

</> </> </> </>

+++

Spring Cloud Alibaba Meetup

杭州站 · 双11特别活动

# 微服务容易挂？ Sentinel Go 让服务稳如磐石

---

赵奕豪（宿何）

Sentinel PMC

+++

</> </> </> </>

01

●  
微服务可用性  
的场景

02

Sentinel Go  
核心场景实践

03

云原生时代的  
高可用防护

04

Roadmap &&  
Community

# 微服务的可用性的问题



4个9 - 52.6 分钟

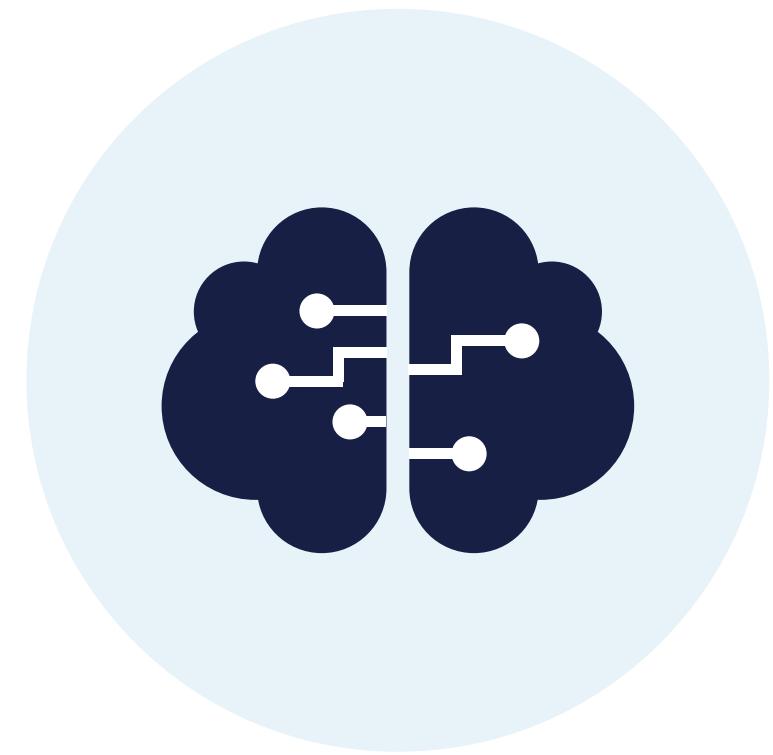
5个9 - 5 分钟

当服务挂掉的时候  
我们首先想到什么原因？



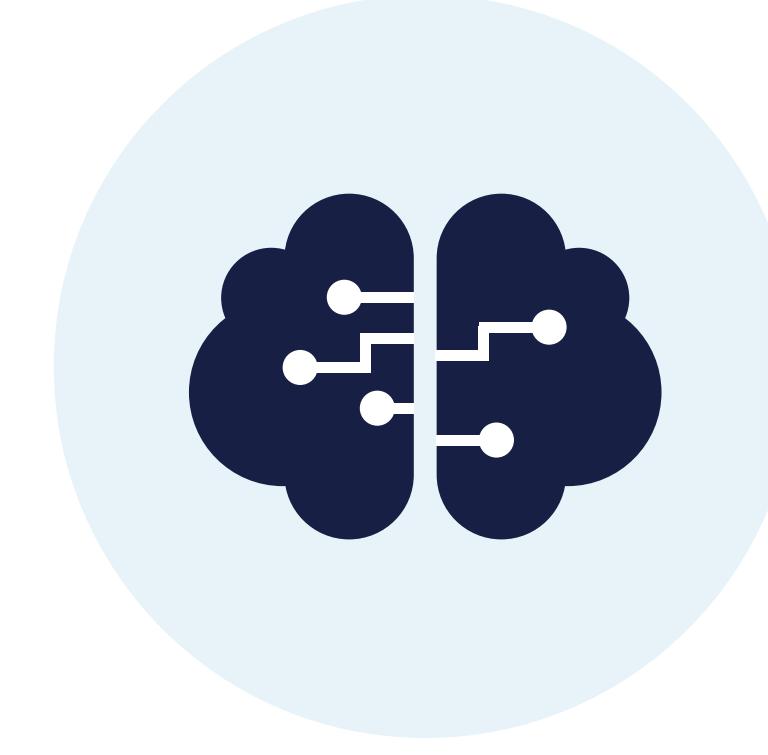
缺乏高可用防护/容错机制  
尤其是针对流量的防护

# 微服务可用性的场景



## 激增流量

- 激增流量导致系统 CPU / Load 非常高，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

Reliability & Resilience

01

微服务可用性  
的场景

02

Sentinel Go  
核心场景实践

03

云原生时代的  
高可用防护

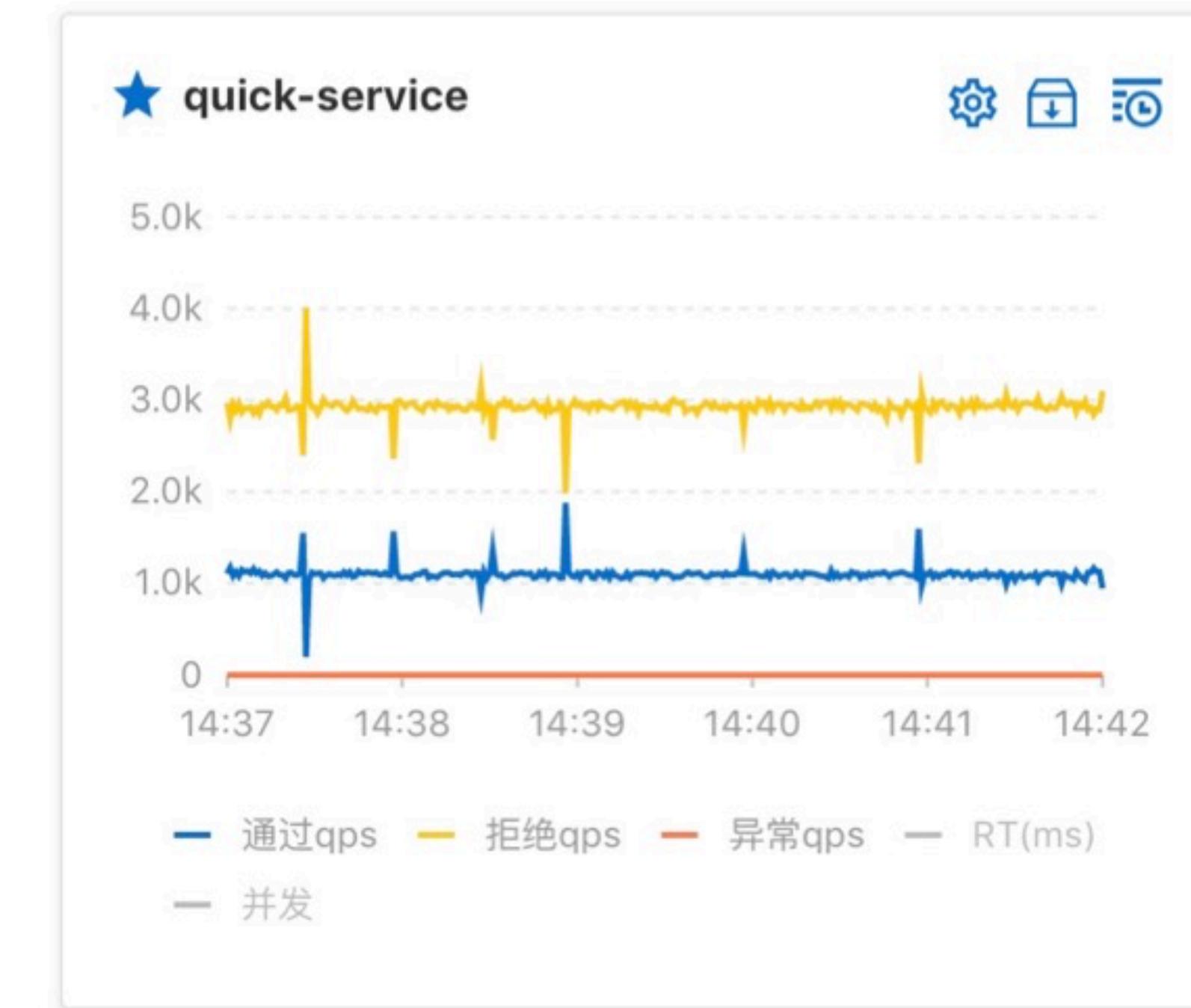
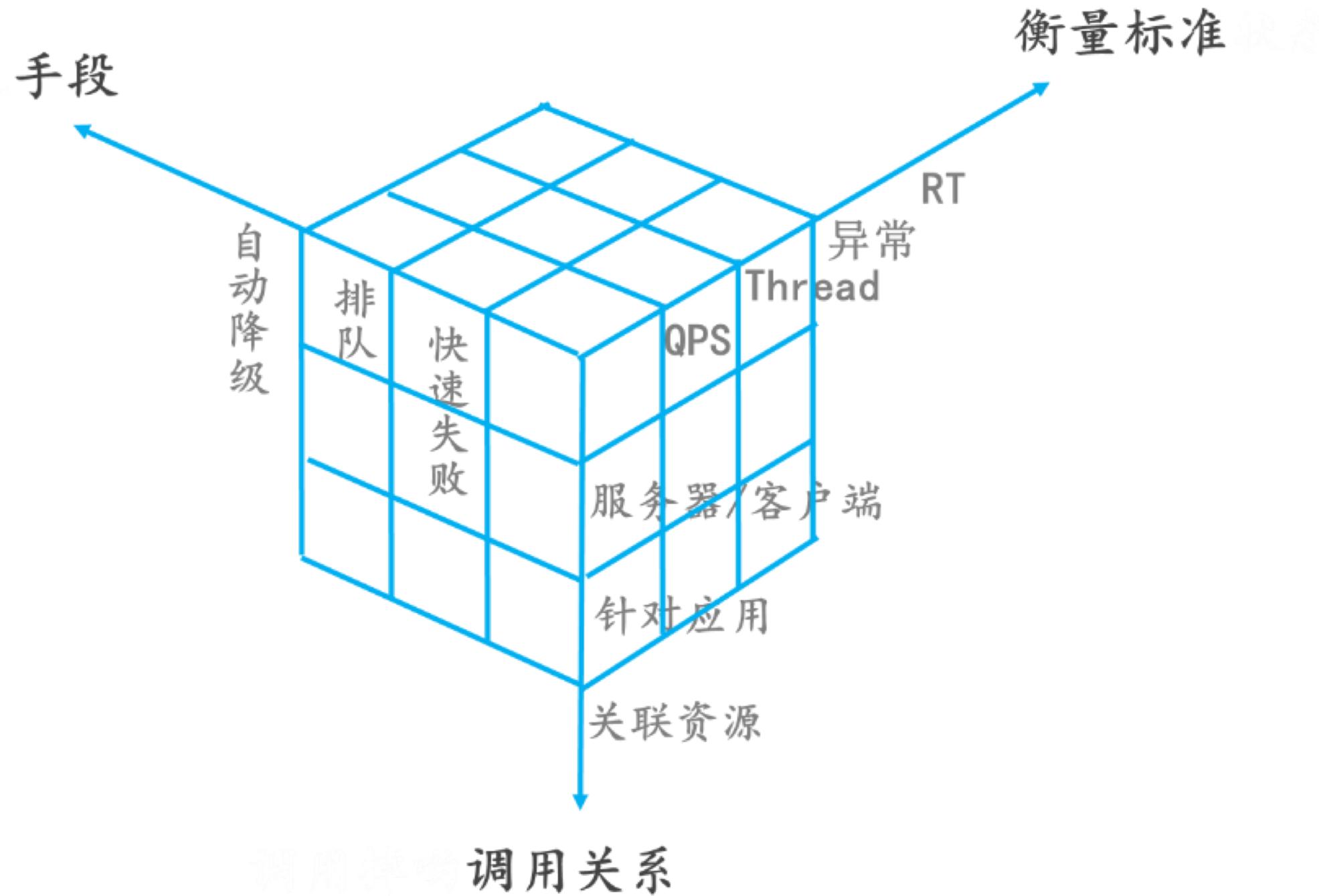
04

