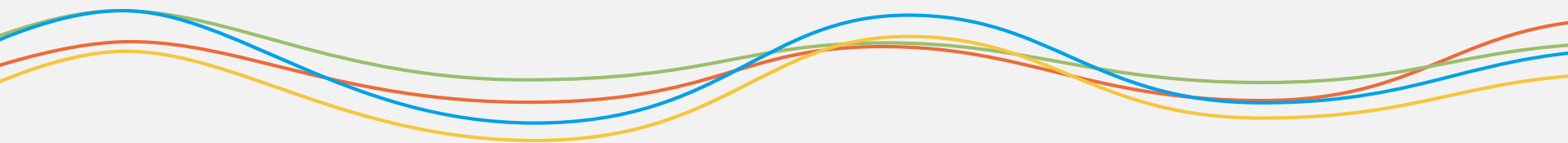




你我贷运维之路

给梦想可能



01

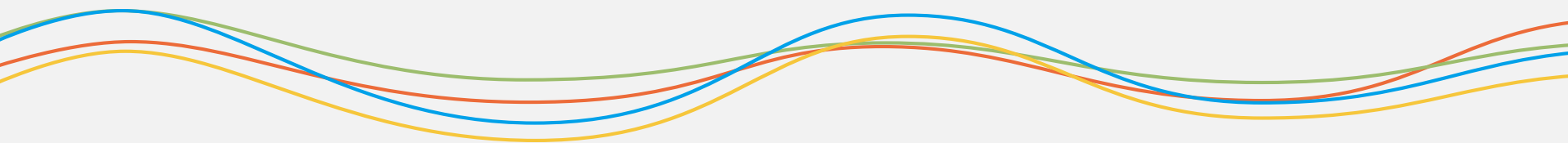
面临的问题

02

运维演变之路

03

决胜未来



01

面临的问题



发现问题 01

监控指标多并且复杂
误报和漏报严重
告警无法合并

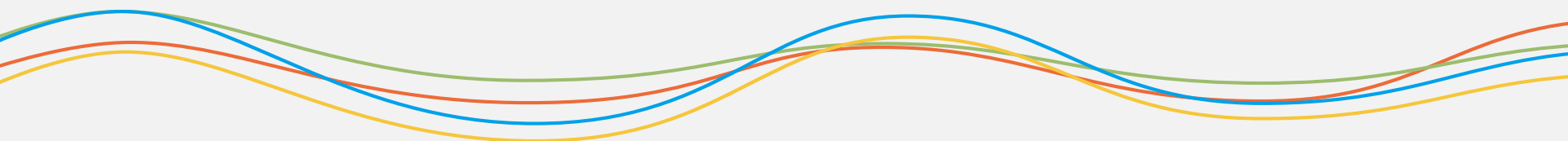


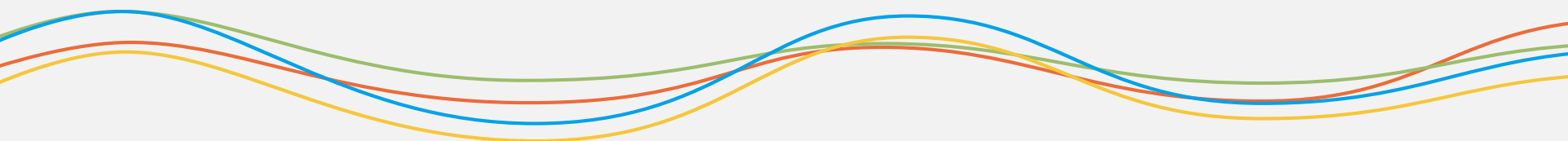
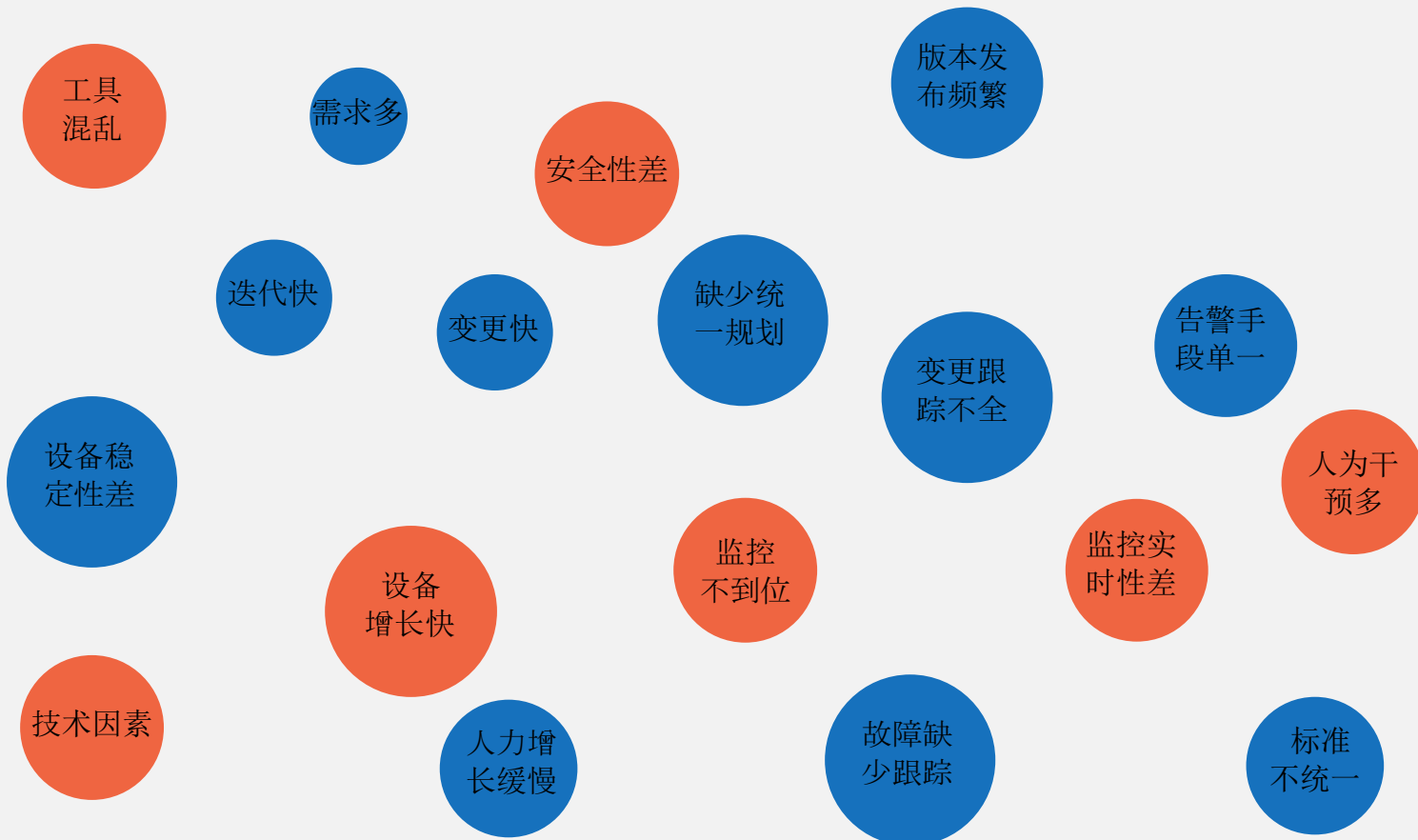
定位问题 02

