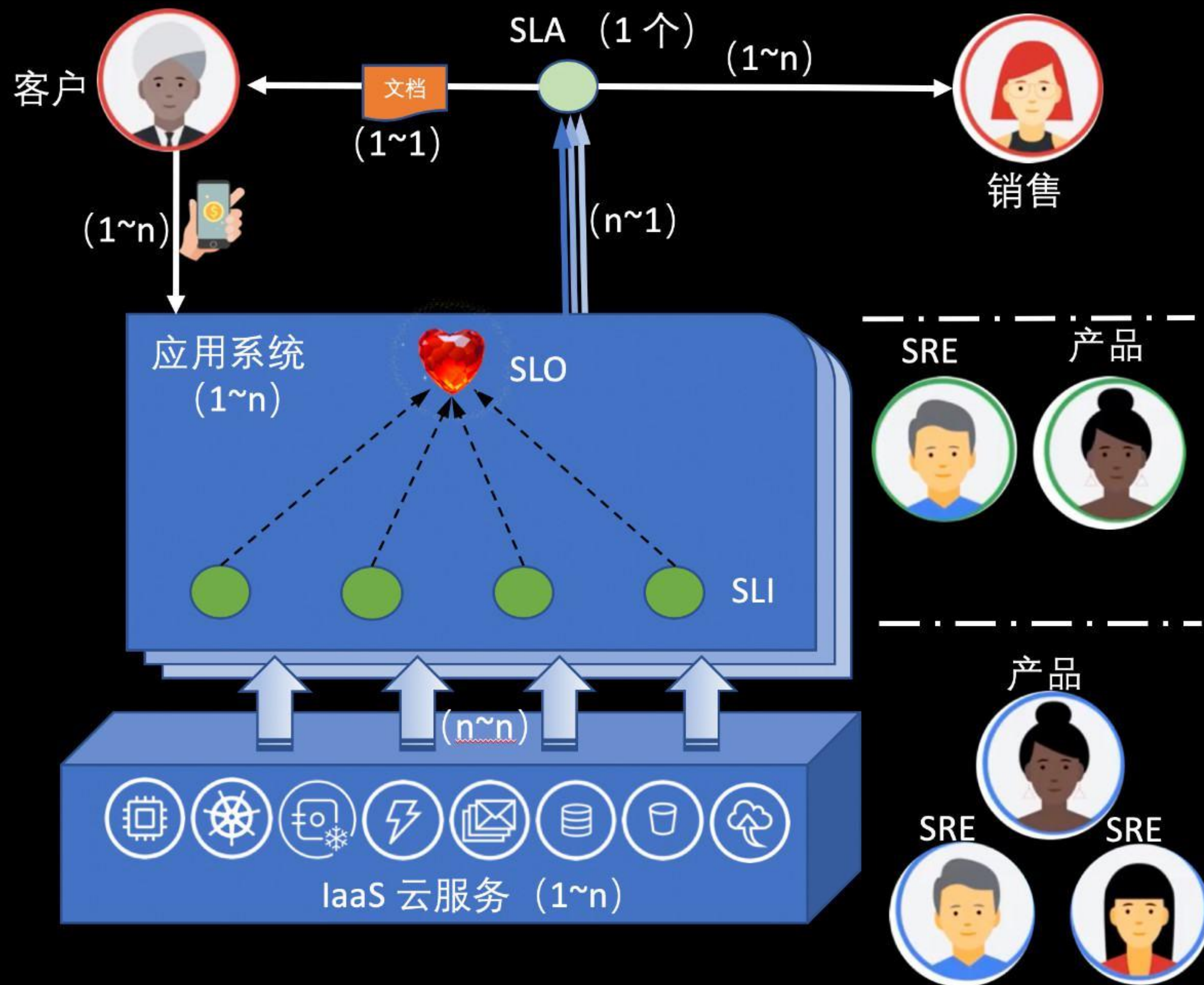


# SLO 实战 7 步法

## 基于系统边界的API服务



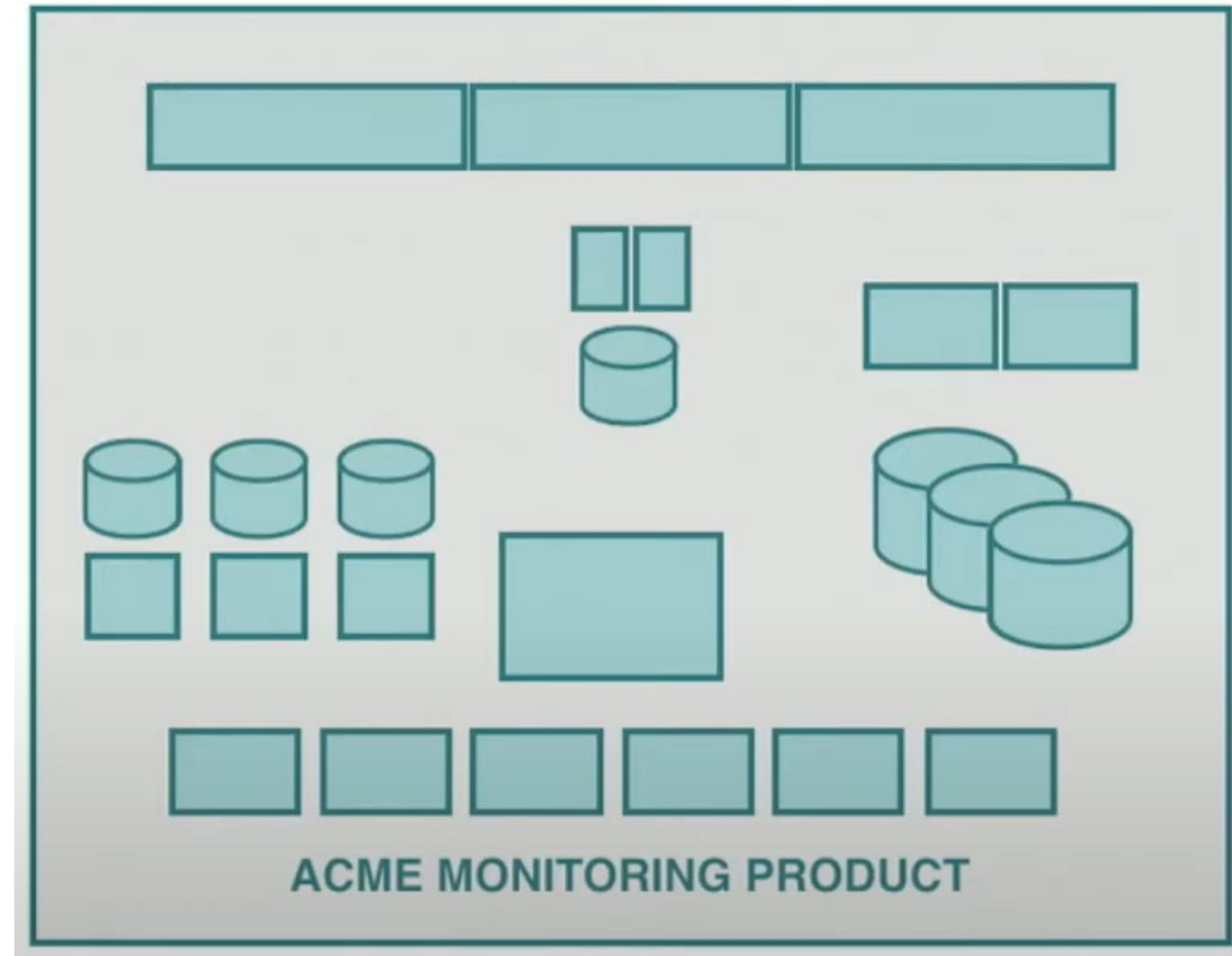




# 示例系统

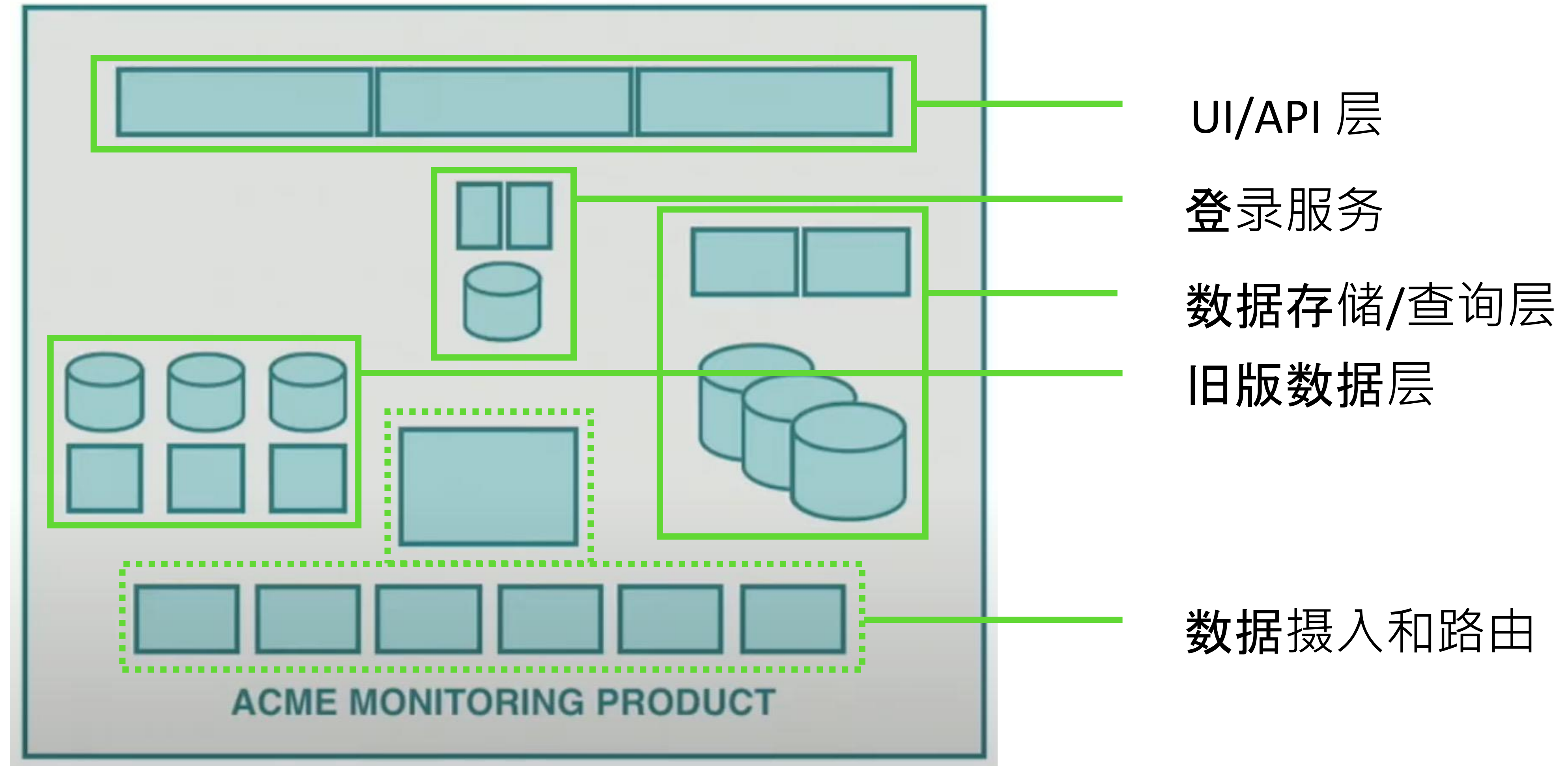
## SaaS版的监控产品

- 数据被正常收集，客户可以在99.9%的时间内登录系统并查看自己的数据.





