腾讯跨平台&动态化开发框架演进之路

Lingtonke（柯灵杰）

InfoQ 写作平台

- InfoQ 写作平台是 InfoQ 开放给开发者的高端技术社区，创作者可以在这里自由创作和发布内容。
- 写作平台将为创作者提供签约、培训、资金扶持等一系列权益，助力作者成长为高精尖技术人才；同时也为企业提供品牌、活动打造、内容传播等服务，与伙伴一同成长。

扫码申请创作者
企业/个人均可申请

扫码进入写作平台
企业/个人均可申请

CONTENTS

跨平台技术演变

跨平台技术在业内的演变，价值，现状。

腾讯的跨平台实践

腾讯在跨平台技术上的发展路线、探索、实践、成果

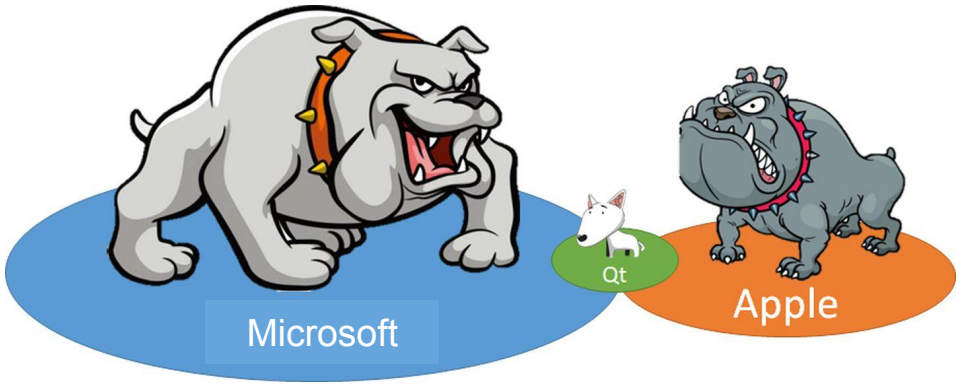
打造自研跨平台框架

为什么要打造自研跨平台框架？如何打造？现有框架如何融合？

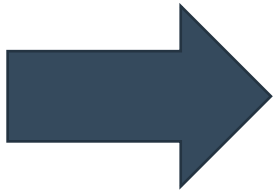
应用场景&未来&挑战

未来的价值，应用场景，面对的挑战和后续的规划

跨平台框架技术的演变



桌面端跨平台



移动端跨平台

QT

JAVA

wxWidgets

GTK+

Xamarin

Google

Flutter

PWA

AMP

Microsoft

Xamarin

Facebook

React Native

Adobe

PhoneGap

Tencent

Hippy

小程序

阿里巴巴

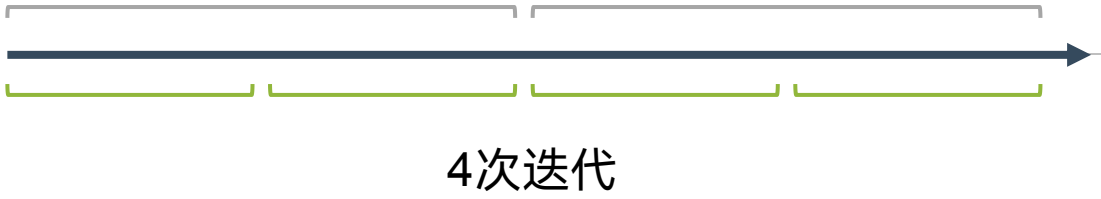
WEEX

Weex

Appcelerator

Titanium

从跨平台到动态化



Q1

提高研发效能

不管是跨平台技术、还是动态化技术，本质上是为了提高研发效能

Q2

持续部署、持续交付

动态化能力为客户端提供了持续部署和持续交付的能力

Q3

加快迭代速度

具有持续交付能力之后，最大的收益是提高了迭代的速度

Q4

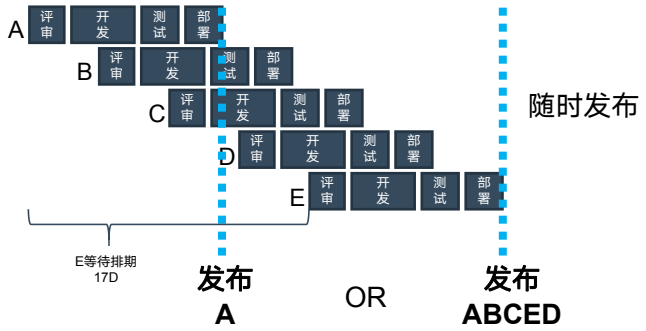
实现数据驱动

高速的迭代速度及方便的A/B实验能力，是数据驱动的基石