故障定位难
缺少拓扑与调用关系
缺乏事件关联关系
主要靠经验判断

解决问题 03

无法故障自愈或隔离
主要依靠人工解决
处理问题效率低下







新技术促进运维能力趋向开发/运维一体化

运维需要更加贴进研发、业务



云计算带来变革

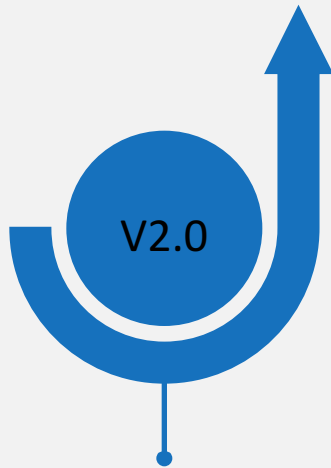
传统运维服务体系需要快速适应变革

02

运维演变之路



手工运维



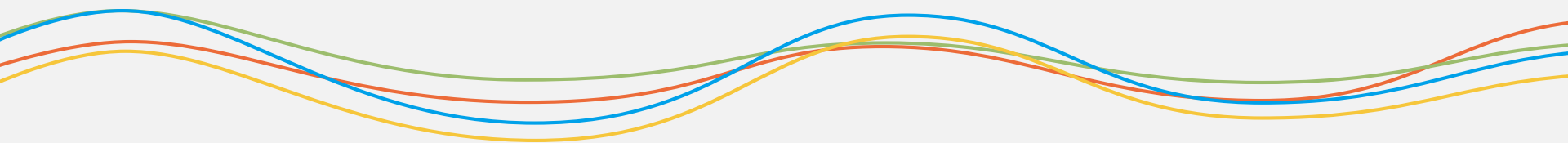
脚本运维



自动化运维



智能化运维





服务器

基础架构的每个组件都很复杂

效率低下的孤岛

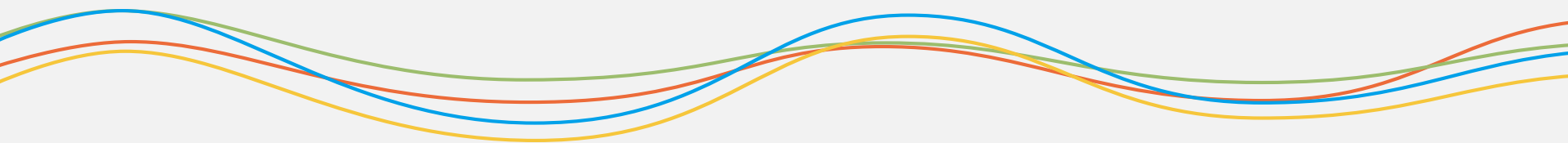
存储网络

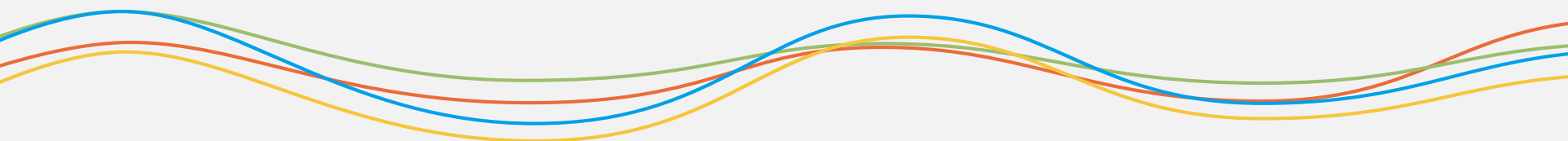
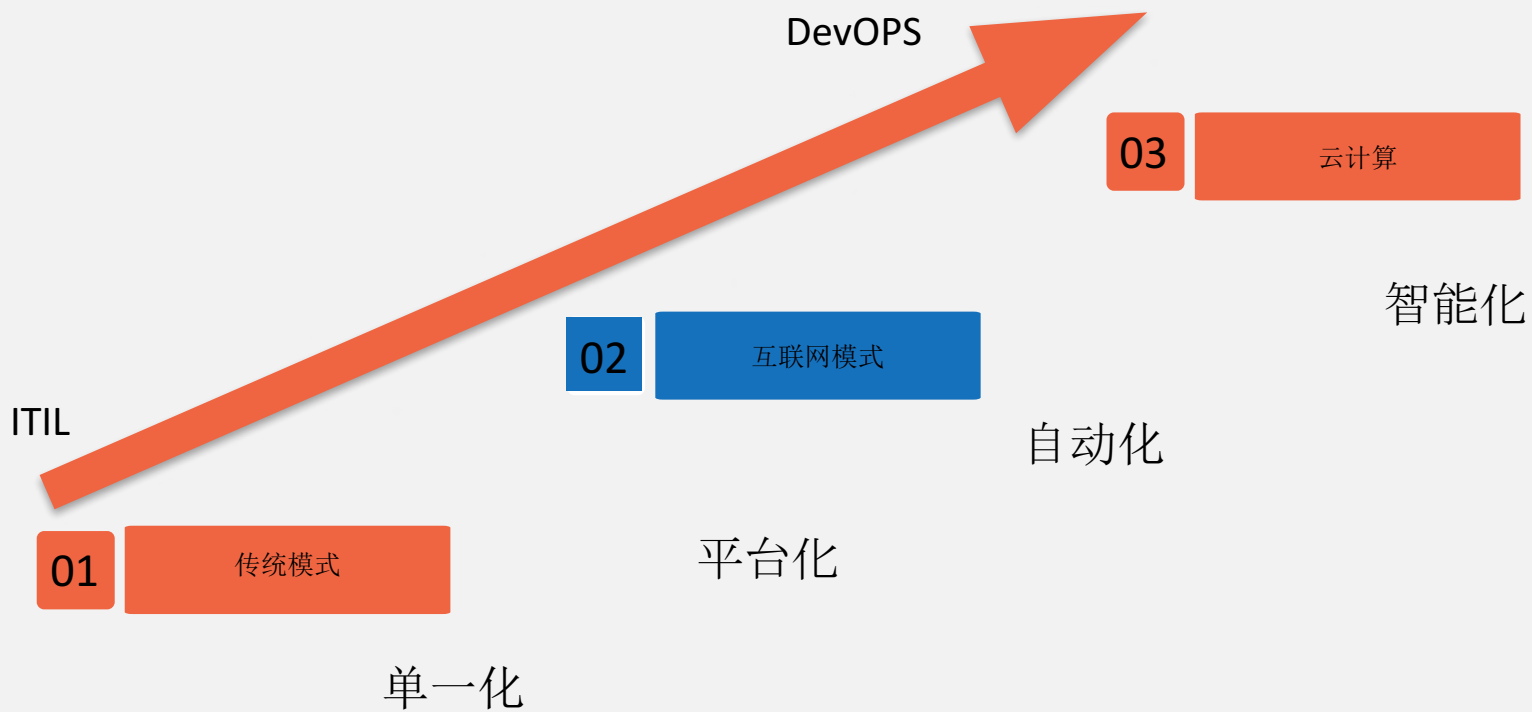
痛苦的升级

