

---

# 前端性能监控平台

推啊-康旭峰

- 为什么需要前端性能监控
- 推啊前端性能监控平台实践及思考
- 性能监控平台如何助力优化产品



# 页面加载够快吗?

老板：你们的页面有点慢？🤔

: 你们页面太慢了!

我觉得还好啊 

: 打开页面要多少时间?

emmm.. 大概两秒吧 

: 我感觉要 5 秒啊, 得优化!

# 为什么差异这么大？

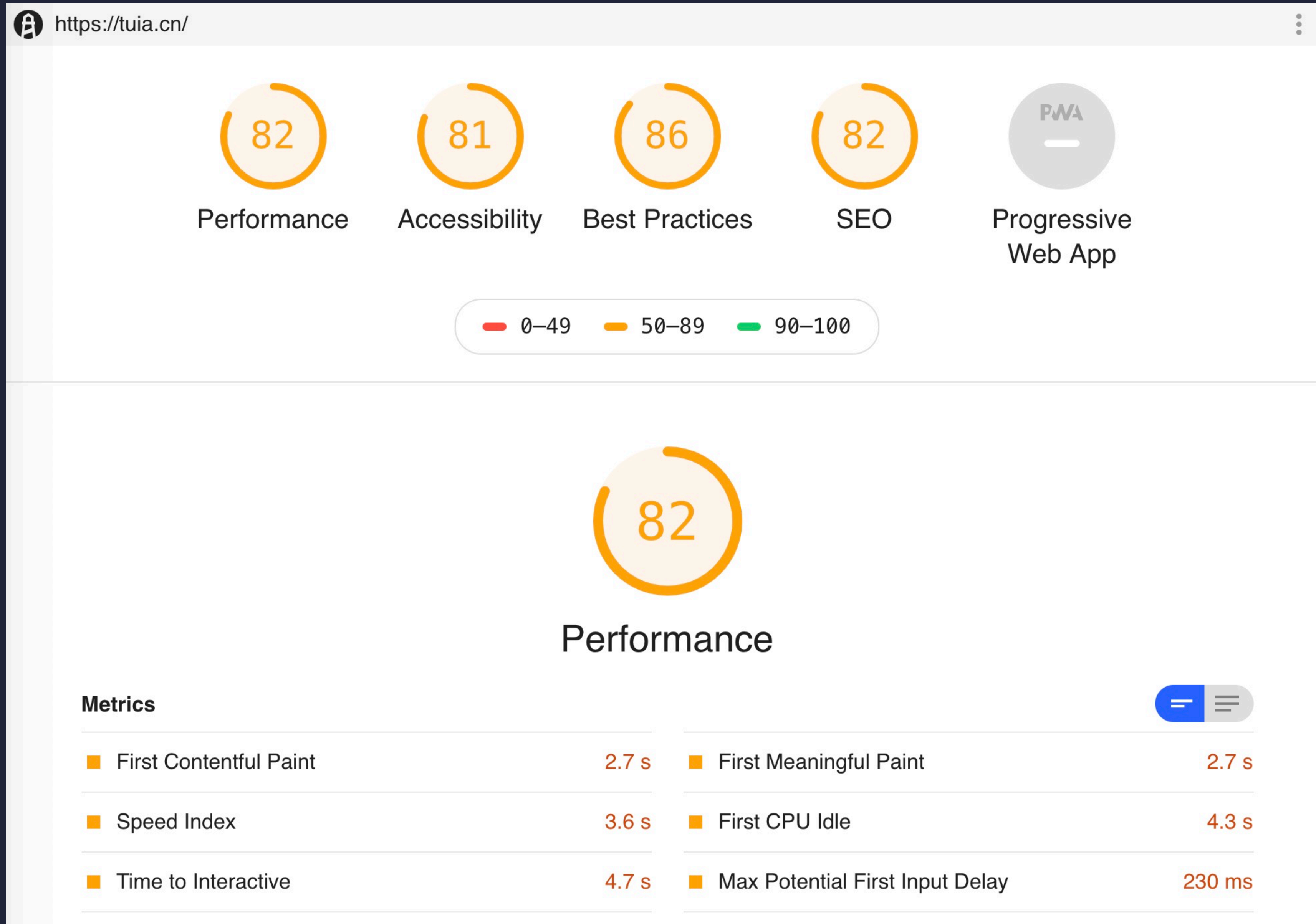
★ 影响因素：

网络、设备、用户主观因素 等等

★ 需要一个标准、一个平台，去衡量页面性能。

# 合成监控





使用lighthouse对 [www.tuia.cn](https://www.tuia.cn) 进行合成性能测试

# 真实用户监控



<https://www.tuia.cn>



/report



监控平台



用户真实访问

提取性能指标

数据清洗加工

性能分析监控



