GIAC
全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

分布式服务架构下的混沌工程实践

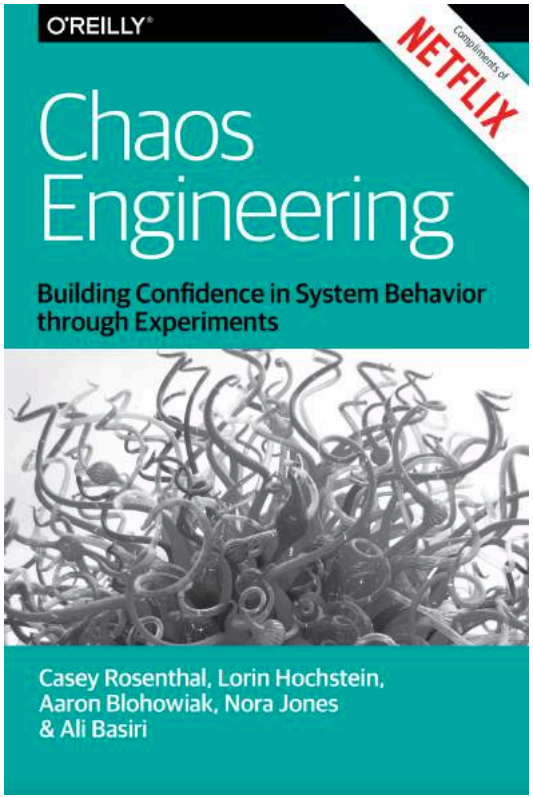
肖长军　　阿里巴巴 高级开发工程师

自我介绍

- 肖长军，花名 穹谷，阿里高可用架构团队
- 多年应用性能监控研发和分布式系统高可用架构经验
- 阿里云应用高可用服务（AHAS）产品核心开发
- 阿里集团故障演练、突袭演练、攻防演练核心开发
- 开源项目 ChaosBlade 负责人
- 混沌工程布道师

