

产品中心系统高可用实战

槐国涛



槐国涛
美团旅行 技术专家

产品中心负责人

纲要

产品简介

可用性管理

假日应对

产品中心简介

业务介绍

酒店频道



门店列表



产品列表



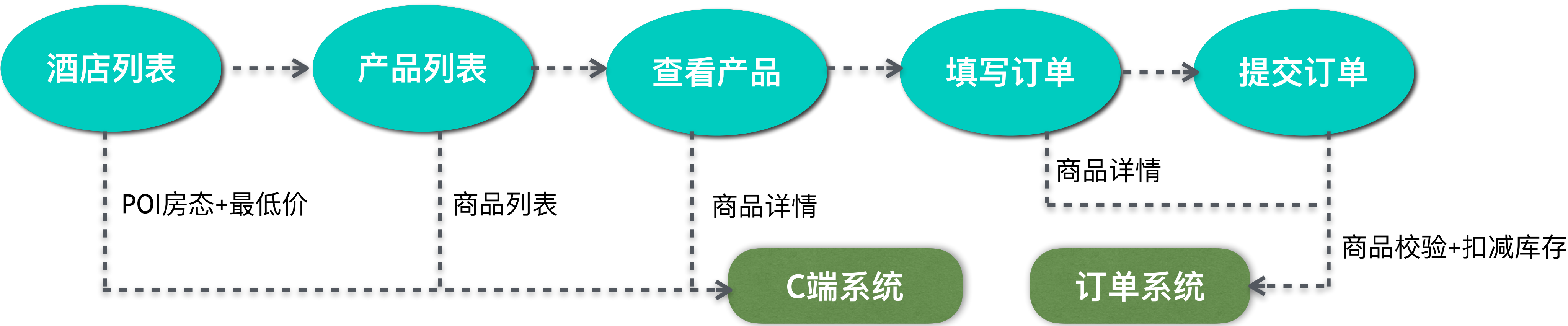
填单页



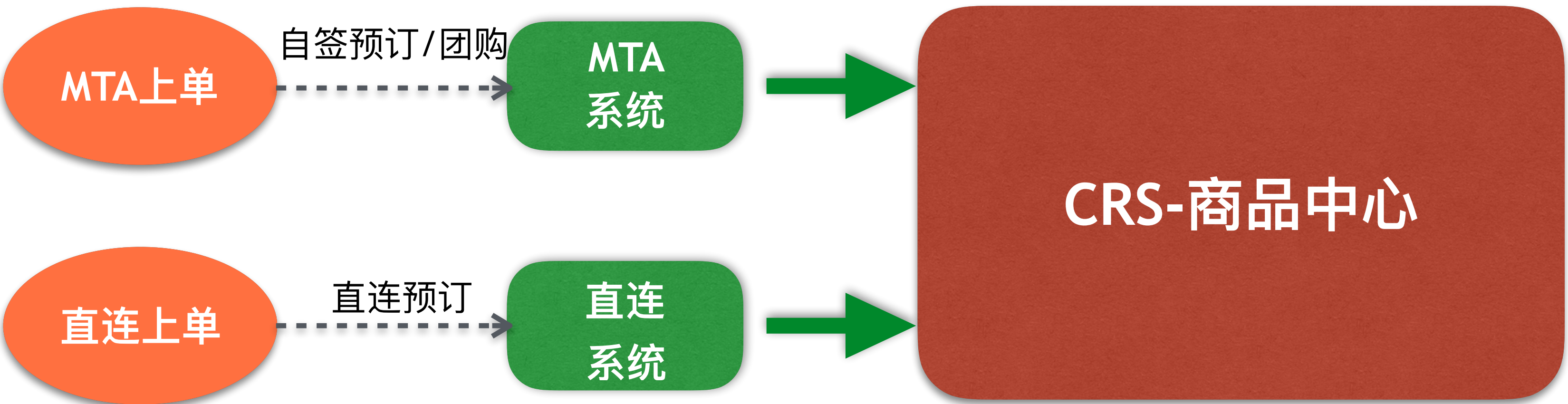
产品中心简介
系统交互



用户流程



上单流程



产品中心简介

系统数据

90000+

QPS

300ms

TP999

9999%+

可用性

纲要



可用性管理

纲要

01 可用性定义

可用性定义，SLA解释

02 可用性提升

持续发现问题，解决问题的过程，循序渐进提升的过程

可用性管理

可用性定义

- 高可用性：High Availability
- **MTTF**：Mean Time To Failure
- **MTTR**：Mean Time To Repair
- **Availability** = $MTTF / (MTTF + MTTR) * 100\%$

描述	通俗叫法	可用性级别	年度停机时间
基本可用性	2个9	99%	87.6小时
较高可用性	3个9	99.9%	8.8小时
具有故障自动恢复能力的可用性	4个9	99.99%	53分钟
极高可用性	5个9	99.999%	5分钟

可用性管理

SLA解释

业界定义

- SLA: 服务等级契约 (SLO + 后果)
- SLO: 服务等级目标
- SLI: 服务等级指标

监控服务定义

- SLA: 成功率<99.99% 服务不达标
- SLO: 成功率>99.99%
- SLI: $\text{成功率} = \frac{\text{成功请求量}}{\text{失败请求量} + \text{成功请求量}} * 100\%$

业务定义

- SLA: P0系统达标率>98% 扣除100元奖金
- SLO: 达标率>98% (原则: 收益>成本)
- SLI: $\text{达标率} = \frac{\text{成功率达标天数}}{\text{P0系统数量} * \text{天数}} * 100\%$

可用性管理

可用性管理

01 可用性定义

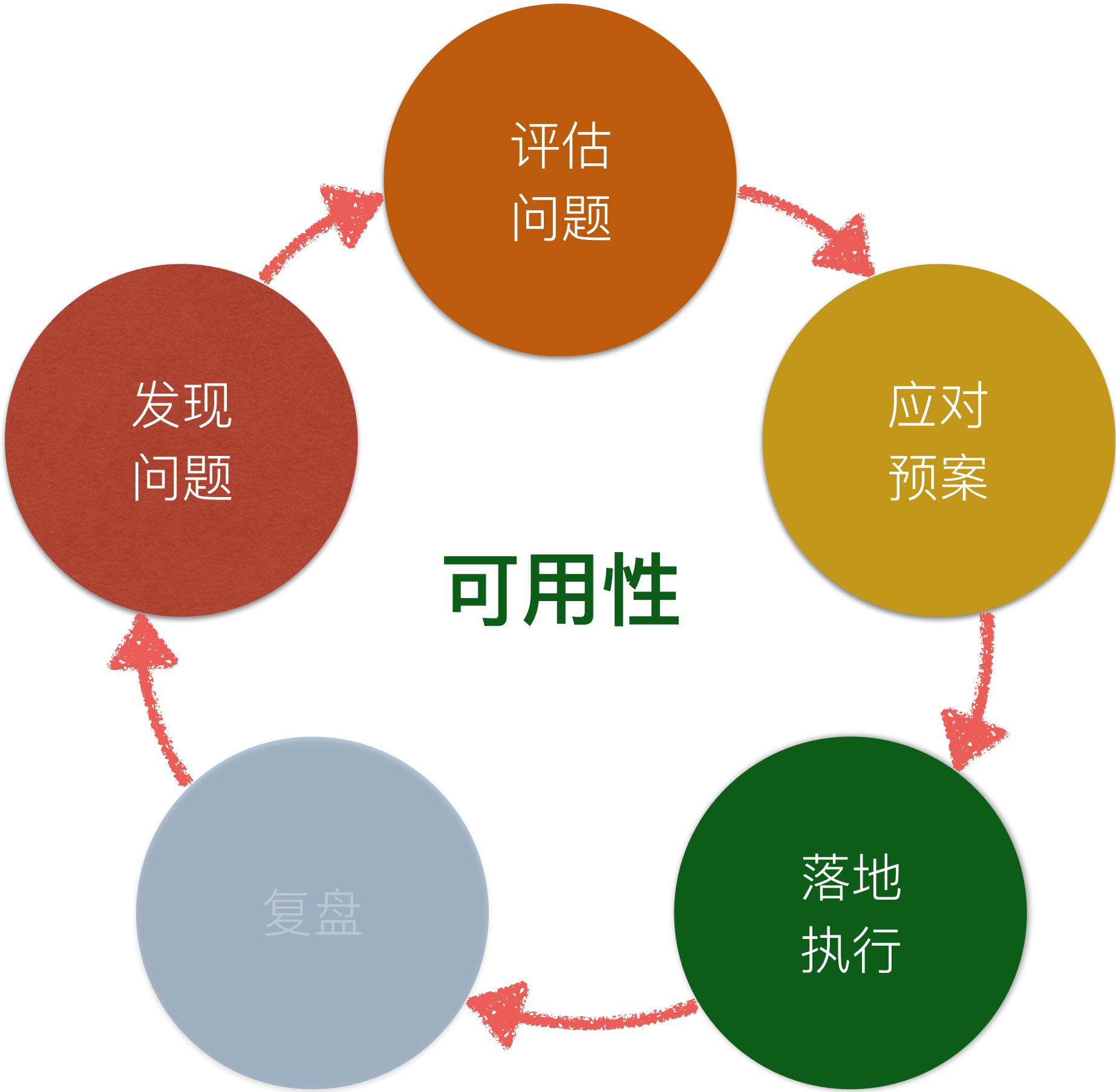
业界定义，基础服务定义，业务定义

02 可用性提升

持续发现问题，解决问题的过程，循序渐进提升的过程

可用性管理

可用性提升



可用性管理－可用性提升

发现问题

- A 影响因素
- B 关注点
- C 思路
- D 方法

可用性管理－可用性提升－发现问题

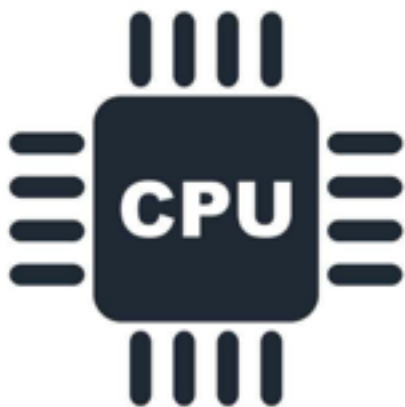
影响因素－硬件因素



交互



传输



计算



存储

硬件

手机
PC
智能设备
.....

网卡
交换机
网线
.....

CPU
.....

内存
磁盘
.....

软件

APP
小程序
浏览器
操作系统
.....

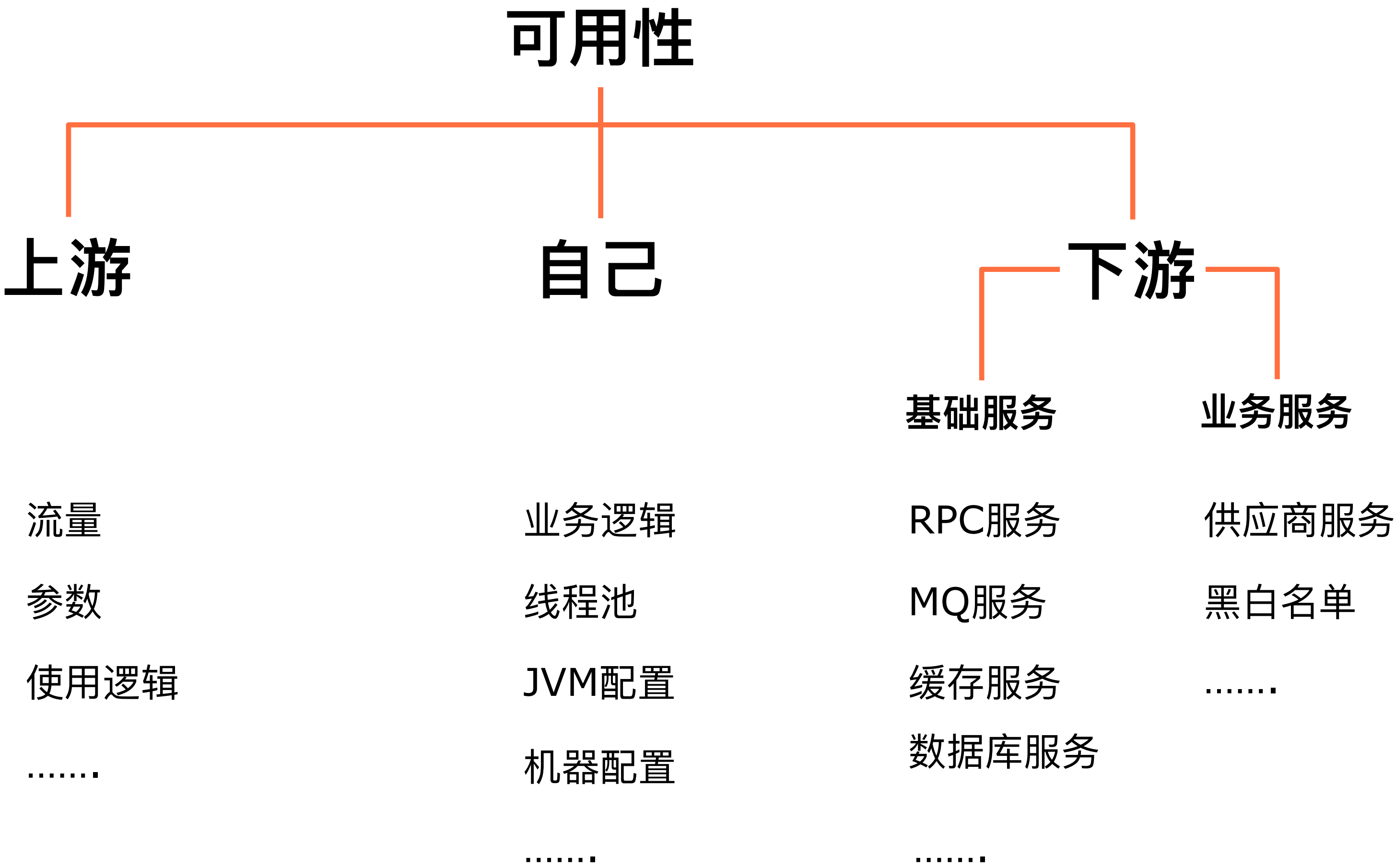
DNS
负载均衡
防火墙
消息队列
.....

批量计算
作业调度
业务服务
.....

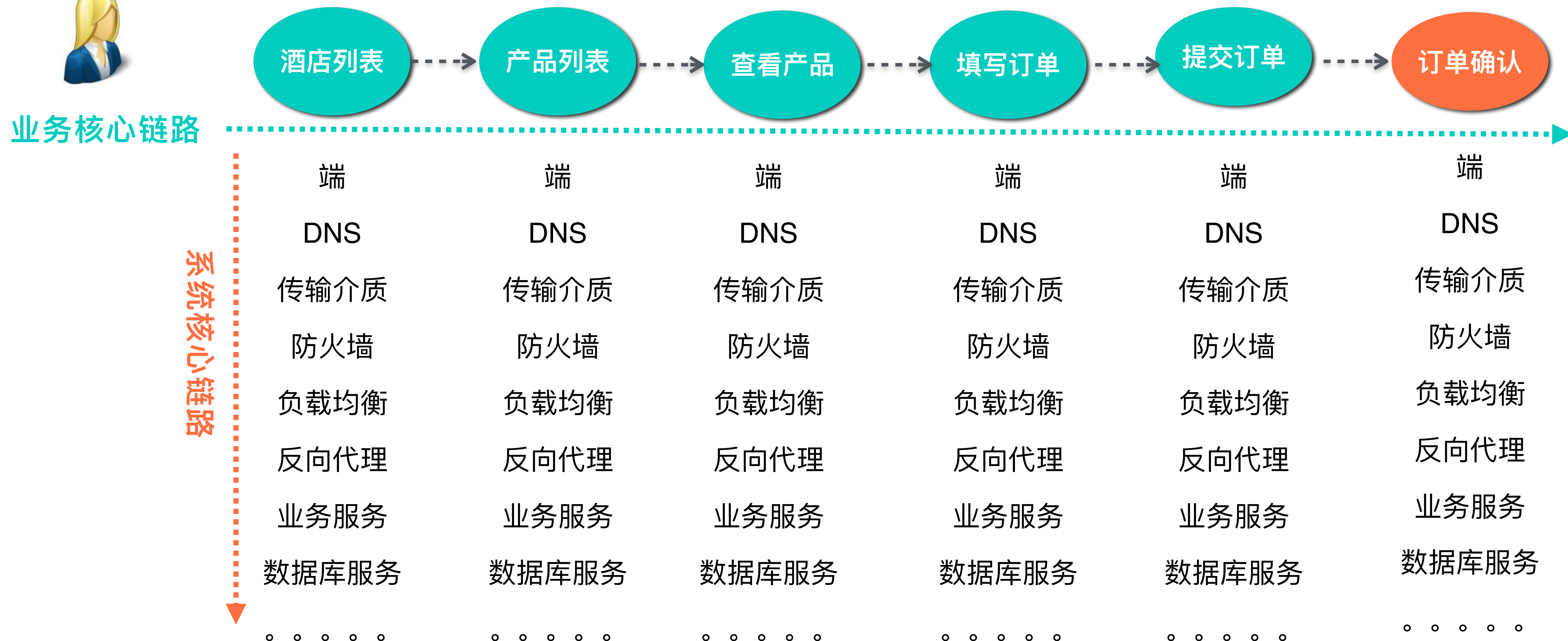
缓存服务
数据库服务
.....