从跨平台到动态化

按版本交付

持续交付



版本

基于版本：传统月版本模式

0~30（等待版本节奏）+30（版本周期）+2（审核）+N（更新，按2算）

= **34~64D** or 加班插需求

动态化

基于动态化：持续部署、持续交付

0~17D（等待排期）+6（评审、开发、测试）+0（审核）+0（更新）

= **6~23D**

Q1 提高研发效能

不管是跨平台技术、还是动态化技术，本质上是为了提高研发效能

Q2 持续部署、持续交付

动态化能力为客户端提供了持续部署和持续交付的能力

Q3 加快迭代速度

具有持续交付能力之后，最大的收益是提高了迭代的速度

Q4 实现数据驱动

高速的迭代速度及方便的A/B实验能力，是数据驱动的基石

游戏引擎 VS UI引擎

为什么不用
游戏引擎做UI？

01 刷新方式：全局刷新 VS 局部刷新

02 更新机制：实时更新 VS 标脏机制

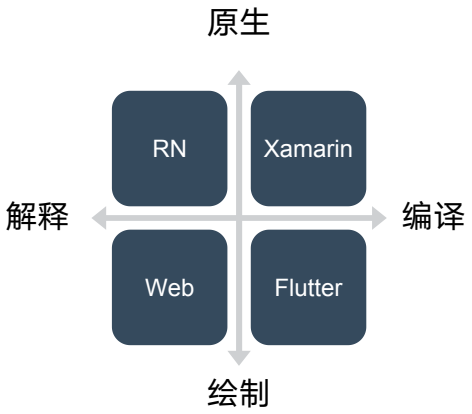
03 渲染缓存：无缓存 VS 多级缓存

04 内存占用：大 VS 小

05 运行功耗 ： 高 VS 低

06 UI组件：少 VS 多

行业现状



Google

Flutter　PWA　AMP

Microsoft

Xamarin

Facebook

React Native

Adobe

PhoneGap

Tencent

Hippy　小程序　MXFlutter

阿里巴巴 Alibaba.com

WEEX

Weex

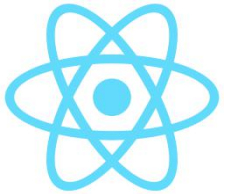
Appcelerator

Titanium

腾讯在跨平台与动态化上的实践

过去：探索 → 现在：并存 → 未来：融合&创新

体验



React Native

自研

Hippy

性能、一致性

腾讯动态化框架
Tencent Dynamic Framework

生态

Flutter

动态化

MXFlutter

早期 Hybrid 实践及遇到的问题

1

线程问题

异步加载，白屏，响应不及时