# 前端性能监控方案对比

合成监控：成本低，数据量小

适合**定性分析**，比如发布前的自动化测试

真实监控：成本较高，数据量大

适合**定量分析**，可结合业务数据做深入挖掘

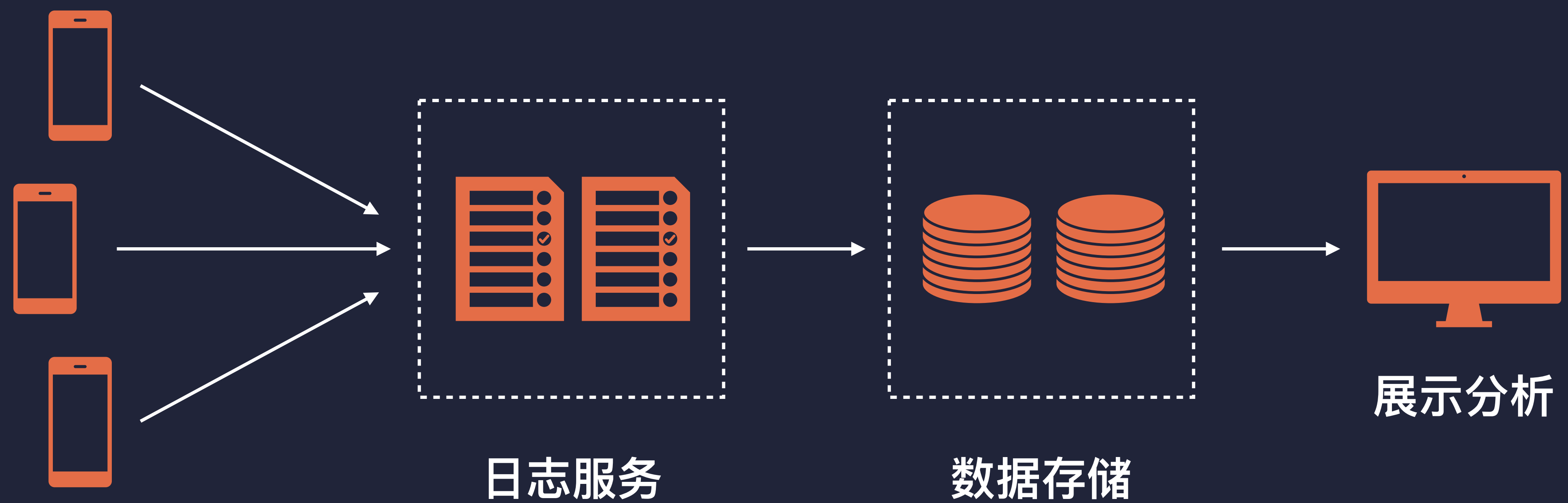
性能监控：

各维度数据量化，去衡量真实用户的加载速度

- 为什么需要前端性能监控
- 推啊前端性能监控平台实践及思考
- 性能监控平台如何助力优化产品

# 性能监控平台 Hunter

# 监控平台链路



采集上报

日志服务

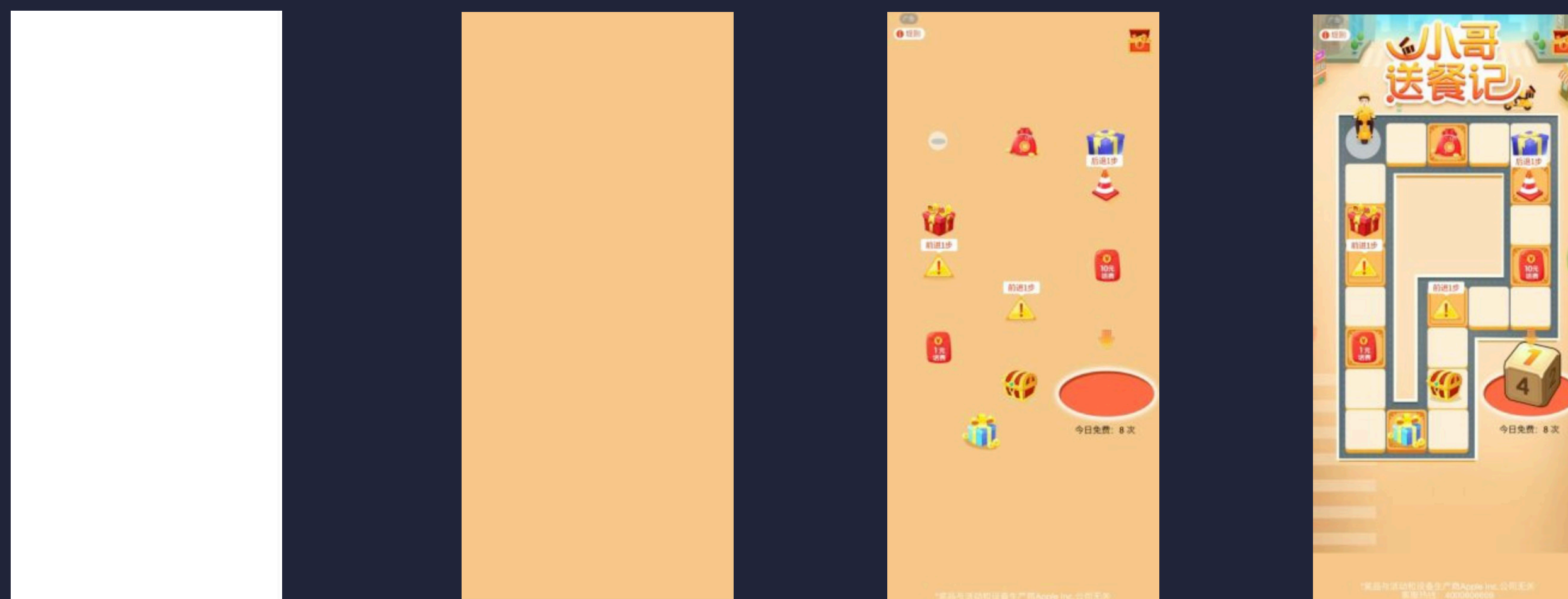
数据存储

展示分析

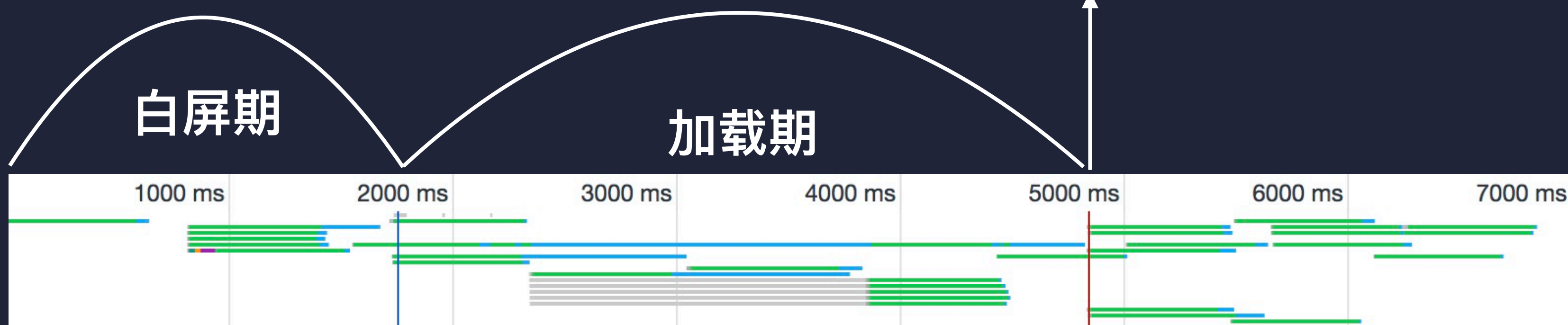
# 采集上报

- 定义性能指标
- 采集准确的时间
- 上报关联纬度

# 定义性能指标

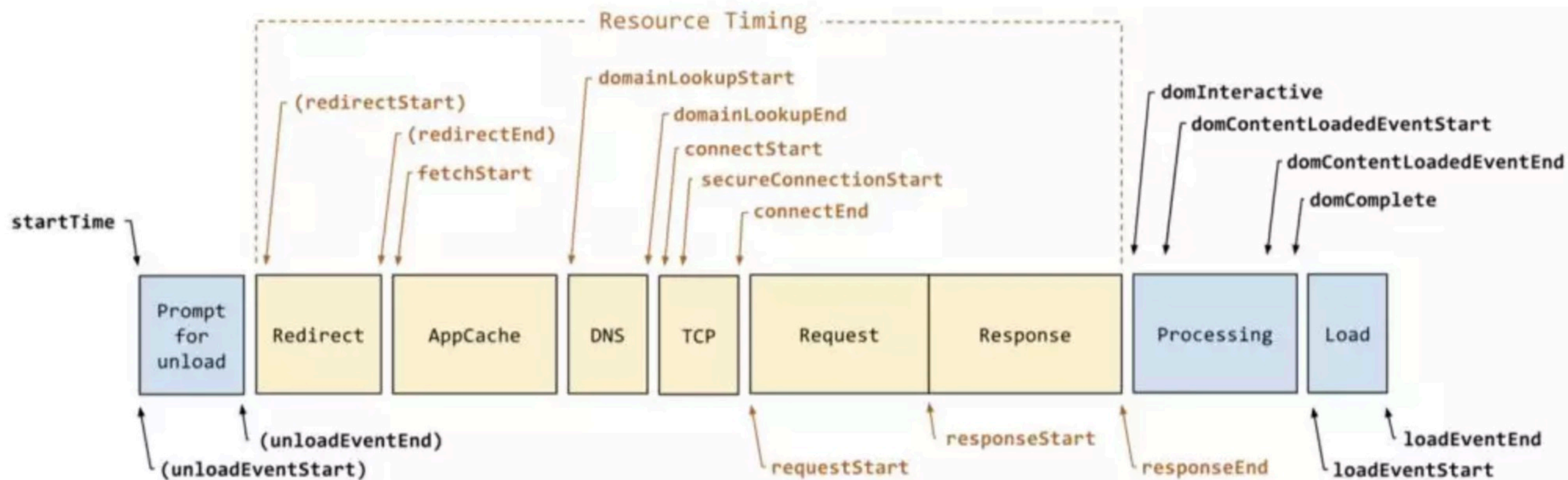


页面加载完成

