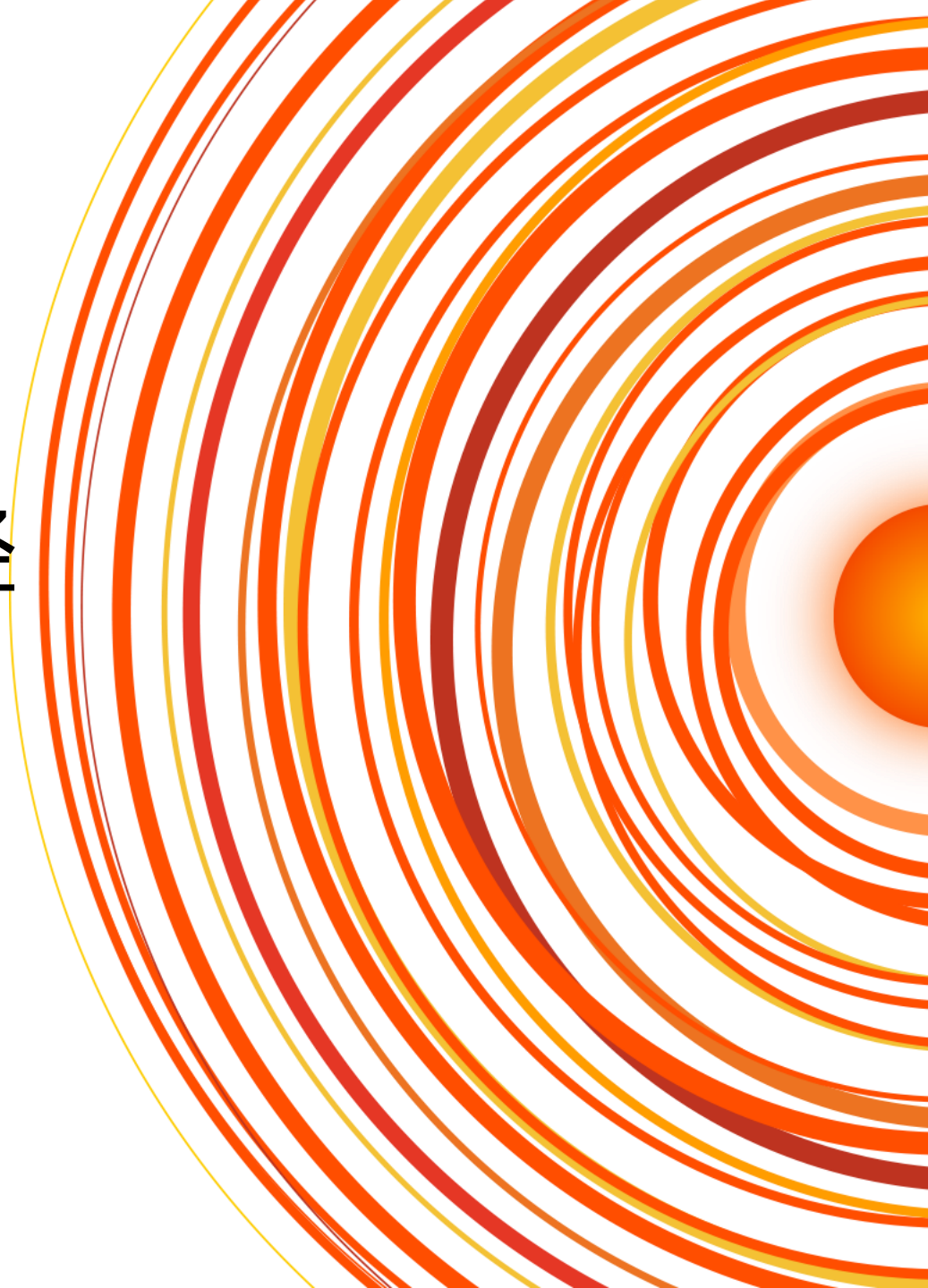
观测云
GUANCE.COM

DevOps团队最佳选择！

系统可观测的价值与实现路径

徐季秋



关于我

观测云
GUANCE.COM

徐季秋@观测云

观测云
GUANCE.COM

目录

contents

CIO们的关键字:

降本增效

数字化转型

云原生

当大家都在用数字去量化业务的时候,谁来量化你的数字化系统本身?

