阿里云
奥运会官方云服务合作伙伴

# 应用高可用解决方案

## 阿里巴巴应用高可用保障最佳实践

# 什么是应用高可用

阿里云
奥运会官方云服务合作伙伴

**无论什么原因应用都不能停止服务**

# 故障不可避免，风险管理常备

突发性热点事件，导致“微博崩了”

网友吐槽：一官宣就宕机

bilibili

7月13日晚间，哔哩哔哩（简称“B站”）突发服务器宕机事故，主站、App 以及小程序都无法使用

facebook

10月5日晚间，Facebook、Instagram和WhatsApp的网站和App出现集体宕机，并持续超过6个小时

富途证券 WWW.FUTU5.COM

10月9日凌晨，富途账户出现了登录后无法交易，出现交易中断，甚至导致资产清零。事故持续2个小时左右。

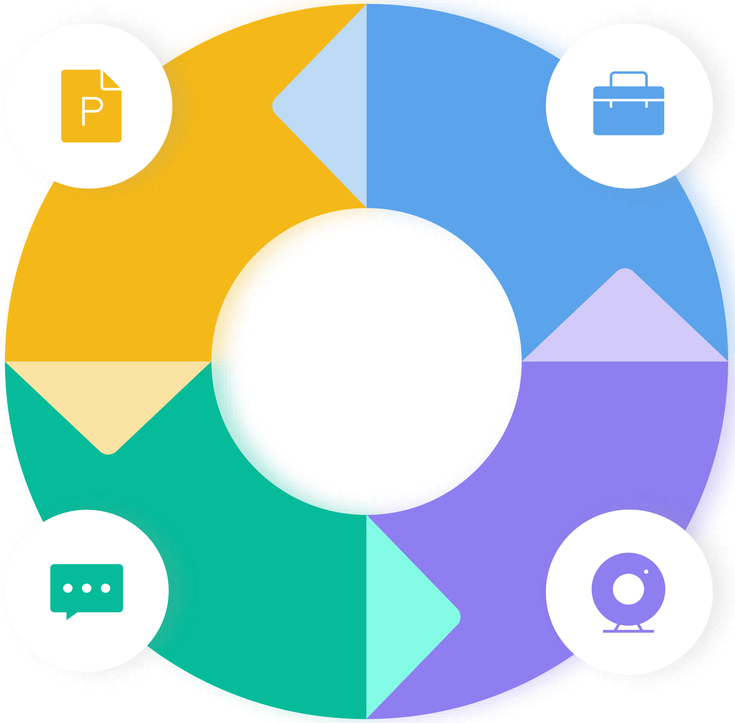
**高可用、稳定性压倒一切**

# 影响因素很多，保障手段各异

# 应用稳定性生命周期

## 风险控制

- 限流降级
- 开关预案
- 容灾多活
- 常态化压测
- 常态化故障演练
- ...



## 流程规范

- 需求评审
- 设计评审
- 架构评审
- 发布管理
- ...

## 风险感知

- 监控
- 报警
- 反馈
- ...

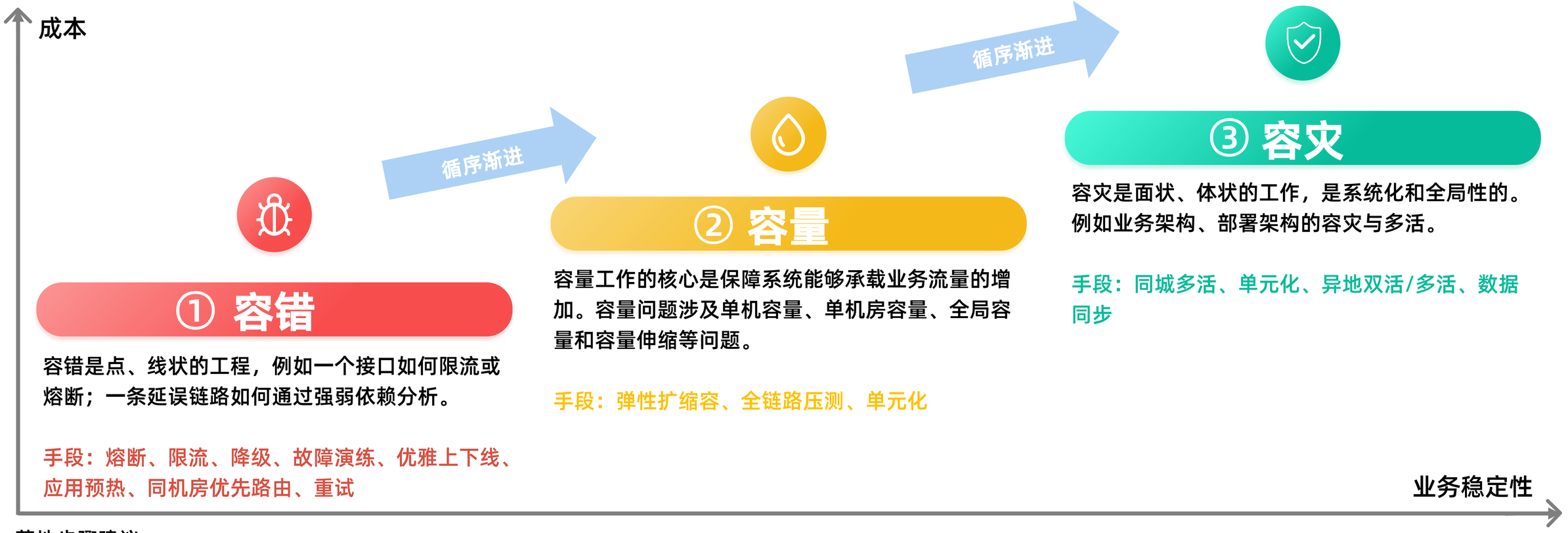
## 代码质量

- 单元测试
- 冒烟测试
- Code Review
- 静态代码扫描
- ...

