微服务和Service Mesh

在多个行业落地实践

网易云
163yun.com

阶段一：单体架构群，多个开发组，统一运维组

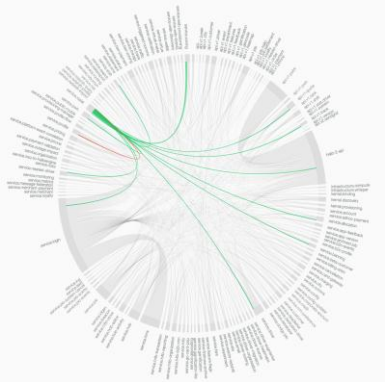
开发组：
- 开发应用

运维组：
- 管理资源
- 部署上线

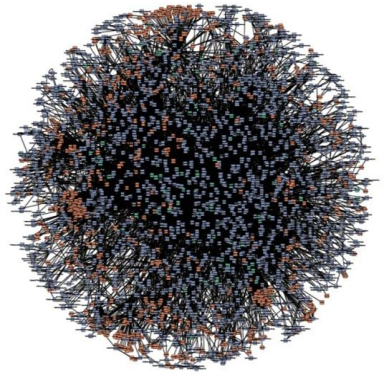
阶段二：组织服务化，架构SOA化，基础设施云化

阶段三：组织DevOps化，架构微服务化，基础设施容器化

微服务成为互联网化架构的典型特征

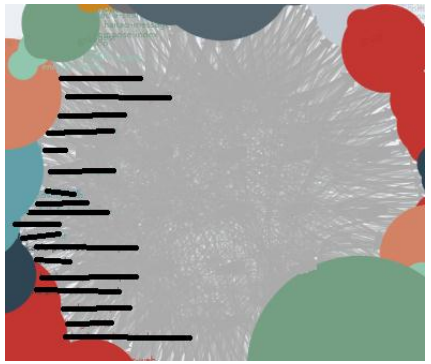


Hailo



Amazon

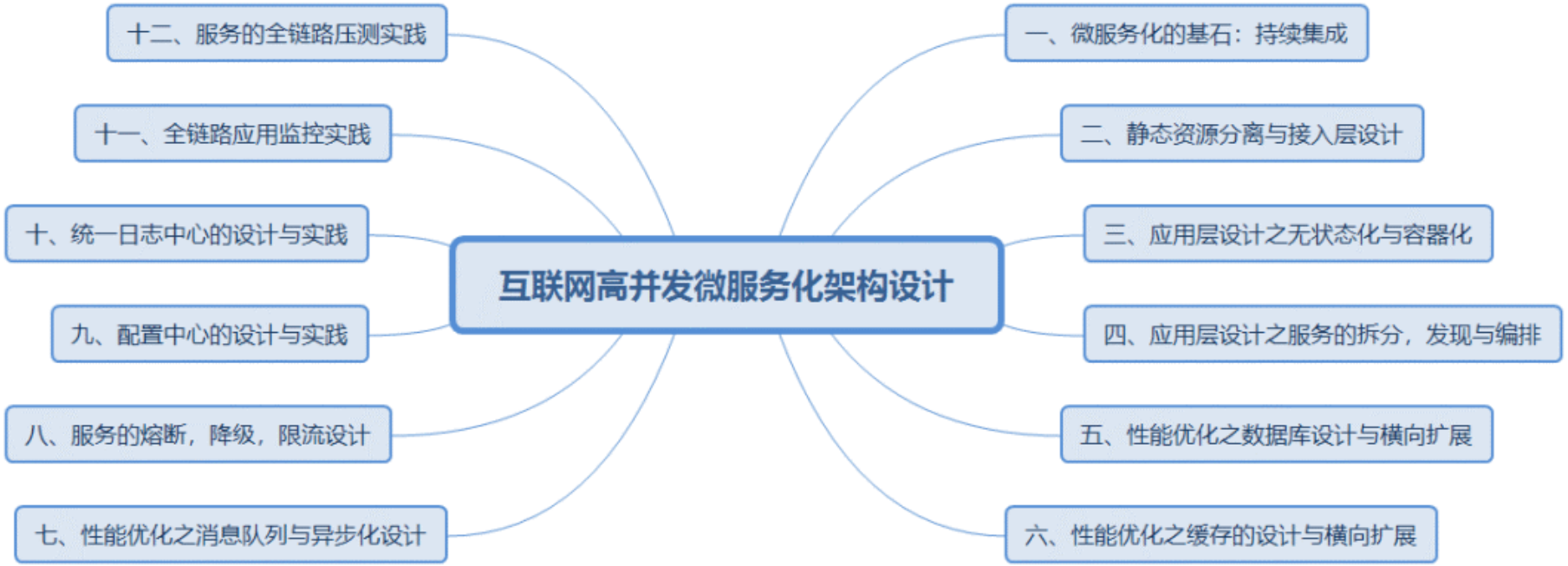
Netflix



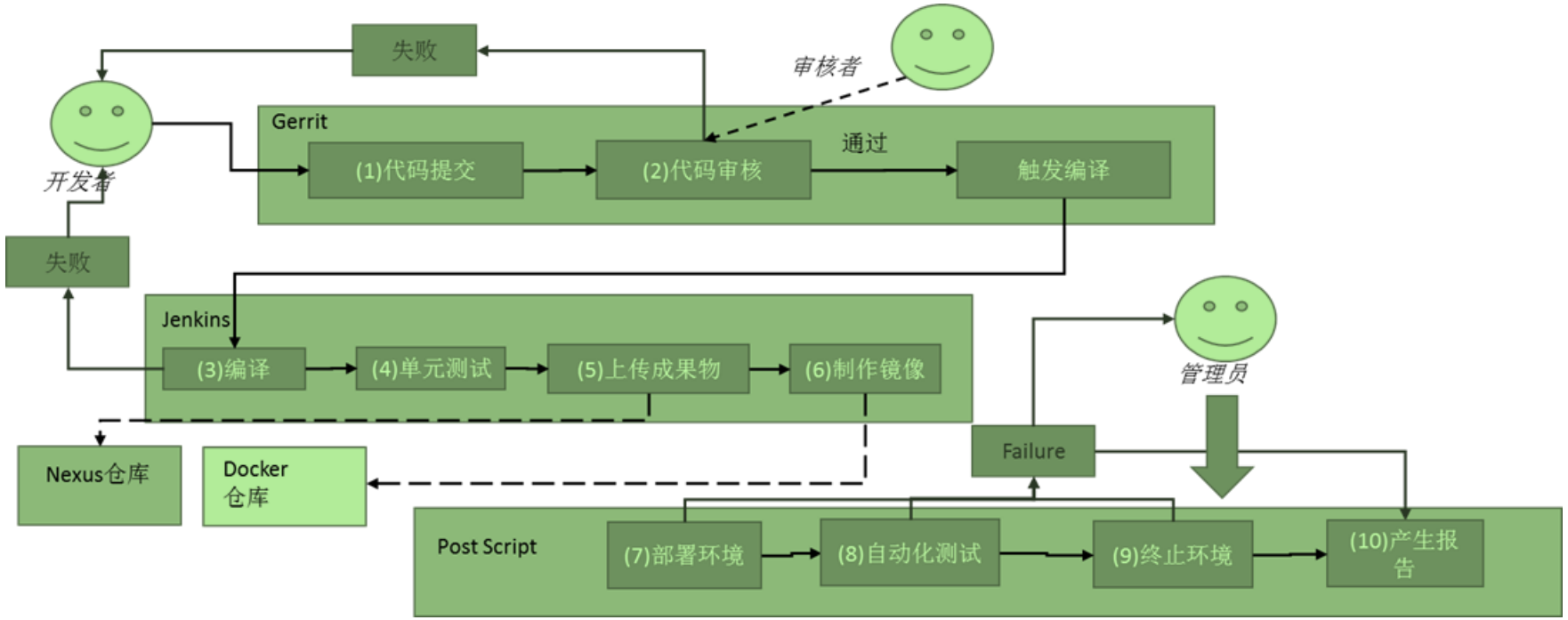
Kaola

来源：https://www.linkedin.com/pulse/astonishingly-underappreciated-azure-service-fabric-ben-spencer/

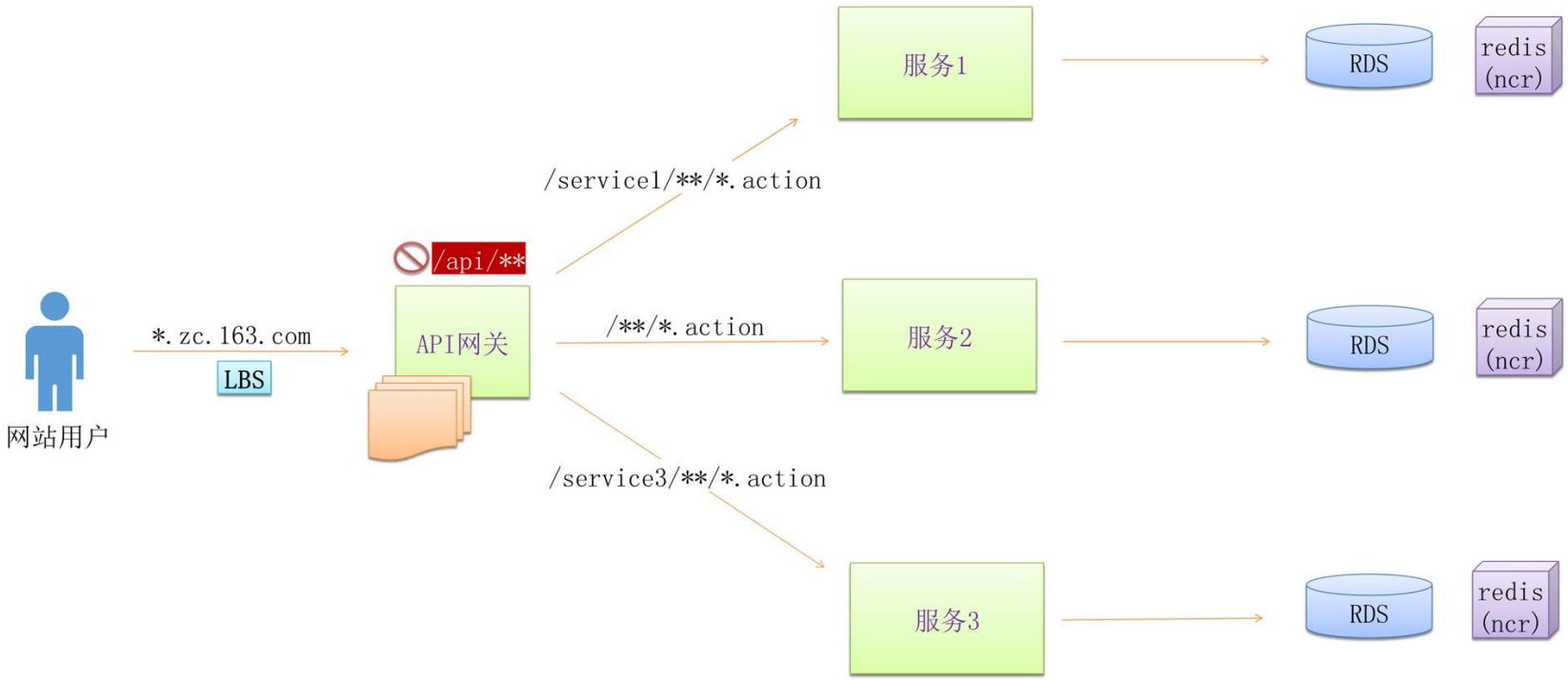
微服务化设计要点


互联网高并发微服务化架构设计
一、微服务化的基石：持续集成
二、静态资源分离与接入层设计
三、应用层设计之无状态化与容器化
四、应用层设计之服务的拆分，发现与编排
五、性能优化之数据库设计与横向扩展
六、性能优化之缓存的设计与横向扩展
七、性能优化之消息队列与异步化设计
八、服务的熔断，降级，限流设计
九、配置中心的设计与实践
十、统一日志中心的设计与实践
十一、全链路应用监控实践
十二、服务的全链路压测实践