目录

混沌工程是什么



混沌工程是在分布式系统上进行实验的学科，旨在提升系统容错性，建立系统抵御生产环境中发生不可预知问题的信心。

What does not kill me, makes me stronger.
-- Nietzsche

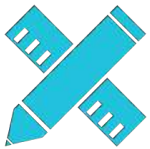
打不倒我的必使我强大。

为什么要实施混沌工程

架构师：验证系统架构的容错能力

开发&运维：提高故障的应急效率

测试：提早暴露线上问题，降低故障复发率



产品&设计：提升客户使用体验

实施混沌工程的原则

建立一个围绕稳定状态行为的假说

- 关注可测量输出，而不是系统内部属性。
- 短时间内的度量结果，代表了系统的稳定状态。
- 验证系统是否工作，而不是如何工作。

多样化真实世界的事件

- 混沌变量反映了现实世界中的事件。
- 通过潜在影响或预估频率排定事件的优先级。
- 任何能够破坏稳态的事件都是混沌实验中的一个潜在变量。

在生产环境中运行实验

- 系统的行为会根据环境和流量模式有所不同。
- 为了保证系统执行方式的真实性与当前部署系统的相关性，混沌工程强烈推荐直接采用生产环境流量进行实验。

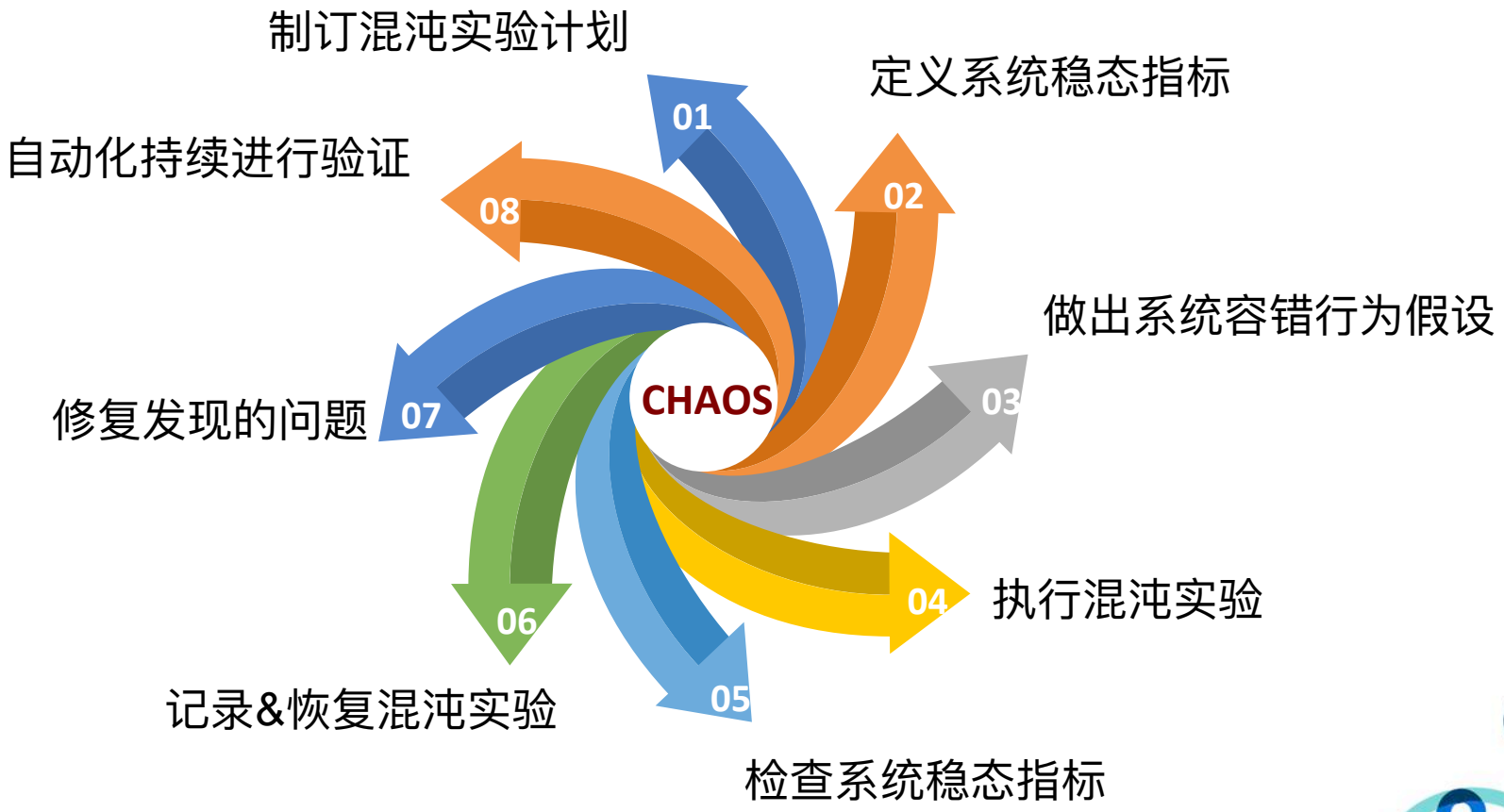
持续自动化运行实验

- 手动运行实验是劳动密集型的，最终是不可持续的，所以我们要把实验自动化并持续运行。
- 混沌工程要在系统中构建自动化的编排和分析。

最小化爆炸半径

- 在生产中进行试验可能会造成不必要的客户投诉。但混沌工程师的责任和义务是确保这些后续影响最小化且被考虑到。

实施混沌工程的步骤


制订混沌实验计划
01
定义系统稳态指标
02
做出系统容错行为假设
03
执行混沌实验
04
检查系统稳态指标
05
记录&恢复混沌实验
06
修复发现的问题
07
自动化持续进行验证
08
CHAOS

混沌工程如何在企业中落地?

落地三阶段

1

坚定混沌工程价值

承受各方挑战，推动项目落地

2

引入混沌工程技术

结合技术架构，选择实验工具

最小爆炸半径，控制实验风险

3

推广混沌工程文化

建立推广门户，宣传混沌工程

制订攻防制度，培育攻防文化

接受挑战，坚定混沌工程价值

业务方：

实施实验的依据是什么?

能给业务带来什么价值?

该如何修复发现的问题?

老板：

如何衡量混沌工程价值?

如何控制演练影响面?

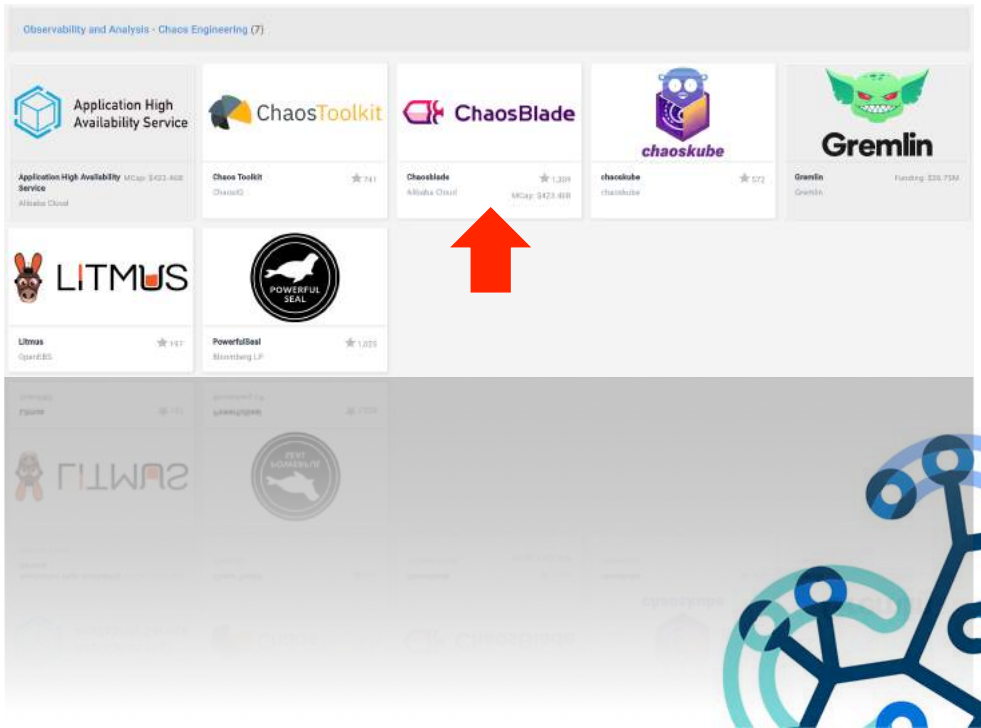
从系统成熟度，了解自身的系统

成熟度等级	1 级	2 级	3 级	4 级	5 级
架构抵御故障的能力	无抵御故障的能力	一定的冗余性	冗余且可扩展	已使用可避免级联故障的技术	已实现韧性架构
指标监控能力	无系统指标监控	实验结果只反映系统状态指标	实验结果反映应用的健康状态指标	实验结果反映聚合的业务指标	有对照组比较业务指标的差异
实验环境选择	只在开发和测试环境中运行	可在预生产环境中运行	复制生产流量在灰度环境中运行	在生产环境中运行实验	包含生产在内的任意环境都可以运行实验
故障注入场景爆炸半径范围	注入一些简单的事件，如CPU 高，IO 高等	进行一些较复杂的故障注入，如终止实例等	注入较高级的故障，如延迟、异常等	引入服务级别的影响和组合式的故障	可注入如对系统的不同使用模式、返回结果和状态的更改等故障

引自：AWS 黄帅-混沌工程实验成熟度等级

选择一款合适的混沌实验工具

- 场景丰富度
  进程、网络、应用、容器 …
- 工具类型
  实验工具、开发框架、产品平台
- 易用性
  低、中、高
- 构建语言
  Go、Java、Python …
- 活跃状态
  已停滞、维护、活跃