# 高可用架构演进思路

*整体思路：*

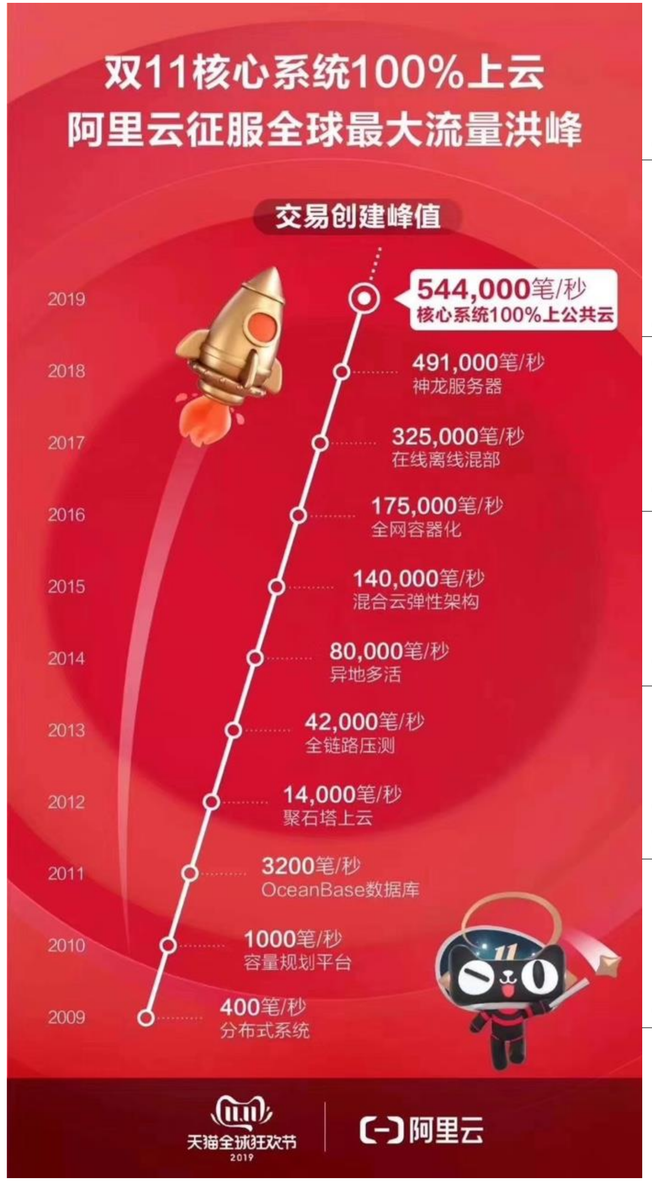
*业务高可用架构建设是以面向失败设计为核心思想，以云原生为主要技术路线，需要应用研发和平台运维等共同参与，按照业务维度，以**容错、容量、容灾**为一般落地顺序，演进式的全局优化工作。*




**落地步骤建议：**

1. 错是小灾，灾是大错。如果系统容错能力不足，小的抖动会导致大的故障，一个点的问题会蔓延至整个链路甚至整个系统。因此，高可用建设一般先从容错优化、从应用的自我保护做起。以业务为维度逐步试点，逐步改造，逐步演练。
2. 容量和容灾是全局系统性工程，需要业务研发、架构、运维的共同参与。整体项目实施周期较长，通常以月计算，从目前经验来看，异地双活的容灾项目周期是同等规模和业务复杂度的同城项目周期的两倍以上，但是带来的稳定性效益也更高。

# 阿里高可用架构演进历史




2020年~，全面推行“集团-商业-开源”三位一体

2018年，AHAS应用高可用服务商业化
2019年，第一个大型异地多活项目落地

2017年，PTS商业化

2015年，千里之外的3地四单元的架构，标志异地多活成为标准

2013年，一系列稳定性产品，限流降级，开关预案，故障演练诞生，稳定性体系雏形出现

2012年，全链路压测出现，阿里的双十一因此而不一样

# 阿里云原生应用高可用架构全景图

--- 阿里巴巴集团高可用保障体系的最佳实践


**本期分享：**

故障演练

流量防护

多活容灾

# 覆盖“事前预防-事中防控-事后恢复”的高可用架构

**高阶能力**

管控大屏 | 性能认证 | 测评报告 | 业务监控 | 智能压测 | 自动巡检 | 自动防护 | 人员协同 | 系统恢复

**场景应用**

| 故障演练 Chaos | 流量防护 Sentinel | 多活容灾 MSHA |
|---|---|---|
| 事前预防 | 线上管控 | 业务恢复 |
| 暴露潜在风险 | 流量洪峰 | 同城多活 |
| 衡量容错能力 | 热点防控 | 异地双活/多活 |
| 锻炼应急能力 | 不稳定依赖 | 异地应用双活 |
| 验证告警有效 | 单机过载 | 混合云/异构云容灾 |

**应用高可用产品功能**

| 故障演练 Chaos | 流量防护 Sentinel | 多活容灾 MSHA |
|---|---|---|
| 故障注入 | 入口流控 | 流量路由 |
| 基础资源演练 | 系统防护 | 应用单元化 |
| 演练编排 | 熔断降级 | 规则管理 |
| 云服务演练 | 网关防护 | 同步管理 |
| 指标监控 | 热点隔离 | 流量纠错 |
| 微服务演练 | Ingress/Nginx防护 | 数据禁写保护 |
| 架构感知 | 集群流控 | 一键切流 |
| 容灾演练 | 实时监控 | 日志监控 |

**基础能力**

面向失败设计实践

ONE-AGENT

解决确定性问题：监控设计 | 管控设计 | 冗余设计 | 容错设计

SRE体系实践

系统可视化：用云最佳实践 | 风险巡检 | 风险拓扑

基础监控：系统监控 | 业务监控

# 故障演练 Chaos
# --- 混沌工程最佳实践

# 阿里混沌工程十年演进之路

*阿里巴巴的混沌工程，从解决自身问题出发，有效支撑了阿里集团的微服务化、同城多活、异地多活、全站上云、全面云原生等技术演进，沉淀了完整的产品能力和落地实践。2018年，商业化和开源以来，累计已完成多个行业头部客户的落地。*