Roadmap &&  
Community

# You NEED Sentinel



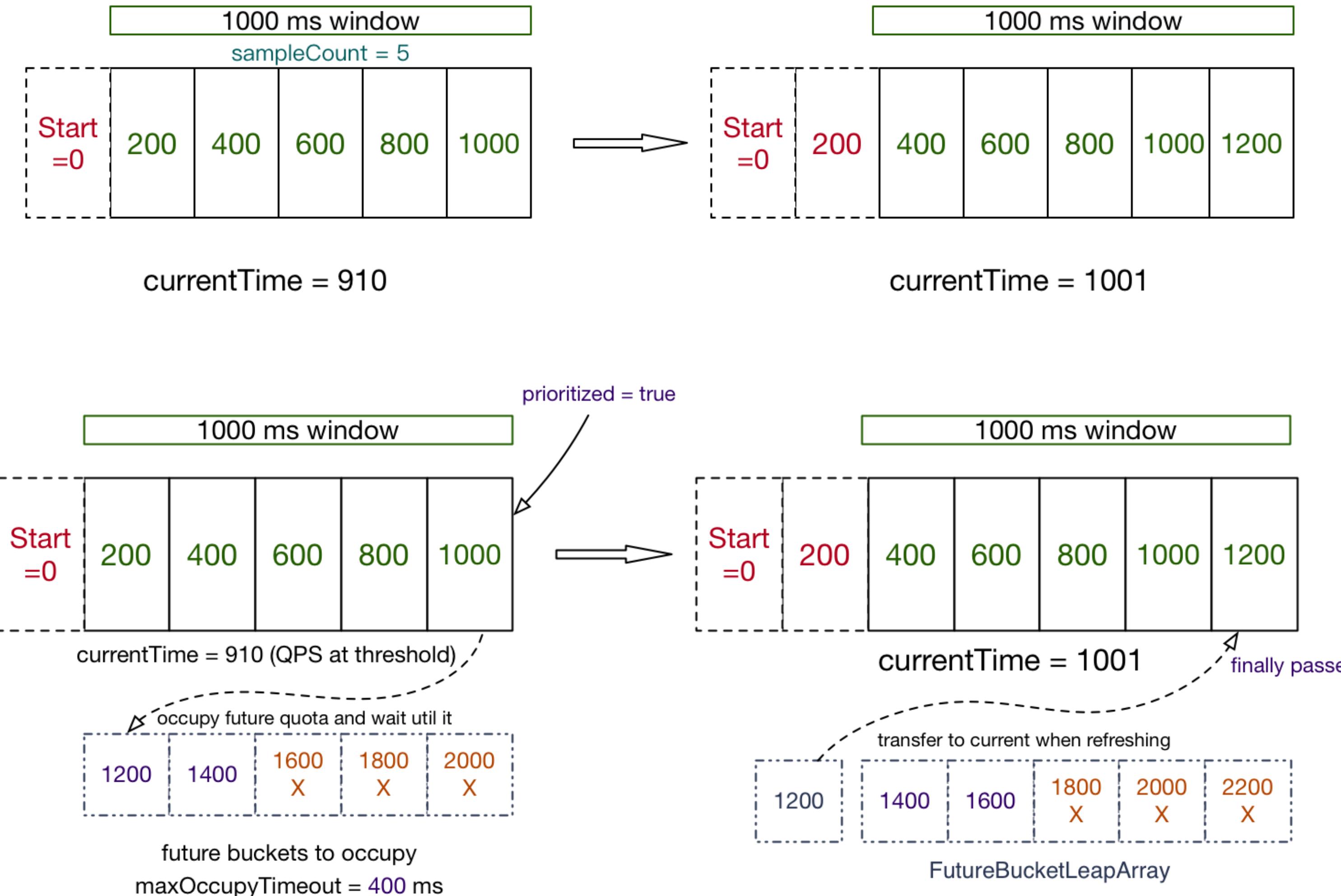
Sentinel 是阿里巴巴开源的，面向云原生分布式服务架构的高可用防护组件



多维度的流控降级能力  
自适应保护系统

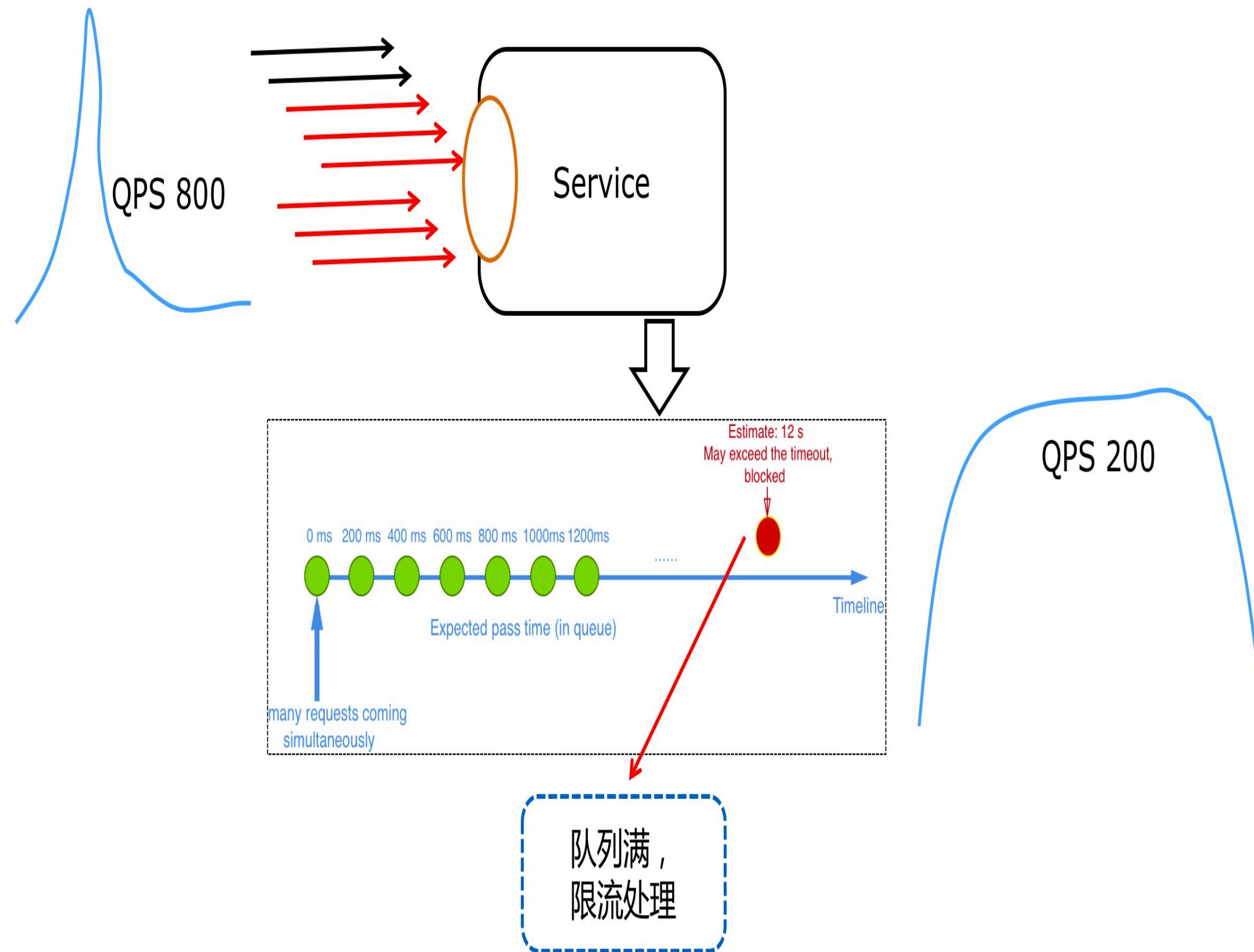
秒级实时监控与动态规则管理

# Sentinel 核心技术-实时统计



滑动窗口-基础统计结构

# 流量控制



## 最普适的场景：

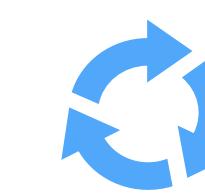
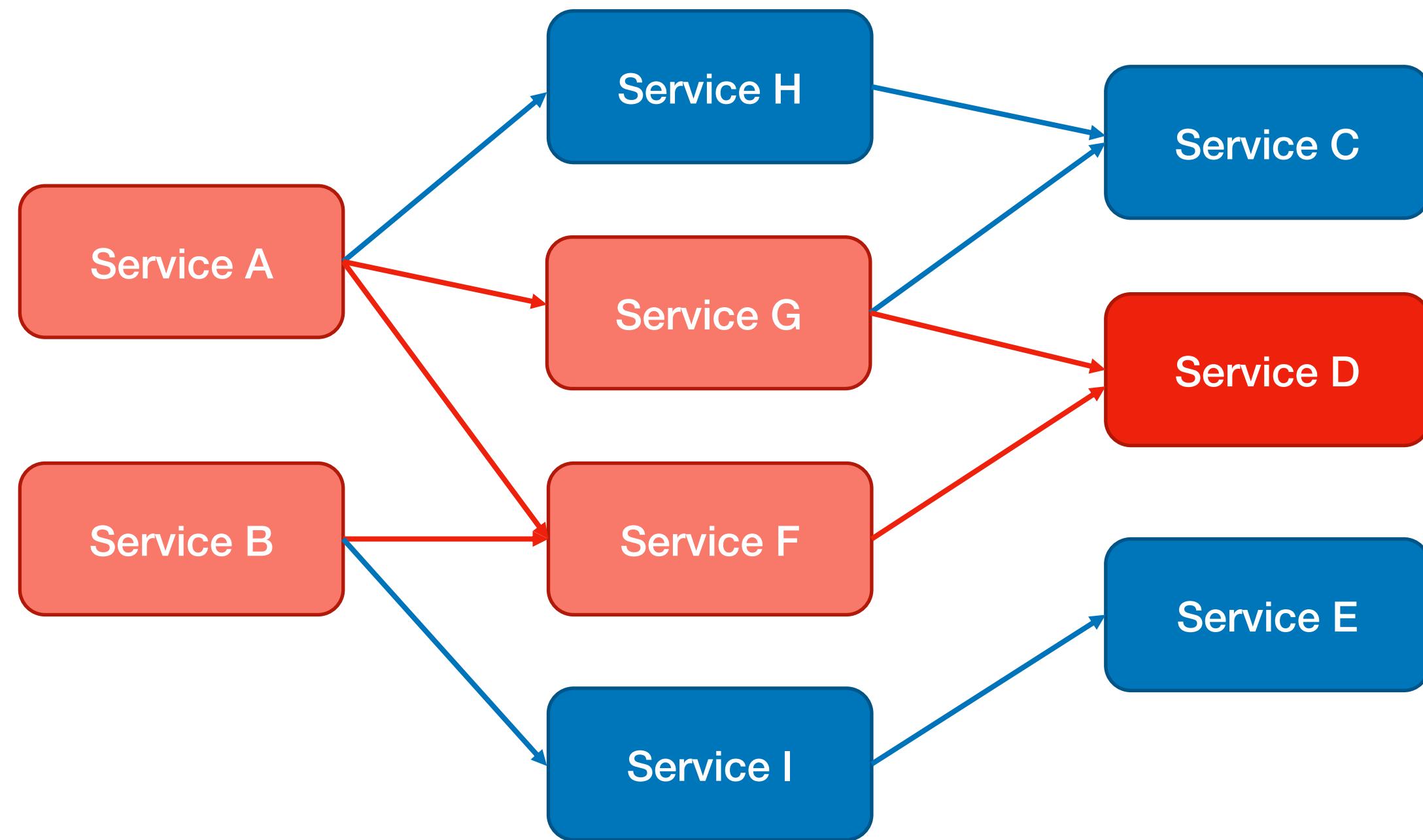
- RPC provider 端控制脉冲流量
- 针对不同调用来源进行流控
- Web 接口流控

## 如何配置规则：

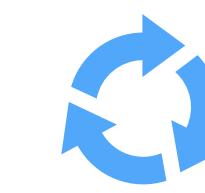
1. 梳理核心接口
2. 通过事前压测评估核心接口的容量，配置 QPS 阈值
3. 注意流控处理逻辑

**流量不大，万事无忧？**

# 熔断降级与隔离的场景



出现了慢SQL，慢SQL导致应用越来越慢，最后整个应用卡挂了。