可用性管理－可用性提升－发现问题

影响因素－上下游

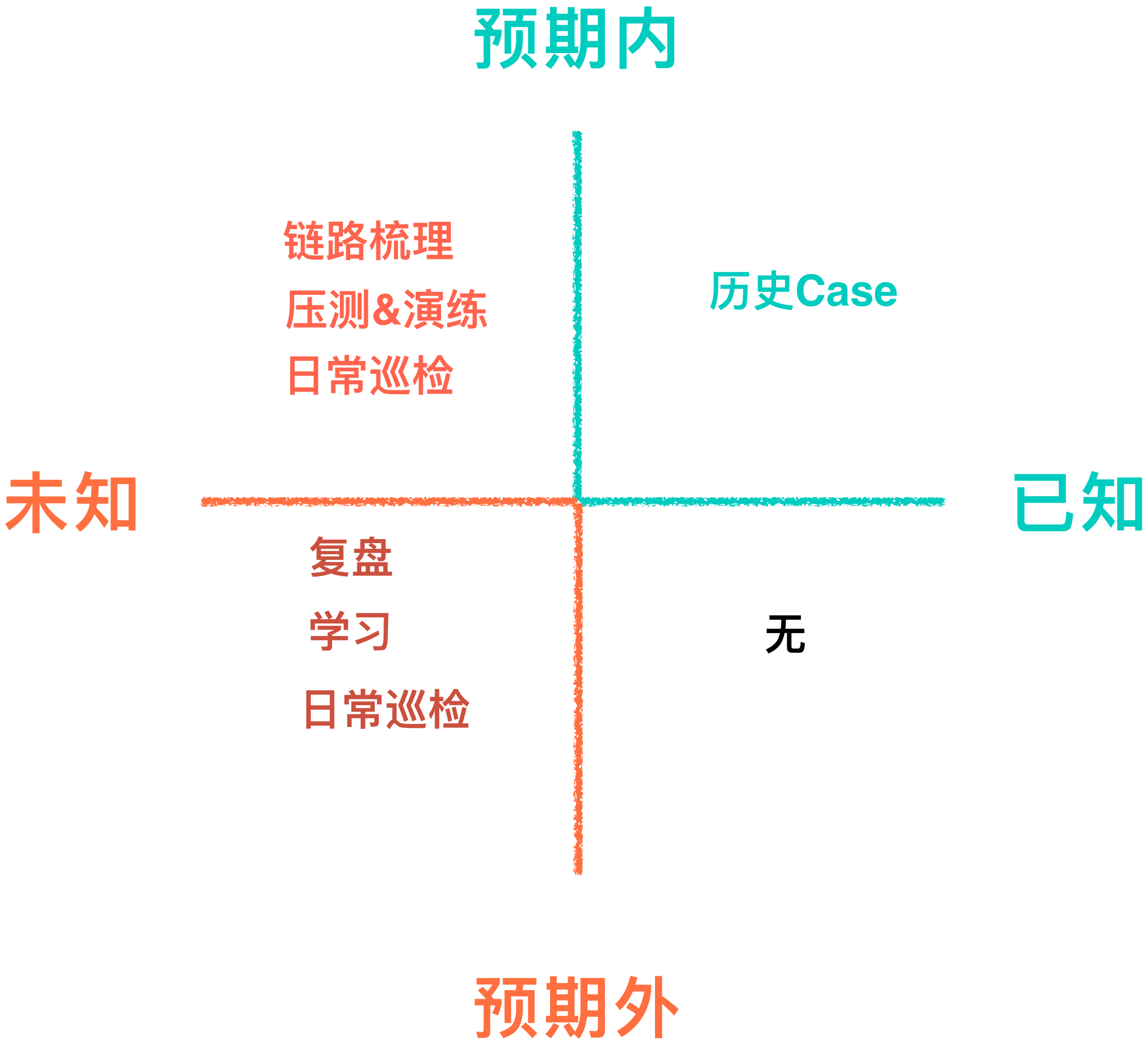


发现问题—关注点



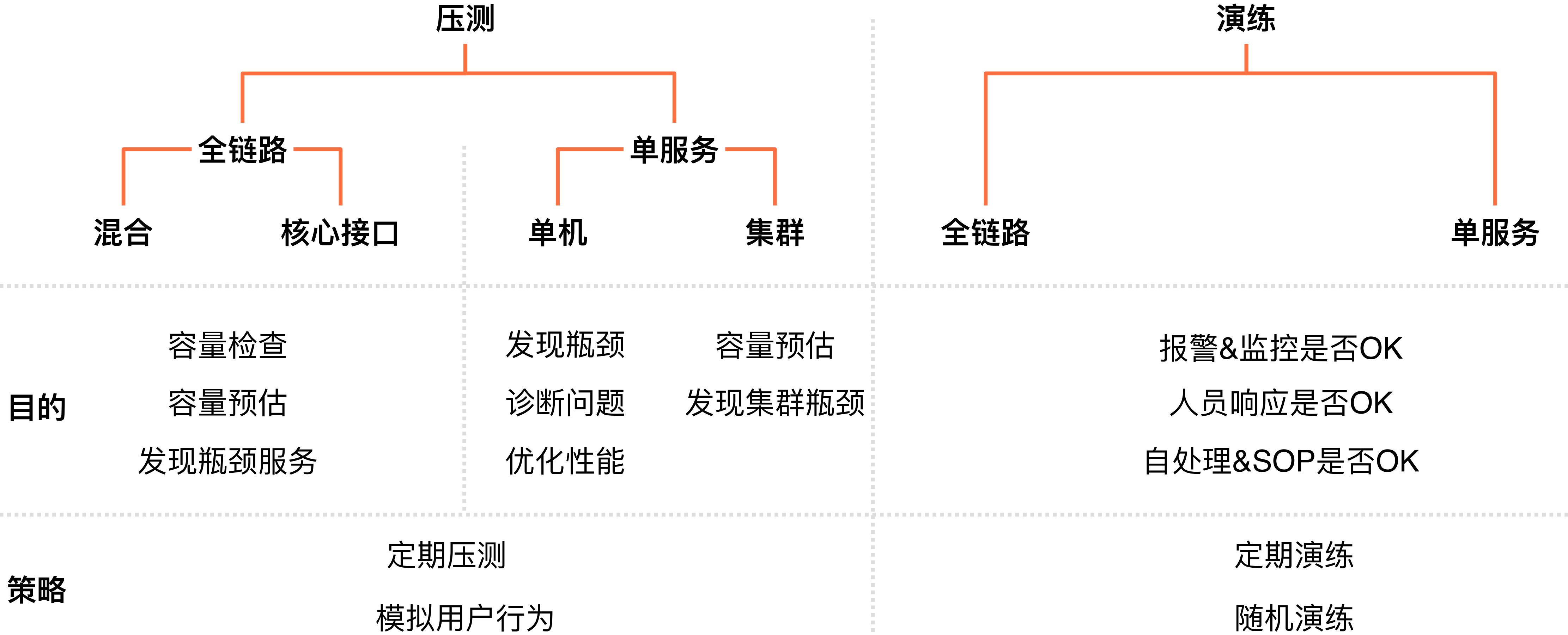
可用性管理－可用性提升－发现问题

思路



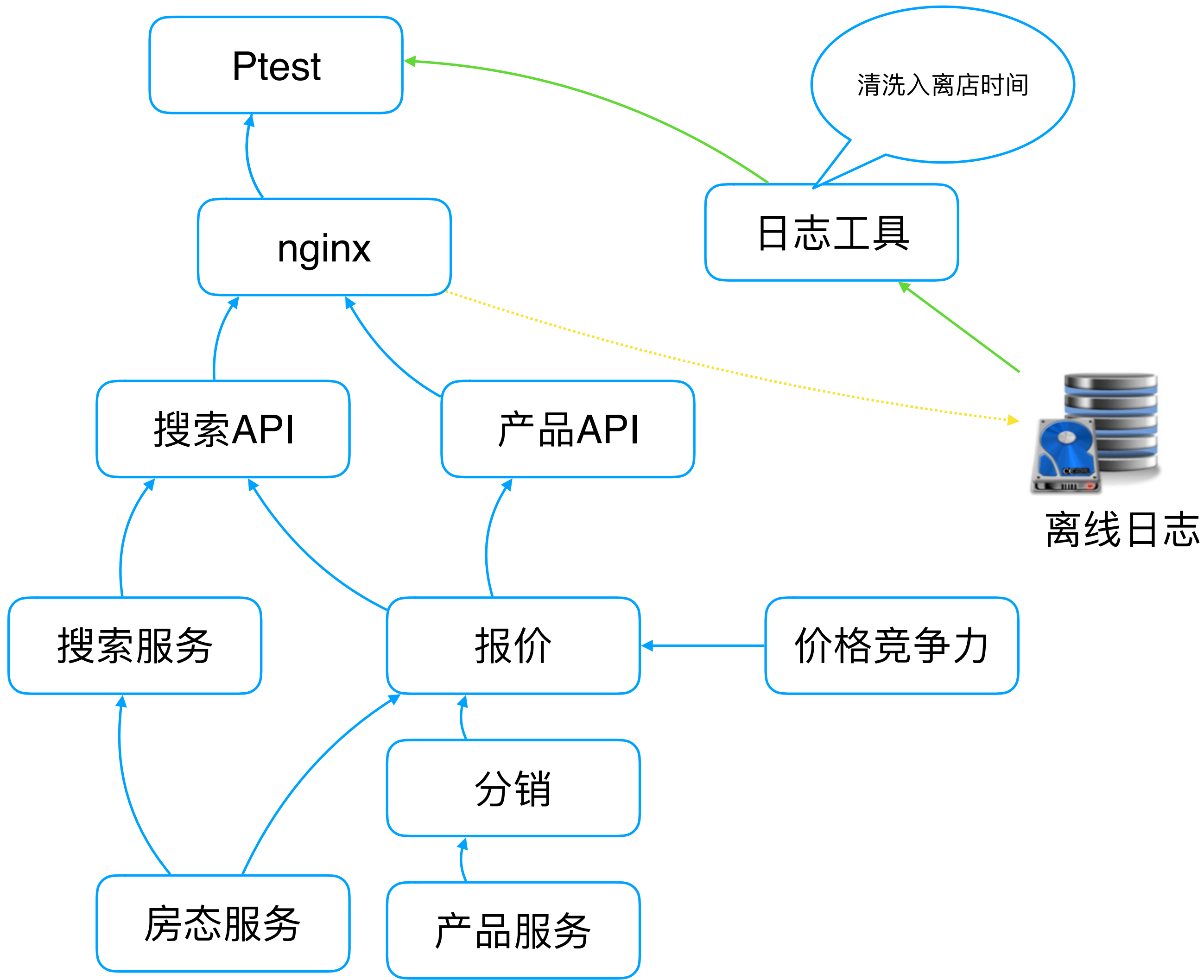
可用性管理－可用性提升－发现问题

方法－压测&演练



可用性管理 - 可用性提升 - 发现问题

方法 - 压测



← 压测请求

← 离线日志

← 离线日志

压测目标

业务量春节1.5倍

系统容量春节3倍

发现问题

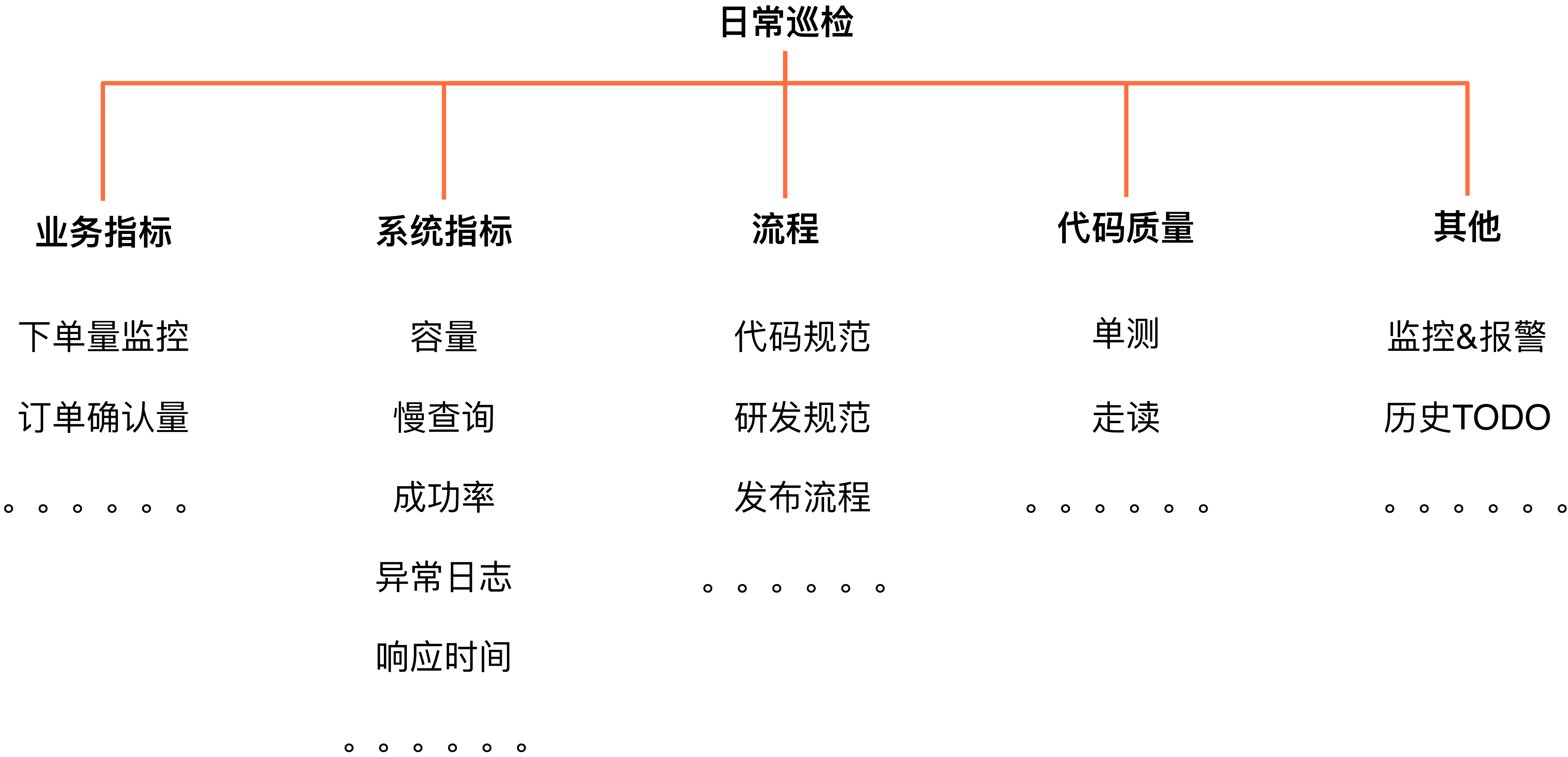
产品API 超时

分销线程池设置不合理

缓存服务流量限制

可用性管理－可用性提升－发现问题

方法－日常巡检



可用性管理－可用性提升－发现问题

方法－日常巡检

可用性

连续3天、5天、7天同一服务稳定性不达标，划分成黄色预警（TL介入）、橙色预警（架构师介入）和红色预警（B端负责人介入）。

1.可用性列表

统计周期：周四～周三（12.21～12.27）

	日期	工程	可用性	异常	原因
1	12.27		99.9387%	<50000	商家营销新接口 PoiFilterService.queryPricePolicyByParamFromCache 限流，导致客户端可用性受影响