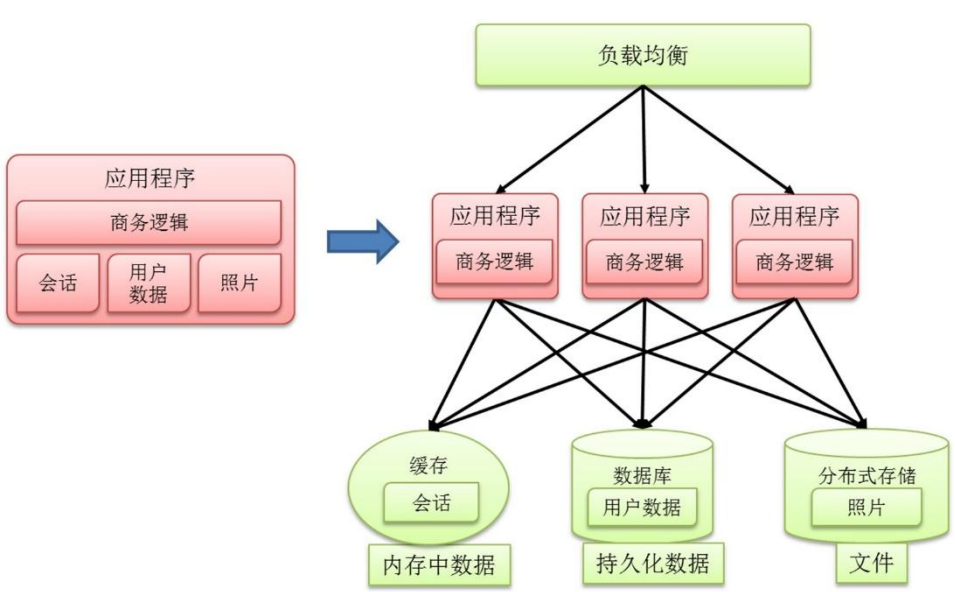
设计要点一：持续集成

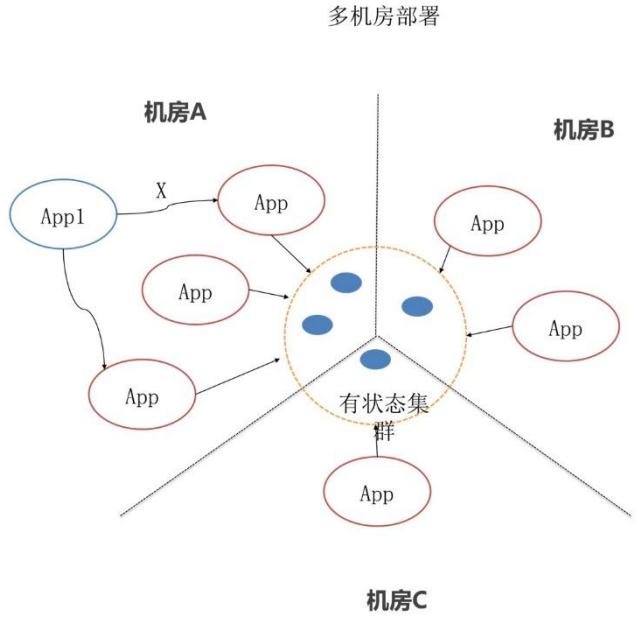

失败
审核者
Gerrit
(1)代码提交
(2)代码审核
通过
触发编译
开发者
失败
Jenkins
(3)编译
(4)单元测试
(5)上传成果物
(6)制作镜像
管理员
Nexus仓库
Docker
仓库
Failure
Post Script
(7)部署环境
(8)自动化测试
(9)终止环境
(10)产生报告

设计要点二：接入层设计

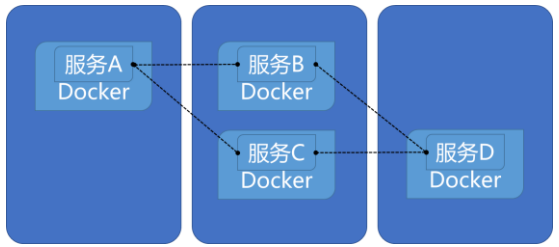

网站用户
*.zc.163.com
LBS
API网关
/api/**
/service1/**/*.action
/**/*.action
/service3/**/*.action
服务1
服务2
服务3
RDS
RDS
RDS
redis
(ncr)
redis
(ncr)
redis
(ncr)

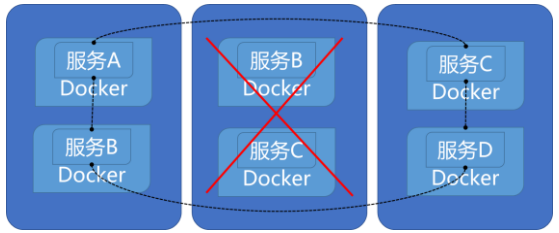
设计要点三：无状态化

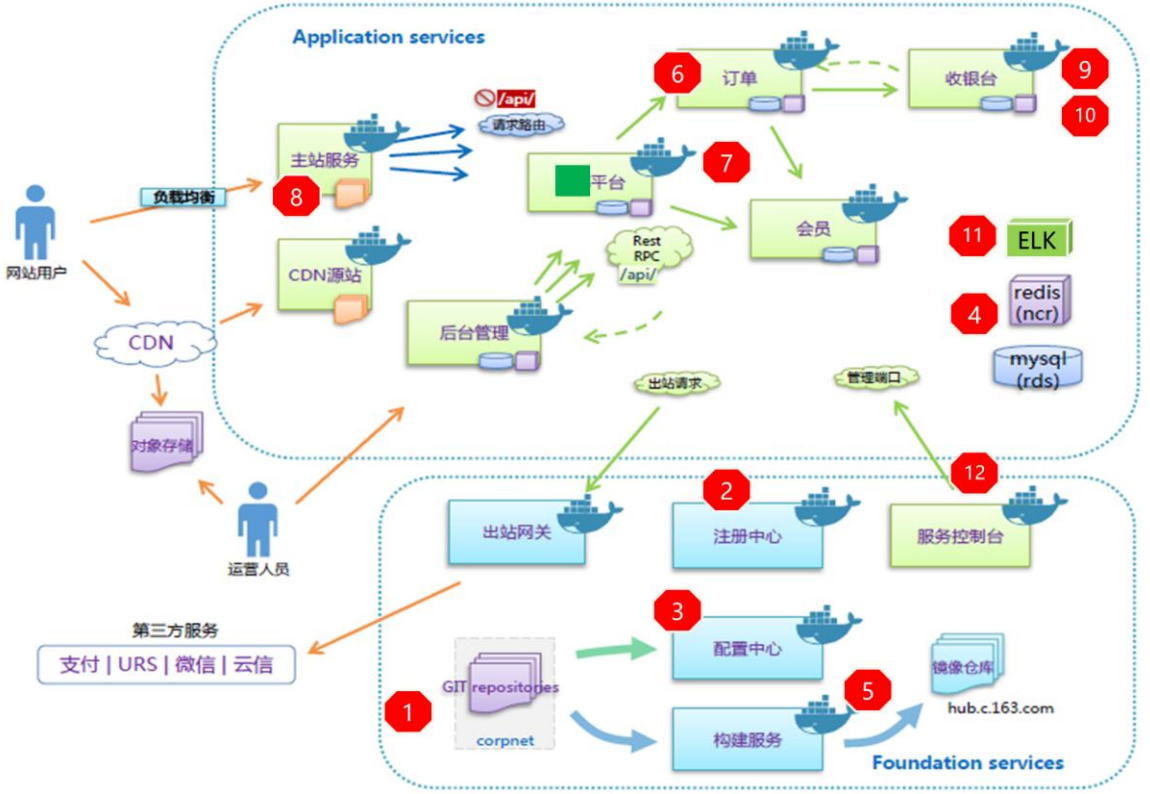

应用程序
商务逻辑
会话
用户数据
照片
负载均衡
应用程序
商务逻辑
应用程序
商务逻辑
应用程序
商务逻辑
缓存
会话
内存中数据
数据库
用户数据
持久化数据
分布式存储
照片
文件