应用依赖了第三方服务，第三方服务突然不响应，造成应用线程也被挂在第三方应用上无法返回，最后自己的线程也耗尽，也无法处理新的请求。



程序内部某个方法调用持续异常。这个时候调用这个方法已经毫无意义了，而且会影响主营业务流程。

# 熔断降级与隔离

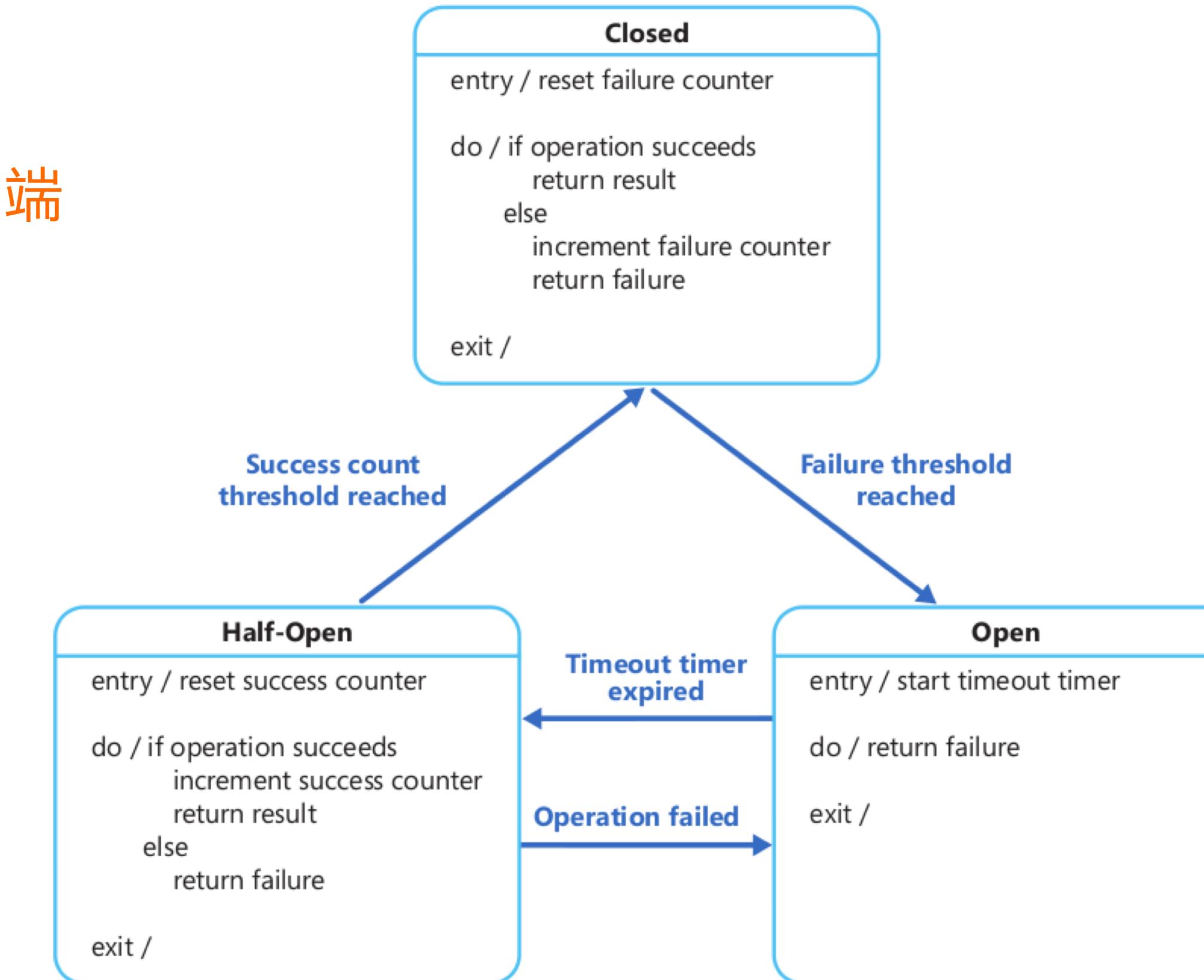
## 保护自身的手段

- 并发控制（信号量隔离）
- 基于慢调用比例熔断
- 基于异常比例熔断

通常在 **consumer** 端  
组合配置

## 触发熔断后的处理逻辑示例

- 提供 fallback 实现（服务降级）
- 返回错误 result
- 读缓存（DB 访问降级）



**熔断降级的前提**：梳理强弱依赖，只有弱依赖才能降级

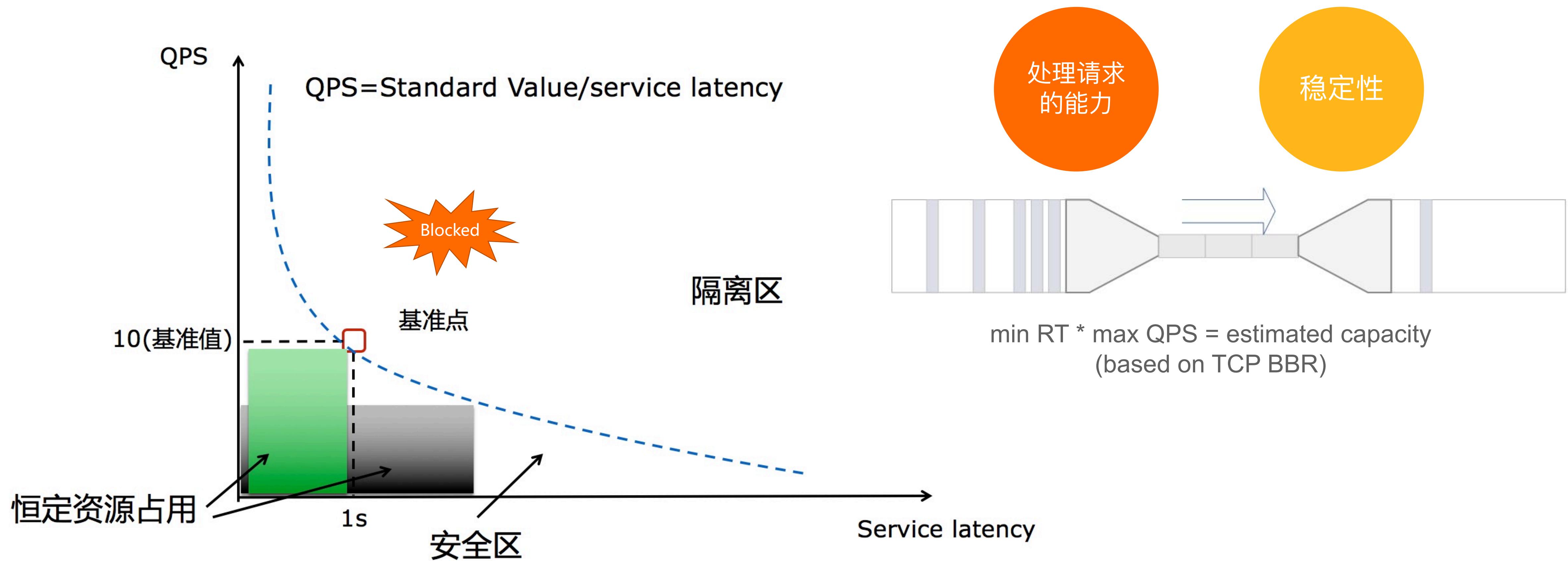
# 系统自适应保护的场景



- 容量评估不到位，某个大流量接口限流配置不合理或没有配置，导致系统崩溃，来不及进行处理。
- 突然发现机器的 Load 和 CPU usage 等开始飚高，但却没有办法很快的确认到是什么原因造成的，也来不及处理。
- 当其中一台机器挂了之后，本该由这台机器处理的流量被负载均衡到另外的机器上，另外的机器也被打挂了，引起系统雪崩。
- 希望有个全局的兜底防护，即使缺乏容量评估也希望有一定的保护机制。

