QCon[上海站]
全球软件开发大会2016

Weex极致性能优化

SPEAKER

隐风

International Software
Development Conference

主办方 Geekbang 极客邦科技 InfoQ

个人简介

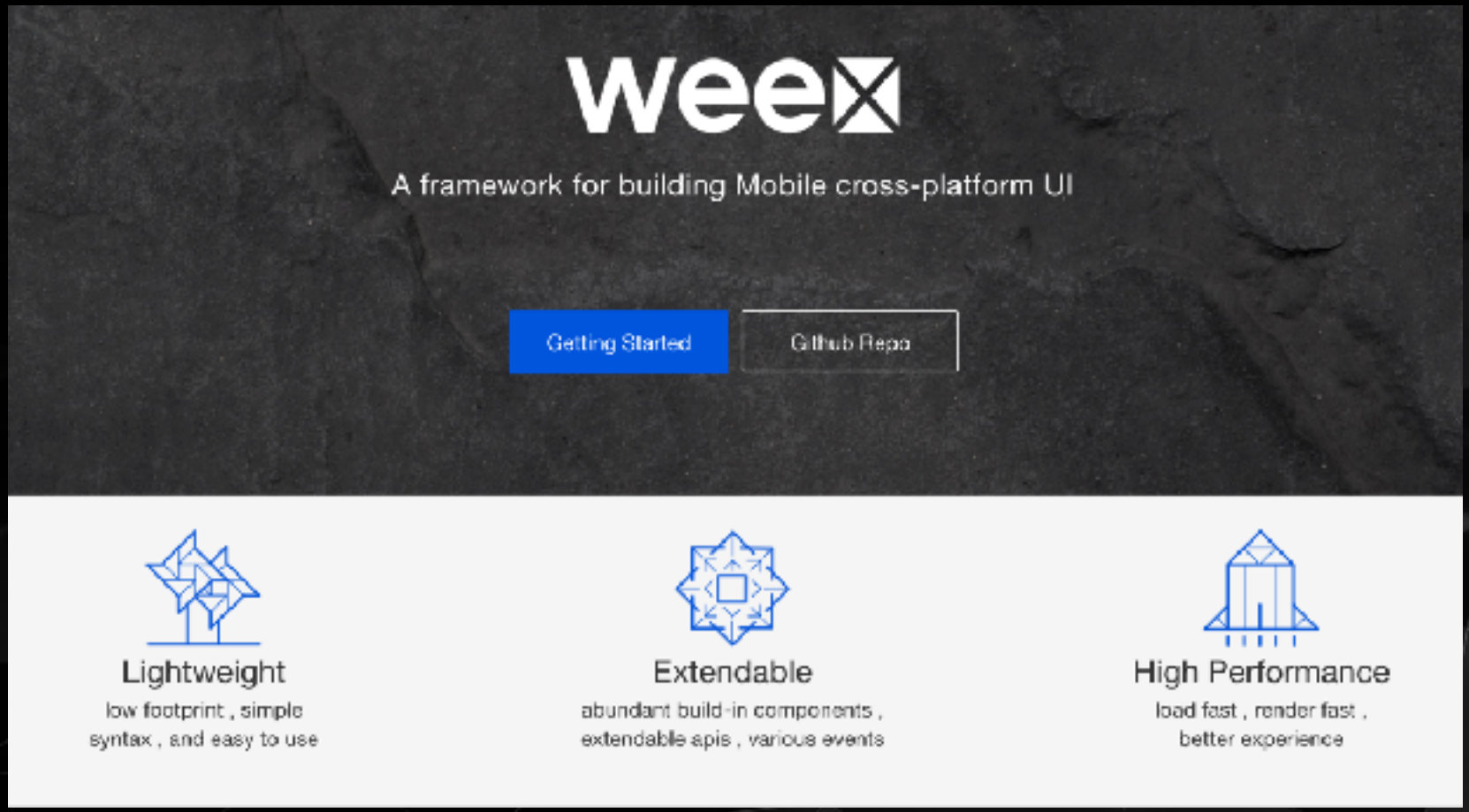
Alibaba Group 阿里巴巴集团

- 冯成晓(花名隐风)
- 2013年加入阿里无线事业部
- Hybrid框架/动态化框架
- Weex 渲染引擎
- iOS/Web

Alibaba Group 阿里巴巴集团

Weex

f(HTML/CSS/JS) = Native UI



https://alibaba.github.io/weex/

WEEX－动态性不懈追求

Alibaba Group 阿里巴巴集团

```
<template>
  <div class="container">
    <div class="logoContainer">
      <image class="logo" src="http://t.cn/RqObcDm">
      </image>
      <text class="desc">
        A framework for building Mobile cross-platform UI
      </text>
    </div>
    <div class="textContainer">
      <text class="text">Hello weex</text>
    </div>
  </div>
</template>
<style>
  .logoContainer {
    width: 750px;height: 500px;
    align-items: center;justify-content: center;
    background-color: #0054DC;
  }
  .logo {
    width: 378px;height: 84px;
  }
  .desc {margin-top: 40px;font-size: 24px;color: #ccc;}
  .textContainer {
    width: 750px;padding-top: 200px;
    align-items: center;justify-content: center;
  }
  .text {font-size: 48px;}
</style>
```

三端一致体验

三端使用体验一致性，渲染一致

语法轻量

语法简单,一小时快速上手。JSFrm 36K(GZIP)