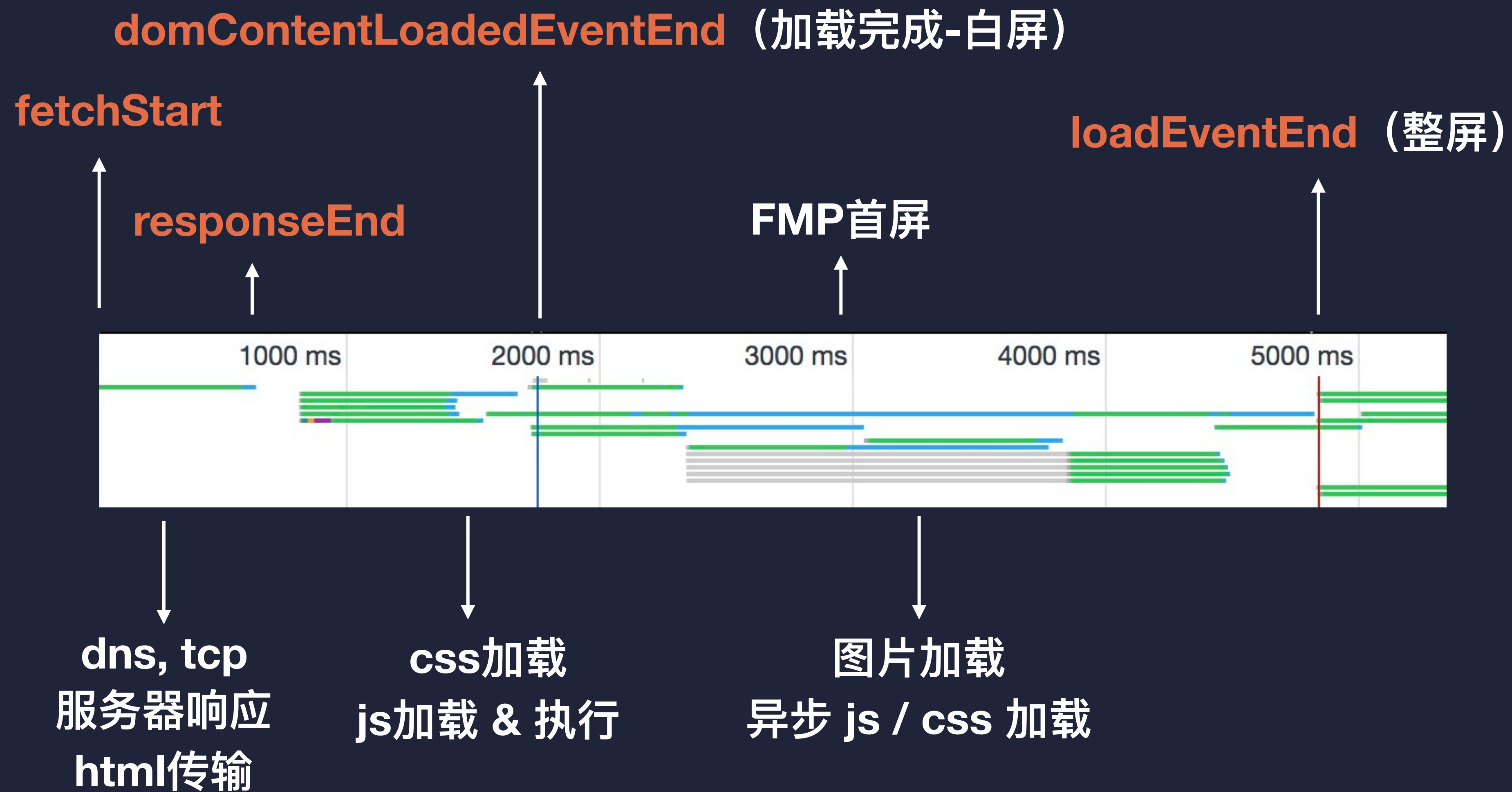




# 采集准确的时间 —— performance.timing

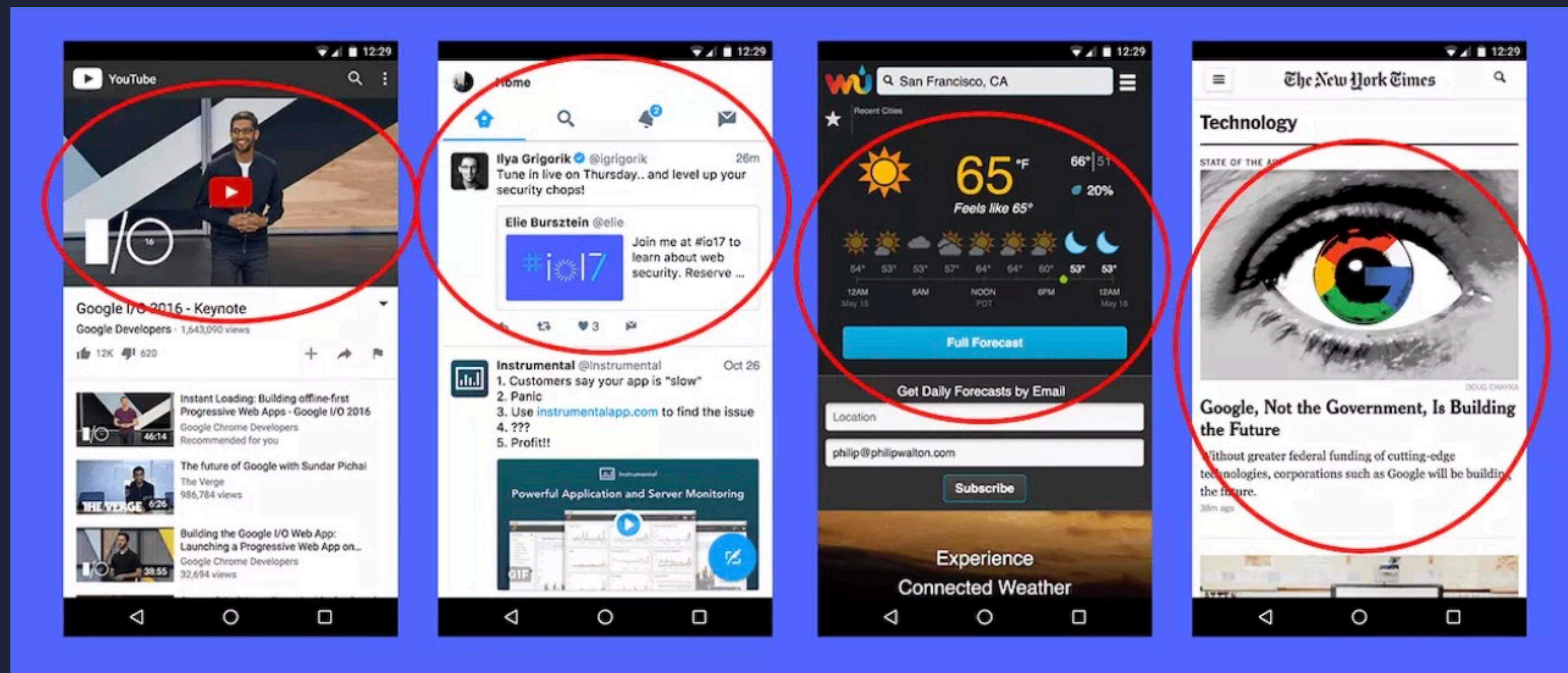


# 采集准确的时间 —— 关键时间点



# 采集准确的时间 —— 首屏加载时间

## FMP ( First Meaning Paint )



最重要区块 \* 最慢元素加载时长

# 采集准确的时间 —— 首屏加载时间

FMP算法：最重要区块 - 定义



```
<body>
  <div id="header">
    <div class="logo"></div>
    <div class="nav"></div>
  </div>
  <div id="content">
    <div class="banner">
      
    </div>
  </div>
  <div id="footer"> ... </div>
</body>
```