阿里混沌工程技术演进

EOS 强弱依赖

2012

2012

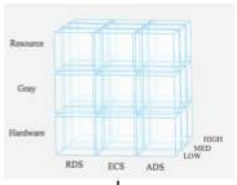
同城容灾 断网演练

虎!虎!虎! 异地多活

2015

2016

MonkeyKing 故障演练



2018

ACP

2018

公有云 AHAS

Application High Availability Service

ChaosBlade

2019

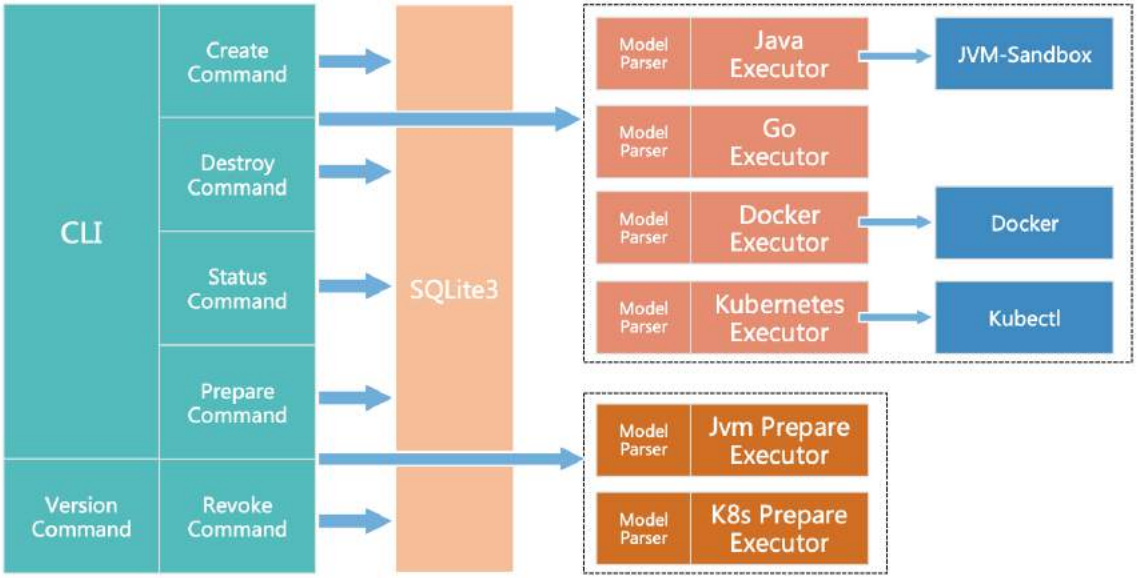
2019

专有云 混沌实验平台

阿里开源工具 ChaosBlade 介绍

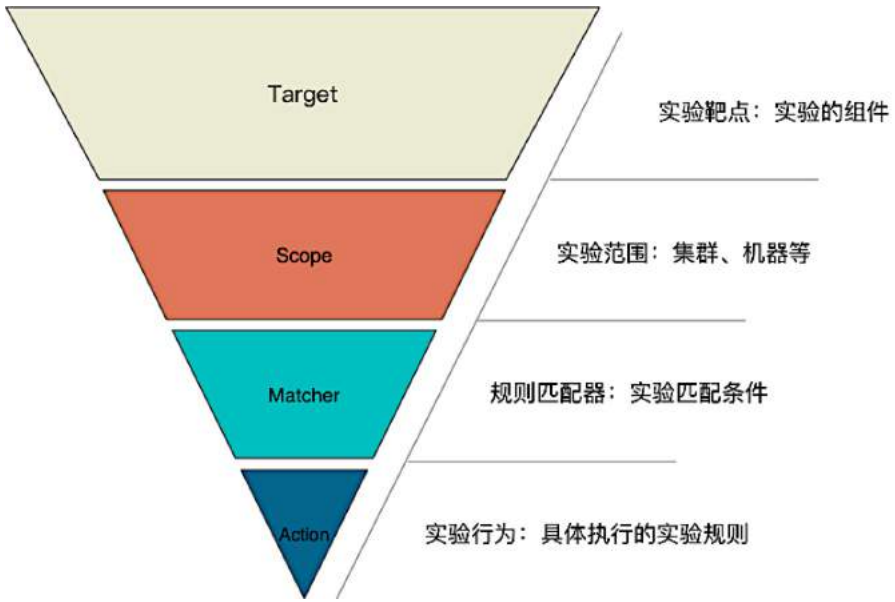
ChaosBlade（混沌之刃）是一款遵循混沌实验模型，提供丰富故障场景实现，旨在帮助分布式系统提升容错性和可恢复性的混沌工程工具。

