

# 云原生架构下的混沌工程实践

周洋（花名：中亭）

阿里云智能事业群-高可用架构团队

# 自我介绍

- 阿里高可用架构团队，花名中亭
  - 多年高可用保障、产品研发和系统架构经验，2015年双11稳定性负责人
  - 目前负责高可用技术云化输出，应用高可用服务（AHAS）、集团突袭演练负责人
- 
- 2017 QCon明星讲师，《阿里电商故障治理和故障演练实践》
  - 2019 混沌工程布道师，开源项目ChaosBlade发起人

# 目录

- 混沌工程三连问

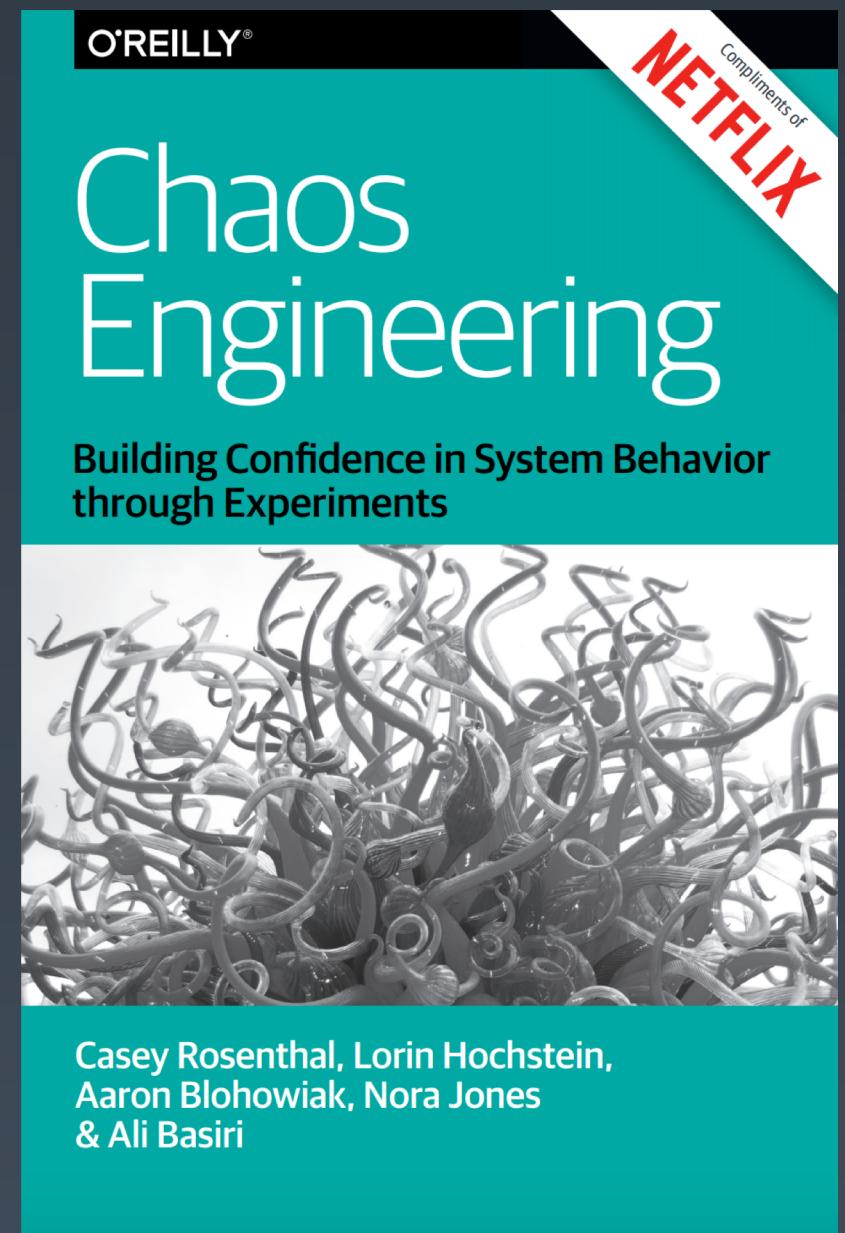
混沌工程是什么？

为什么要实施混沌工程？

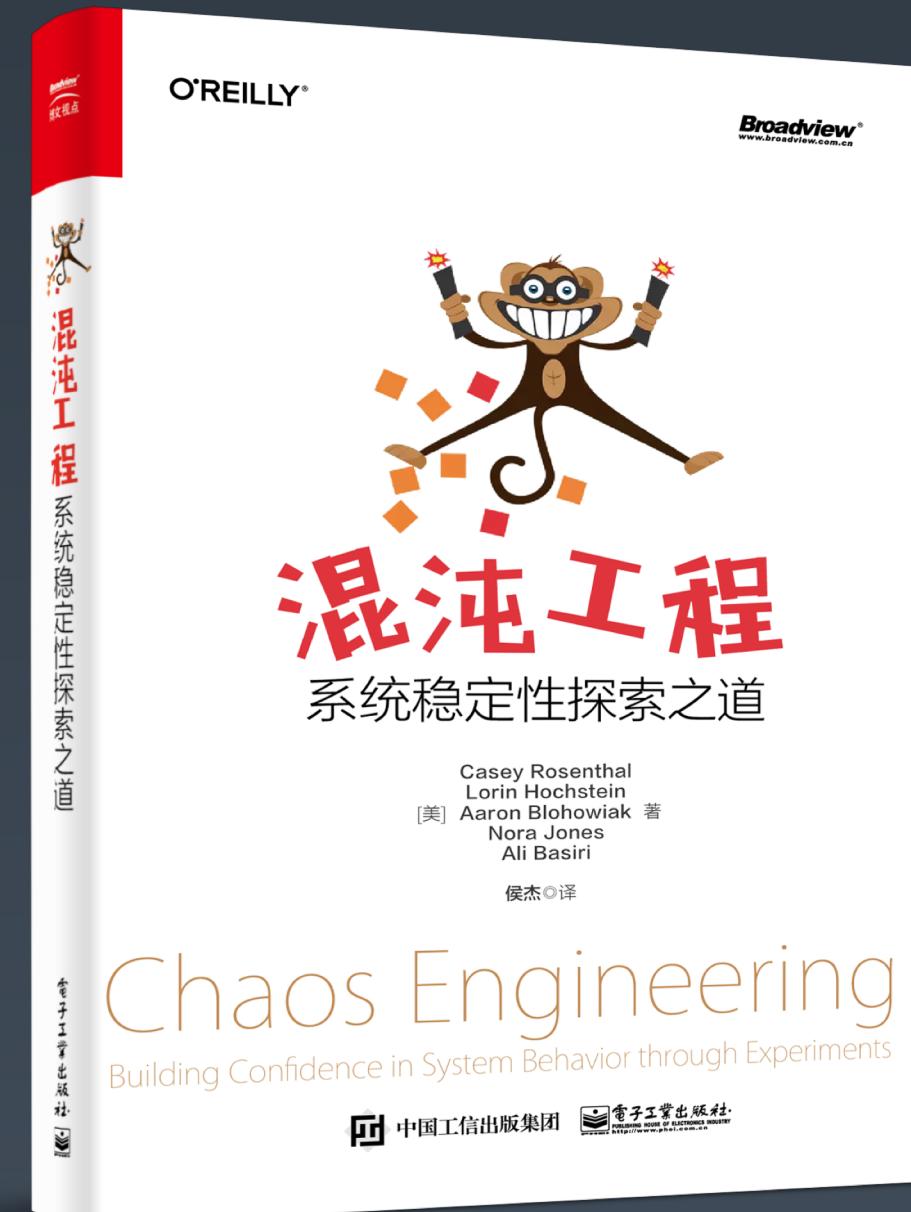
怎样实施混沌工程？

- 落地案例介绍
- 未来展望

# 混沌工程的概念



2017.8月出版



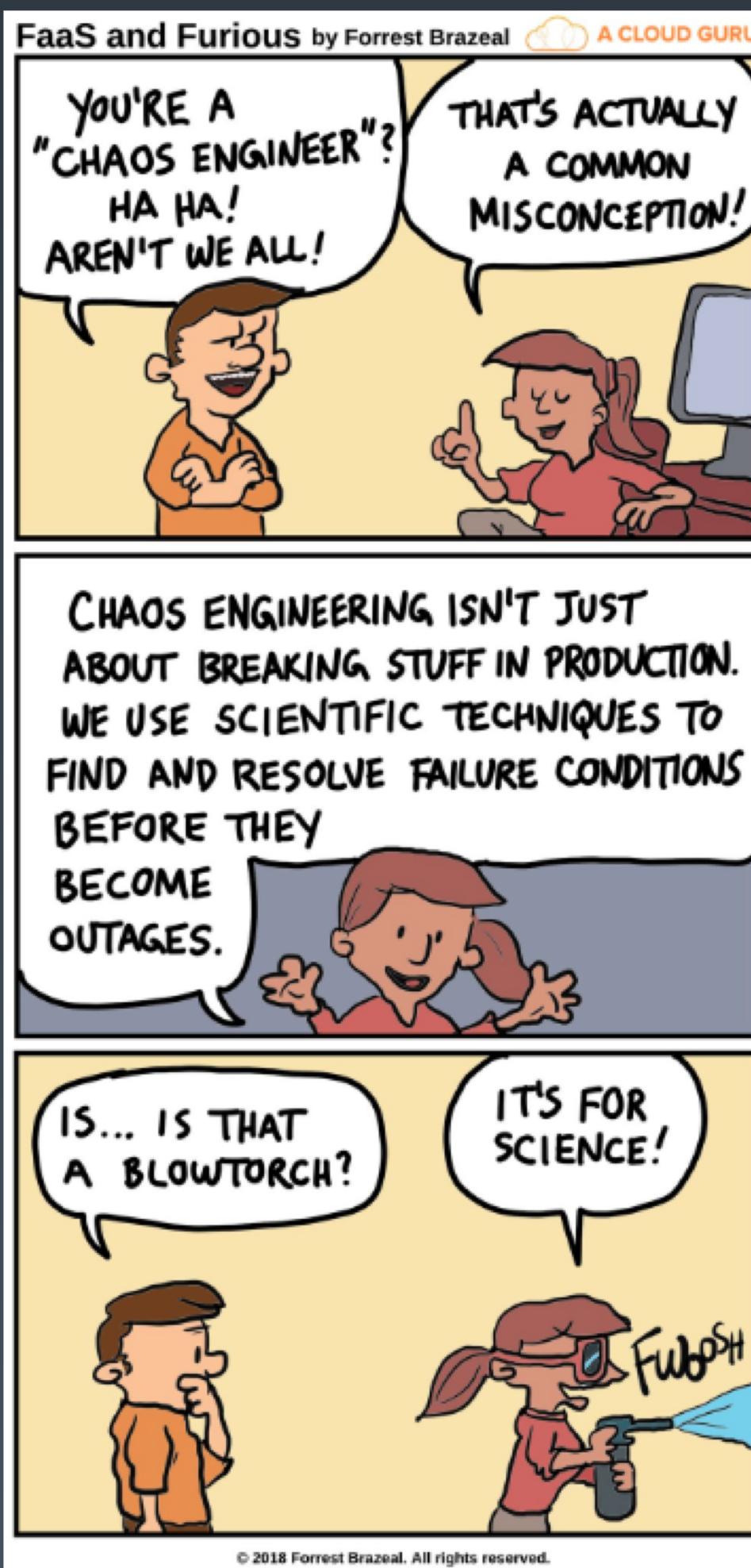
预计2019.6月上市

## PRINCIPLES OF CHAOS ENGINEERING

*Chaos Engineering is the discipline of experimenting on a distributed system  
in order to build confidence in the system's capability  
to withstand turbulent conditions in production.*