**2012年**

电商业务就开始尝试通过故障注入技术去解决微服务的依赖问题

**2016年**

通用故障模式沉淀为系统，通过在生产环境验证稳定性能力

**2018年**

内部多年沉淀的实践正式在阿里云商用

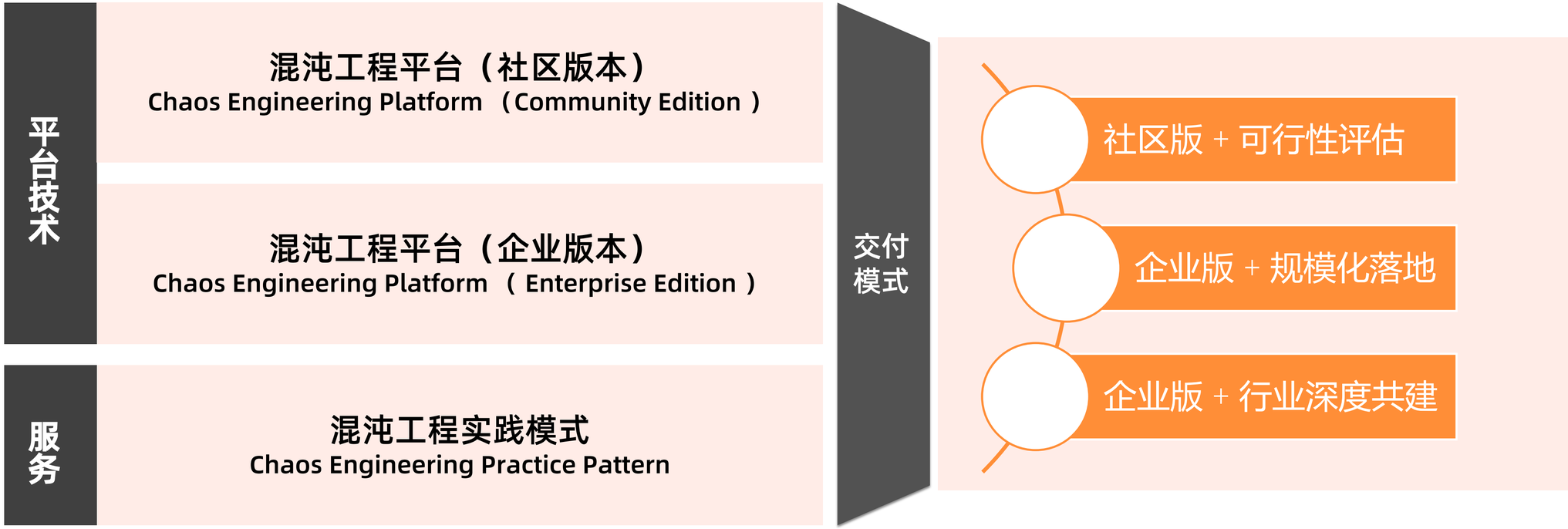
**2019年**

故障模拟能力开源，国内首个混沌工程开源产品

**2021年**

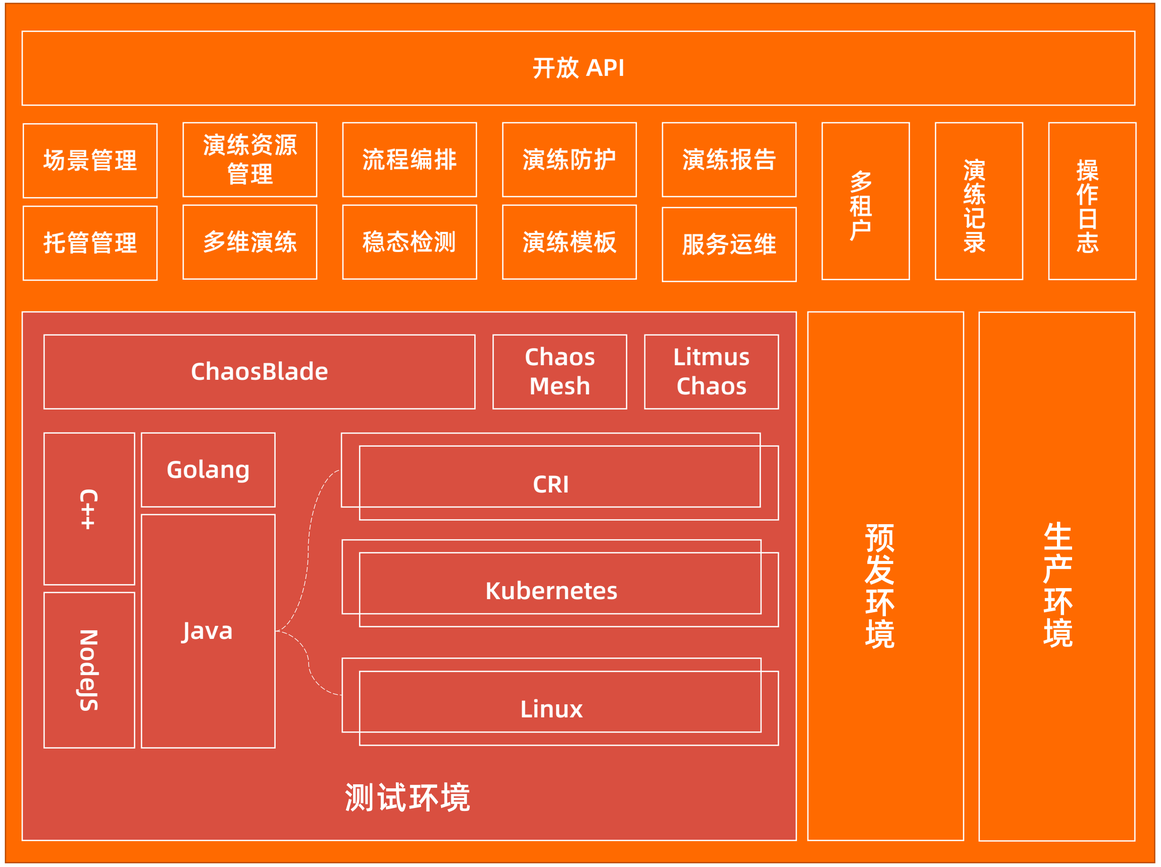
开源产品进入CNCF Sandbox项目孵化器

# 混沌工程行业化落地解决方案



# 混沌工程平台社区版

*混沌工程平台社区版定位：面向多集群、多环境、多语言的通用混沌工程平台，解决混沌工程如何启动问题 。*



# 混沌工程平台企业版

*混沌工程平台企业版定位：企业级混沌工程平台，提供规模化、场景化、自动化、安全的产品能力，覆盖IaaS、PaaS、SaaS的全栈场景。*

**混沌工程社区版**
真实事件

一键升级

**客户运维系统**

- 全链路压测：生产环境、爆炸半径
- 环境技术：生产环境、爆炸半径
- 单元化/容灾：爆炸半径
- 预案系统：爆炸半径
- 持续集成：自动化实验
- 可观测设施：稳态识别

适配集成

**混沌工程企业版 AHAS**

适配集成

**客户应用架构**

- 多语言架构：Java、Go、C++ ...
- 多平台架构：Windows、Linux
- 异构云架构：私有云、公共云...
- 多供应商架构：厂商A、厂商B ...