# 采集准确的时间 —— 首屏加载时间

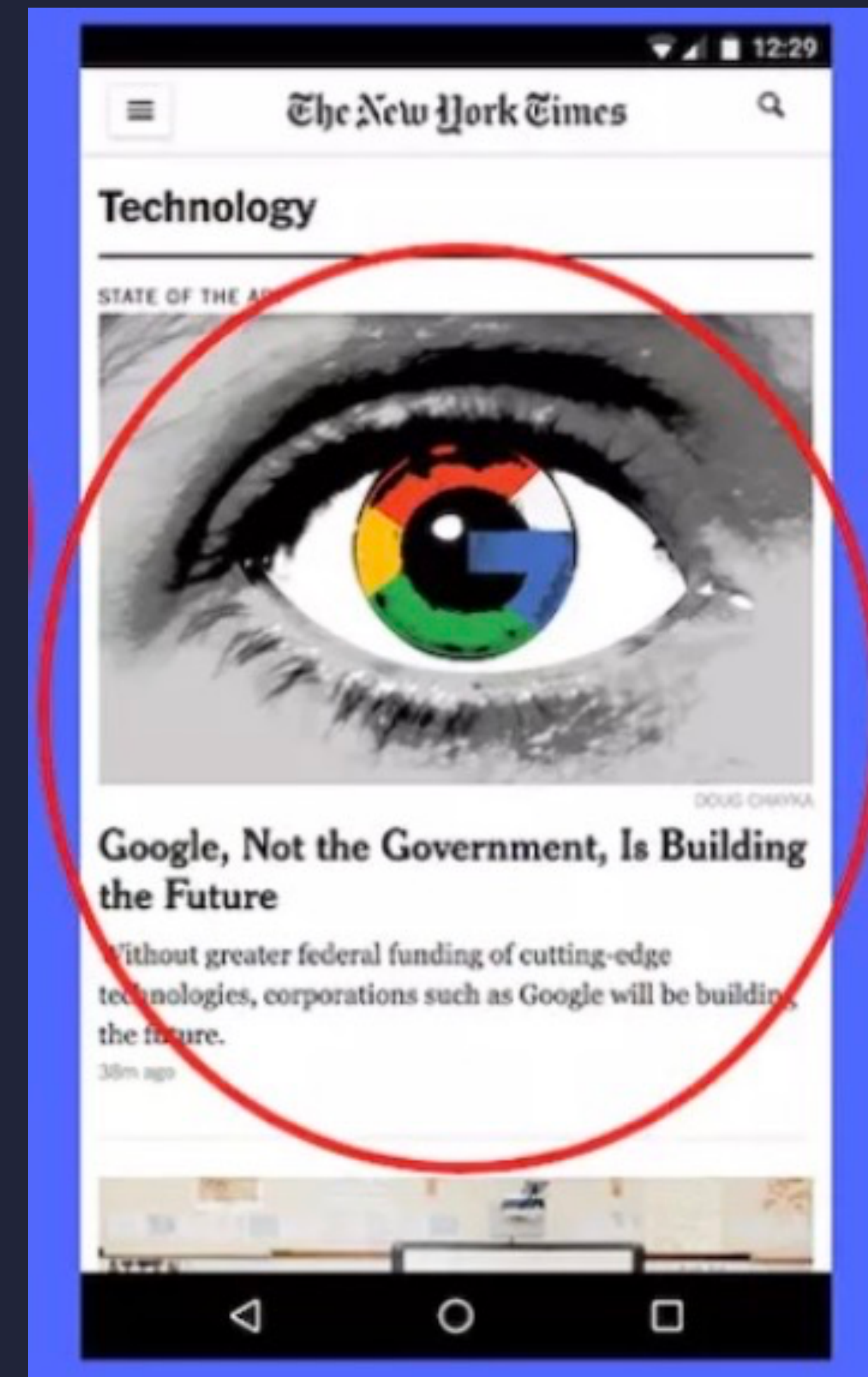
FMP算法：最重要区块 - 计算

面积：元素区块在首屏范围面积

权重：div: 1分, img: 2分, ...

得分：面积 \* 权重

★ DFS：深度优先遍历



# 采集准确的时间 —— 首屏加载时间

FMP算法：最慢元素加载时长

DOM: MutationObserver

资源: `performance.getEntries()`



# 采集准确的时间 —— 首屏加载时间

## FMP算法存在的问题

- 算法的权重，不一定是业务中真正最重要的元素
- 非常规页面布局问题
- 计算性能消耗

# 采集准确的时间 —— 首屏加载时间

有万金油方案吗？

追求准确，指定监听元素

推啊游戏活动统计方案  $fmp \approx window.onload$

★ 灵活应变



# 上报关联纬度

常规纬度

时间

页面

浏览器

机型

系统

业务纬度

媒体ID

广告位ID

用户ID

...

专业纬度

页面是否可见

页面加载方式

是否启用http2

...