We NEED  
结合系统指标和服务容量，**自适应**动态调整流量

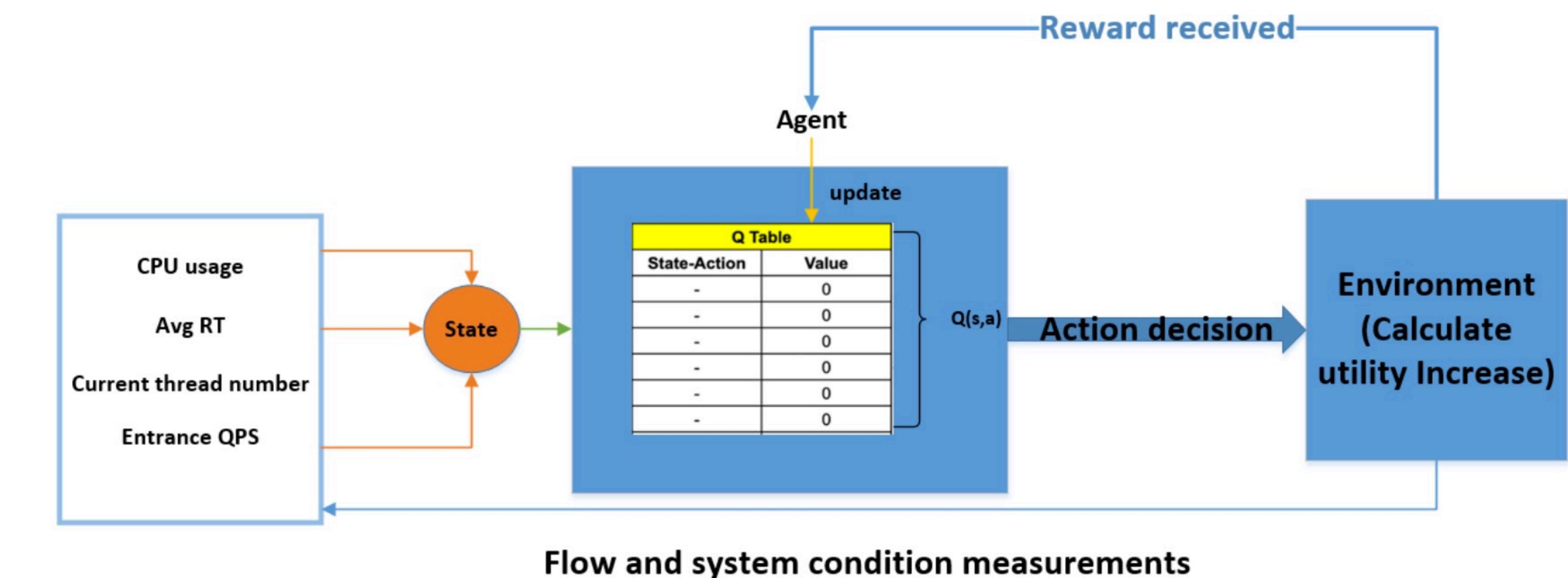
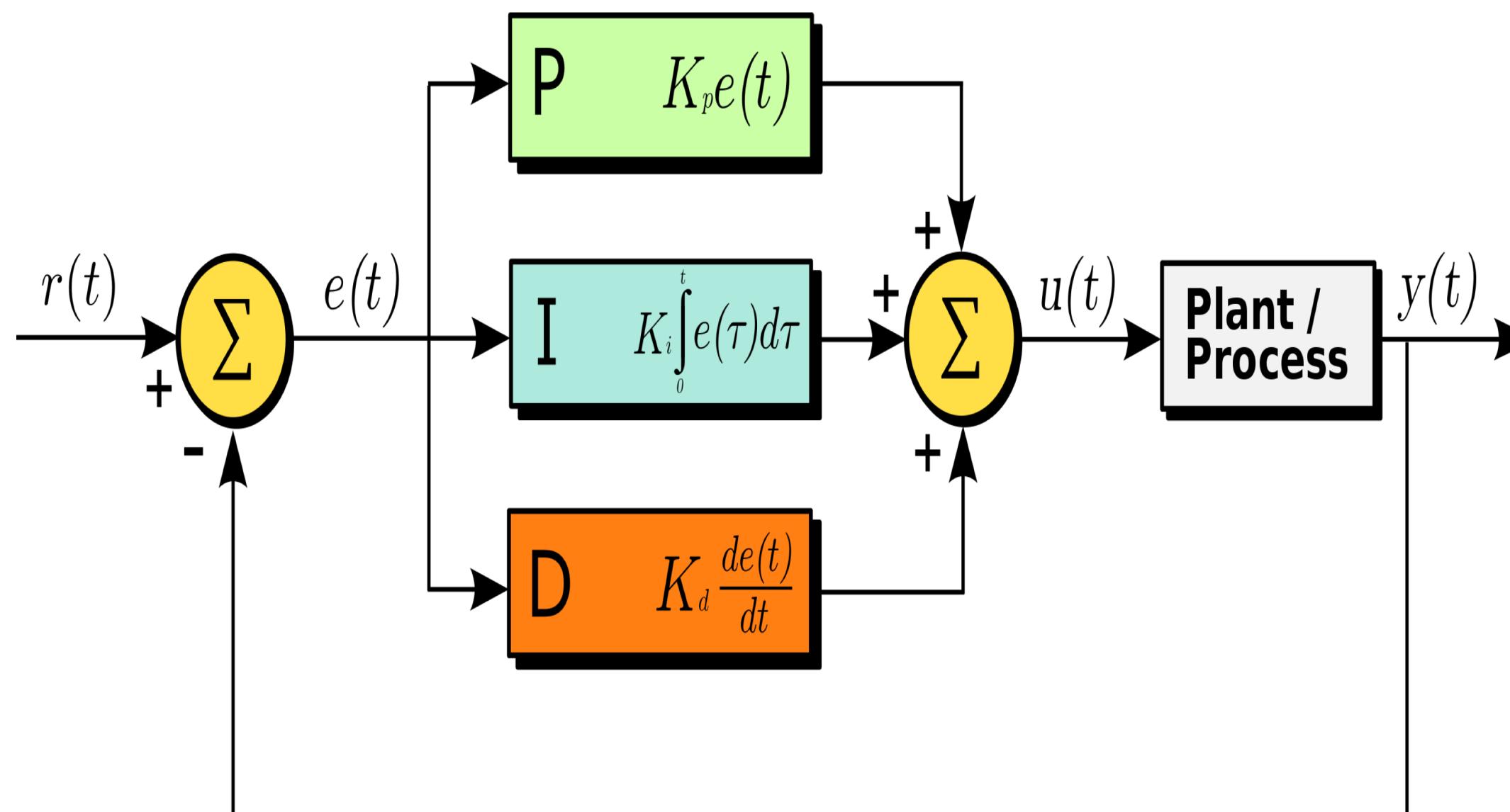
# 系统自适应保护原理



系统指标 (load/CPU usage) 作为启发条件，动态探测系统预估容量 (并发) 并对入口流量进行自适应调整

兜底必配！！！

# 自适应流控演进



$$u(t) = K_p e(t) + K_i \sum_{i=1}^k e(t_i) \Delta t + K_d \frac{e(t_k) - e(t_{k-1})}{\Delta t}$$

State  $\leftrightarrow$  Action  $\leftrightarrow$  Reward

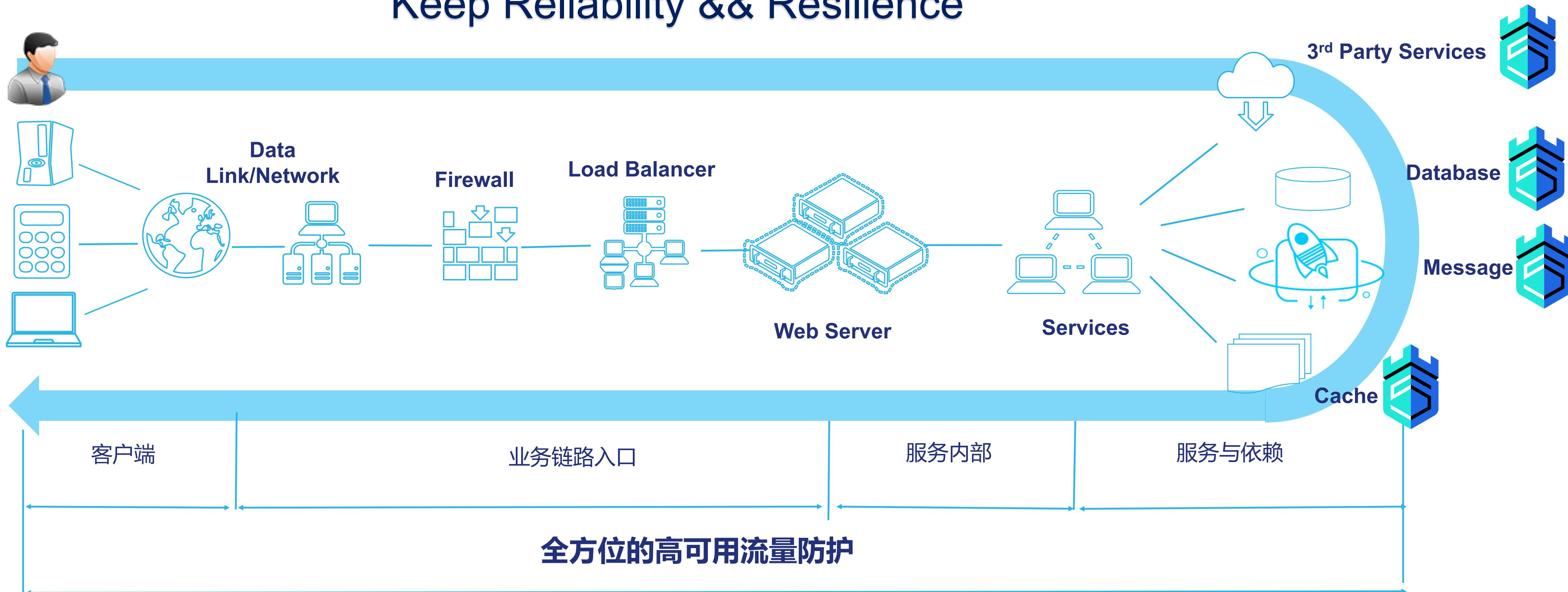
**Proportional–Integral–Derivative Controller**