ChaosBlade



Github 地址：https://github.com/chaosblade-io

统一实验模型，沉淀故障场景



简洁，层次清晰通俗易懂

四层，边界清晰

通用，覆盖目前所有故障场景

基础资源、应用、容器或 serverless 架构

易实现，实验场景共建简单

定义清晰的接口规范

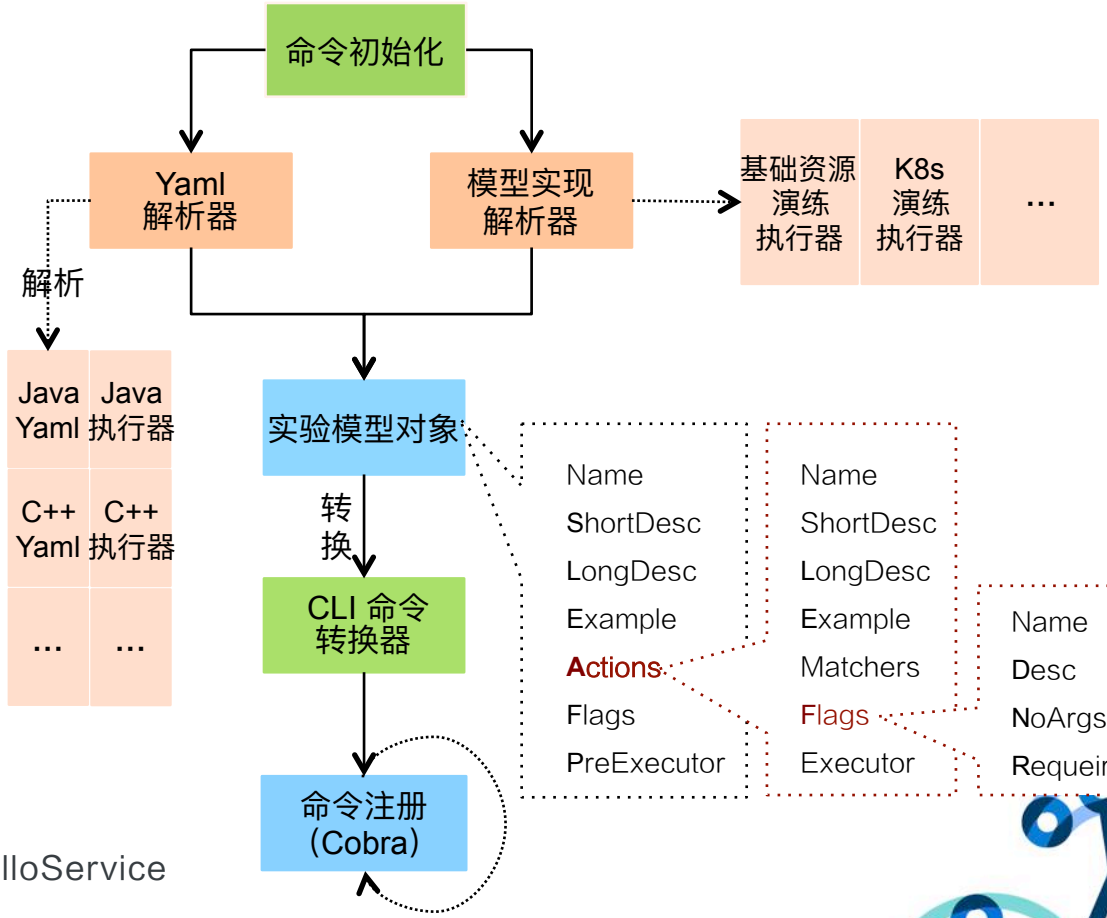
语言、领域无关

可以扩展多语言、多领域实现

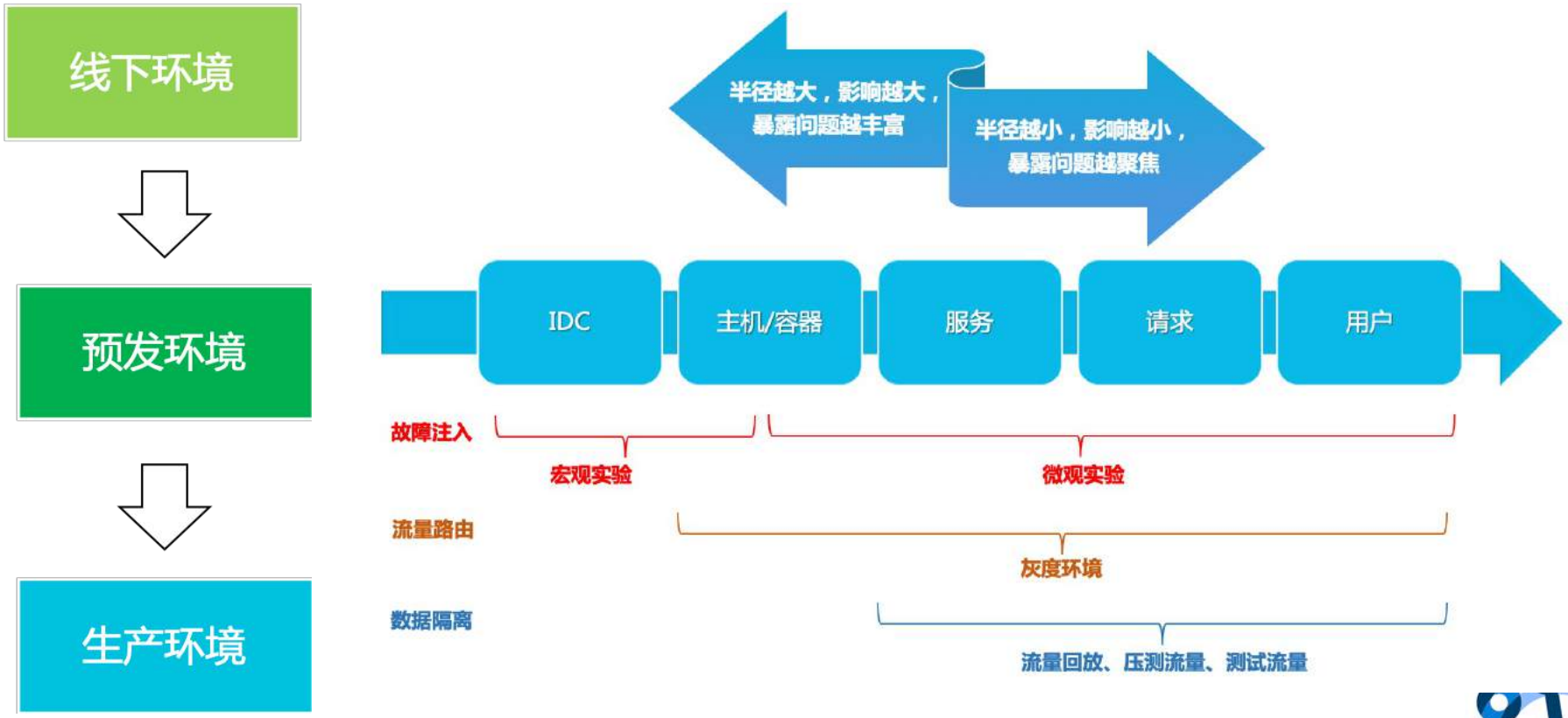
ChaosBlade 基于实验模型的架构设计

