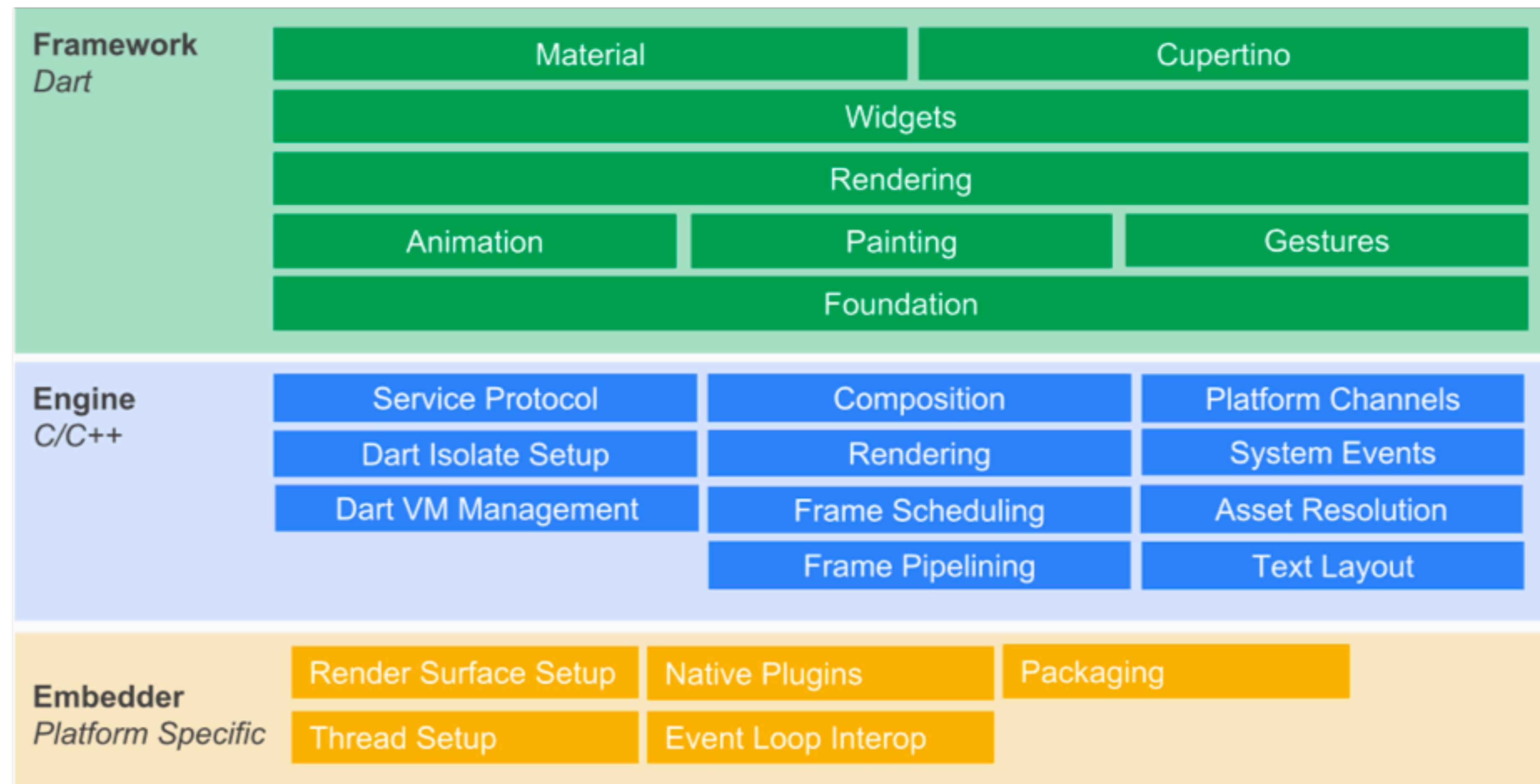
- Flutter 框架介绍
- 跨平台方案对比
- 好好学习 Flutter 解决方案进展

Flutter 框架介绍





Flutter 引擎结构





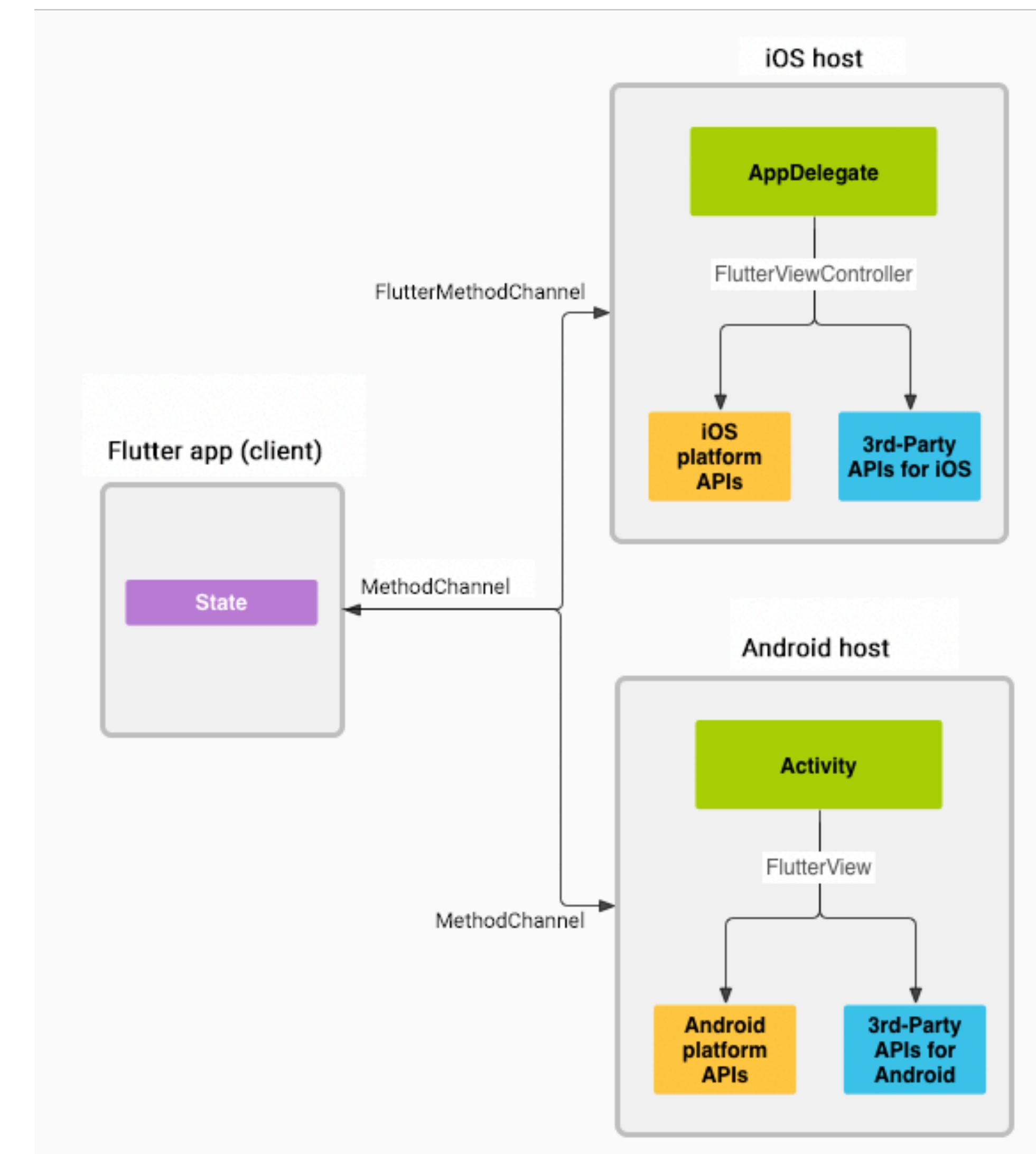
AOT JIT 多模式支持



| | Android | iOS |
|---------|---------|-------|
| Debug | 22.28M | 31M |
| Release | 3.46M | 13.8M |