**客户界面**

**丰富的演练场景**

- 200+故障场景，覆盖基础设施到业务应用
- 支持SLB、ECS、RDS等多项阿里云服务
- 主机节点兼容Linux、Windows多操作系统 *NEW*
- 支持预检、断网、恢复、复盘一站式的容灾断
- 微服务演练2.0，支持服务级别强弱依赖自动化验证

**多样的演练形式**

- 自定义演练机器与场景，灵活自由
- 经验库一键演练，简单快速
- 高阶演练方案，按需配置
- 可视化演练，支持基于业务架构拓扑一键发起演练 *NEW*

**易用的演练平台**

- 业务0改造，一键接入
- 业务系统架构自动化感知
- 故障效果可视化、爆炸半径可视化
- 社区版在线托管至企业版，且支持一键升级至企业版 *NEW*

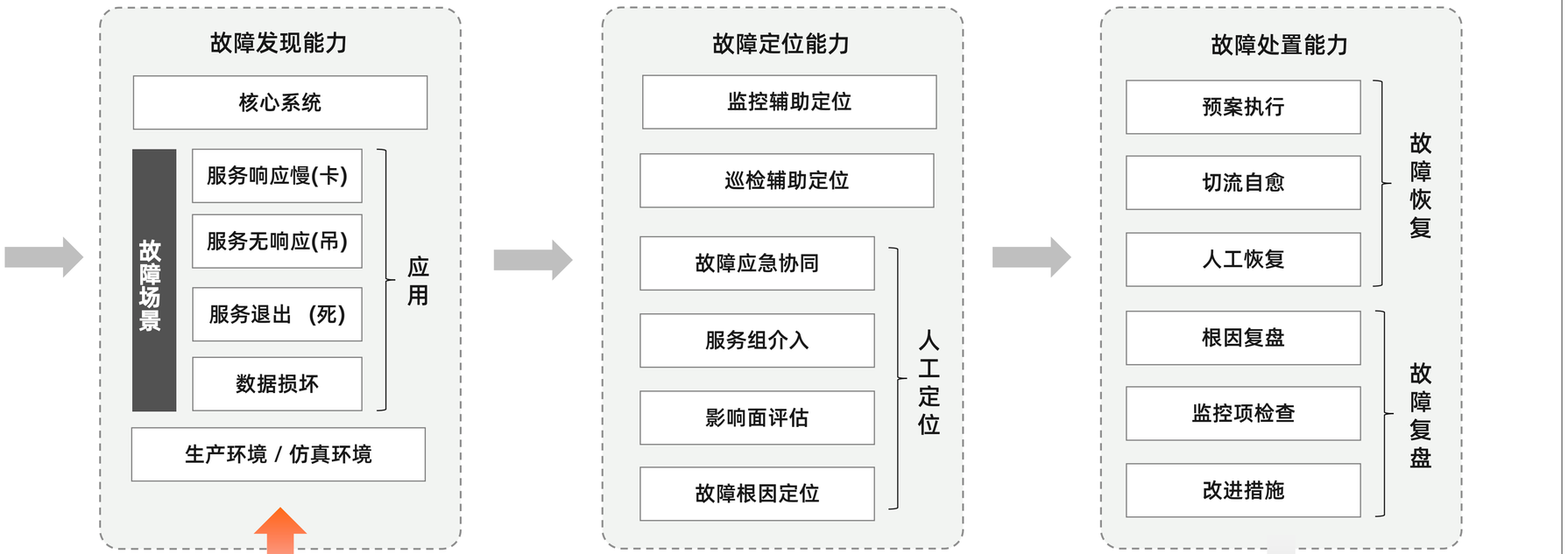
**安全的演练保障**

- 多种演练恢复策略，安全可控
- 多维度权限管控，可控演练
- 应用级别的爆炸半径 *NEW*

# 行业落地典型案例：企业版 + 规模化落地

*借助混沌工程企业版帮助客户缩短建设大规模演练实施的进程，加速演练执行效率，让客户更聚焦在架构风险识别与系统优化上，保障混沌工程实验投入产出比最大化。*

## 实施流程




故障模拟　　演练工单　　经验沉淀

### 研发

人工触发

CI / CD

### 混沌工程企业版

**演练场景**
- 基础设施演练
- 容器平台演练
- 业务应用演练
- 云服务演练

**一站式演练**
- 微服务演练
- 容器演练
- 容灾演练
- 可视化演练

**平台易用**
- 0改造、一键接入
- 架构自动感知
- 故障效果可视化
- 爆炸半径可视化

**演练防护**
- 多样化演练制动
- 环境/资源隔离
- 多维度权限管控
- 应用级爆炸半径

## 实施收益

发现风险：23 类，超过 300 项

演练次数：超过 2000 次

当日演练次数： 287 次

覆盖核心系统：超过 300 个

监控2分钟发现达标率： X %

演练成熟度： 3级

平台等级：先进级

## 组织运营

双随机演练模式

演练大屏看板

生产质量分析报告

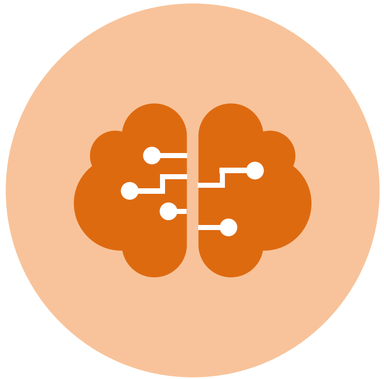
演练数据运营

专题保障项目

跨部分分享

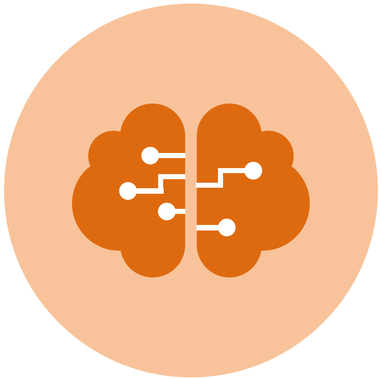
# 流量防护 Sentinel
## --- 服务治理限流降级最佳实践

# 影响服务稳定运行的典型流量场景



## 激增流量

- 激增流量导致系统 CPU / Load 飙高，无法正常处理请求
- 激增流量打垮冷系统（数据库连接未创建，缓存未预热）
- 消息投递速度过快，导致消息处理积压