多机房部署
机房A
机房B
App1
X
App
App
App
App
App
有状态集群
App
机房C

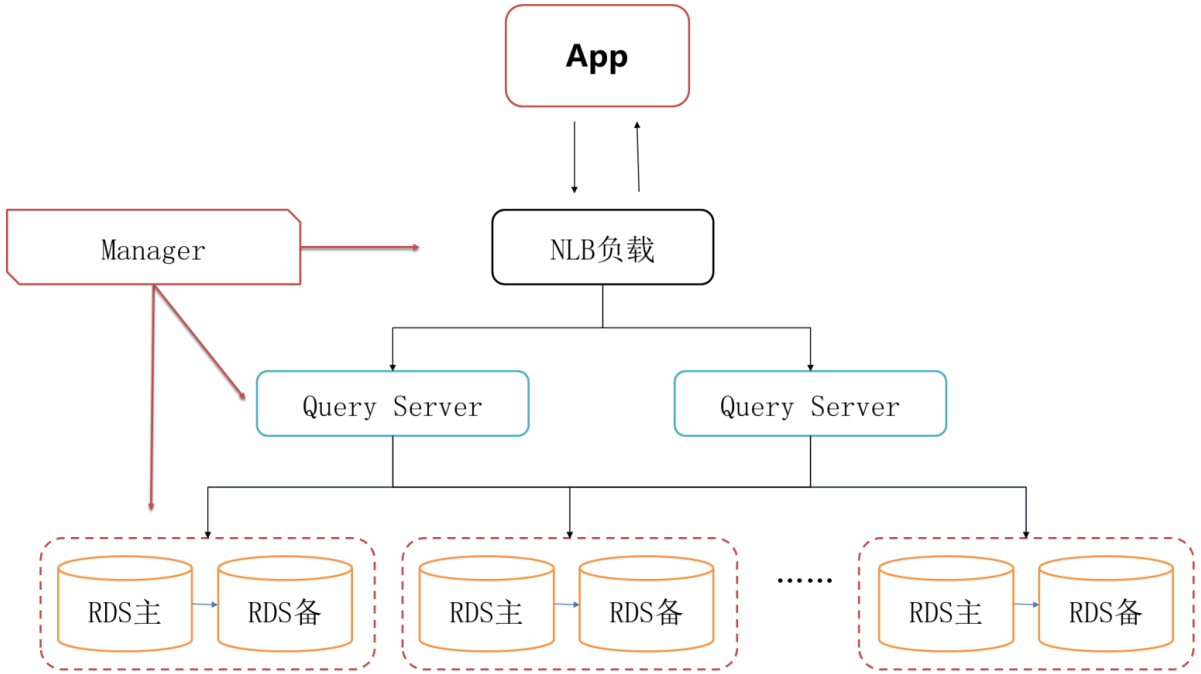
设计要点四：服务拆分与服务发现







设计要点五：数据库横向扩展



- 分布式部署
- 底层节点主备
- 负载均衡
- QS自动切换
- 双机房部署
- 读写分离

设计要点六：缓存的设计

APP缓存

CDN

接入层

静态资源

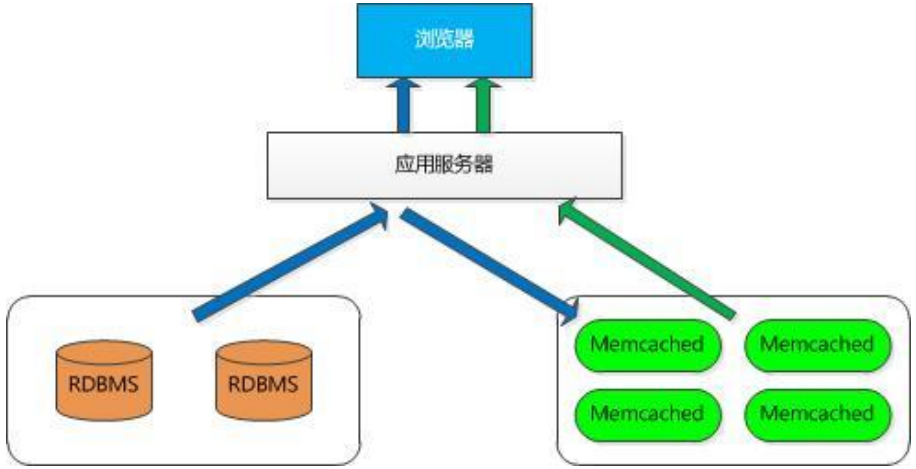
动态资源静态化

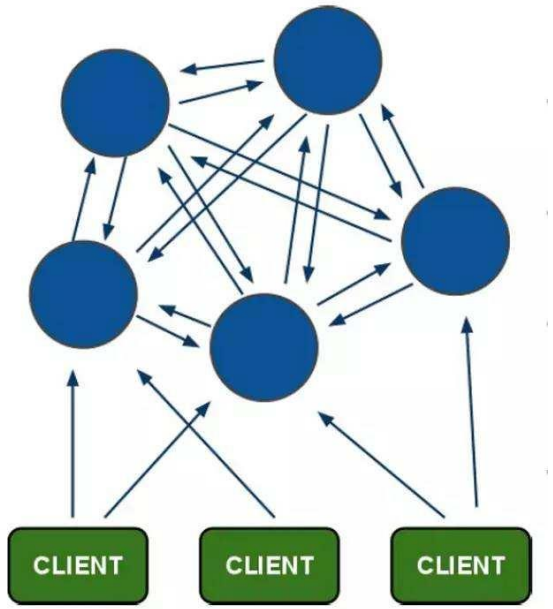
应用本地缓存

分布式缓存

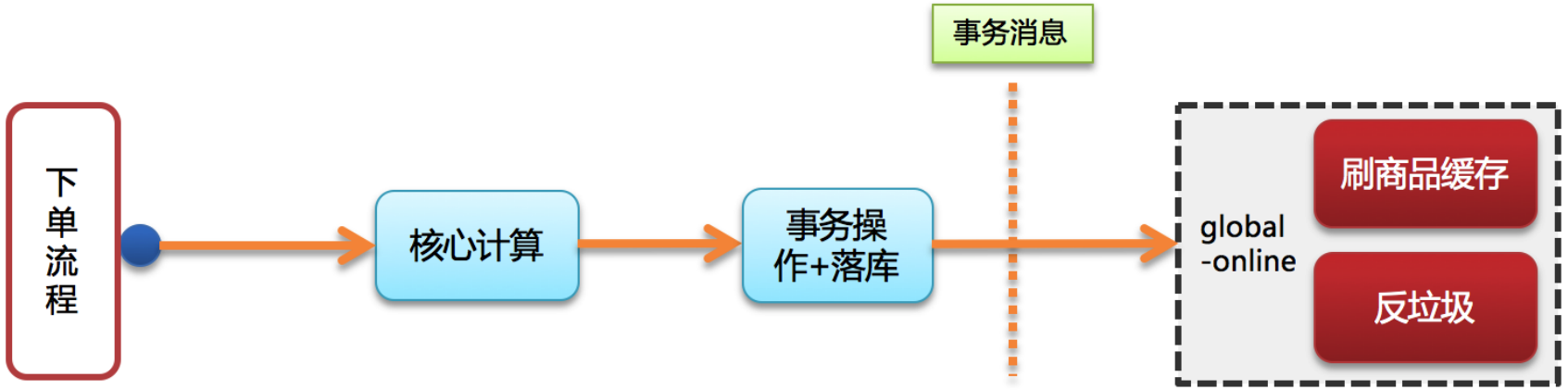
数据库为中心

缓存为中心

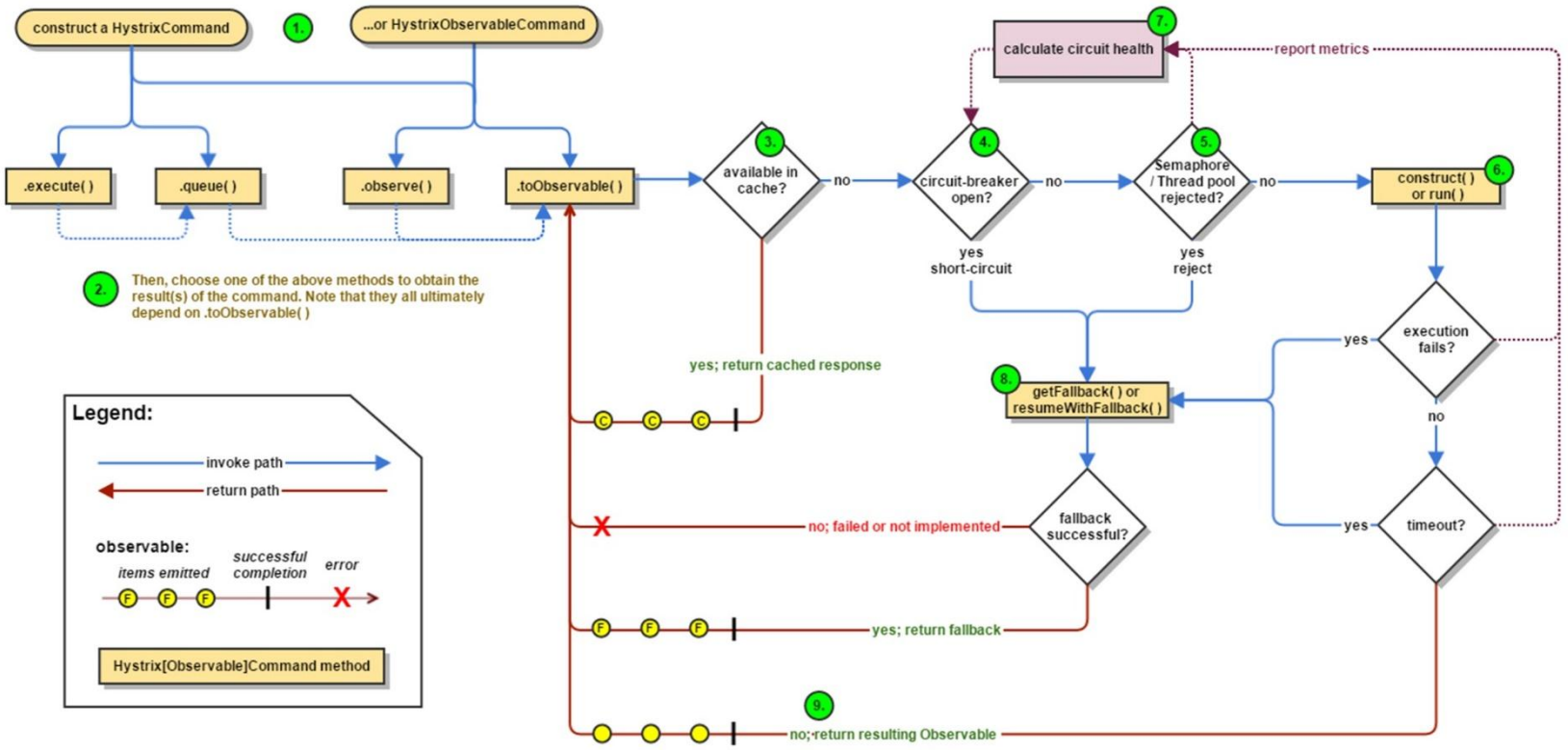




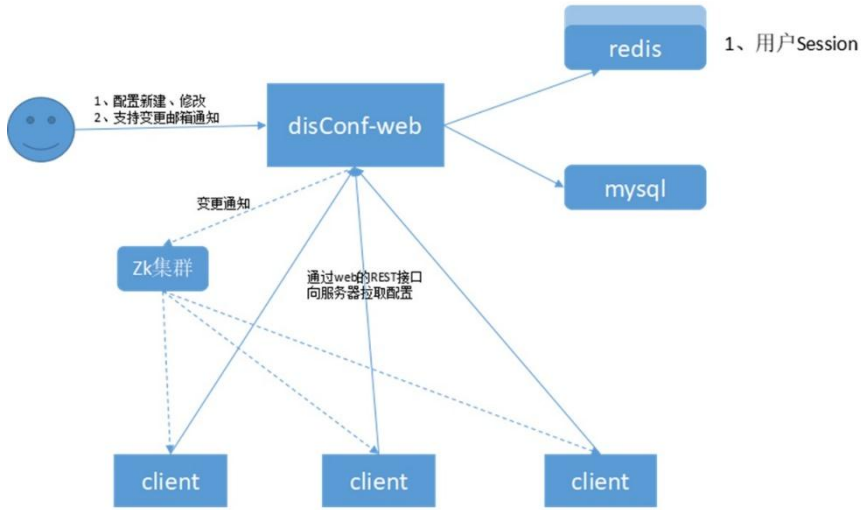
设计要点七：消息队列与异步化

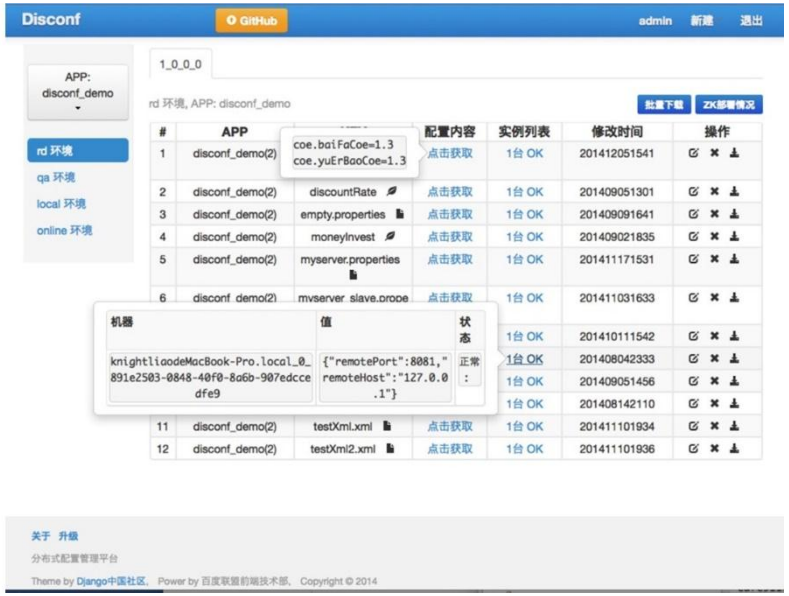


设计要点八：熔断，限流，降级

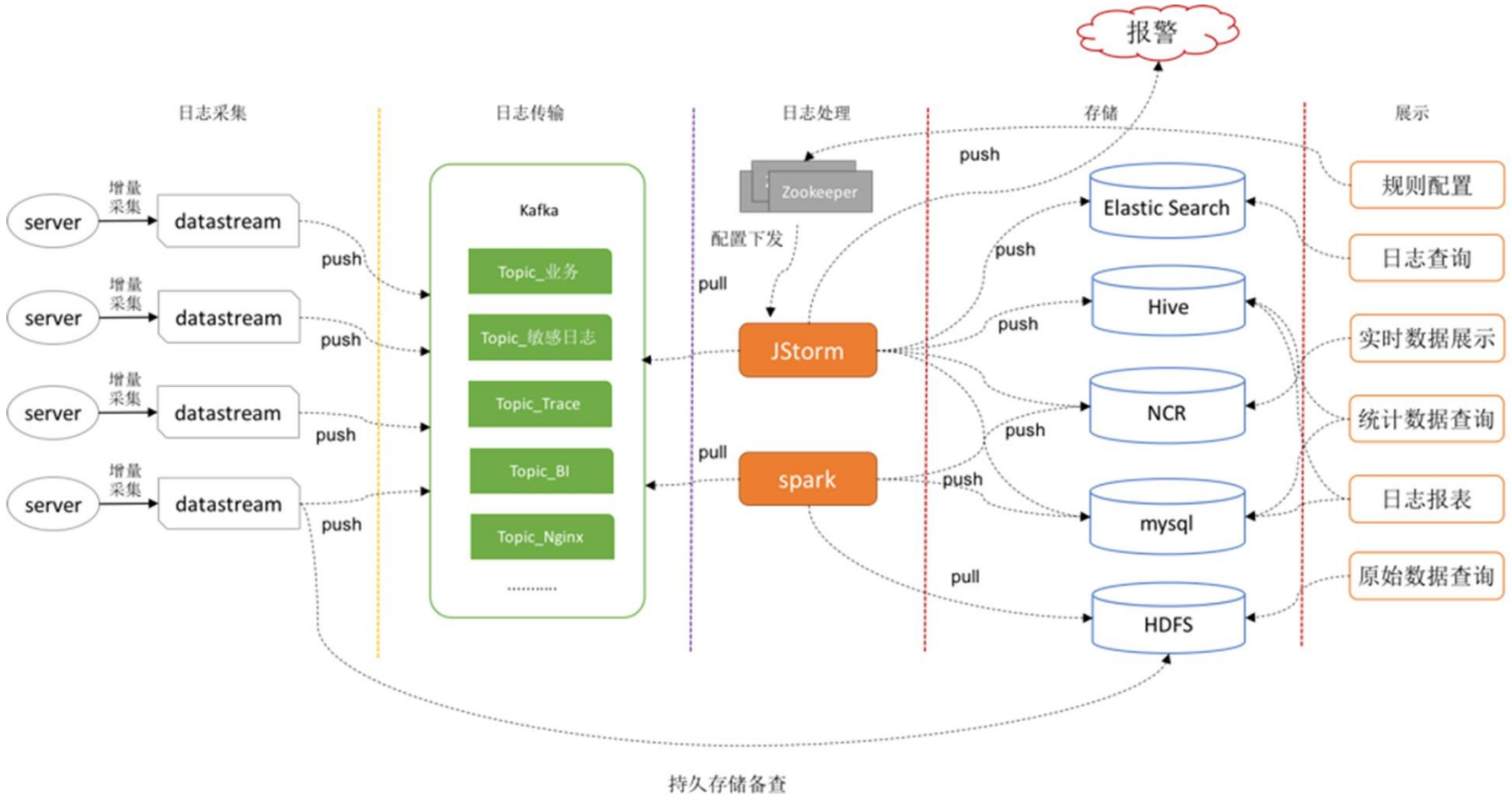


设计要点九：配置中心


redis
1、用户Session
1、配置新建、修改
2、支持变更邮箱通知
disConf-web
mysql
变更通知
Zk集群
通过web的REST接口
向服务器拉取配置
client
client
client


Disconf
GitHub
admin
新建
退出
APP:
disconf_demo
rd 环境
qa 环境
local 环境
online 环境
1_0_0_0
批量下载
ZK部署情况
coe.baiFaCoe=1.3
coe.yuErBaoCoe=1.3
机器
值
状态
knightliaodeMacBook-Pro.local_0_891e2503-0848-40f0-8a6b-907edccedfe9
{"remotePort":8081,"remoteHost":"127.0.0.1"}
正常
关于 升级
分布式配置管理平台

设计要点十：日志中心


报警
日志采集
日志传输
日志处理
存储
展示
server
增量采集
datastream
push
Kafka
Topic_业务
Topic_敏感日志
Topic_Trace
Topic_BI
Topic_Nginx
.....
Zookeeper
配置下发
pull
JStorm
spark
Elastic Search
Hive
NCR
mysql
HDFS
规则配置
日志查询
实时数据展示
统计数据查询
日志报表
原始数据查询
持久存储备查