采集上报

日志服务

数据存储

展示分析

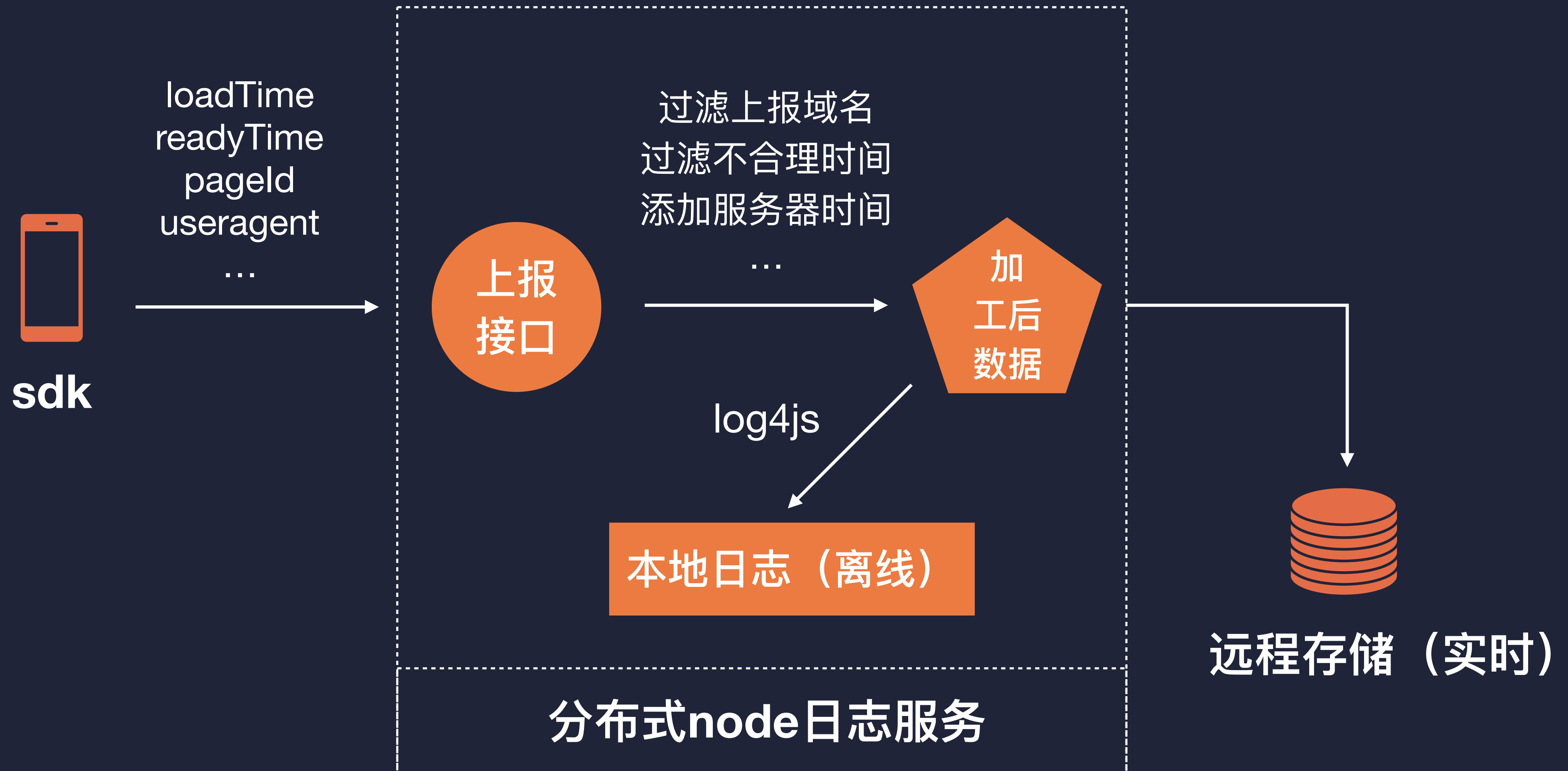
# 日志服务 —— 核心功能

- 接收上报信息
- 数据清洗加工
- 入库

# 日志服务 —— 分布式架构



# 日志服务 —— 接收 & 清洗 & 入库



采集上报

日志服务

数据存储

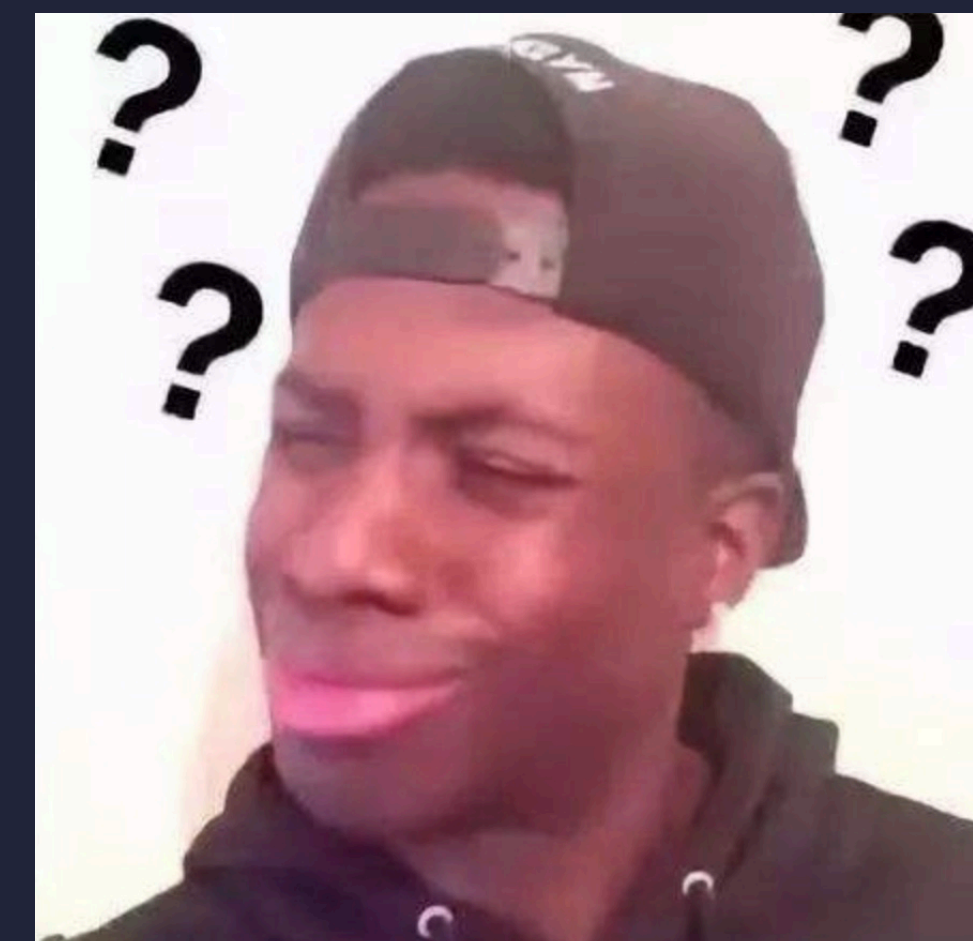
展示分析

## 数据存储 —— 技术选型

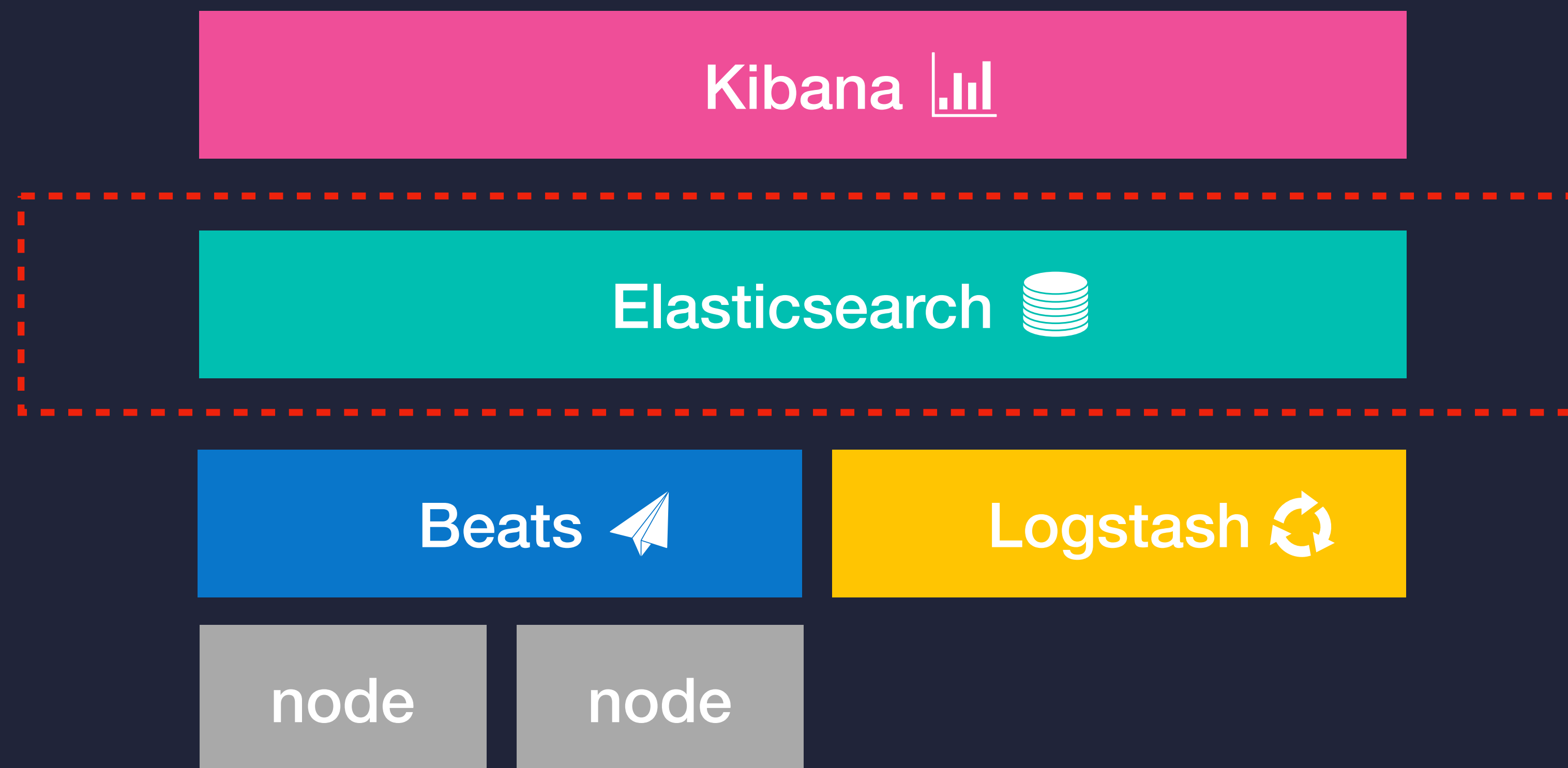
- Mysql ?
- MongoDB ?