内核更轻量

Android 1.9M、iOS 700K

高性能

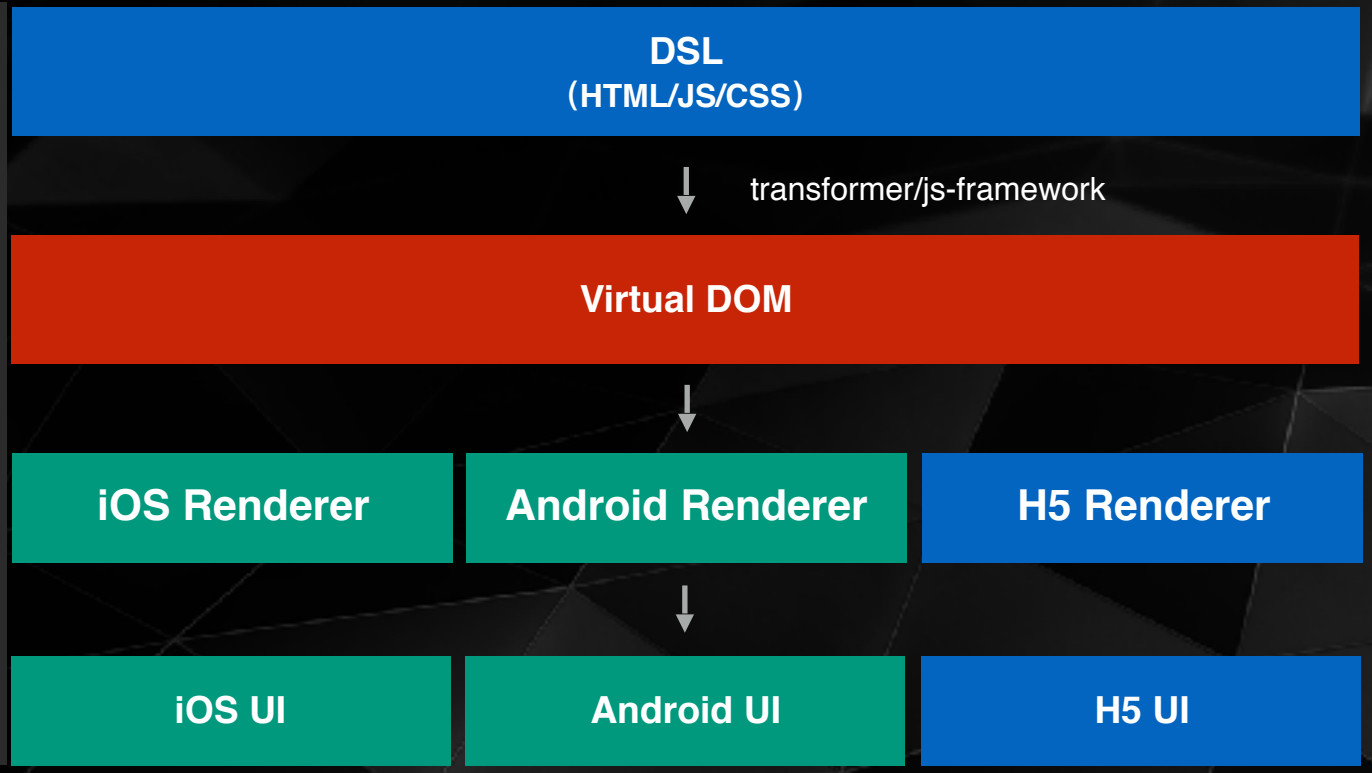
低端机性能基本解决，首屏加载秒开

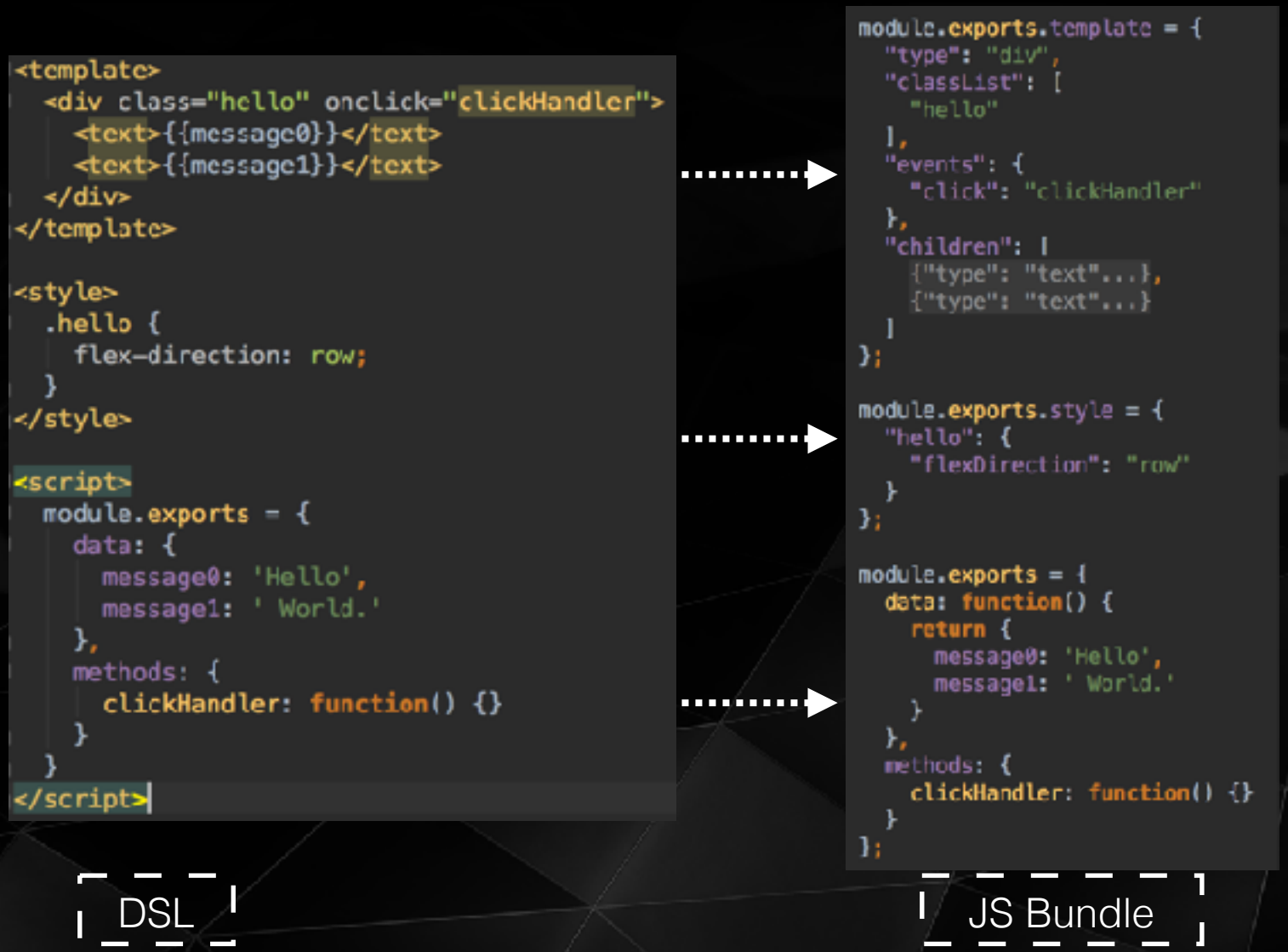
- write once, run everywhere
- 三端一致的开发体验和渲染效果
- web的开发效率，native的渲染性能

f (HTML/CSS/JS) = Native UI

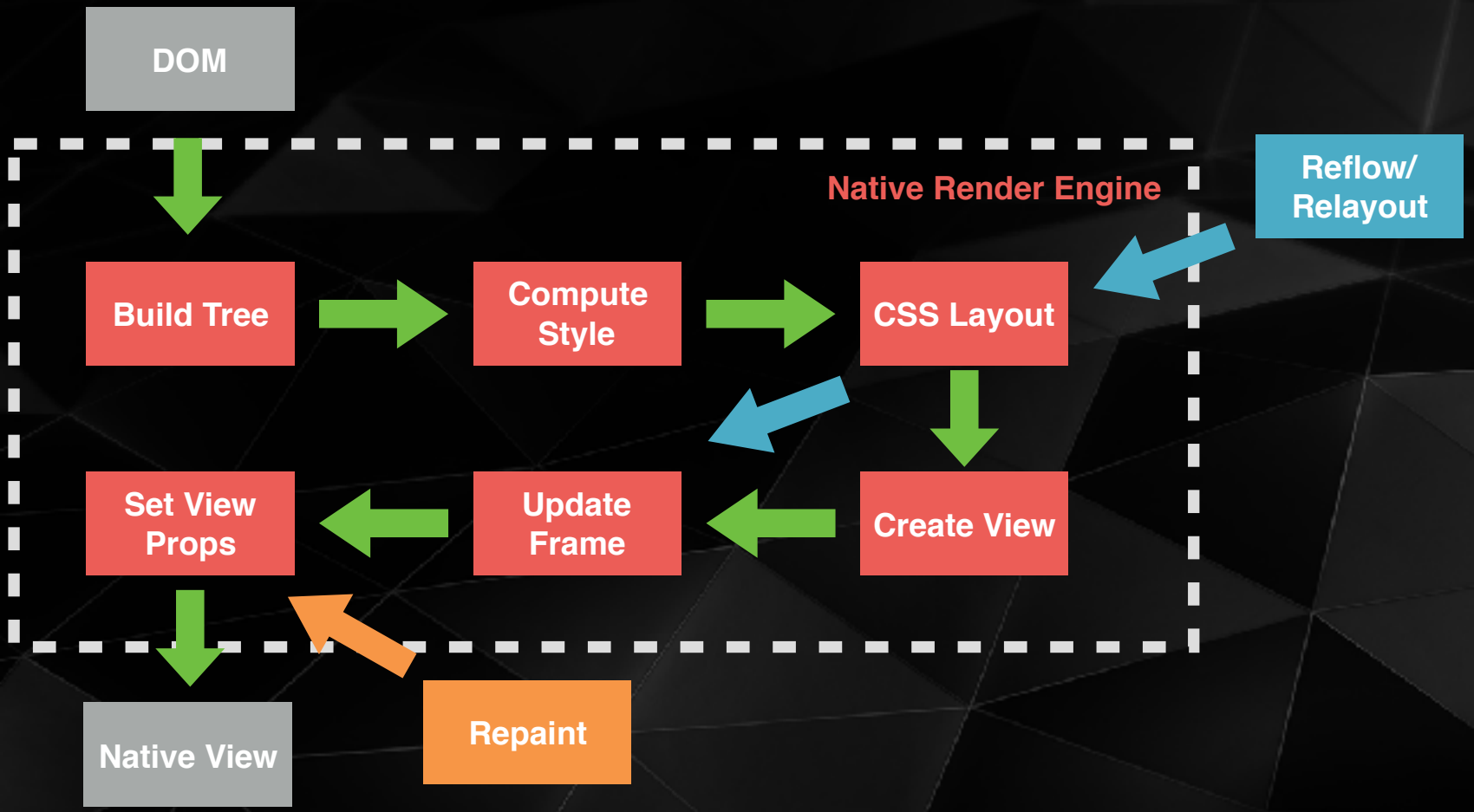
Alibaba Group 阿里巴巴集团

```
{
  ref: 1,
  type: 'div',
  attr: {append: 'tree'},
  event: ['click'],
  style: {
    backgroundColor: "#ffffff",
    flexDirection: 'column',
    fontSize: 28,
    paddingBottom: 0,
    paddingLeft: 24,
    paddingRight: 24,
    paddingTop: 24,
    position: 'relative'
  },
  children: [
    {
      ref: 2,
      type: 'image',
      attr: {},
      event: [],
      style: {
        position: 'absolute'
      },
      children: []
    }
  ]
}
```