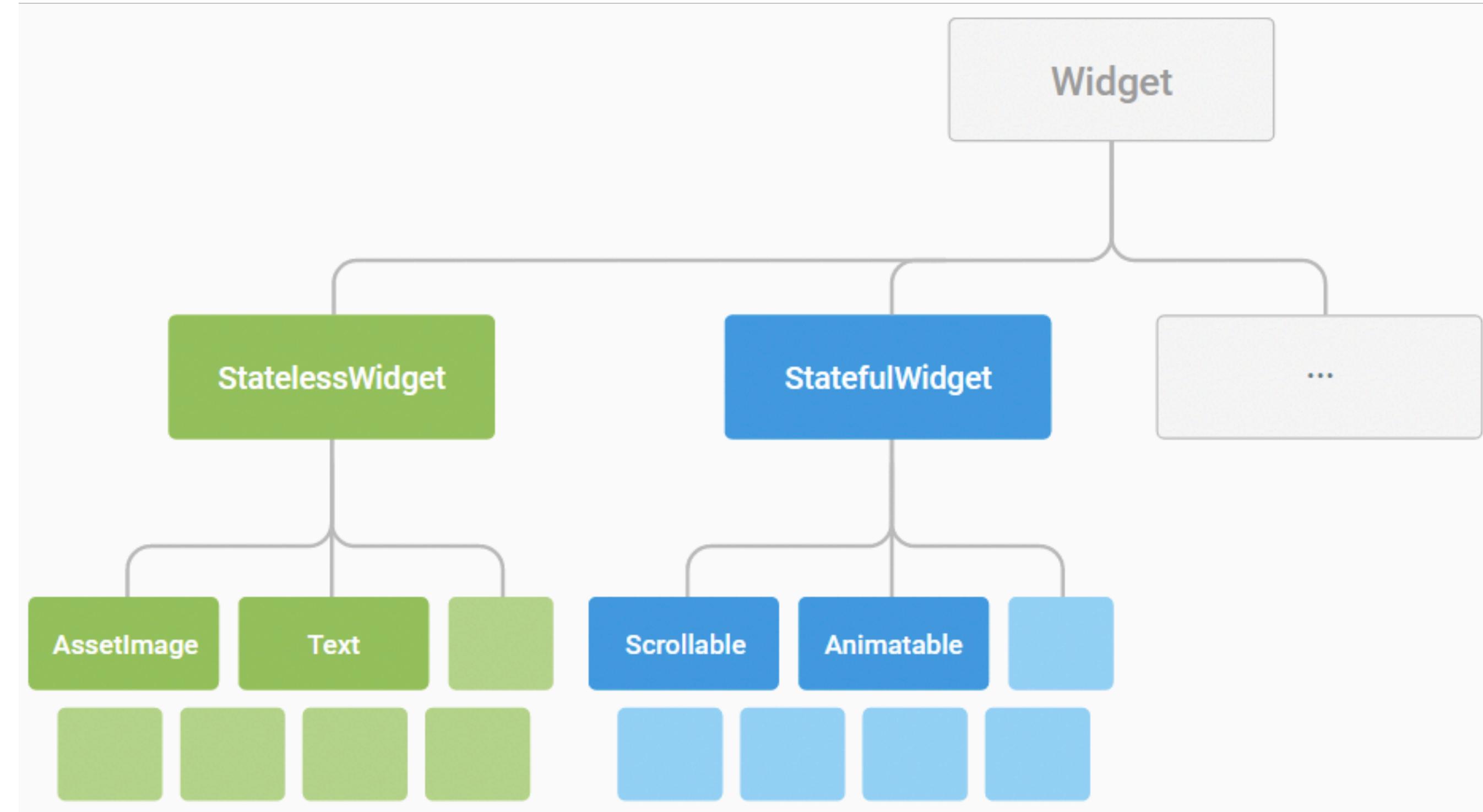
与 Native 沟通

Platform Channel

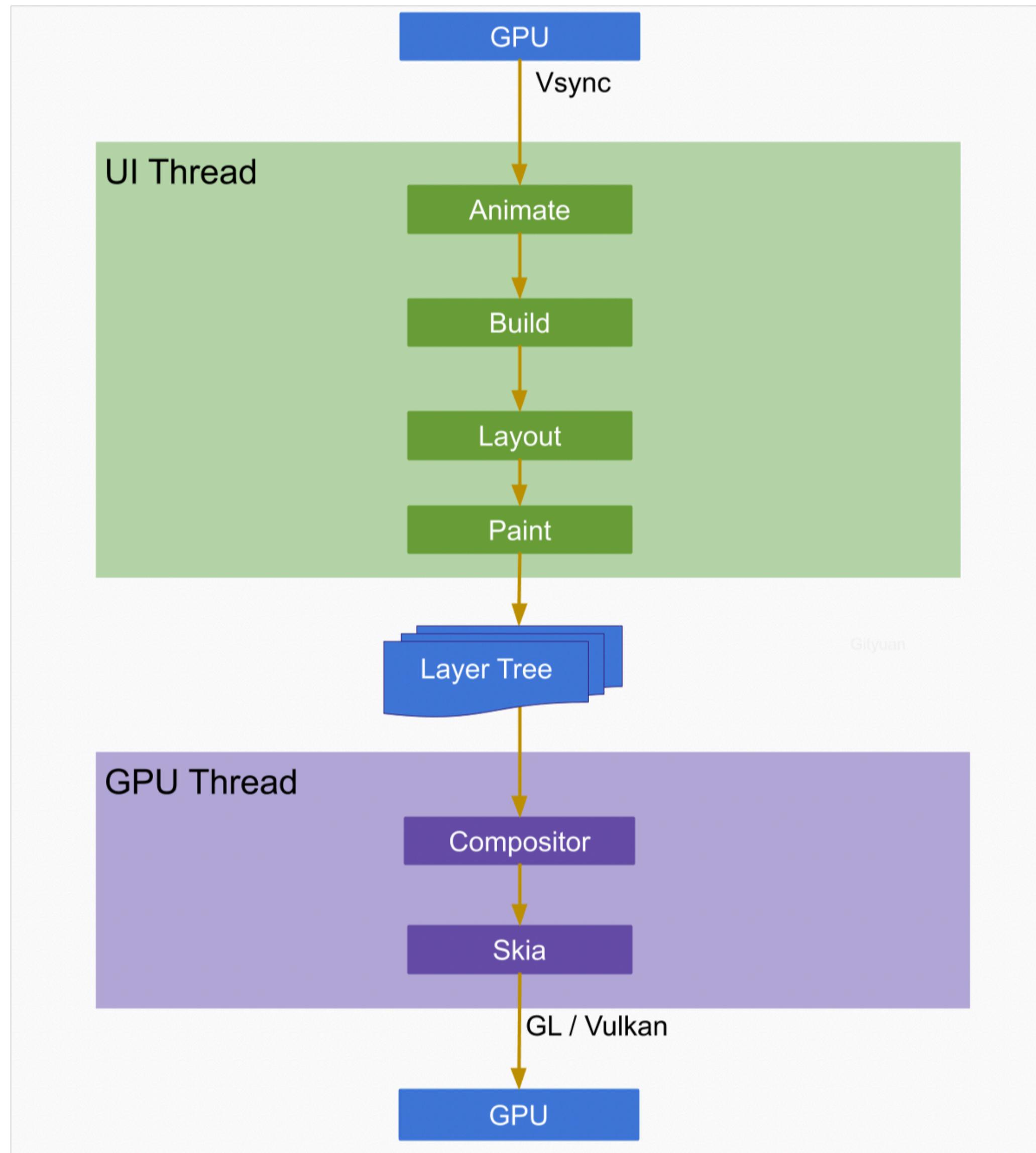


渲染

万物皆 Widget



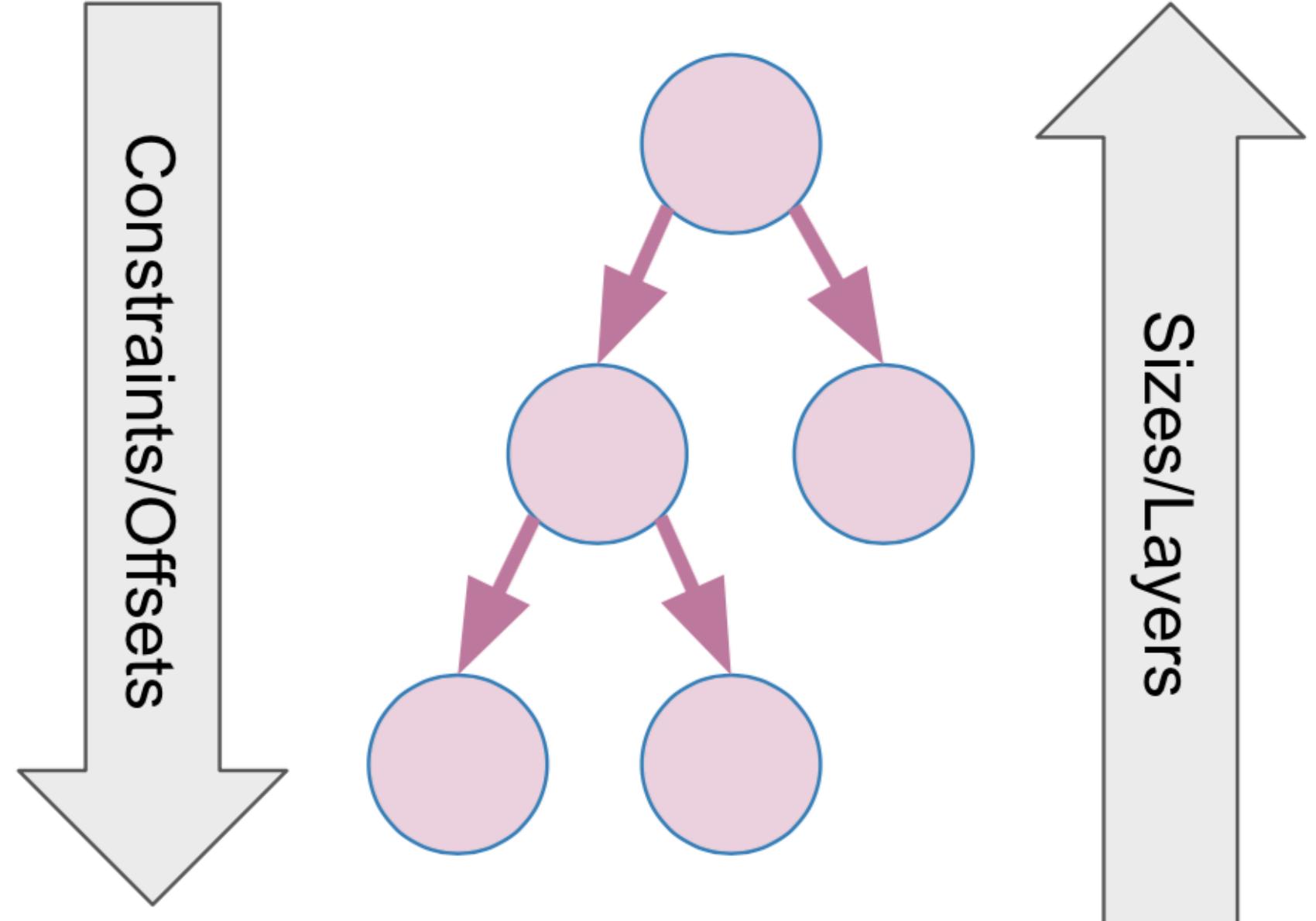
渲染 渲染管线



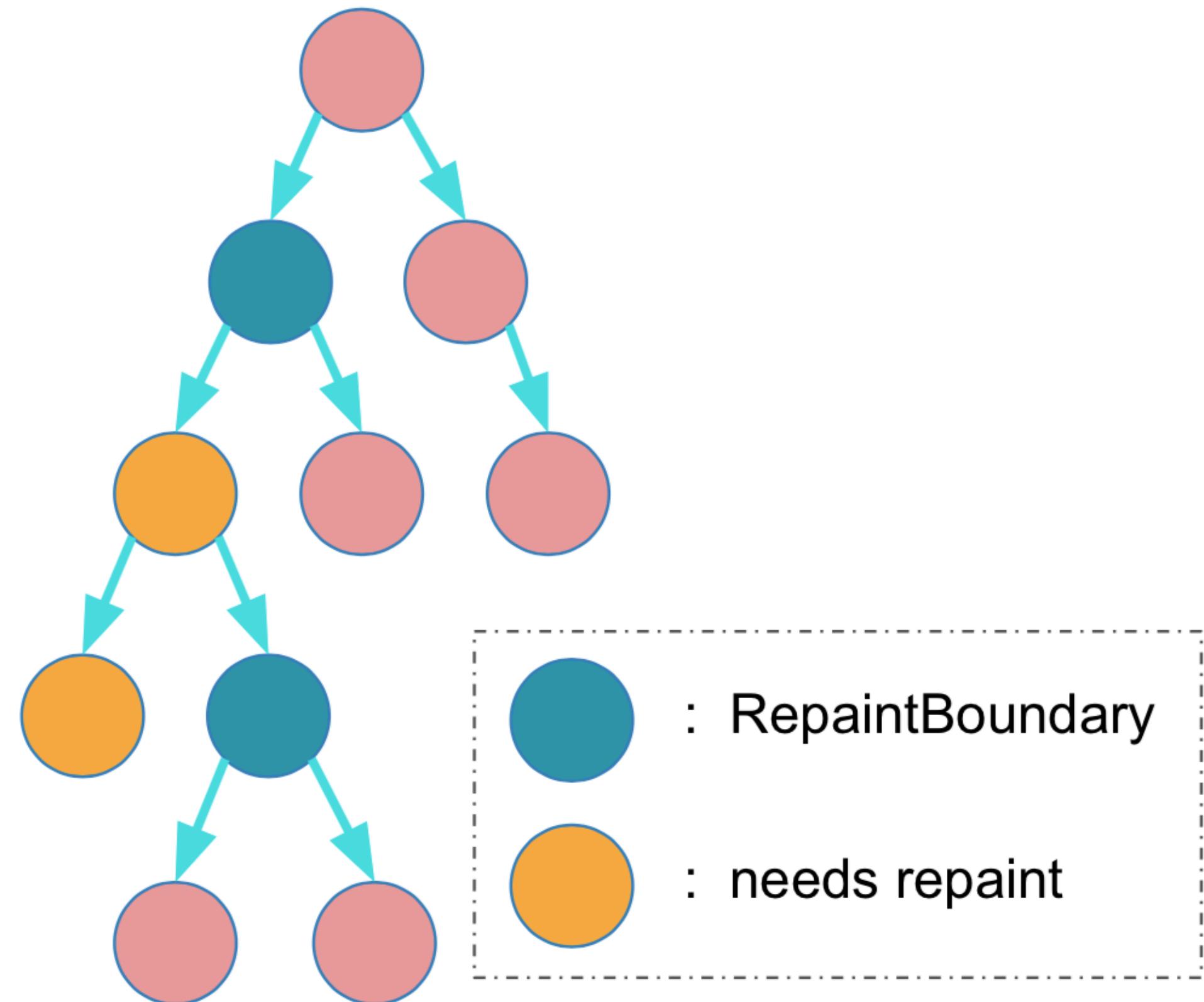
Widget
Element
RenderObject

渲染

布局更快



one-pass, linear time layout and painting
instead of $O(n^2)$



: RepaintBoundary

: needs repaint

Tree diagrams copied from Xiao's GDD China 2018 slides.