2

性能问题

运行性能差，通信性能差，体验天花板低

3

离线问题

默认只支持在线浏览，离线需要改造，体验不好

4

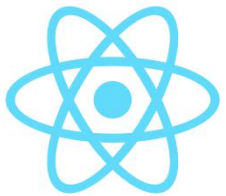
生态问题

客户端生态、Web 生态不一致

从 Hybrid 到 React Native 再到 Hippy

——Web前端生态的变革

- 基于 WebView
- 浏览器内核不可控
- 不支持离线
- 无法和原生控件混合

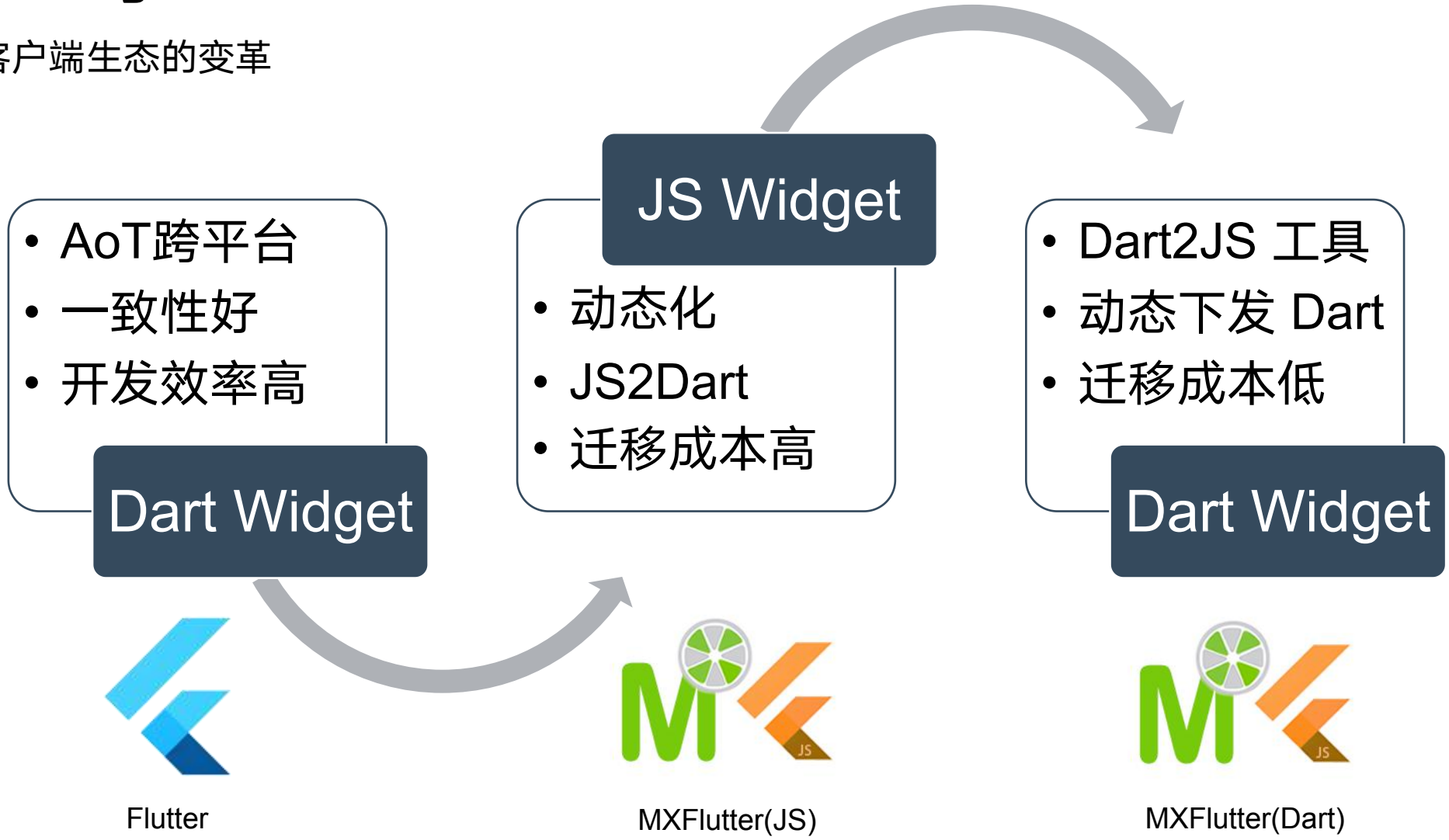


- 基于 Native
- 更高性能
- 开源内核
- 支持离线
- 可以和原生控件混合

- 无协议风险
- 同时支持 React / Vue
- 首屏直出
- 支持分包加载
- 支持局部更新
- 包体更小
- C++ 排版引擎，性能更高

Flutter 与 MXFlutter

——客户端生态的变革



打造腾讯自研跨平台动态化框架

腾讯动态化框架
Tencent Dynamic Framework

生态

Hippy

MXFlutter

小程序

TDF View
TDF Widget
TencentOffice

其他领域

+

底层

AppCenter
平台

Core
内核

DevTools
工具

打造高性能的跨平台渲染内核

兼容多种生态的架构

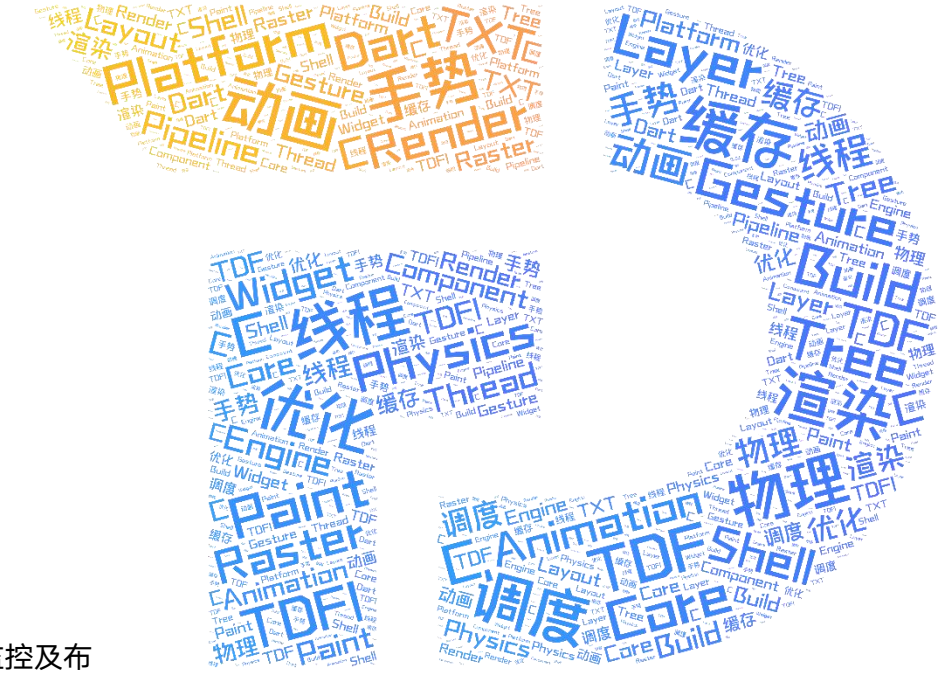
支持前端、客户端多种生态，为不同需求提供完善的底层支持

自优化的超流水线设计

采用PGO优化、多阶段并行超流水线设计，进一步释放硬件潜能提高性能

健壮的内存和线程模型

稳定精确的线程调度策略，结合内存持续监控及布局分析，保障内核整体的稳定性



灵活快速渲染树结构

多级分层树结构，在充分满足渲染能力需求的情况下极致压榨性能

智能高效的缓存结构

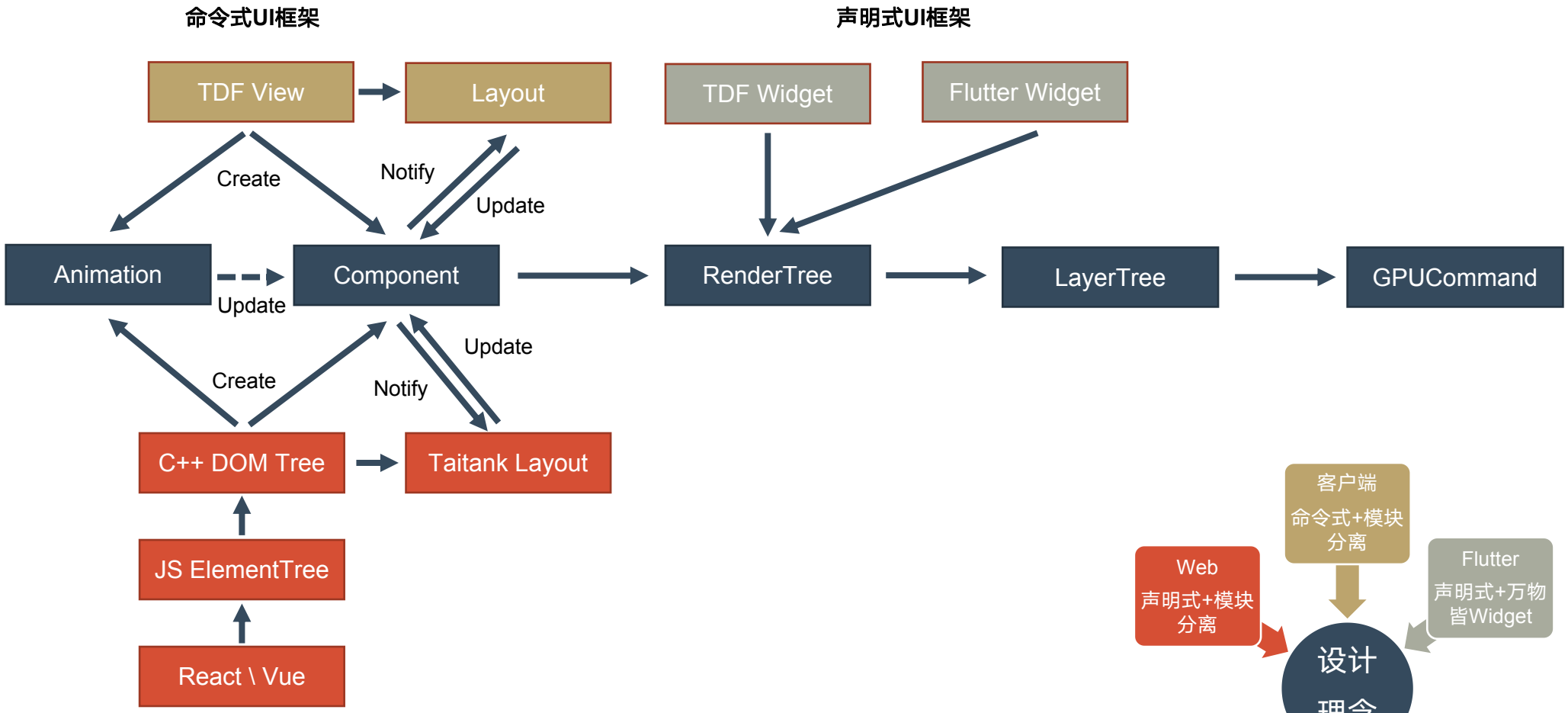
TreeCache、L2 CommandCache、L1 RasterCache。多级缓存为性能保驾护航