DSL
JS Bundle

Weex Render Engine

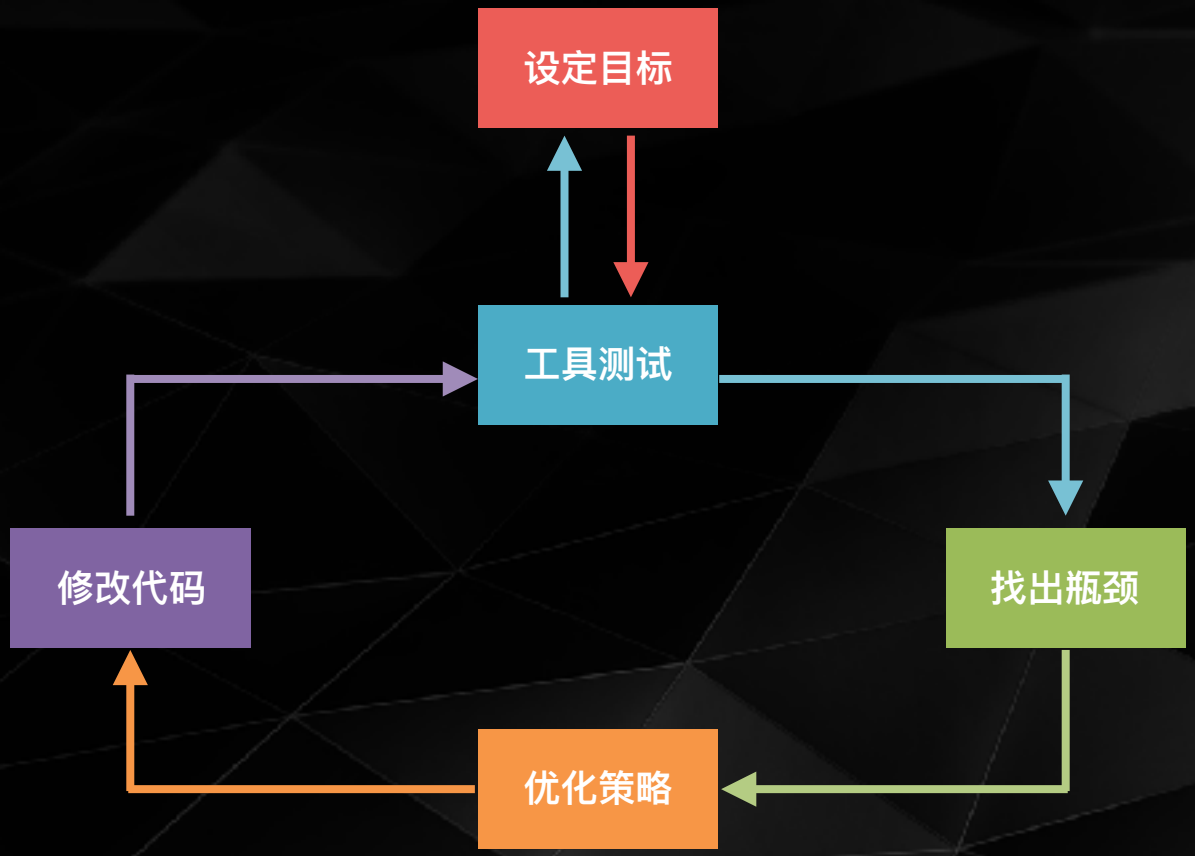


Write once, run anywhere

性能优化

提升用户体验

性能优化



Alibaba Group 阿里巴巴集团

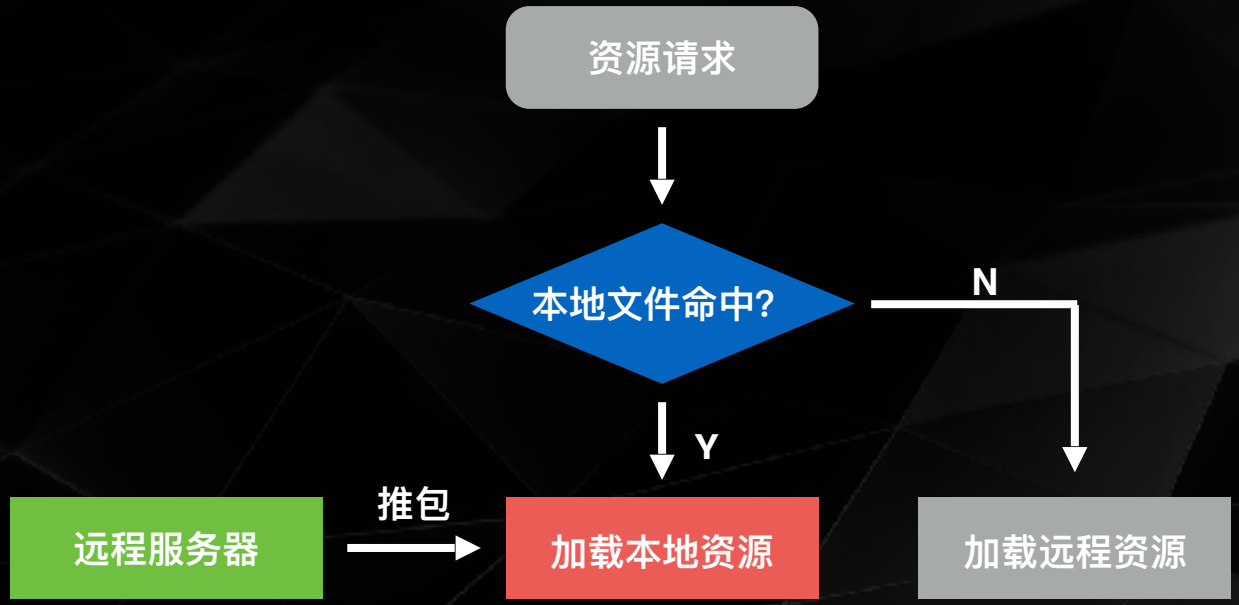
秒开

点击到首屏显示<1秒

Weex加载周期

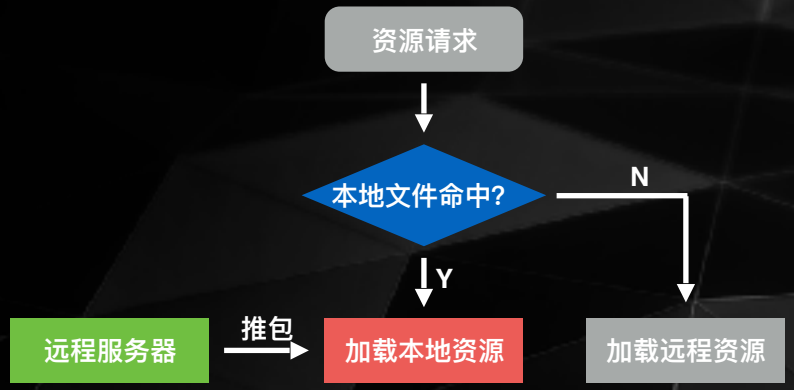
- 下载JS
- 执行JS
- 解析DOM
- 渲染UI

JS预加载

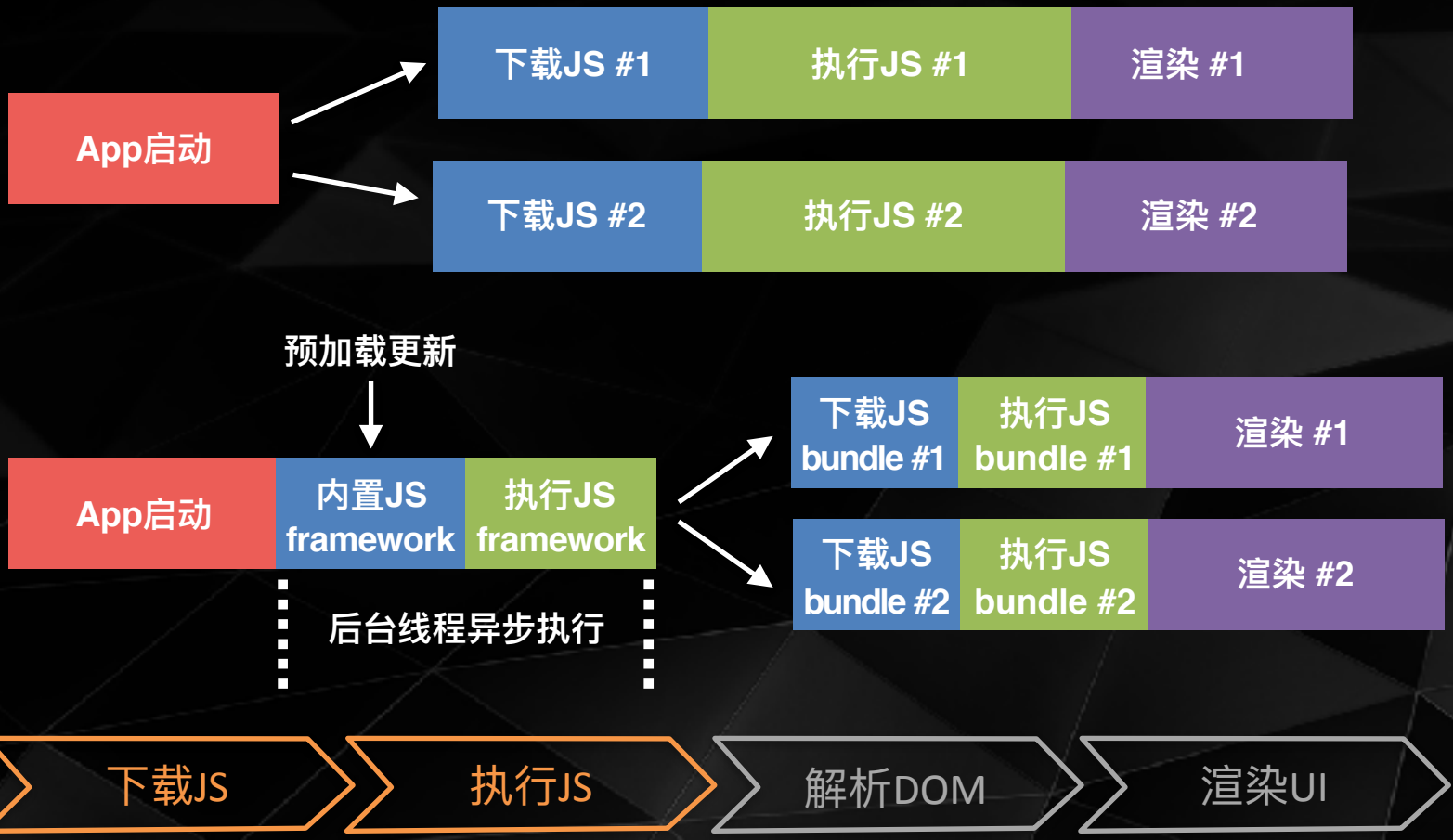


JS预加载

- App启动预先下载JS
- 通过长连通道push
- 全量/增量、被动/强制更新结合
- 400ms->10ms(iOS)
- 850ms->35ms(Android)

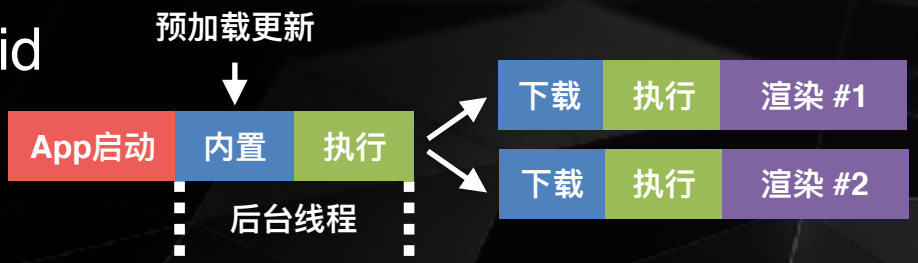


framework & bundle 分离



framework & bundle 分离

- js framework 36KB(gzip)
- js bundle 10-50KB(gzip)
- 异步初始化(iOS 170ms, Android 600ms)
- 单一Context的挑战