状态管理

- `setState()`
- Redux
- Bloc
- MobX

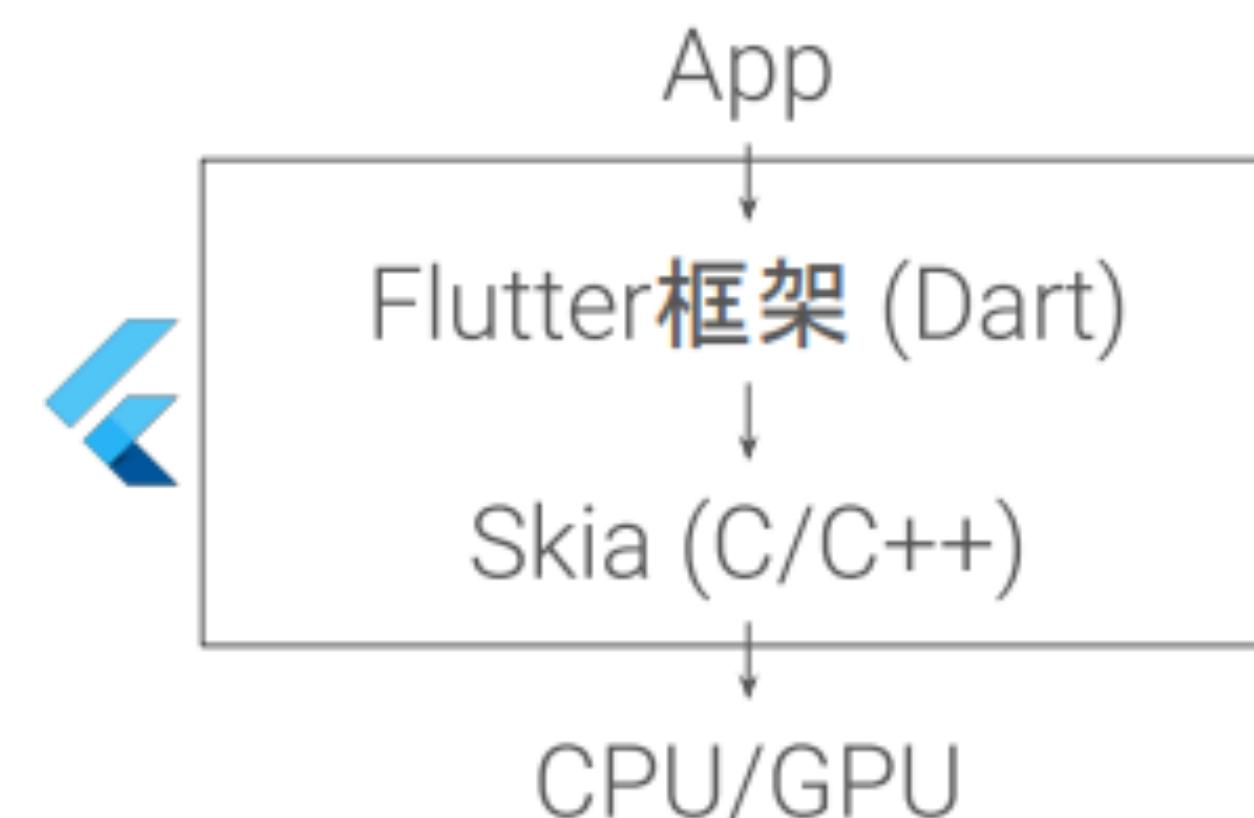
Codegen

```
1 import 'package:json_annotation/json_annotation.dart';
2
3 part 'request_params.g.dart';
4
5 @JsonSerializable()
6 class RequestParams {
7   String name;
8   int age;
9   String description;
10
11   RequestParams({
12     @required this.name,
13     @required this.age,
14     this.description,
15   });
16
17   factory RequestParam.fromJson(Map<String, dynamic> json) =>
18     _$RequestParamsFromJson(json);
19
20   Map<String, dynamic> toJson() =>
21     _$RequestParamsToJson(this);
22 }
```