存储设备

复杂的管理

扩展性差







技术驱动 + 数据驱动



手工运维

业务运行环境复杂，故障定位难，效率低，重复性、手工性工作量大，费时费力，容易出错

脚本运维

缓解了手工运维的部分压力，架构越来越复杂，故障定位慢、缺少流程化、规范化、电子化，工作效率低。重复性、手工性工作量大，并且容易出错，缺少自动化工具和手段

自动化运维

统一标准化、规范化、流程化、电子化、保证业务7×24小时高效稳定运行.缺少事前预警、事中恢复、事后存档 .CMDB与应用配置管理为基础，实现可持续集成发布

智能化运维

智能告警、故障自愈、运营辅助等。数据采集量较大，建模过程较久。需要数据支持、规则建立、机器学习以及各类决策算法应用的引入，对技术挑战性非常大。



01

从0到1
从1到N
优先核心功能
逐步完善
小步快跑
收益最大化

02

标准化
规范化
流程化

03

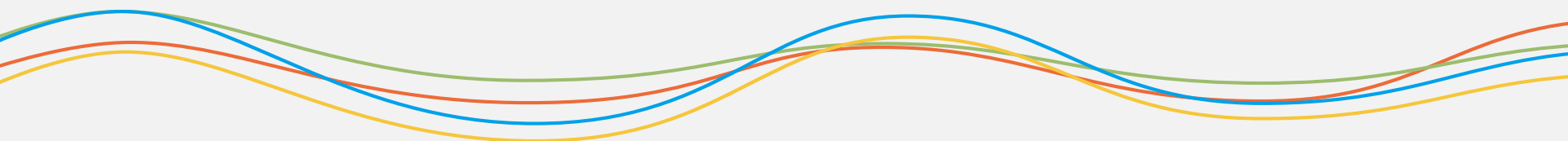
以大化小
优先最核心
最迫切功能
功能插件化

04

完善监控
从底往上
由精至细
分步落地

05

容忍缺陷
业务导向
快速反馈
不断改进





01

从0到1

摸石头过河
学习、交流、借鉴
从一头雾水到思路明确
难在规划

02

整合阶段

需求拆分
以大化小
分析、总结运维场景
逐步丰富功能
功能开发插件化
难点在于对运维场景和业务的理解

03

稳定期

智能告警
故障自愈
决策辅助
转向智能化运维
难点在于数据支持、规则的建立



可视化

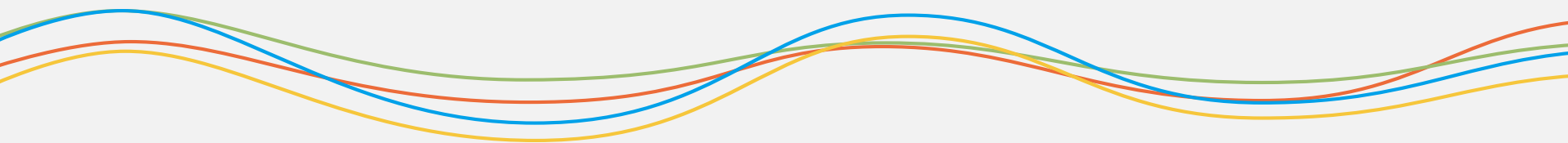
资源视图、网络视图、系统管理视图、监控视图、报表视图、告警视图、安全事件等多维报表