## 不稳定服务依赖

- 慢 SQL 查询卡爆连接池
- 第三方服务不响应，卡满线程池
- 业务调用持续出现异常，产生大量的副作用

# 漏斗形流量防护原则

CDN

静态页面

动态请求

动静请求分离

SLB/NGINX

非法请求

有效请求

安全策略防攻击；粗放型流控

超容

容量之内

热点/普通流量探测分离；精细流控

后端应用

机器A

机器B

机器C

机器D

机器E

机器F

- 根据应用容量流控；
- 实时探测应用内部不稳定调用，及时隔离；
- 自适应系统防护

下游依赖

DB

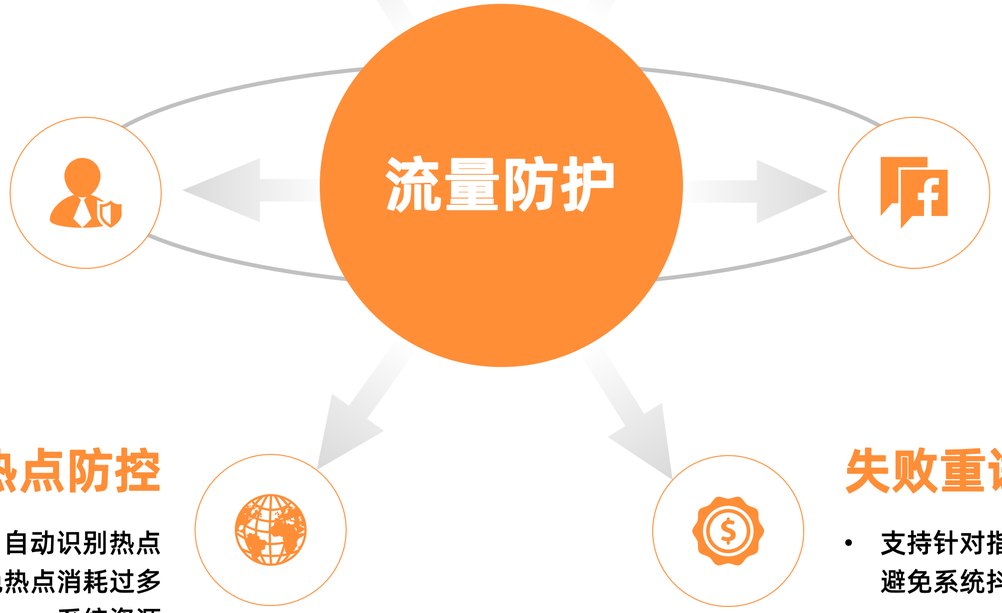
- 慢SQL；
- 缓存击穿防护；
- 不稳定第三方应用探测，隔离

其它应用

缓存

数据库

# 流量防护 Sentinel 企业版核心能力




## 流量控制

- 提供单机限流、集群限流、分钟小时限流多种限流方式
- 支持滑动窗口、令牌桶、漏桶多种限流算法

## 并发控制

- 强依赖场景下，支持设定接口并发数，精准控制强依赖消耗的线程资源

## 系统保护

- 自适应系统保护，自动根据系统实时处理能力调节处理流量大小
- 可根据 CPU、LOAD 等系统资源指标，设定系统保护规则

## 熔断降级

- 弱依赖场景下，支持根据接口调用RT、异常比例做异常熔断，保证弱依赖不拖垮系统
- 提供无嵌入的主动降级功能，活动前动态降级弱依赖

## 热点防控

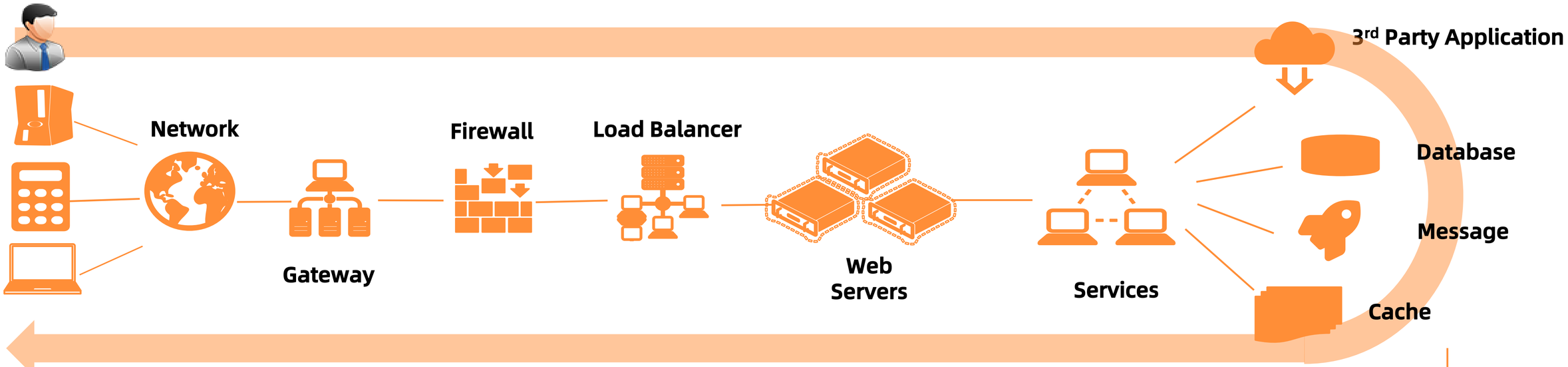
- 自动识别热点
- 支持设定单个热点流量请求大小，避免热点消耗过多系统资源

## 失败重试

- 支持针对指定的非致命错误、异常自动重试，最大限度避免系统抖动

# 全链路场景流量防护全覆盖

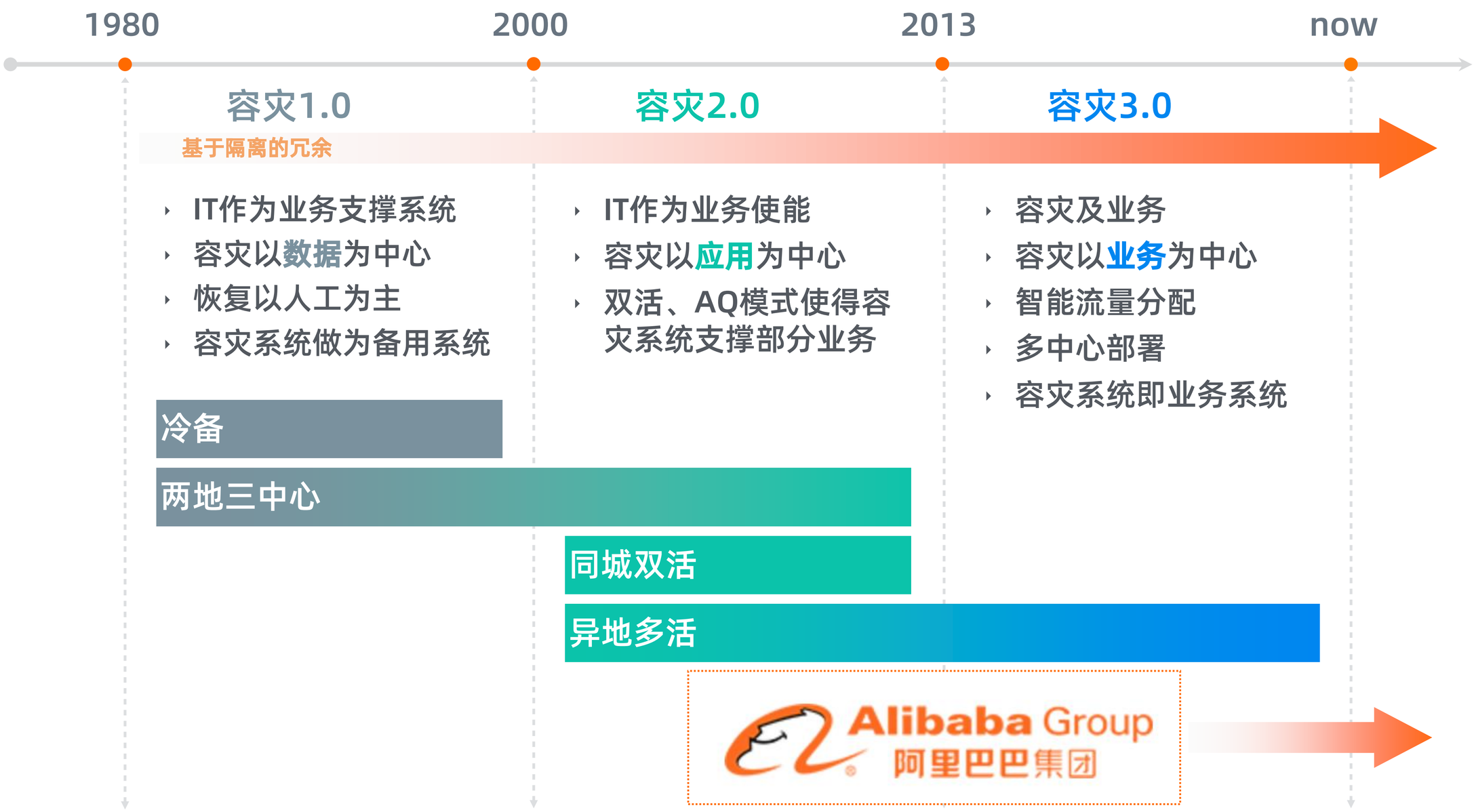
*以流量控制为抓手，提供线上流量防护，热点探测，异常发现，慢方法熔断，系统防护等一系列防护手段，帮助用户提高业务在各种情况下的稳定性。*