全面有效的质量保障

内存监控、线程检测、设计先行、高覆盖率单元测试、结对编程、多轮CR，保障质量

兼容多种生态的框架设计

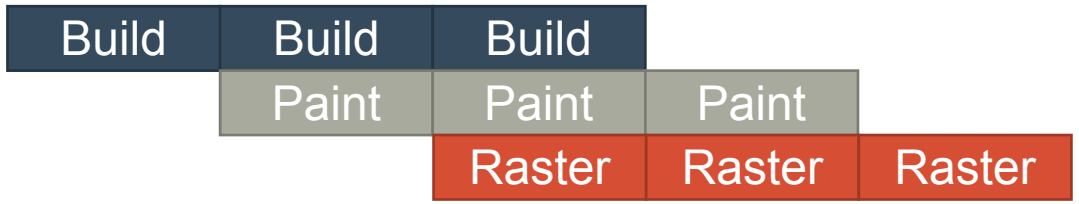
灵活快速渲染树结构

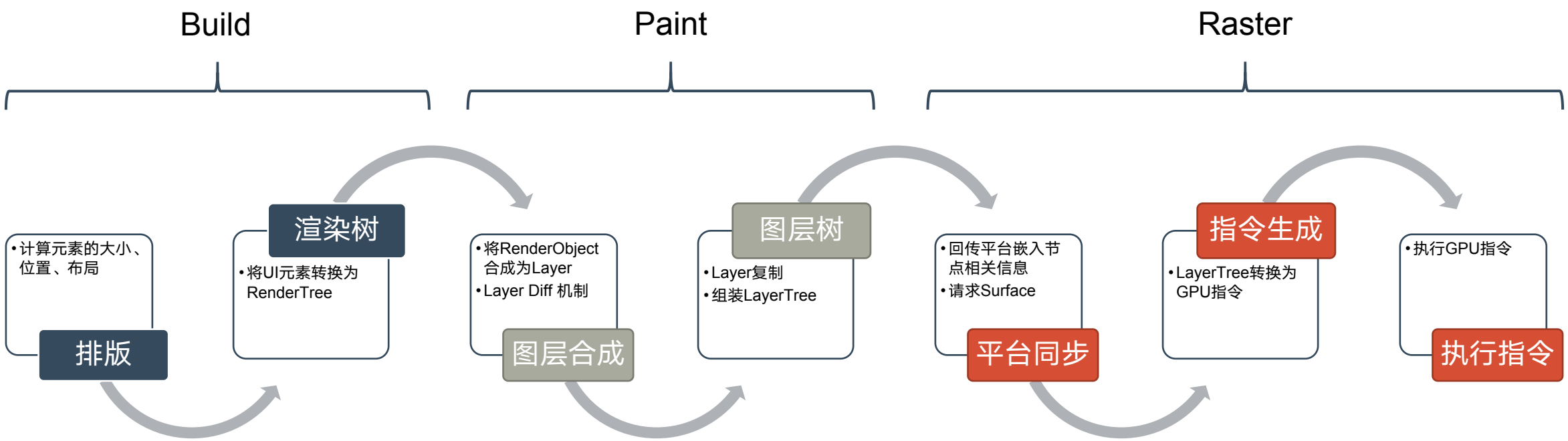

命令式UI框架
声明式UI框架
TDF View
Layout
TDF Widget
Flutter Widget
Create
Notify
Update
Animation
Update
Component
RenderTree
LayerTree
GPUCommand
Update
Create
Notify
C++ DOM Tree
Taitank Layout
JS ElementTree
React \ Vue
Hippy