全自动化管理

自动发打包、自动安装部署、自动配置管理更新、自动扩展、自动备份与恢复、自动数据采集、自动故障隔离等

统一资源管理

资源统一管理、运维操作在平台界面完成、运维流程实现自助



终端

网站

移动APP

ERP

外部系统

CDN Server Radware

公共服务

权限管控

字典管理

短信服务

邮件服务

微服务接口

网站运营

会员中心

合作中心

礼券中心

开放平台

内容管理

产品中心

微信平台

活动中心

离线计算

交易

账户中心

撮合中心

债转中心

清算中心

还款服务

财务

银行卡

收款

付款

呼叫中心

客服服务

营销服务

投资服务

贷款服务

数据服务

大数据风控

数据仓库

驾驶舱

用户画像

算法平台

服务监管与治理

dubbo

红警

TCC

服务授权

MQ

日志监控

测试自动化平台

运维自动化平台

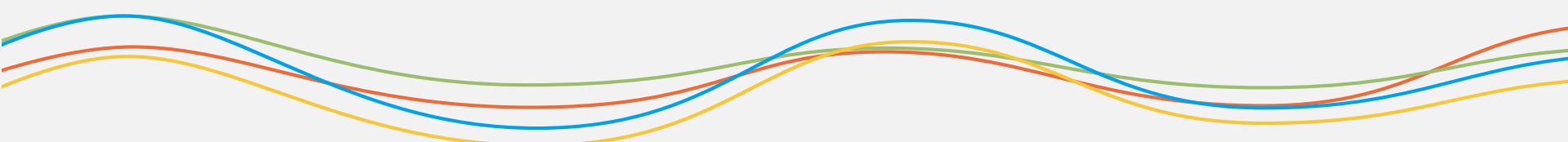
数据系统

数据库

redis

cache

文件





安全监控

行为分析、羊毛党监控、CC攻击、DDOS、SQL注入等

性能监控

响应时间、网络利用率、吞吐量、慢查询、调用栈、cache等

业务监控

交易数据、资金监控、信审业务监控、负载使用、流量分布等

应用监控

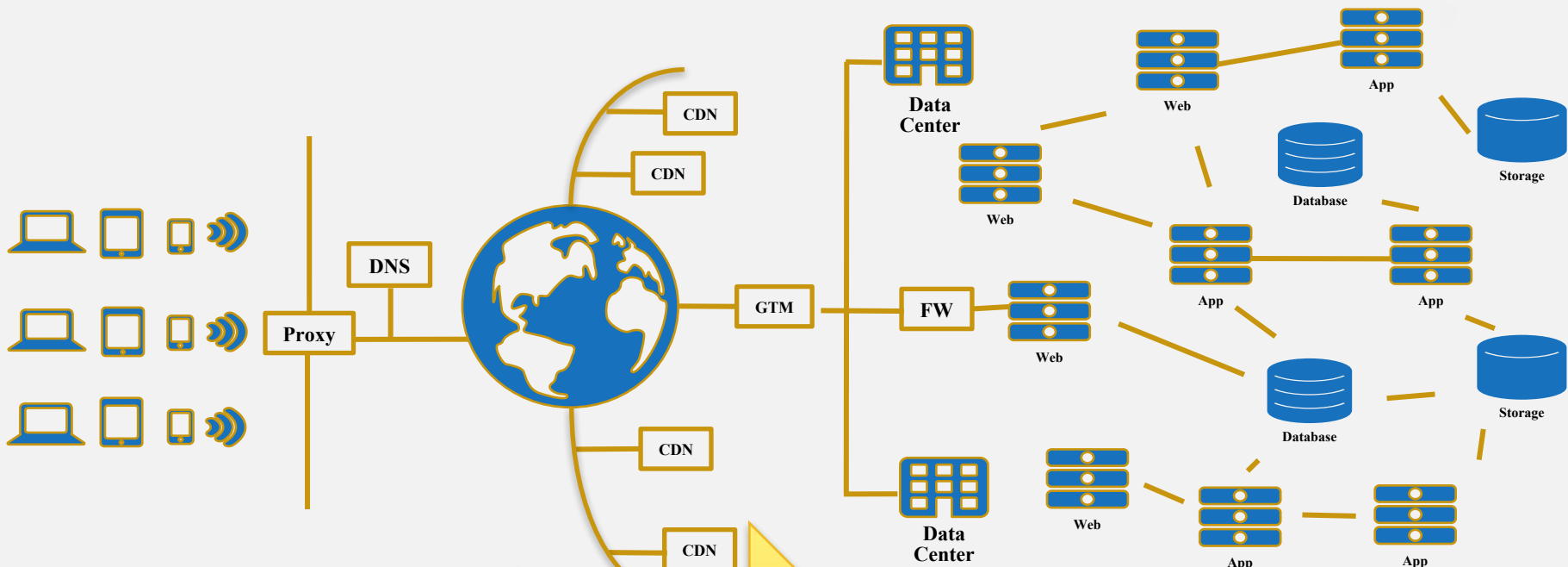
服务探活、作业调度、批处理、日志等

系统监控

CPU、内存、磁盘IO、网络IO、连接状态、进程端口、存储容量等

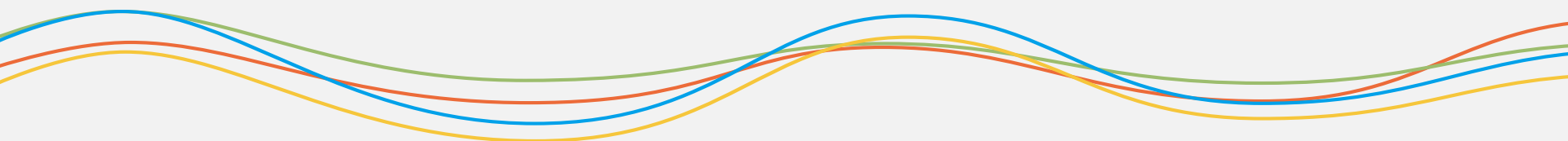
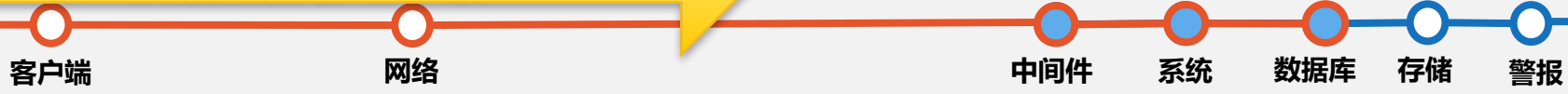
硬件监控