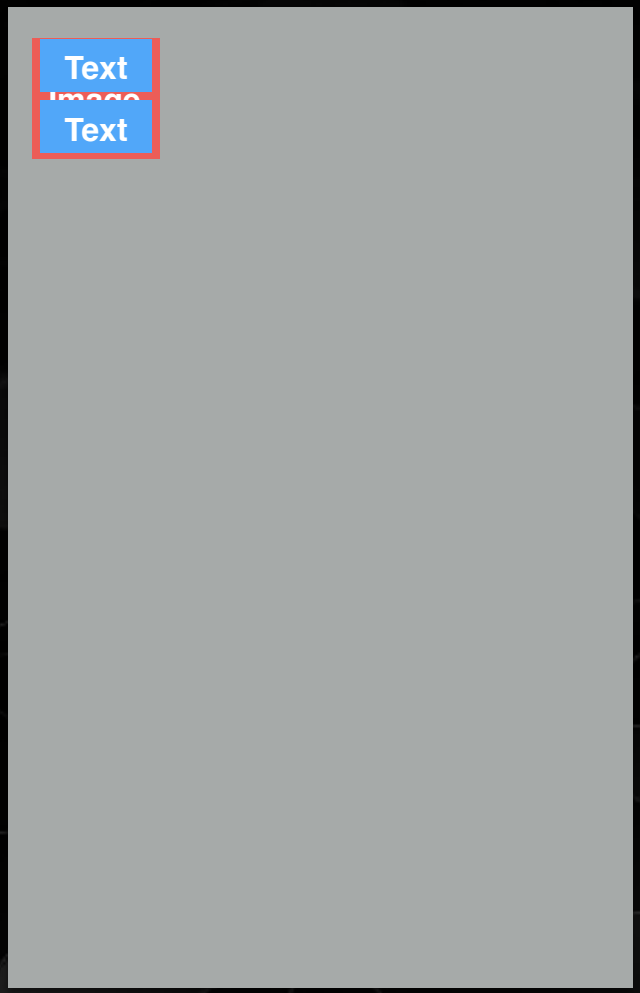
Node & Tree 渲染结合

- node: 最小颗粒度逐个渲染
- tree: 整棵树一起渲染

```
<template>
  <div>
    <image ...></image>
    <text>...</text>
  </div>
  <div append="tree">
    <image ...></image>
    <text>...</text>
  </div>
  <div append="node">
    <image ...></image>
    <text>...</text>
  </div>
</template>
```