**Reinforcement Learning**

Sentinel 也在结合自动化控制理论（如 PID controller）和强化学习算法来更好地完善自适应流控的效果

# Sentinel 高可用防护布局

Keep Reliability && Resilience



**Sentinel Go 不仅仅针对微服务自身**

Kubernetes

Open  
Telemetry

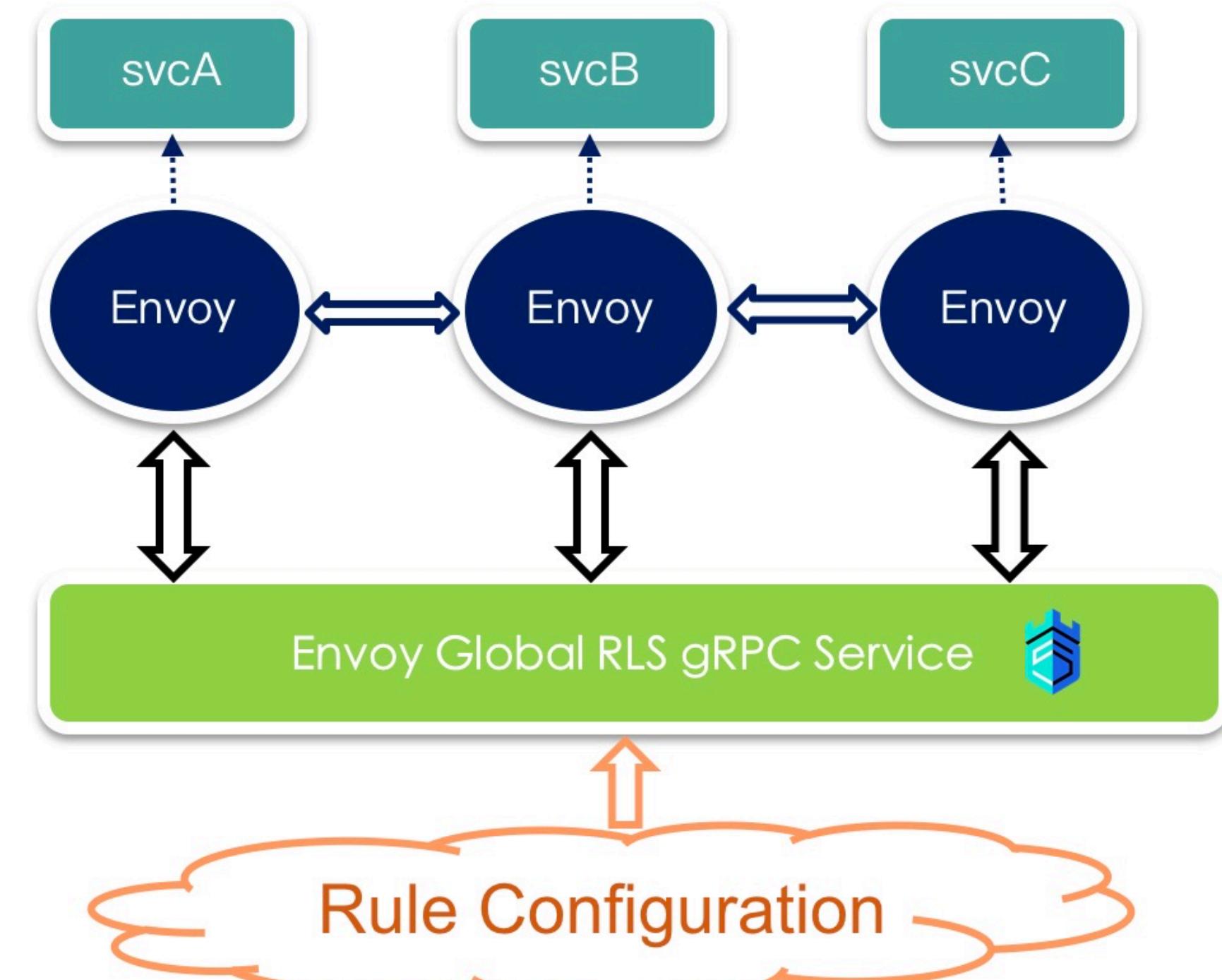
Service  
Mesh

Serverless

# 云原生时代的高可用防护形态？



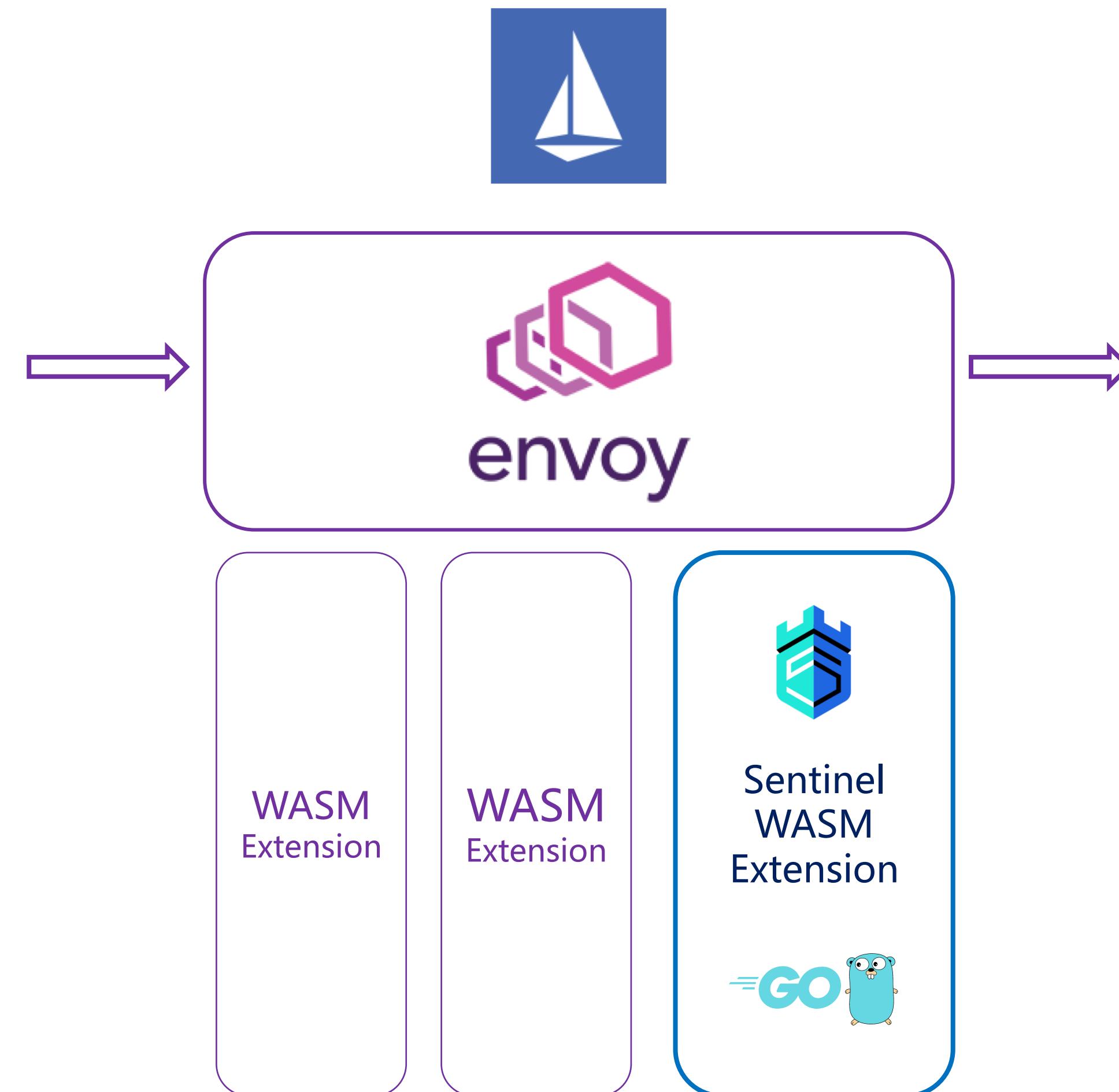
# Service Mesh 全局流量控制



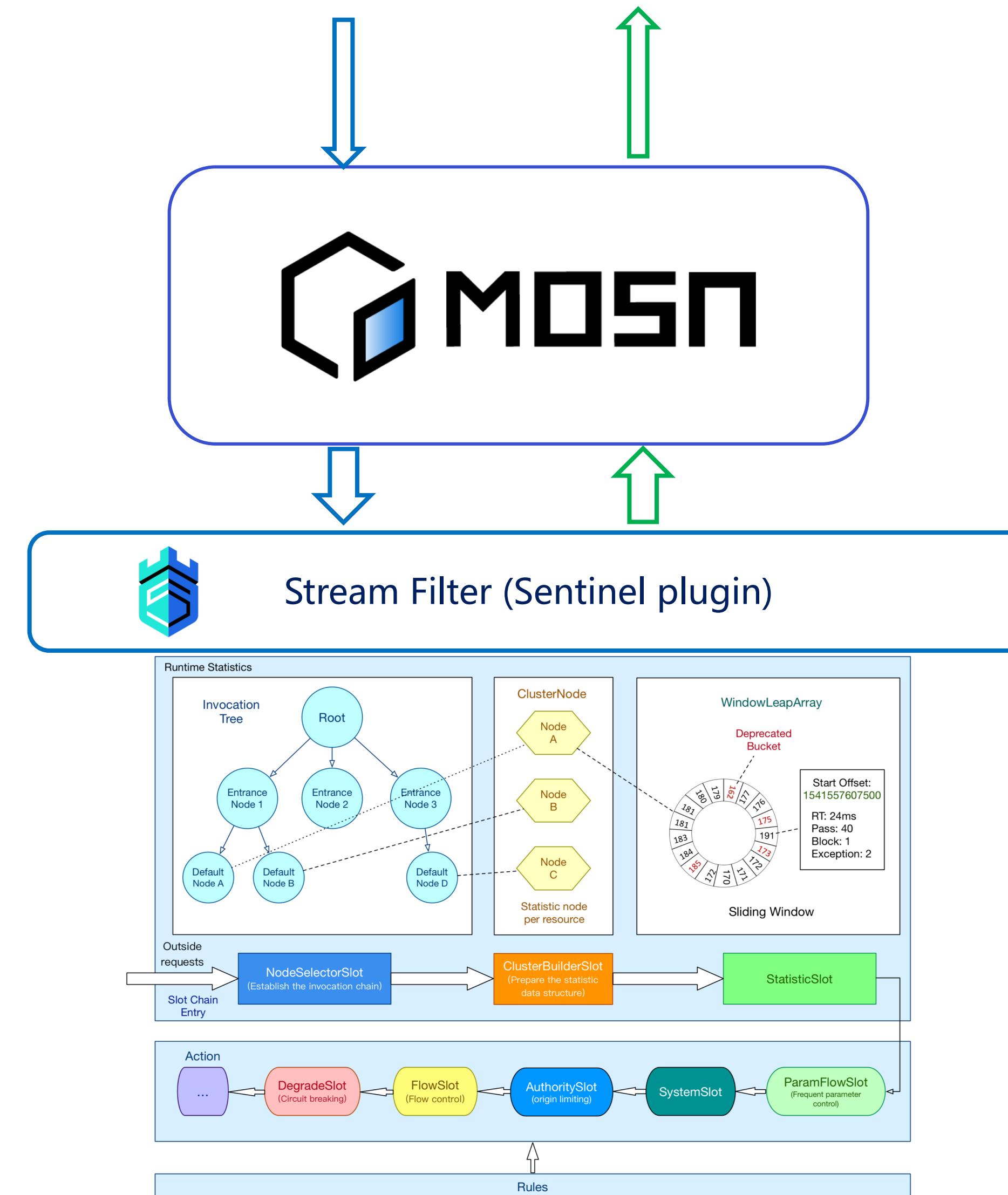
Envoy + Istio 场景，借助 Sentinel RLS token server 为 Mesh 层提供全局流量控制的能力

- K8S 环境示例：<https://github.com/alibaba/Sentinel/tree/master/sentinel-cluster/sentinel-cluster-server-envoy-rls/sample/k8s>
- 阿里云 AHAS Sentinel Mesh 流控：<https://ahas.console.aliyun.com/flowPriotecton/meshProtection>