# 系统边界



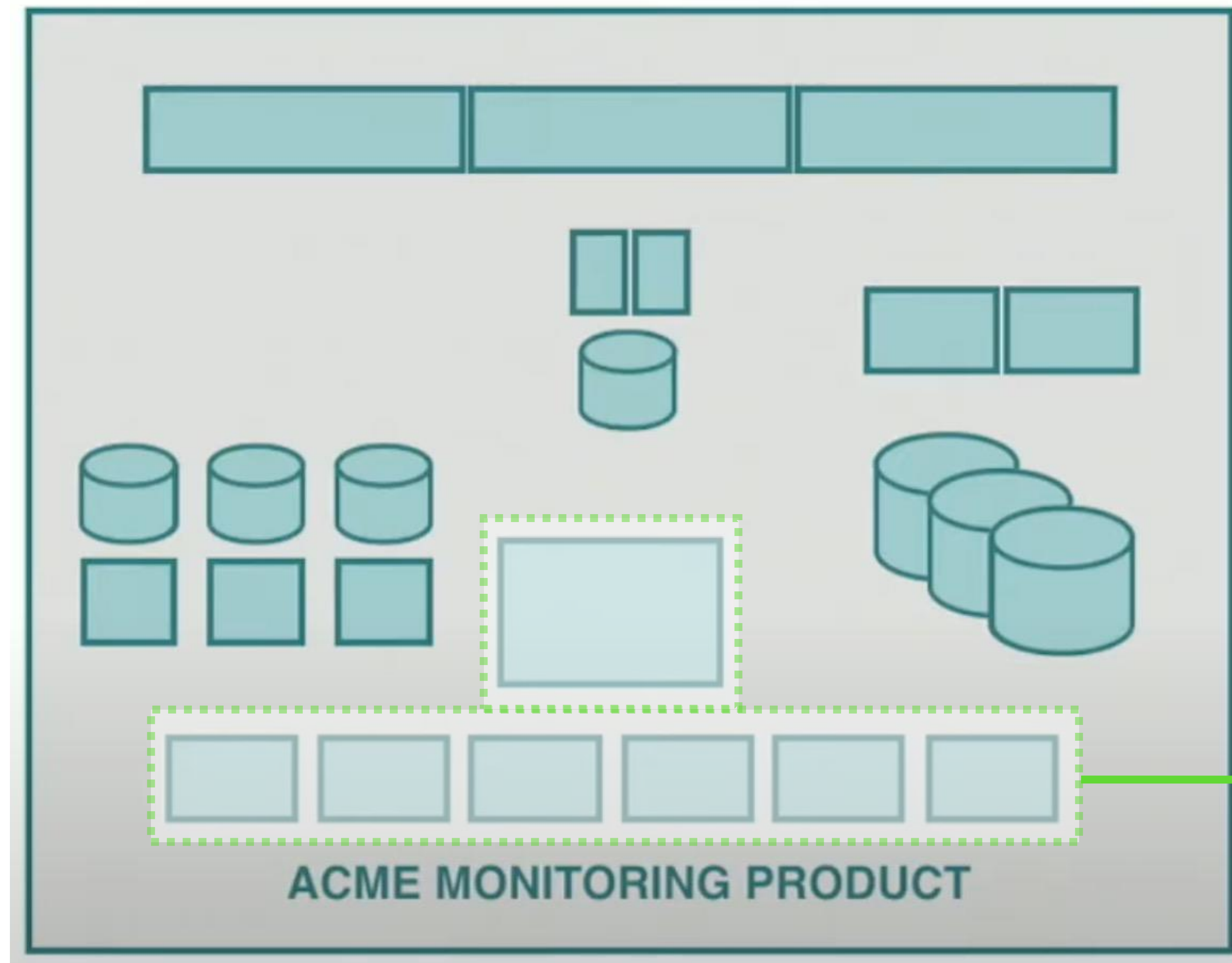
# SLO 实战 7 步法

一个简洁攻略：SLI + SLO



# 系统能力驱动的 SLI

## 唯一或多个系统级能力



数据摄入和路由

多项子系统能力

- 数据摄入
- 数据路由

# 每个系统能力确定一或多个SLI

高效、快速地摄入监控数据

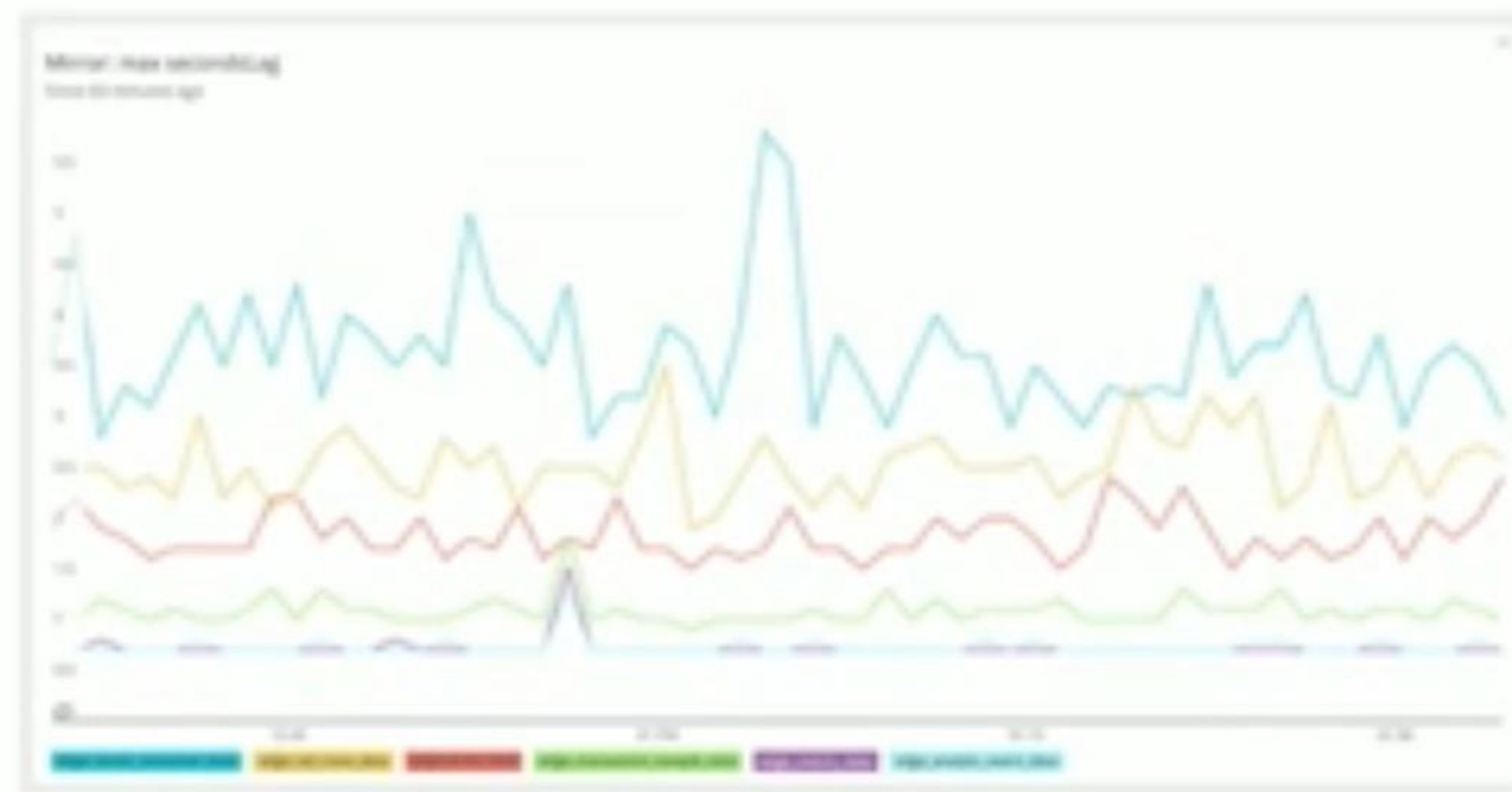


数据摄入 SLI

接收到格式良好的有效数据包的百分比

数据摄入 SLO

99.9%



数据路由 SLI

将信息送达正确目的地的时间

