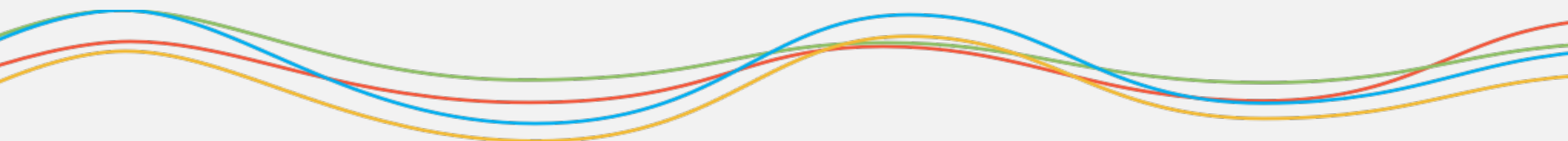


基于 Web 前端的可用性探索

百度 网页搜索部 李文倩





01.

前端可用性意义与现状

02.

前端可用性标准探索与实践

03.

前端可用性保障体系

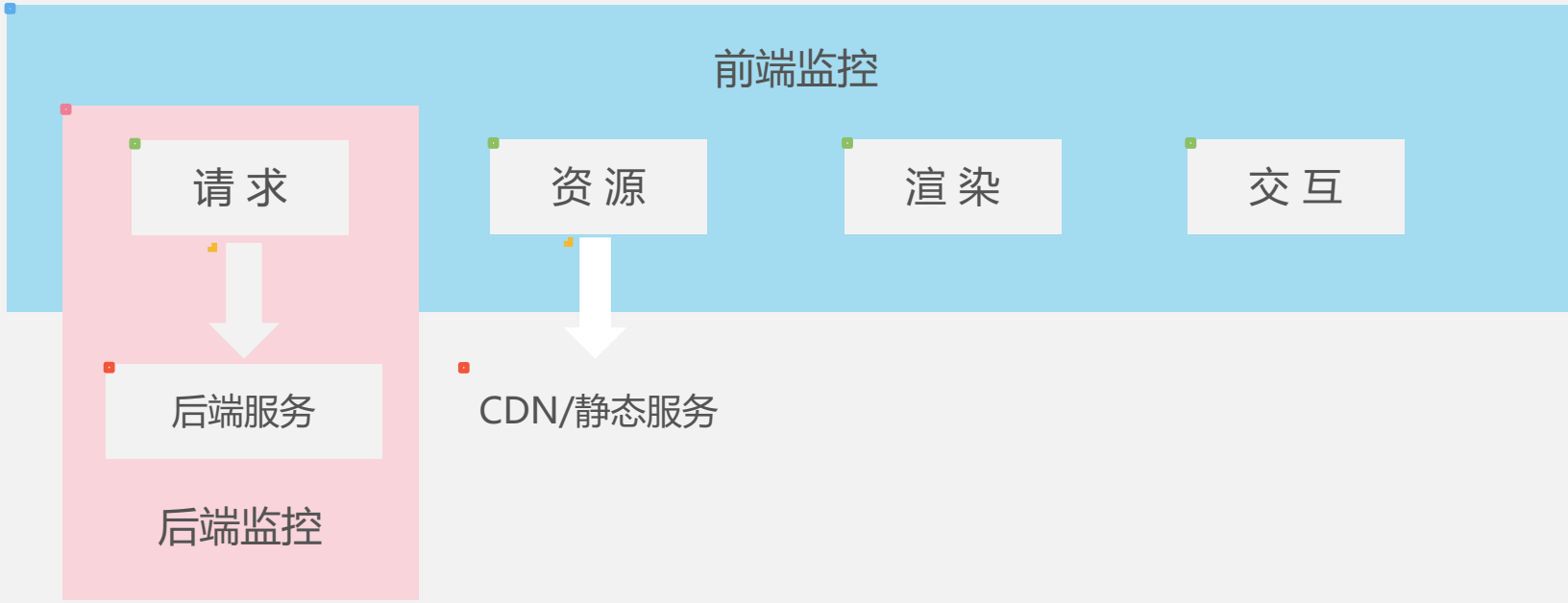
04.