| 客户端 | 业务链路入口 | 服务内部 | 服务与依赖 |
| --- | --- | --- | --- |
| • 流量实时监控<br>• 水位诊断分析<br>• 热点防控<br>• 业务排序 | • 链路入口流控<br>• 热点漏斗 | • 服务水位流控<br>• 同步匀速器<br>• 异步消峰填谷 | • 慢SQL发现熔断<br>• 慢方法熔断<br>• 过载保护 |

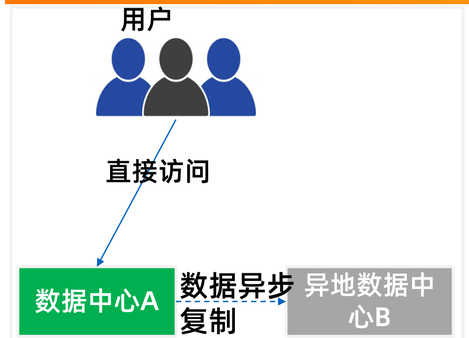
# 多活容灾 MSHA
## --- 业务级多活最佳实践

# 容灾系统职能的变化


1980
2000
2013
now
容灾1.0
容灾2.0
容灾3.0
基于隔离的冗余
IT作为业务支撑系统
容灾以数据为中心
恢复以人工为主
容灾系统做为备用系统
IT作为业务使能
容灾以应用为中心
双活、AQ模式使得容灾系统支撑部分业务
容灾及业务
容灾以业务为中心
智能流量分配
多中心部署
容灾系统即业务系统
冷备
两地三中心
同城双活
异地多活
Alibaba Group
阿里巴巴集团

# 几种主流的容灾架构

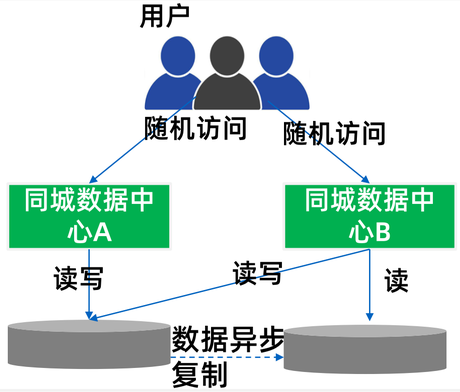
## 异地冷备




**主要特征：**

- 简单，对于业务侵入较少；
- 不提供在线服务；
- 应用和数据库均是冷备；
- 支持异地部署；
- **发生故障时难以保证切换可靠性，且备份全站成本非常高。**

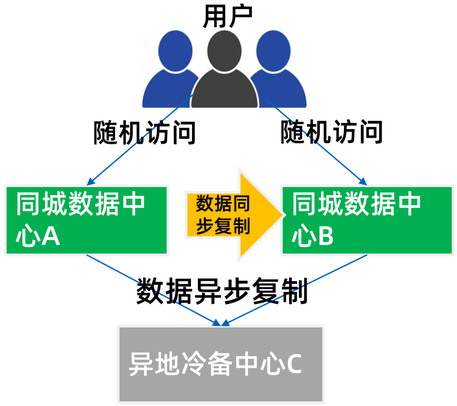
## 同城双活




**主要特征：**

- 数据热备；
- 应用分担；
- 数据可读；
- 同城部署；
- **无法做到跨省机房间的无缝切换；**

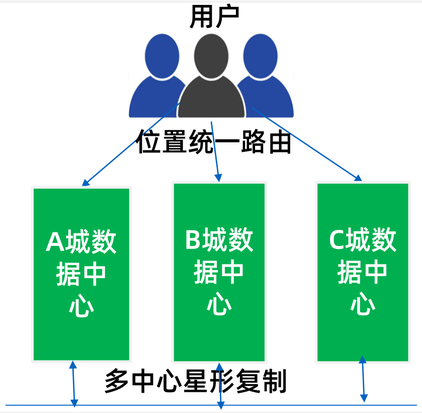
## 两地三中心




**主要特征：**

- 业务侵入少，建设速度快；
- 同城范围有效地保证了数据的安全性和业务系统；
- **冷备中心成本浪费；**
- **关键时刻不敢切换；**
- **数据单点写存在瓶颈；**