- 开箱即用，无需安装
- 支持命令提示
- 所有变量参数化
- 所有参数规范化
- 模块化，支持动态扩展
- 对象化，方便管理

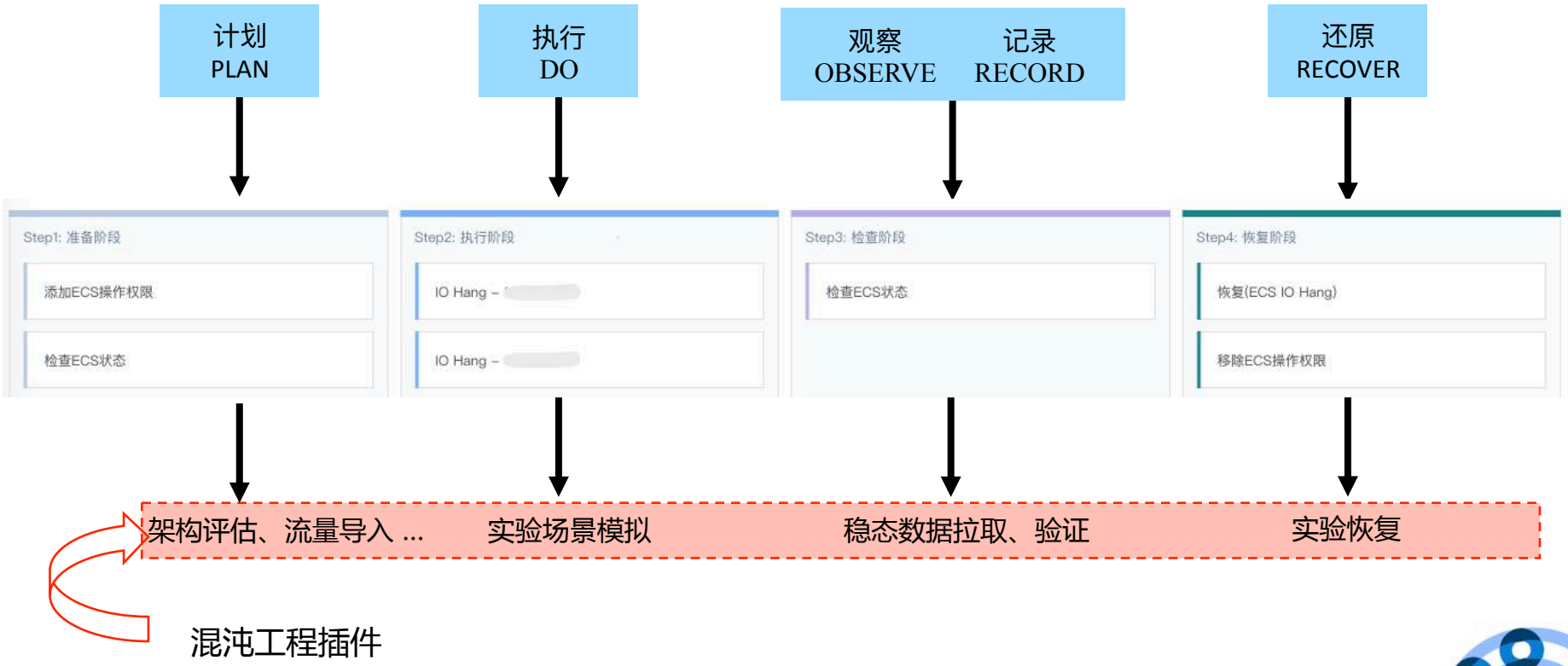
```
blade create cpu fullload

blade destroy 7c1f7afc281482c8

blade create dubbo delay
        --time 3000
        --service com.alibaba.demo.HelloService
        --consumer
```