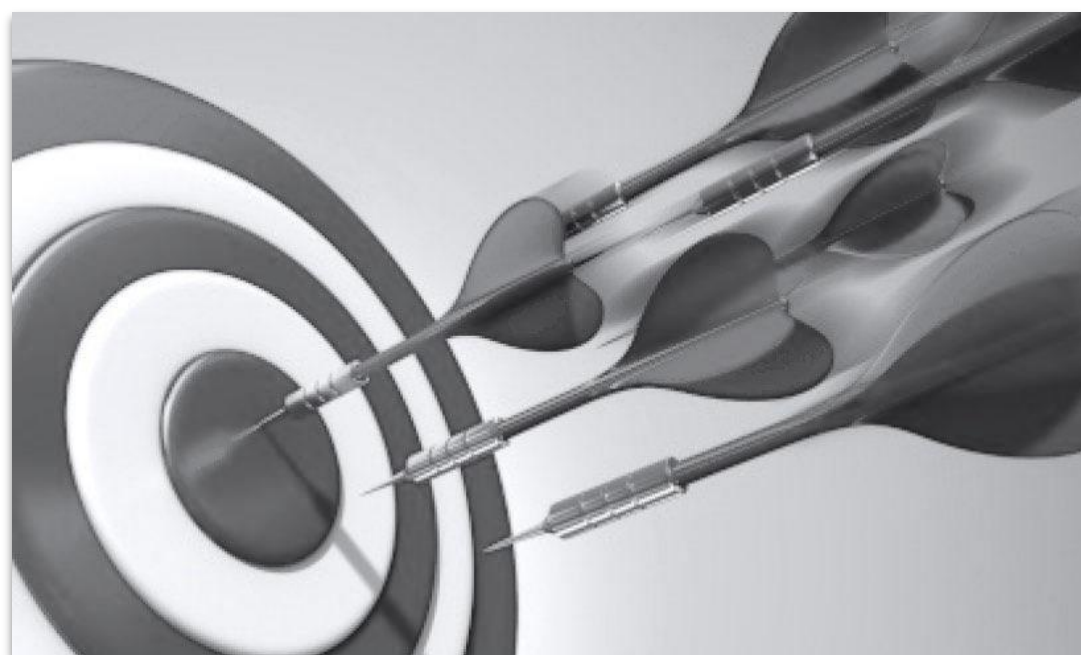
数据路由 SLO

99.5%的消息在 5 秒钟内送达



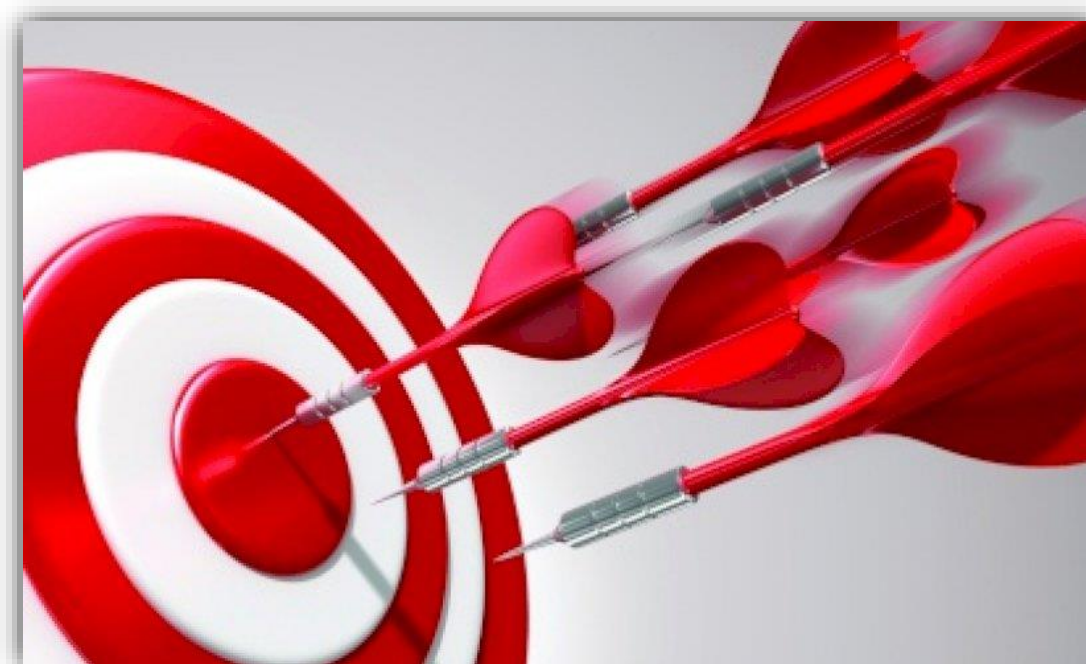
# 选择 SLO 目标

‘有的放矢’的原则应该如何执行



**SLO 的数值应该是：**

- 团队事实上承诺支持到的
- 组织事实上承诺支持的
- 反应出技术上的现实



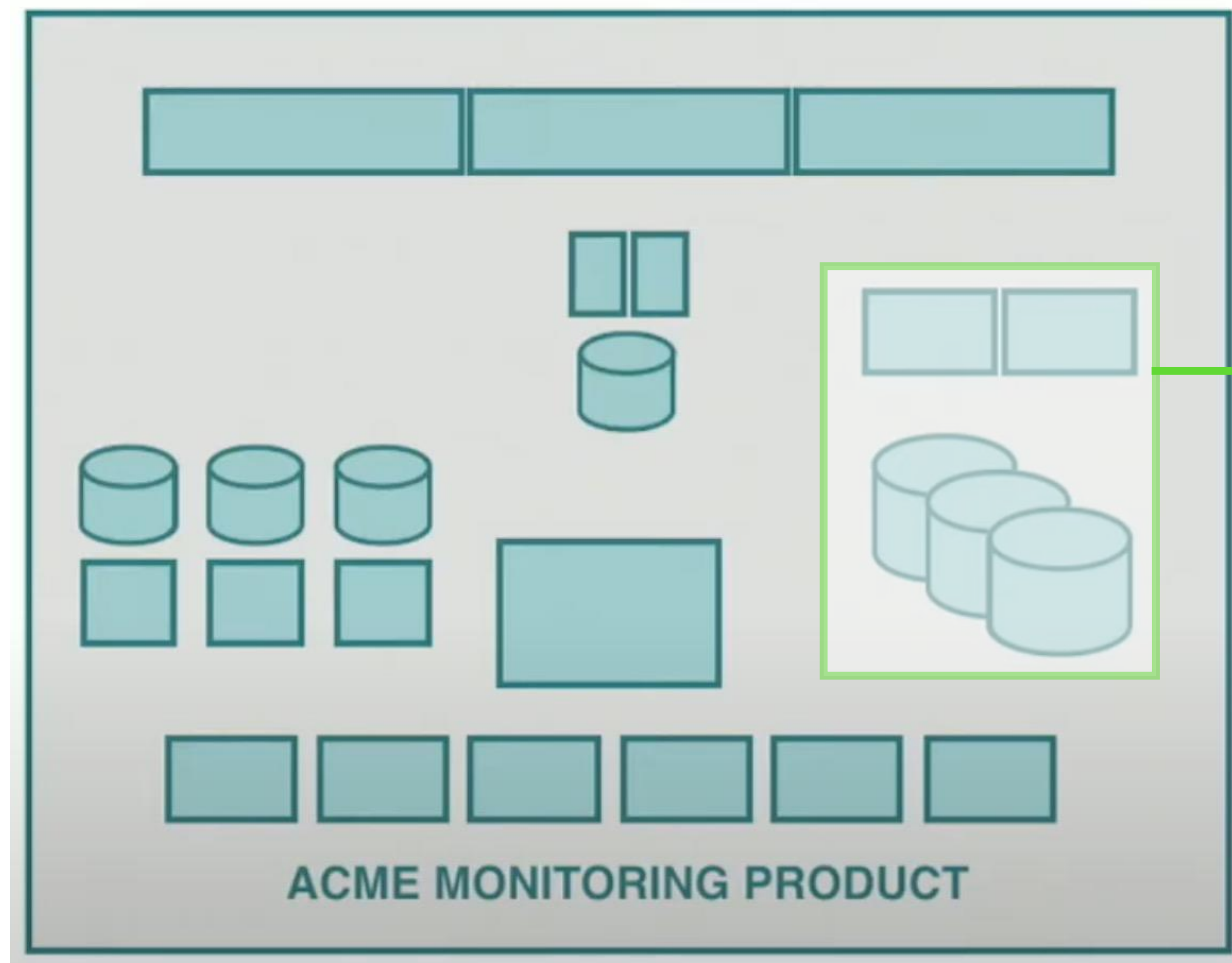
**SLO 代表着现行的  
承诺！**



**当对其不确定/怀疑时，  
先度量之**