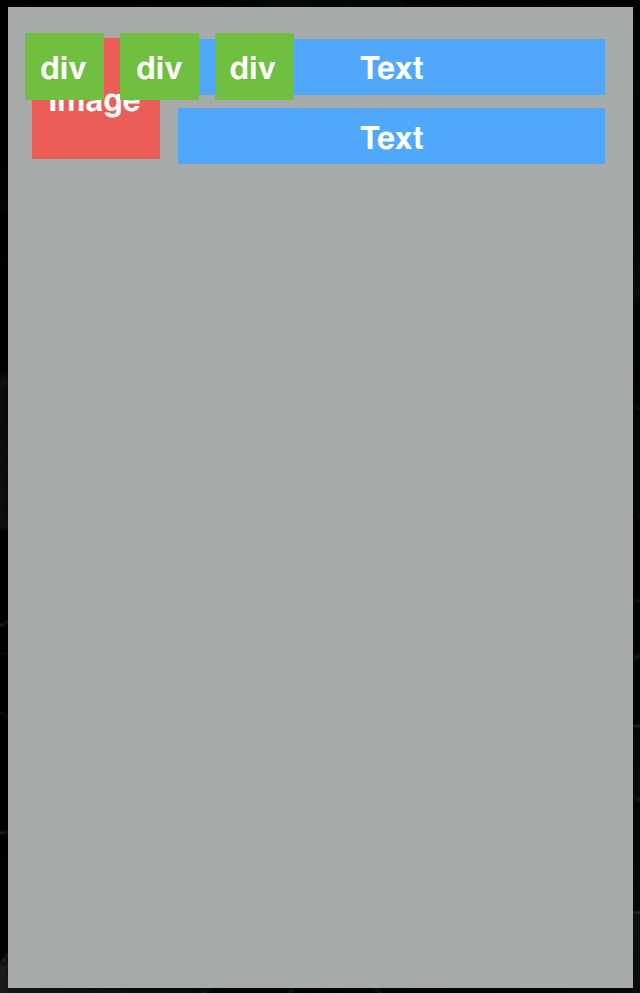
Alibaba Group 阿里巴巴集团

Node



Layout

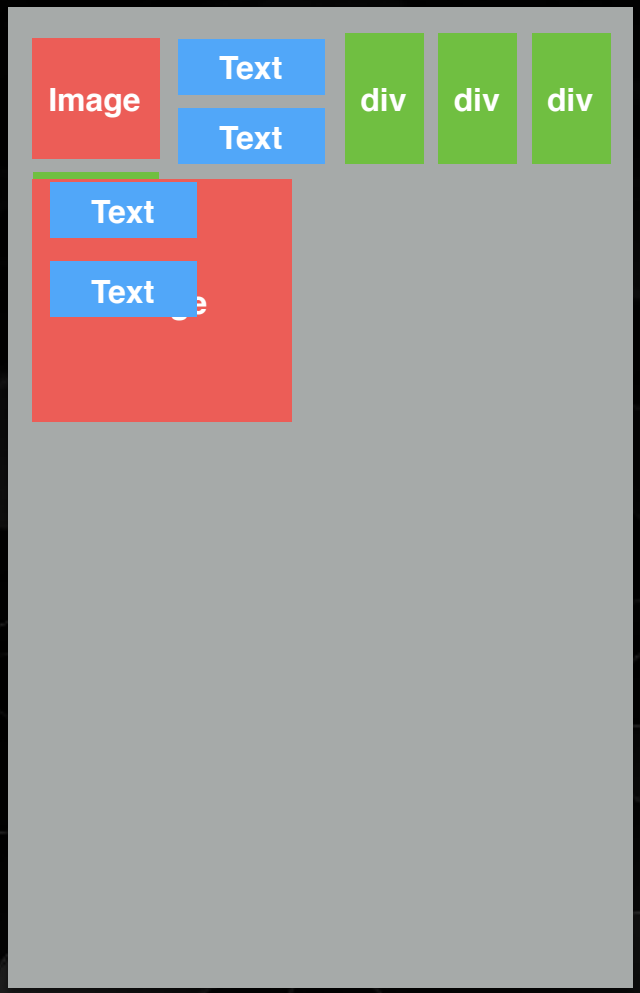
Node



Layout

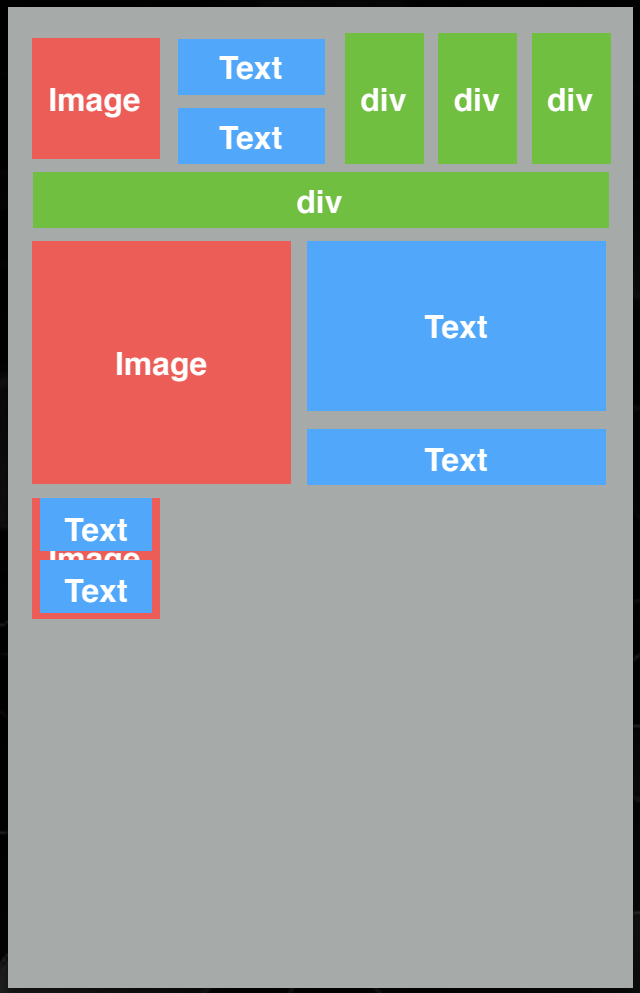
Alibaba Group 阿里巴巴集团

Node



Layout

Node



Layout

Node

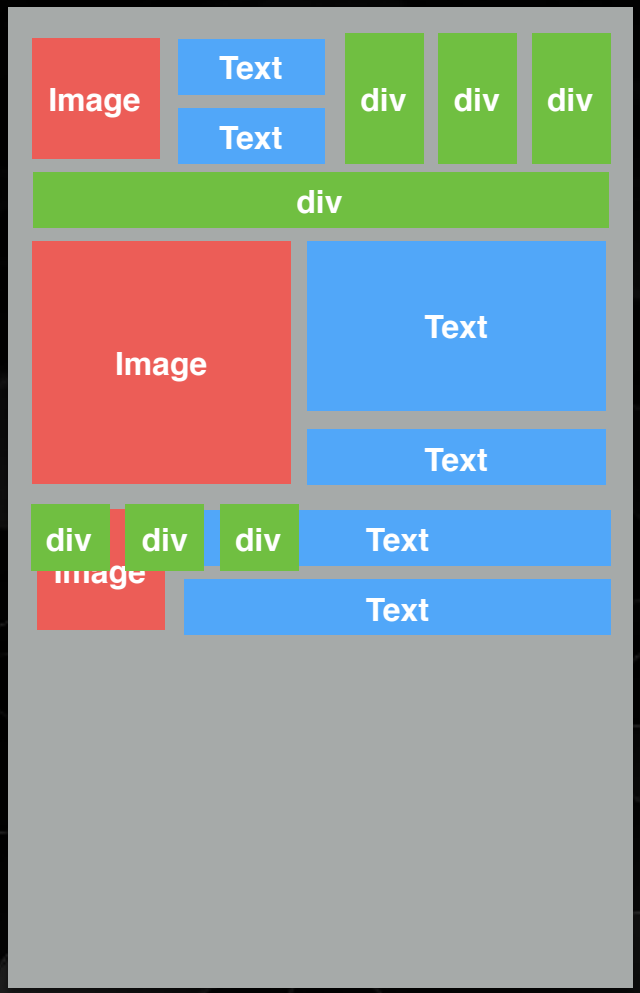


Image
Text
Text
div
div
div
div
Image
Text
Text
div
div
div
Text
Text

Layout

Node

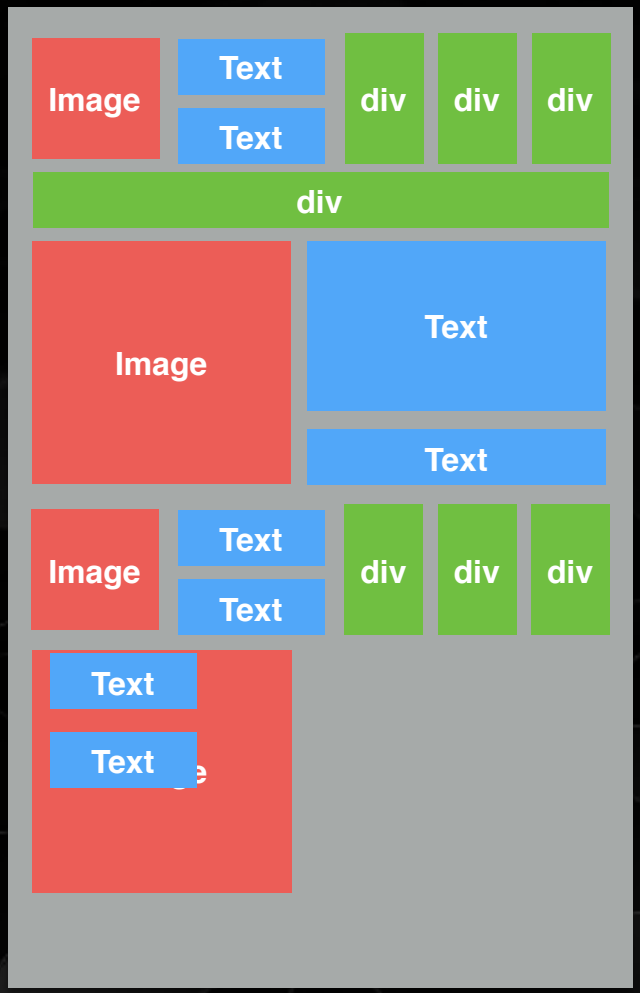


Image
Text
Text
div
div
div
div
Image
Text
Text
Image
Text
Text
div
div
div
Text
Text

Layout

Node

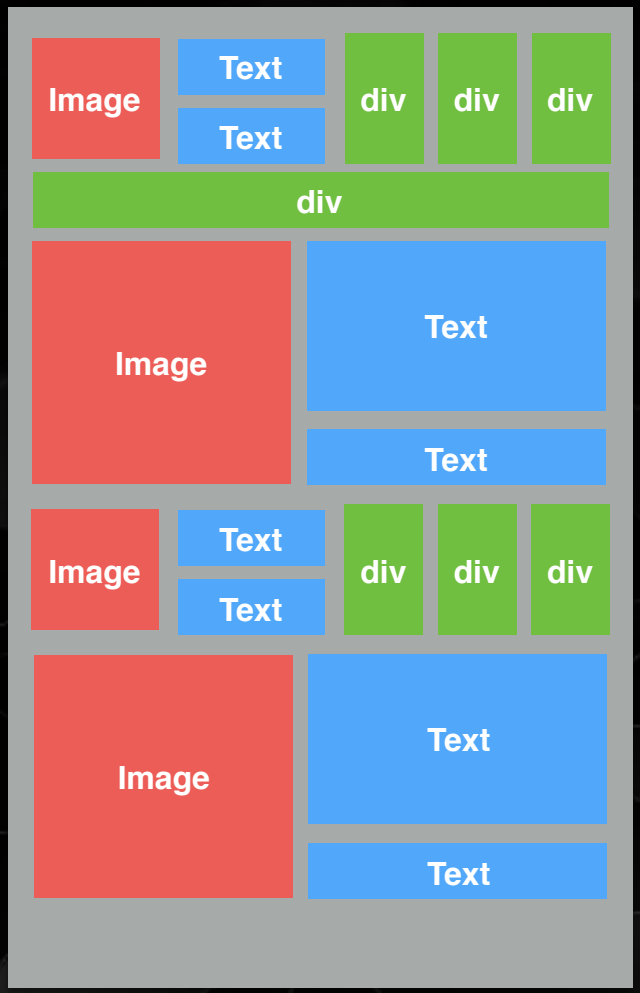
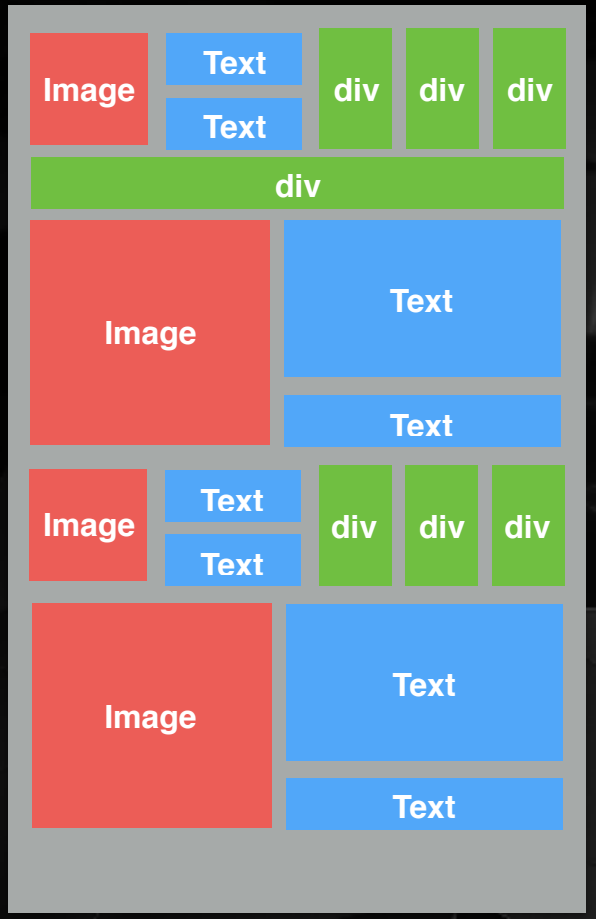


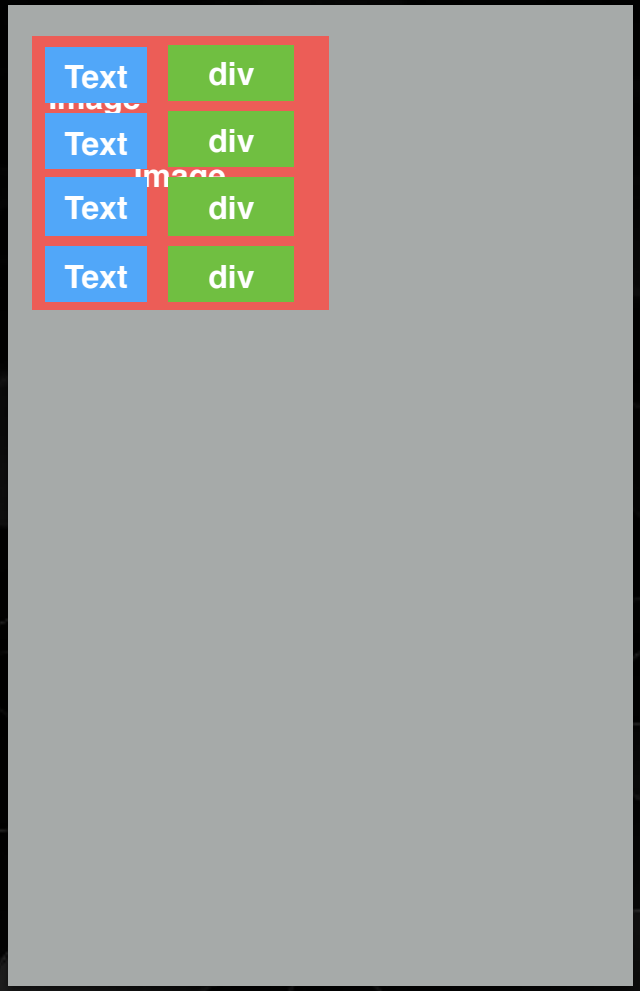
Image
Text
Text
div
div
div
div
Image
Text
Text
Image
Text
Text
div
div
div
Image
Text
Text