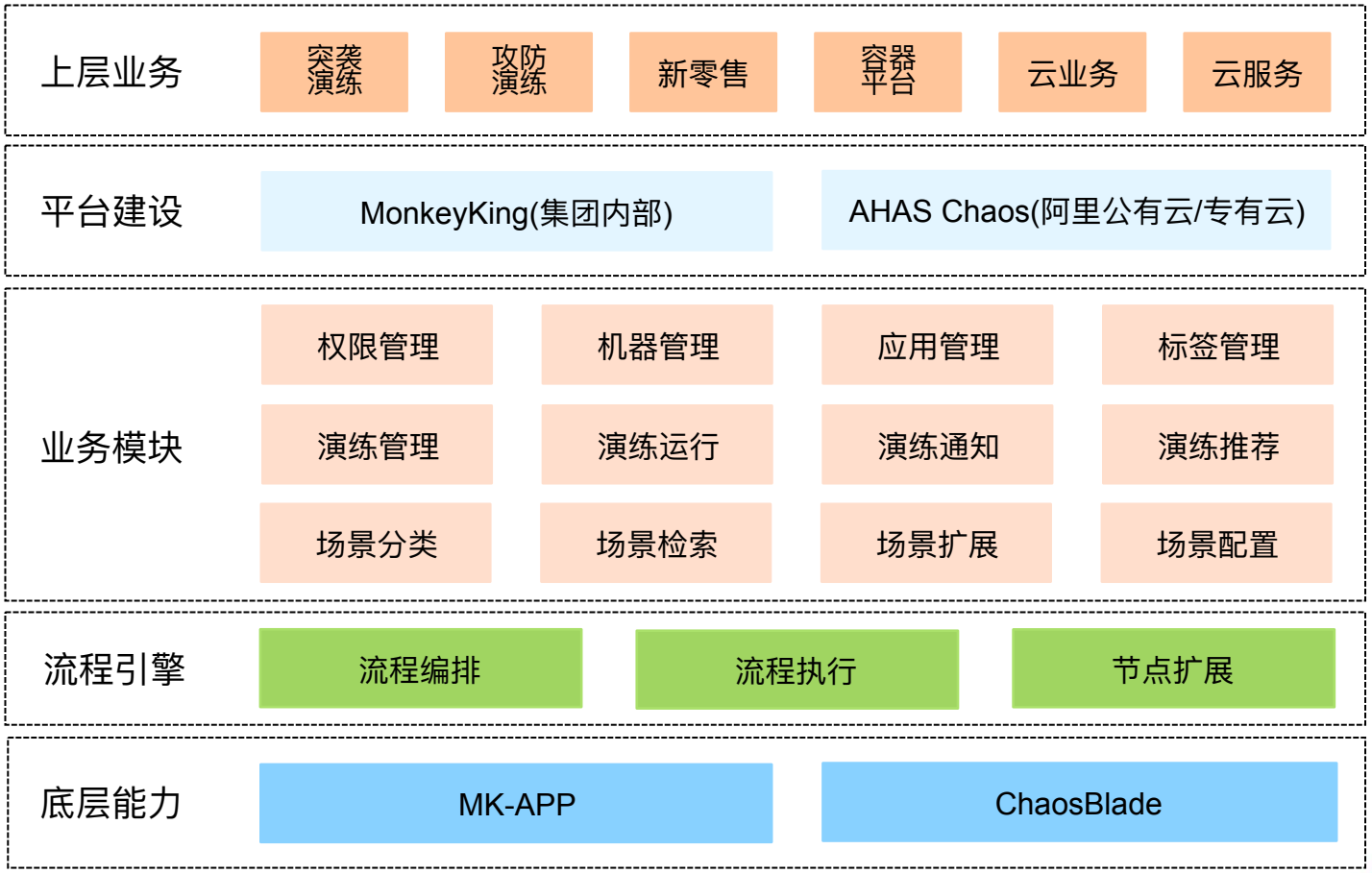
控制爆炸半径，减小实施风险



通过平台能力，标准化实验流程



建设实验平台，提升规模化能力



建立混沌工程文化

建立推广门户

- 日常红黑榜，每周推送
- 技术专栏，推广好的架构

制订攻防制度

- 设定故障分，推动常态化演练
- 设定演练分，衡量突袭演练
- 常态攻防，培养风险氛围
- 大型攻防，建立固定攻防日

分布式服务下混沌工程实践

分布式服务系统面临的问题

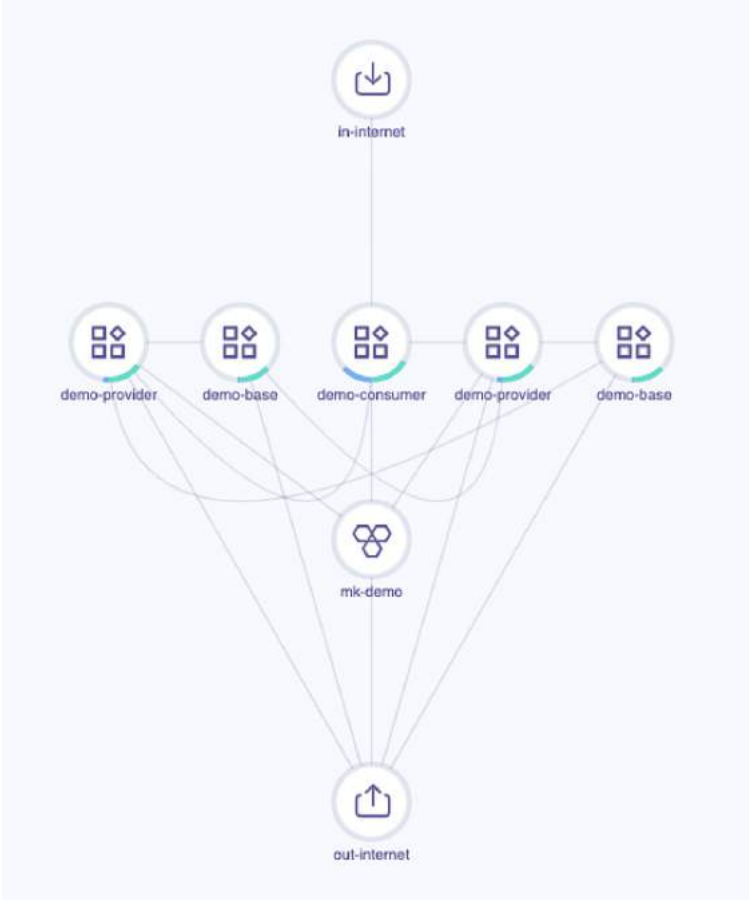
- 分布式系统日益庞大
  很难评估单个故障对整个系统的影响
- 服务间的依赖错综复杂，配置不合理
  单个服务不可用可能拖垮整个服务
- 请求链路长，监控告警、日志记录等不完善
  定位问题难
- 业务、技术迭代速度快
  系统稳定性受到更大的挑战