# SLI 在广义上代表着可用性



**水平扩展的数据存储层**

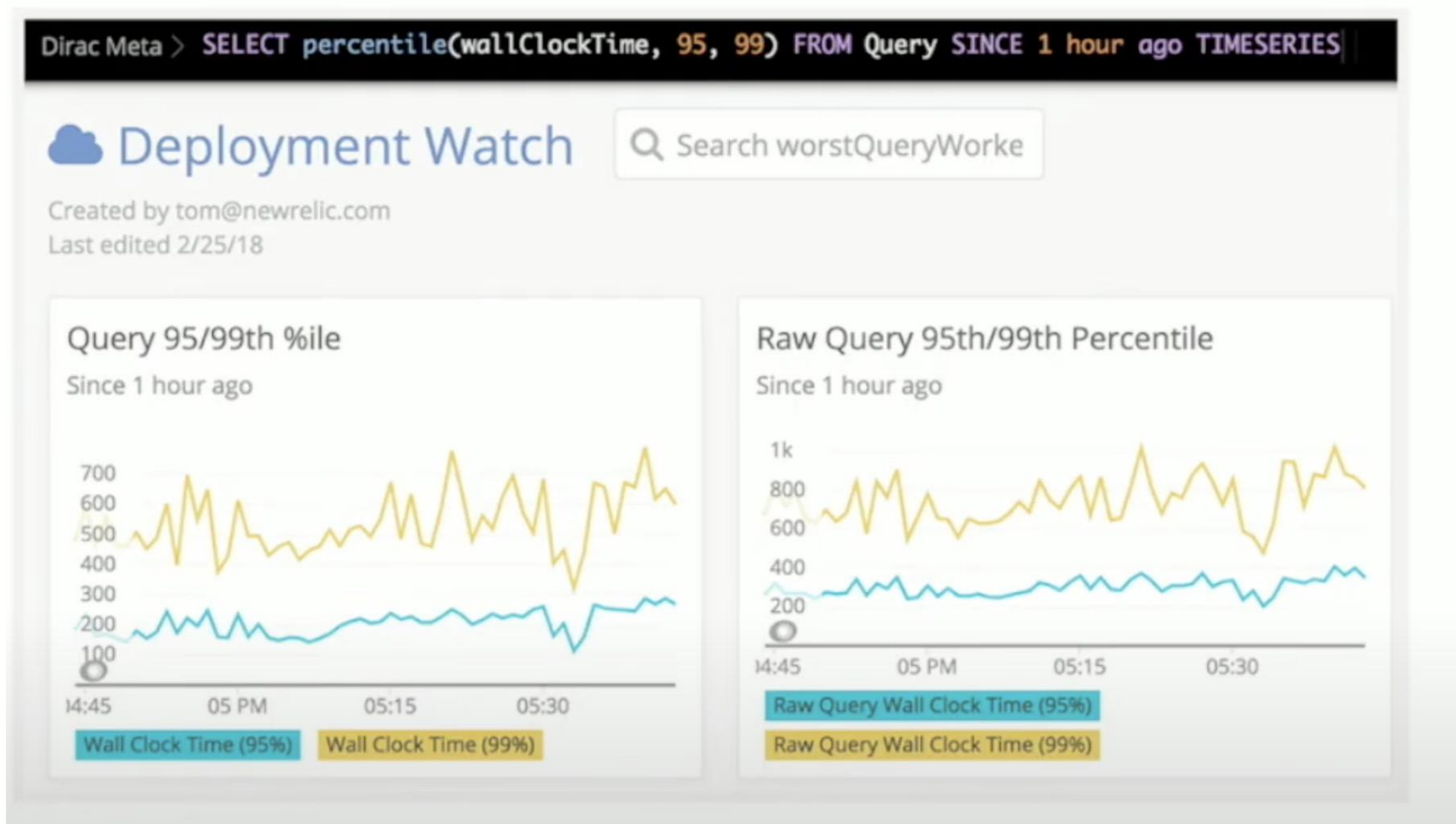
单一能力：查询数据

多个 SLI

- 延迟
- 正确率/错误率

# 程序埋点监控查询响应速度

平均查询时间、P99 都不合适





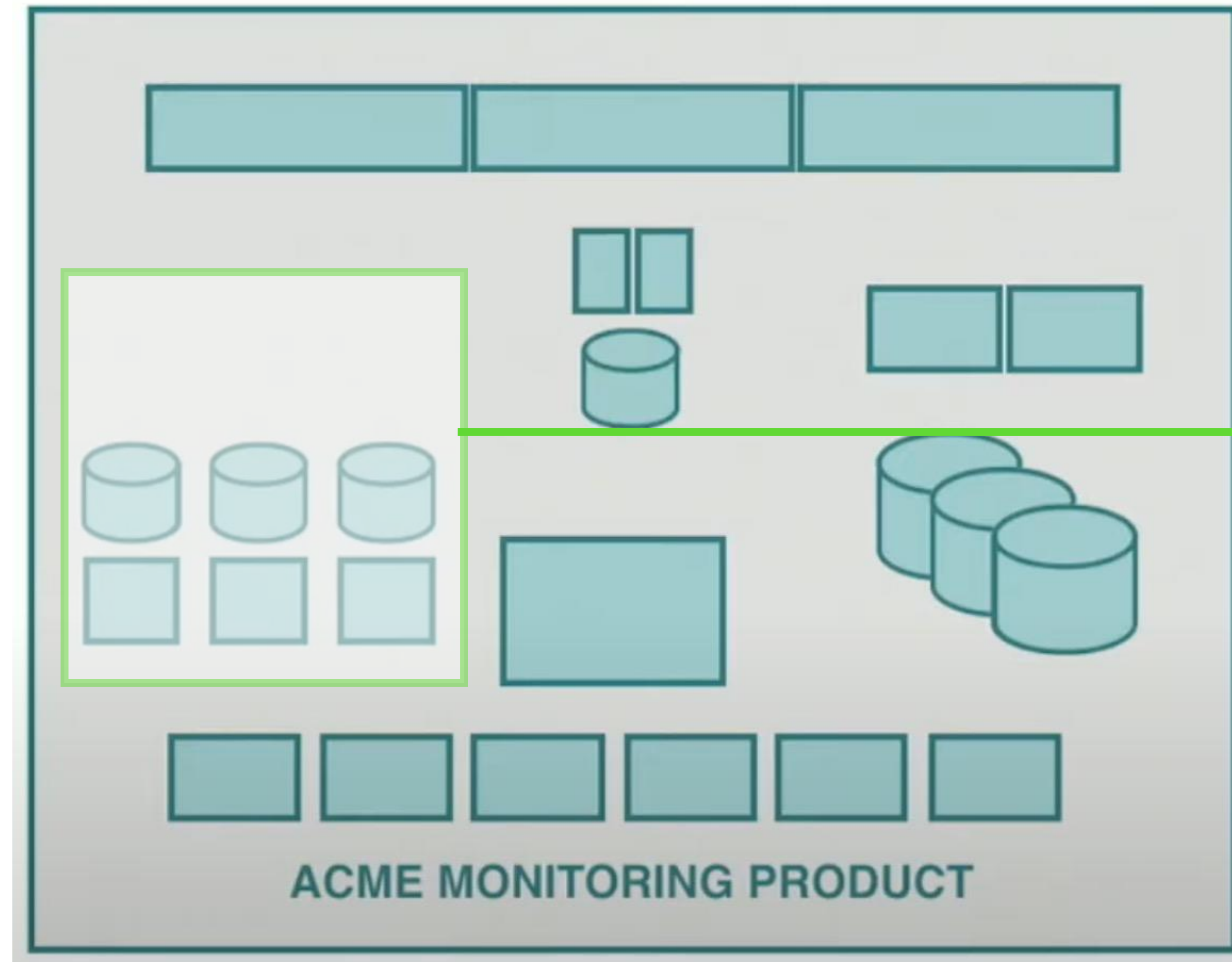
# 复合型 SLO

99.95% 的有效查询得到了正常的查询结果。

99.9% 的查询将被响应的时间小于 1000ms。

99.9% 的有效查询得到了正常的查询结果的时间小于 1000ms。

# 硬分片系统的 SLI 和 SLO



## 硬分片的旧数据库

单一能力：查询性能