- 日5000万数据存储
- 近实时分析线上数据



# 数据存储 —— ELK Stack



大规模日志实时处理系统



# 数据存储 —— Elasticsearch

- 支持PB级数据、分布式
- 高性能实时检索、聚合分析数据
- 支持 Node.js 读写, Restful 风格接口

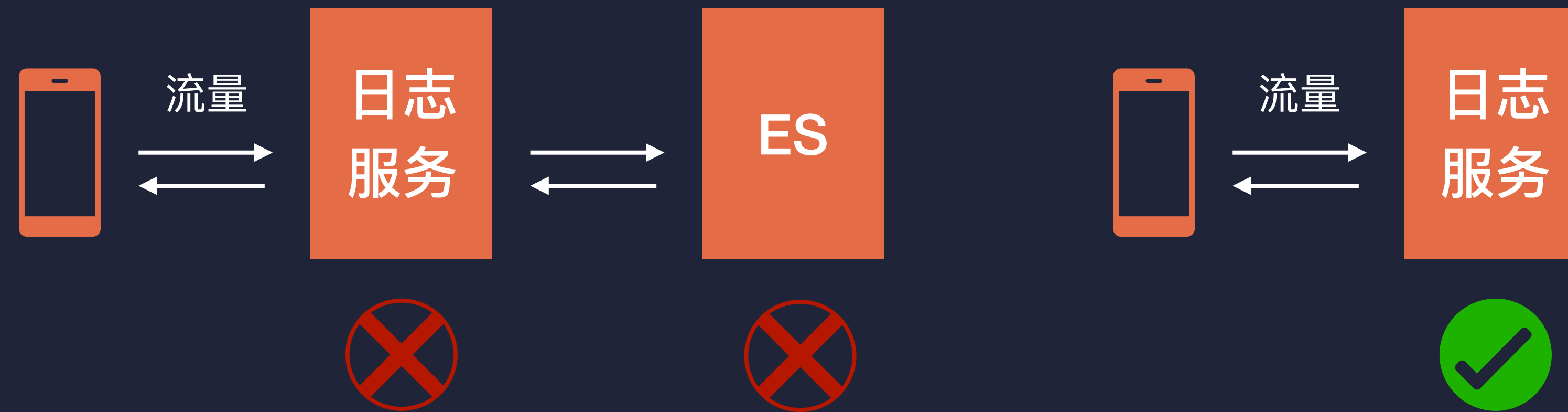
# 系统服务性能问题 —— 扛不住的压测

Node: 1核1G \* 2      ES: 1核4G \* 3

上报接口高峰期预计 1500 QPS, 压测到 500 QPS 时:

- node服务大量500错误
- es服务cpu飙升90%+

# 系统服务性能问题 —— ES使用姿势



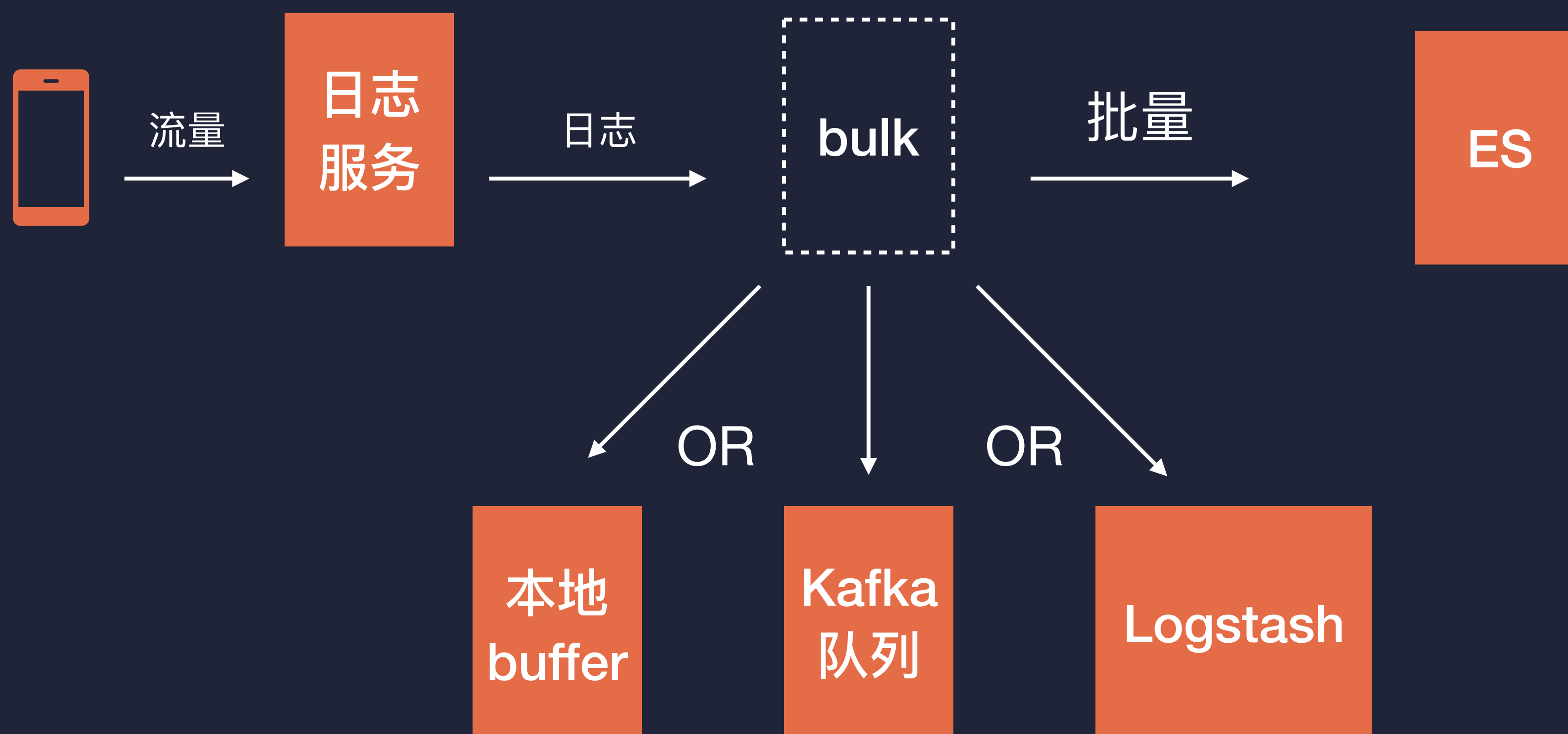
问题出在 ES:

1. 写入频率
2. 刷新频率
3. Node异步写



1. ES批写入 (bulk 10 -> 1000)
2. ES刷新频率 (1s -> 60s)

# 系统服务性能问题 —— Node



## 系统服务性能问题 —— 现状



Node: 1核1G \* 4      ES: 1核4G \* 3

QPS: 1700+, 日pv5000万+, 性能稳定

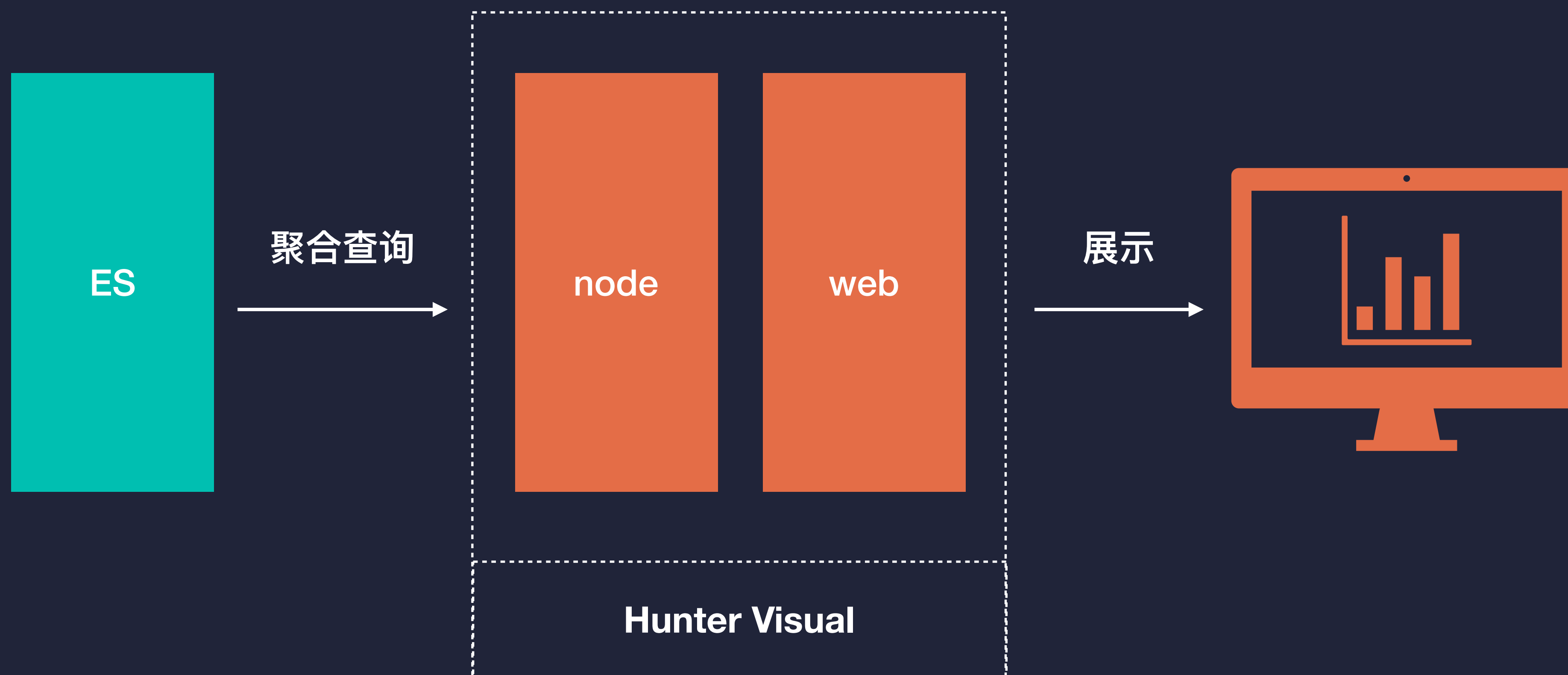
采集上报

日志服务

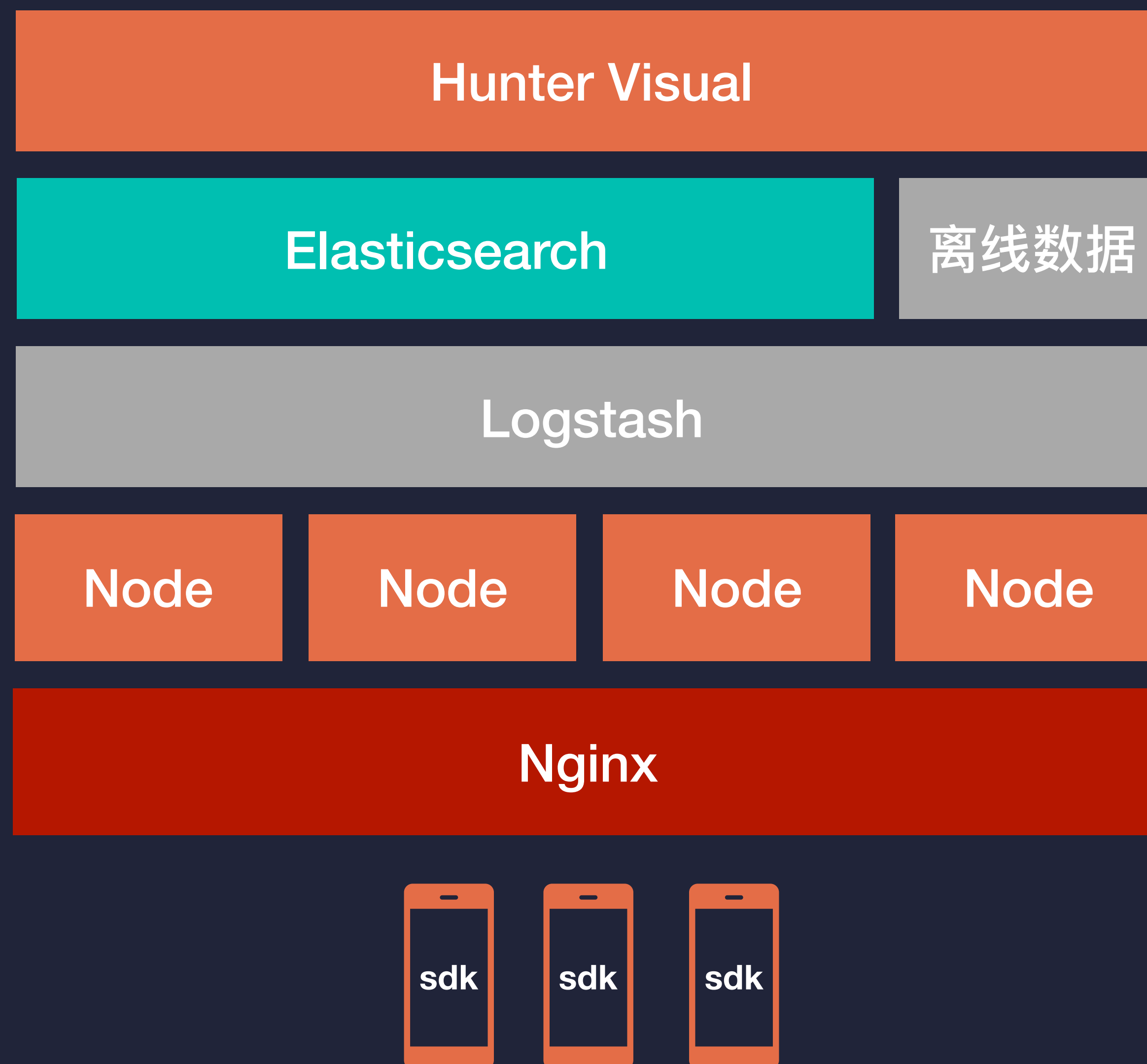
数据存储

展示分析

# 展示分析 —— 可视化图表应用



# 整体架构





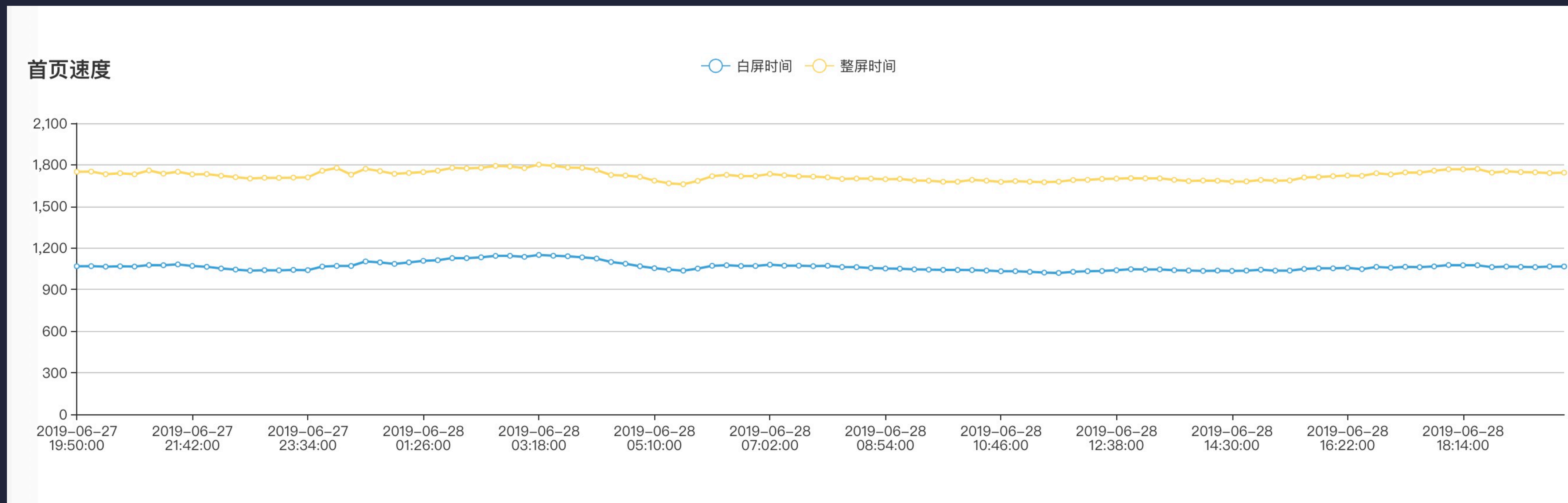
采集准确的数据，做好合适的技术选型和架构

- 为什么需要前端性能监控
- 推啊前端性能监控平台实践及思考
- 性能监控平台如何助力优化产品

# 性能监控平台如何助力优化产品

- 发现问题
- 定位问题
- 提升数据

# 发现问题



大盘：白屏1秒左右，整屏不到1.8秒

# 发现问题

大盘好就代表没问题吗?

- 慢页面是哪些?

页面名称	白屏时间 ⚡	整屏时间 ⚡
custom_smash_v2	1700	4924
drawLot	1955	3662
custom_gashapon_v5	1590	3651
ops_wallet3_2	2493	3602