# Service Mesh 高可用防护



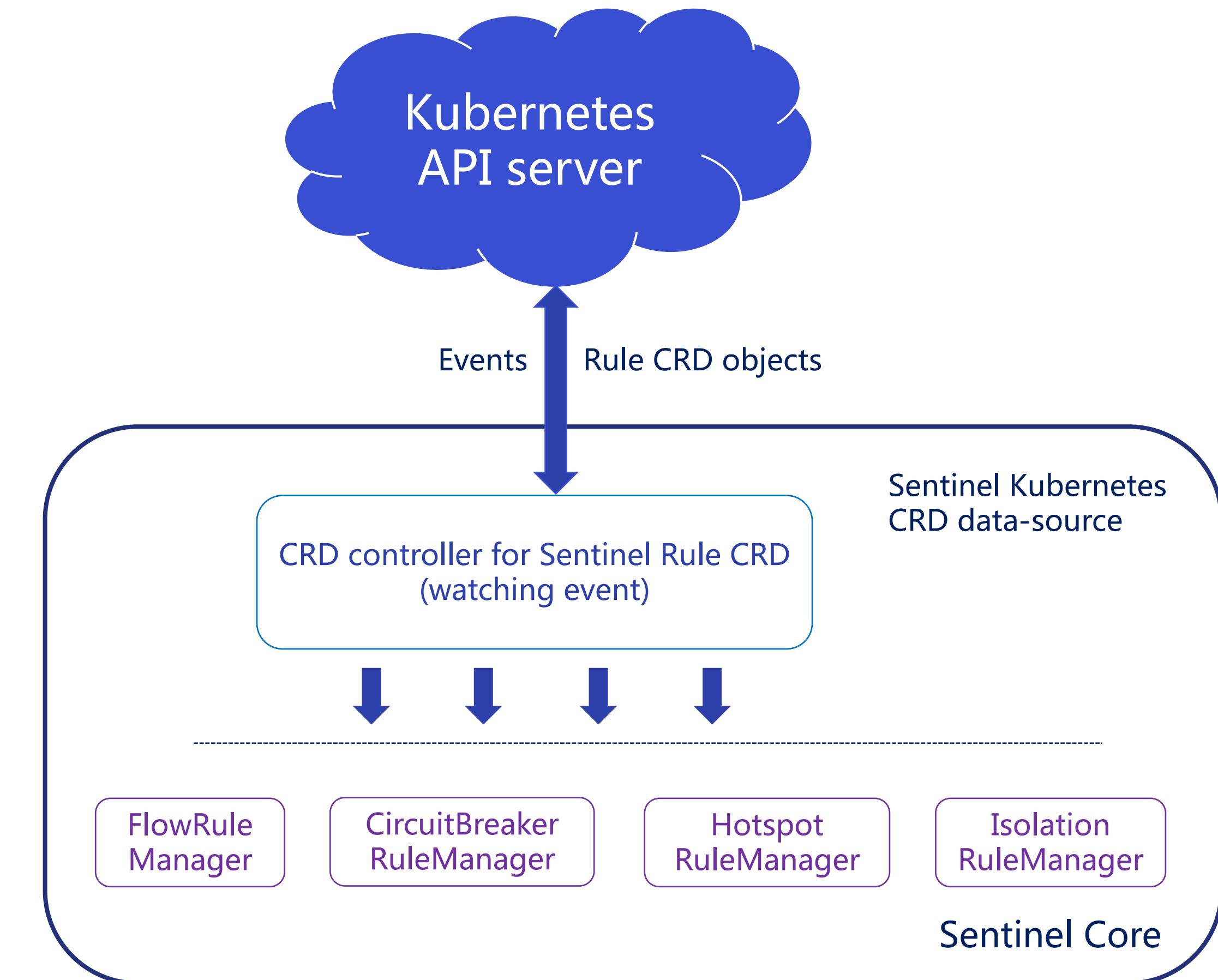
# Envoy Proxy WASM Extension



# MOSN Plugin

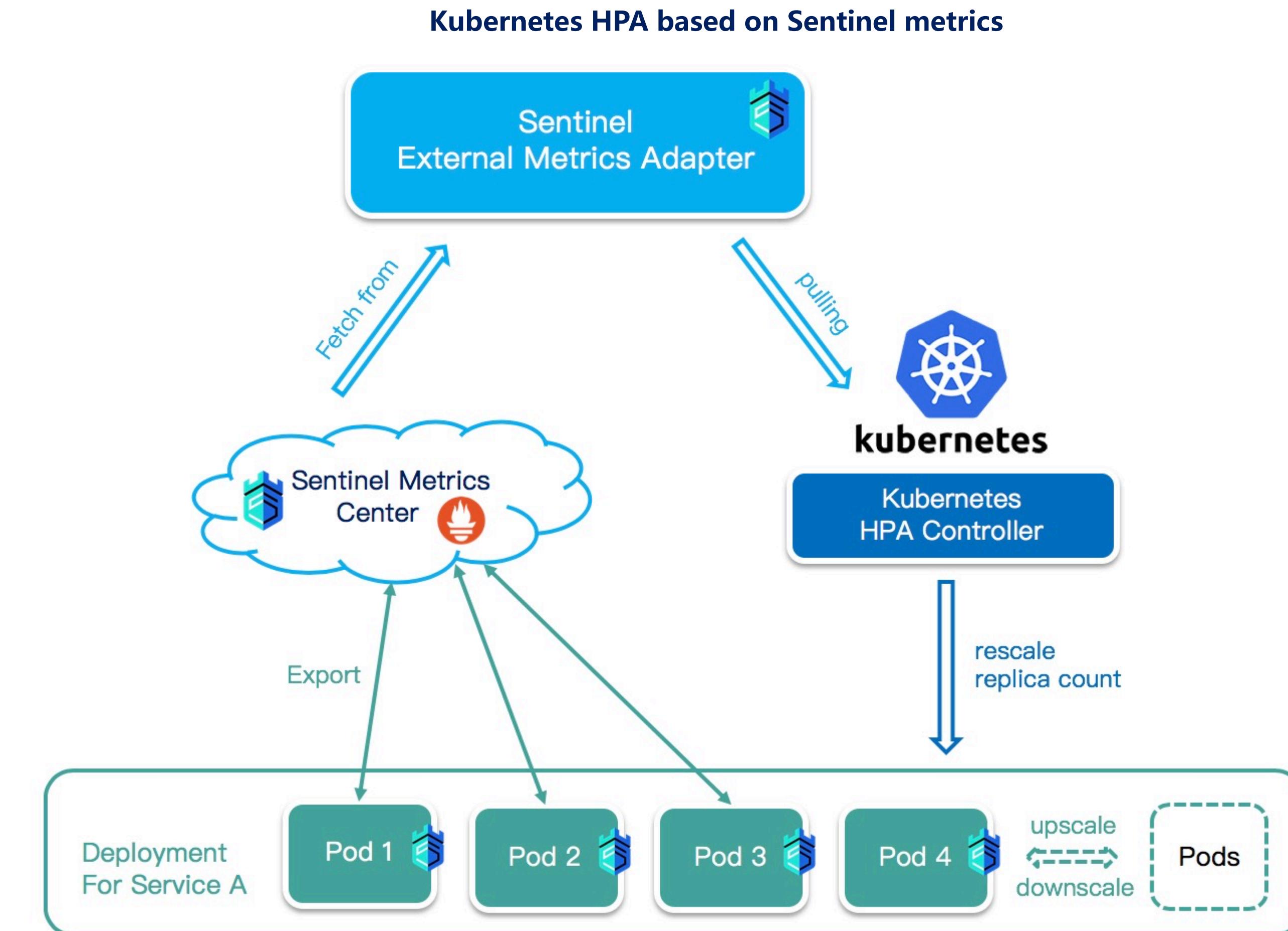
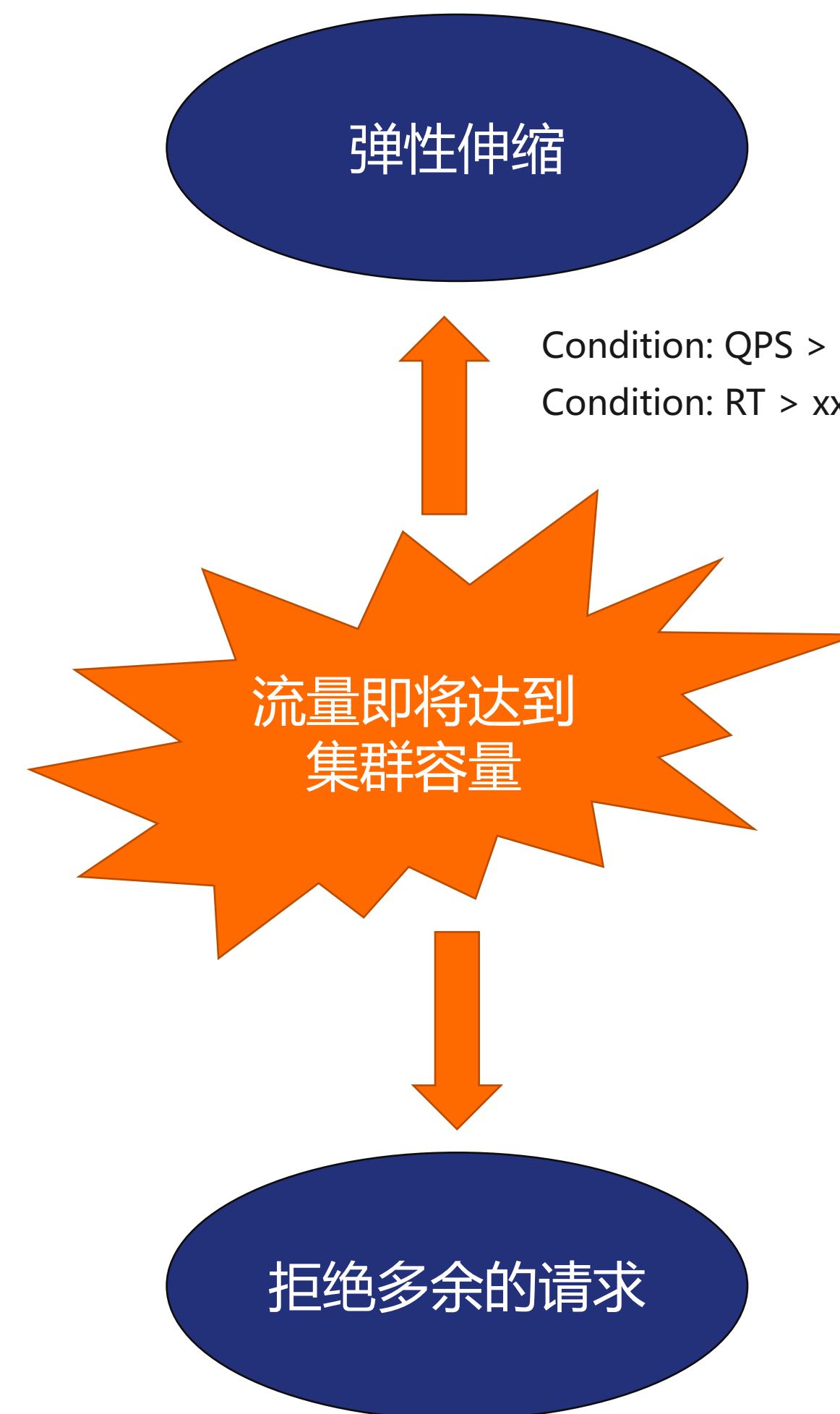
# 动态规则管理的方式 - Kubernetes CRD data-source

```
apiVersion: datasource.sentinel.io/v1alpha1
kind: FlowRules
metadata:
  name: foo-sentinel-flow-rules
spec:
  rules:
    - resource: simple-resource
      threshold: 500
    - resource: something-to-smooth
      threshold: 100
      controlBehavior: Throttling
      maxQueueingTimeMs: 500
    - resource: something-to-warmup
      threshold: 200
      tokenCalculateStrategy: WarmUp
      controlBehavior: Reject
      warmUpPeriodSec: 30
      warmUpColdFactor: 3
```