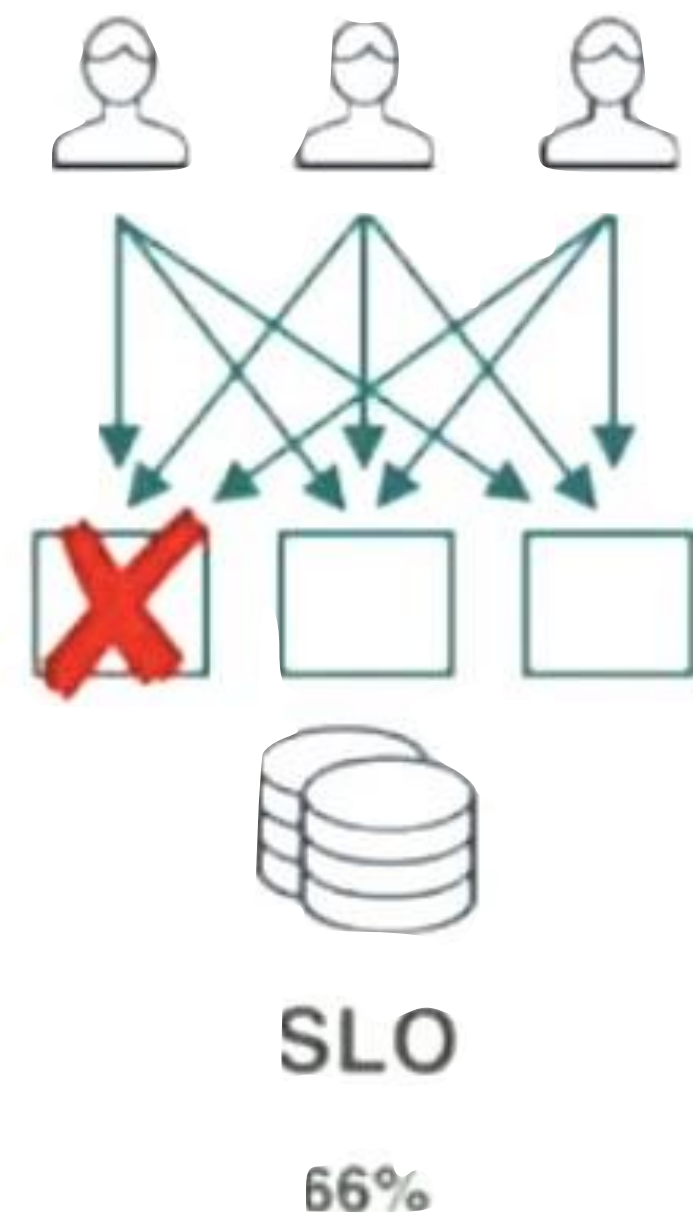
多个 SLI

- 延迟
- 正确率
- 新鲜度

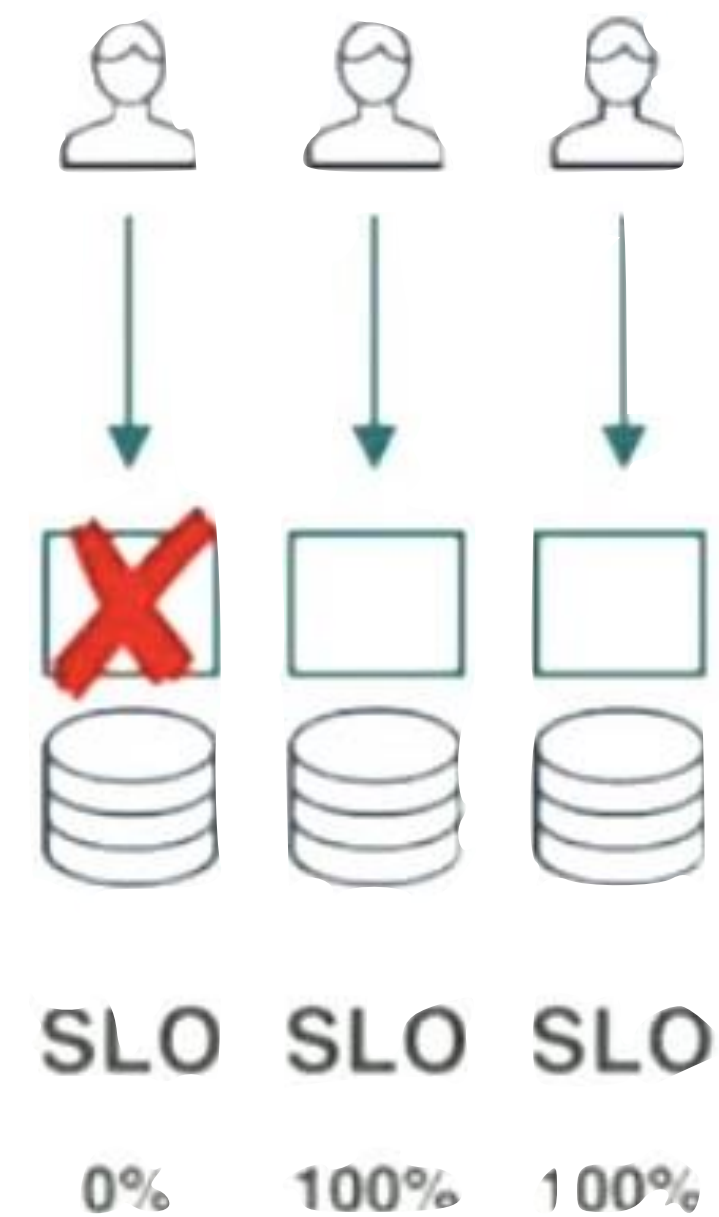


# 单一系统实例和多系统实例的 SLO

集体降级 - 单个宕机，整体度量和独立度量



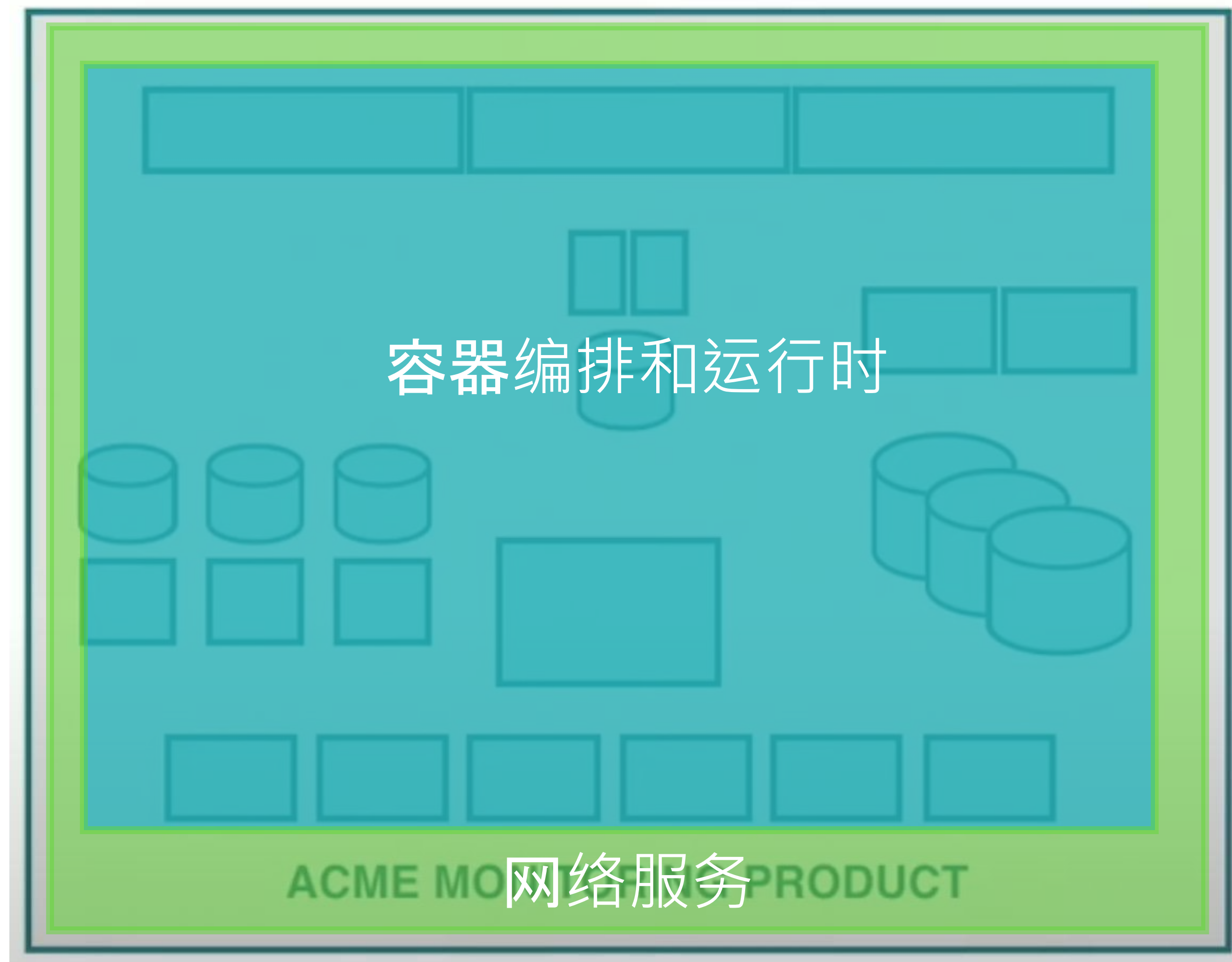
水平的切分



垂直/硬切分

# 为核心基础设施定义 SLI/SLO

逐层级的落实下去





# 客户分组讨论

## 基础服务的使用者和提供者

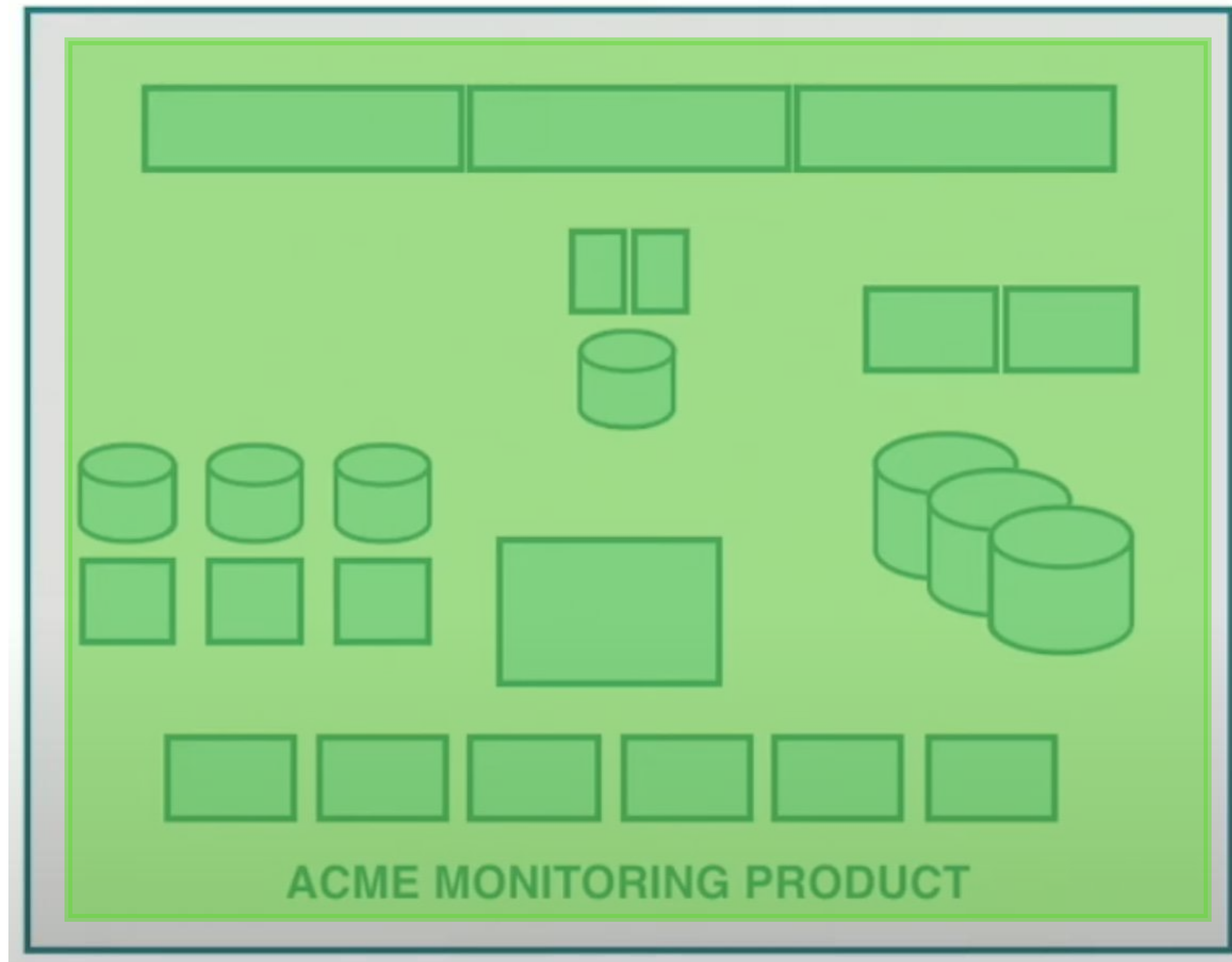