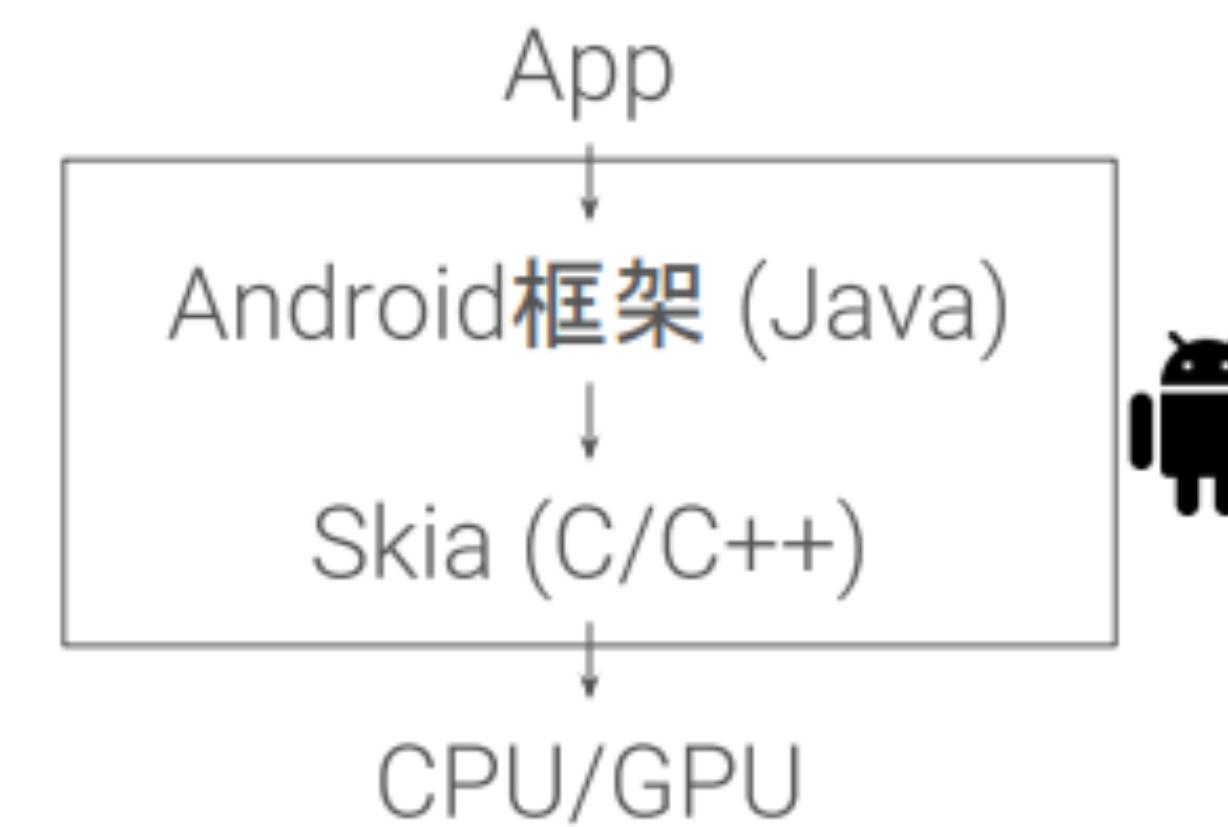
跨平台方案对比



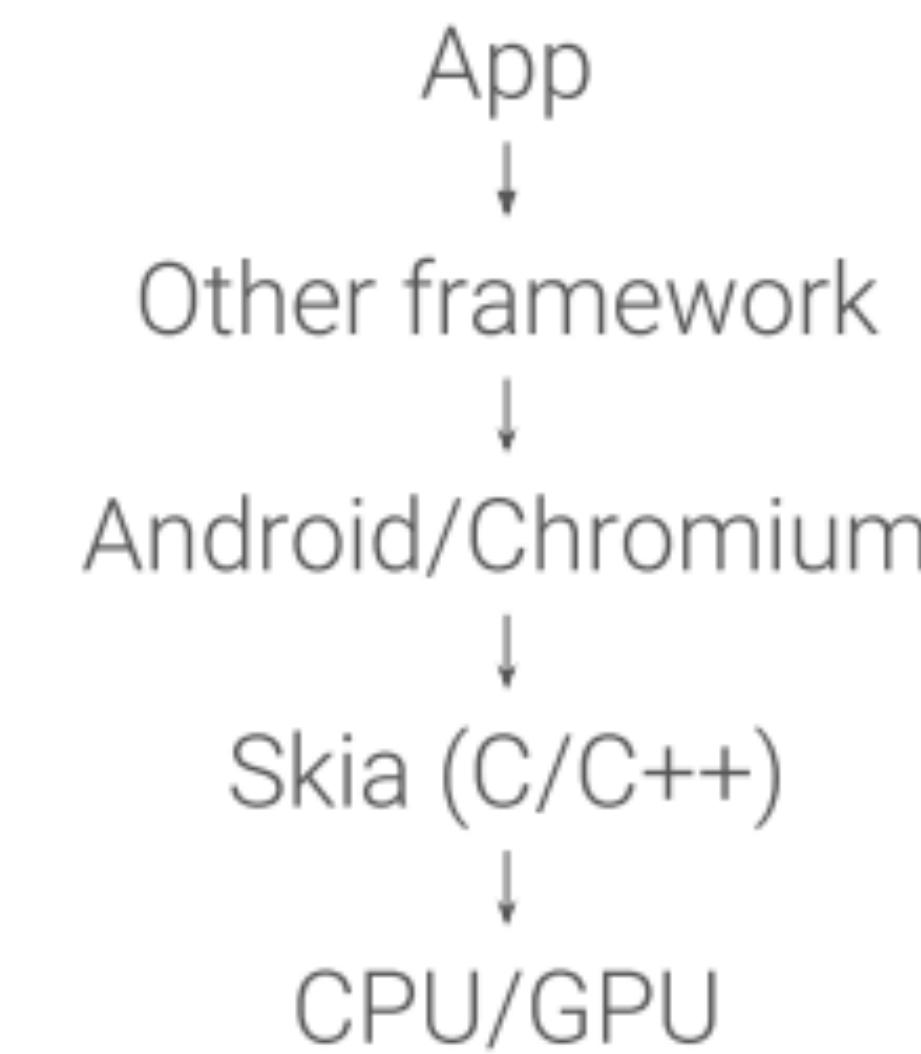
绘制层级



Flutter on iOS/Android



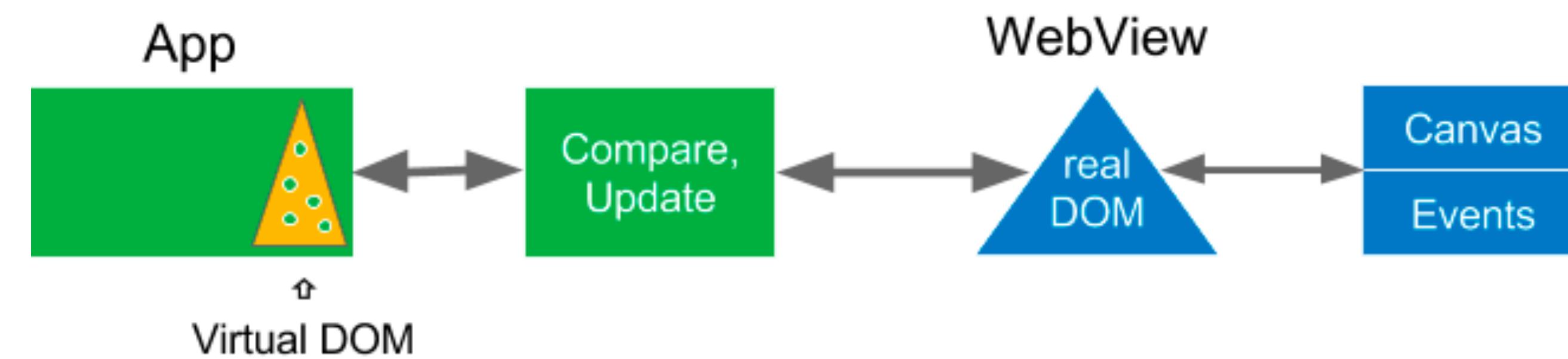
原生Android



其它跨平台框架

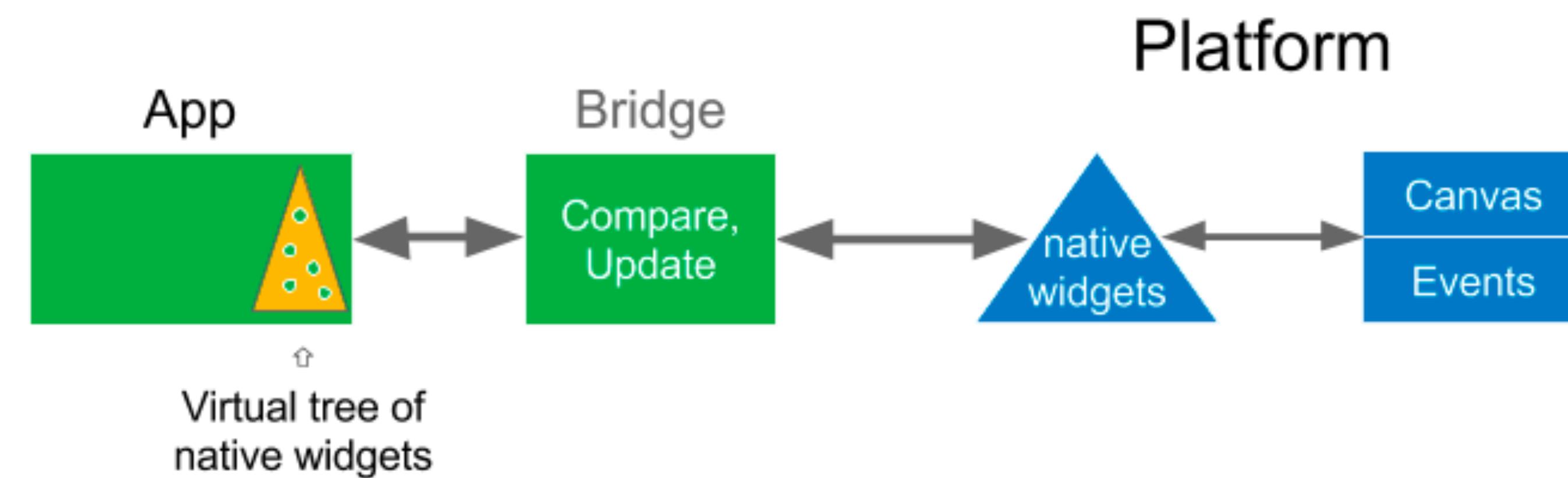
Widget 重绘

WebView



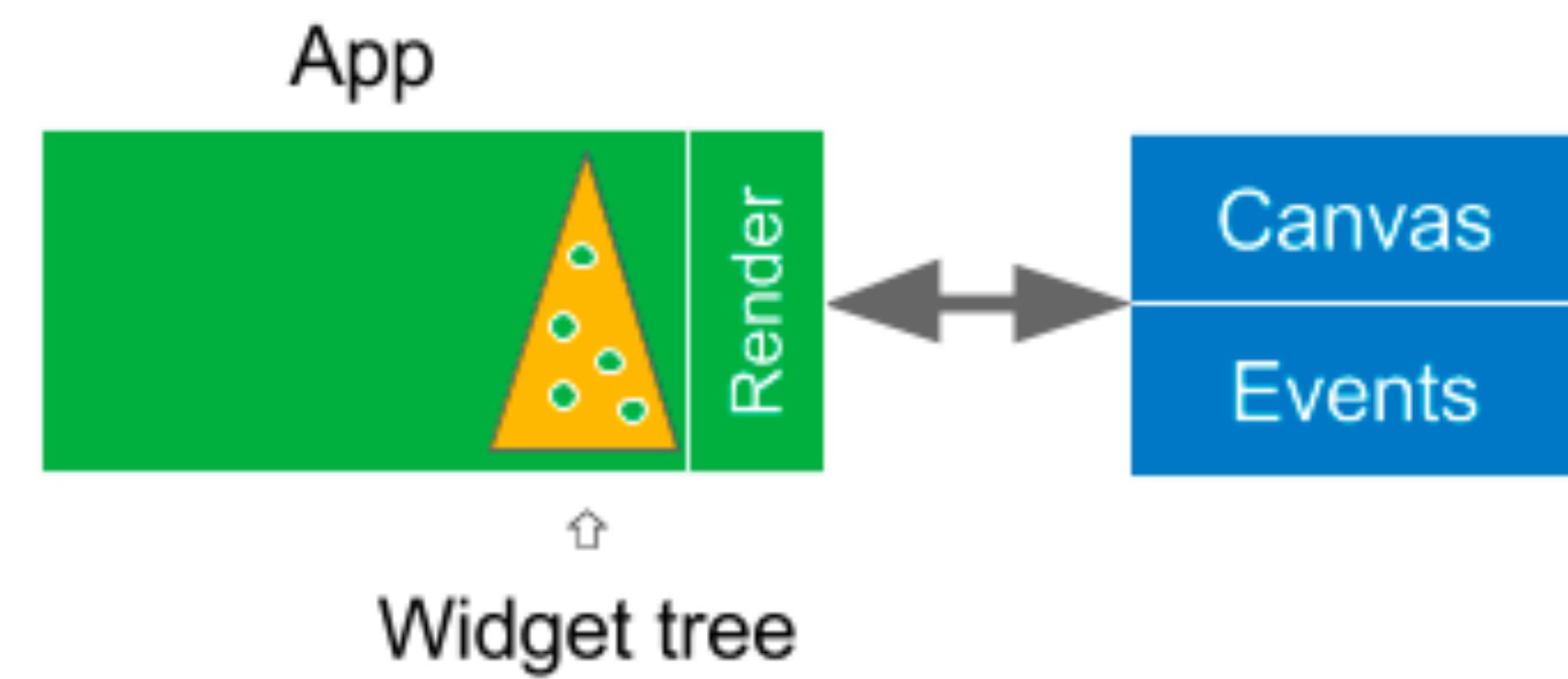
Widget 重绘

React Native



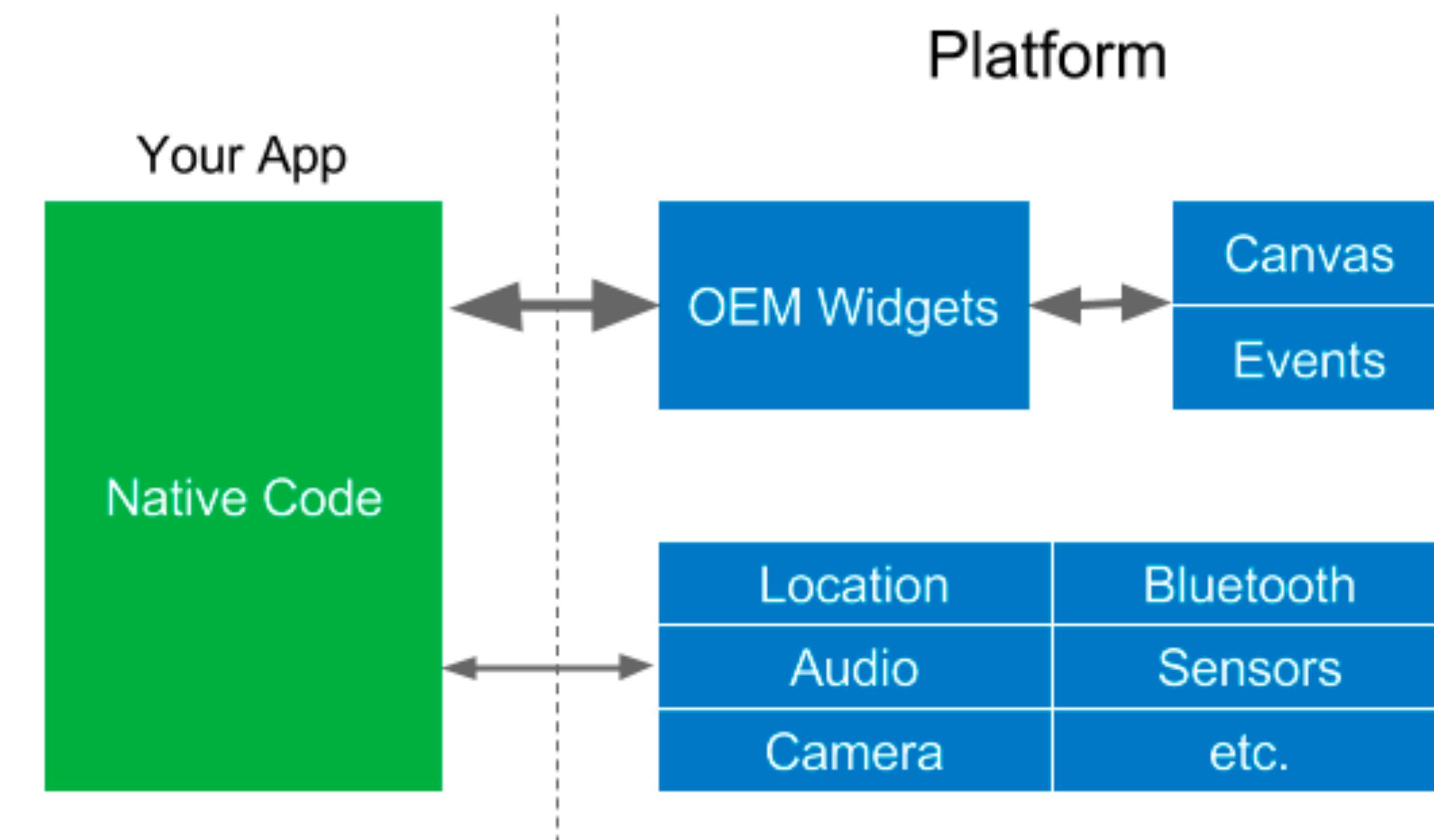
Widget 重绘

Flutter



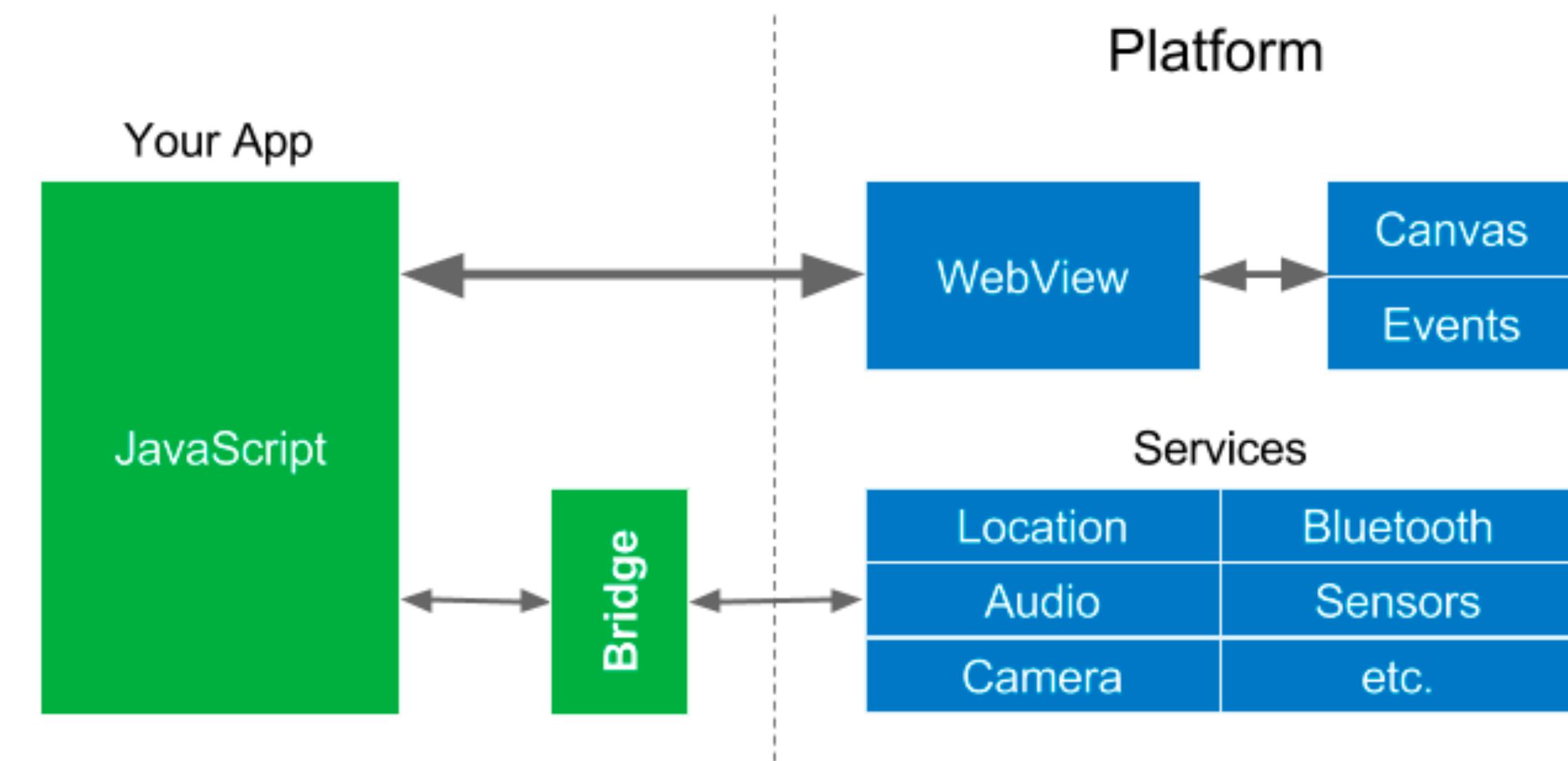
Structure

Native



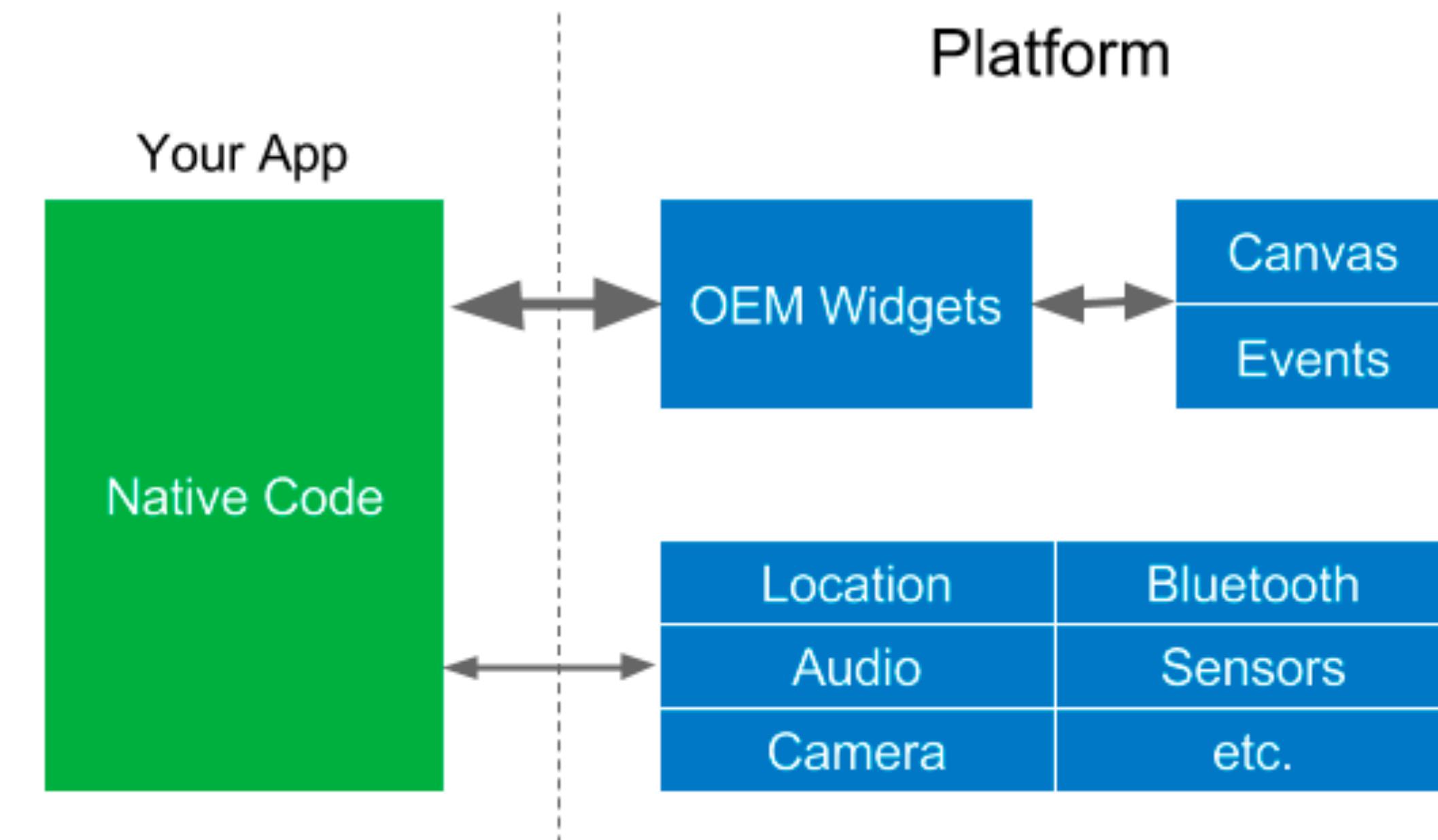
Structure

WebView



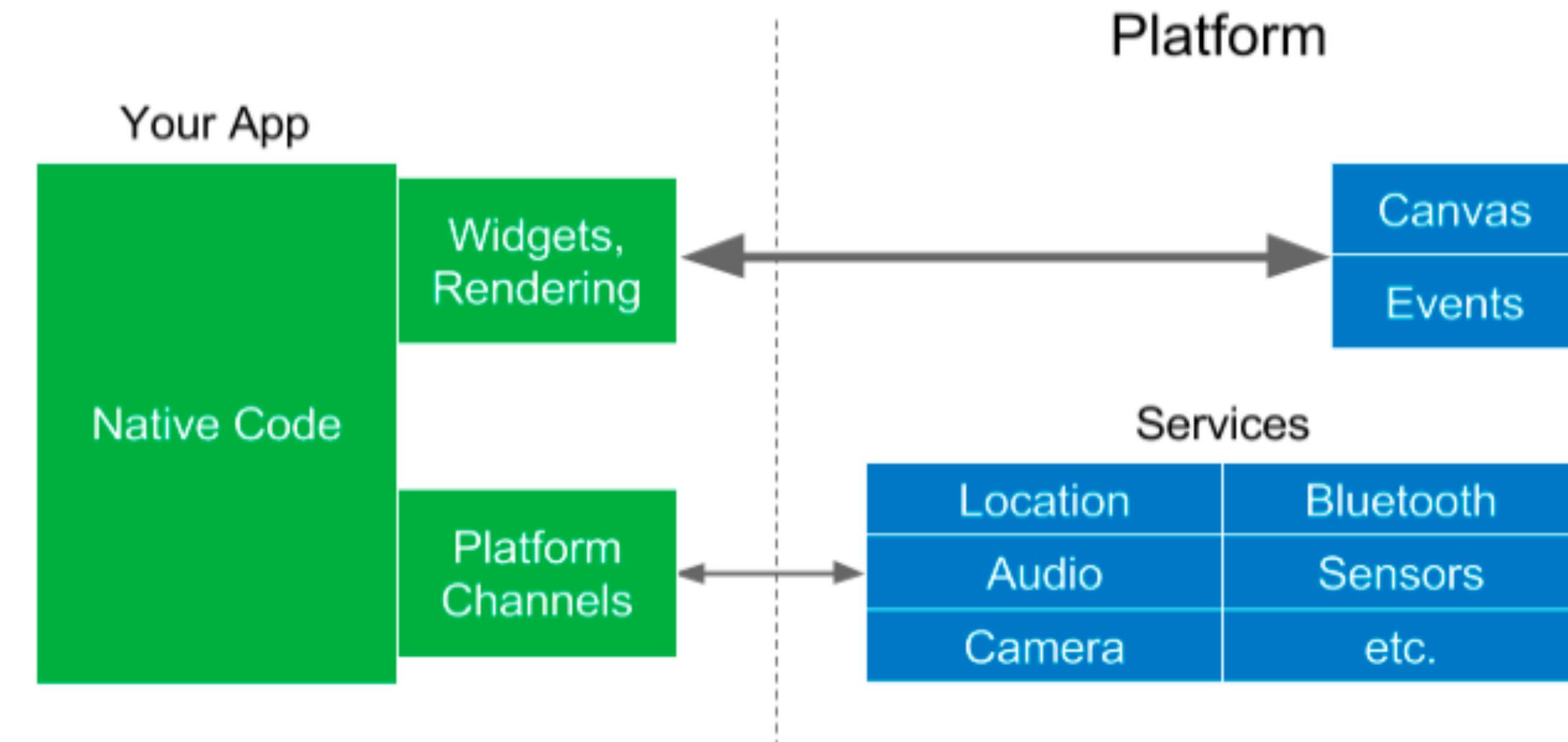
Structure

React



Structure

Flutter



好好学习 Flutter 建设



基础建设

Wrapper, Engine

- 更好的 Hybrid 混合应用接入