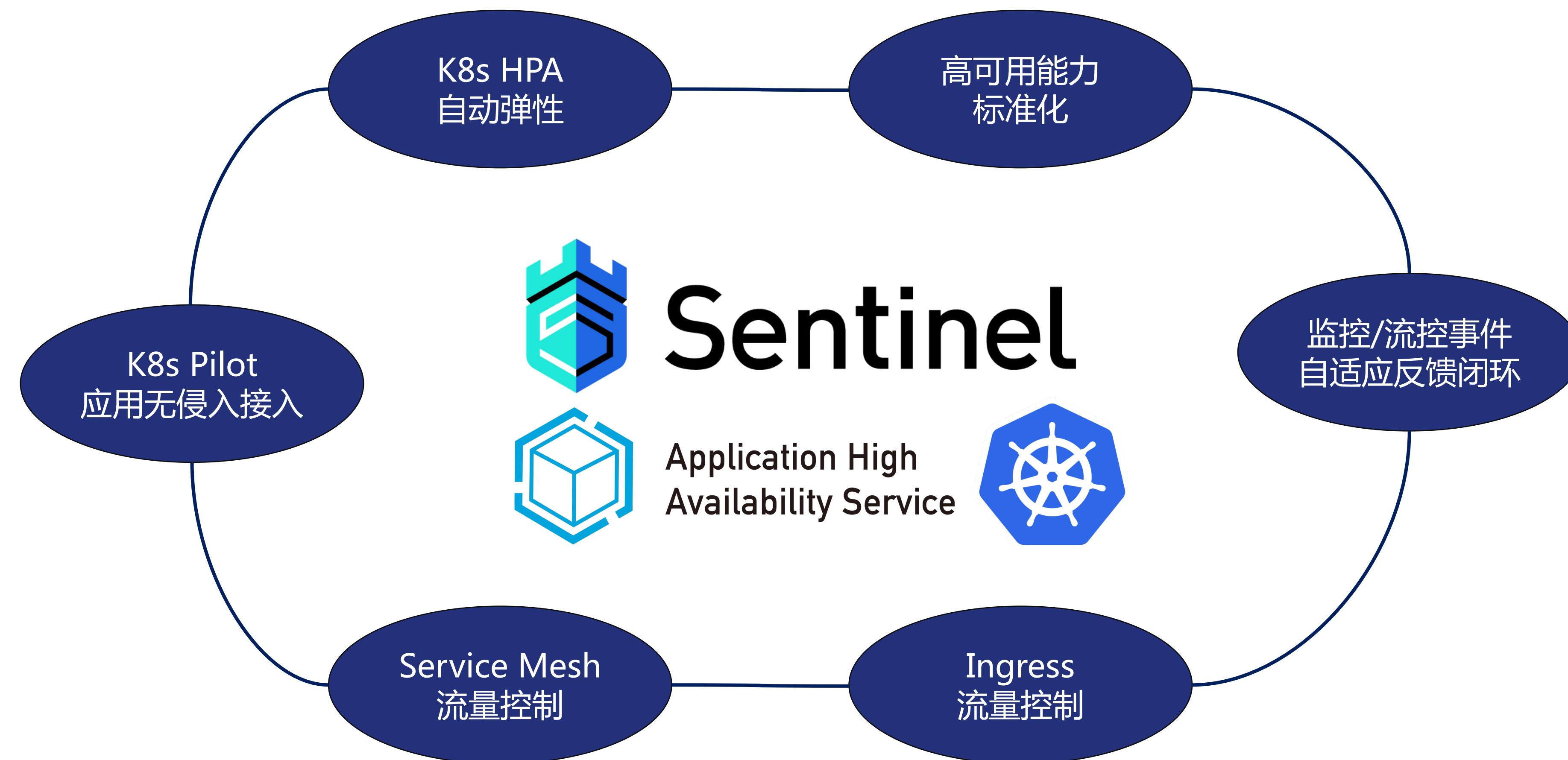
kubectl apply -f flow.yml

# 高可用的另一种思路 - 弹性



**局限：**适合于启动快的无状态服务；  
无法解决非本服务容量的问题（如依赖的 DB 容量不够）

# 云原生时代的高可用防护



阿里云 AHAS 高可用防护: <https://ahas.console.aliyun.com/>

01

微服务可用性  
的场景

02

Sentinel Go  
核心场景实践

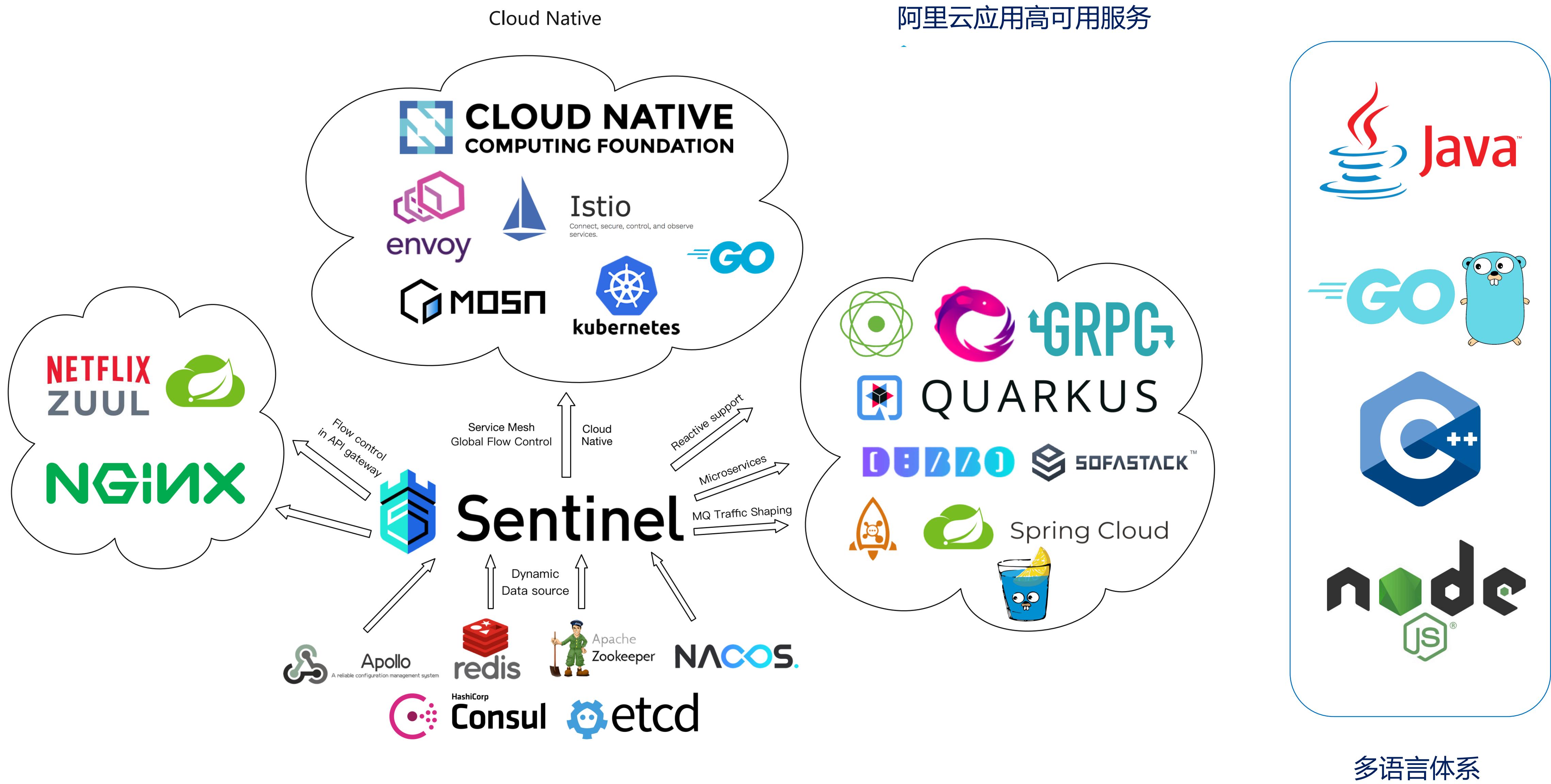
03

云原生时代的  
高可用防护

04

Roadmap &&  
Community

# Sentinel 生态大图



多语言体系

# Roadmap

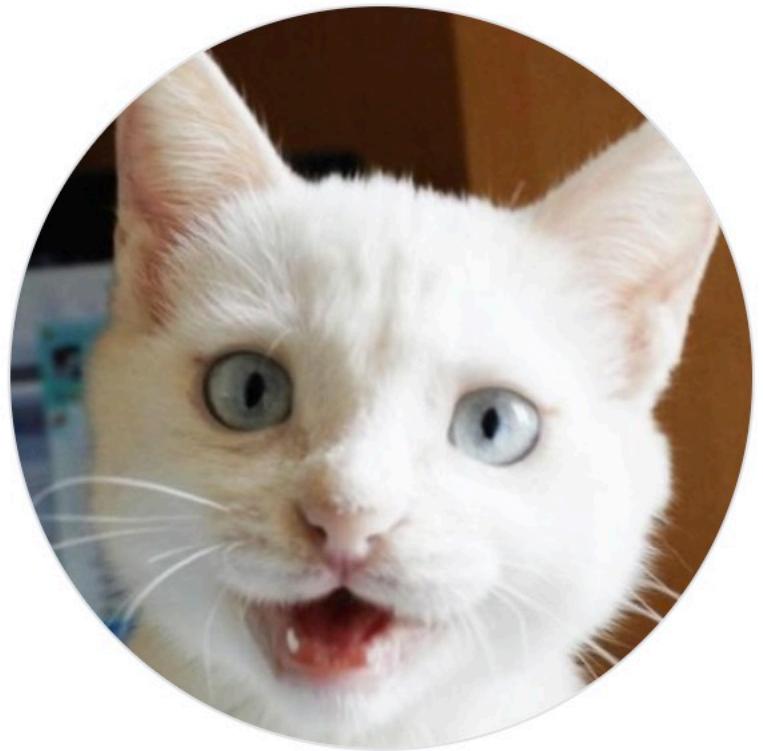


云原生

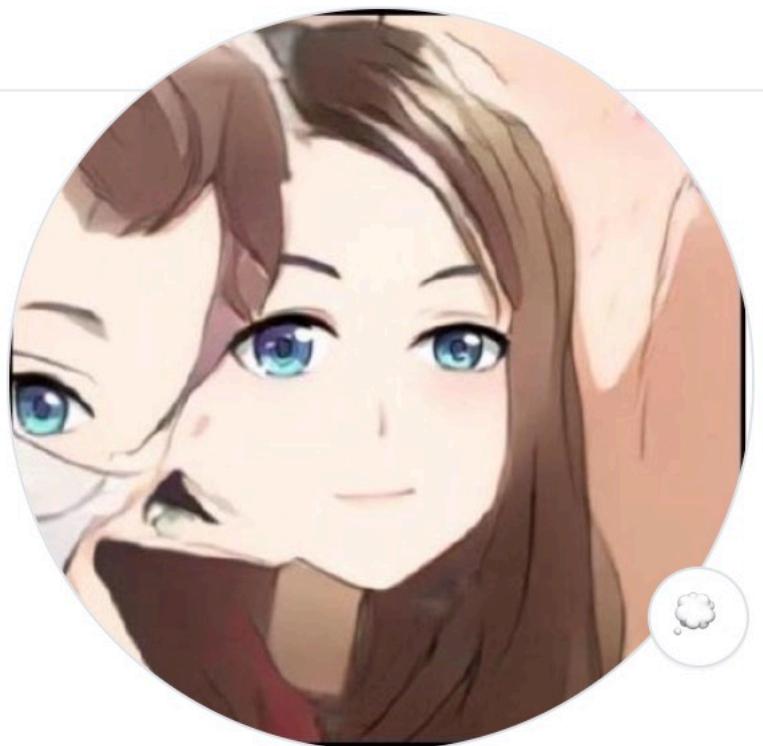


自适应、智能化

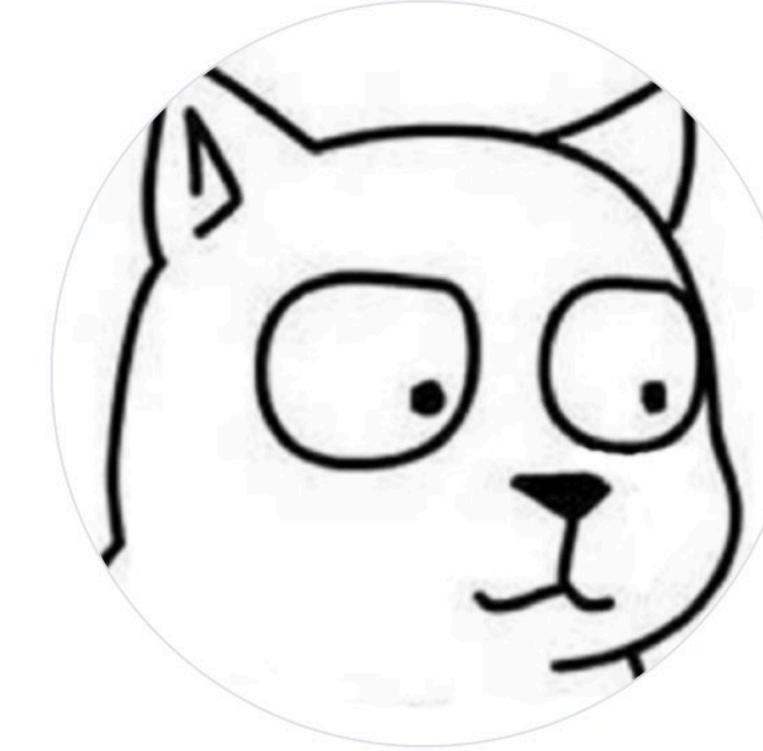
# Our Contributors/Committers



**Jason Joo**  
jasonjoo2010



**louyuting**  
louyuting



**于玉桔**  
zhaoyuguang



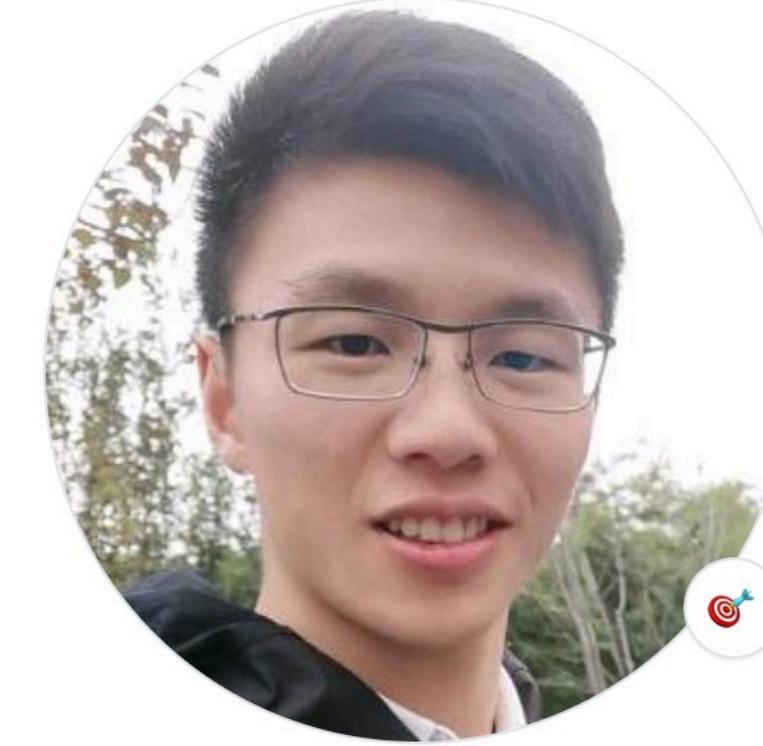
**binbin.zhang**  
sanxun0325



cdfive



**Lin.Liang**  
linlinisme

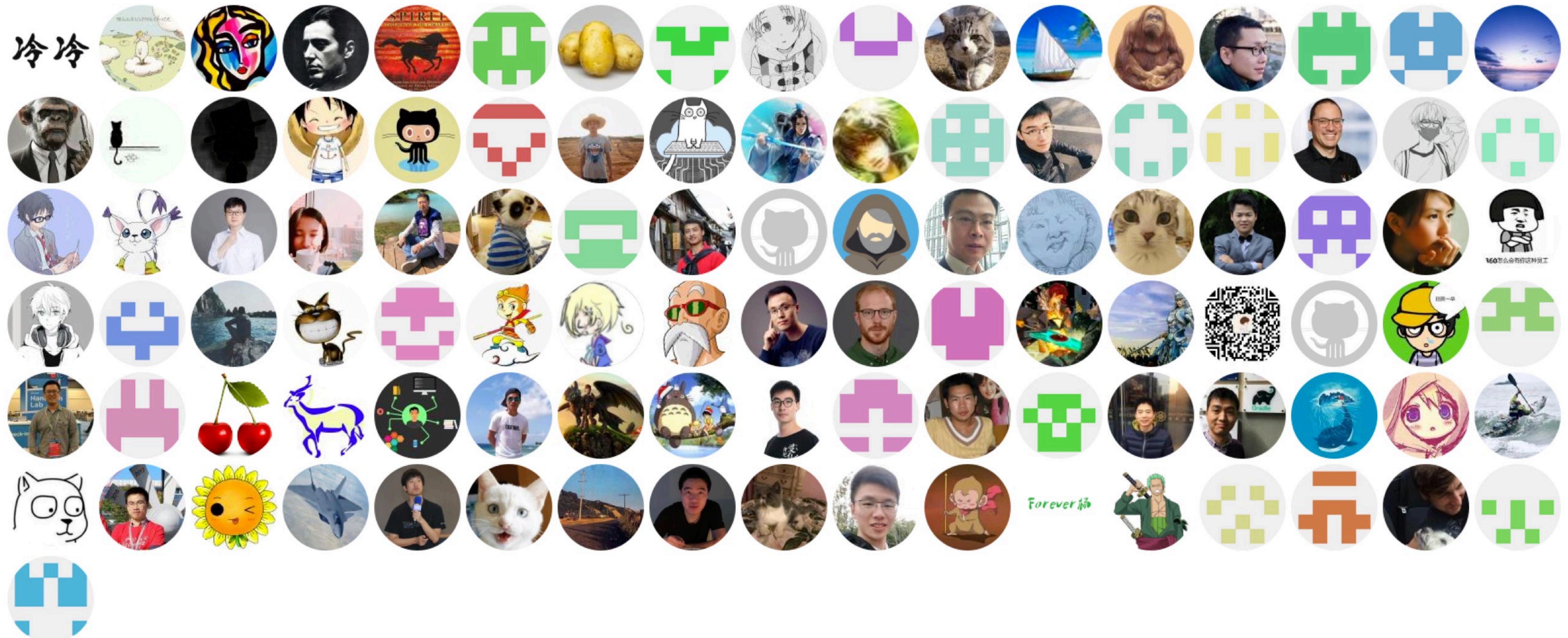


**ZhiQiang Gu**  
yunfeiyanggzq



luckyxiaoqiang

# Our Contributors/Committers



# From zero to contributor

## 参与贡献的途径

- GitHub 上面挑选 issue / 提出新的 idea 进行贡献
- 参与 **核心贡献小组**，认领任务
- 一起 review 社区提交的 PR

## 激励

- 技术成长
- 贡献者激励
- 核心贡献者成为 **Committer**

Now start hacking!