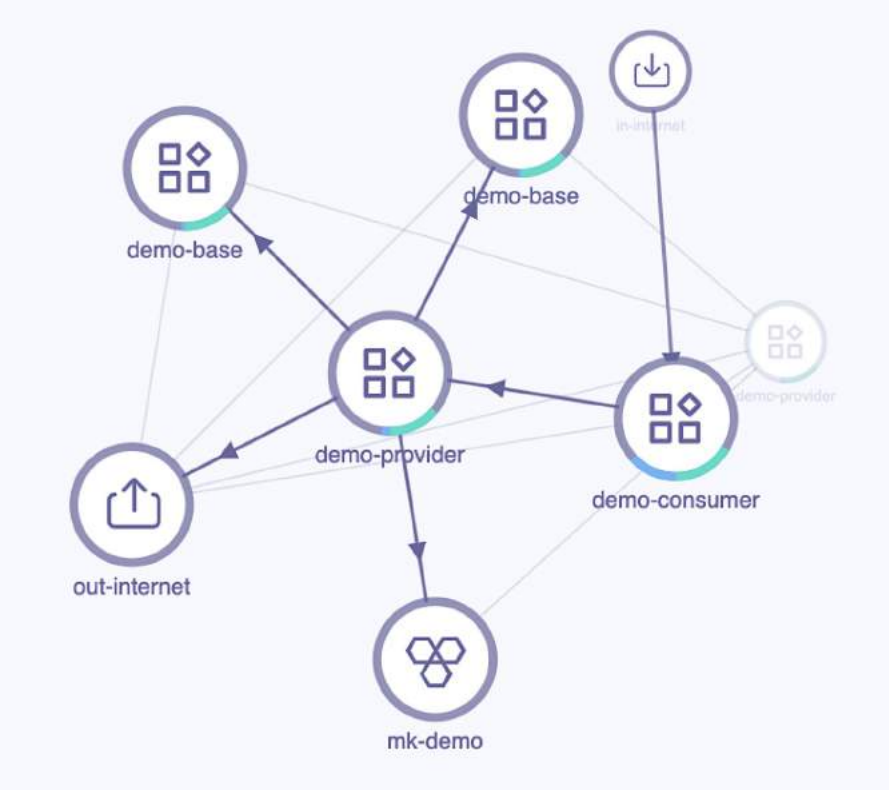
分布式服务系统高可用原则

入口服务	下游服务	应用进程	消息服务	数据缓存	数据存储	系统运维
• 负载均衡	• 超时重试	• 资源隔离	• 异步传递	• 热点隔离	• 读写分离	• 监控告警
• 流量调度	• 服务降级	• 异步调用	• 消息分级	• 热点散列	• 分库分表	• 日志跟踪
• 请求限流	• 调用熔断	• 热点防护	• 削峰填谷	• 主从备份	• 主从备份	• 健康检查
	• 强弱依赖		• 消息存储		• 一致性保障	• 灰度发布
	• 幂等处理					• 发布回滚
	• 最优调用					• 弹性伸缩
						• 容量规划
						• 服务治理
						• 异地多活

混沌工程

案例 Demo 拓扑图



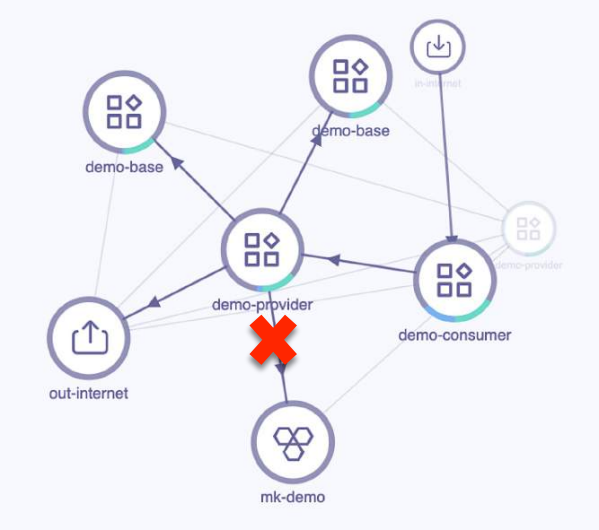


拓扑图来自于-阿里云 AHAS 产品架构感知功能

案例一：验证监控告警

场景：数据库调用延迟
监控指标：慢 SQL 数，告警信息
期望假设：慢 SQL 数增加，钉钉群收到慢 SQL 告警
混沌实验：对 demo-provider 注入调用 mk-demo 数据库延迟故障
监控指标：慢 SQL 数增加，钉钉群收到告警，符合预期
问题排查：通过 ARMS 慢调用链路排查

备注：以上告警和链路跟踪来自于阿里云 *ARMS* 产品

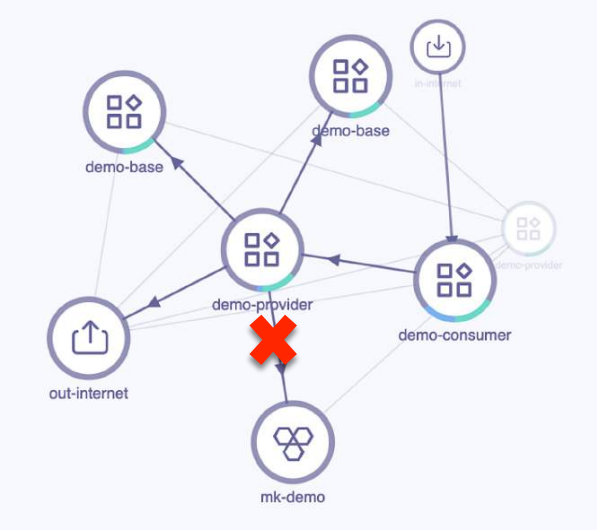



```
blade create mysql delay
        --time 600
        --database demo
        --table d_discount
        --sqltype select
        --effect-percent 50
```