前端可用性优化思路

01. 前端可用性意义与现状

- 什么是前端可用性
- 前端可用性现状
- 前端可用性建设意义

前端可用性是从用户的角度出发，检测整个系统的可用性，系统任何一个环节的缺失都会对体验造成影响。





页面功能和交互复杂度增加




前端功能测试局限性



各种前端渲染框架引入




运营式线上问题反馈



全面覆盖

覆盖系统请求、资源、渲染、交互各个环节



主动反馈

主动发现和反馈线上问题



优化提升

优化前端可用性
提升用户可用性体验

- 01. 前端可用性意义与现状
- 02. 前端可用性标准探索与实践
- 03. 前端可用性保障体系
- 04. 前端可用性优化思路

02. 前端可用性标准探索和实践

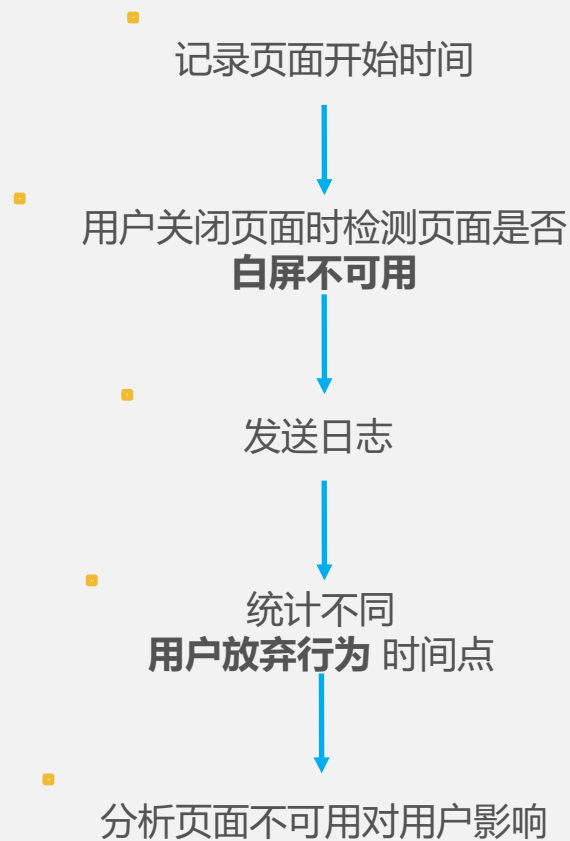
- 前端可用性评估指标
- 前端可用性数据现状
- 前端可用性标准定义

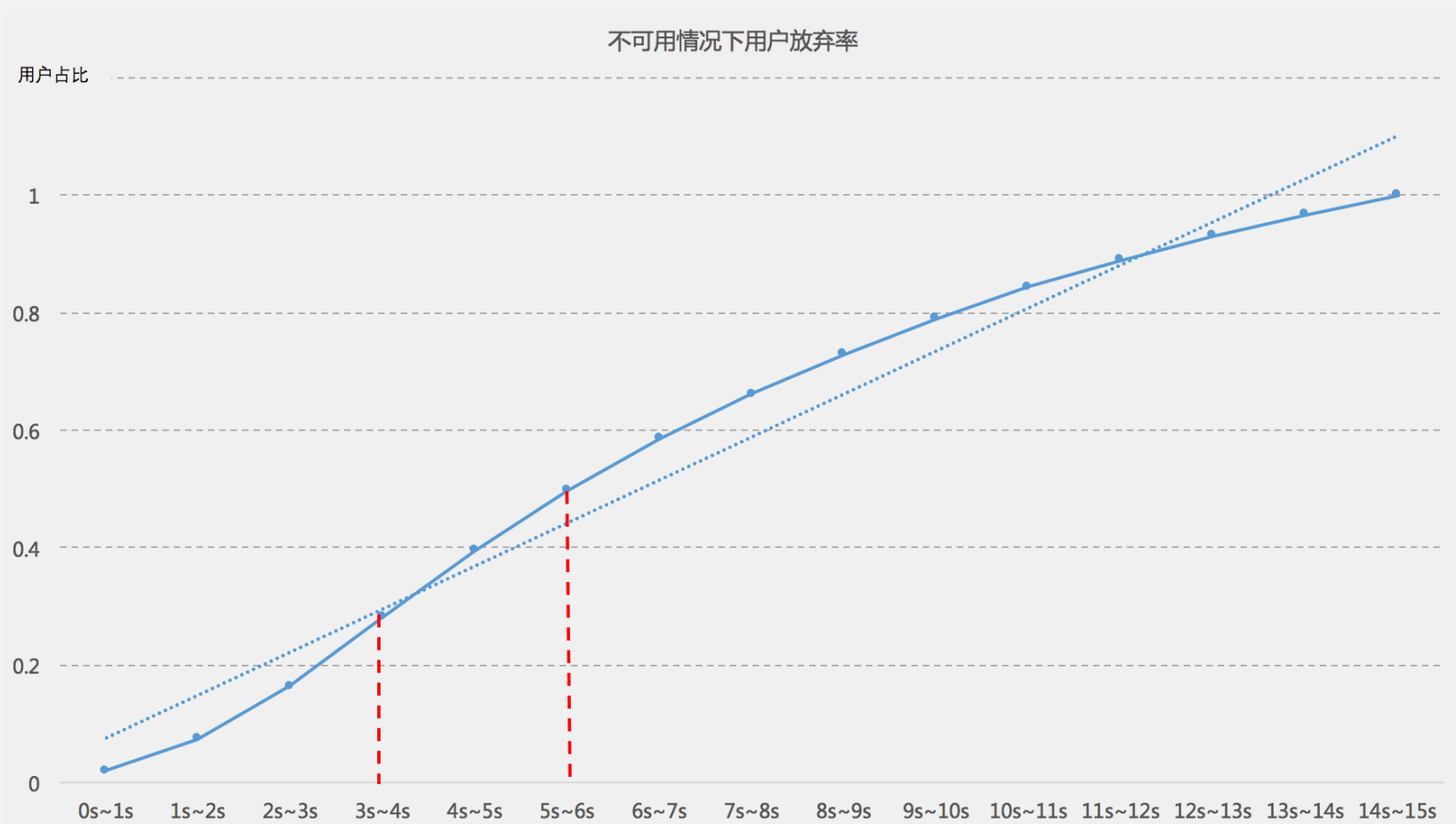


白屏不可用：在等待首屏渲染成功之前，用户会看到页面呈现白屏状态。

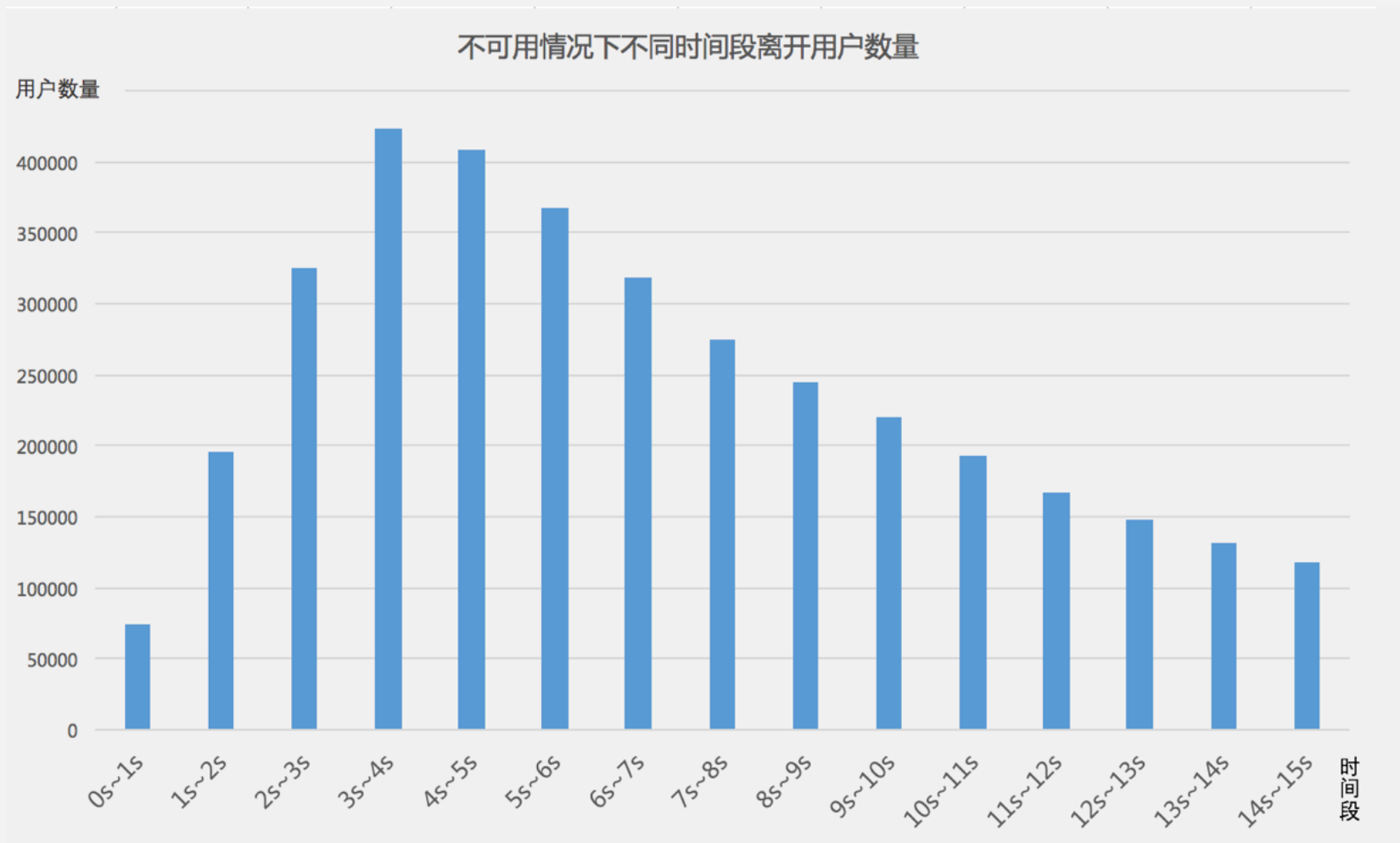
用户放弃行为：用户在等待页面内容成功渲染过程中，离开并返回上一页的行为。

对页面从不可用到可用过程中，对用户感知情况进行了实际测试，测试过程如图：





数据结论：在 **4s ~ 6s** 时用户放弃率增长较为显著



数据结论：**50%** 的用户会在白屏不可用 **6s** 内放弃浏览页面

对统计回传数据进行分析，在首屏成功渲染之前，白屏不可用极大影响了前端可用性，造成了大量的用户放弃，因此确立 **端上白屏不可用时间 6s** 为前端可用性标准核心指标之一。