客户端
命令式+模块分离
Web
声明式+模块分离
Flutter
声明式+万物皆Widget
设计理念

自优化的超流水线设计

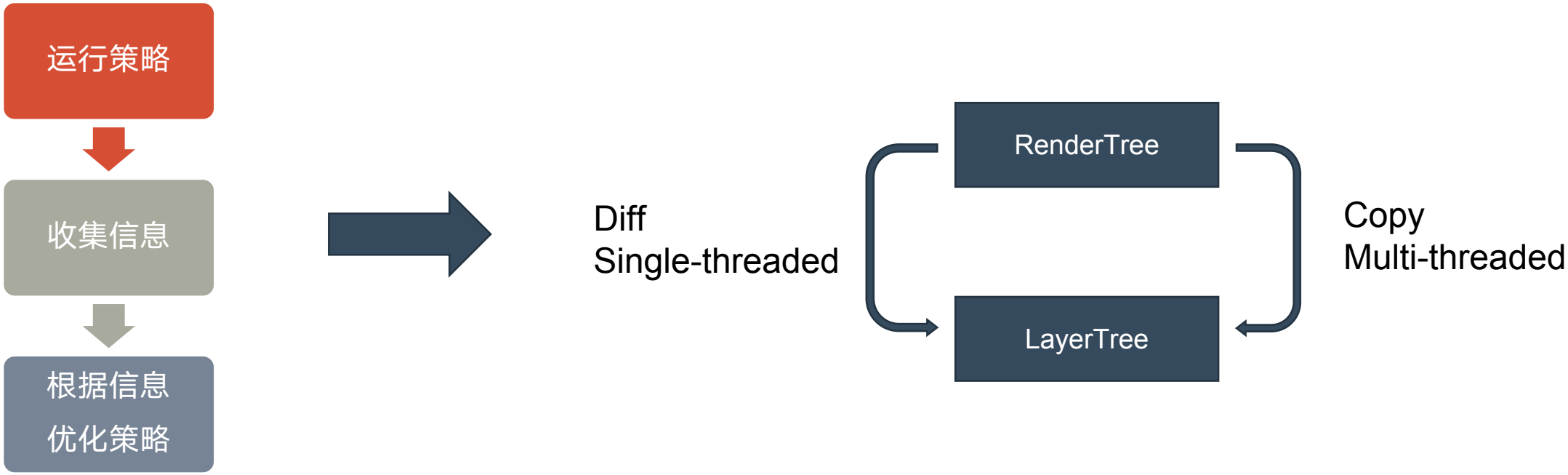
—— 流水线流程



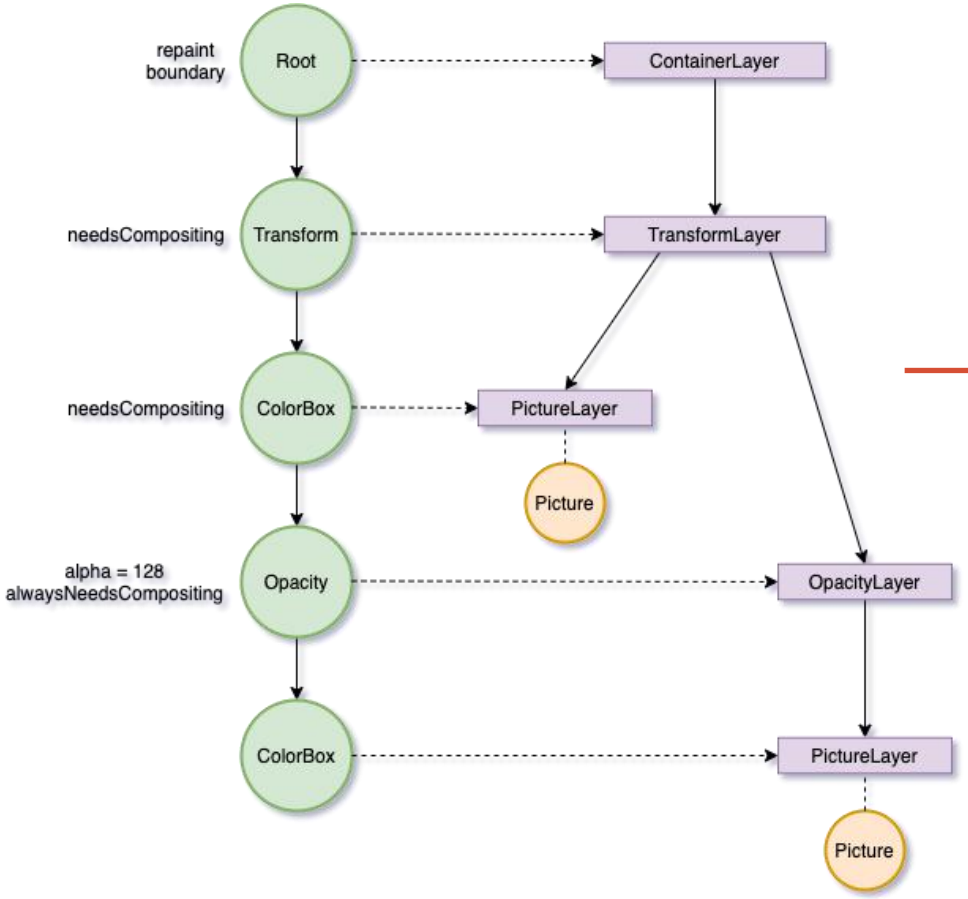


自优化的超流水线设计

—— PGO（ Profile-guided optimization ）：让程序自己优化自己



智能高效的缓存结构



TreeCache

通过节点合并、渲染树diff，实现局部更新逻辑。减少页面刷新耗时，提高效率

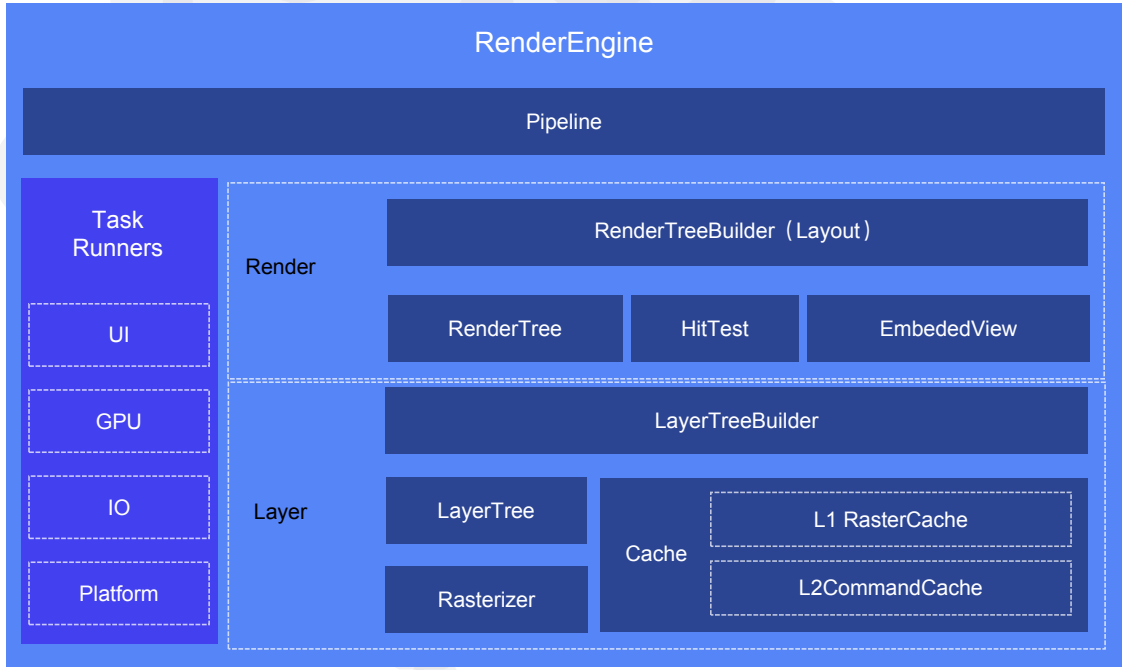
L1 RasterCache

渲染节点具有保存纹理缓存的能力。
对于没有发生变化的节点，优先使用纹理缓存，最大幅度减少UI、GPU开销

L2 CommandCache

渲染节点具有保存指令缓存的能力。
在RasterCache不命中的情况下，复用之前生成的渲染指令，一定幅度减少开销。

健壮的内存和线程模型



内核——渲染引擎

01 每个函数恒定执行线程

02 跨线程函数专有命名规范