混沌工程是在分布式系统上进行实验的学科，  
目的是建立对系统抵御生产环境中失控条件  
的能力以及信心。

# 我对混沌工程的理解



一种拥抱失败的技术文化

一套抽象严谨的实践原则

一种主动防御的稳定性手段

一个高速发展的技术领域

# 混沌工程的起源



源自《*Chaos Engineering*》书籍

在过去五年左右的时间里，只有仅有的一次节点掉线影响了我们的服务。当时正是混乱猴子终止了一个由于部署失误而没有冗余的服务节点造成了问题。幸运的是，这个故障发生在白天工作时间，在这个故障的服务刚刚部署不久后，对用户的影响也非常小。

混乱猴子的美妙之处就在于此，它能尽可能地将服务节点失效的痛苦提到最前，同时让所有工程师在构建一个具有足够弹性应对失败的系统上，达成一个一致的目标。

# 混沌工程原则

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。
- ◆ 任何能够破坏静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。

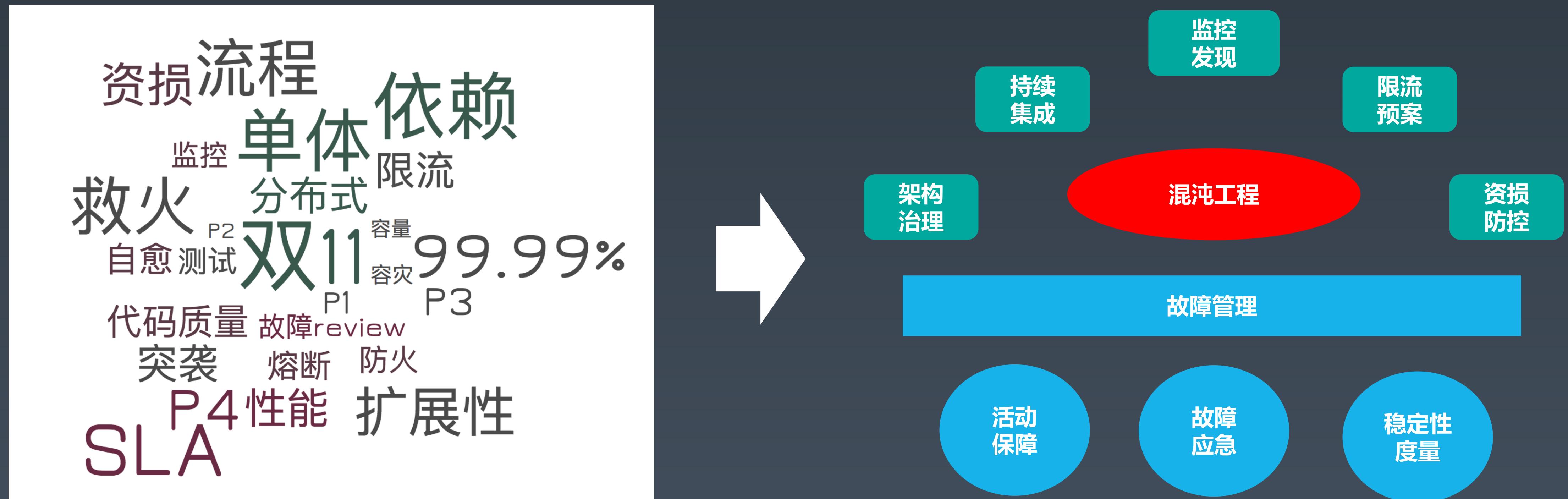
持续自动化运行实验

- ◆ 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。

# 从"故障驱动"到"故障"驱动



# 一个高速发展的技术领域



# 小结

## 混沌工程

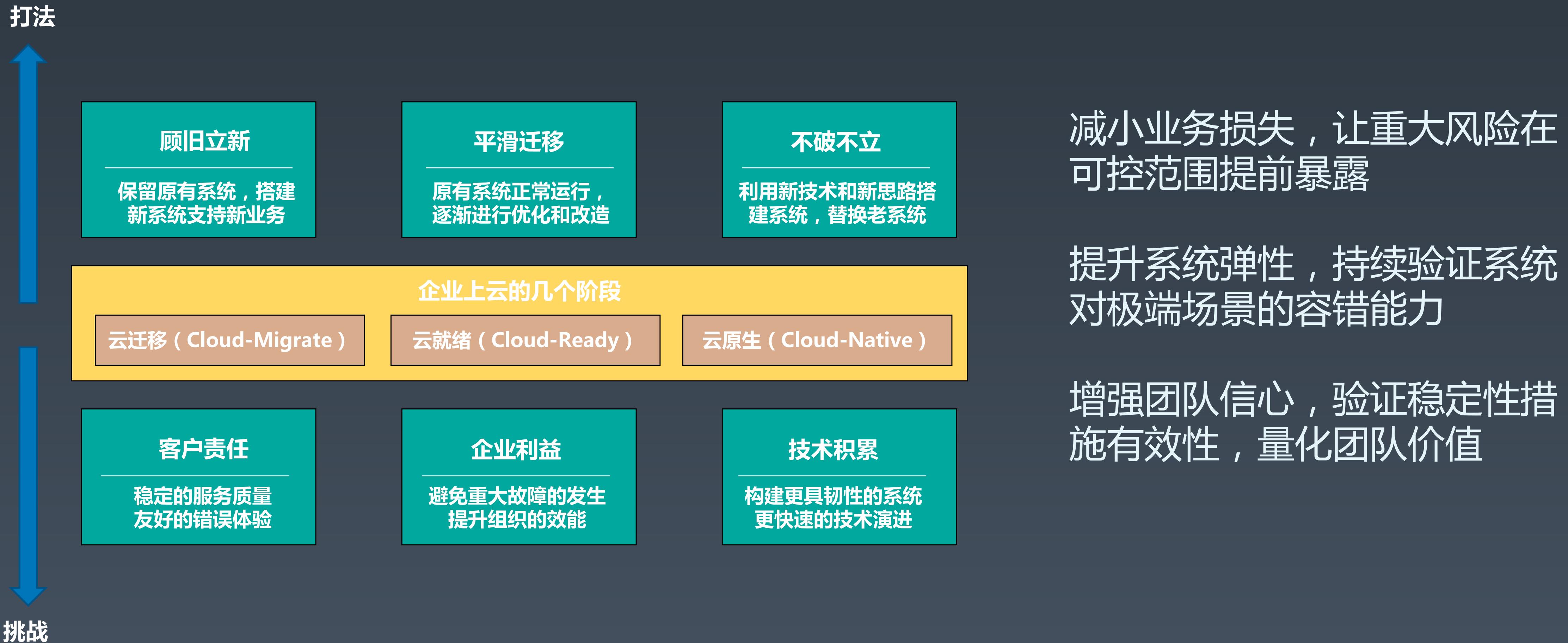
作为一个蓬勃发展的技术领域，

体现一种反脆弱的技术思想，

通过一套严谨的实践原则，

帮助企业更主动的提升稳定性

# 为什么要实施混沌工程



# 企业如何开始实施混沌工程？

- 混沌工程的引入 ( $0 \rightarrow 1$ )

结合技术架构，选择实验工具

最小爆炸半径，控制实验风险

- 混沌工程的推广 ( $1 \rightarrow N$ )

建立面向失败设计的技术文化

围绕战略制定目标，围绕目标设计组织

复用成熟产品，提升效能

# 一款好的实验工具需要满足哪些条件？

**CNCF Cloud Native Interactive Landscape**

The Cloud Native Trail Map ([png](#), [pdf](#)) is CNCF's recommended path through the cloud native landscape. The cloud native landscape ([png](#), [pdf](#)) and serverless landscape ([png](#), [pdf](#)) are dynamically generated below. Please [open](#) a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2019-05-03 19:53:13Z

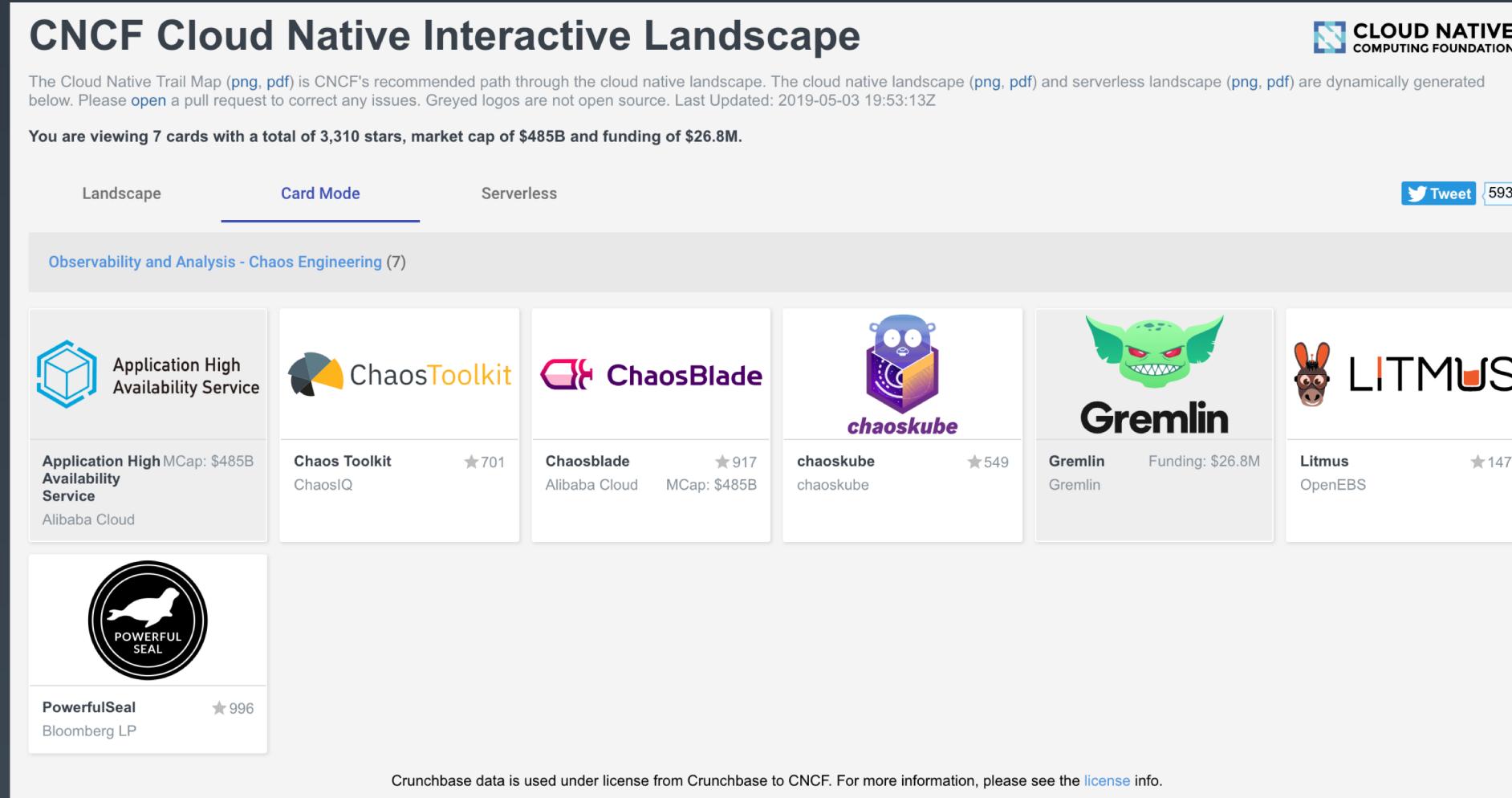
You are viewing 7 cards with a total of 3,310 stars, market cap of \$485B and funding of \$26.8M.