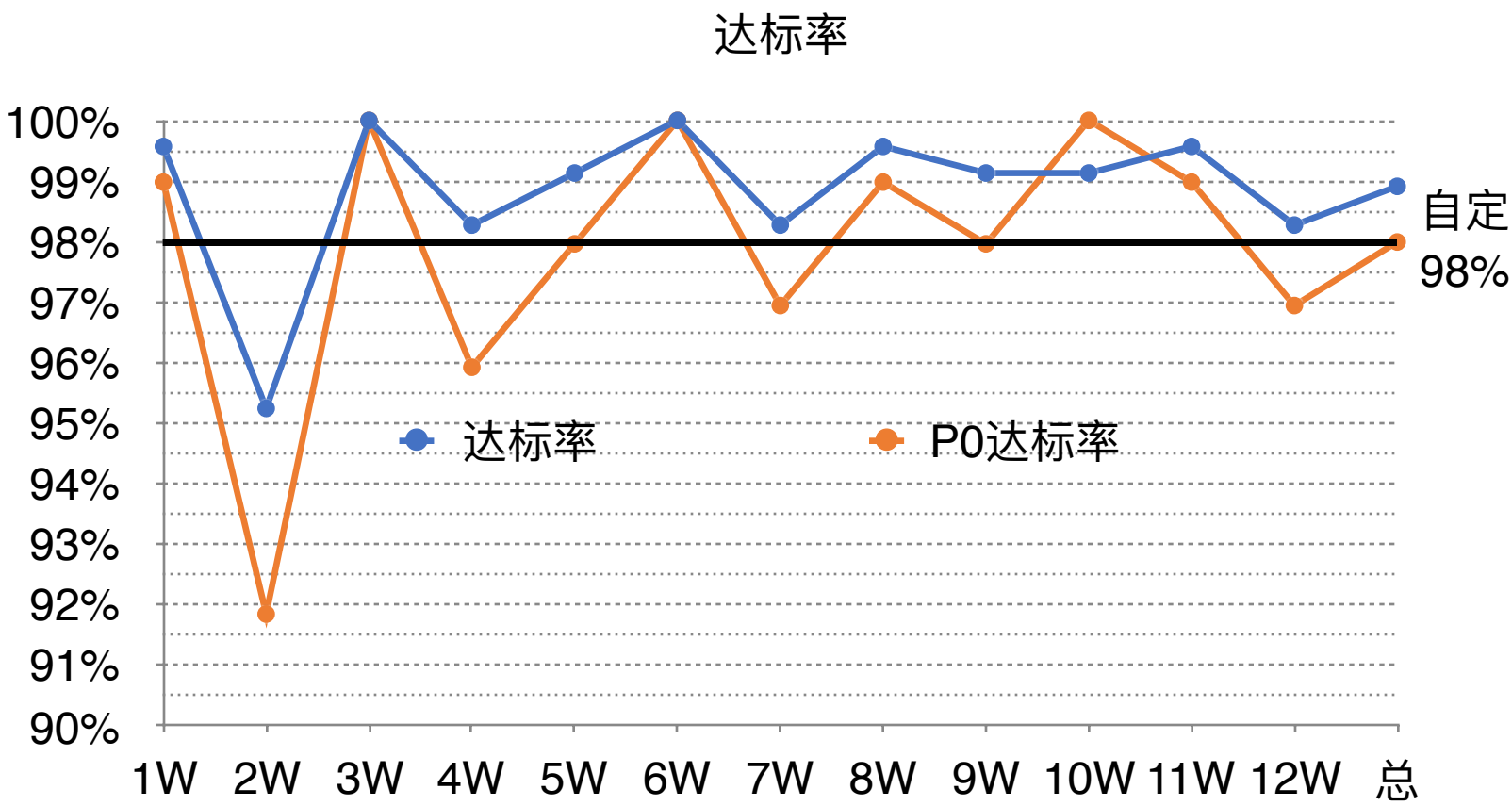
响应时间

服务等级	方向	服务名（Appkey）	标准	可用性	不达标明细	服务端异常	TP90	TP99
P1	商家业务		99.99% / 50000 / 200 / 1000	99.9522%	99.9664%（12.27）、 99.9272%（12.26）、 99.8922%（12.25）、 99.9863%（12.24）、 99.902%（12.22）	3	353（12.25）、 351（12.22）、 334（12.24）、 330（12.21）	2280（12.2）、 1561（12.2）、 1201（12.2）、 1134（12.2）

容量

服务等级	方向	服务名（Appkey）	标准	QPS	请求量	原因分析
P0	CRS		74000	55095	2708785120	

达标率(SLO)



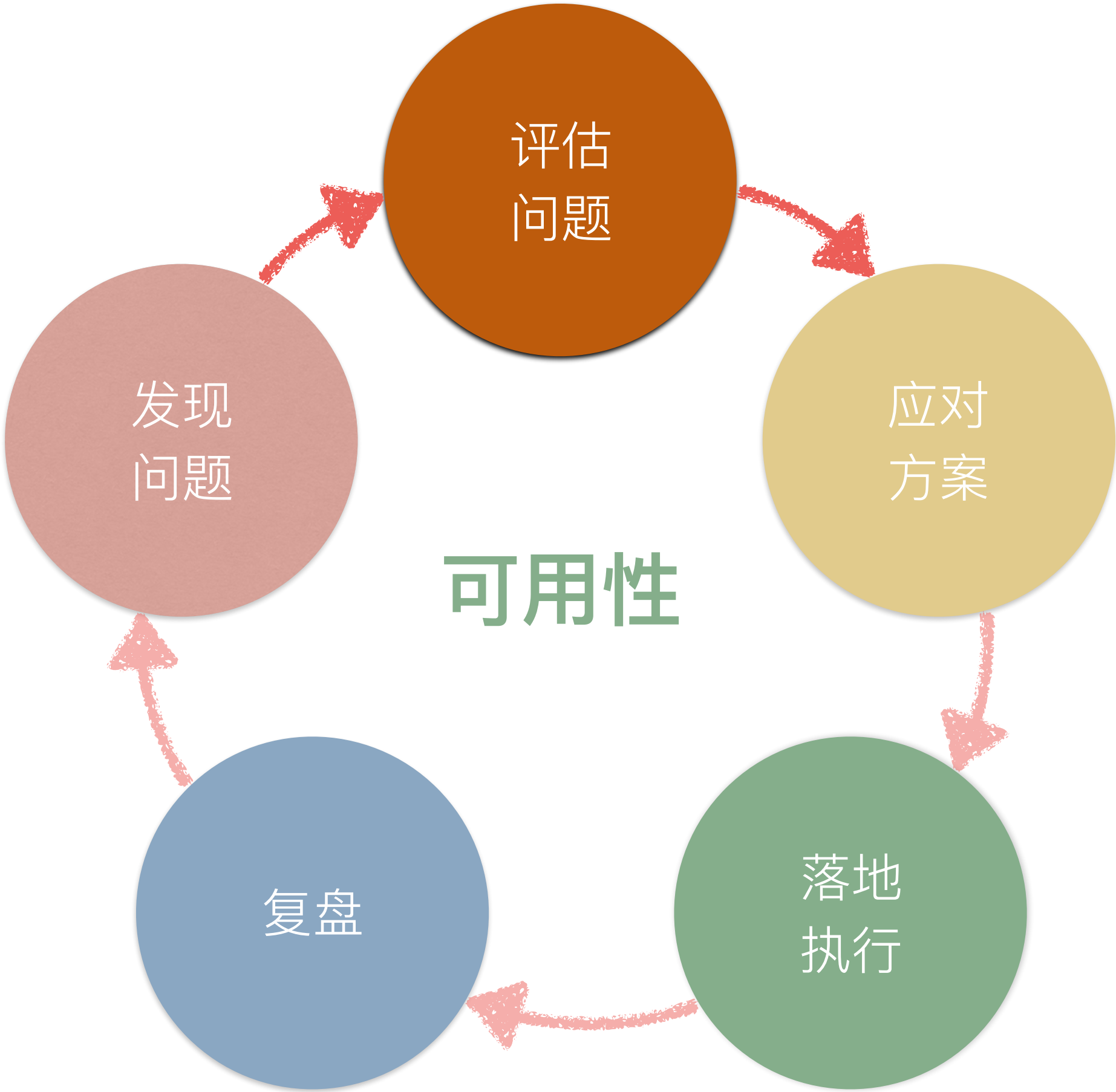
代码质量

服务等级	方向	服务名（Appkey）	标准	代码 bugs	代码缺陷	代码问题	代码重复率	代码复杂度	原因分析
P1	CRS		100 / 100 / 2000 / 20.0% / 10000	456	196	5286	17.2%	21221	