Node & Tree 渲染结合

- Node: 最小颗粒度逐个渲染
  - 解析完DOM后可以立刻显示
  - 保证不会长时间阻塞主线程
  - 可能会造成多次冗余Layout

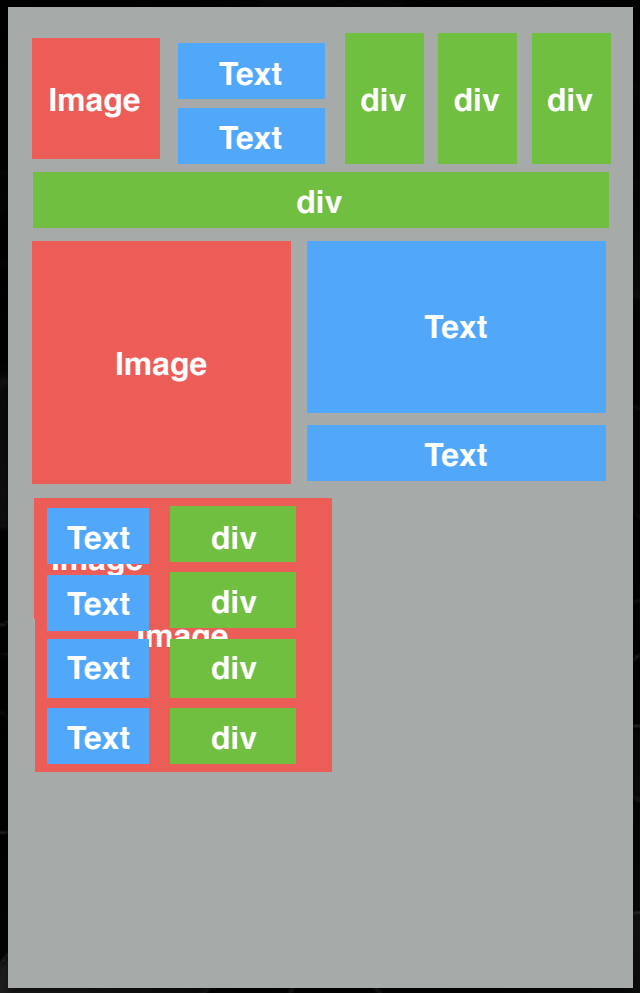


Tree



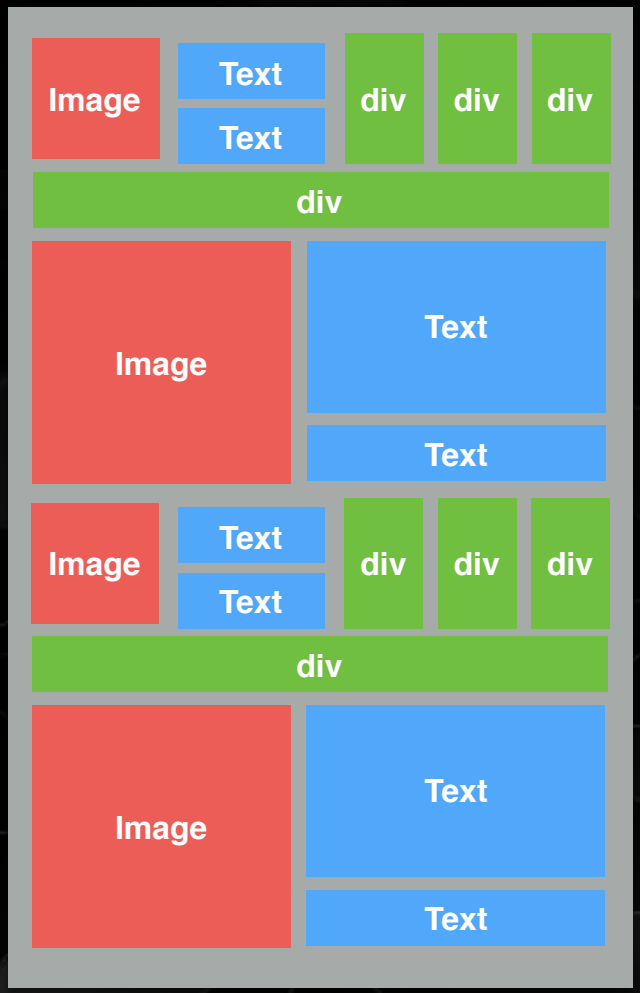
} append = “tree”

Tree



} append = “tree”

Tree

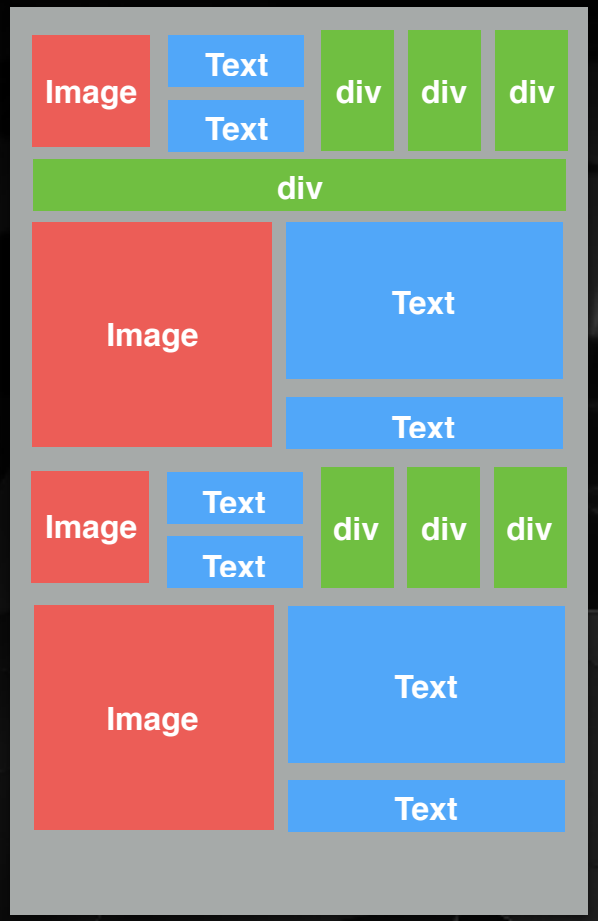


append = “tree”

append = “tree”

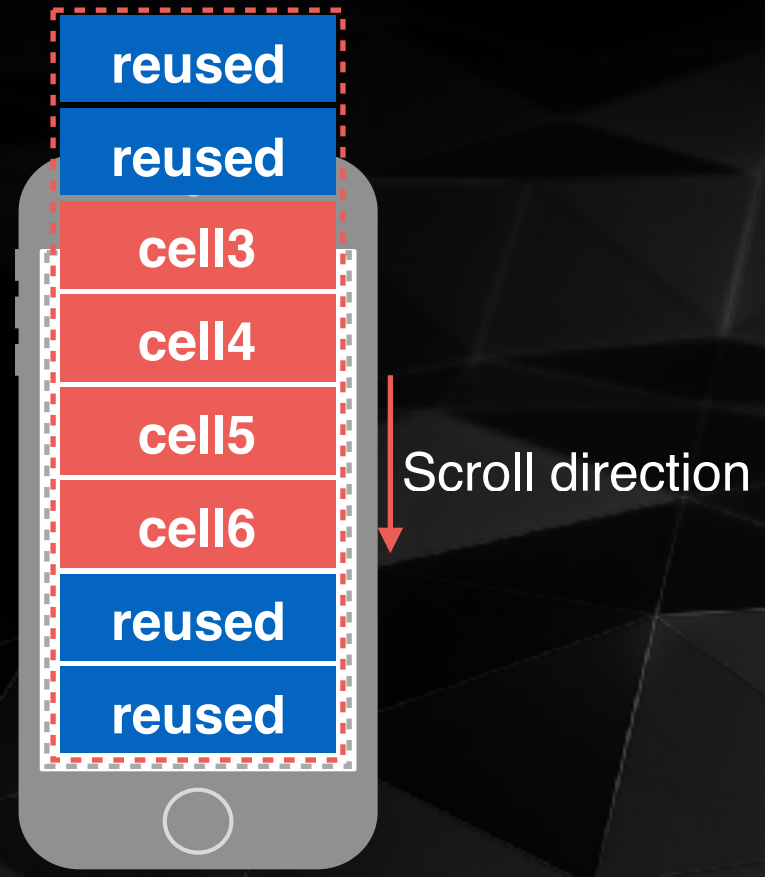
Node & Tree 渲染结合

- Node: 最小颗粒度渲染
  - 解析完DOM后可以立刻显示
  - 保证不会长时间阻塞主线程
  - 可能会造成多次冗余Layout
- Tree: 整棵树一起渲染
  - 只需布局一次, 渲染更高效
  - 可能会阻塞主线程