01. 前端可用性意义与现状

02. 前端可用性标准探索与实践

03. 前端可用性保障体系

04. 前端可用性优化思路

03. 前端可用性保障体系

- 前端可用性保障系统设计
- 监控与预警
- 兜底容错机制

可用性系统要求：实时性、全面性

数据采集

请求异常

资源异常

渲染异常

交互异常

监控预警

实时监控

阈值报警

兜底容灾

容错机制

快速降级

数据采集：从可能异常点出发，通过 JS 进行打点回传

请求异常

请求状态码异常
请求超时
返回数据格式错误

■ AJAX 监控

资源异常

资源加载失败

■ CDN 监控

渲染异常

渲染、交互逻辑异常

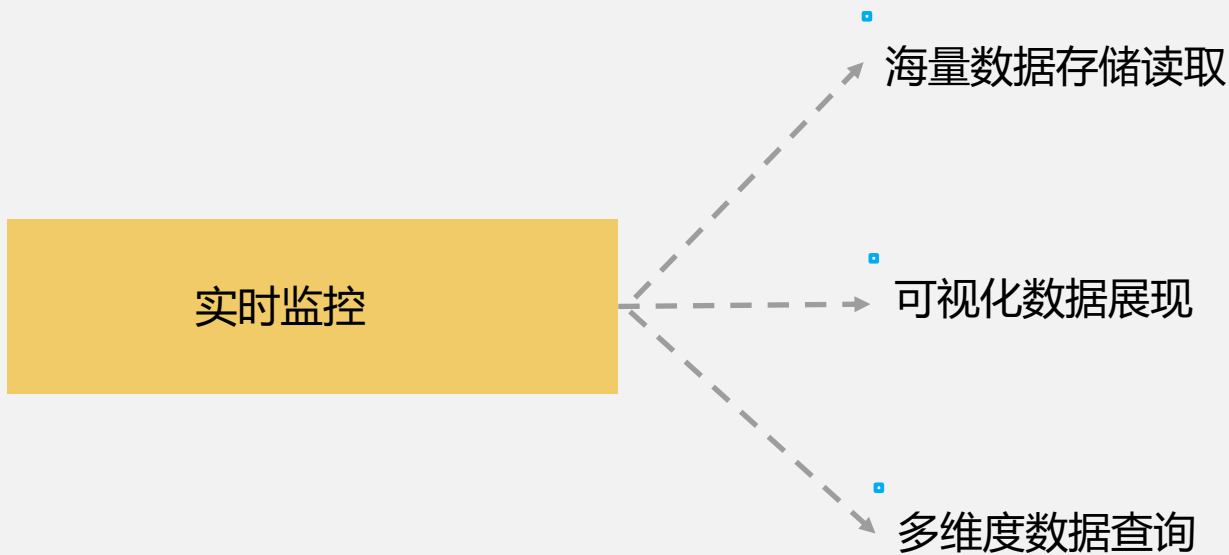
■ DOM 检查

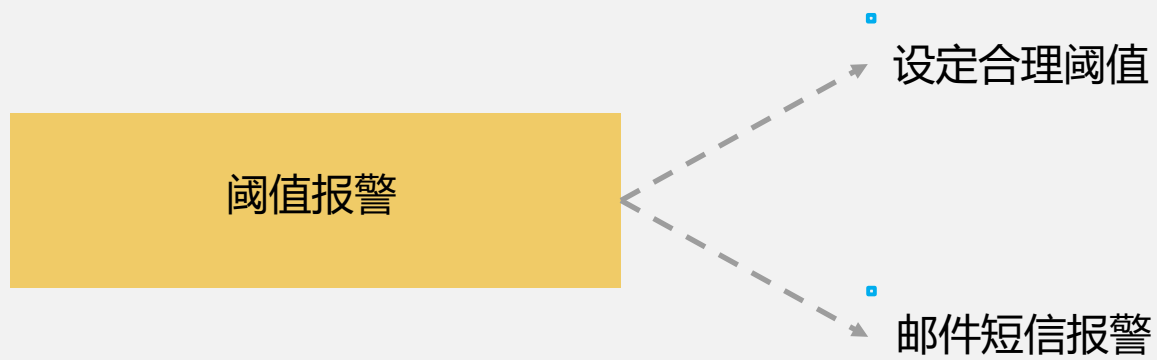
交互异常

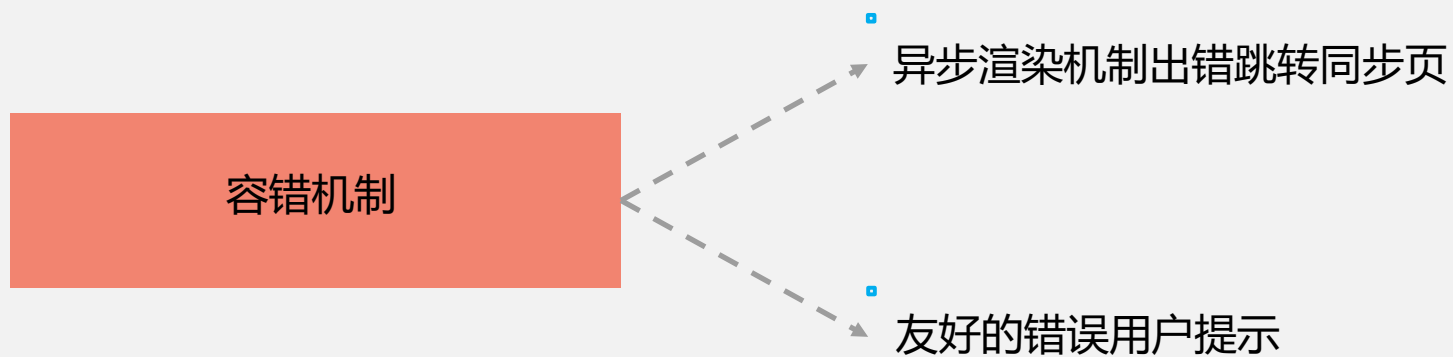
■ JS 错误监控

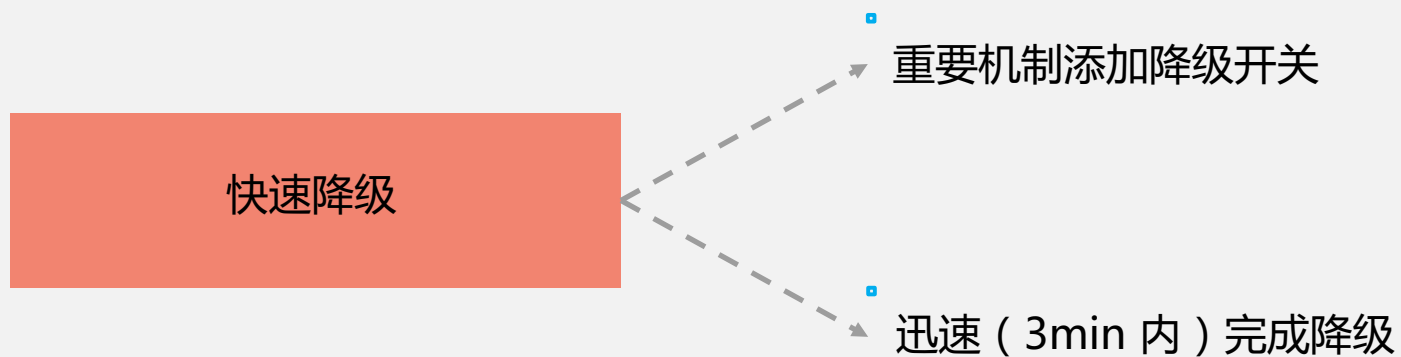
■ 兜底白屏监控

实时监控：完善的支撑系统









01. 前端可用性意义与现状

02. 前端可用性标准探索与实践

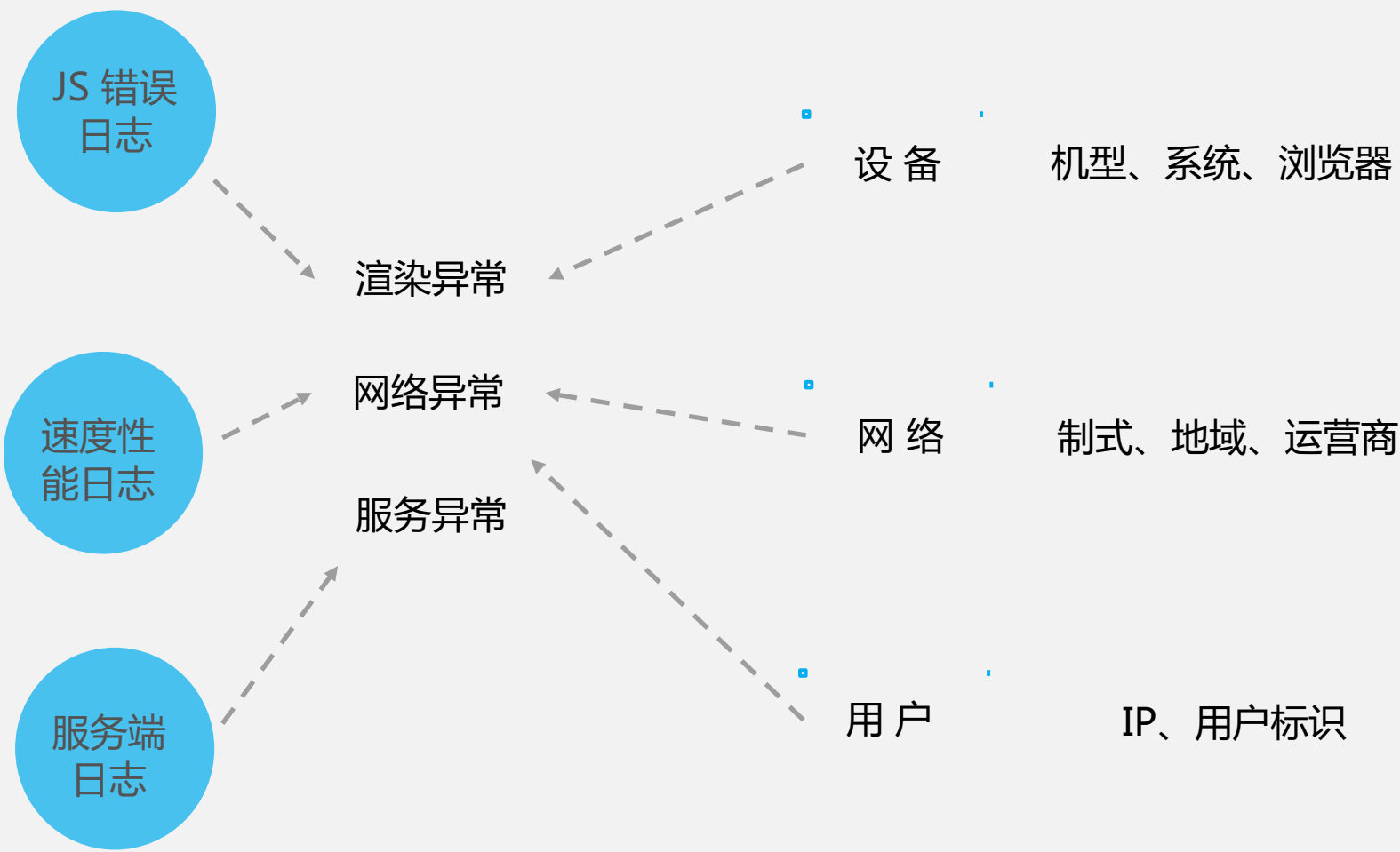