## 异地多活




**主要特征：**

- **实现单元化的业务跨省机房的无缝切换；**
- 部署层面单元化，多个单元之间无访问；
- 调度层面切换禁写局部化、入口引流统一化；
- 数据层面局部化，基于数据时间戳局部同步；
- 数据修复后项化，禁写后的后向数据修复；

# 业务灾备与多活容灾的差异

## 主备容灾

针对某一业务，生产状态下只在主数据中心运行；当灾难发生，主数据中心瘫痪时，业务需要在灾备中心恢复对外提供访问；

- 备中心业务静默
- 服务能力正常
- 灾难故障切换时间长，用户感知
- RPO，视灾备建设而定数据丢失量
- RTO > 0

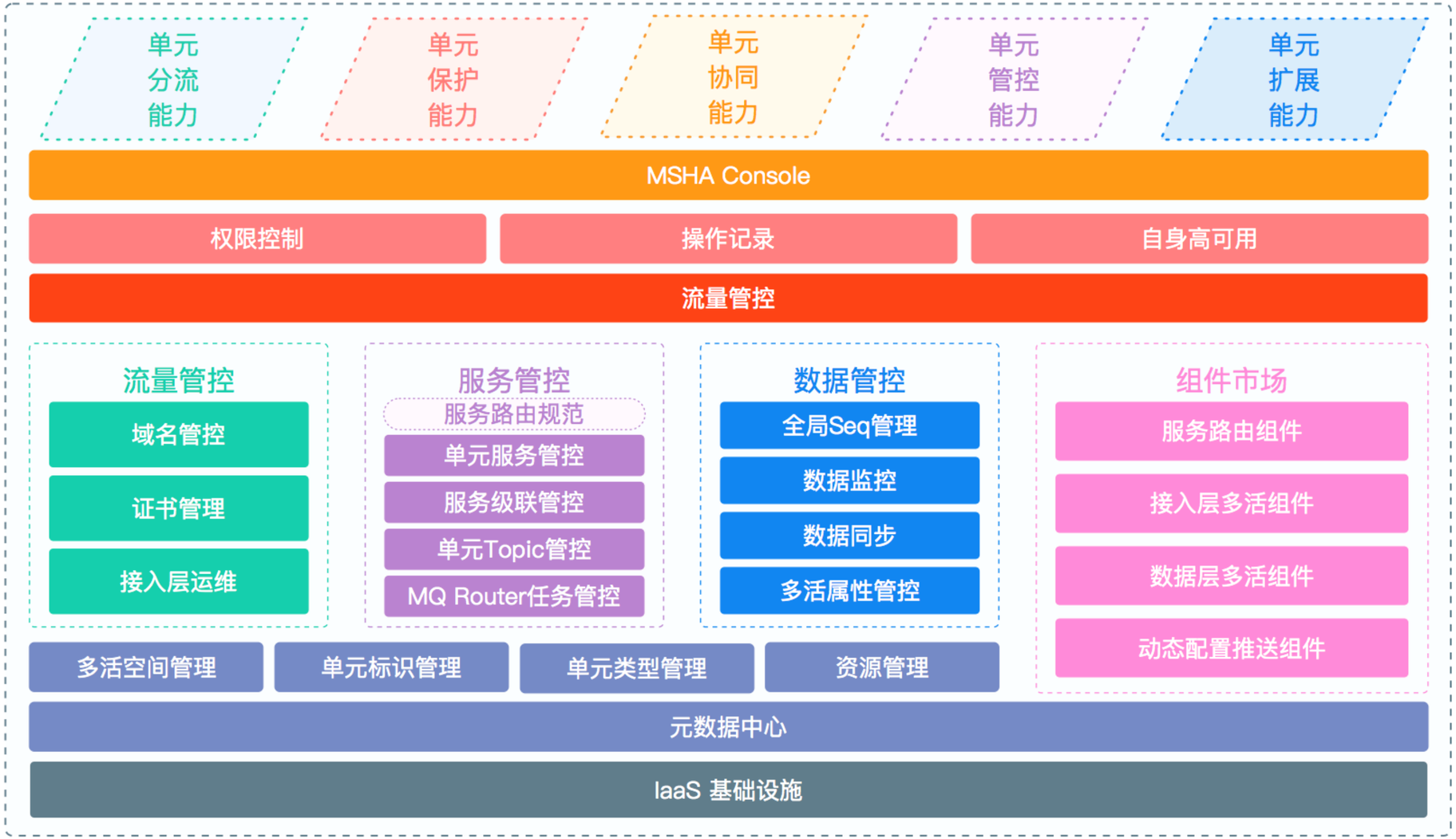
## 业务双活

同一业务在两个数据中心同时运行，通过跨中心的负载均衡对外提供访问；当灾难发生，主数据中心瘫痪时，业务不需要再恢复，通过全局负载均衡自动导流由备中心承担；

- 充分利用资源
- 服务能力翻倍（负载均衡）
- 灾难发生时，业务切换迅速，用户无感知