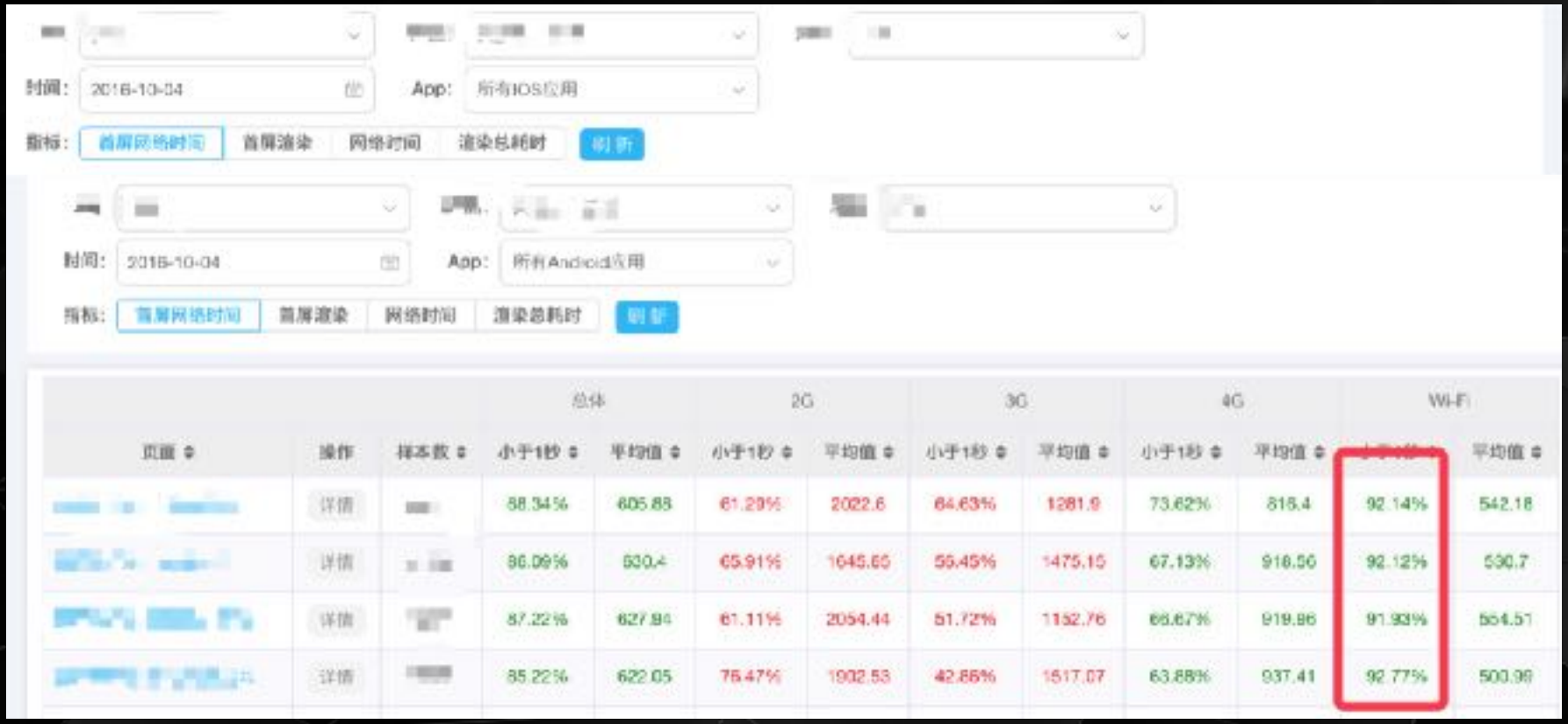


List

- 使用UITableView/RecyclerView
- 只渲染可见区域
- 内存复用
- 天然的cell之间互相隔离
- 原生的交互体验



秒开数据


时间: 2016-10-04
App: 所有iOS应用
指标: 首屏网络时间 首屏渲染 网络时间 渲染总耗时 刷新
时间: 2016-10-04
App: 所有Android应用
指标: 首屏网络时间 首屏渲染 网络时间 渲染总耗时 刷新
总体 2G 3G 4G Wi-Fi
页面 操作 样本数 小于1秒 平均值 小于1秒 平均值 小于1秒 平均值 小于1秒 平均值 小于1秒 平均值
详情 58.34% 605.88 61.29% 2022.6 64.63% 1281.9 73.62% 815.4 92.14% 542.18
详情 86.09% 630.4 65.91% 1645.65 59.45% 1475.15 67.13% 918.56 92.12% 530.7
详情 87.22% 627.84 61.11% 2054.44 51.72% 1152.76 66.67% 919.86 91.93% 554.51
详情 85.22% 622.05 76.47% 1902.53 42.86% 1517.07 63.88% 937.41 92.77% 503.99


电脑桌
SONOS
无线智能音响
100元预定 抢 430元优惠
聚
进口
超市
充
旅行
到家
淘抢购
有好货
爱逛街
必买清单

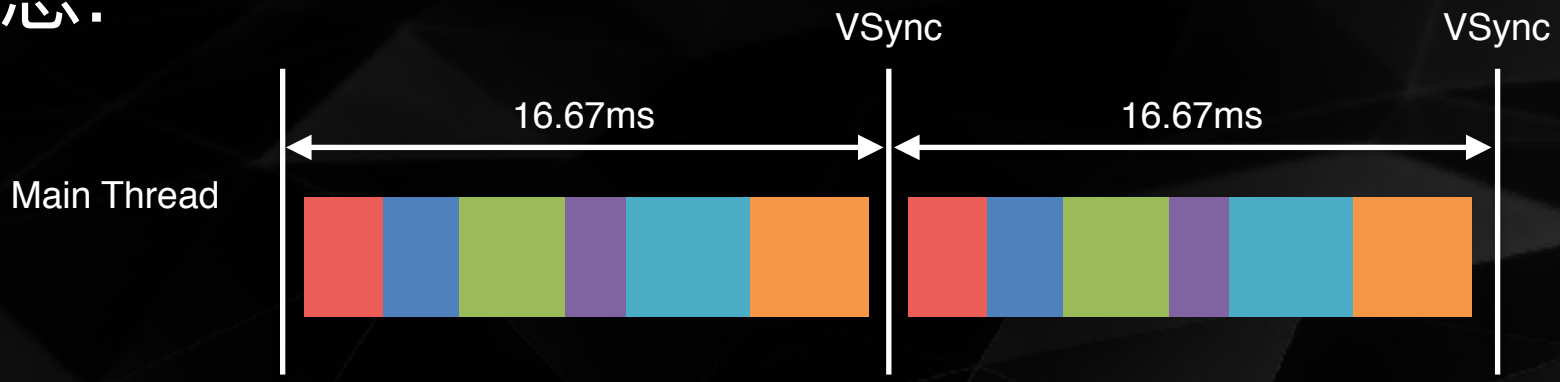
Alibaba Group 阿里巴巴集团

流畅

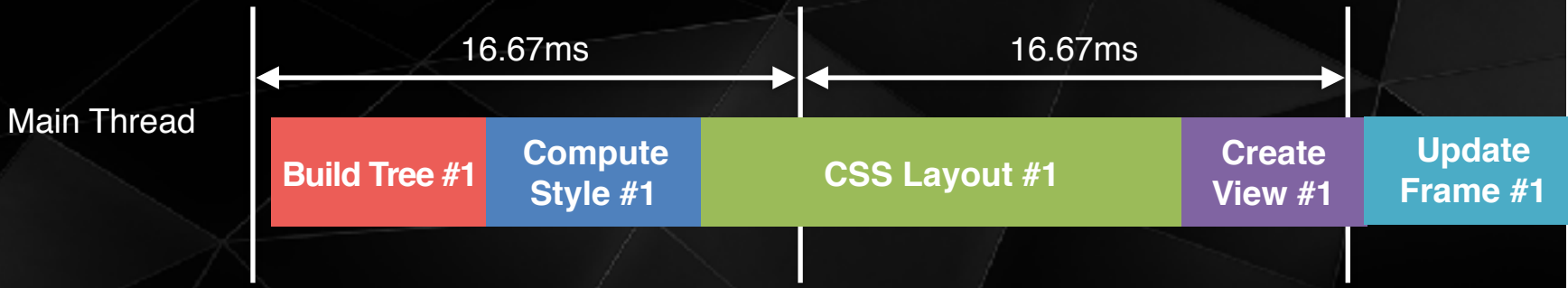
60FPS


Build Tree
Compute Style
CSS Layout
Create View
Update Frame
Set View Props

理想:


VSync
VSync
16.67ms
16.67ms
Main Thread

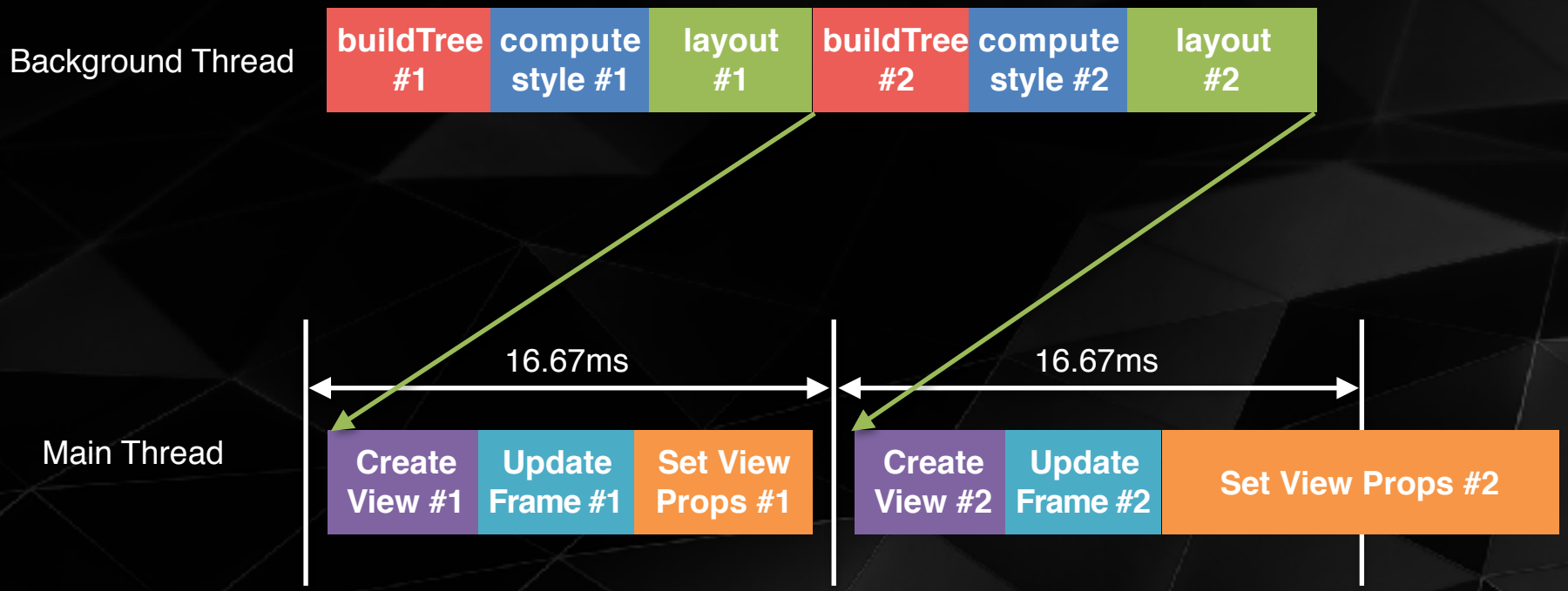
现实:


16.67ms
16.67ms
Main Thread
Build Tree #1
Compute Style #1
CSS Layout #1
Create View #1
Update Frame #1