服务分级，重点关注P0服务

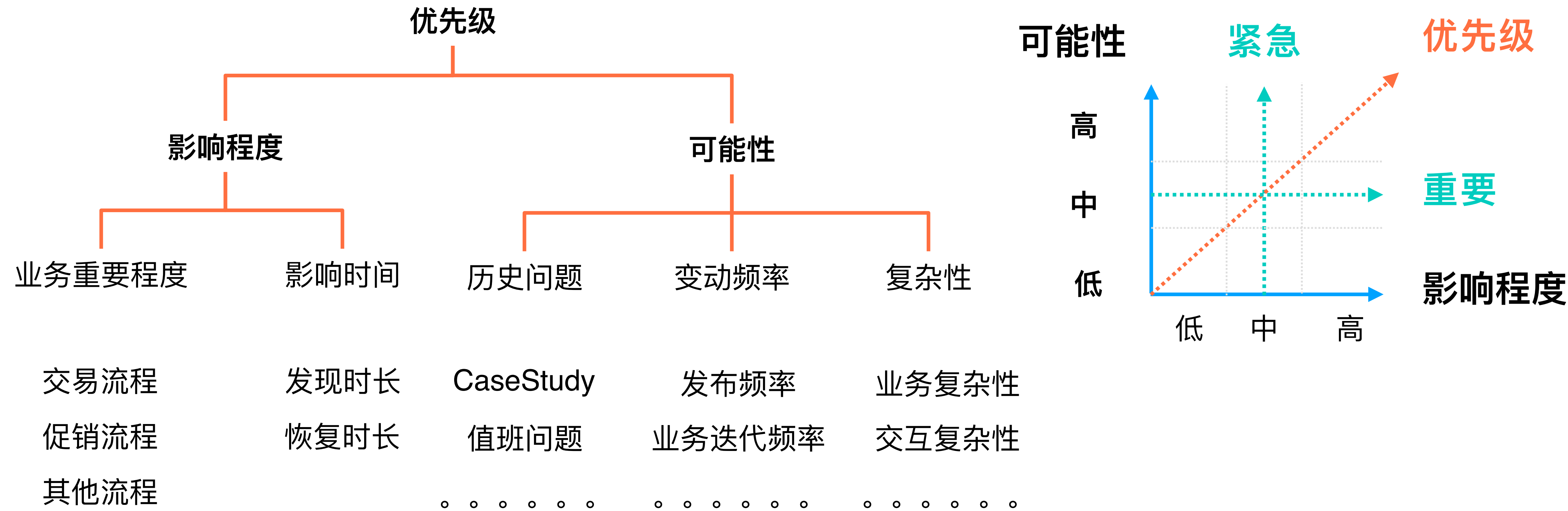
P0服务:影响交易，用户体验影响大的服务

可用性管理
评估问题



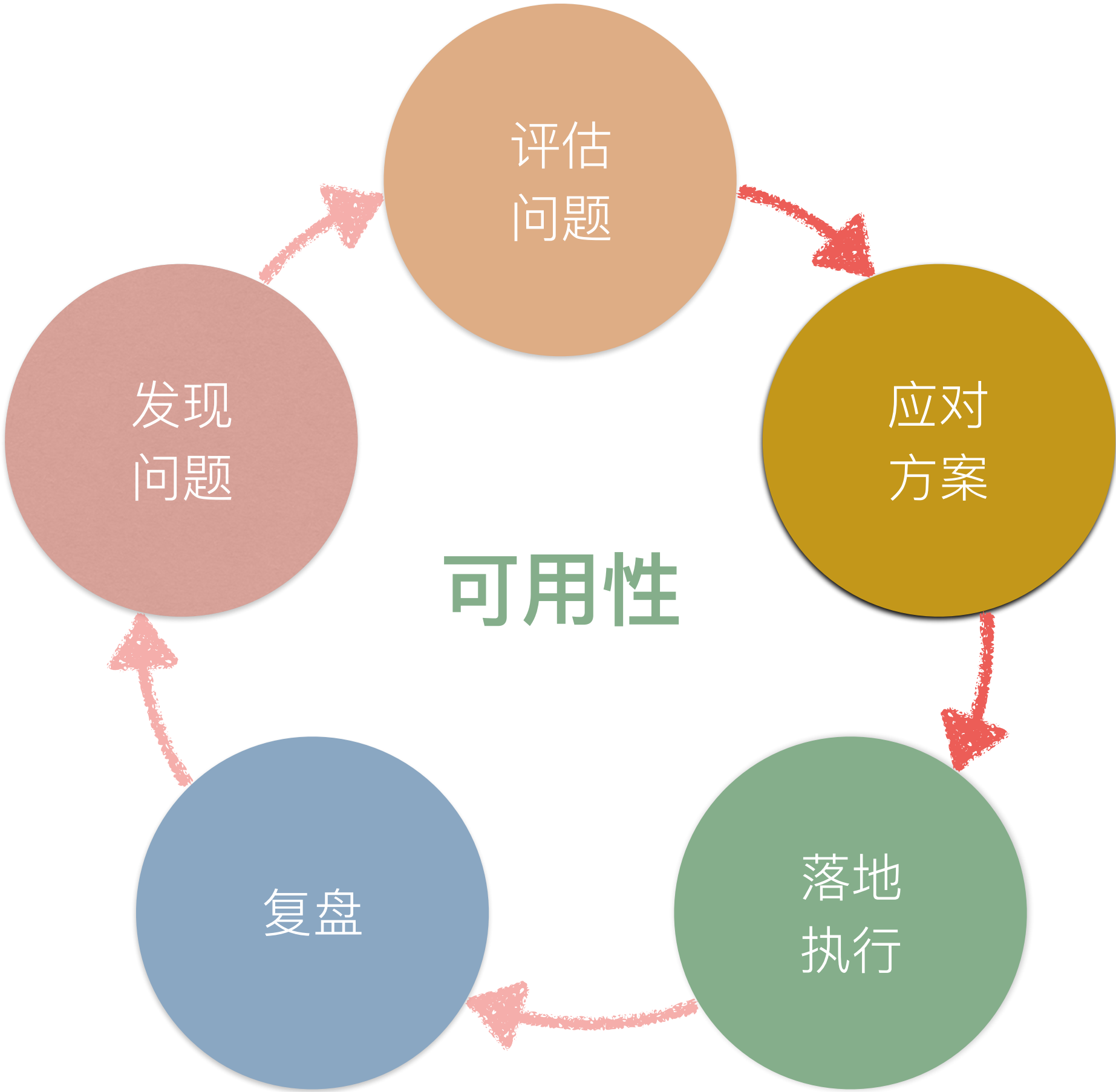
可用性管理

评估问题



可用性管理

应对方案



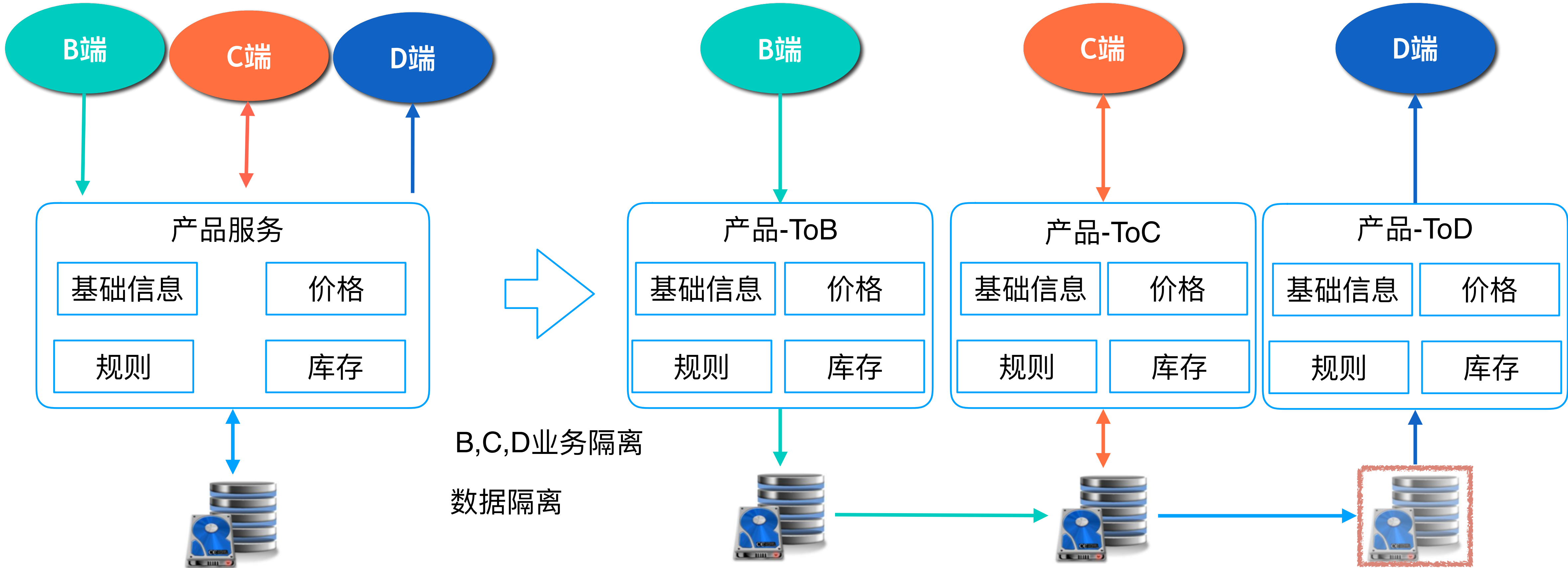
可用性管理

应对方案



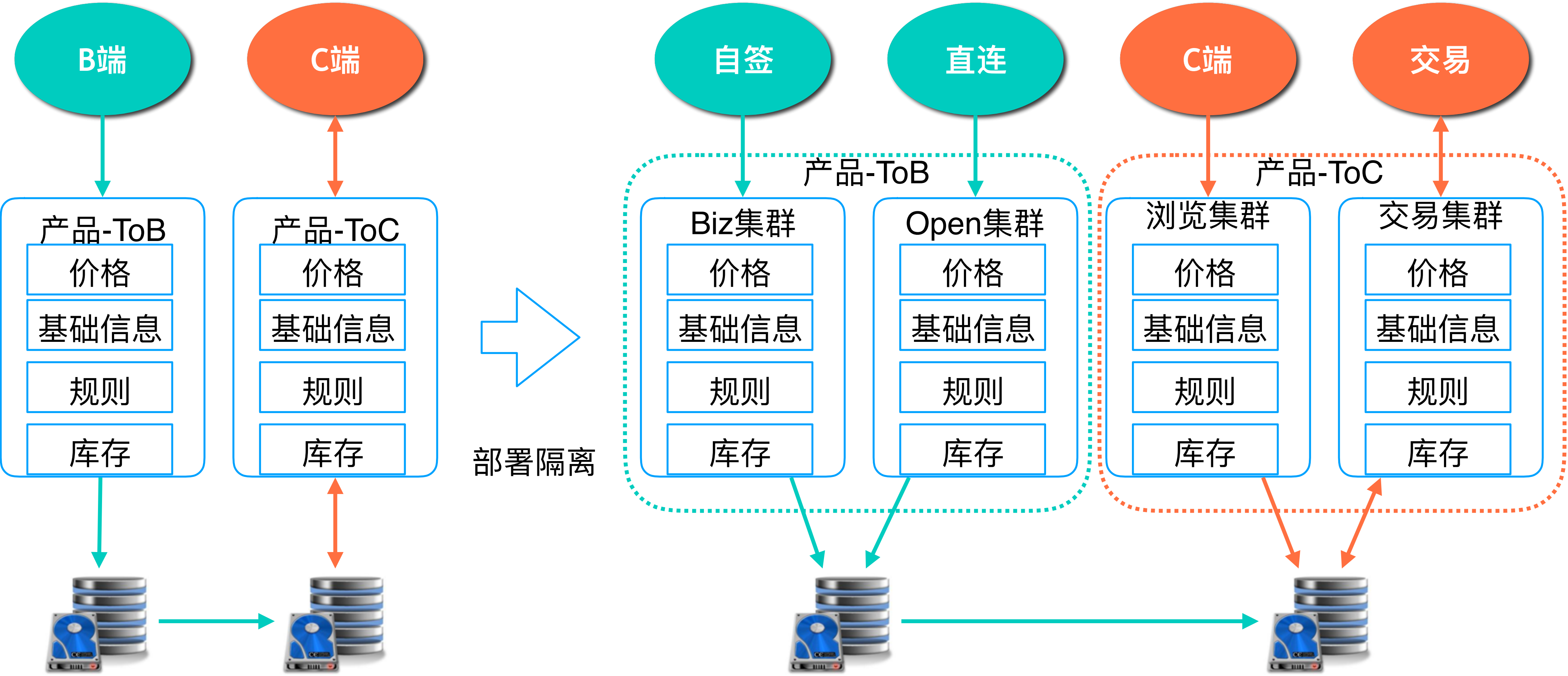
可用性管理 - 应对方案 - 控制影响

隔离 - 业务拆分



可用性管理 - 应对方案 - 控制影响

隔离 - 部署隔离



可用性管理－应对方案－控制影响

交互隔离

依赖强度			
强	超时	超时 有损降级 紧急预案	核心
	熔断降级	熔断降级	
		业务重要性	非核心

- 超时：按业务要求细化超时时间，保证依赖服务不拖垮自己
- 熔断：不拖慢服务
- 降级：启用备用方案

销售规则

商家详情

预订 [8] 预订成功后100%有房

盘多了测
不含早
不可取消
¥108
LH测试间夜减 已返2元

预订

测试1
有窗 | 不含早
入住当天18点前免费取消
¥108
LH测试间夜减 已返2元

测试3
30m² | 有窗 | 不含早
不可取消
¥108
LH测试间夜减 已返2元

测试通用房-多门店
不含早
入住当天18点前免费取消
¥108

有损降级

商家详情

预订 [8] 预订成功后100%有房

盘多了测
不可取消
¥108
LH测试间夜减 已返2元

预订

测试1
有窗
不可取消
¥108
LH测试间夜减 已返2元

测试3
30m² | 有窗
不可取消
¥108
LH测试间夜减 已返2元

测试通用房-多门店
不可取消
¥108

可用性管理－应对方案－控制影响

其它隔离

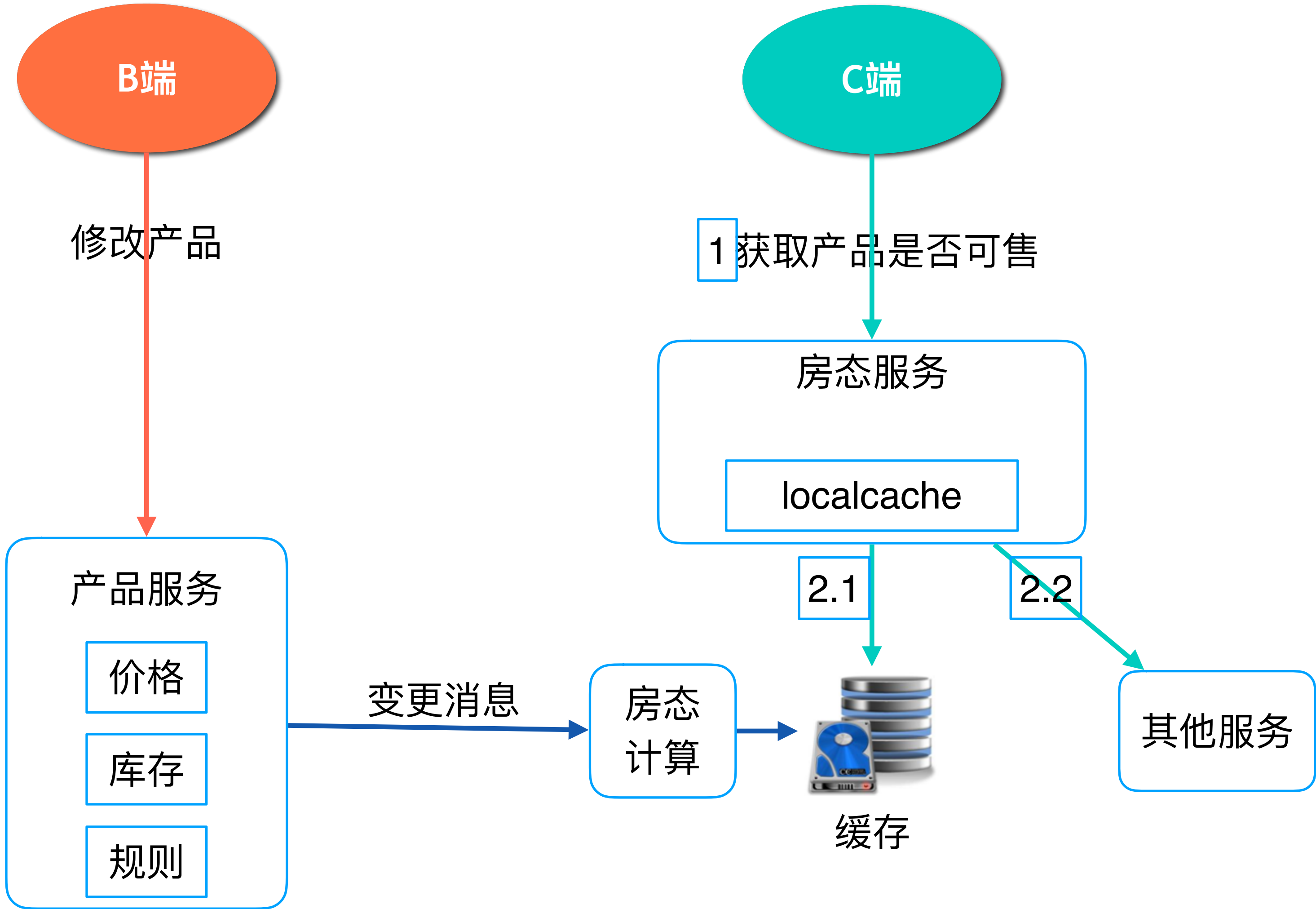
- 线程池隔离
 - 核心流程与非核心隔离
- 动静隔离
 - 不变与动态变更隔离