cus_zadan_v1	1370	3413
fanpaizi_v3	897	3325

# 发现问题

平均时长快就一定快吗？

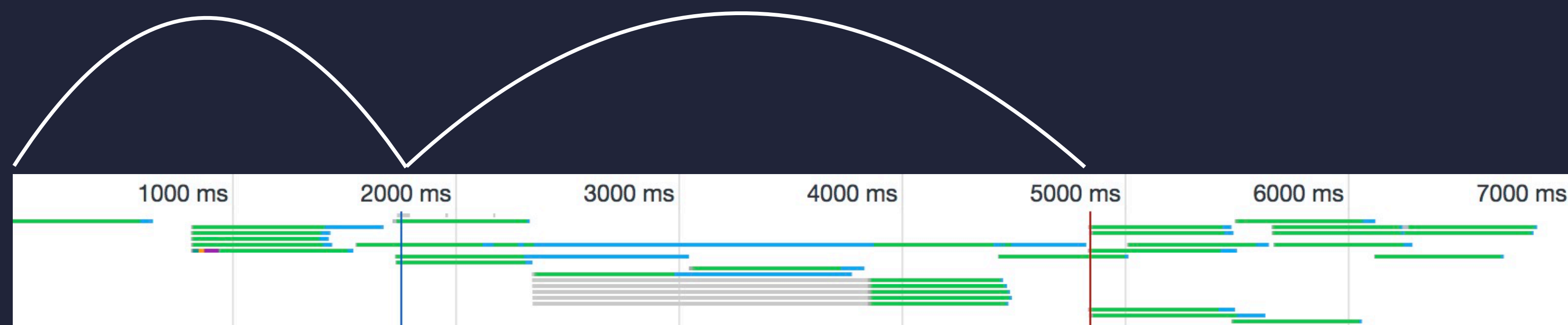
整屏时间：1.72秒

2秒内占比：73.87%

大盘报表											
页面名称	白屏时间	整屏时间	1秒内量	1秒内占比	2秒内量	2秒内占比	3秒内量	3秒内占比	大盘总量	流量占比	操作
大盘均值	1053	1721	11959013	38.26%	23085857	73.87%	27119535	86.77%	31253201	100%	

# 定位问题

一切指标都慢，但是调试却很正常？



白屏期

加载期

# 定位问题

害群之马

页面名称	白屏时间	整屏时间	1秒内量
大盘均值	1161	2459	63664

页面报表 媒体 广告位 导出报表

媒体ID	白屏时间	整屏时间	1秒内量
42211	1280	2699	24845
53289	1103	2281	27571
55793	591	1665	11088

正常环境：1.6秒

某环境：2.7秒



## 更多分析

- 性能数据：加载时长、不同纬度分析；
- 业务数据：结合业务参数，加载完成率，转化率等；
- 异常分析：第三方webview强缓存等；

前端性能监控，通过采集真实用户数据，  
帮助分析并推动产品优化

# Thanks.



暖风 

浙江 杭州



扫一扫上面的二维码图案，加我微信