03 Debug模式线程正确性检测

04 大量无锁算法应用

05 支持ABA线程模型、ABC线程模型切换

06 无裸指针使用、内存自动监控

全面有效的质量保障

——研发流程

概要设计

- 文档编写：架构图、流程图
- 结对审阅
- 负责人审阅

代码编写

- 编写功能代码
- Public接口注释覆盖率100%
- 单元测试覆盖率75%

合流验证

- DevOps流水线检查（代码规范、圈复杂度、单测覆盖率）
- 结对代码CR
- 负责人CR

合流完成

- DevOps流水线二次验证，失败回滚
- 每日全量用例自动化测试
- 性能测试、生成报告

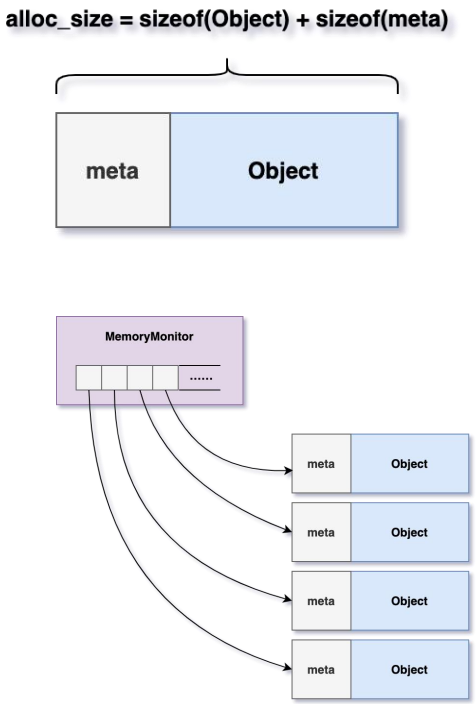
发布

- DevOps流水线编译、验证
- 全量用例自动化测试
- 性能测试、生成报告
- 关键功能人工CheckList
- 线上监控告警，自动回滚

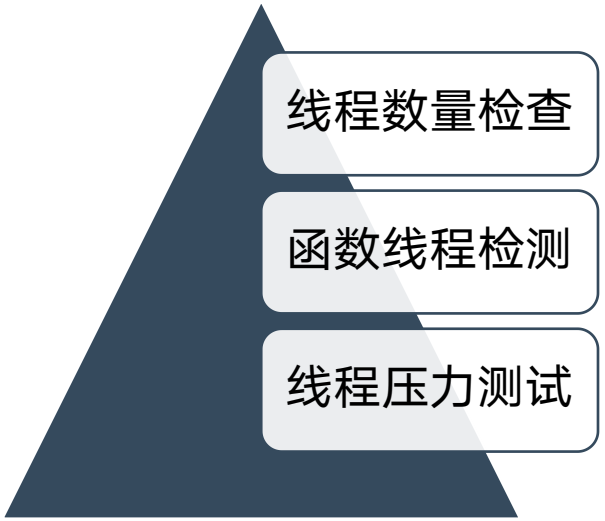
全面有效的质量保障

——系统、质量保障

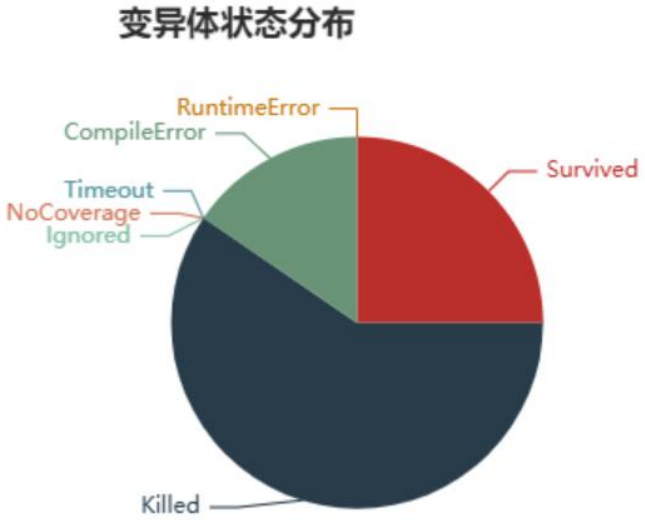
内存监控



线程检测