案例一：验证监控告警

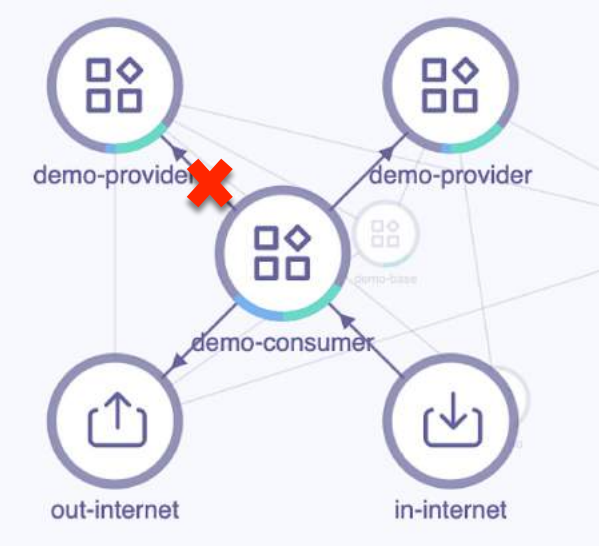
场景：数据库调用延迟
监控指标：慢 SQL 数，告警信息
期望假设：慢 SQL 数增加，钉钉群收到慢 SQL 告警
混沌实验：对 demo-provider 注入调用 mk-demo 数据库延迟故障
监控指标：慢 SQL 数增加，钉钉群收到告警，符合预期
问题排查：通过 ARMS 慢调用链路排查

备注：以上告警和链路跟踪来自于阿里云 *ARMS* 产品



案例二：验证异常实例隔离

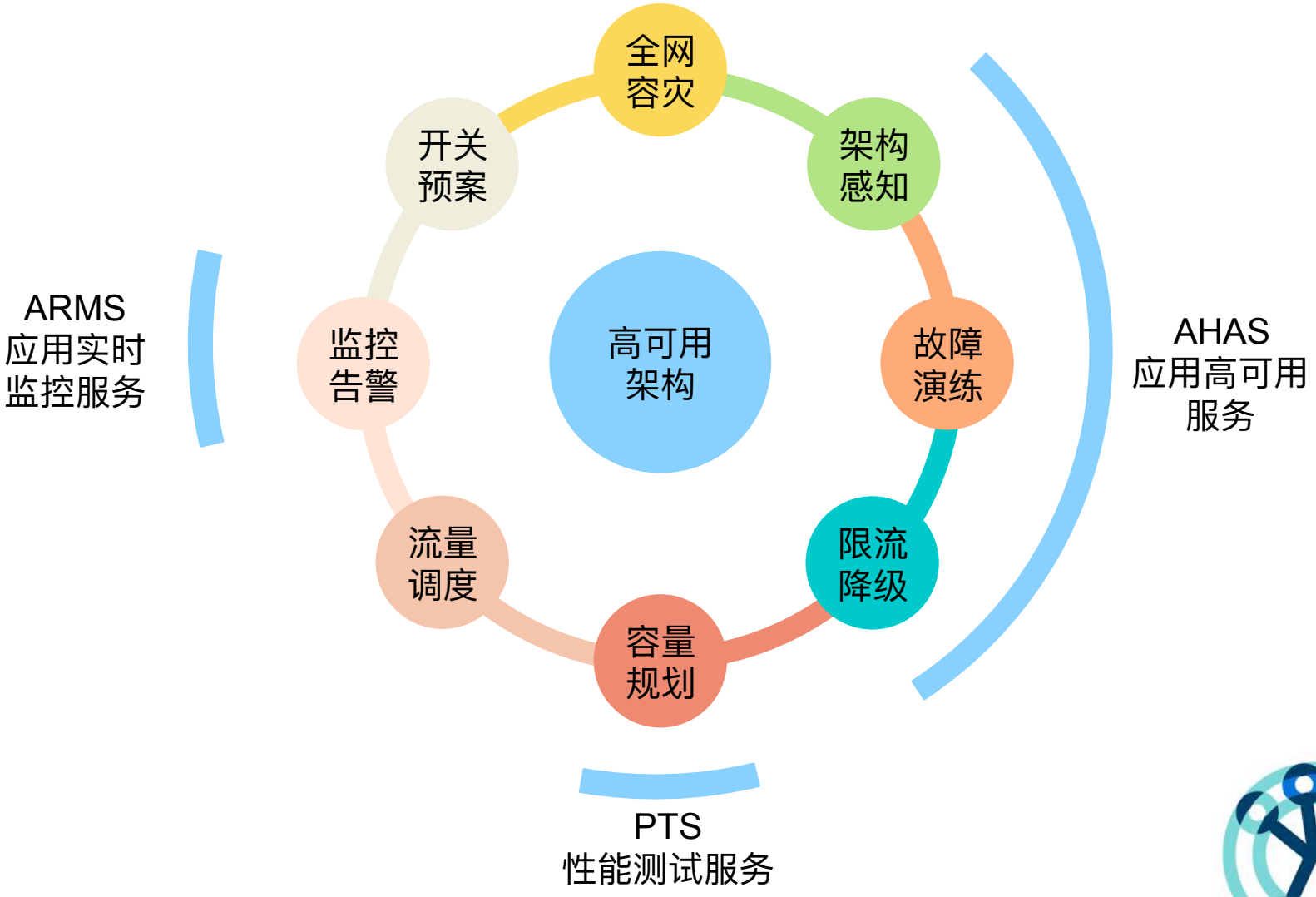
场景：下游一个服务实例出现延迟
监控指标：QPS，稳态在 510 左右
容错假设：QPS 会出现几秒的下跌，但很快恢复；系统会自动隔离或下线出问题服务实例，防止请求路由到此实例
混沌实验：对 demo-provider-1 注入延迟故障
监控指标：QPS 下跌到 40，不会自动恢复，不符合预期
业务方应急处理：下线出问题的实例，QPS 恢复
问题记录：系统缺失服务质量检查，不能对异常服务实例做隔离



回顾总结

- 混沌工程是一种主动防御的稳定性手段，体现了反脆弱的思想
- 落地混沌工程会遇到很多挑战，坚持原则不能退让
- 实施混沌工程不能只是把故障制造出来，需要有明确的驱动目标
- 选择合适的工具和平台，控制演练风险，实现常态化演练

阿里云高可用架构产品图（部分）



主办方：msup | 架構 ARCHNOTES

钉钉扫描

欢迎关注msup微信公众账号

关注大会微信公共账号，及时了解大会动态、日程及每日更新的案例！