。 。 。 。 。 。

可用性管理－应对方案－控制影响

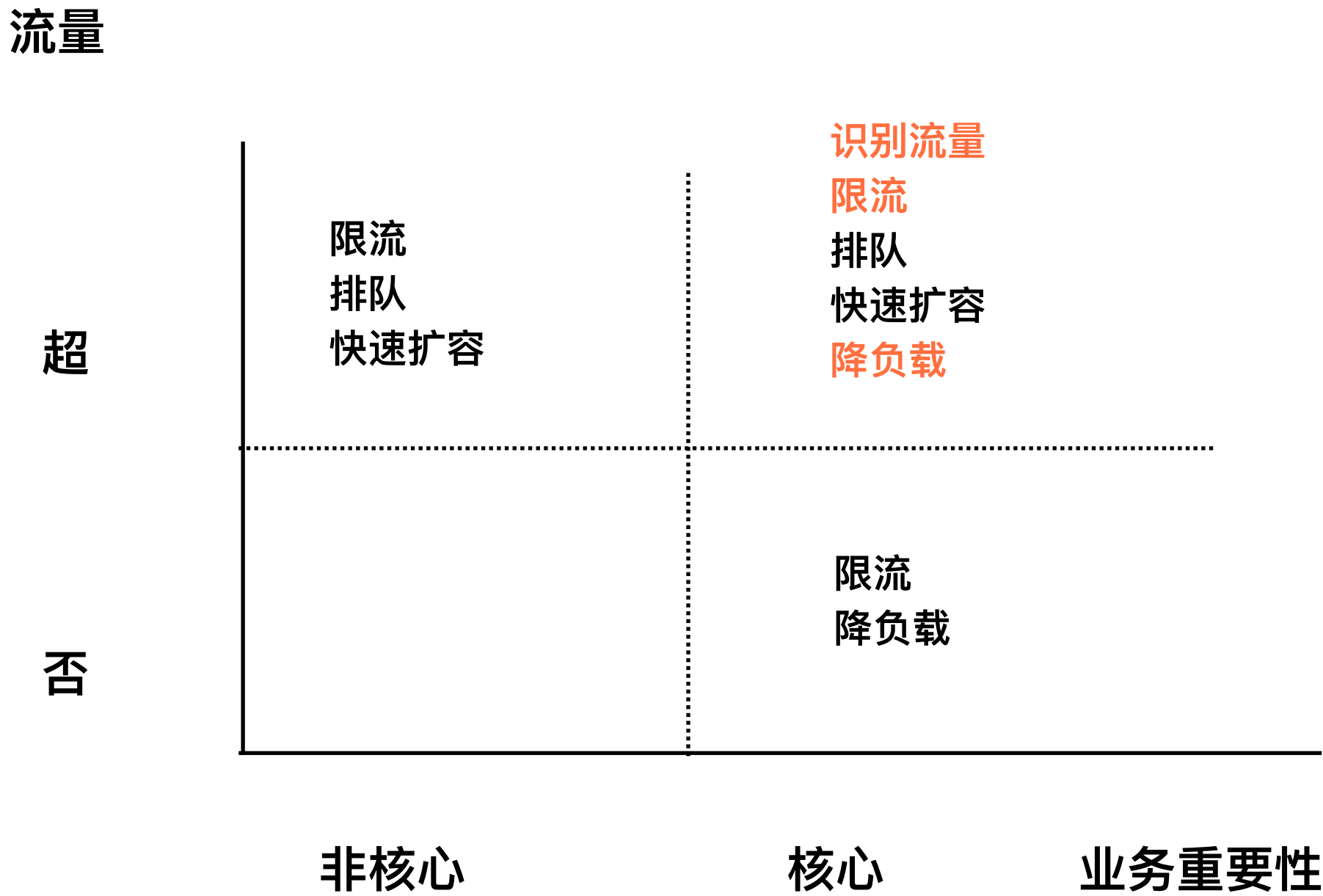
性能优化

- 异步化 (MQ)
- 并行化 (多线程)
- 本地化 (localCache)
- 预计算



可用性管理－应对方案－控制影响

过载保护



识别流量

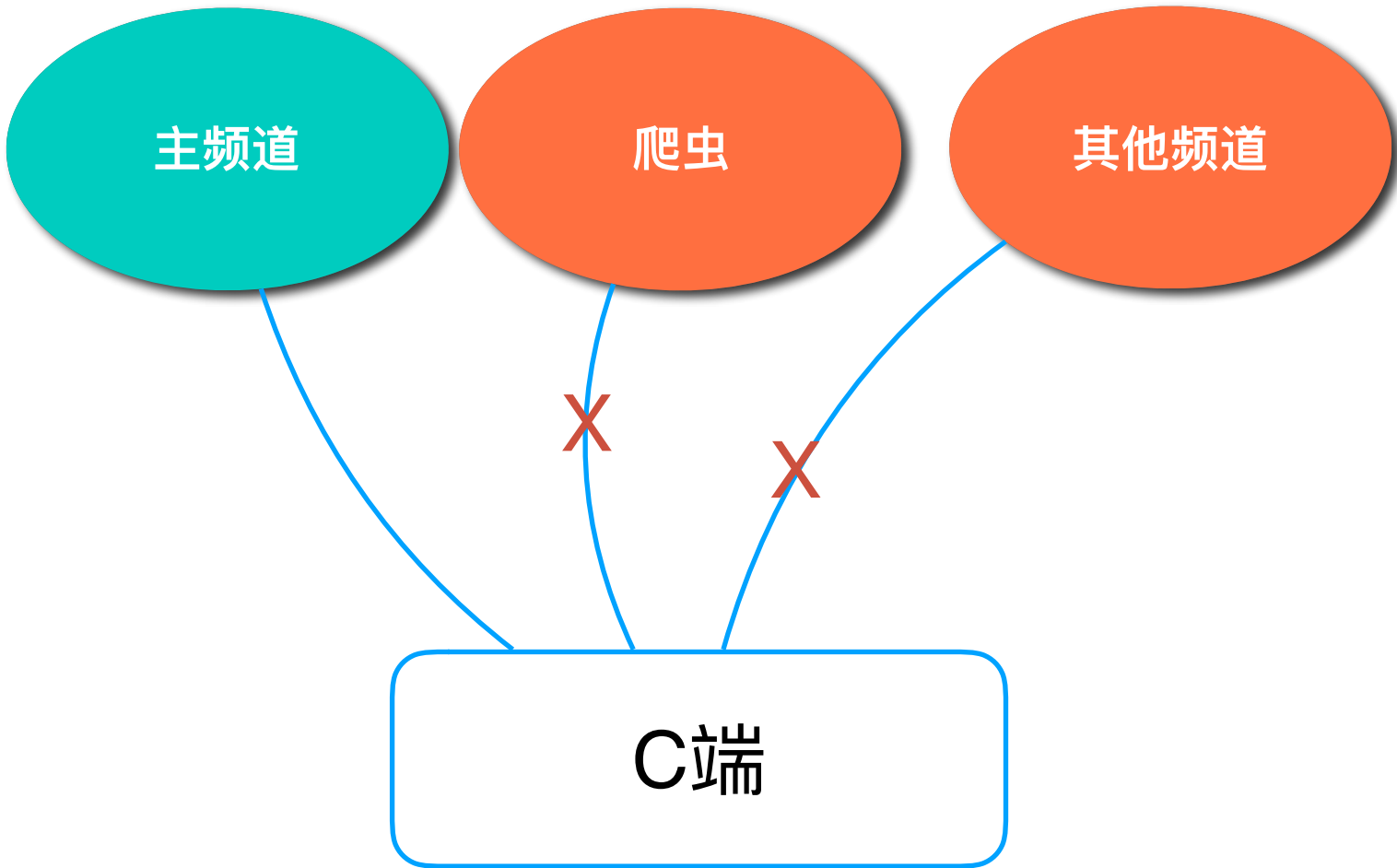
功能价值

限流策略

来源限流
接口限流
容量限流
单机限流

降负载

降级
熔断



可用性管理

应对方案



可用性管理—应对方案—可能性

消除单点

部署

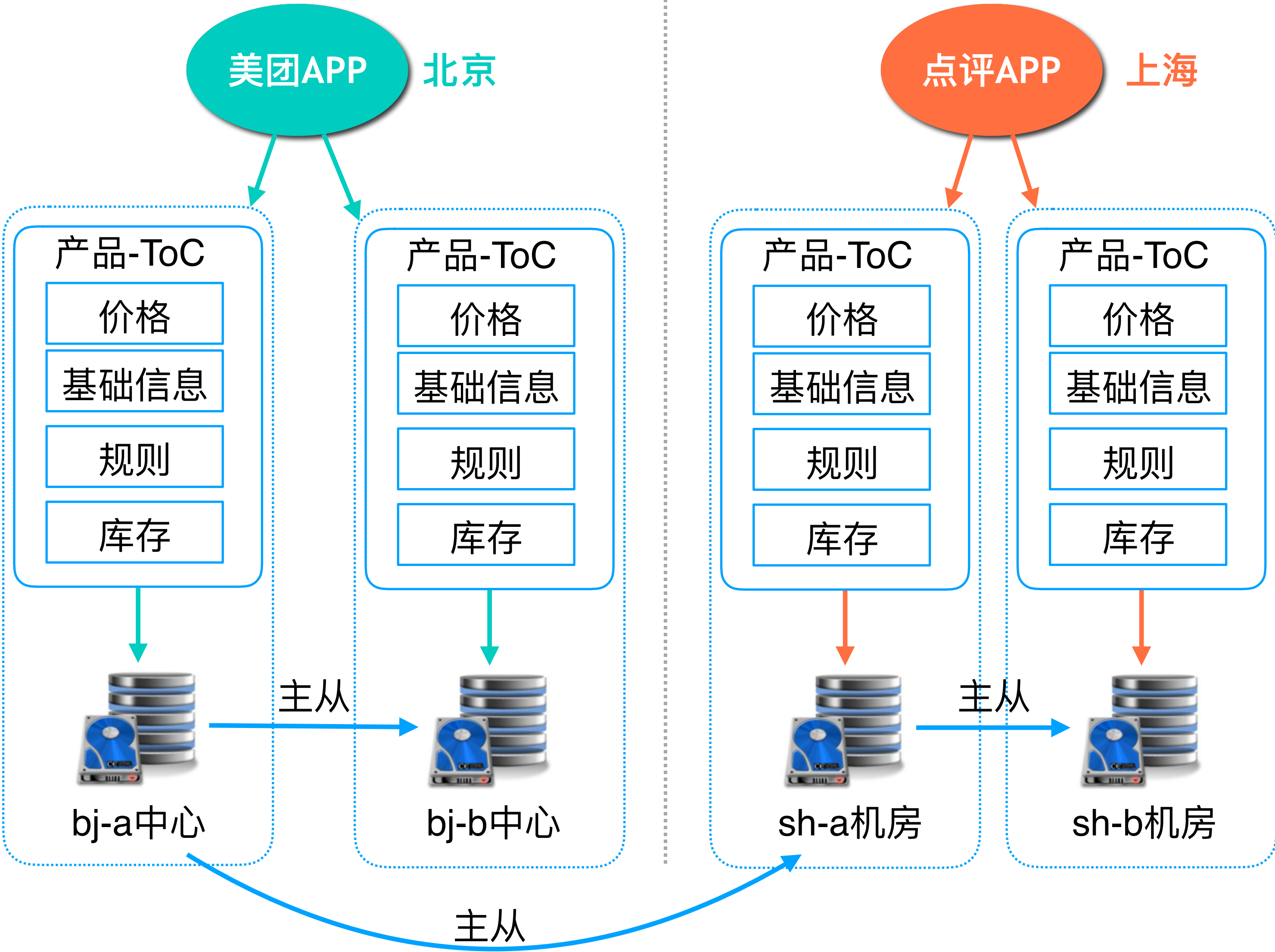
- 冗余N+1
- 多中心
- 多机柜
- 异地多中心
- 多机房

服务

无状态，可扩展，可失效转移

策略

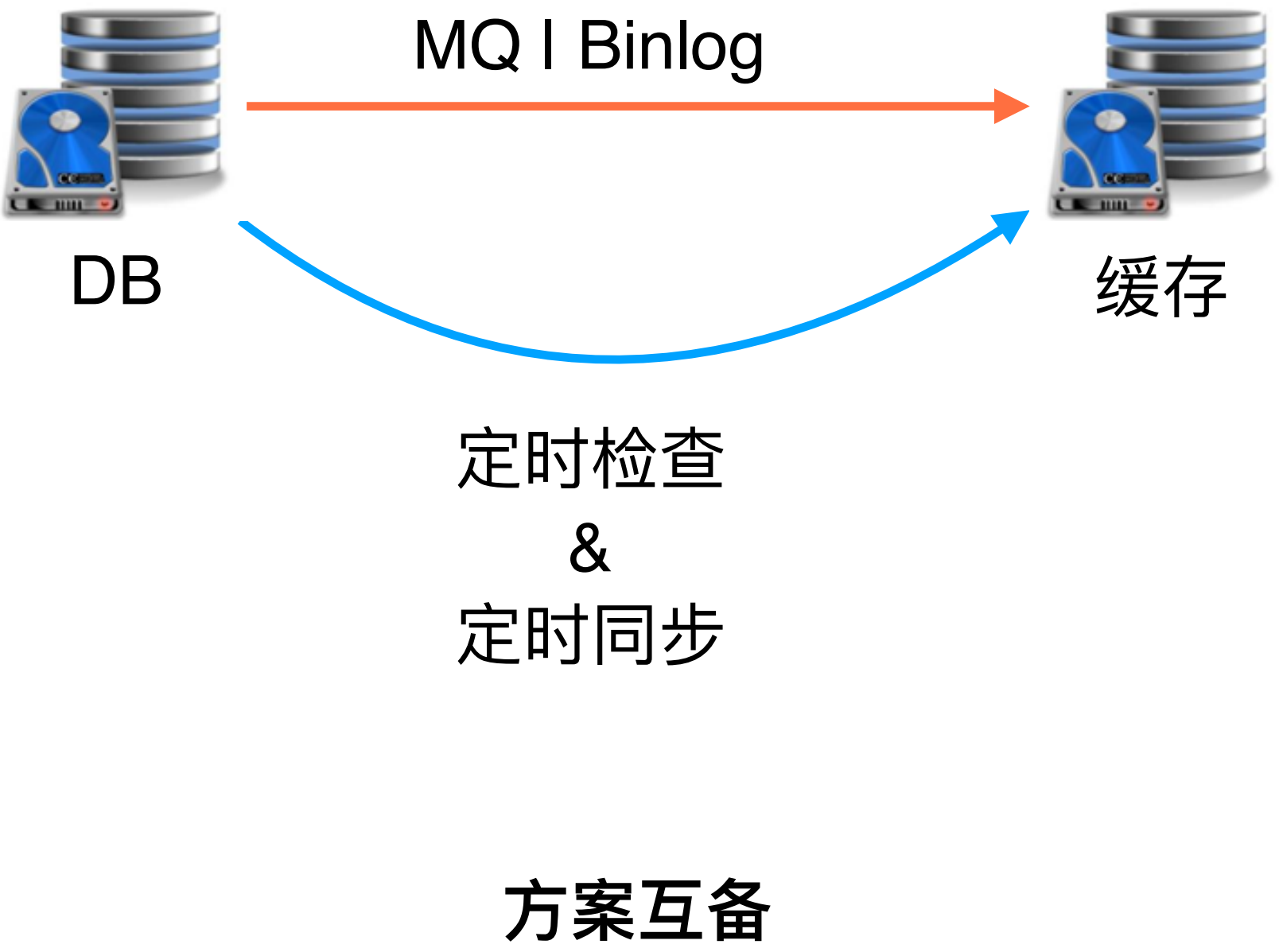
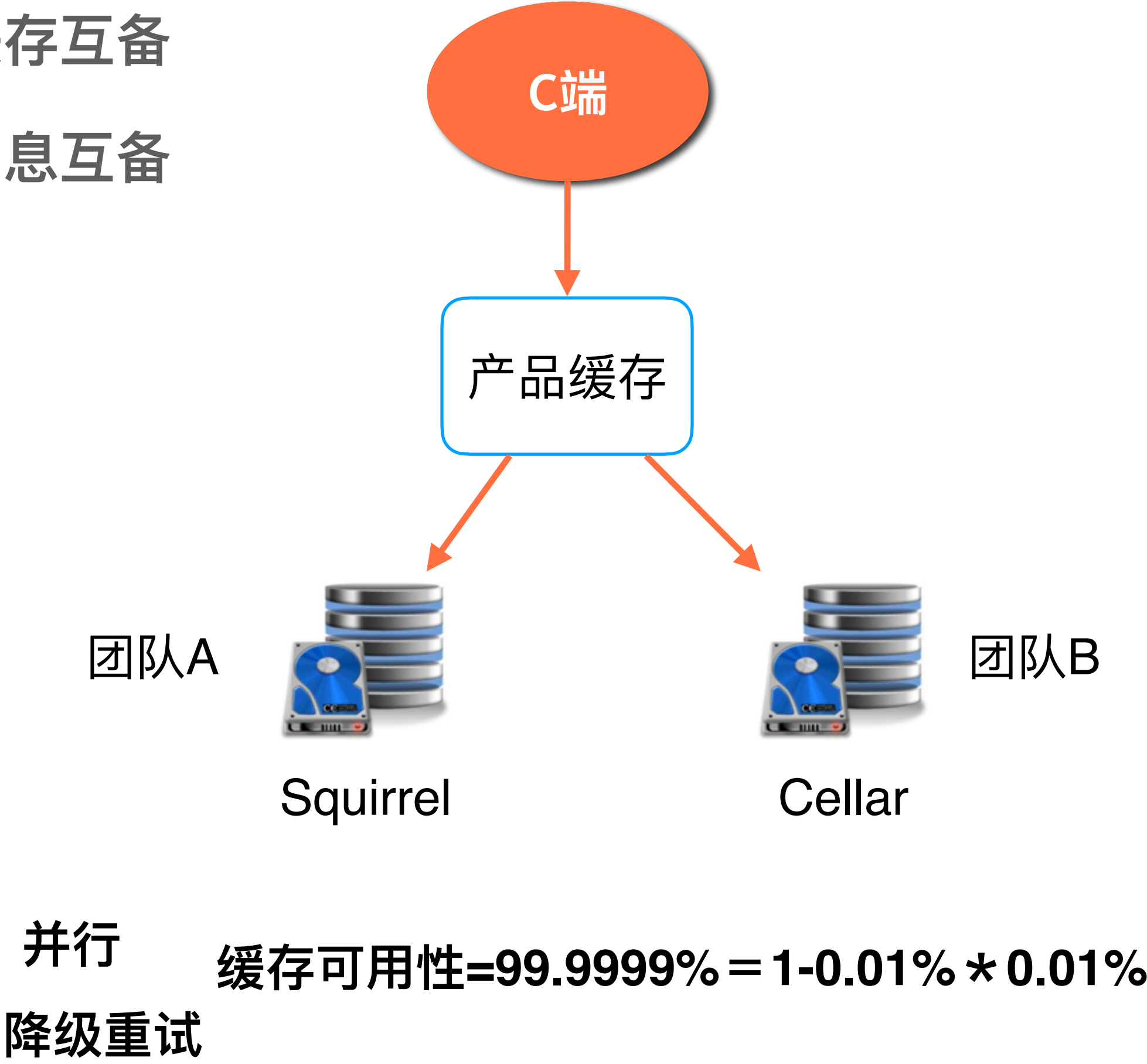
核心连路服务		
...		
业务服务	多实例	异地多机房多中心
基础组件	多实例	异地多机房多中心
DB&缓存	多实例	异地多机房多中心
	非核心	核心
	业务重要性	