Landscape    Card Mode    Serverless    [Tweet](#) { 593 }

Observability and Analysis - Chaos Engineering (7)

 Application High Availability Service Application High MCap: \$485B Availability Service Alibaba Cloud	 Chaos Toolkit ChaosIQ ★ 701	 Chaosblade Alibaba Cloud ★ 917 MCap: \$485B	 chaoskube chaoskube ★ 549	 Gremlin Gremlin Funding: \$26.8M chaoskube	 LITMUS Litmus OpenEBS ★ 147
 PowerfulSeal ★ 996 Bloomberg LP					

Crunchbase data is used under license from Crunchbase to CNCF. For more information, please see the [license info](#).

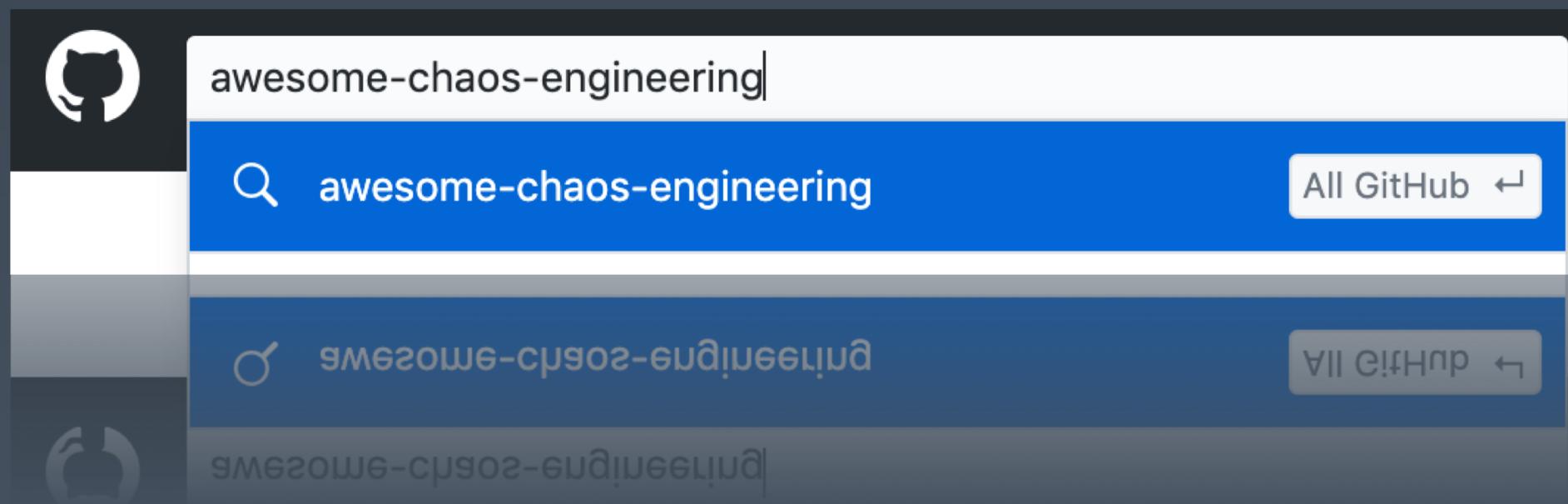


awesome-chaos-engineering

All GitHub ↴

  awesome-chaos-engineering

  awesome-chaos-engineering



- **丰富度**  
资源、主机、容器、应用 .....
- **易用性**  
开发框架、实验工具、产品平台
- **开放程度**  
闭源、OpenAPI、支持扩展、开源
- **集成方式**  
代码依赖、架构依赖、无依赖
- **多语言**  
Java、Go、C++、语言无关 .....
- **活跃状态**  
已停滞、维护、活跃

# 阿里混沌工程的技术演进路线

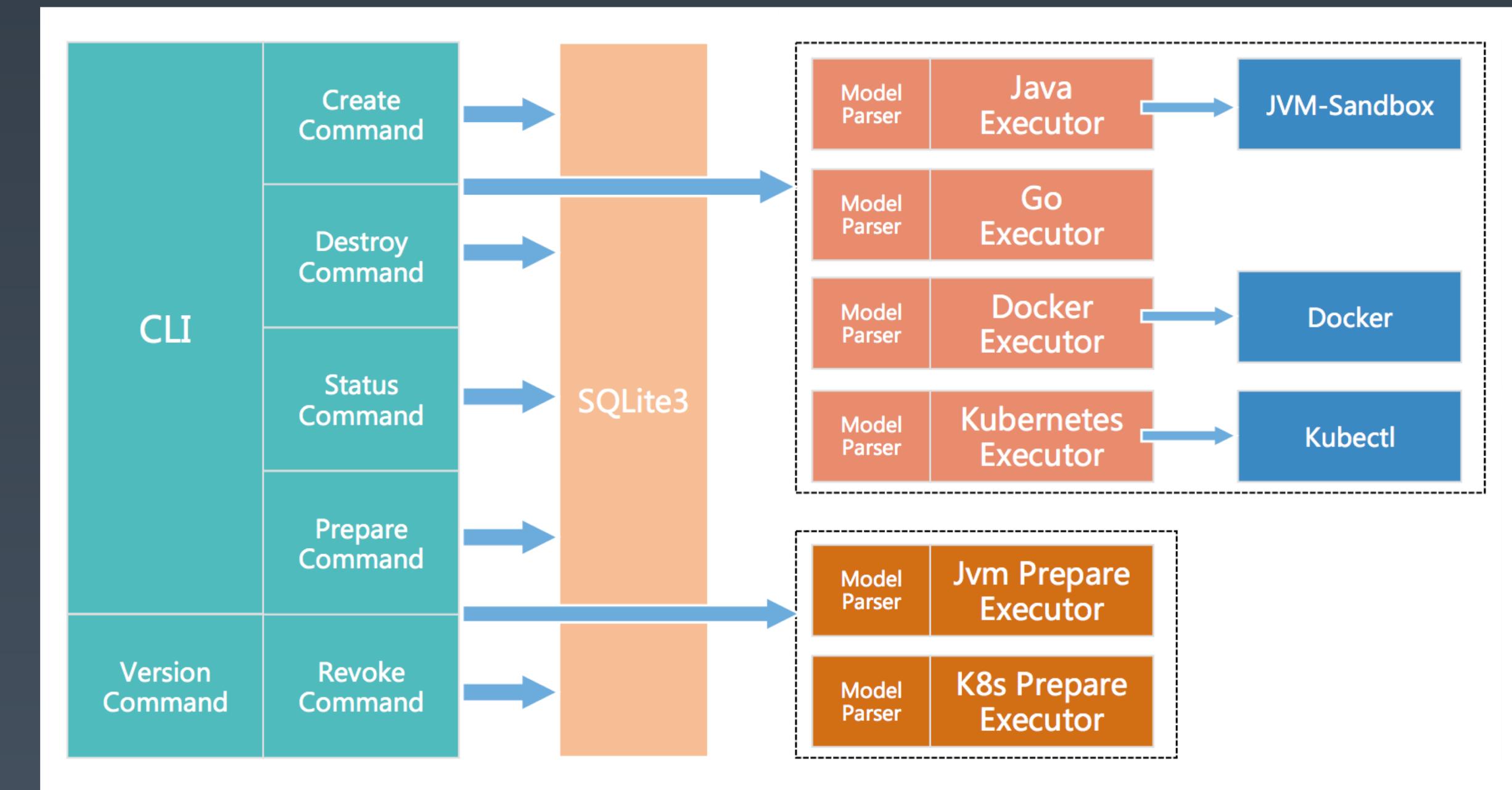
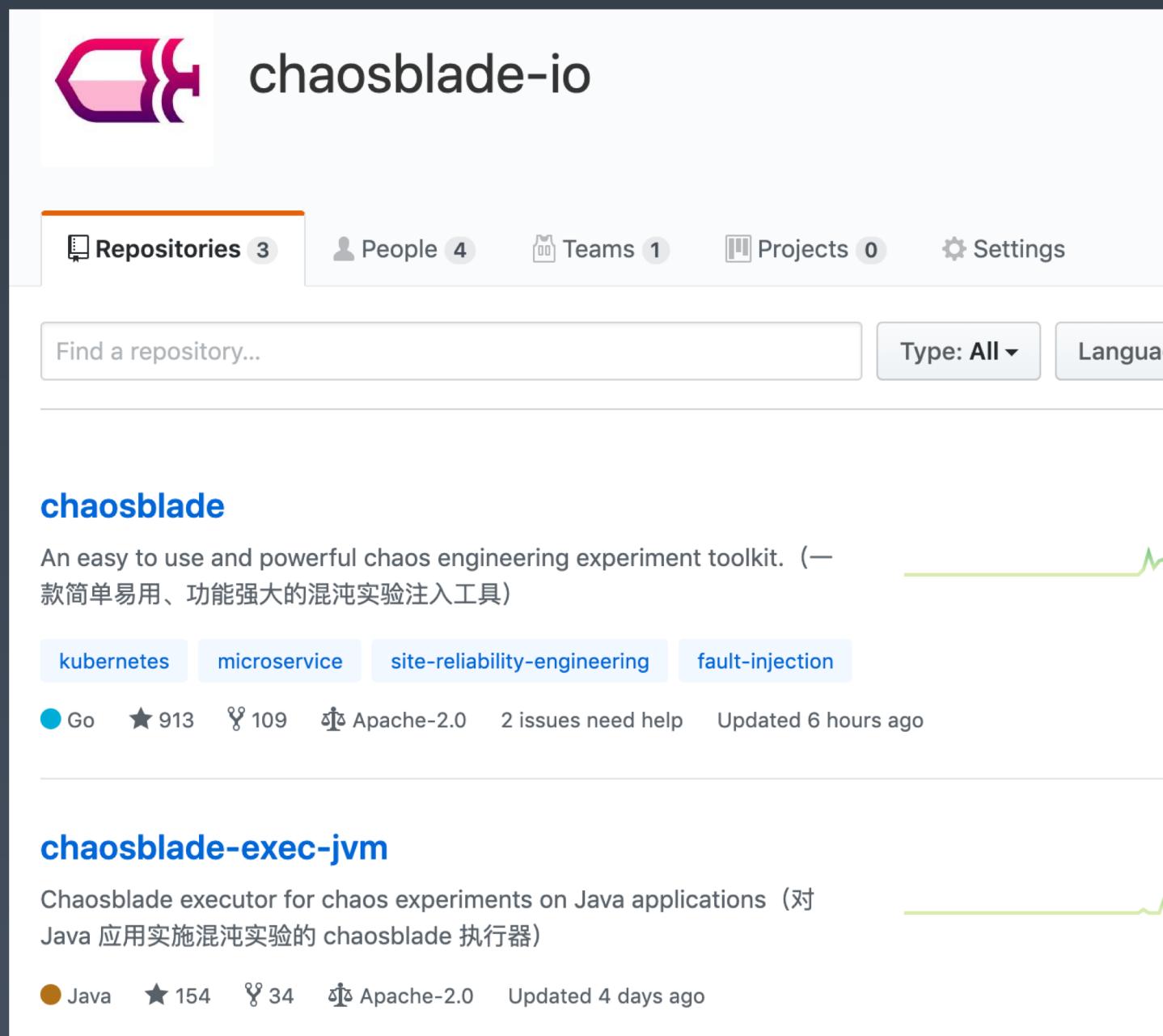


延伸阅读：[《阿里电商故障治理和故障演练实践》](#)