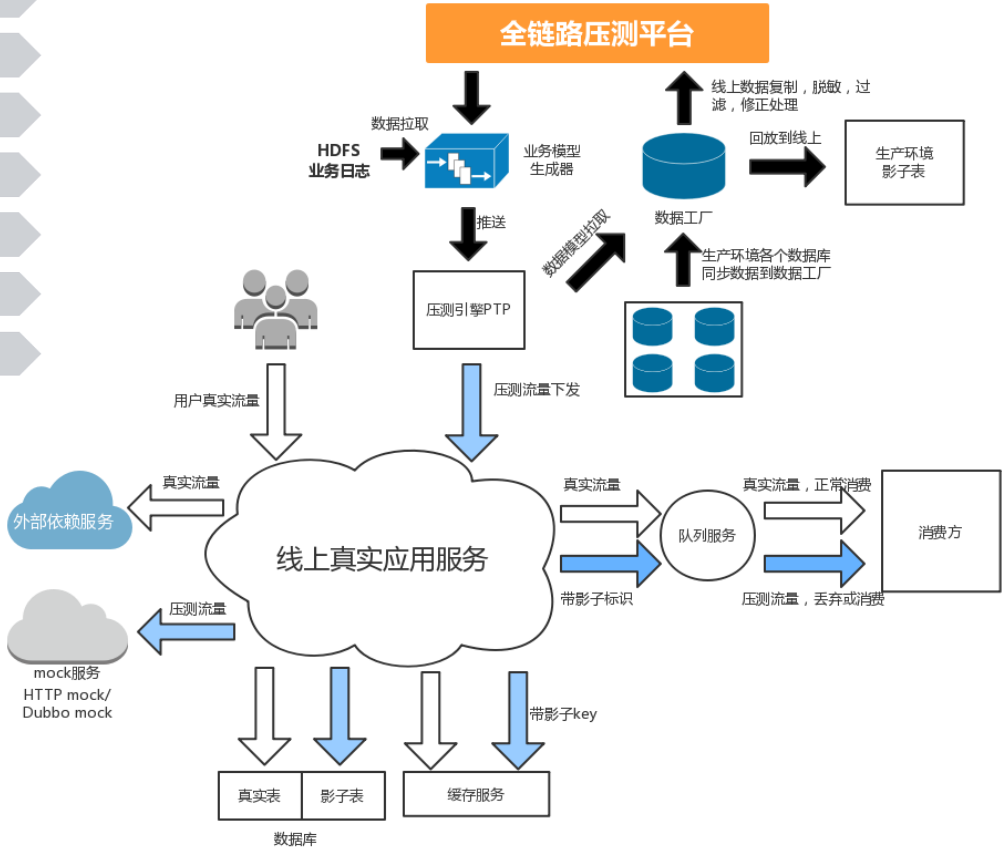
设计要点十一：全链路监控

网易云

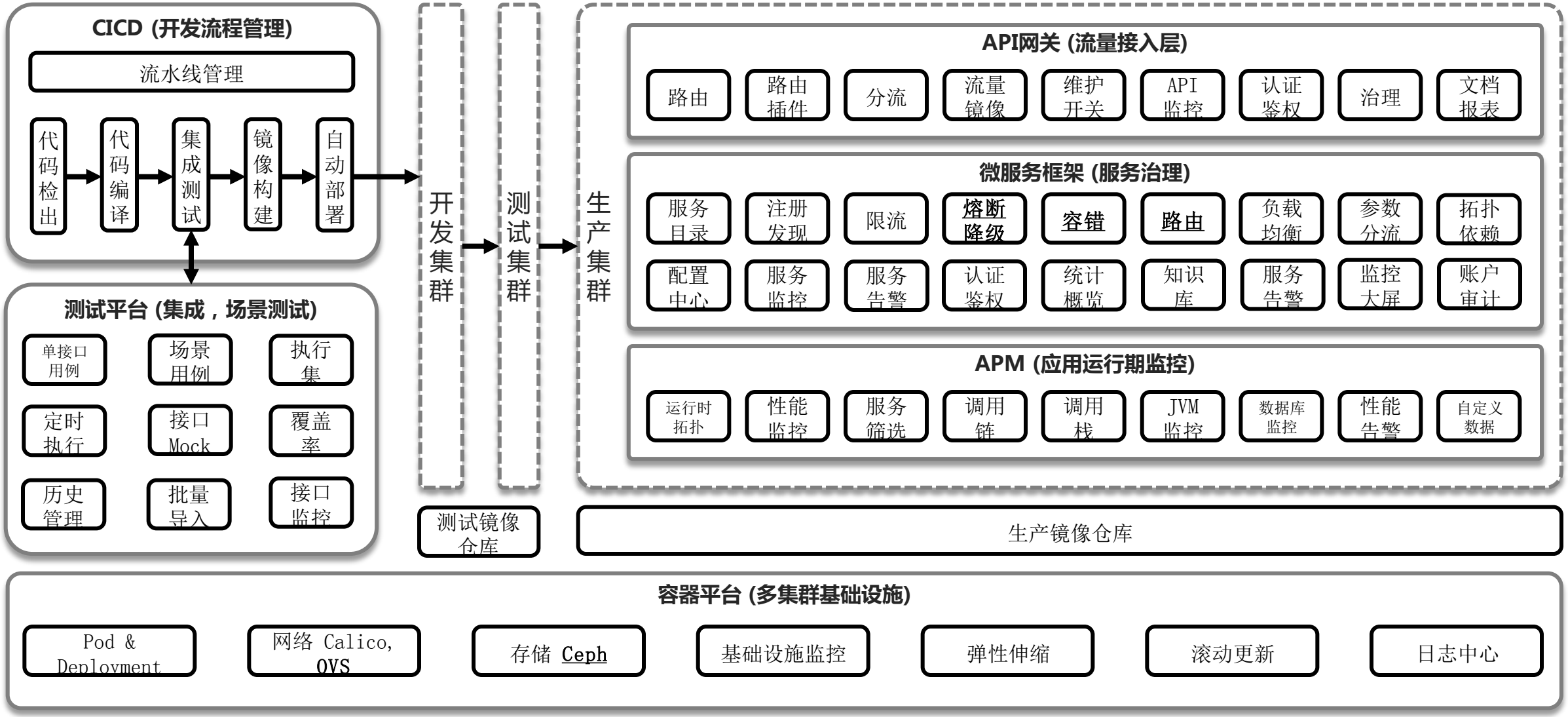
服务器性能监控	CPU、内存、网卡、磁盘、TCP……
应用性能监控	JVM堆内存、GC、Thread、CPU利用率、Method性能……
业务指标监控	下单数、支付数、购物车请求数……
调用链路监控	RT、TPS、Exception……
底层组件监控	RT、TPS、连接数、连接状态、消息积压、zk节点数……
系统异常监控	流量尖刺、Exception log、服务线程数、异常报警……

设计要点十二：全链路压测

容量测试	采用梯度压力，看服务的性能变化情况，评估出服务的最大容量值。
摸高压测	在达到停止条件之后，继续增加压力，检验服务集群在失效状态下的表现。
峰值稳定性测试	在峰值压力下，保持30分钟（可讨论）稳定
脉冲流量测试	制造脉冲式的压力，检验系统在脉冲压力下的表现是否稳定。
秒杀场景测试	针对秒杀类业务，制定秒杀测试场景
限流演练	多级限流，保护系统稳定提供服务
降级演练	非核心业务降级，提升整体服务能力
预案演练	实施预案演练，应对突发问题
故障演练	针对特定服务故障注入，观察服务的高可用、稳定可靠性
安全测试演练	负责人给出安全测试演练用例说明



微服务平台总览


CICD (开发流程管理)
流水线管理
代码检出
代码编译
集成测试
镜像构建
自动部署
测试平台 (集成，场景测试)
单接口用例
场景用例
执行集
定时执行
接口Mock
覆盖率
历史管理
批量导入
接口监控
开发集群
测试集群
测试镜像仓库
生产集群
API网关 (流量接入层)
路由
路由插件
分流
流量镜像
维护开关
API监控
认证鉴权
治理
文档报表
微服务框架 (服务治理)
服务目录
注册发现
限流
熔断降级
容错
路由
负载均衡
参数分流
拓扑依赖
配置中心
服务监控
服务告警
认证鉴权
统计概览
知识库
服务告警
监控大屏
账户审计
APM (应用运行期监控)
运行时拓扑
性能监控
服务筛选
调用链
调用栈
JVM监控
数据库监控
性能告警
自定义数据
生产镜像仓库
容器平台 (多集群基础设施)
Pod & Deployment
网络 Calico, OVS
存储 Ceph
基础设施监控
弹性伸缩
滚动更新
日志中心