服务器温度、电压、风扇、电源模块、网络设备软硬件状态等



听云 全栈溯源 — 用户体验为核心的业务
质量监控管理体系

Zabbix, Cat, NAGIOS, Grafana (生产)、ELK、响应系统



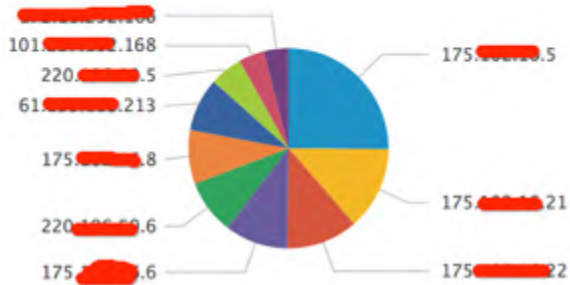


WEB-log-analysis

WEB-log-analysis

编辑 ▾ 更多信息 ▾ 下载 打印

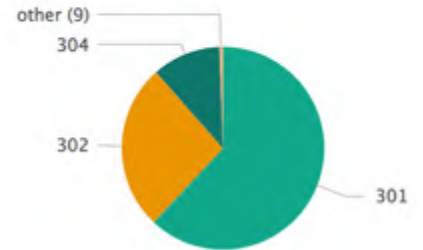
访问网站排名前10的IP地址



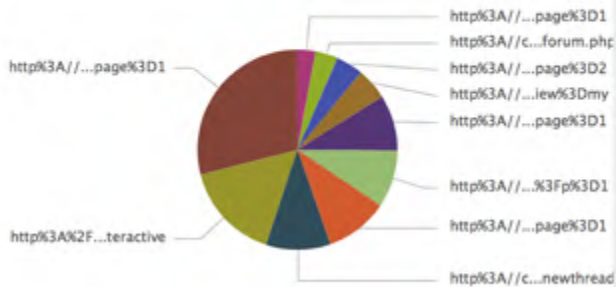
有多少404事件



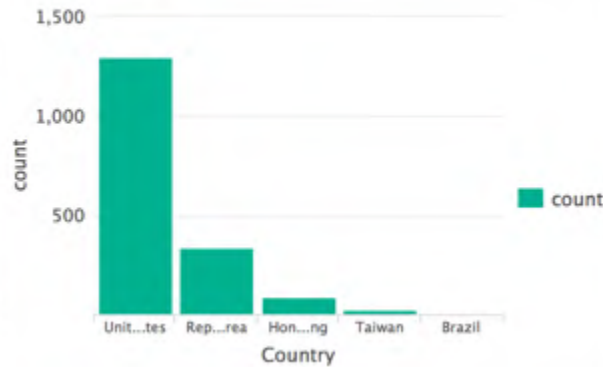
HTTP错误状态码分析



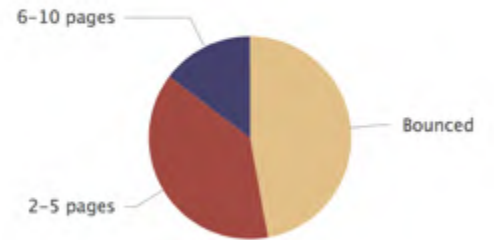
统计浏览来源最多的网站



来自国外的事件统计

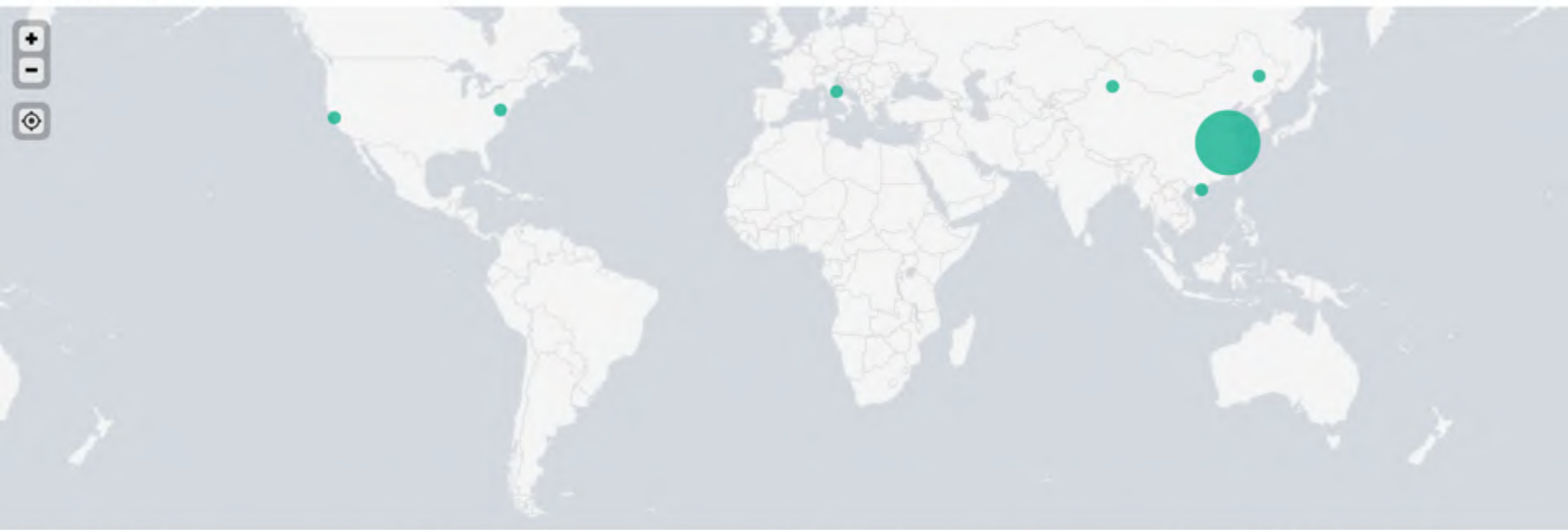


网站跳出率