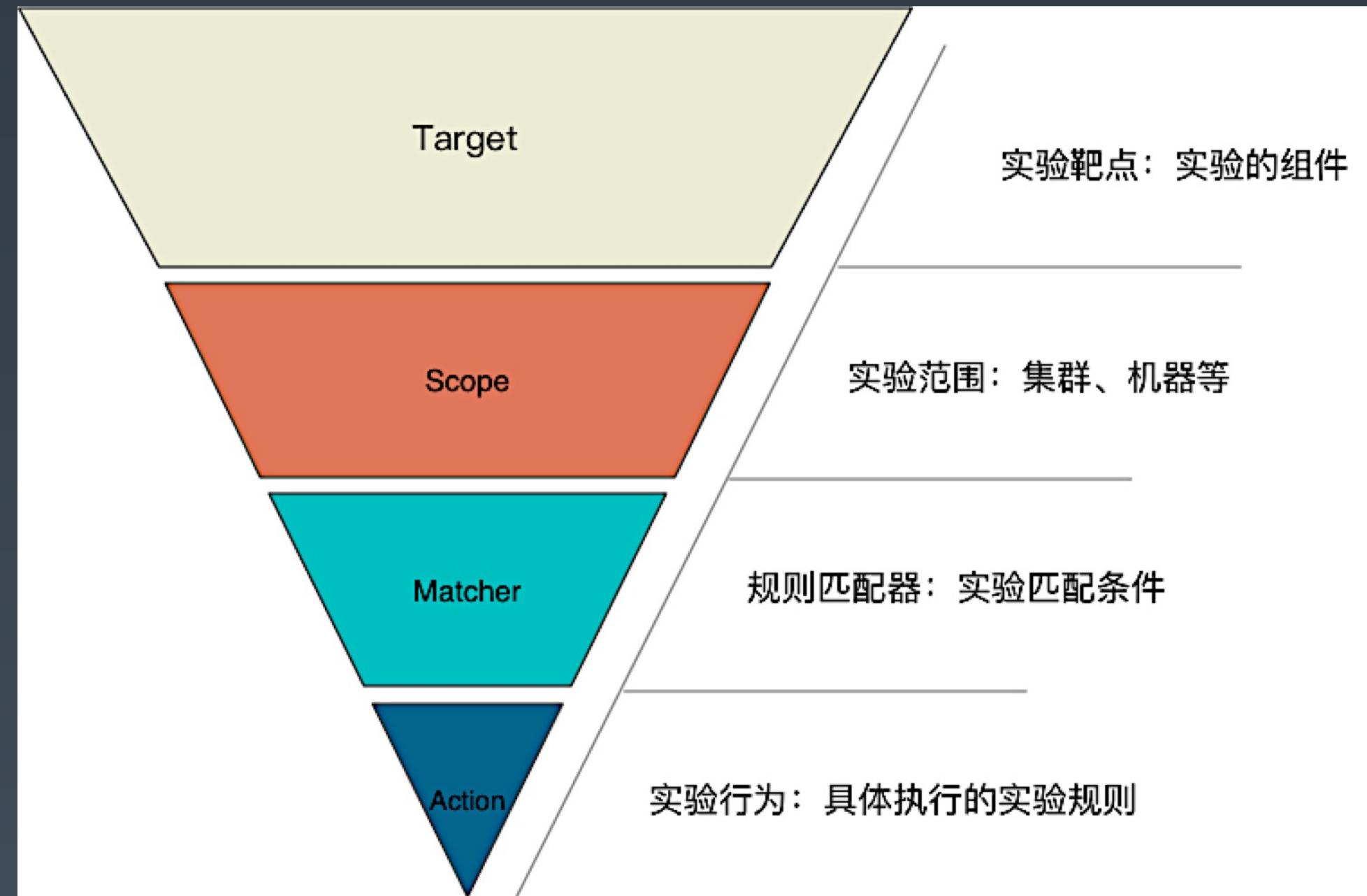
# 开源工具——混沌之刃（ChaosBlade）

ChaosBlade是一款遵循混沌实验模型，提供丰富故障场景实现，帮助分布式系统提升容错性和可恢复性的混沌工程工具，它的特点是操作简洁、无侵入、扩展性强。

GitHub 地址：<https://github.com/chaosblade-io>



# 统一实验模型，描述混沌事件



## 实验示例：

```
blade create dubbo delay --time 3000  
--consumer  
--service com.example.HelloService  
--version 1.0.0
```

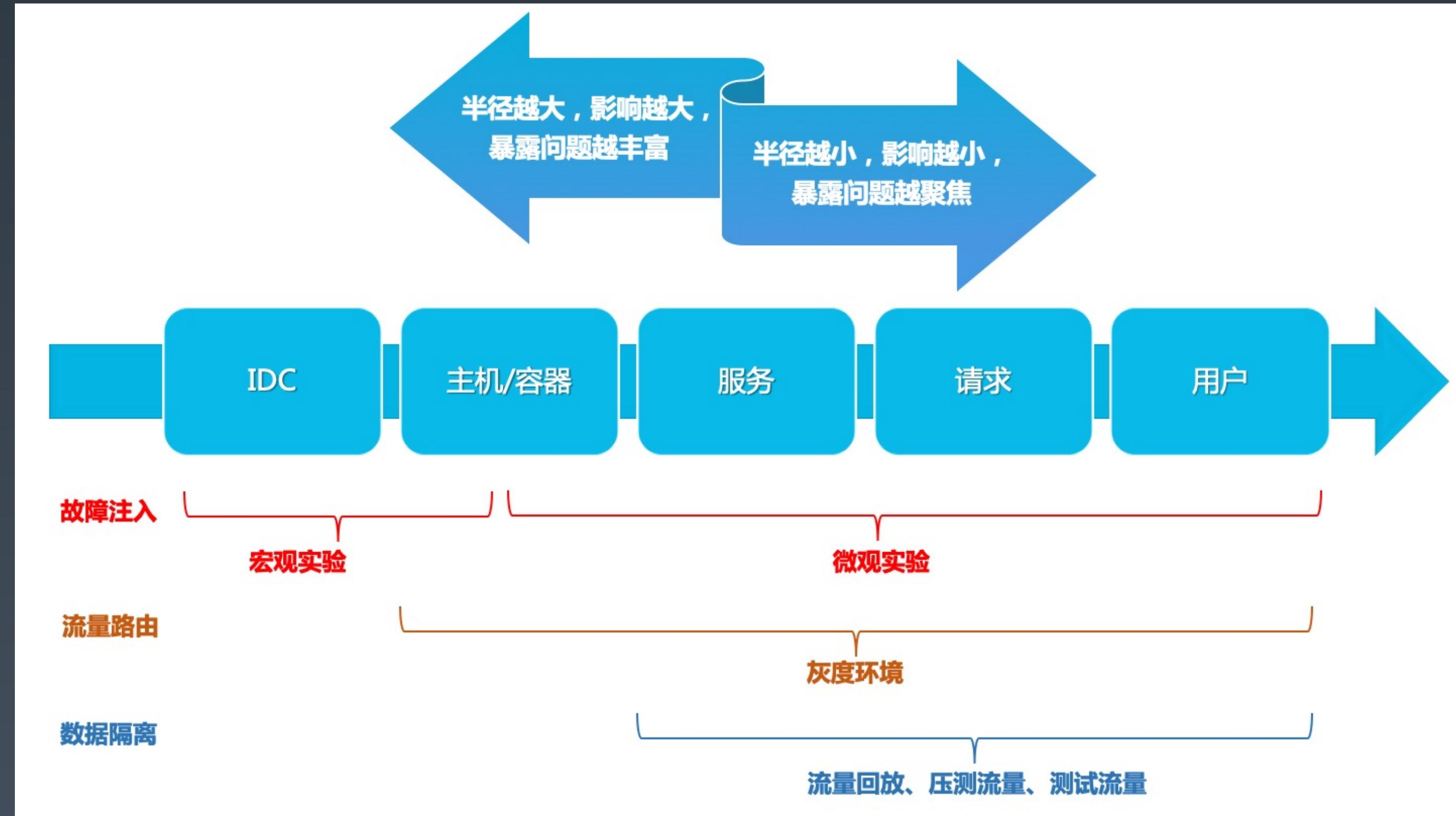
```
blade create cpu fullload --cpu-count 4
```

## 结束示例：

```
blade destroy 6435335635bbaca5 (实验ID)
```

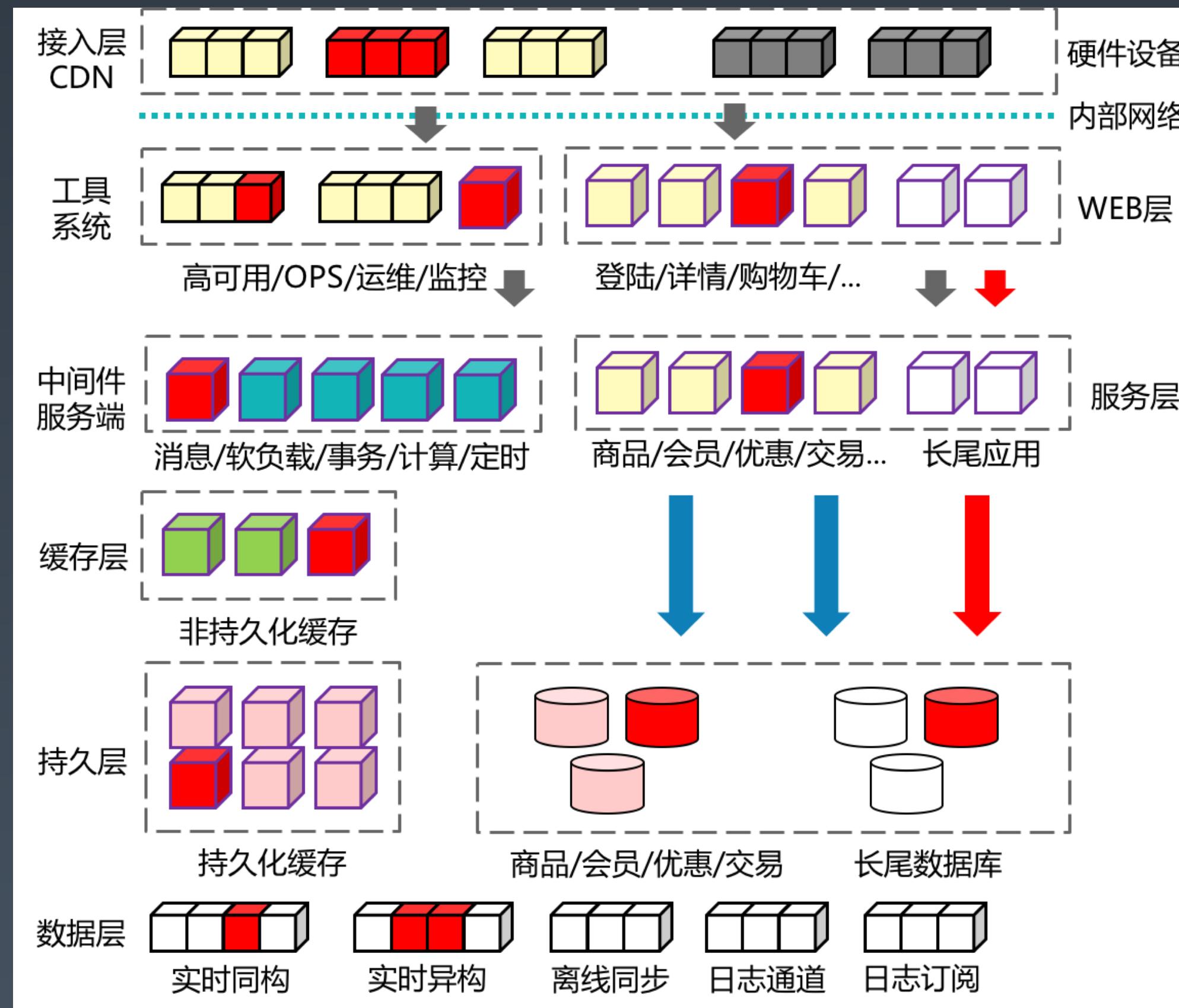
```
{  
  "code": 200,  
  "success": true,  
  "result": "command: cpu fullload --cpu-  
count 4 --debug false --help false"  
}
```