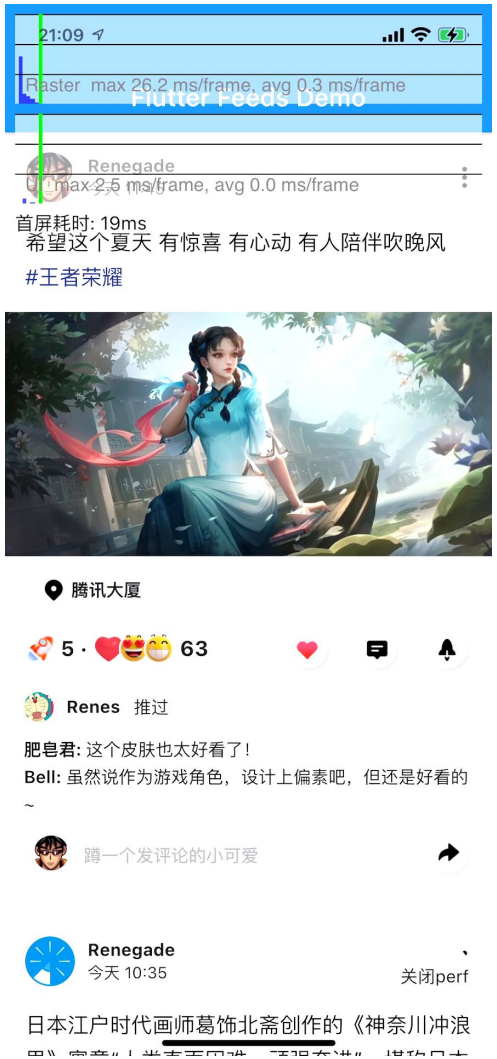


变异测试



性能测试

Flutter



TDF Widget+内核

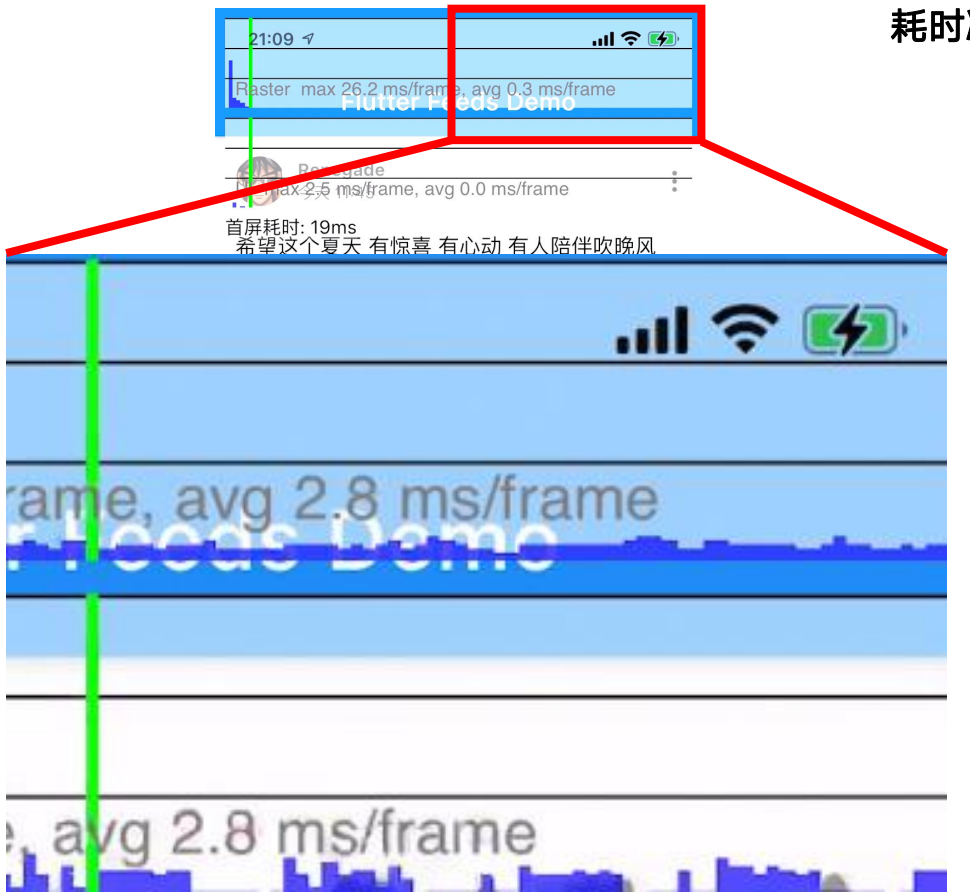
性能测试

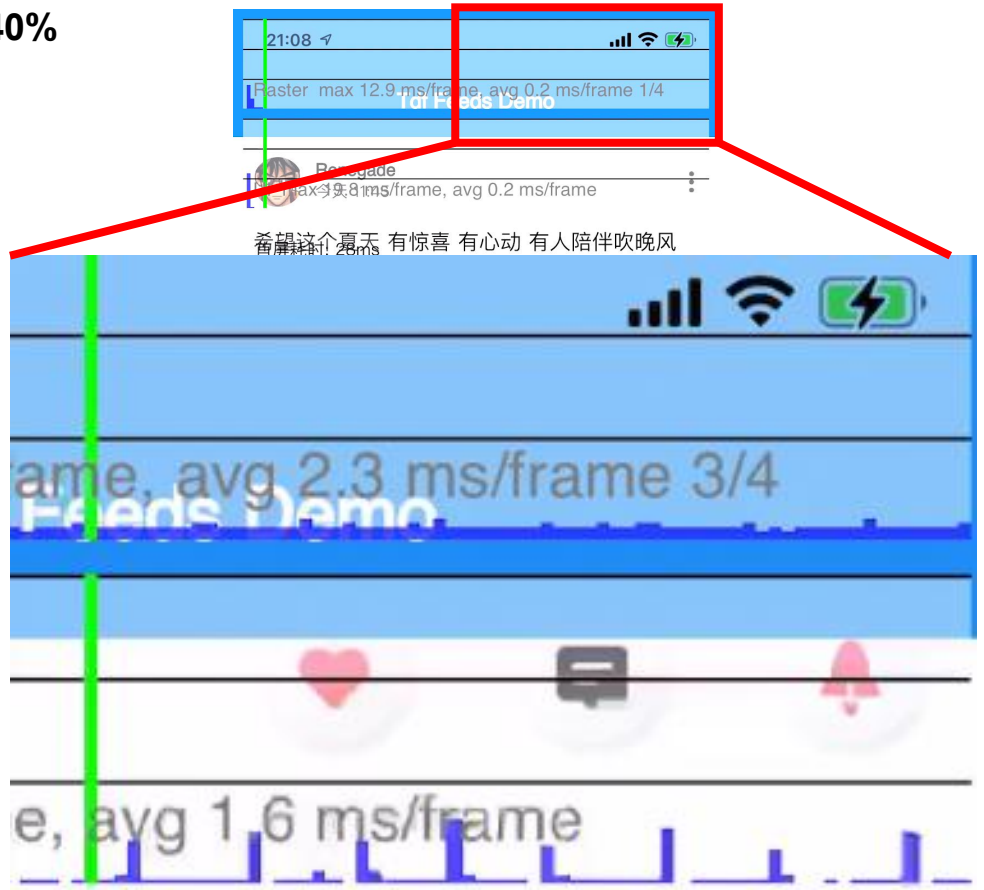
Flutter

2.8ms **VS** 1.6ms

耗时减少40%

TDF Widget+内核





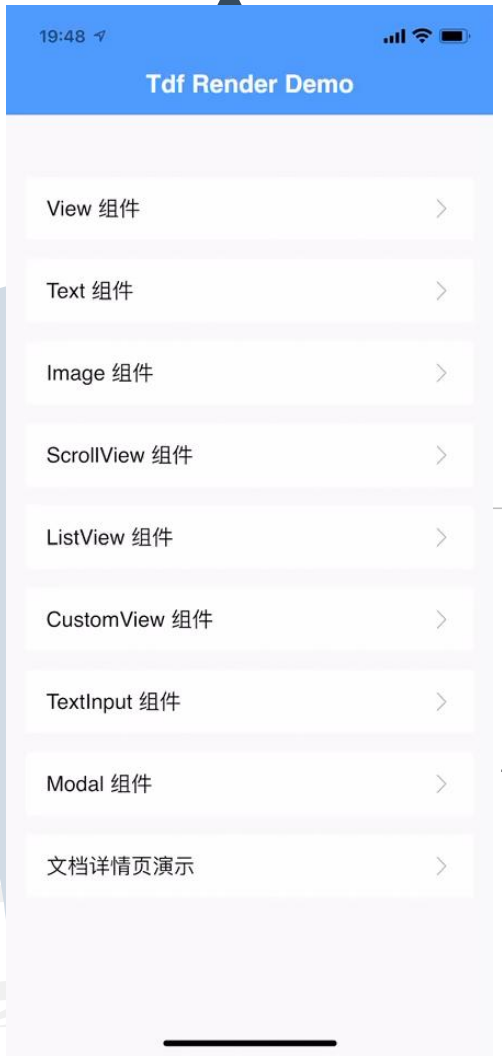
应用场景

独立跨端、UI 框架

借助 TDF 内核提供的 Component 基础 UI 组件，结合排版、手势、动画、渲染模块，形成可以独立使用的UI框架

现有框架内核

通过将 Hippy 内核及 C++DOM 树对接到 TDF 内核，实现 Hippy 从 NativeUI 到自渲染 UI 的升级



TencentOffice 内核

腾讯文档将升级为 TencentOffice，加强在多端原生能力上的投入。TDF 内核提供了强大的底层能力支持，为业务保驾护航

后台渲染支持