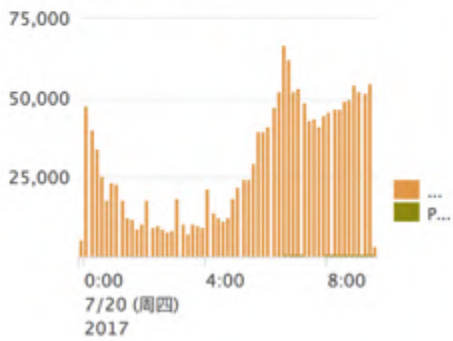




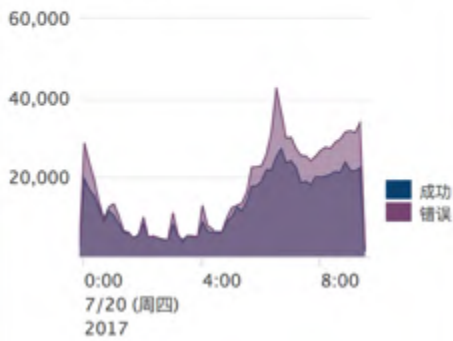
在世界地图上显示IP分布



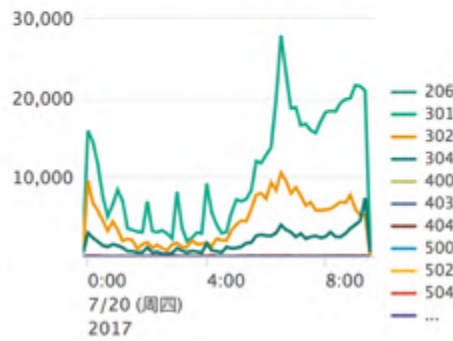
GET/POST请求分析



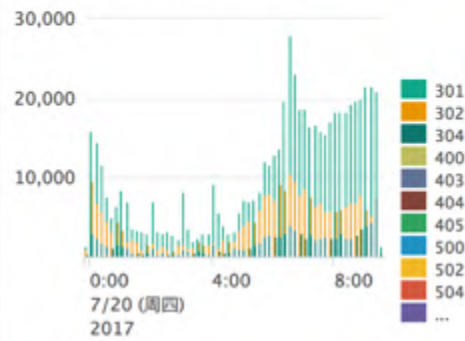
成功事件vs错误事件



统计4XX和5XX事件的时间趋势图



HTTP请求未成功事件时序图





Redis

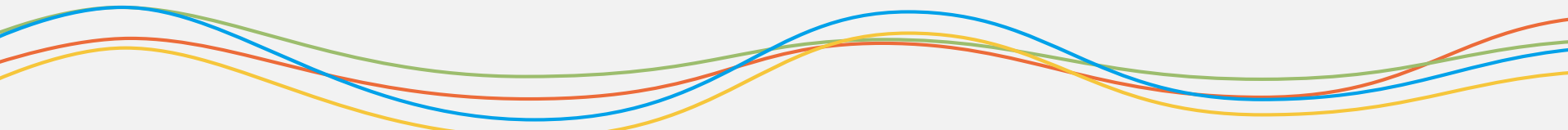
初期使用，可以满足要求，但是随着管理服务器增加，处理消息量剧增，出现过消息丢失情况，造成运维故障。(返回结果的确认很重要)

ActiveMQ

能满足我们的运维使用需求，但是管理页面不友好，不支持完善的事务型消息解决方案，HA方案存在一些问题。

PostMQ

在RocketMQ基础上自研了PostMQ：
提高了消费者获取消息的实时性；实现了单消息、批量消息的处理模式；HA方案中实现了raft一致性算法；功能更加完善的管理后台。