可用性管理—应对方案—可能性

方案互备

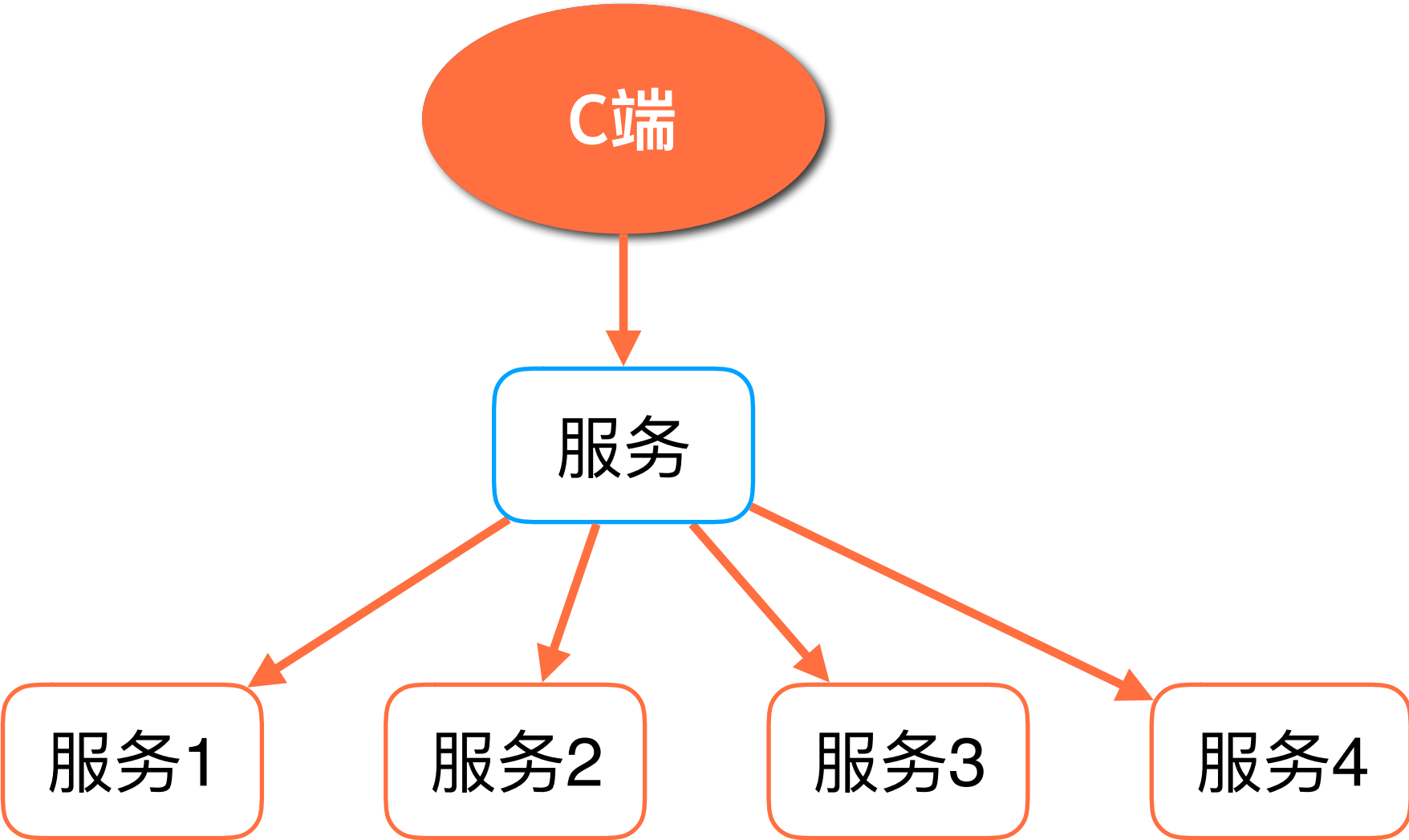
- 缓存互备
- 消息互备



可用性管理－应对方案－可能性

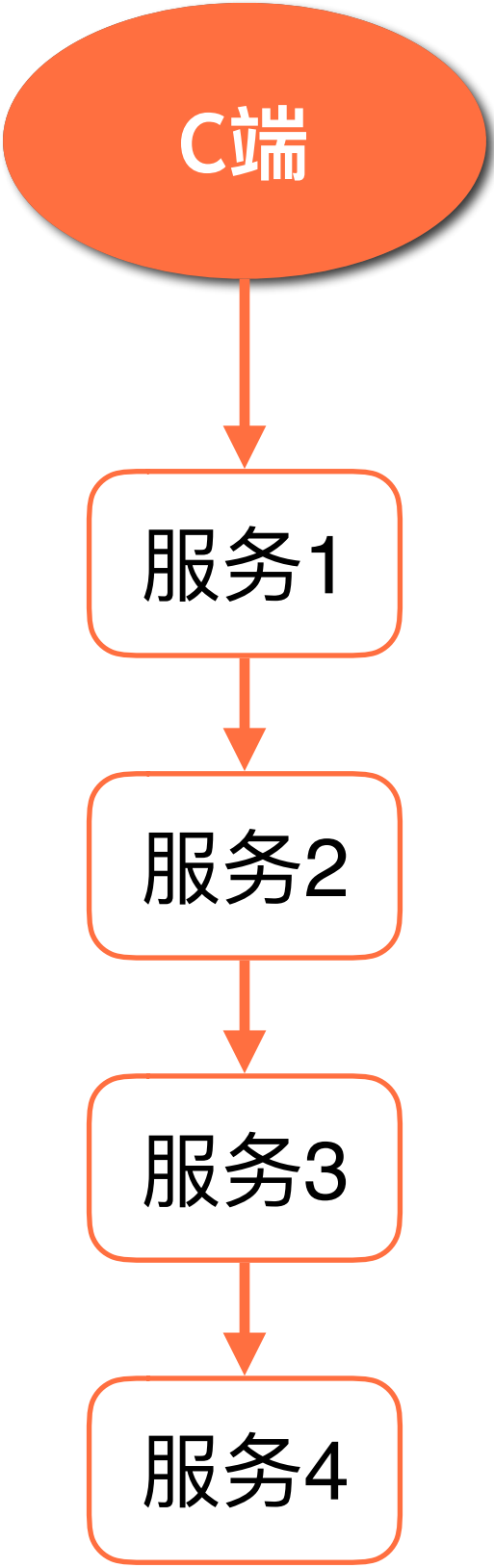
控制复杂性

- 多 VS 少
- 长 VS 短



可用性 = $0.9999^4 = 0.9996 < 0.9999$

减少核心依赖



缩短核心调用链

可用性管理

应对方案



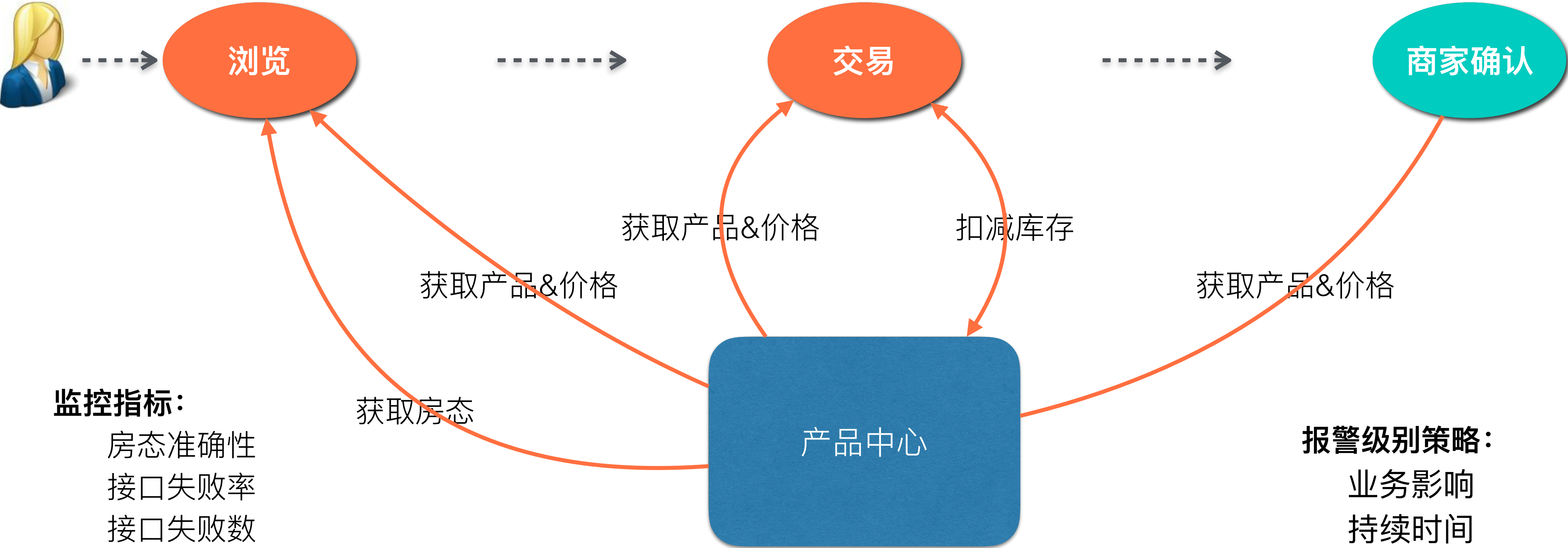
可用性管理－应对方案－可能性－影响时长

监控&报警策略

报警级别	P0	P1	P2
监控范围	业务指标 服务指标 硬件&容器	业务指标 服务指标 硬件&容器	服务指标 硬件&容器
通知范围	全体组员 维护人+关注人	维护人+关注人	维护人+关注人
通知渠道	短信 大象 邮件 电话（非工作日）	短信 大象 邮件	大象 邮件
应对方案	5分钟相应 大组通报 先止损再定位 1天内定位根因 复盘形成TODO	10分钟相应 小组通报 判定是否升级P0 1天内定位根因 形成TODO（3天内）	30分钟相应 判定是否升级P0,P1 记录问题和原因 Review（1周）

可用性管理－应对方案－可能性－影响时长

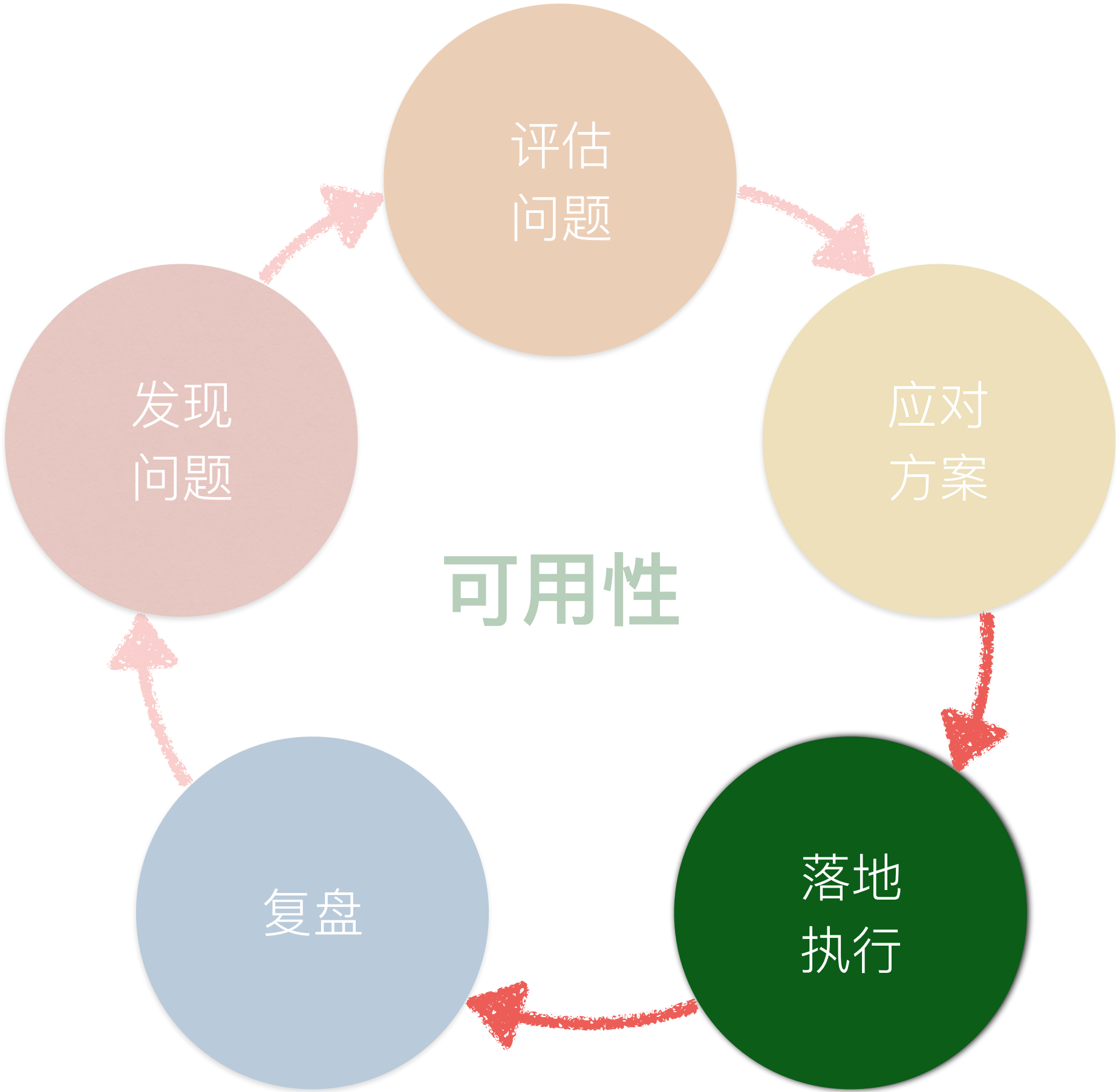
监控&报警设置



- 监控指标：**
- 房态准确性
 - 接口失败率
 - 接口失败数
 - 接口响应时间
 - 接口成功数对比基线的浮动

- 报警级别策略：**
- 业务影响
 - 持续时间
 - 波动幅度

可用性管理
评估问题



可用性管理－落地执行

需求管理



需求管理

需求反讲



研发质量

5PD以上设计评审

双人Review

新代码单测80%行覆盖率

回归测试

合并代码Sonar扫描

上线前读接口Diff

线下泳道环境



发布管理

发布计划Review

灰度一台机器

按机房发布

低峰发布

节前提前3天封板

需求合并发布

灰度&滚动发布



运维工具

根据Trace打印日至定位问题

数据库与缓存数据一致性监测

快速重新同步数据

满房原因定位

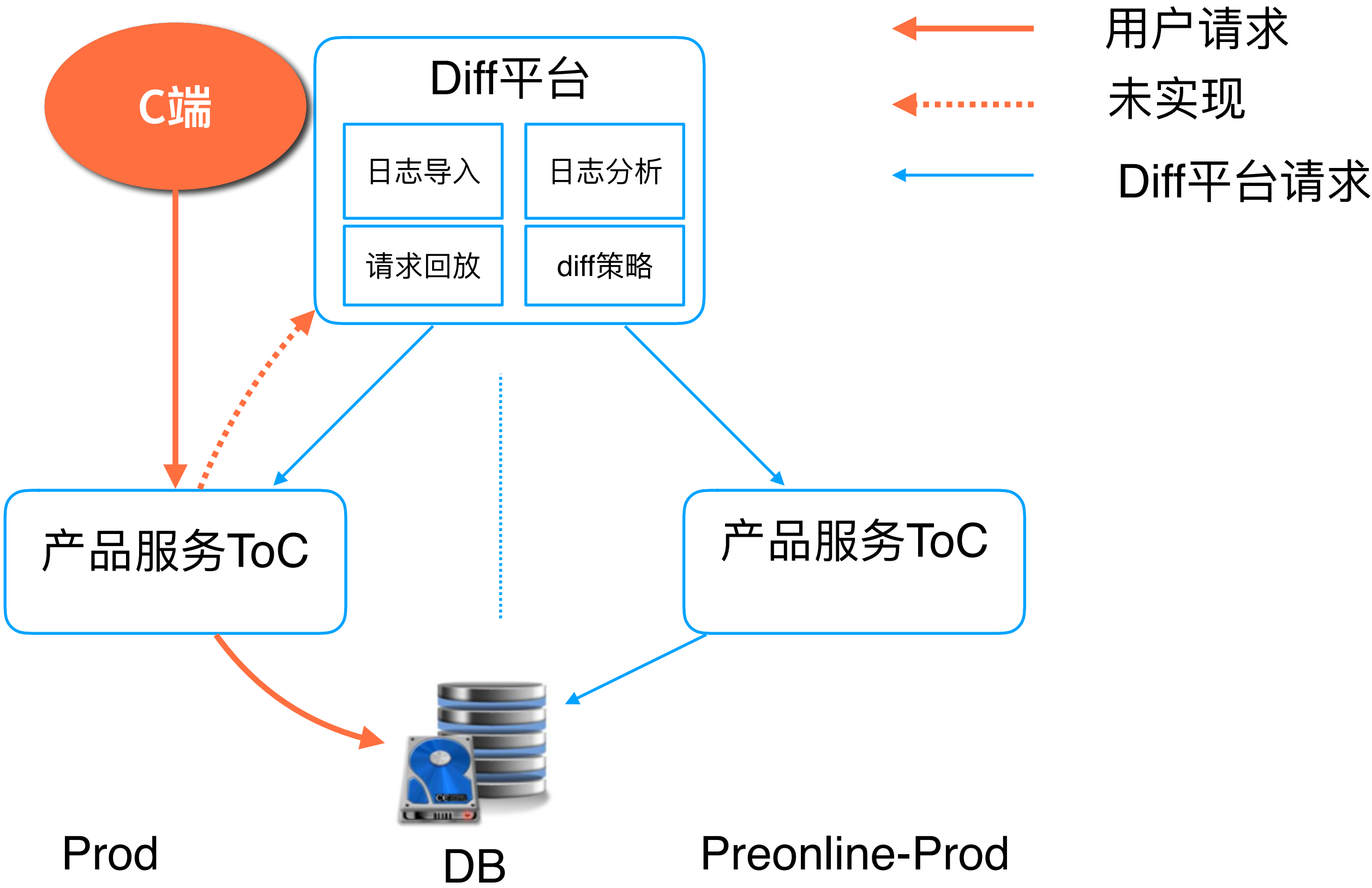
可用性管理－研发质量

发布环境

	环境名称	机器	数据库	用途
线上	Prod	online	online	生产环境
	Preonline-Prod	online	online	压测，灰度发布
	Preonline-Stage	online-stage	online	回归测试
线下	Stage	online-stage	offline	直连业务测试
	Test	offline	offline	测试环境