你用这个系统是做什么的？

你希望达到怎样的保证？

你的代码运行在什么假设条件之下？

# 为网络服务定义 SLI/SLO

## 虚拟网络 — VPC



### 多种能力

- 负载均衡
- 基础设施级别路由
- AZ 级别路由

### 多个 SLI

- 负载均衡端点的 Uptime
- 基础设施网络延迟/丢包
- AZ 网络延迟丢包

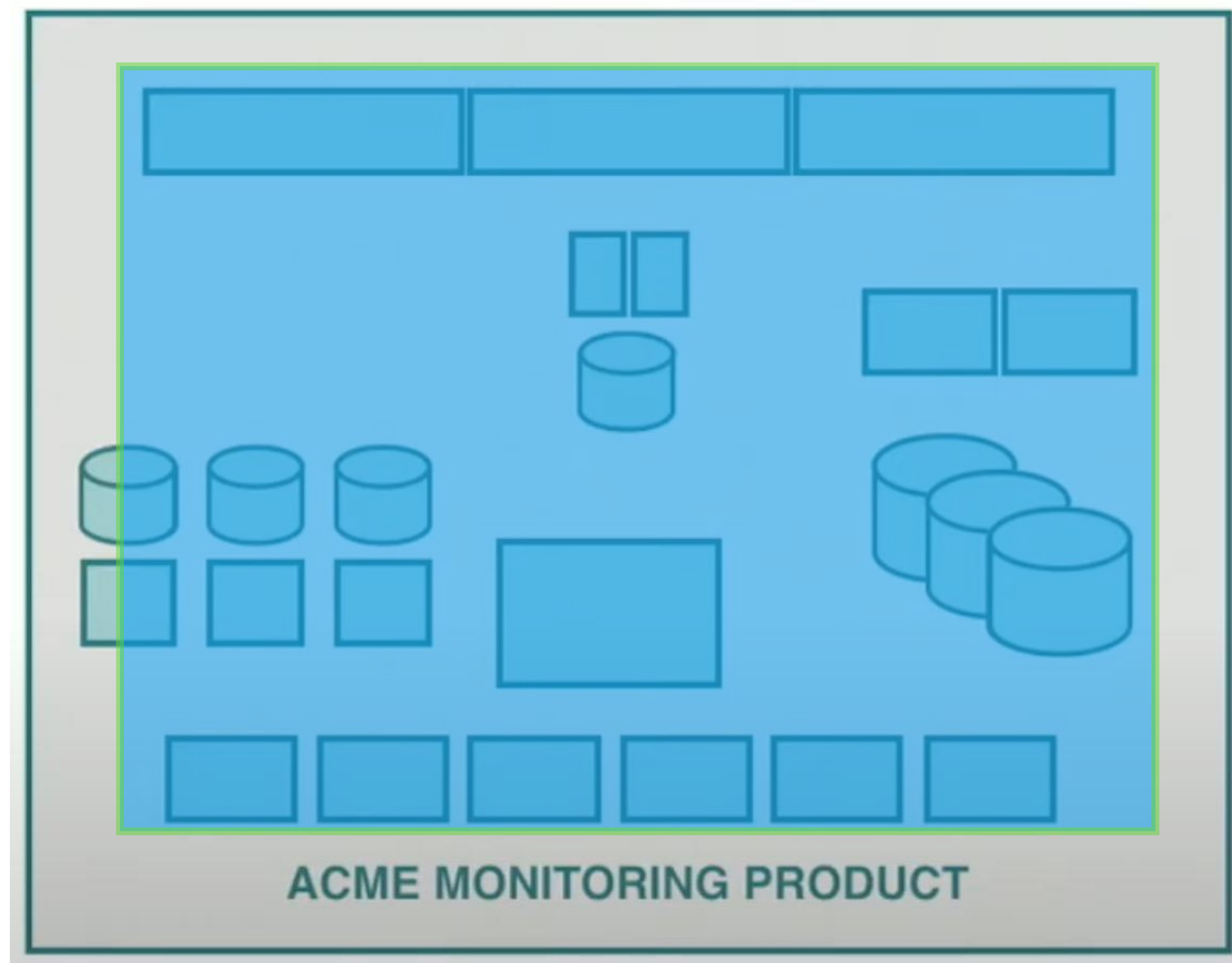
### 每个能力一个 SLO

- 99.99% 目标



# 容器交付相关的 SLI/SLO

Kubernetes 集群提供的容器编排和运行时环境