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with **help wanted** or **good first issue** ← pick your issue here

Pinned issues

Wanted: Who is using Sentinel #18 opened on 1 Aug 2018 by sczyh30 贡献指南

How to contribute | 如何参与开源贡献 #391 opened 7 days ago by sczyh30

Filters is:issue is:open search issue here Labels Milestones

78 Open 215 Closed

sentinel 客户端的向控制台提交的 和端口 能否支持 spring boot 的配置文件, 而不是使用 JVM参数 kind/question 2

How to do persistence for parameter flow rules and cluster config kind/question 1

Add id field or extended identifiers in flow rules kind/enhancement

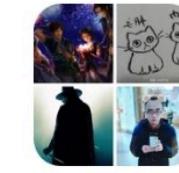
sentinel-dashboard 的监控数据持久化扩展方式希望可以更优雅一些 area/dashboard 1

New issue

# Contact Us

**Sentinel开源讨论群-2群**

999人



扫一扫群二维码，立刻加入该群。

**Sentinel Golang 讨论群**

220人



扫一扫群二维码，立刻加入该群。

**Sentinel 开源3群（临时...）**

52人



扫一扫群二维码，立刻加入该群。

- Sentinel: <https://github.com/alibaba/Sentinel>
- Sentinel Go: <https://github.com/alibaba/sentinel-golang>
- 企业用户欢迎登记：<https://github.com/alibaba/Sentinel/issues/18>

</> </> </> </>

+++

# Spring Cloud Alibaba Meetup

杭州站 · 双11特别活动

# Thanks

---

# 谢谢观看！

+++

</> </> </> </>