多样需求

持续集成

中台化

全链路压测

服务管理

性能监控

服务治理

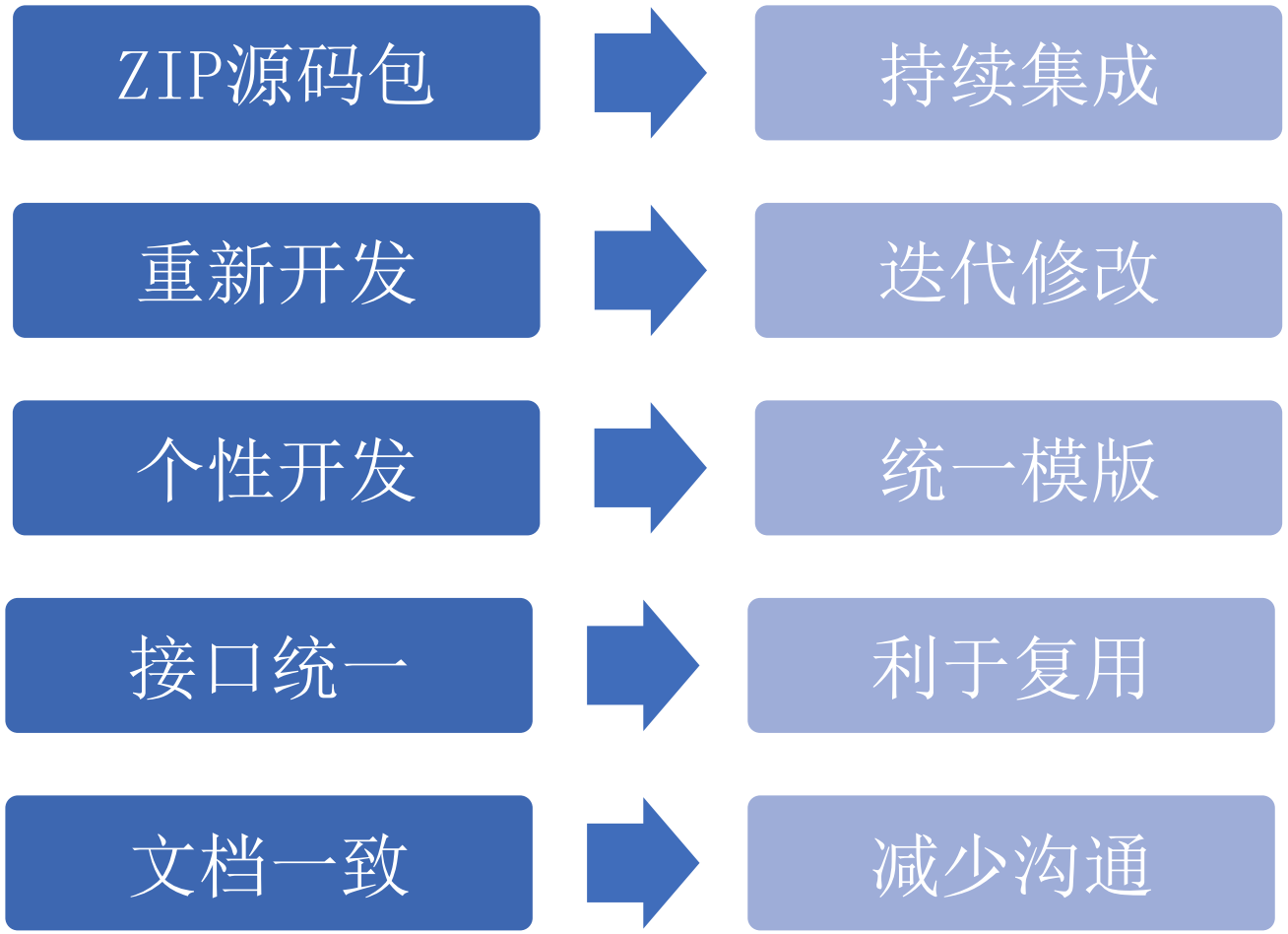
注册发现

容器化

分布式事务

PaaS托管

某视频监控企业：IT资产沉淀与IT能力复用



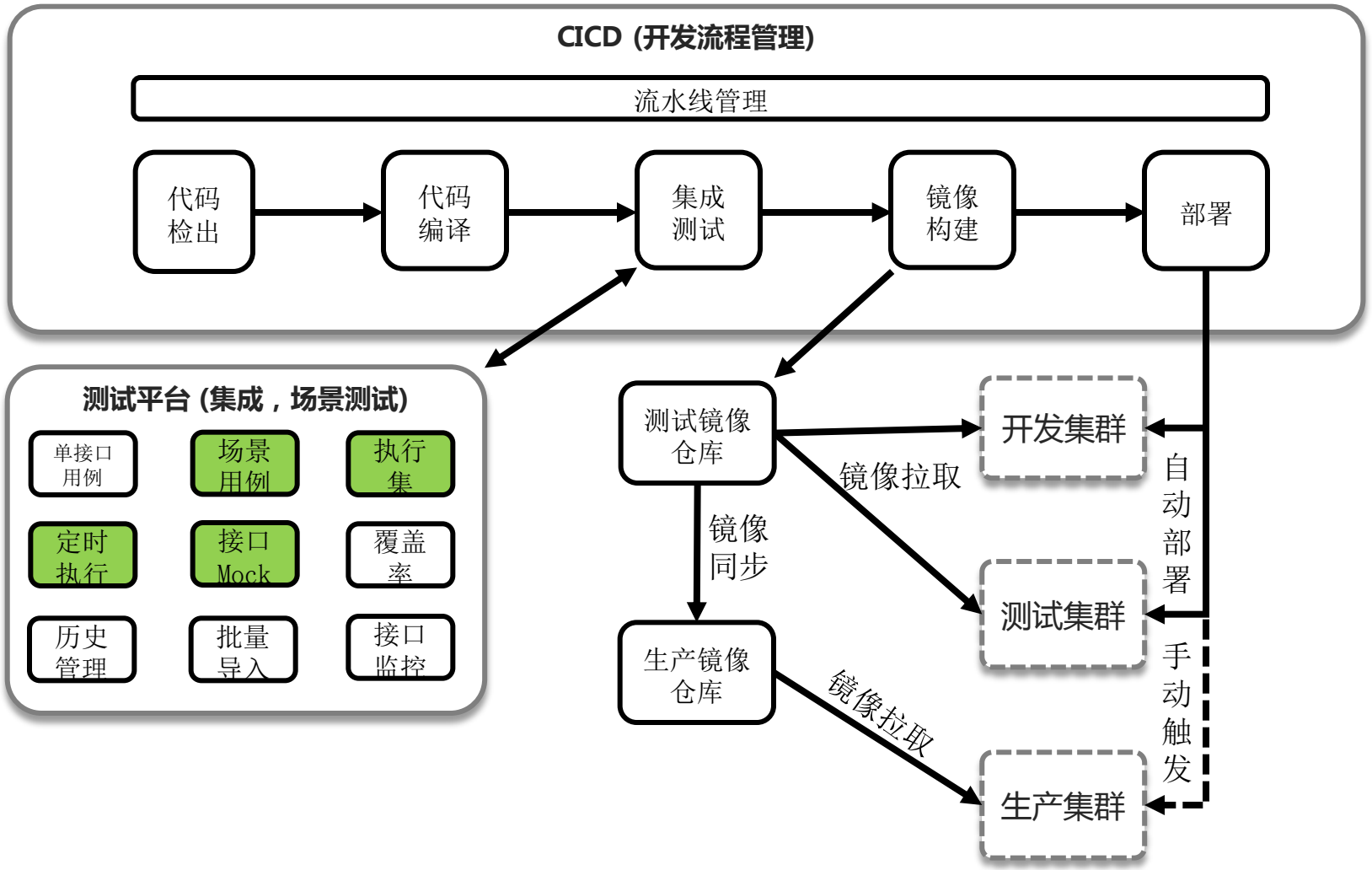
持续集成

服务管理

注册发现

容器化

持续集成



代码到线上全流程管理

互联网化接口测试，场景测试

灵活执行集：冒烟，日常，回归

定时测试，MOCK测试

测试，生产镜像仓库分离，自动同步

开发测试环境自动部署，生产环境手动触发

微服务框架负责服务之间的调用——企业级特性

认证鉴权　　注册，发现，调用都提供鉴权

知识库　　接口文档统一维护
文档与运行时一致
减少调用沟通成本

账户审计　　根据平台、租户、项目三个层次区分权限作用域
操作记录，审计日志，事件查询

某证券公司

中台化

持续集成

注册发现

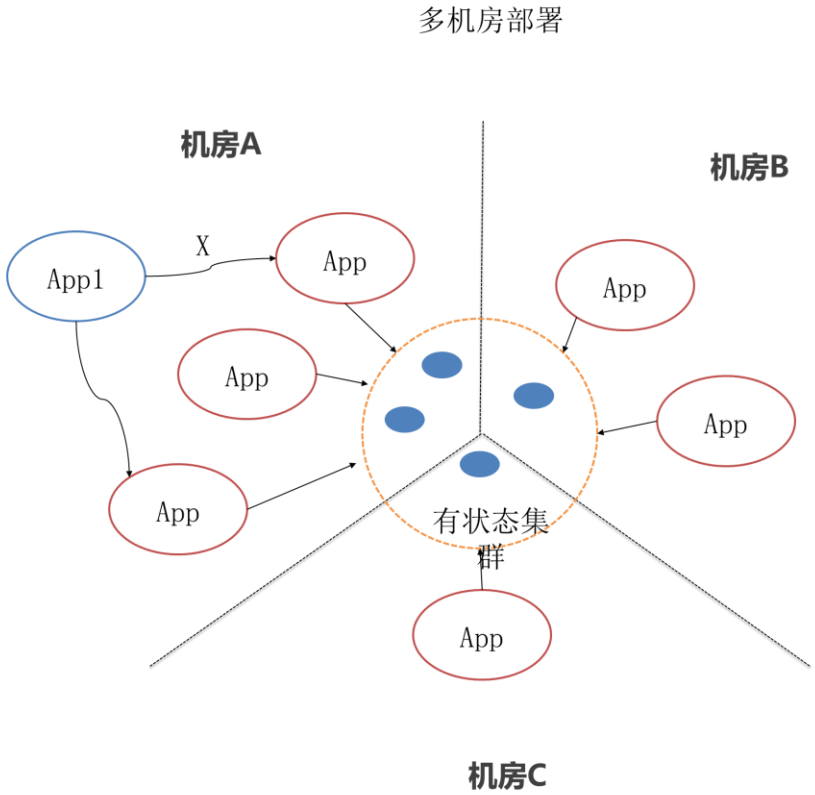
容器化

独立状态集群

服务自动发现

失败自动熔断

多机房部署



网易容器平台优势

集群规模大：30000+节点

全球首批通过K8S一致性认证

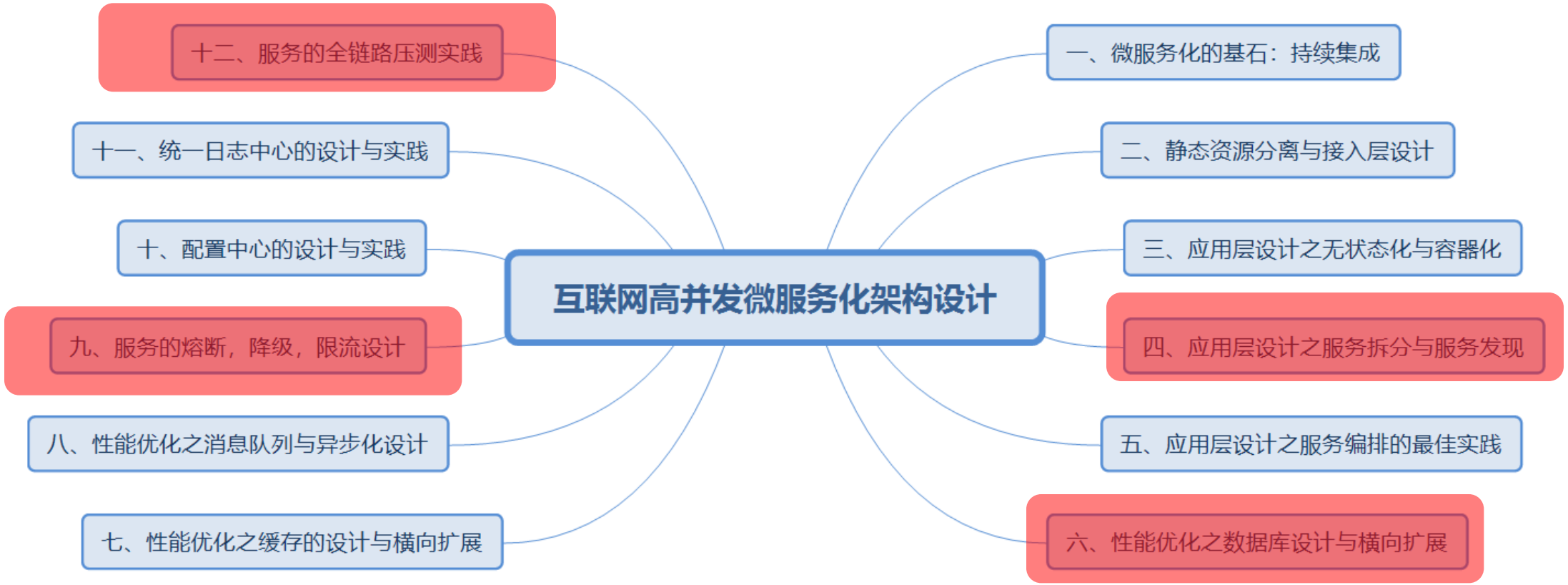
生产检验时间长：国内首个K8S公有云容器平台稳定运行1000+天

基于OVS的网络性能优化

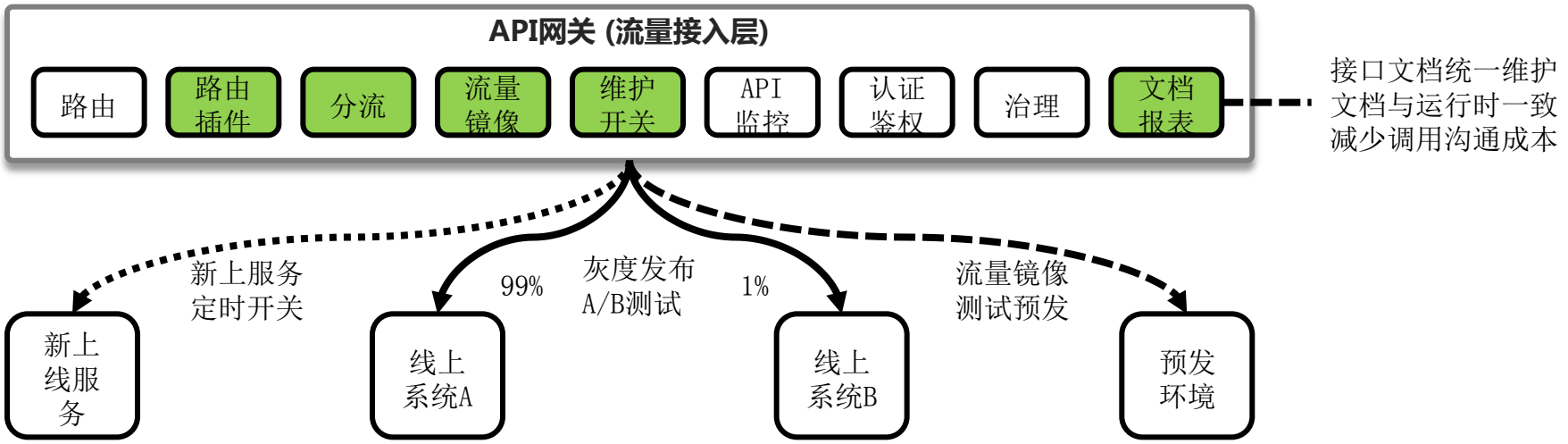
基于Ceph的存储性能优化

多集群统一管理

某物流企业


互联网高并发微服务化架构设计
一、微服务化的基石：持续集成
二、静态资源分离与接入层设计
三、应用层设计之无状态化与容器化
四、应用层设计之服务拆分与服务发现
五、应用层设计之服务编排的最佳实践
六、性能优化之数据库设计与横向扩展
七、性能优化之缓存的设计与横向扩展
八、性能优化之消息队列与异步化设计
九、服务的熔断，降级，限流设计
十、配置中心的设计与实践
十一、统一日志中心的设计与实践
十二、服务的全链路压测实践

API网关负责流量接入



可自行定制：路由插件，可开发插件拦截请求，进行定制化

微服务框架负责服务之间的调用——熔断与容错

熔断　　粒度更细：可指定服务版本，类，方法级别
配置灵活：可配置检测粒度为每M毫秒N个请求P%的错误率
指标多样：RT值，错误率，线程池参数

容错　　粒度更细：可指定调用者和被调用者服务版本，支持failover、failfast、failback容错机制。
配置灵活：支持自定义超时时间和重试次数。
可自行定制：通过暴露自定义异常NSFExcetion解决任意业务方法的容错，支持超时、failover，failfast容错。