引子 2

观测云
GUANCE.COM

程序员的关键字

我在我的环境跑的好好的，为啥你就不行？

这个问题我重现不了！

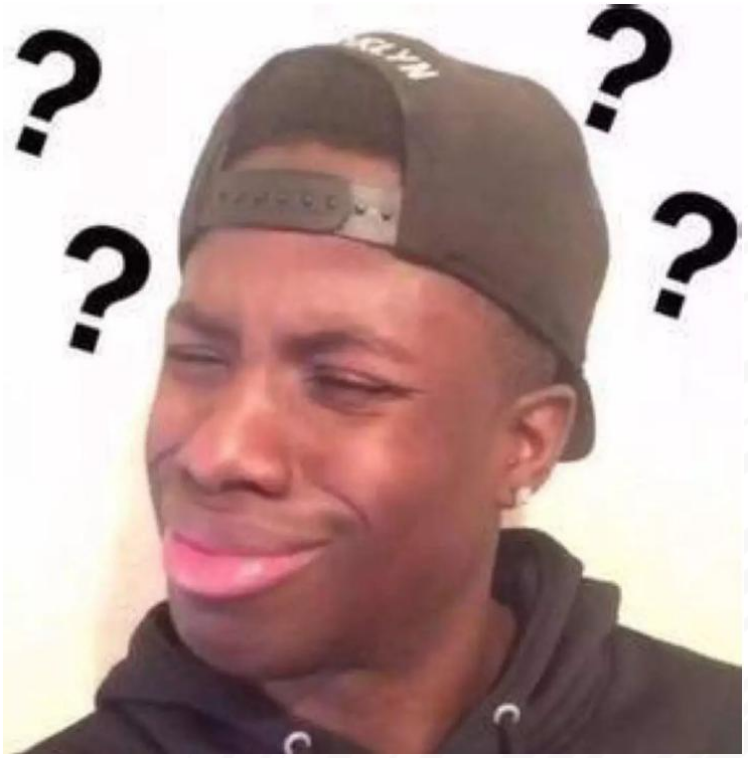
解决问题靠碰运气？

引子 3

运维监控室一片祥和

却收到用户投诉系统不能用了或者太卡了?

给老板解释一下?



解决问题靠大神?

系统可靠性工程 & 系统可观测性

观测云
GUANCE.COM

让我们一起建设系统可靠性和可观测工程吧

系统可靠性工程

工程理论中可靠性工程包括系统可靠性数据的收集、分析，可靠性设计、预测、试验、管理、控制和评价，是系统工程的重要分支.

常用的度量指标主要有可靠度、故障率、平均无故障工作时间和平均故障修复时间等。

SRE

Site Reliability Engineering

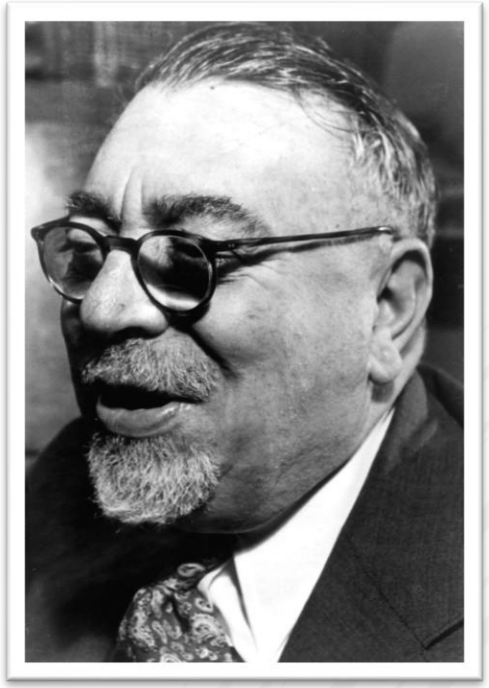
- Define availability
- Level of availability
- Plan in case of failure