- 修改 Shell 层仅需要修改 wrapper
- 使用 socket push 调试应用

Bridge

- 前端同质化的 Bridge 体验
- 复用 web、react Bridge 现有基础
- 减少 Native 侧、Dart 侧的 channel 封装

Router

- 兼容现有的双端的 Router 方案
- 对 Router 提供了更高层次的封装
- 通过对规则配置的下发实现了页面升降级

体验优化

产物分离

为多端同学提供更简便的开发流程，
不会因为 Flutter 工程、
交叉编译的问题产生困扰。

Chanel Bridge 化

收敛于两端的 native 工程，
并且能够直接使用原有的 Web、
React-Native 的 Bridge，减少了开发成本。

多环境分离

支持多平台架构，
支持移动端之后开发流程更为方便。

Flutter 远程调试

Socket 远程连接的 Debug 模式
能够支持在 Hybrid 混合工程之中得到与
Flutter Debug 模式默认获得的相同体验效果。

场景化应用探索

Melon 壳工程



场景化应用探索

搜索页演示



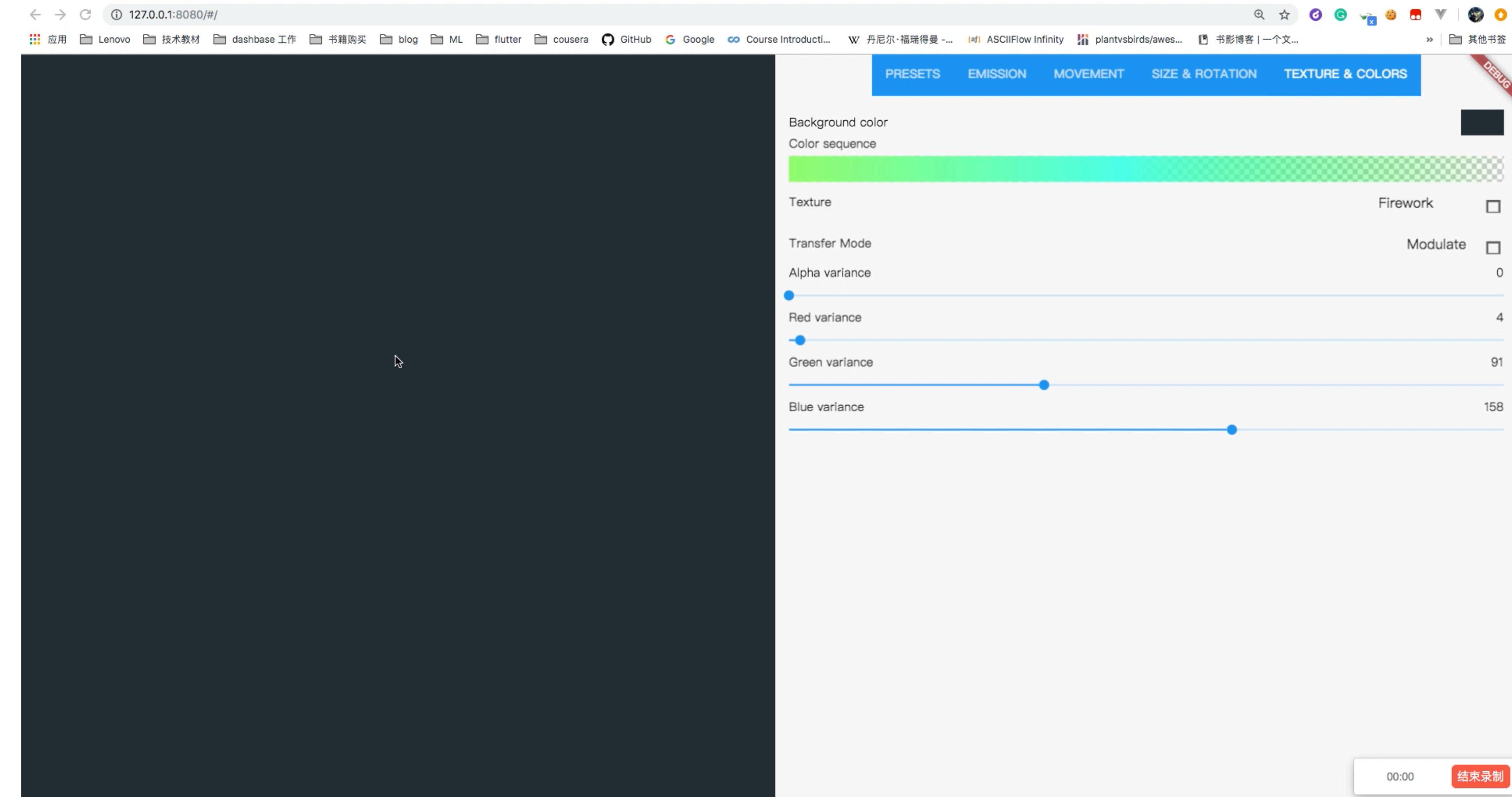
场景化应用探索

Valkyrie SDK



场景化应用探索

Flutter Web 粒子效果



https://github.com/flutter/flutter_web/pull/116