可用性管理－研发质量

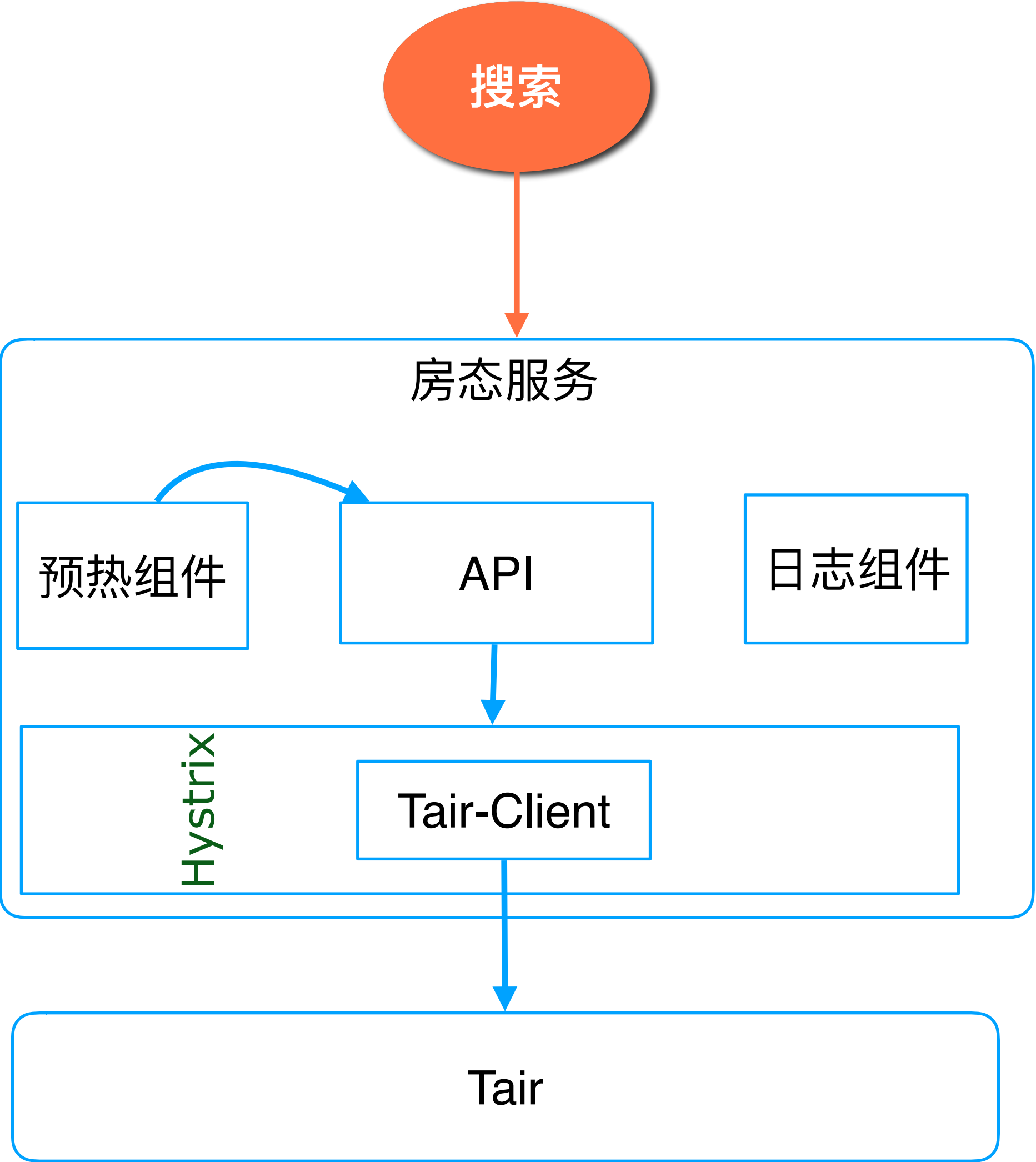
Diff平台



上线前读接口Diff

可用性管理 - 发布管理

启动优化



← 预热请求
← 业务请求

- 接入OCTO平滑发布
- Hystrix首次初始化
- Tair 连接池初始化
- 模拟请求
- 清理无效日志
- 异步打印Error日志栈

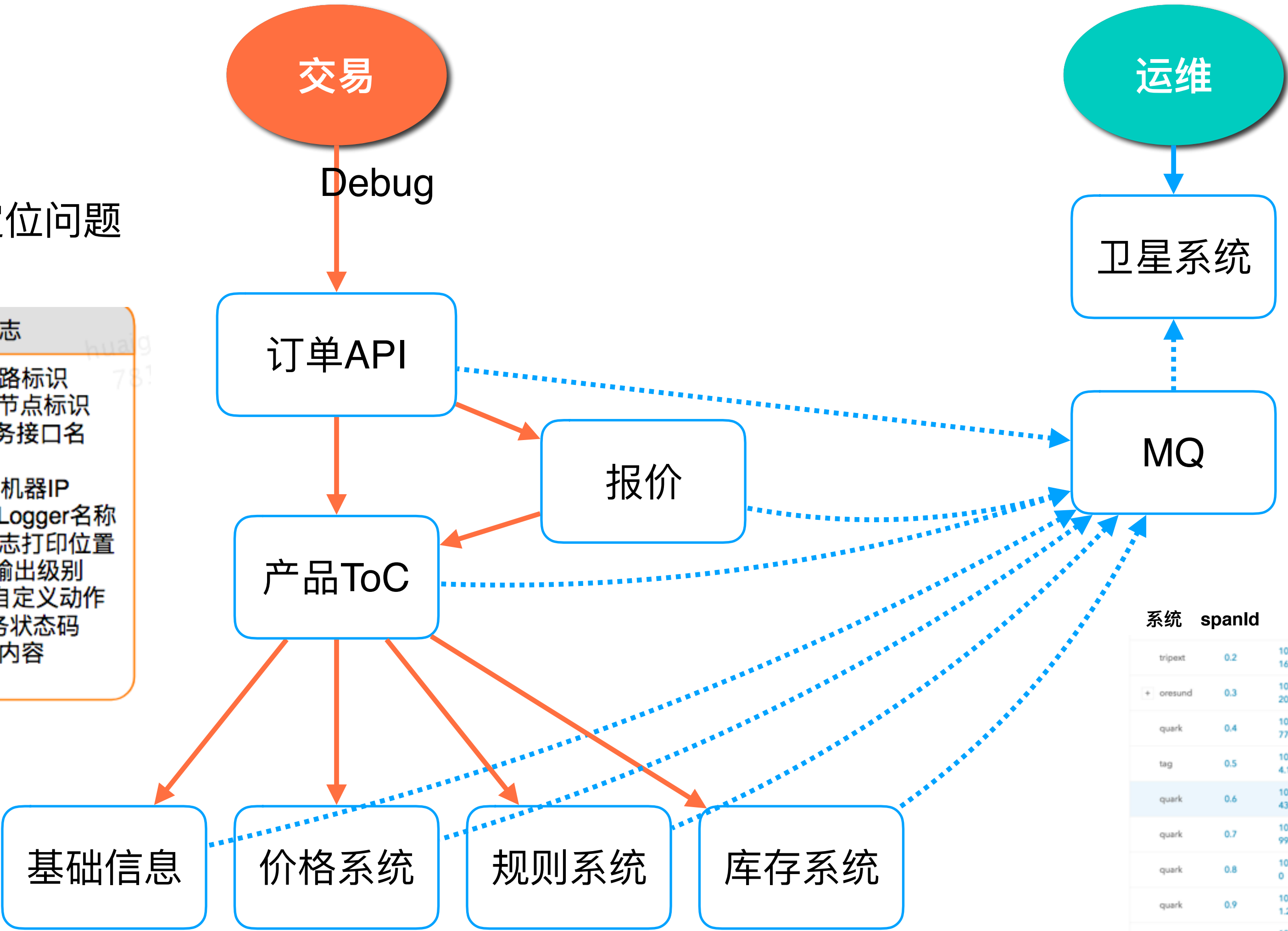
可用性管理 - 发布管理

定位问题

利用Trace 定位问题

统一日志

traceld: 全局链路标识
spanId: 链路中节点标识
spanName: 服务接口名
ip: 当前机器IP
clientIp: 调用方机器IP
logName: 业务Logger名称
logPos: 业务日志打印位置
logLevel: 日志输出级别
action: 业务方自定义动作
stateCode: 业务状态码
message: 日志内容



系统	spanId	方法名
tripext	0.2	10.4.208.168 3 1 mtthri ft S RpcTdcPoiImageService.getImagesByPoiIdAndParam:3ms
+ oresund	0.3	10.4.212.208 77 1 mtthri ft S GetGoodsInfoByGoodIdModelService.queryGoodsInfoList:77ms
quark	0.4	10.4.92.177 2 1 mtthri ft S PoiRoomService.getRoomBaseByRoomIds:2ms
tag	0.5	10.32.10.4.122 3 2 mtthri ft S RPCTagService.getGoodsTagItemList:3ms
quark	0.6	10.4.88.243 3 2 mtthri ft S RealRoomService.getEquipRealRoomDetail:3ms
quark	0.7	10.32.75.99 2 1 mtthri ft S PoiRoomService.getRoomBedByRoomIds:2ms
quark	0.8	10.4.54.60 3 2 mtthri ft S PoiRoomService.getRoomExtendByRoomIds:3ms
quark	0.9	10.32.14.1.226 3 1 mtthri ft S PoiExtendService.getAllPoiExtend:3ms
- priceapi	0.10	10.21.19.7.192 39 1 mtthri ft S PriceCalcService.queryProductPrice:39ms