微服务框架负责服务之间的调用——路由

路由　　可配置多条规则，按优先级匹配

可配置消费端黑白名单：只有A服务能访问B服务，只有IP1能访问B服务，更加安全

可配置服务端黑白名单：A服务版本1访问B服务版本1，A服务版本2访问B服务版本2，更灵活

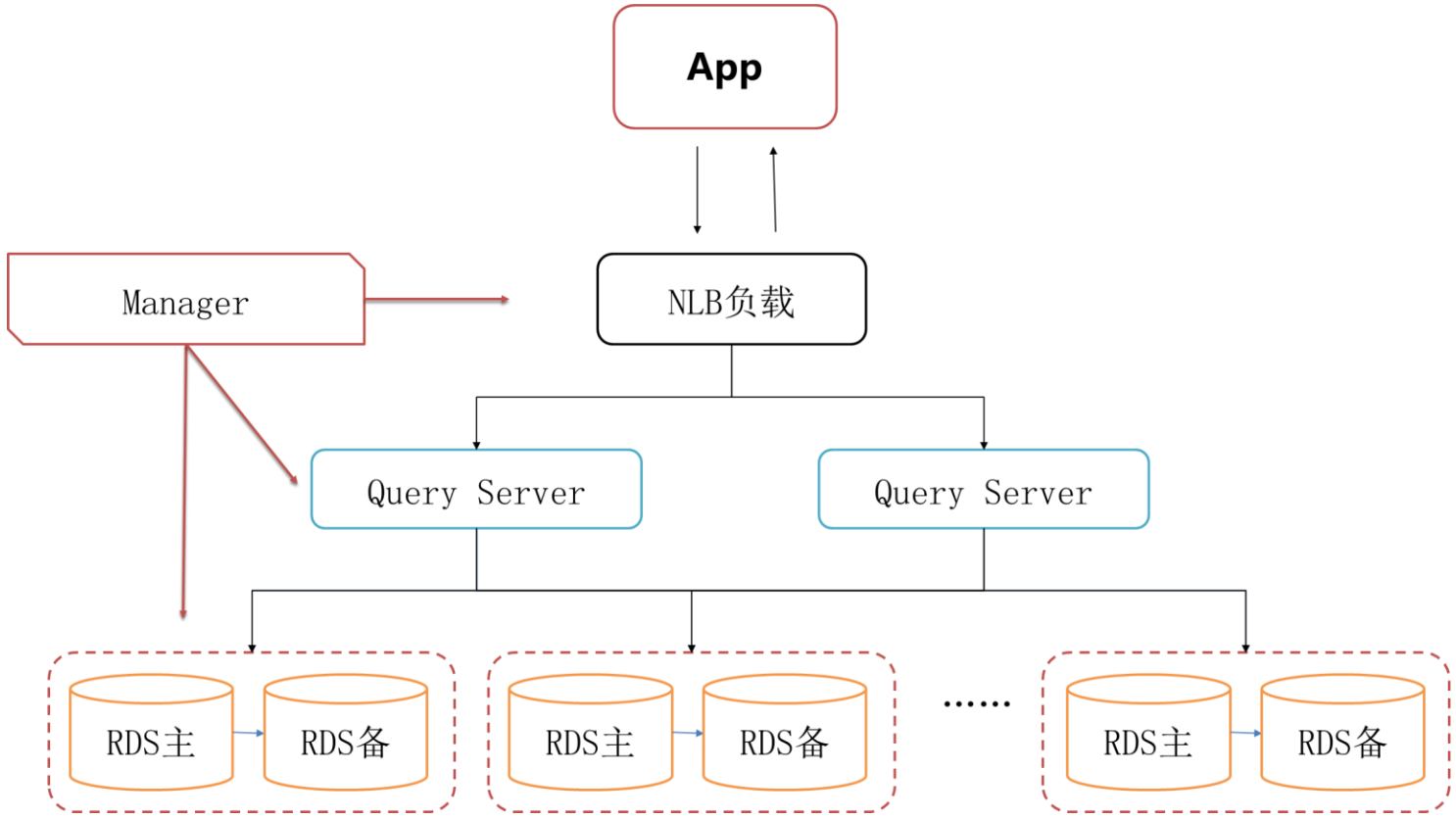
微服务框架负责服务之间的调用——负载均衡与参数分流

参数分流　　负载均衡规则的高级补充，可以通过参数判断，决定请求流量的流向。
参数取模时，可针对 Cookie、HTTP Header 或 Query String 的参数进行取模运算。
名单分流时，名单支持正则表达式。

A用户永远只访问A服务v1

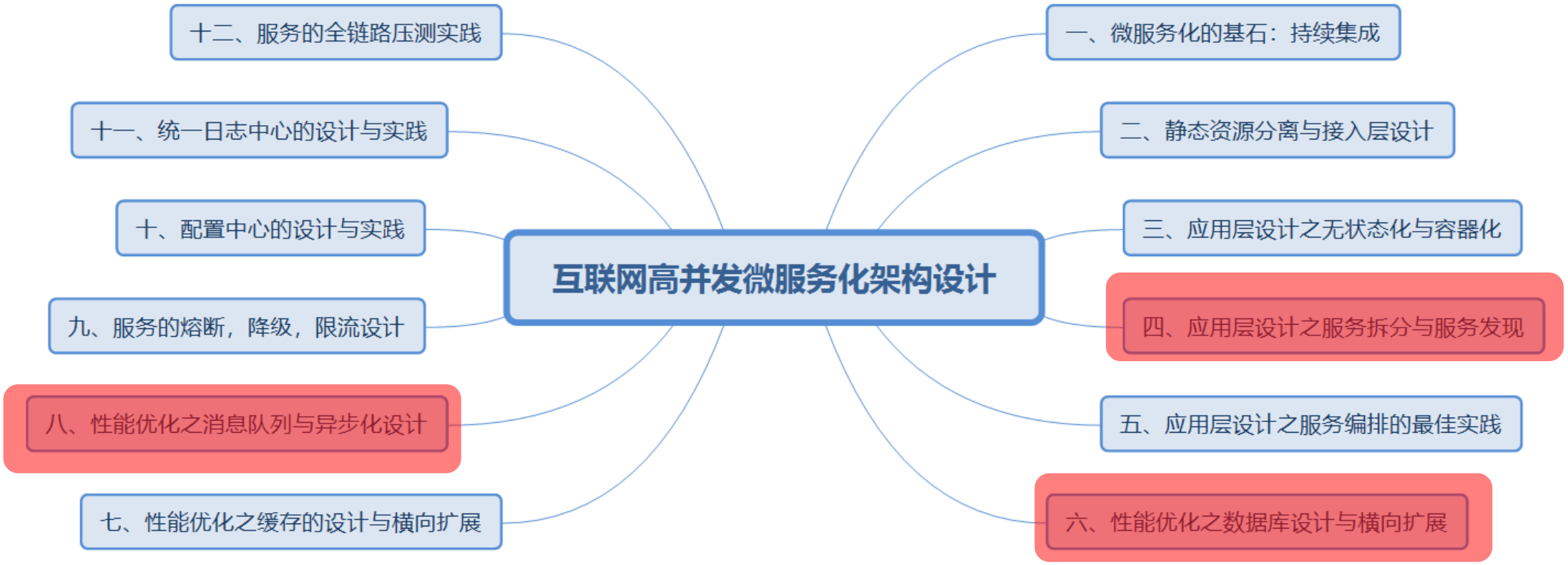
VIP用户访问A服务V2，非VIP用户访问A服务V1

分布式数据库



- 分布式部署
- 底层节点主备
- 负载均衡
- QS自动切换
- 双机房部署
- 读写分离

某大型银行


互联网高并发微服务化架构设计
一、微服务化的基石：持续集成
二、静态资源分离与接入层设计
三、应用层设计之无状态化与容器化
四、应用层设计之服务拆分与服务发现
五、应用层设计之服务编排的最佳实践
六、性能优化之数据库设计与横向扩展
七、性能优化之缓存的设计与横向扩展
八、性能优化之消息队列与异步化设计
九、服务的熔断，降级，限流设计
十、配置中心的设计与实践
十一、统一日志中心的设计与实践
十二、服务的全链路压测实践

网易分布式事务的实现机制

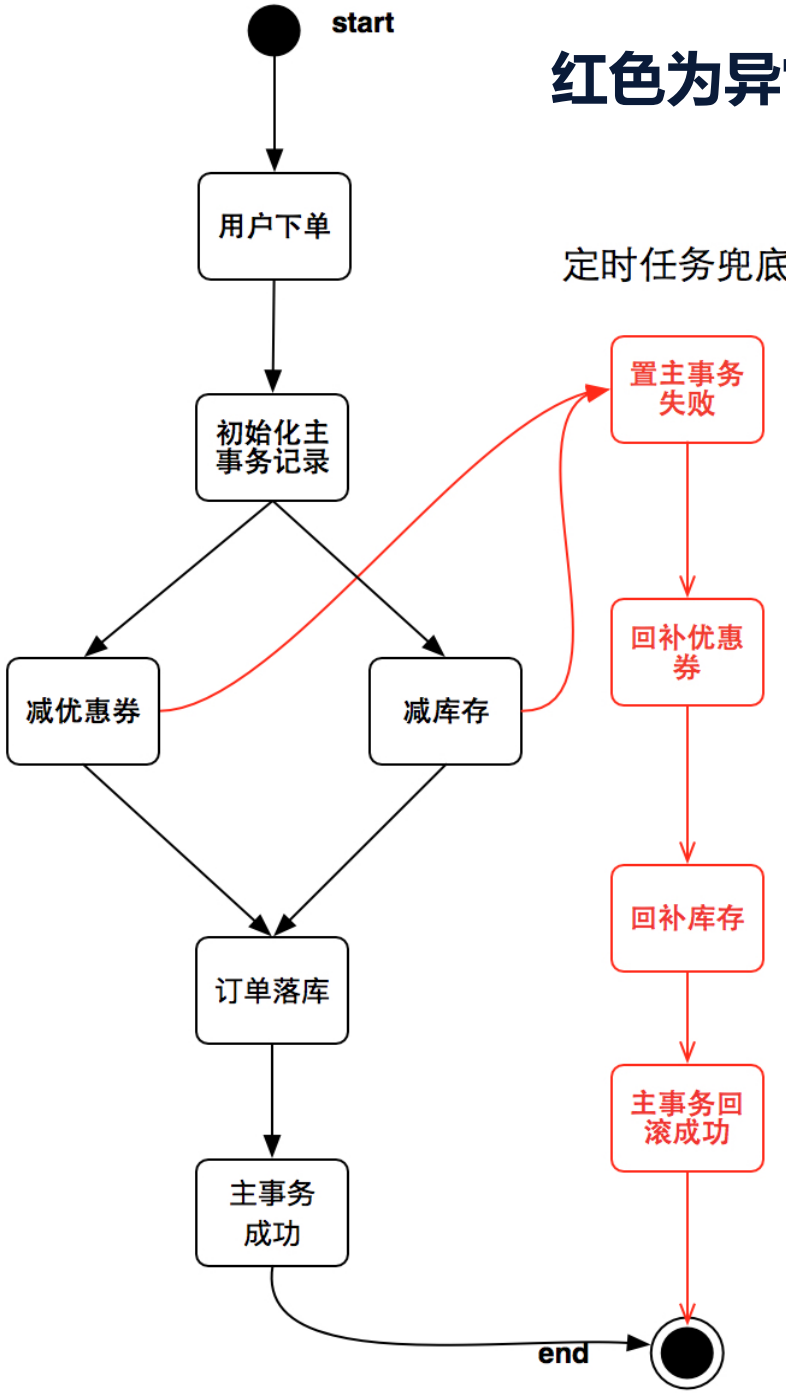
- 两阶段提交XA——中间件DDB
- TCC——中间件 Dubbo + DTS
  - Try 预留 + Confirm 提交 + Cancel 还预留
  - Try 操作 + Confirm 无操作 + Cancel 补偿
- 事务消息——中间件 TMC