03. 前端可用性保障体系

04. 前端可用性优化思路

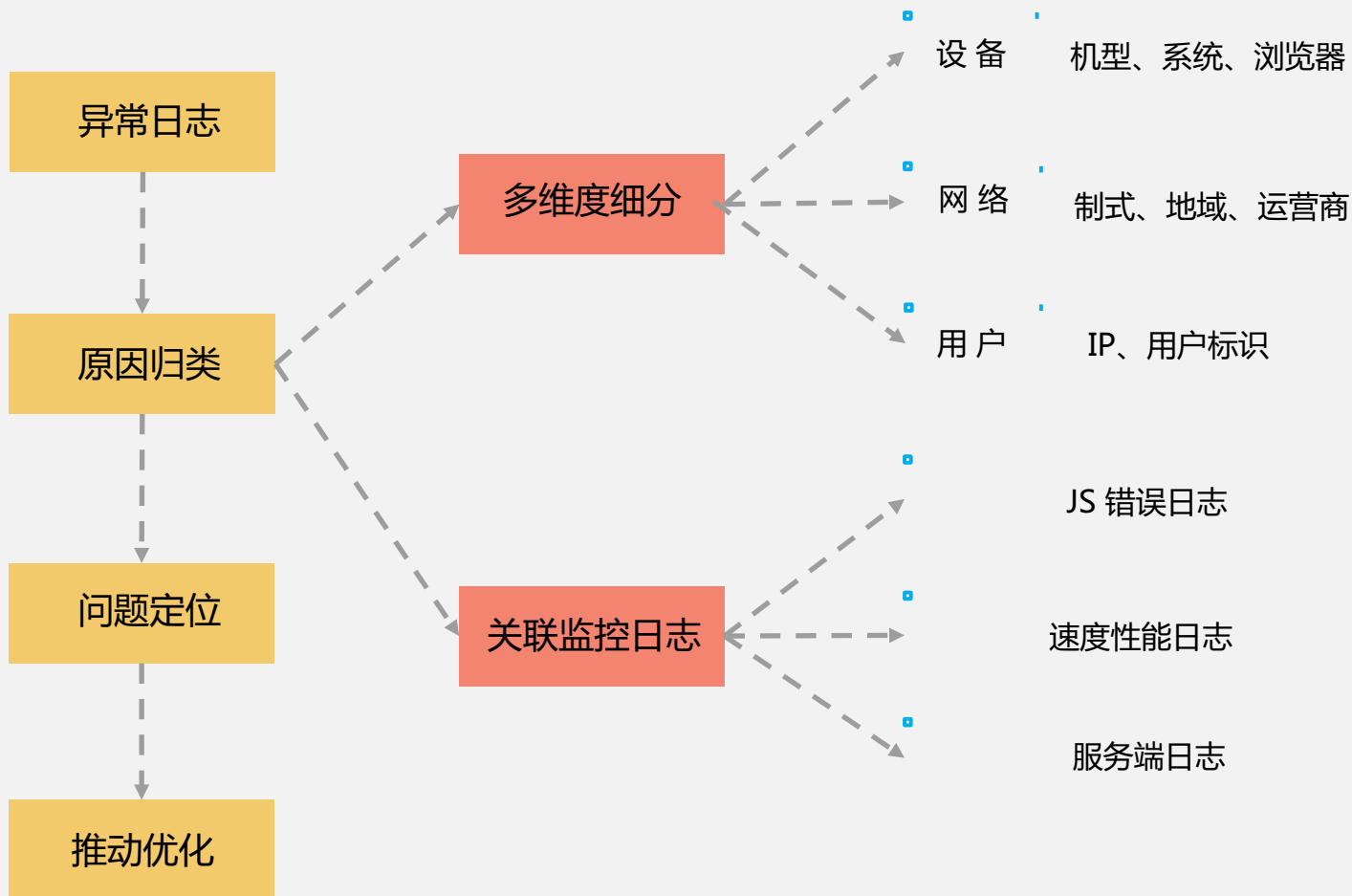
04. 前端可用性优化思路

- 优化案例分析
- 优化思路总结

如何优化某产品的前端可用性？



运用统计思维分析数据，还原用户交互过程。



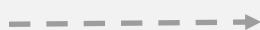
Good Case1: 实时性

某产品线前端配置上线



多模块耦合，功能测试局限

上线过程中发现端上白屏指标数量突增



兜底监控，端上核心服务指标

触发报警机制



实时反馈问题

迅速进行降级回滚



快速降级挽回损失

Good Case2: 全面性

某小流量实验转全，后端服务监控表现正常



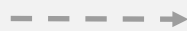
后端监控覆盖不全

白屏展现数量大幅上升



无法立即定位问题

数据分析前端代码有浏览器兼容性问题，
某些浏览器下发生白屏、样式错乱等问题



结合优化分析思路

迅速修复问题，挽回流量损失



主动反馈线上问题

24H

全天监控预警服务

数十亿

日均搜索 PV

10+

搜索点出产品类型

提升整体业务故障响应能力

01

智能化分析辅助

02

03

异常采集机制标准化&通用化

THANK YOU