# 控制爆炸半径，减小实施风险

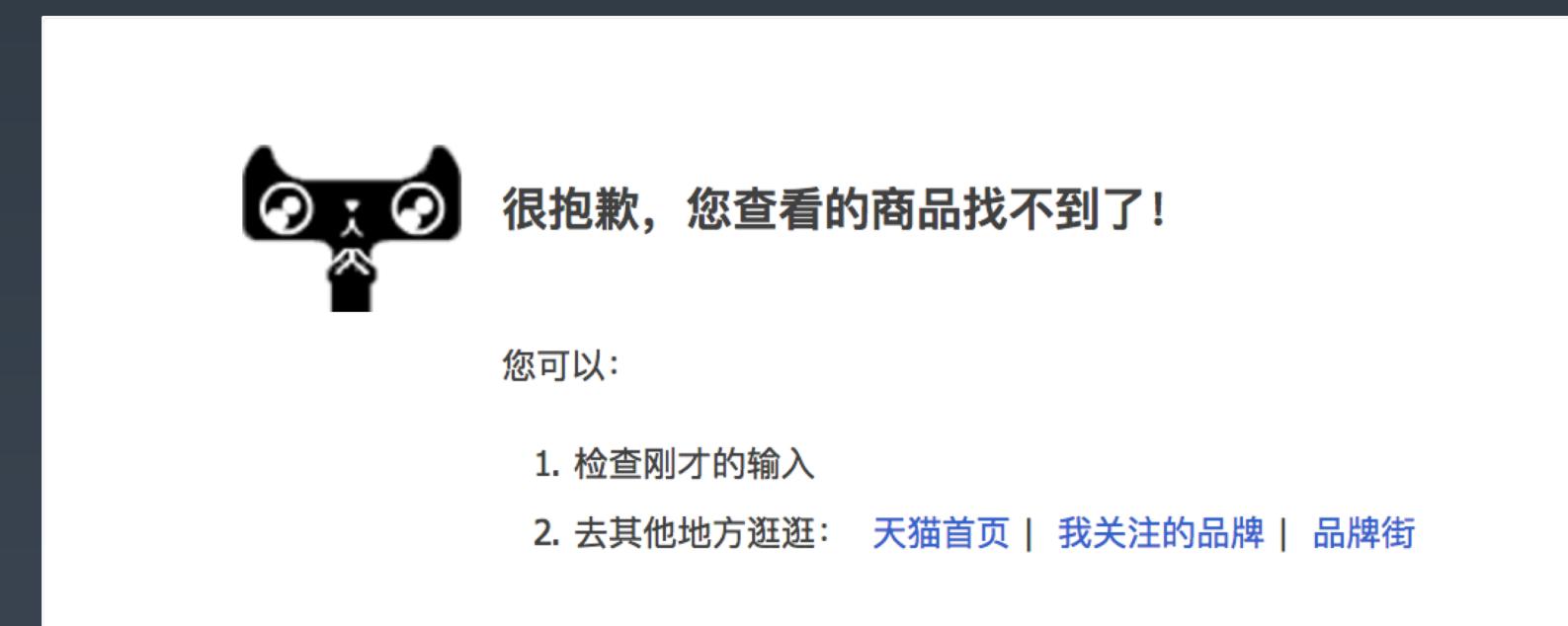


# 建立面向失败设计的技术文化

面向失败设计，因为每一块硬盘，每一个业务系统，每一种技术组件都有出错的可能！



分布式系统需要制定分级策略，防止非核心业务拖垮核心业务！！



工具系统需要优先实现容灾！！！

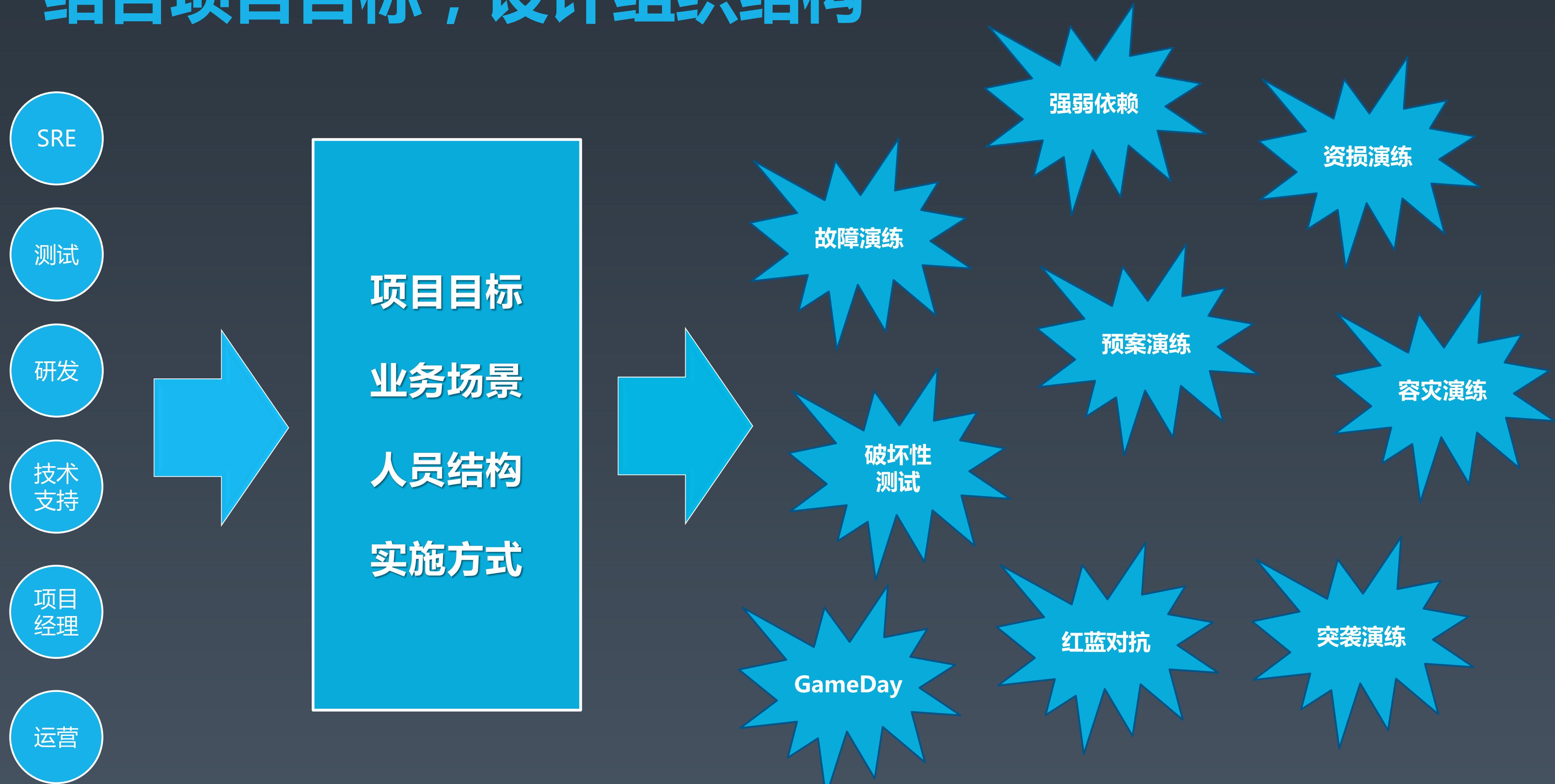


故障处理流程和人员能力也非常重要！！！！

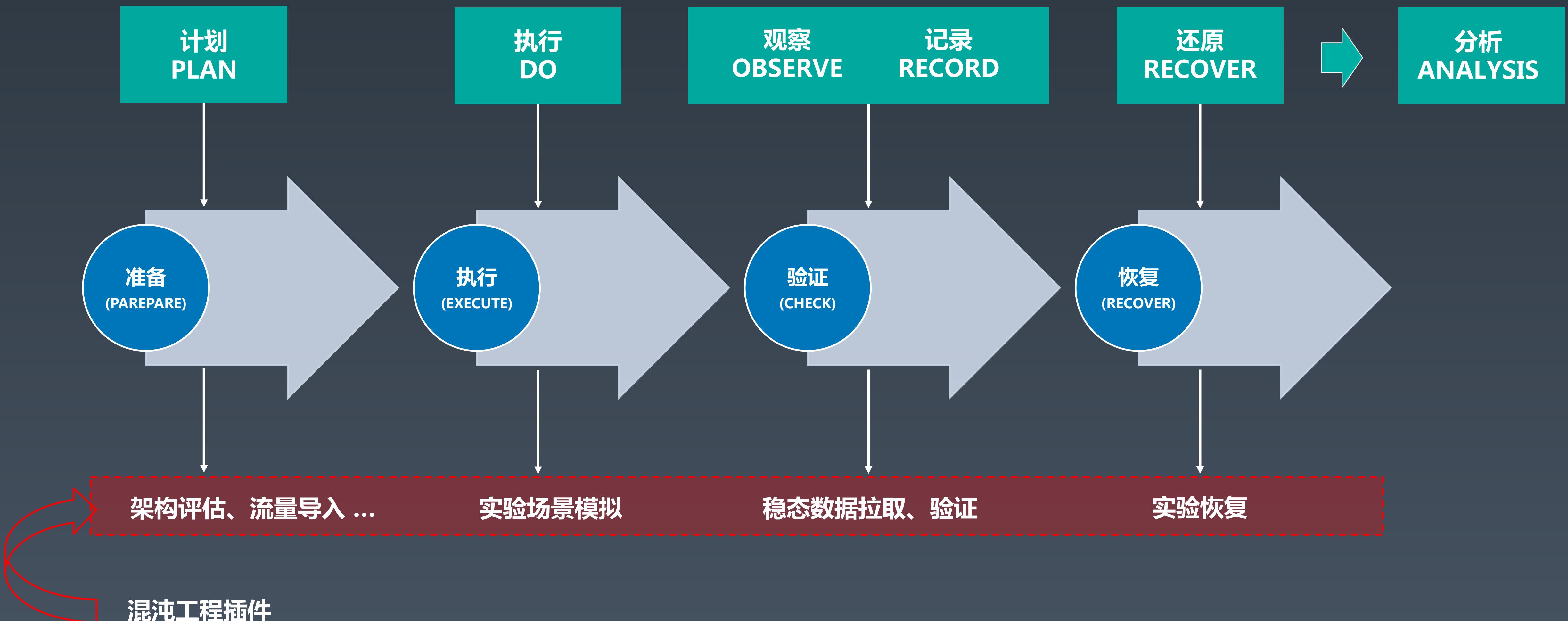
# 围绕企业战略制定项目目标



# 结合项目目标，设计组织结构



# 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



# 小结

引入和推广混沌工程，您需要

结合企业特点，选择适合当下的工具或产品；

最小爆炸半径，控制实验风险；

建立面向失败的技术文化，接受不确定性；

围绕企业战略，有针对性的设计组织和实施；

建设实验平台，提升规模化能力；

# 落地场景举例



新零售



云服务



云业务

# 新零售业务稳定性的挑战

## 挑战

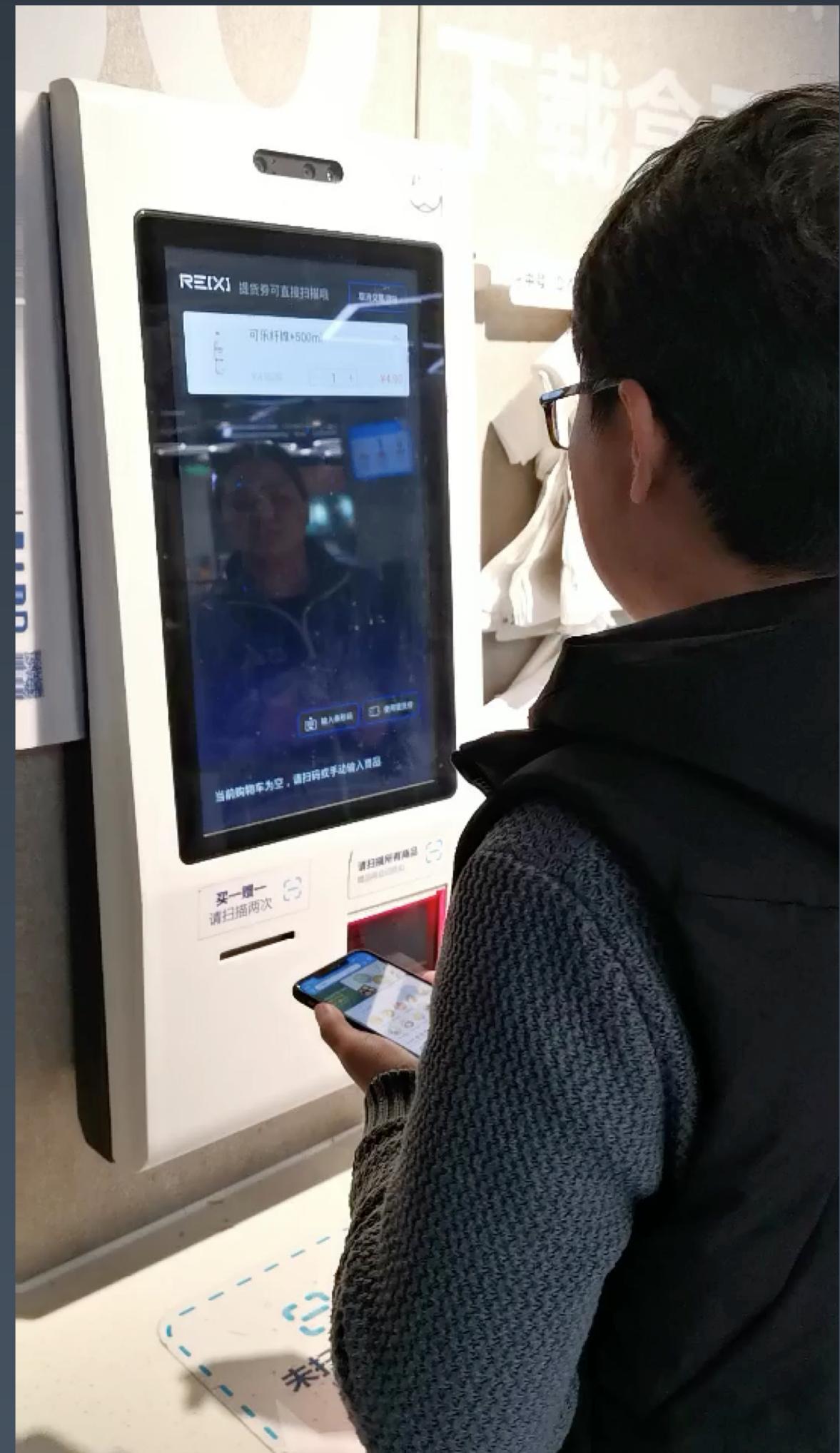
- 线下场景，用户对故障容忍度低