- RPO≈0
- RTO≈0

# 多活容灾 MSHA 产品能力

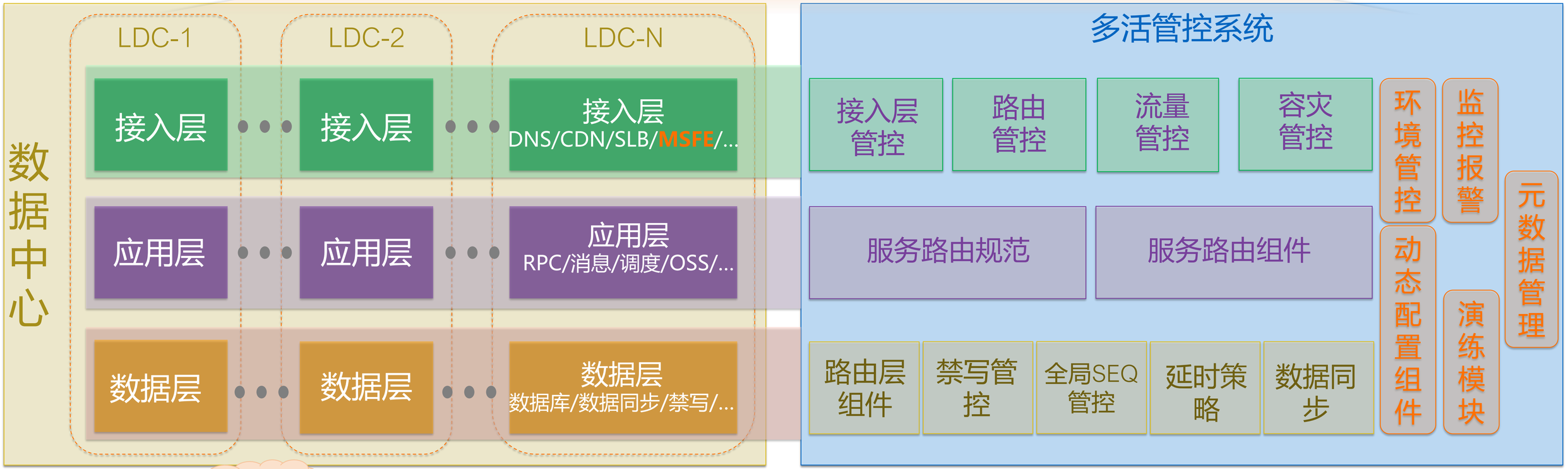
*多活，顾名思义就是分布在多个站点同时对外提供服务。与传统的灾备的最主要区别就是“多活”里的所有站点同时在对外提供服务，不仅解决了容灾本身问题，提升了业务连续性，并且实现了容量的扩展。*




- 支持丰富的容灾架构
- 一站式容灾管控
- 分钟级容灾切换
- 自动化容灾运维

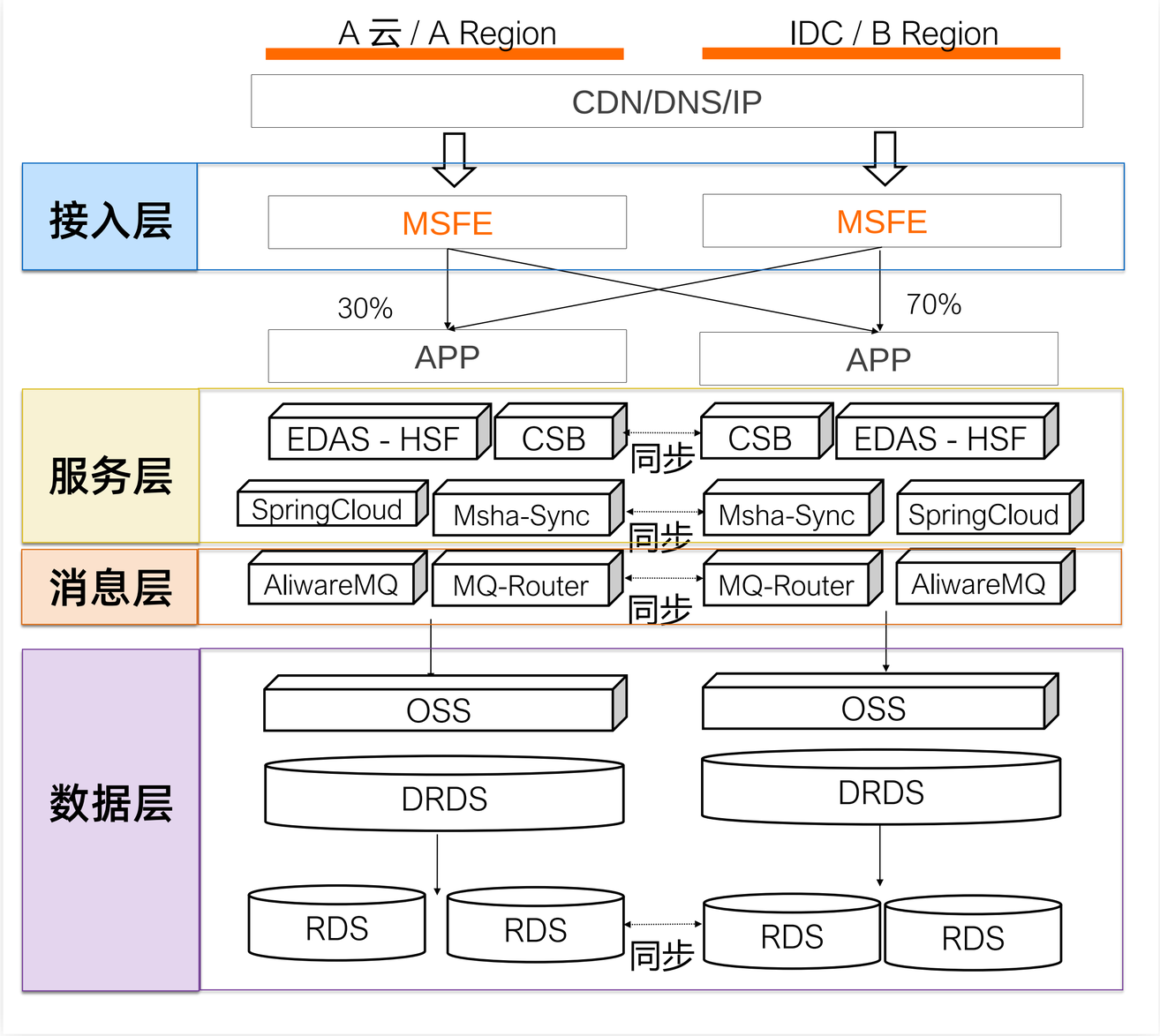
# 多活容灾 MSHA 产品架构

同城多活 | 异地灾备 | 异地双读 | 异地双活 | 异地多活 | 异地应用双活




多活接入层组件 + 应用层插件 + 数据层插件 + 数据同步 + 多活控制台

# 业务多活容灾典型场景：异地多活




01 单元化部署

02 单元请求分流

03 业务代码改造

04 数据单元保护

05 数据星型复制

06 容灾切流




WWW.ALIYUN.COM