下单分布式事务

- 事务发起者、事务参与方
- 分支事务用事务id做幂等
  - 生成对应分支事务记录
- 并行调用
- 定时任务兜底

Q：回滚的时候，如果分支事务记录不存在，该怎么返回？

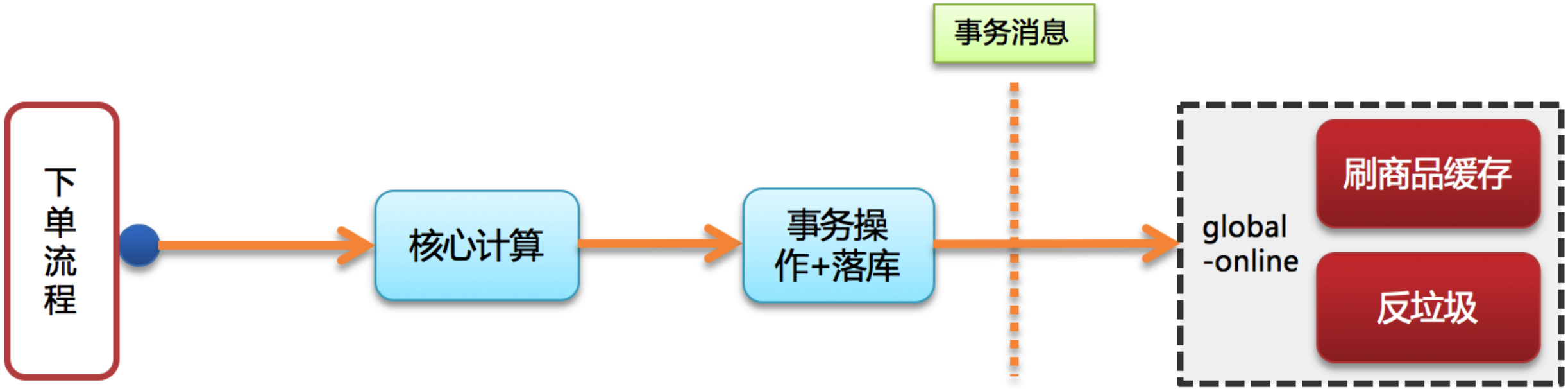
option1：返回成功 ✘
option2：返回失败 ✘
option3：引入超时判断机制 ✔



红色为异常流程

定时任务兜底执行

交易消息



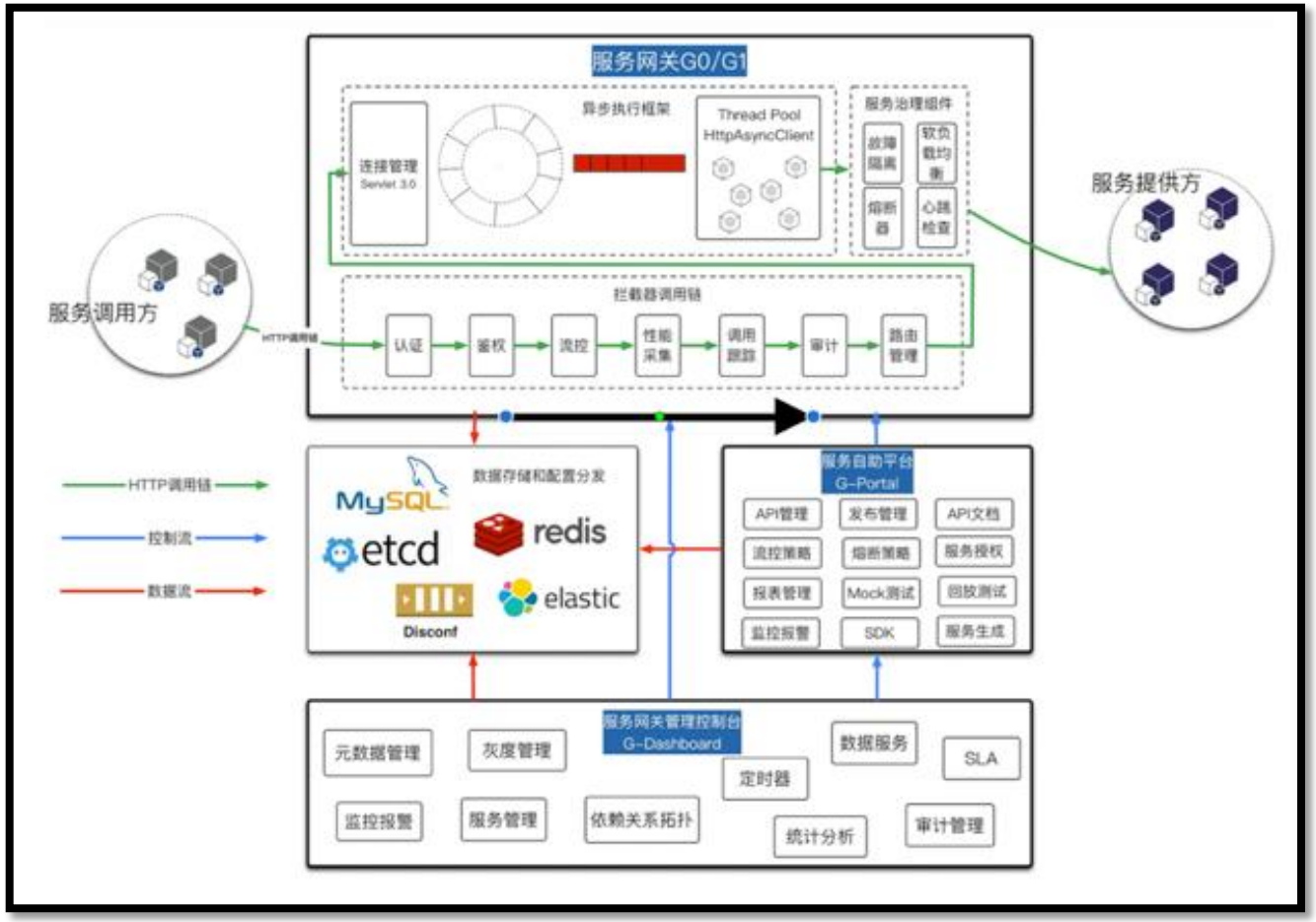
某制造企业

- 大量遗留应用，多种语言开发
- 服务间直接调用，依赖混乱
- 调用记录无迹可寻
- 环境与接口规范缺失，维护困难
- 安全靠白名单各自为战
- 调用统计与分析无从谈起
- 无服务治理能力，或各自造轮子

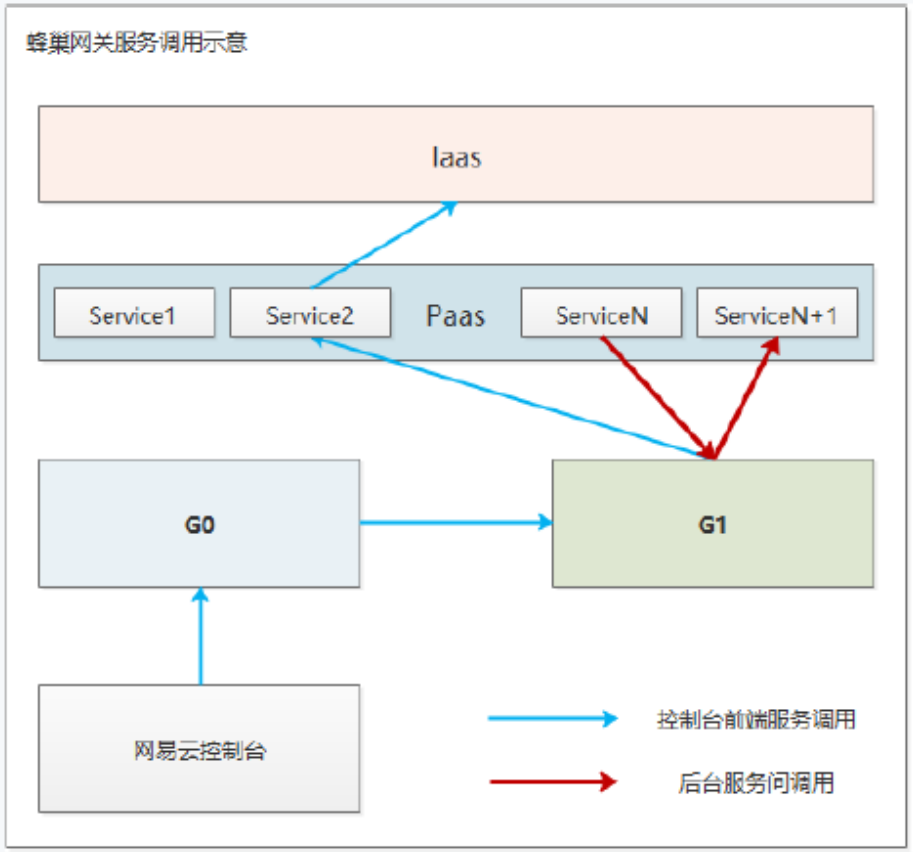
API网关

- API网关带来的能力
  - 请求透明代理与路由
  - 调用审计
  - 统一API规范与接口管理
  - 基于ak/sk的认证鉴权
  - 服务/API审计与调用分析
  - 不同维度故障隔离、服务熔断与降级
  - 不同维度的灵活流量控制
  - 自定义分流插件

……



从API网关到Service Mesh



现有网关调用示意图

目前G0/G1网关在一定程度上充当了“代理”的角色，基于Java语言实现

- G0本质上是一个FrontProxy
- G1本质上是一个ServiceToServiceProxy
- 功能方面支持流控、熔断（Hystrix）、负载均衡
- 服务治理方面有G-Dashboard
- 监控方面有APM

从API网关到Service Mesh