纲要

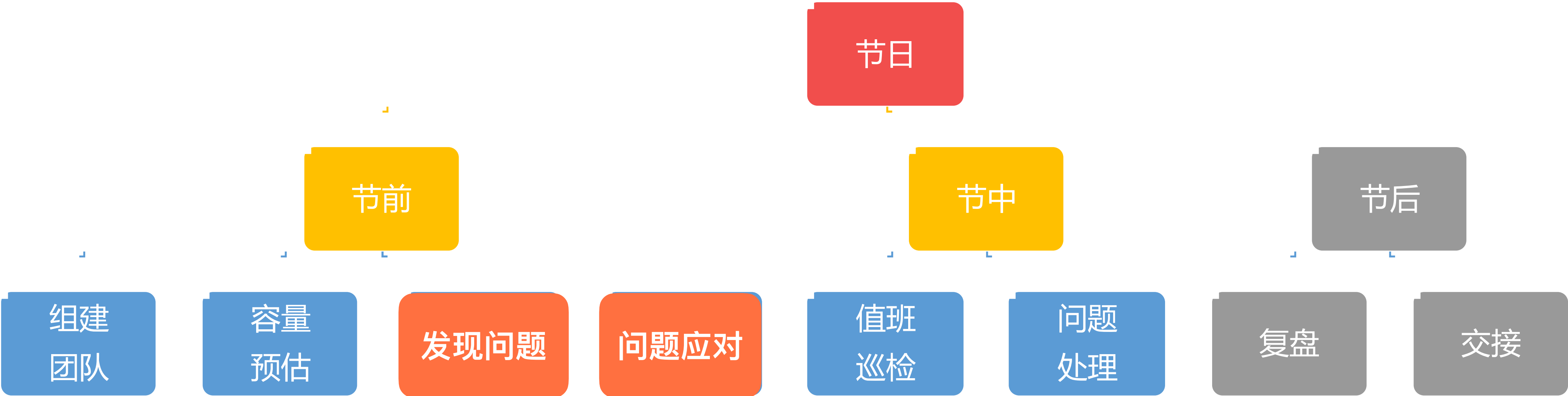
产品简介

可用性管理

假日应对

节假日实战

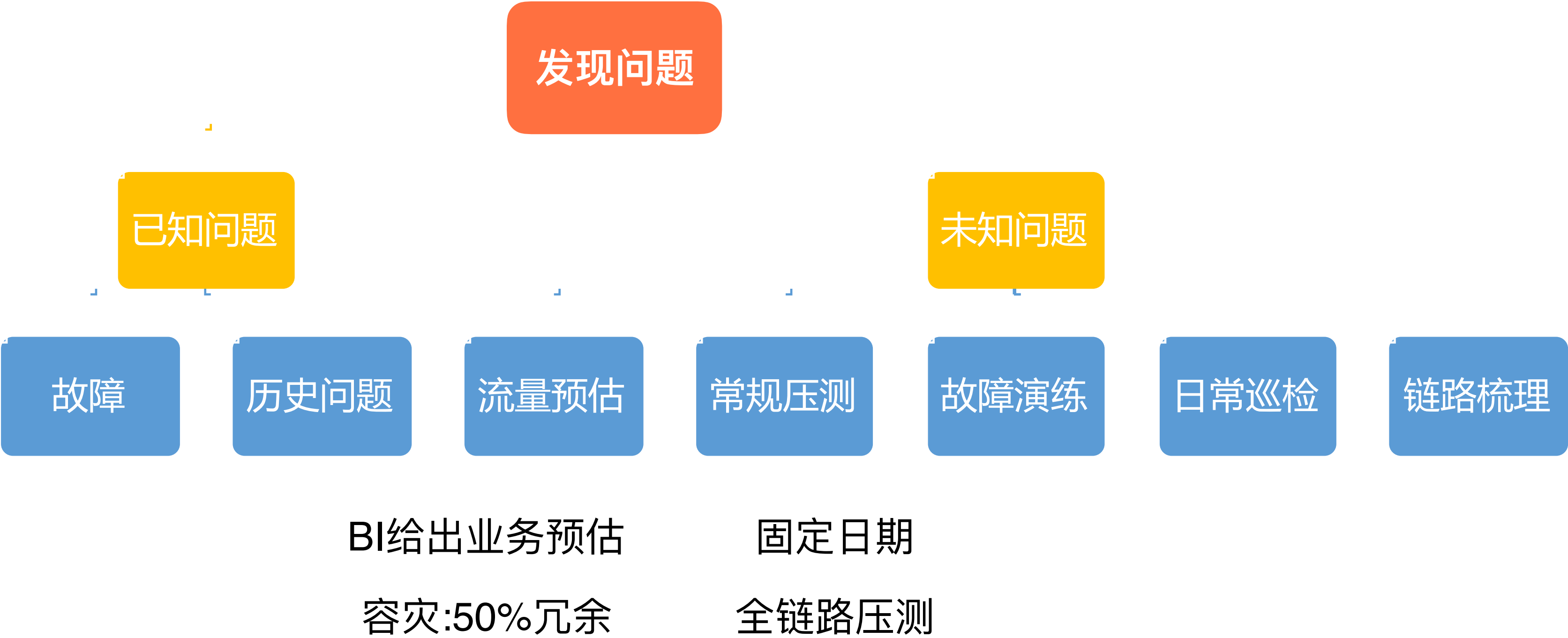
节日计划



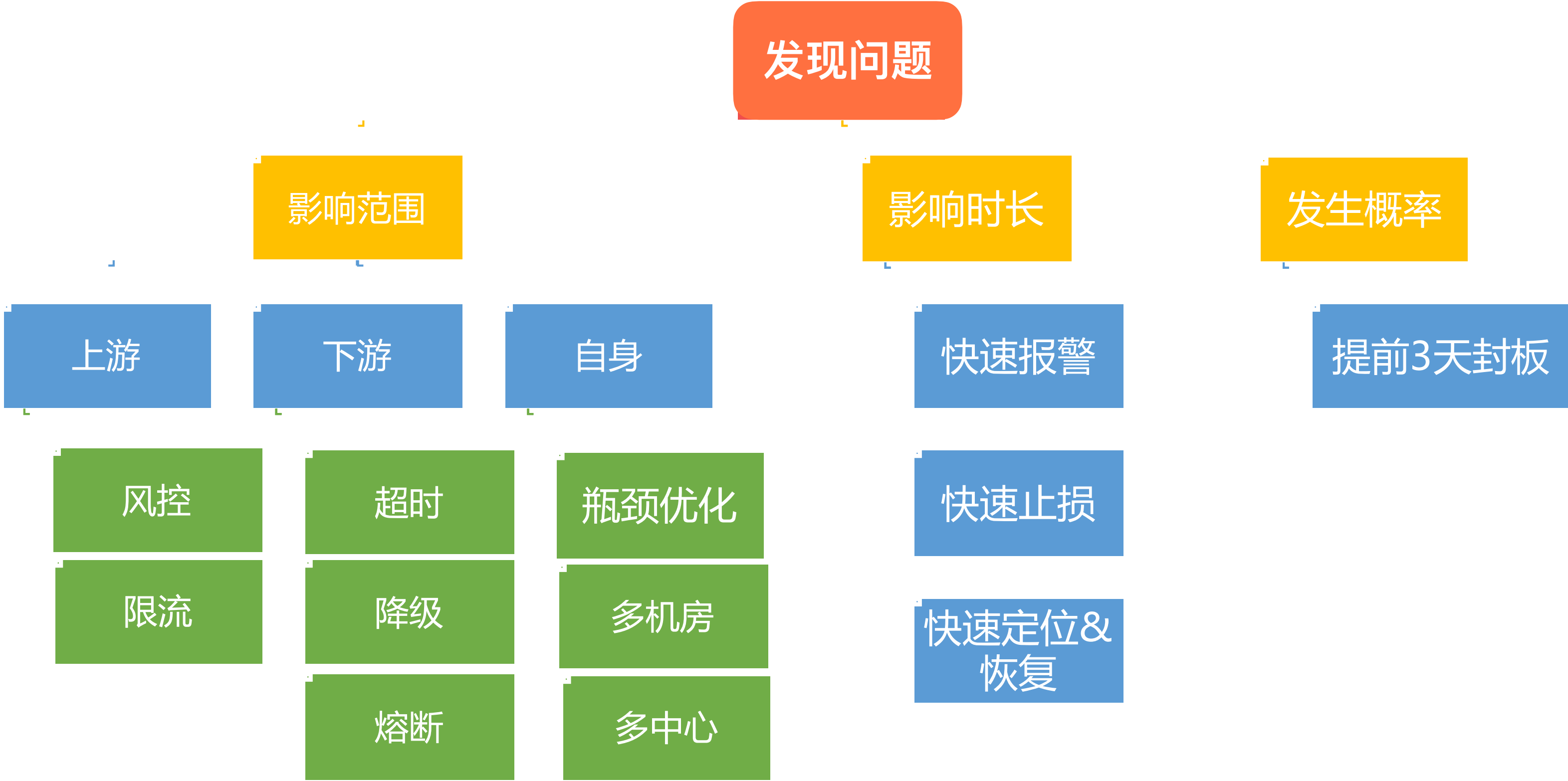
稳定性指挥官 历史峰值1.5倍

各组出稳定性负责人

节假日实战
发现问题



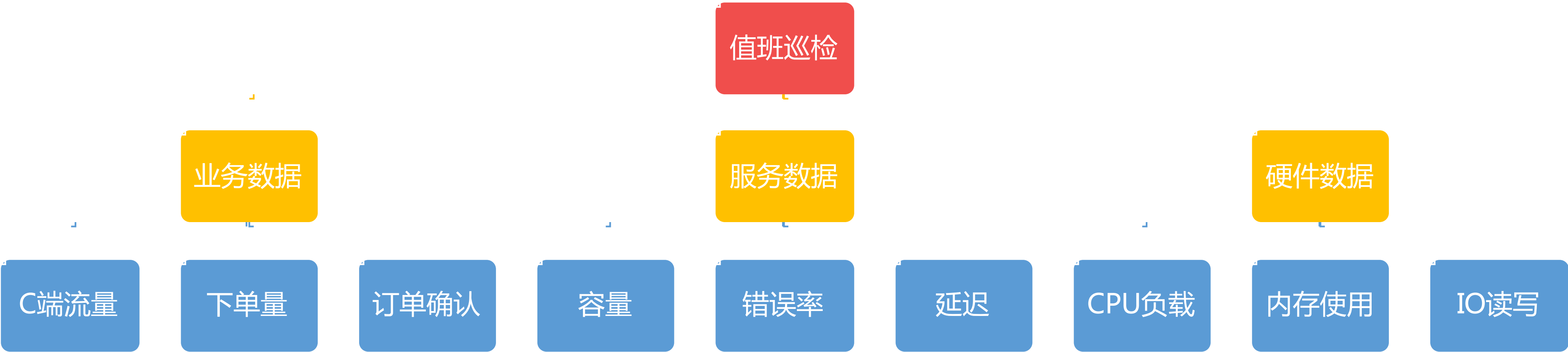
节假日实战
问题应对



接入UAC系统 调整线程池大小上海侧多机房部署 一键降级平台接入
对恶意流量限流
对价值较小流量限流

节假日实战

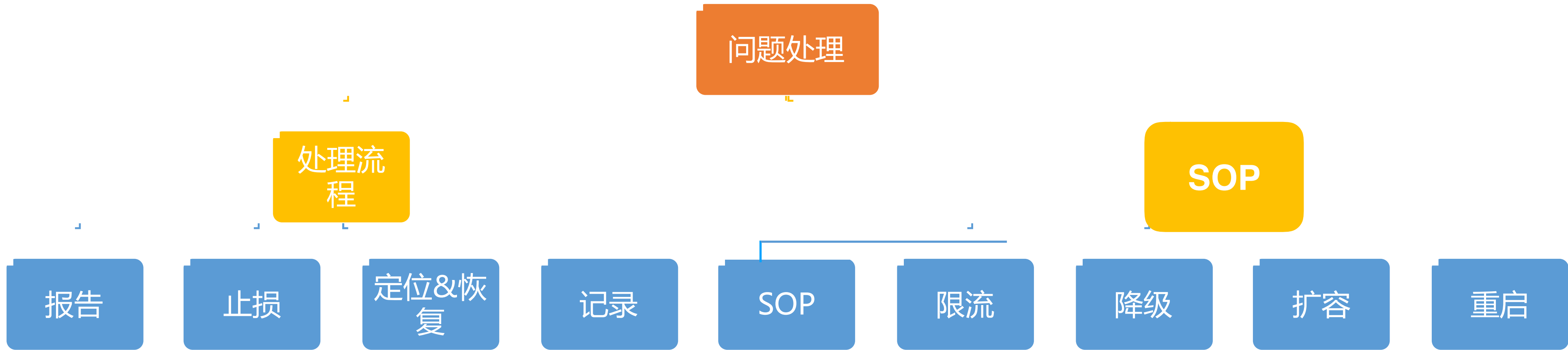
值班巡检



13:00,18:00,22:00定时巡检

容量占比
响应时间波动
系统Load值

节假日实战
问题处理



问题原则

1. 及时止损：例如上线导致故障，第一时间应该快速回滚，及时止损，而不是追问题改代码
2. 无脑预案：因为在故障发生时，最困难的就是知道“发生了啥”，“该干啥”，
3. 工具化、平台化：预案不能依赖特定的人，例如降级要有一键开关，而不是需要某个人去x平台改个参数，y平台改个超时
4. 及时通报：及时对外同步故障的影响，跟进人，修复进度，便于相关团队知晓、协助
5. 经常演练：没有经过经常性演练的预案==没有预案

Q&A

附录

参考资料

- 复盘
- 构建高可用系统-之稳定性篇
- 酒旅稳定性建议方案
- SRE++Google运维解密

附录

项目复盘

个人复盘教练工作底稿			
过程	提问句式	操作指引&提示	谈话要点记录
开场	XX, 你好! 我是XX, 今天我们来做一次个人复盘。你知道复盘的目的是什么吗?	目的是建立亲和, 语气应亲切、柔和、自然	
	嗯, 就像你所说, 复盘的目的在于.....	复述被教练对象的核心观点, 重申复盘的目的在于学习	
	那么, 你今天想对什么进行复盘呢?	个人复盘的议题不宜太大, 不要相隔很久。否则, 建议更换主题	
	【确定复盘的主题】		
回顾目标	通过本次复盘, 你希望能有哪些收获?	明确本次复盘的目的 (而非复盘主题的目的)	
	好的, 我们本次复盘可能需要____分钟。今天的谈话只限于我们两人之间, 希望你畅所欲言, 不要有任何顾虑。我们可以开始了吗?	使被教练者消除顾虑, 敞开心扉、畅所欲言	
	首先, 请你回想一下, 你当初做这件事的目的是什么?	注意区分目的与目标; 目的是做某件事的动机或想实现的价值; 目标则应尽可能明确、具体、可衡量。	
	你觉得做到什么程度, 就算达成了你的目的? 也就是说, 有没有一些具体的目标呢?	如果目的不清或没有目标, 应引导其意识到做事情应事先制定目标, 并指导其事后梳理一下思路: 若回到当时, 比较科学合理的目标应该是什么?	
评估结果	实际结果是怎样的?	实际结果要具体, 最好有数据或具体的事实、行为表现作为支撑, 不能只是个人主观感受或笼统的评价。 注意: 在这里, 只讲结果即可, 不必详述过程, 也无需探讨这个结果是怎么形成的	
	对照目标, 你觉得哪些地方做得好?	评价要客观、具体。 此处只是陈述观点即可, 不必展开原因分析。 若教练了解复盘主题相关的信息, 可以通过提问引导被教练对象对自己的观点进行反思, 但不建议给出自己的看法, 更不得强加于人	
	哪些地方还存在不足?		

可用性管理思路 故障复盘

莎士比亚新韵文事故总结（事故编号 #465）

日期：2015-10-21

作者：jennifer、martym、agoogler

目前状态：已经终稿，待办事项正在进行中

摘要：莎士比亚搜索服务出现 66 分钟的故障，由于新发现了一篇韵文导致用户流量暴涨。

事故影响^{注 1}：预计 12.1 亿个请求丢失，没有损失任何收入。

根源问题^{注 2}：由于异常的高负载情况以及搜索词语在 Shakespeare Corpus 中不存在时的一项资源泄露导致了连锁故障的发生。新发现的韵文使用了一个之前从未在莎士比亚文献中出现的词语，这恰恰是用户大量搜索的关键词。在日常情况下，这种资源泄露导致的任务崩溃现象由于出现非常不频繁而没有被注意到。

触发条件：潜伏性的 Bug 被大量上涨流量所触发。

解决方案：将流量导向滞洪集群，同时增加了 10 倍容量以应对连锁故障。部署了更新过的索引，绕过了潜在的 Bug。在公众对新韵文的兴趣消退之前，保持额外的容量。资源泄露的问题已经被找到，并且修复已经上线。

检测：Borgmon 检测到大量 HTTP 500 的情况，向 on-call 发送了紧急警报。

待办事项：^{注 3}

待办事项	类型	负责人	Bug
更新运维手册（Playbook），加入有关应对连锁故障的指示	缓解	jennifer	n/a DONE
利用 flux capacitor 在多个集群中均衡负载	预防	martym	Bug 5554823 TODO
在下次 DiRT 练习中进行连锁故障演习	流程	docbrown	n/a TODO
调查是否可以持续性地进行搜索更新	预防	jennifer	Bug 5554824 TODO
在搜索排序子系统中修复文件描述符的泄露	预防	agoogler	Bug 5554825 DONE
给莎士比亚搜索服务增加流量抛弃功能	预防	agoogler	Bug 5554826 TODO
增加回归测试，保障系统在收到致死请求时保持正常	预防	clarac	Bug 5554827 TODO
将更新过的搜索排序子系统部署到生产环境	预防	jennifer	n/a DONE
由于错误预算耗尽，停止更新生产环境直到 2015-11-2。或者寻求管理层对特殊情况的批准	其他	docbrown	n/a TODO

经验教训

做得好的地方：

- 监控系统在大量 HTTP 500（接近 100%）的情况下快速发出了紧急警报
- 在所有集群中快速更新了 Shakespeare Corpus

做得不好的地方：

- 对连锁故障的处理不够熟练
- 由于这次大幅度的流量增长（几乎全部请求都失败了），导致我们超过了可用性错误预算数个数量级

幸运的因素^{注 4}

- 莎士比亚速的邮件列表里刚好有一份新的韵文文本
- 服务器日志中包括了指出文件描述符耗尽问题导致崩溃的栈跟踪
- 致死请求（query-of-death）通过推送新的索引关键词就解决了

附录

故障复盘

故障描述
故障等级：S(必填，参考故障等级定义)。事故定级规范 境内度假事业部线上故障处理流程及定级标准 酒店线上故障定级及处理说明 V1.0 （以SRE推荐的酒店业务线内部的标准进行量化。响应时间部分另行讨论）
故障类型：流程规范类
起止时间：2016-07-16 09:22 - 2015-06-16 09:33(必填，需要精确到分)
责任组/人：酒店旅游业务测试组
故障详情
事故情况描述，简明扼要的说明什么时间，什么直接原因，导致了本次的线上事故。
故障处理过程
这里描述事故的来龙去脉。需要包含整个事故经过的时间、事件。在什么时间节点什么人做了什么事。 包括：事故什么时候开始发生 包括：什么人什么时间报告故障 包括：如何追查原因 包括：什么时间定位到原因 包括：问题如何解决 包括：什么时候故障得以解决
影响与损失
依照所属故障定级的指标，写清楚此次事故造成的影响和损失具体数据。 比如后台对账结算故障，需写明造成的打款金额错误有多少、打款账户错误有多少、打款延时有多少。 如果存在新业务，需及时调整故障定级规范。明确指标。
解决方案
如果为代码问题，需给出具体的代码变更及说明。 如果为架构设计问题，需给出具体的架构调整，应急处理总结wiki，其中应包含具体的架构设计图。
原因分析
线上故障原因分析至关重要，如果不能追根溯源找到问题的关键，改进、避免就无从谈起 美团线上故障统一采用5 Whys分析法，相关介绍参见5 whys分析法在美国工程师中的实践，使用“5 whys”调查方法来建立一个通向根本原因的原因/效果关系链。 1. 为什么出现这个故障？ 2. 为什么...
故障总结与改进
首先总结本次故障的根本原因，其次，围绕本次故障的根本原因，制定改进措施。 从事前预防、及早发现、快速定位、举一反三等角度，认真思考改进措施。 改进事项应避免空洞无物，确切针对原因分析中的问题，建立对应的改进事项，符合SMART原则，明确（与故障原因相关）、具体（明确的改进产物，如代码修正，checklist）、可执行（符合我们现有的环境）、可衡量（有明确的改进前和改进后对照的指标）、有确切完成时间（建议在1个月以内需要完成所有的后续todo）。
后续To-do
明确需要采取的行动、负责人、时间计划。 改进事项完成情况包括：未开始、进行中、已完成、取消 改进事项完成情况请 及时更新 ，定期统一回顾之前CaseStudy的改进情况

附录

发布环境

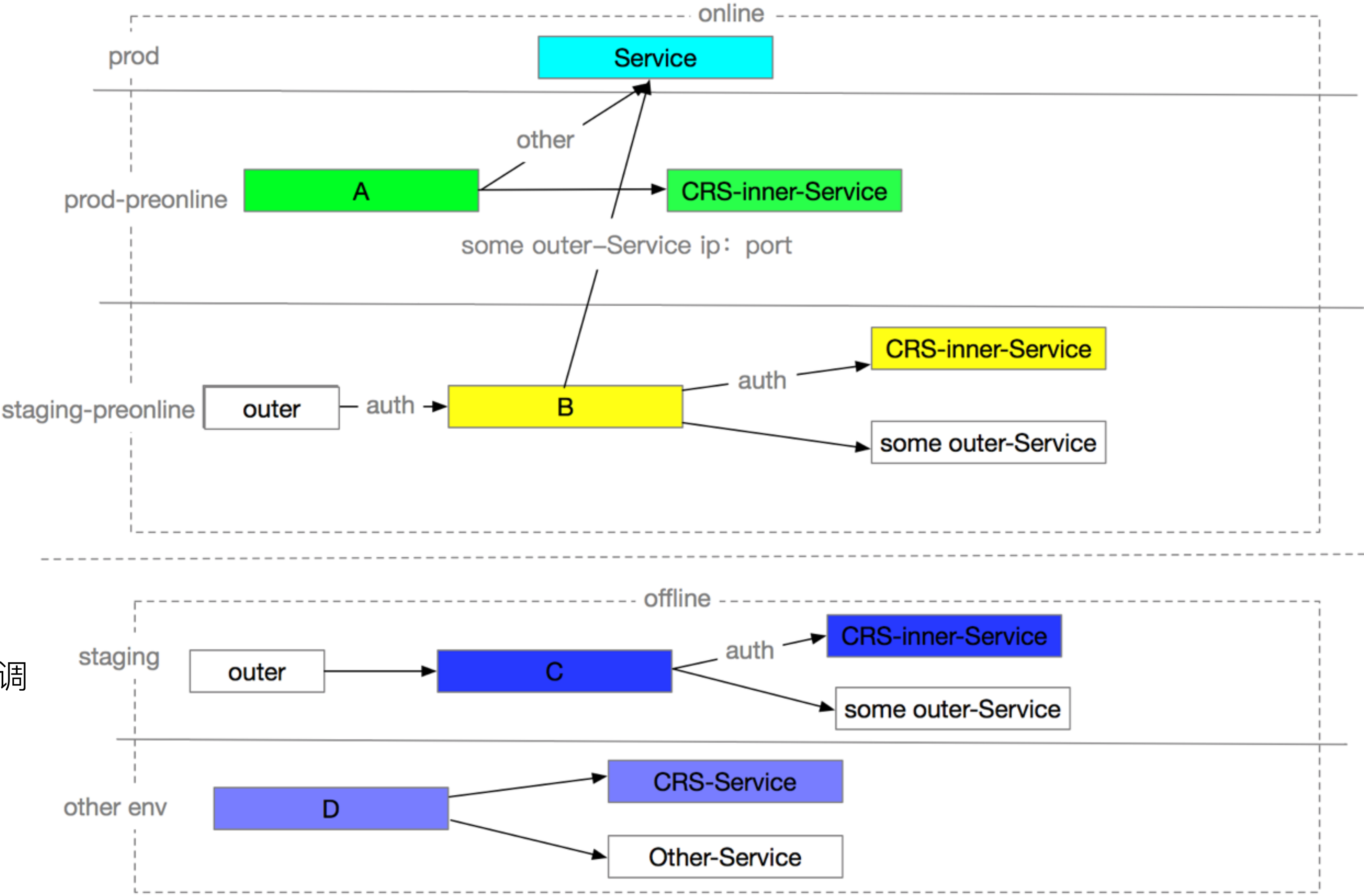
线上环境

线上压测

线上联调试

线下库直连联调

线下测试



可用性管理

复盘



项目复盘

- 回顾目标
- 评估结果
- 分析差异
- 总结经验
- 后续TODO



故障复盘

- 故障描述
- 损失评估
- 处理过程
- 5W根因定位
- 解决方案
- 后续TODO

可用性管理

复盘 - 发现新的问题