借助 TDF 内核跨平台能力，及多平台渲染一致性，实现后台渲染支持。在预览图生成，文档离线渲染方面，有重要作用

挑战&展望未来

大家都在基于 Flutter 的时候
我们在打造新的框架和内核
在基础框架领域发出中国人的声音

今年下半年在腾讯内部项目试点
线上验证后，年底开源
期待与大家相遇

站在巨人的肩膀上，我们能做的更好

- 规模大：超过 30 个大型模块，数百个小型模块
- 领域多：深入数十个不同细分领域
- 周期长：需要持续多年投入才能有产出
- 门槛高：多名T12、T11带队，大量T9以上工程师投入
- 开发难：保障质量，每人每天输出少量高质量代码
- 配套多：包含平台、框架、工具、文档等配套设施

谢谢大家

Lingtonke(柯灵杰)
腾讯动态化框架中台PMC

InfoQ 写作平台

- InfoQ 写作平台是 InfoQ 开放给开发者的高端技术社区，创作者可以在这里自由创作和发布内容。
- 写作平台将为创作者提供签约、培训、资金扶持等一系列权益，助力作者成长为高精尖技术人才；同时也为企业提供品牌、活动打造、内容传播等服务，与伙伴一同成长。

扫码申请创作者
企业/个人均可申请

扫码进入写作平台
企业/个人均可申请

腾讯跨平台&动态化开发框架演进之路

扫描二维码　提交议题反馈