- 无法彻底规避网络问题，尤其是门店网络

## 要求

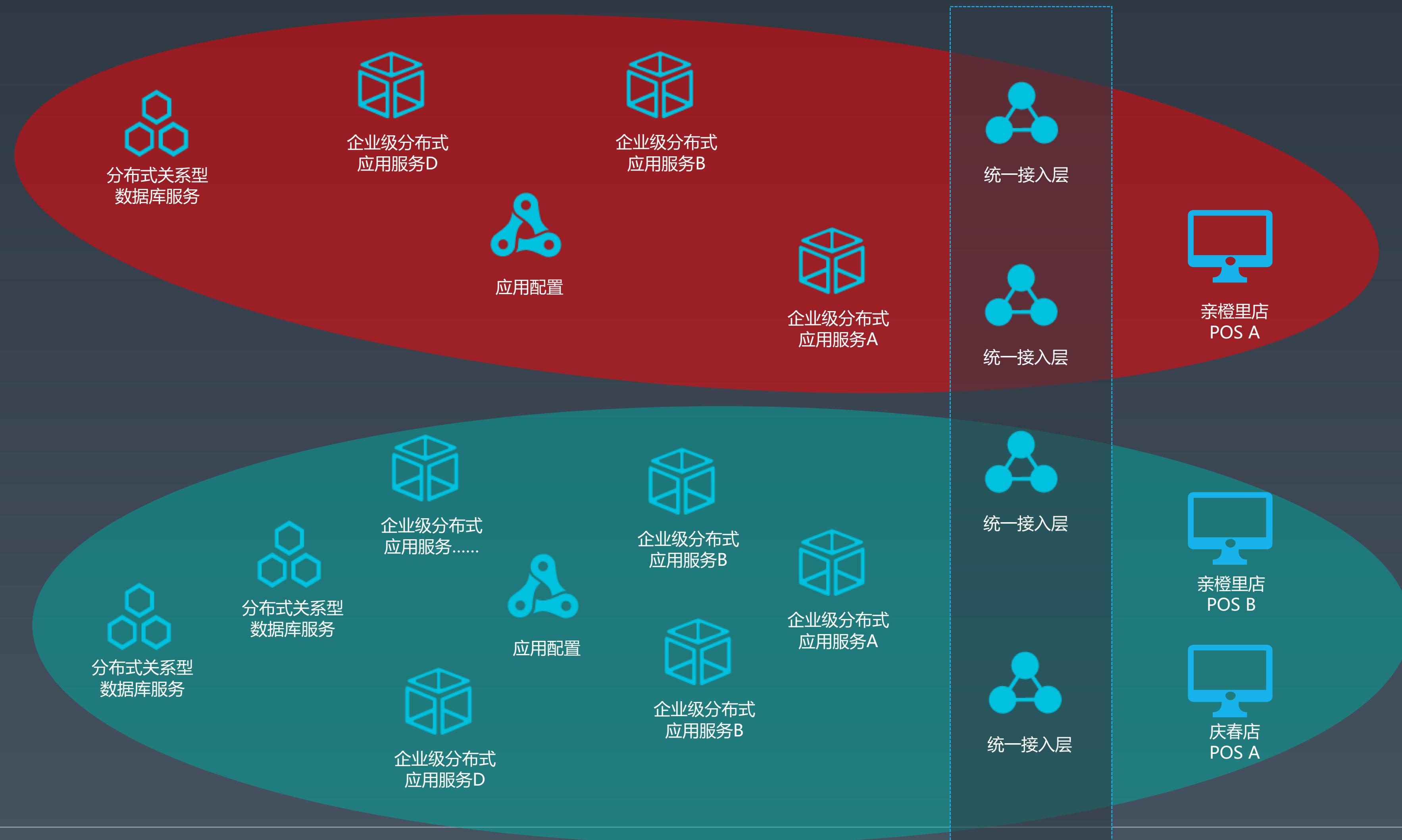
- 云端服务要具备较高的可用性
- 终端的异常提示要面向用户友好
- 现场人员要熟悉处理手段

## 难点

- 如何证明稳定性措施有效性？
- 如何减少实验对业务带来的影响？
- 如何常态化的实施实验？



# 最小化爆炸半径，实现常态化的实验



# 云服务稳定性——专有云混沌工程实践

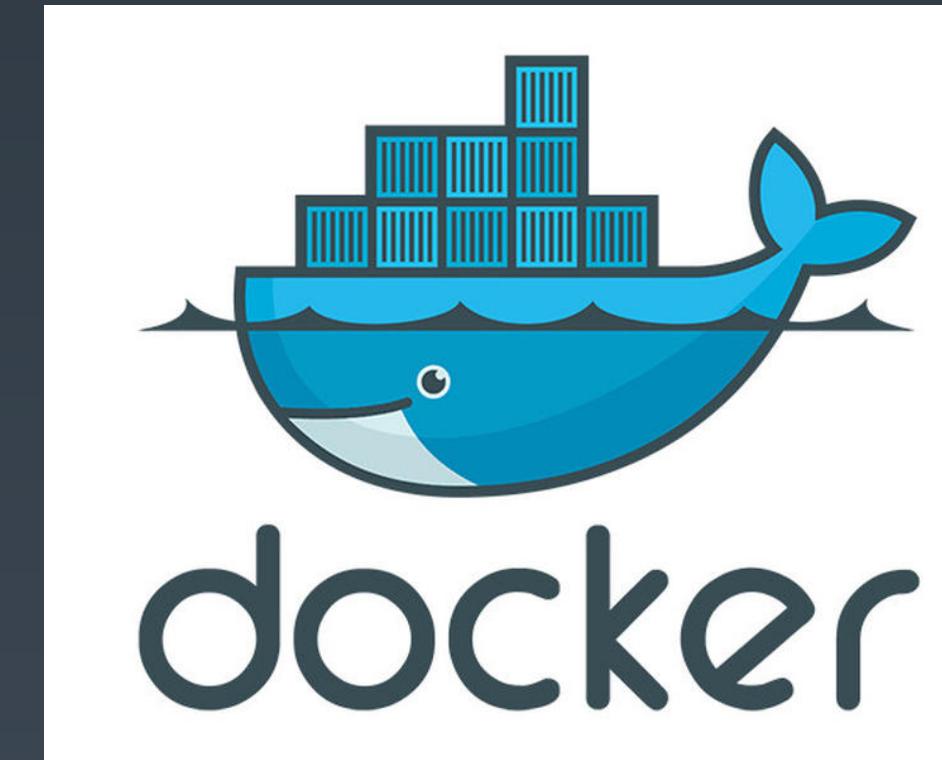


# 云业务稳定性保障的挑战

微服务



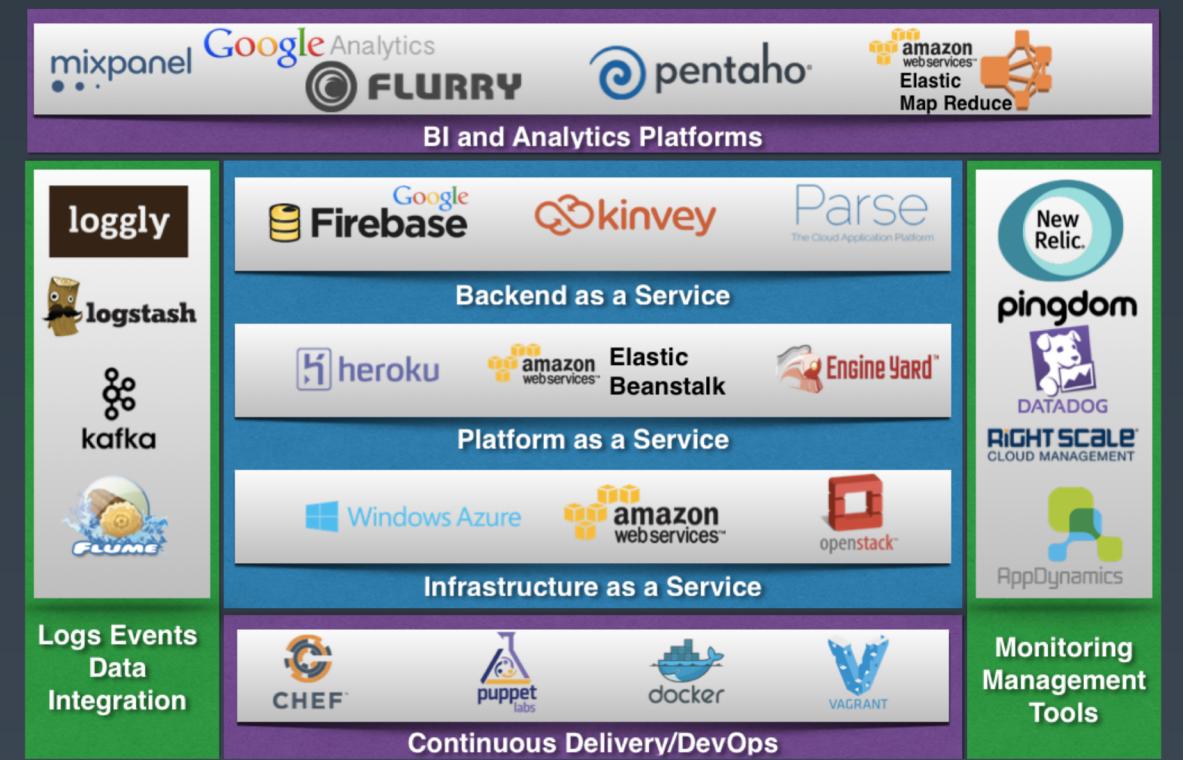
容器



开源组件



云服务



有什么

有哪些第三方组件，用了哪些云服务，有哪些自开发的应用服务，他们之间的关系是什么样的，他们和底层的容器，云服务器的关系是什么样子的？

做什么

我的第三方组件，云服务和应用服务需要具备哪些高可用能力

怎么做

如何提高这些组件的高可用能力

# 面向云业务的高可用服务

## 应用高可用服务

应用高可用服务（Application High Availability Service）是一款专注于提高应用高可用能力的SaaS产品，提供应用架构自动探测，故障注入式高可用能力评测和一键流控降级等功能，可以快速低成本的提升应用可用性。