规范再统一

再次制定新的标准和规范，推行实施，自动化运维中后期。

规范不适用

制定的标准和规范发现不能满足业务的发展或者是运维平台的建设。
(运维平台简化上线流程时，需要直接从对接北斗系统的时候，发现之前制定规范根本行不通))

统一规范

制定运维平台需要的统一规范，统一标准，推行实施，自动化运维初期。
(制定统一的war版本规范，用研发部名+项目名+包名)

杂乱无序的标准

规范杂乱无章，各个研发团队各自为政，生产上线手动运维阶段。
(例如：生产环境的war包命名规范)



出现这种情况后，怎么去协调各个部门，解决问题，这也是一门大的学问。

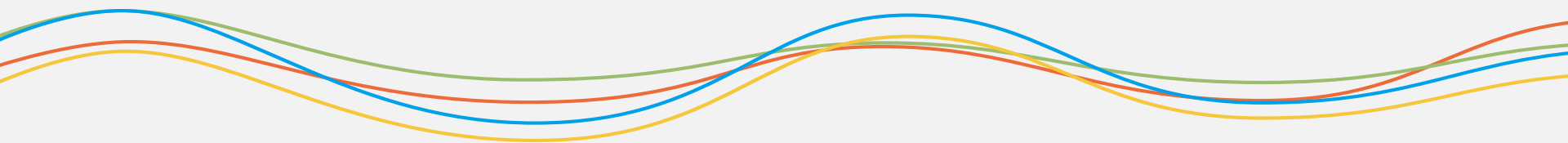




全生命周期
自动化管理

故障预防、
发现与自愈

智能化容量
运营





01

全链路监控、海量日志

用户端、链路、服务端等多维度进行数据采集、分析，并进行自动化检测

02

大数据分析可视化

通过大数据平台，建立分析模型，发现问题以及相关联信息，进行可视化展示

03

自动学习、分析

通过机器学习算法，自动学习并发现故障趋势。自动发现环境、服务或应用依赖关系的变更

04

预测故障、自愈

在故障发生前发出预警或根据预先设立的规则进行故障自动处理



01

机器数据

服务器、操作系统及相关
日志输出到ELK

02

网络通信数据

网络抓包，流量分析，协议分析

03

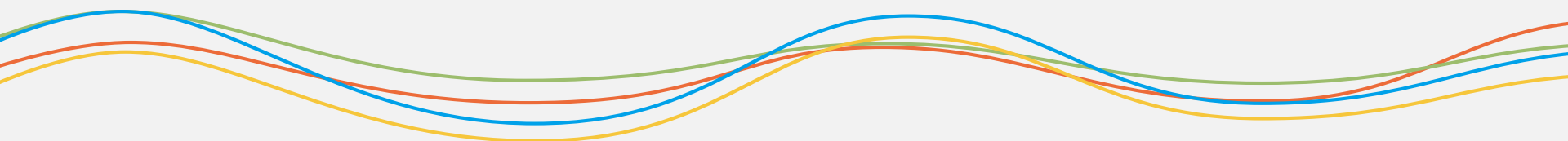
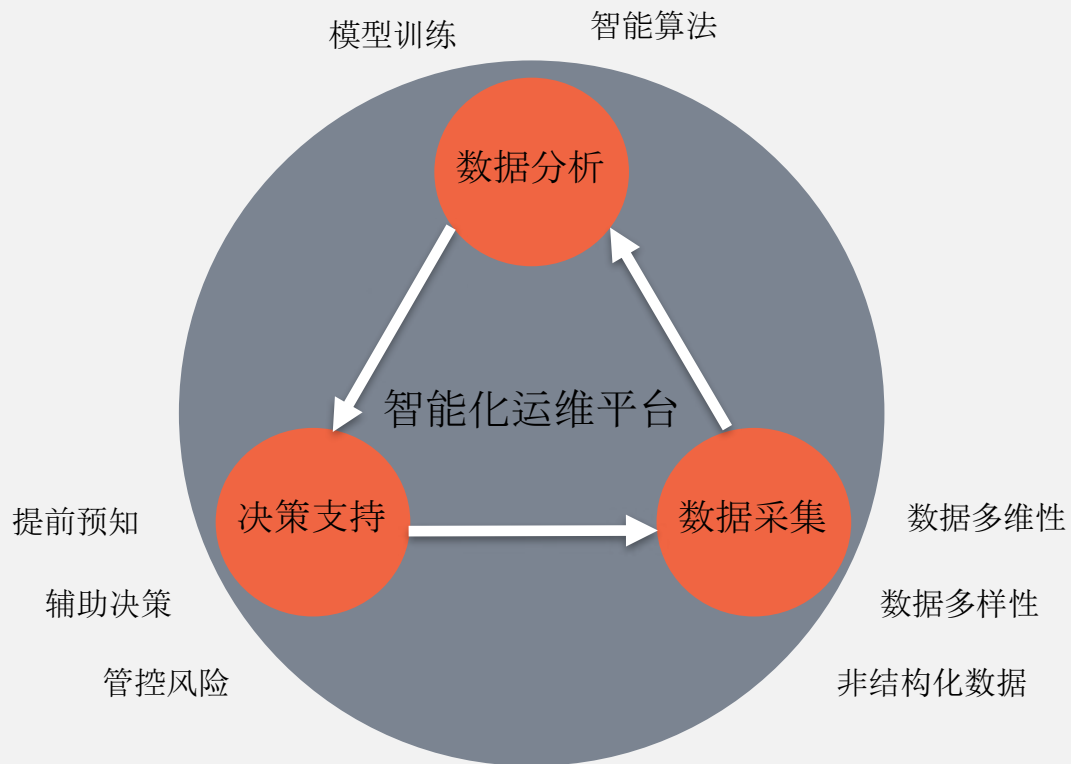
埋点

