大前端新技术Flutter实践

CONTENTS

1 客户端挑战

2 思路 业内方

3 案 语言对比

4 Flutter的优势

5 业务特征

6

客户端挑战

效率

- UI还原
- 多端开发

质量

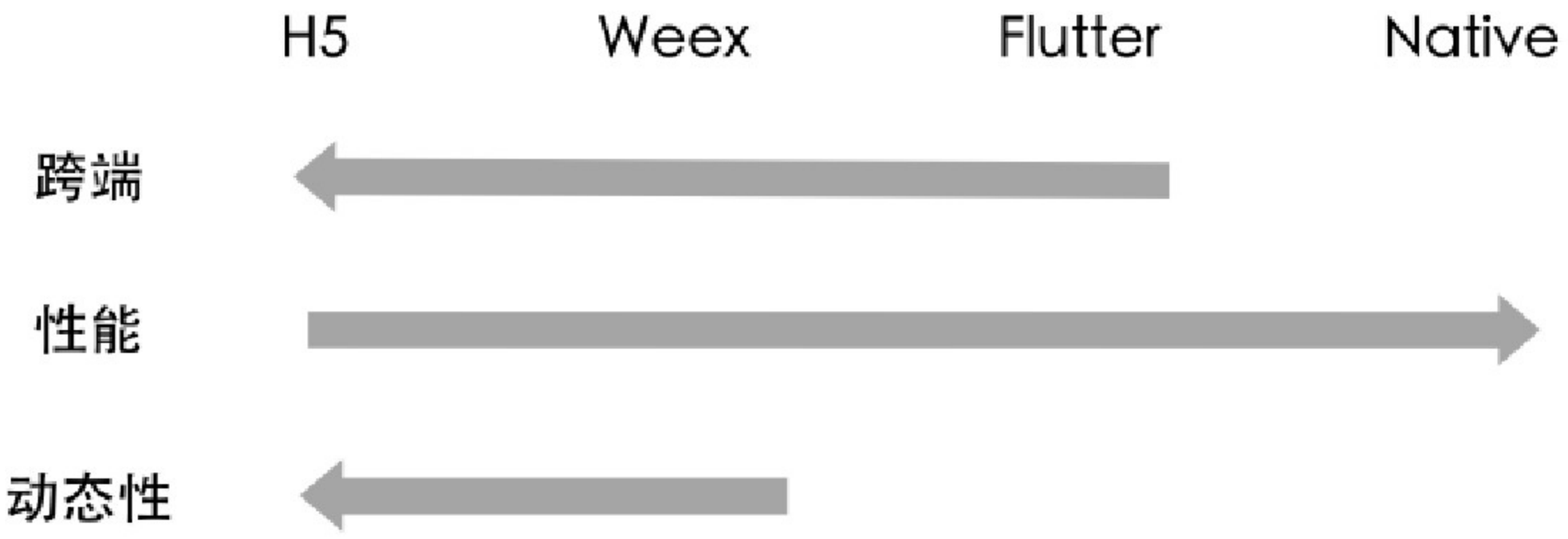
- 帧数/流畅度
- Crash率/Abort率

思路

跨平台的意义在于解决的是端逻辑的统一，至少避免了逻辑重复实现

保证质量的前提下，寻找一种可以接近原生体验的跨端方案

业内方案



语言对比

	HTML	Hybrid	Weex	React Native	Flutter
开发语言	JavaScript	JavaScript	JavaScript	JavaScript	Dart
成熟度	高	较高	较高	较高	一般
原生功能	差	差	高	高	高
开发效率	高	一般	高	高	高
兼容性	差	差	良	良	优
渲染性能	差	差	良	良	优

Flutter的优势

➢ 原生性能：AOT/更快的渲染引擎
➢ 多端一致性：多平台高度的一致性
➢ 友好而高效的研发体验：hot reload/attach debug/coroutine

Flutter 的平台无关能力能带来什么？

1、某些功能页面，可以一套代码实现，利用插件安装引入，在web、移动app、甚至 pc 上，都可以编译出对应平台的高性能代码，而不会像 RN 等一样存在各种兼容问题。

2、在应用上可以快速实现“降级策略”，比如某种情况下应用产生奔溃了，可以替换为同等 UI 的 h5 显示，而这些代码只需要维护一份。

React Native VS Flutter

- 性能:框架底层决定性能上 Flutter 的确实会比 React Native 好
- 支持:第三方包质量参差不齐,Flutter 是弱于 React Native
- 上手难易:Flutter 配置环境和运行的“成功率”比 React Native 高不少
- 稳定性:Flutter 中大部分异常是不会引起应用崩溃，更多会在 Debug 上体现为红色错误堆栈，Release 上 UI 异常等等

如果你是前端，推荐你学 React Native，如果你是原生开发，推荐学 Flutter

业务特征

- 一次性的活动页面-H5
- 布局简单的页面-RN
- 对动画、手势有特殊要求-原生 or Flutter
- 在旧客户端版本上迭代-H5 or RN
- 创新性业务-RN or Flutter

技术交流群

企业驾驶舱