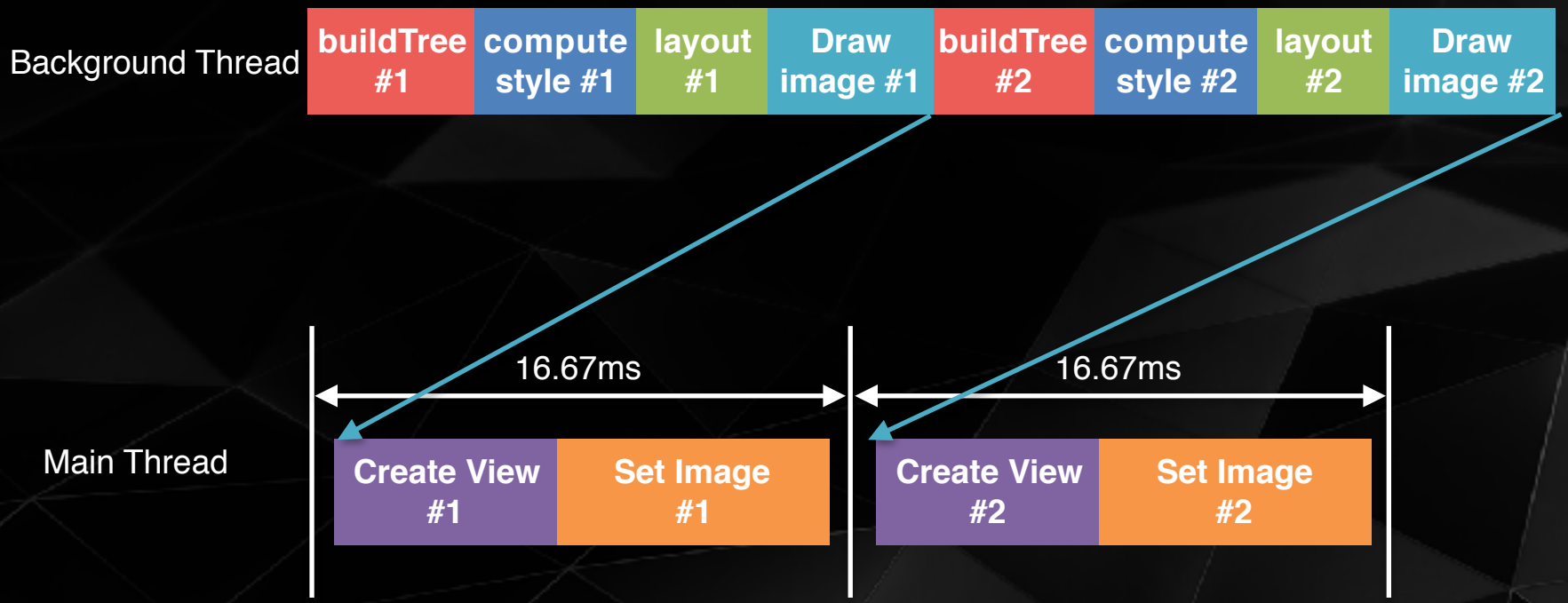
Weex帧率优化

- 60FPS = 一次Cell渲染主线程 <16ms
- 主线程同步 = 卡顿
- 想尽办法让渲染脱离主线程

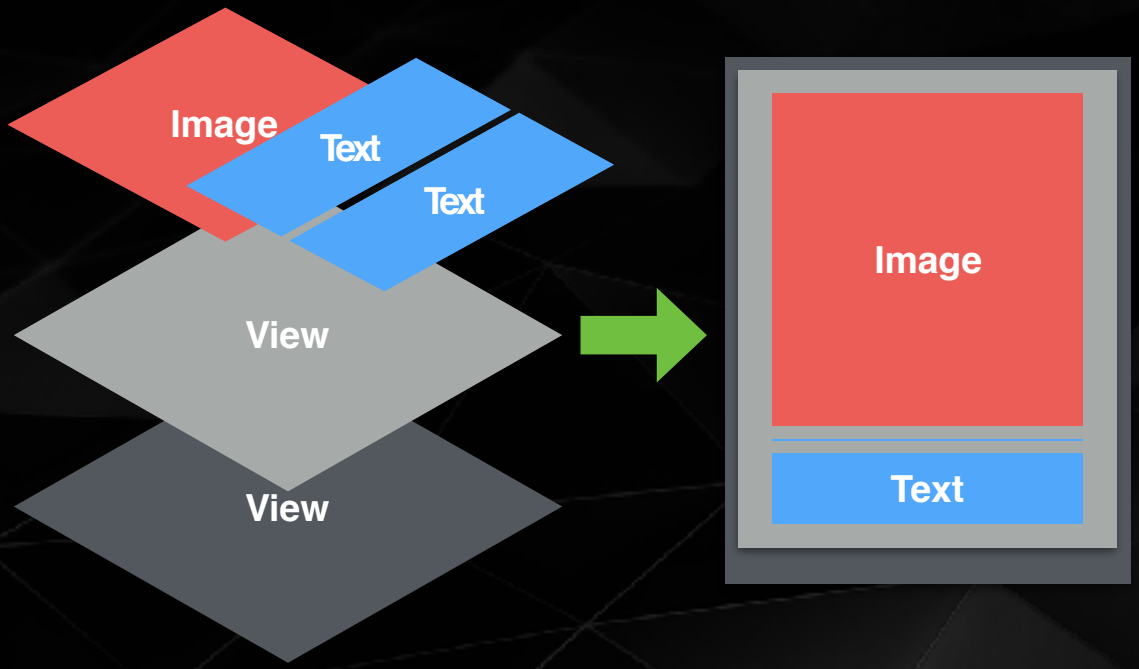
异步Layout

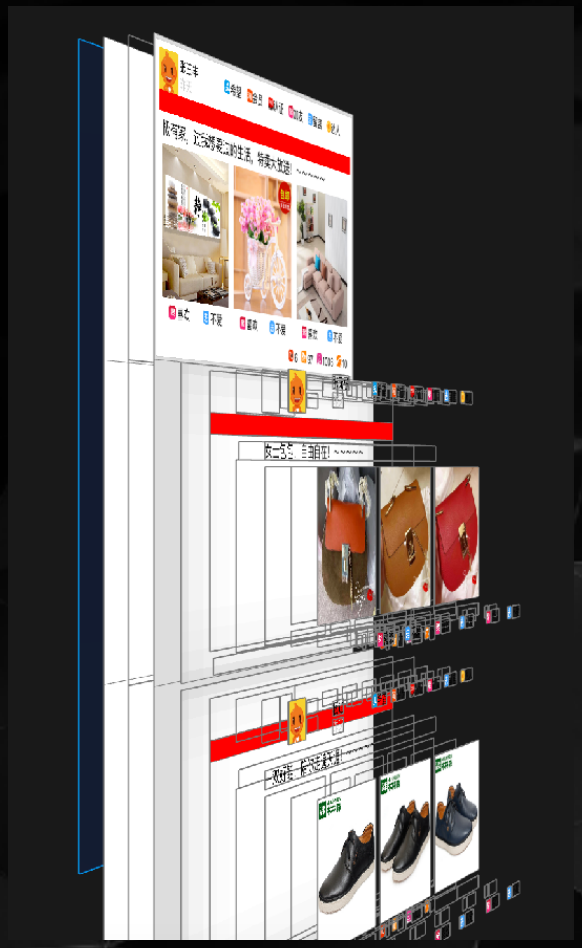

Background Thread
buildTree #1
compute style #1
layout #1
buildTree #2
compute style #2
layout #2
16.67ms
16.67ms
Main Thread
Create View #1
Update Frame #1
Set View Props #1
Create View #2
Update Frame #2
Set View Props #2

异步绘制


Background Thread
buildTree #1
compute style #1
layout #1
Draw image #1
buildTree #2
compute style #2
layout #2
Draw image #2
16.67ms
16.67ms
Main Thread
Create View #1
Set Image #1
Create View #2
Set Image #2

Compositing(Experiment)





Alibaba Group 阿里巴巴集团

节源

降低内存、CPU开销

JSCore内存回收机制

- JSCore 使用 mark-and-sweep GC
- GC 时机
  - 系统剩余内存较低(至少足够执行一次GC)
  - 内存碎片化严重
  - 刚分配了一块较大内存
  - 主动调用
- 循环引用不造成内存泄露

mark-and-sweep GC

```
void mark(block)
{
    if (!isMarked(block)) // 查询标记位
    {
        setMarked(block); // 标记

        // 递归往下标记
        Array *children = block->children();
        for(int i = 0; i < children.count(); ++i)
        {
            setMarked(children[i]);
        }
    }
}

void Sweep()
{
    forEachBlock({ // 遍历所有内存 block
       if (!isMarked(block))
            free(block); // 释放内存
        else
            clearMarked(block); // 重置标记位
    });
}
```

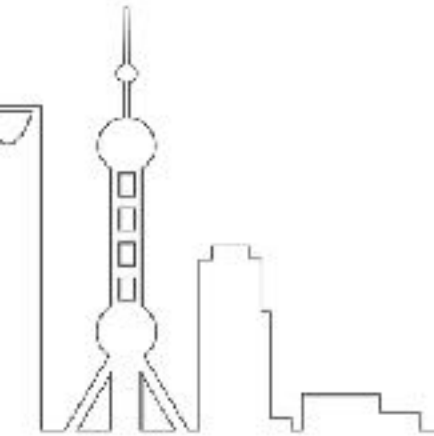
Weex JSCore 内存排查工具

- Xcode debugger/Allocation/Leak
- JSSynchronousGarbageCollectForDebugging
- JSC_logGC
- Chrome Devtool

未来的Weex

- 在秒开、流畅、节源的目标下持续优化
- 精简、高性能、移动定制化的WeexCore
- 充分利用多线程， 性能真正接近Native

Alibaba Group
阿里巴巴集团



Thanks!

International Software Development Conference

主办方 Geekbang. 极客邦科技 InfoQ