## 单一能力

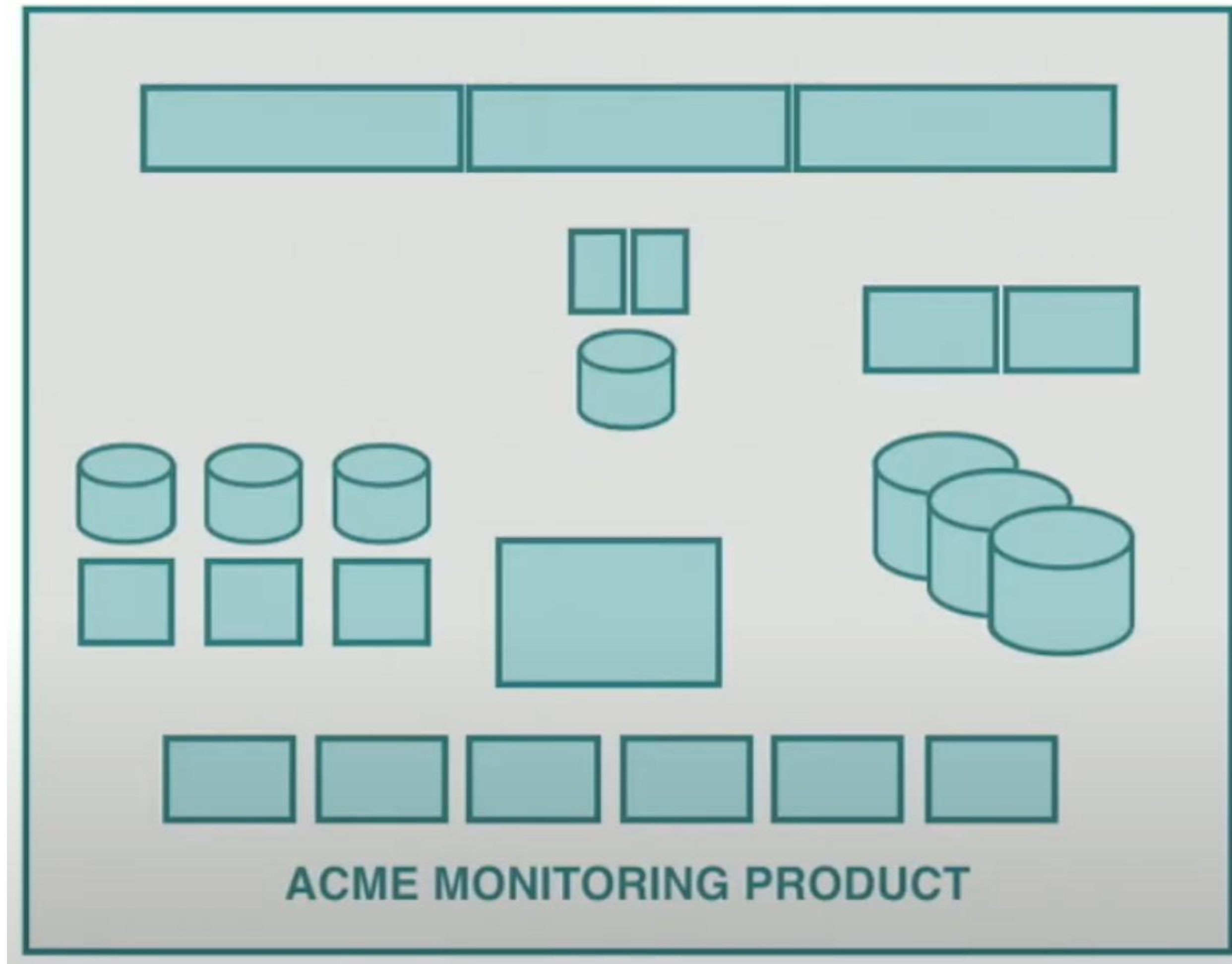
- 作业有预期的资源正常运行

## SLI

- 99.99%的作业在预期的资源下正常运行
- 作业执行的 Uptime 和状态正确率
- 网络服务的饱和度
- 主机的资源的饱和度

# 不要忘了全局服务的可用性

从外到内的黑盒测试



## 多个能力

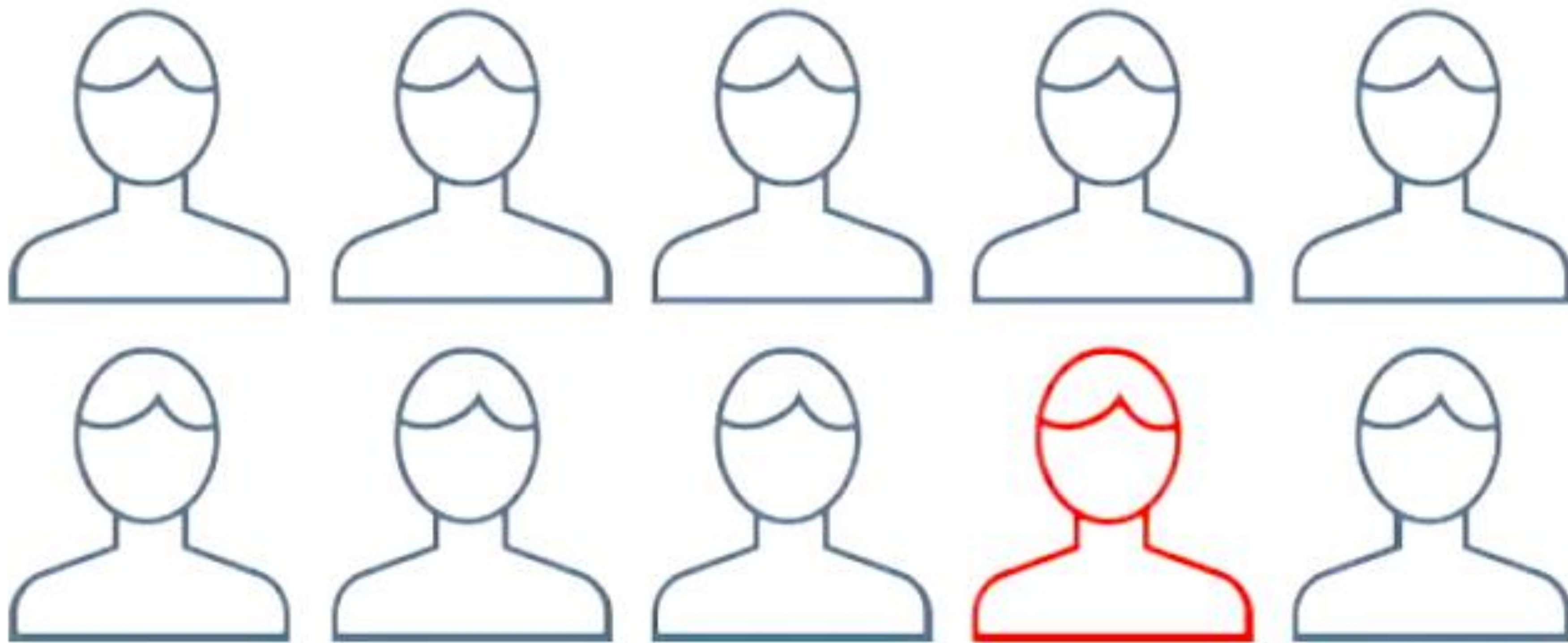
- 收集数据
- 用户登录
- 查看数据

## 傻傻的 1 个 SLI

- 简单的工作流程成功（人工模拟流量）

# 客户特定的 SLO

是否存在人为的分级，客户付费与否的差异







边界切分-定义模块的 SLI/SLO



用人话描述可用性/技术



每一个逻辑实例为一个系统



SLI 高于 SLO



不要把SLI和告警混为一谈



多SLI 可汇聚为单SLO

## 总结



SLO 即现世承诺



文档/分享 SLI/SLO



SLI/SLO 必将与时俱进



关键客户可能需要特供 SLI/SLO



# Google 的 SRE 讲解视频

总共 10 期，见我的微信公众号和 B 站



微信公众号



个人微信号

B 站 <https://space.bilibili.com/477542716/>