在各接口或关键核心业务点
插入统计或函数调用等代码
数据汇总到ELK或CAT里面

04

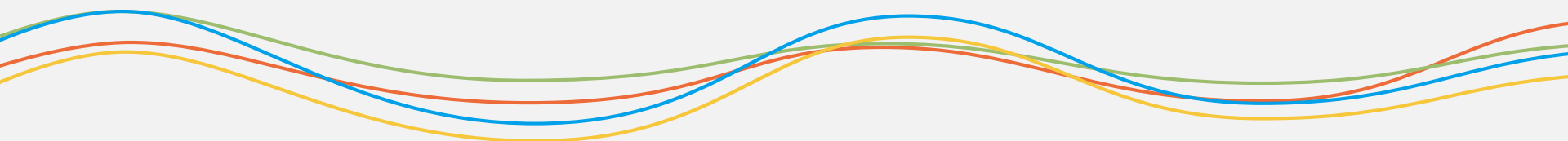
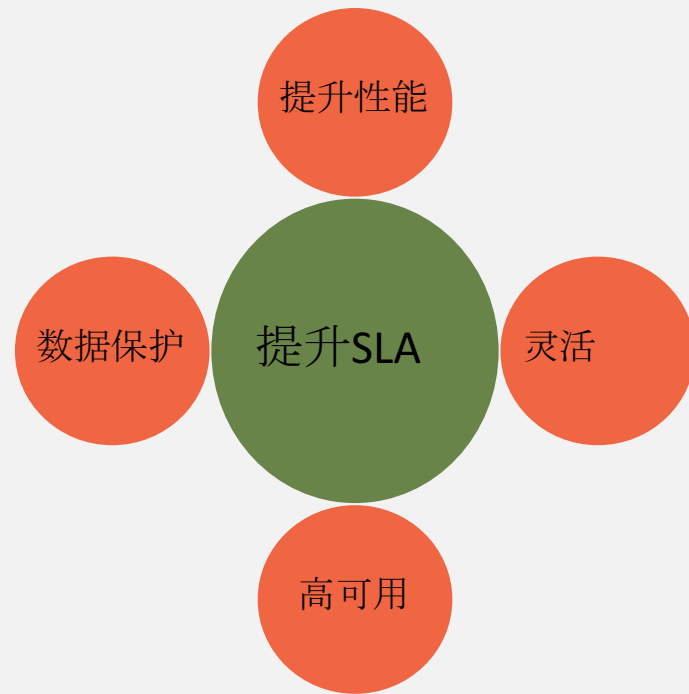
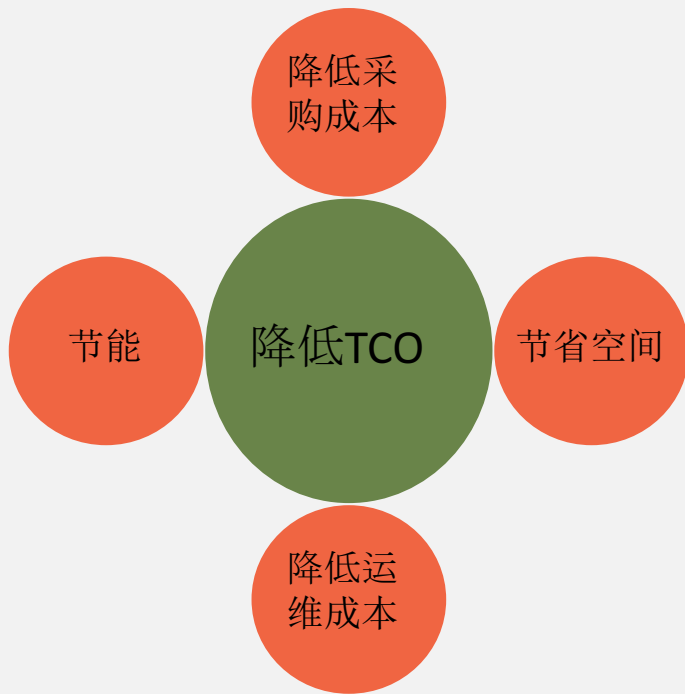
探针

通过听云在各地模拟ping，HTTP GET
等操作或SDK对系统检测



03

决胜未来





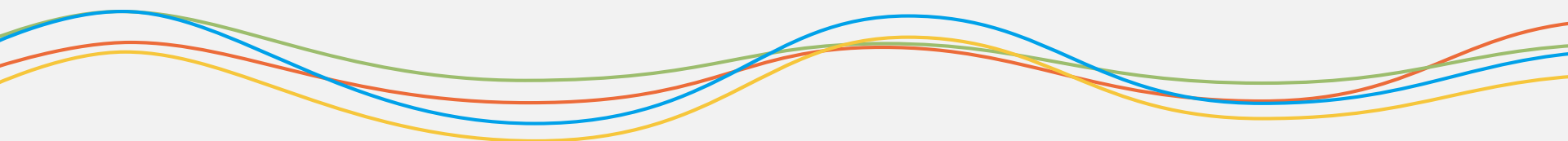
状态监控及信息收集
监控应用、硬件、网络等
整体运行状态，并收集各
指标

数据分析
分析历史数据
辅助问题解剖

预防
问题定位
流量预测
决策支持
故障自愈

报表统计
统计分析历史数据
辅助运维系统管理

预测判断
分析历史数据
构建分析模型
预测事件发生
提供决策支持



THANK YOU