场景化应用探索

多源输入富文本组件



The screenshot shows a mobile application interface for a rich text editor. At the top, there is a blue header bar with the text "Techo 富文本演示". Below the header, there is a section titled "简介" (Introduction) containing the following text:

Techo 富文本渲染组件

- 使用独有的多输入源进行解析 rich_syntax
- 支持多数 quill clipboard 支持的 html 转换
- 提供 datus 对富文本数据结构进行支持
- 解析、渲染，前后端分离，使用 delta 数据结构进行数据连接

Below this, there is a section titled "基本用法" (Basic Usage) with a code snippet link:
[https://code.byted.org/learning_client/Techo](https://code/byted.org/learning_client/Techo)

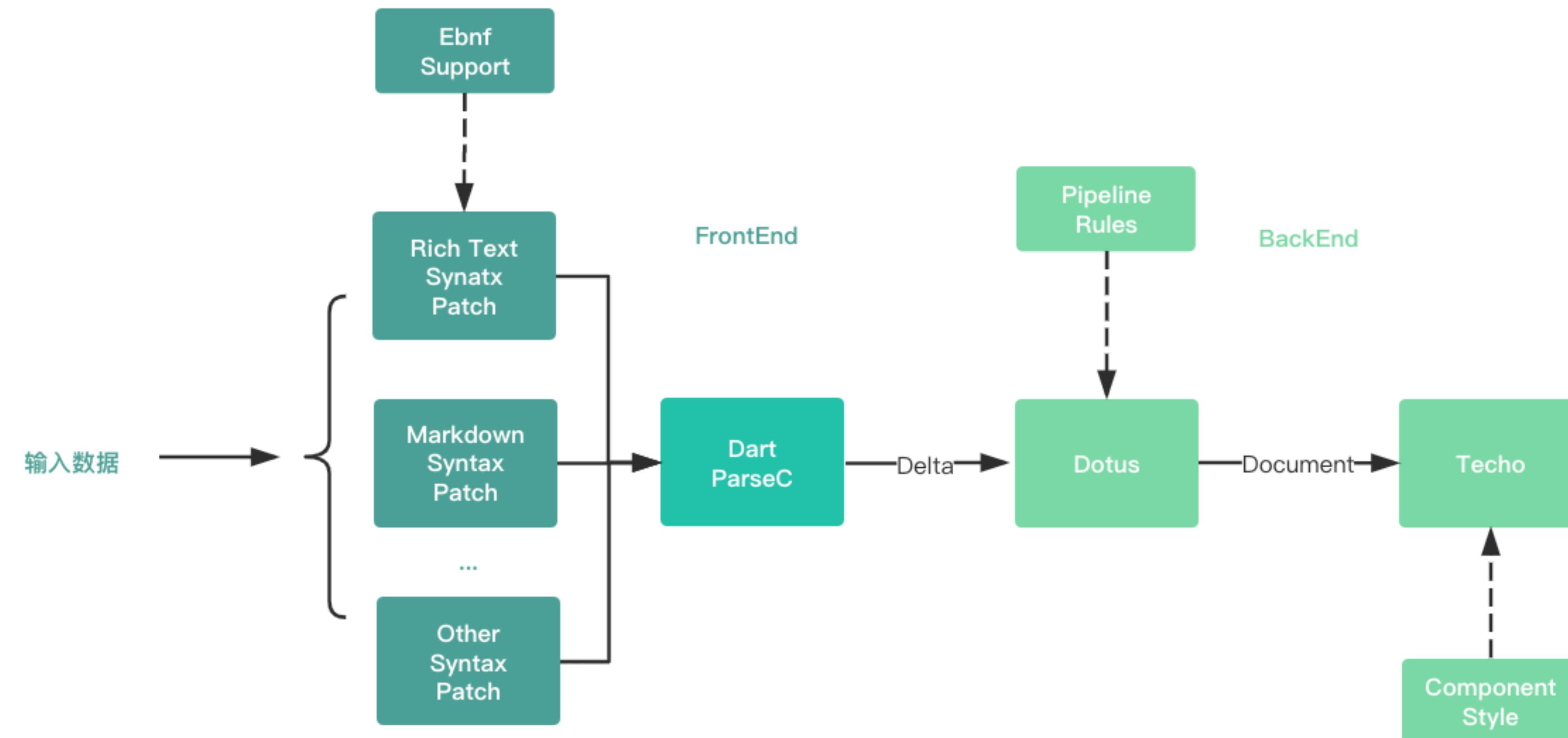
Further down, there is a section titled "输入样例" (Input Examples) containing a large block of HTML code:

```
<body>
<!-- Test Heading -->
<h1>Heading</h1>
<h1>H1</h1>
<h2>H2</h2>
<h3>H3</h3>

<h1>Paragraph</h1>
<!--文本段-->
<p>换行 <br> <strong>加粗</strong> <i>斜体</i></p>
<!--链接-->
<p>
    <a class="pgc-link"
        href="http://toutiao.com/group/6715326135770546695/?iid=0&app=news_article"
        target="_blank">文章</a> Ifkdsksk <i> i </i>
    </p>
    <!--序列-->
    <h1>List</h1>
    <ol>
        <li>Order Item 0</li>
        <li>Order Item 1</li>
        <li>Order Item 2</li>
    <ul>
        <li>UnOrder Item 0</li>
        <li>UnOrder Item 1</li>
```

场景化应用探索

多源输入富文本组件 Techo





THANKS.