[免费开通](#)[产品文档](#)

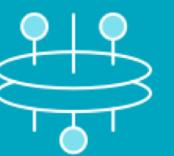
### 自动感知架构

自动探测应用的架构组件和依赖关系，构建架构拓扑并持续跟踪变化



### 智能识别组件

覆盖主流的三方组件和大部分阿里云服务，通过AI不断学习架构特征



### 丰富的故障演练场景

来自线上真实的故障类型和演练模板，全方位评测应用的高可用能力

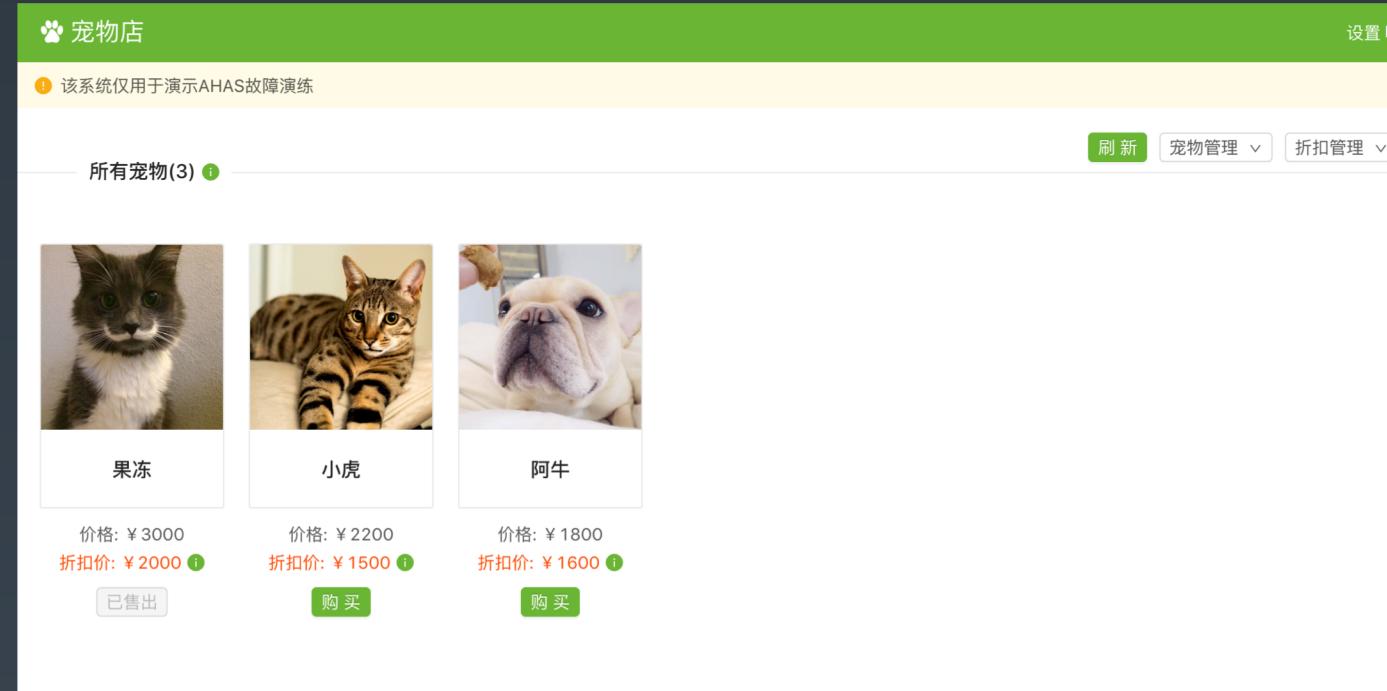


### 成熟的限流降级能力

历经双十一等技术大考的流控降级保护等防护手段，确保应用万无一失

阿里云菜单搜索『AHAS』，免费公测中

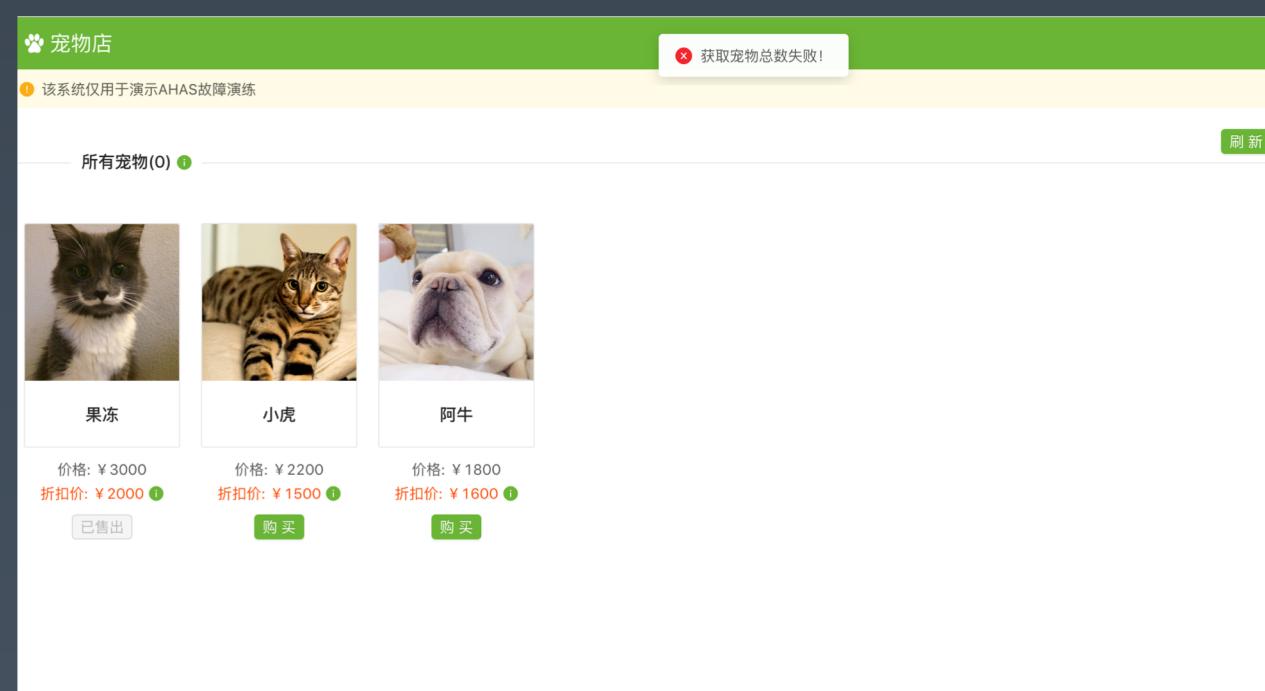
# 云业务混沌实验方案



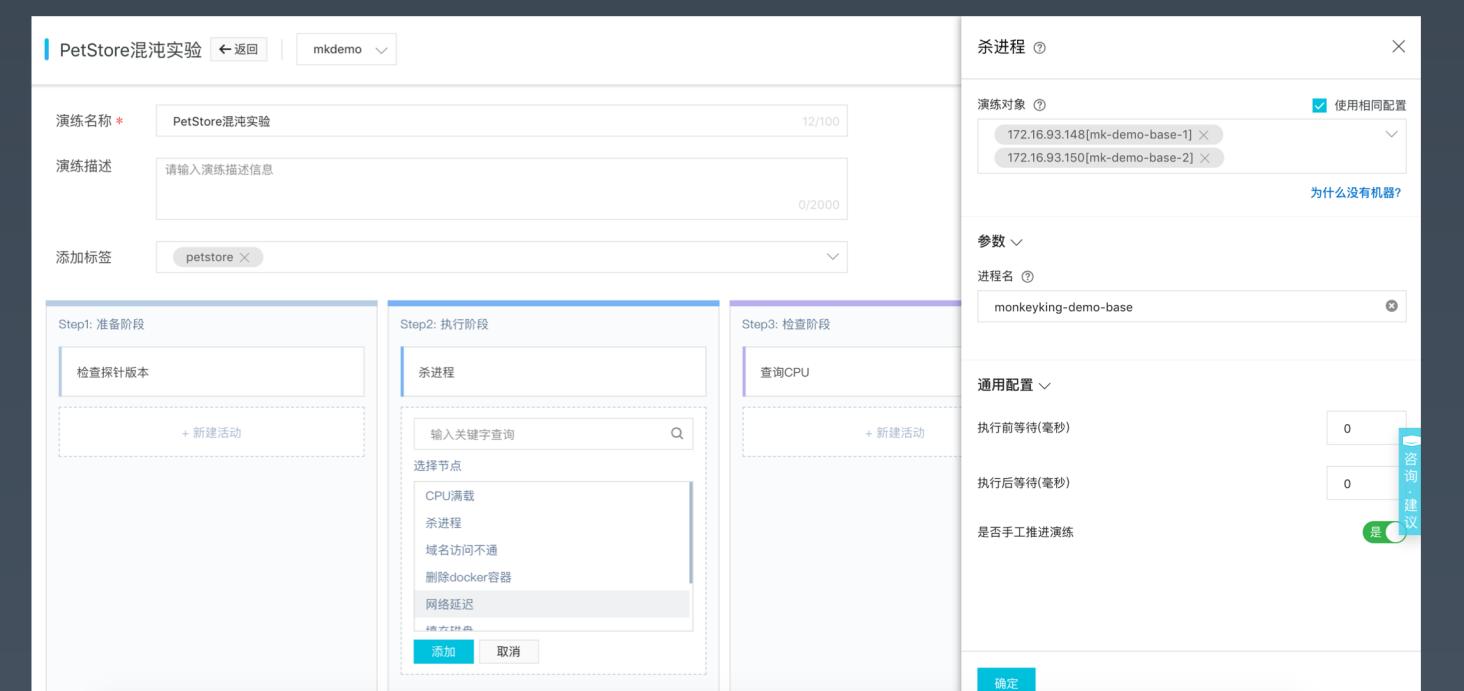
云原生业务（稳态分析）



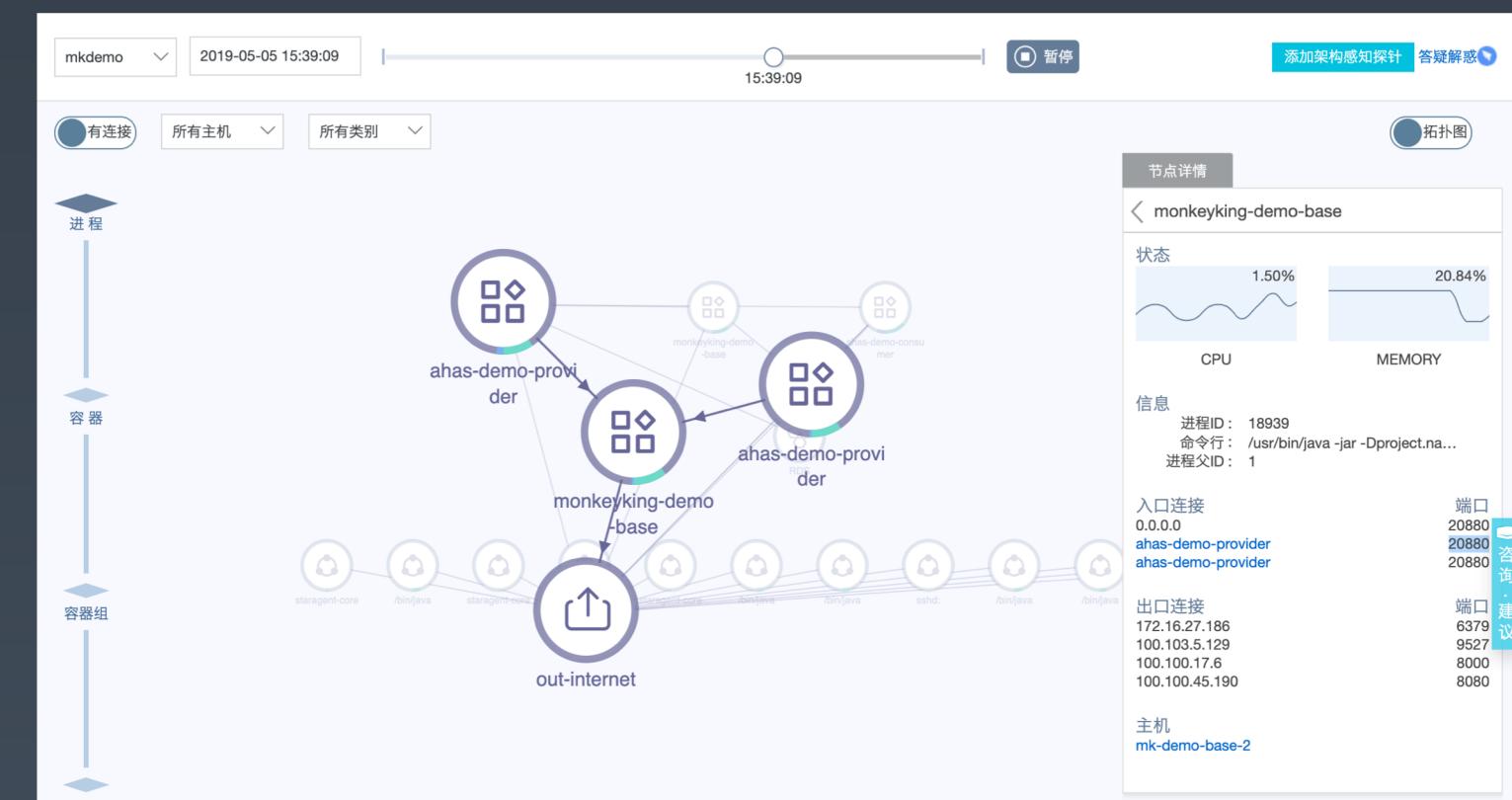
架构感知&组件识别



云原生业务（稳态验证）



白屏化式实验



架构组件分析

# 未来规划

方案

帮助云原生业务提升高可用能力的云服务 ( AHAS )

特性

架构感知

故障演练

限流降级

高可用分析

连接

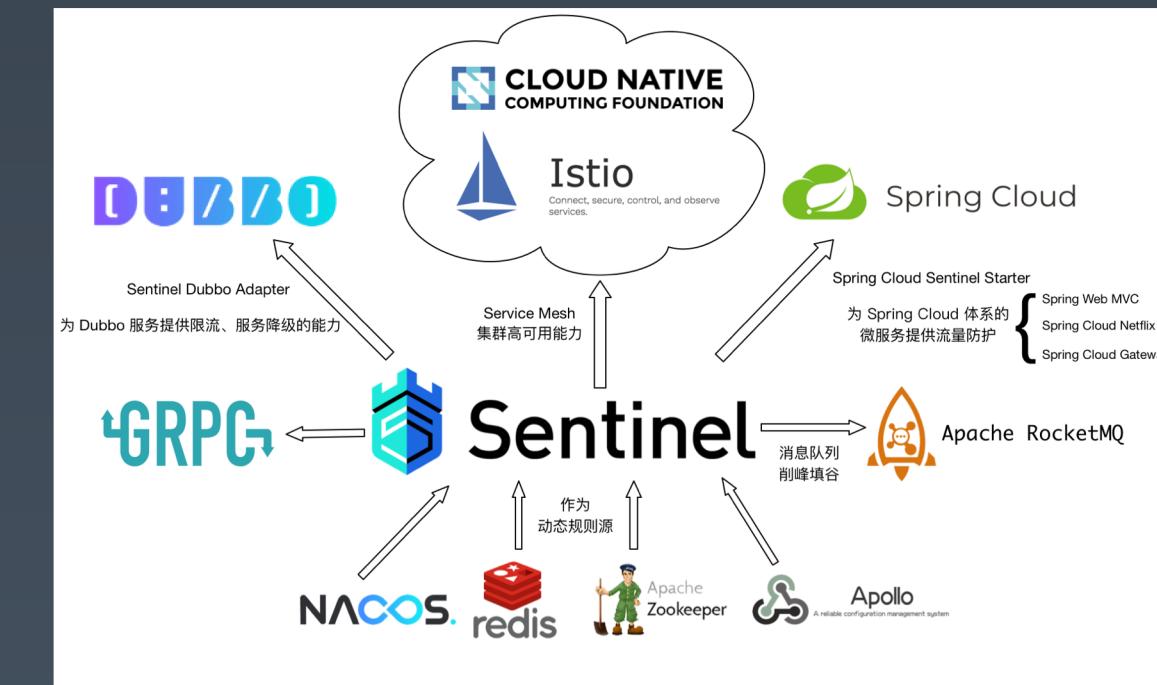
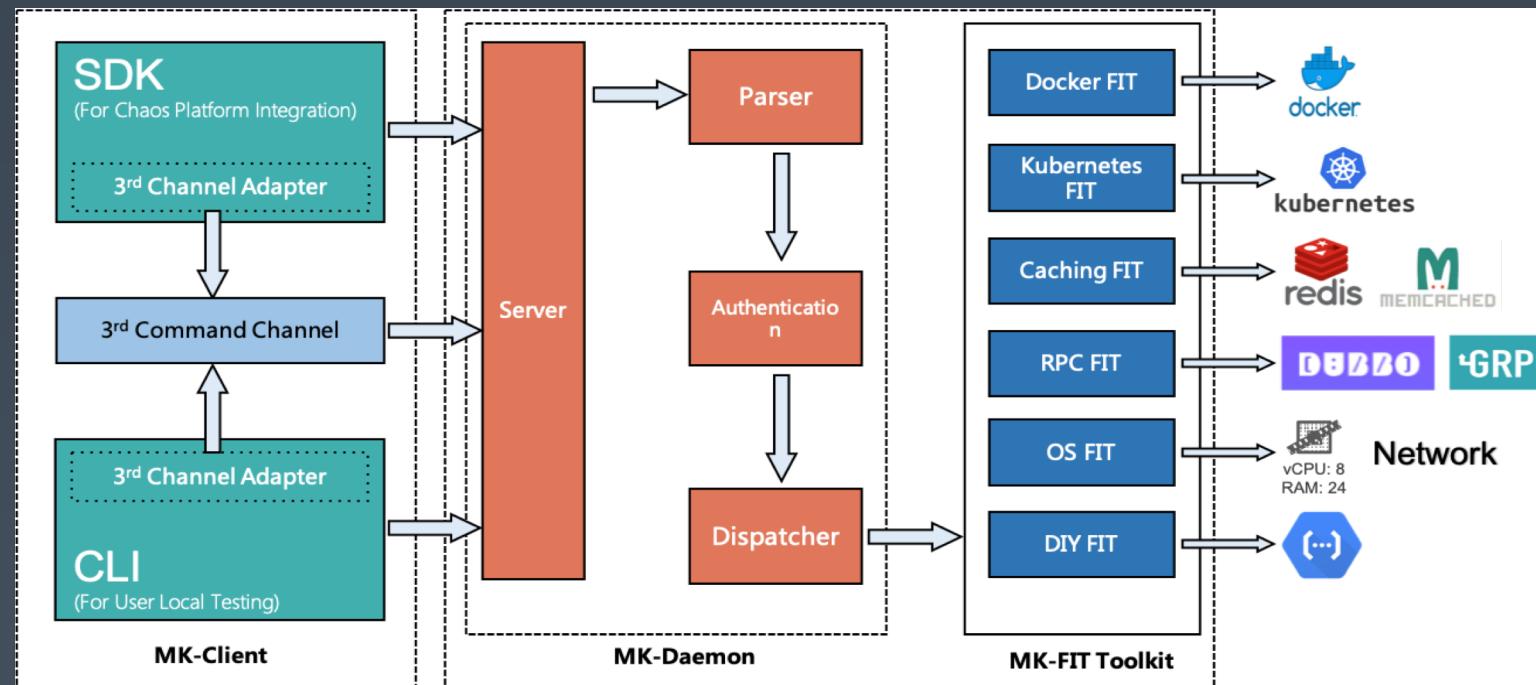
容器服务

日志服务

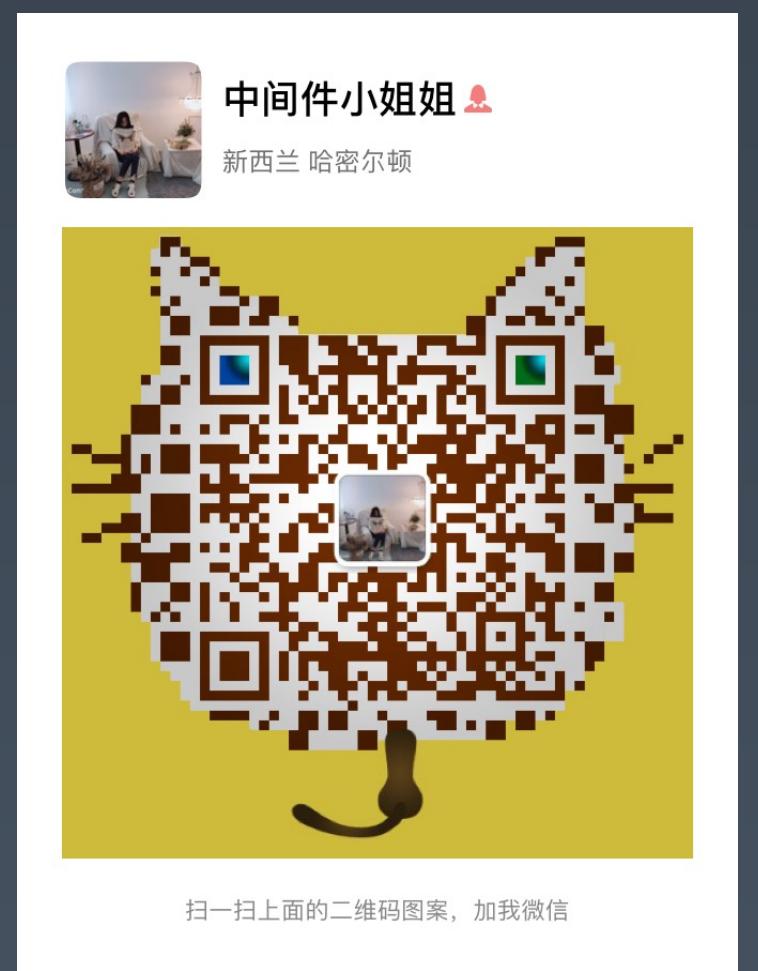
云监控

其他...

开源



社区  
建设



口令 : chaos

THANKS! | QCon IC<sup>th</sup>