系统可观测性

观测云
GUANCE.COM

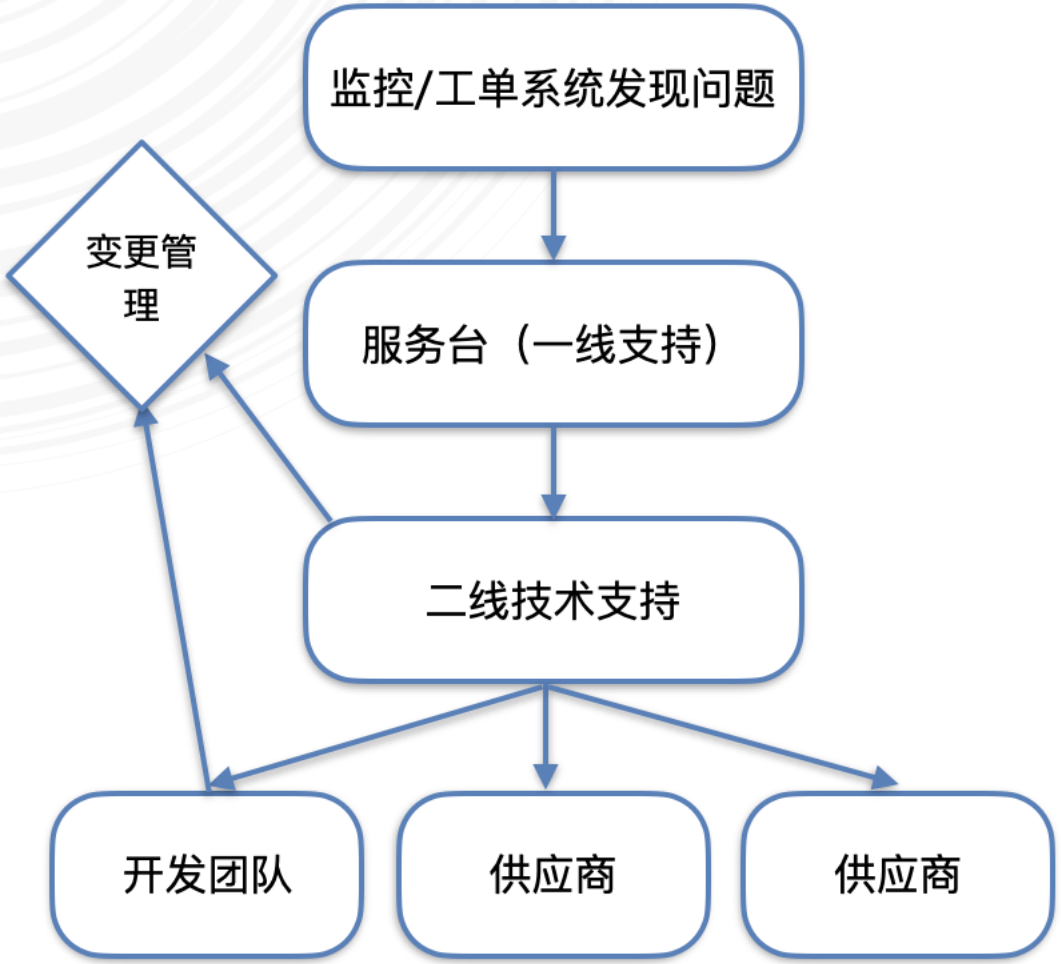
控制理论中的可观测性（Observability）是指系统可以由其外部输出推断其内部状态的程度。

系统的可观测性和可控制性是数学上对偶的概念，可观测性最早是匈牙利裔工程师鲁道夫·卡尔曼针对线性动态系统提出的概念。若以信号流图来看，若所有的内部状态都可以输出到输出信号，此系统即有可观测性。



"我们对一个系统的各种信号越掌握，就对这个系统越有控制力。"

传统的故障处理模型已经失效



- 问题1：监控系统如何定义哪些情况是Incident？过多的incident会导致后端工作量大，只关注资源的incident会导致业务系统问题而无法触发告警。
- 问题2：一线支持工程师能力有限，二线支持如果仍然处理不了就需要继续上升到开发团队。导致了MTTR（平均恢复时间）完全不可控。
- 问题3：当一个故障被确定并定位，可能需要走一个变更管理，如何判断这种变更是不是会导致新的问题，或者谁来判断这个变更有效？
- 问题4：在这种模式下，实际上是一种责任切割模式，大家只为自己的职责范围内的事情负责，容易形成甩锅文化。

结论：这是传统IT时代的方案，特点是内部办公系统对于SLA要求不高，系统不更新或者较少更新。而如果是面向互联网，面向企业最终用户的系统，已经不适用这种类型的办法了。

新模型

观测云
GUANCE.COM

指标 SLI

该服务的某项服务质量的一个具体量化指标。如果触发，会影响SLO。

- 5分钟内的用户P99时平均延迟 > 200ms
- 5分钟内的接口错误率 >1%
- 5分钟内触发了服务不可用

目标 SLO

服务质量的目标、当前状态、以及余额（Budget即犯错空间）

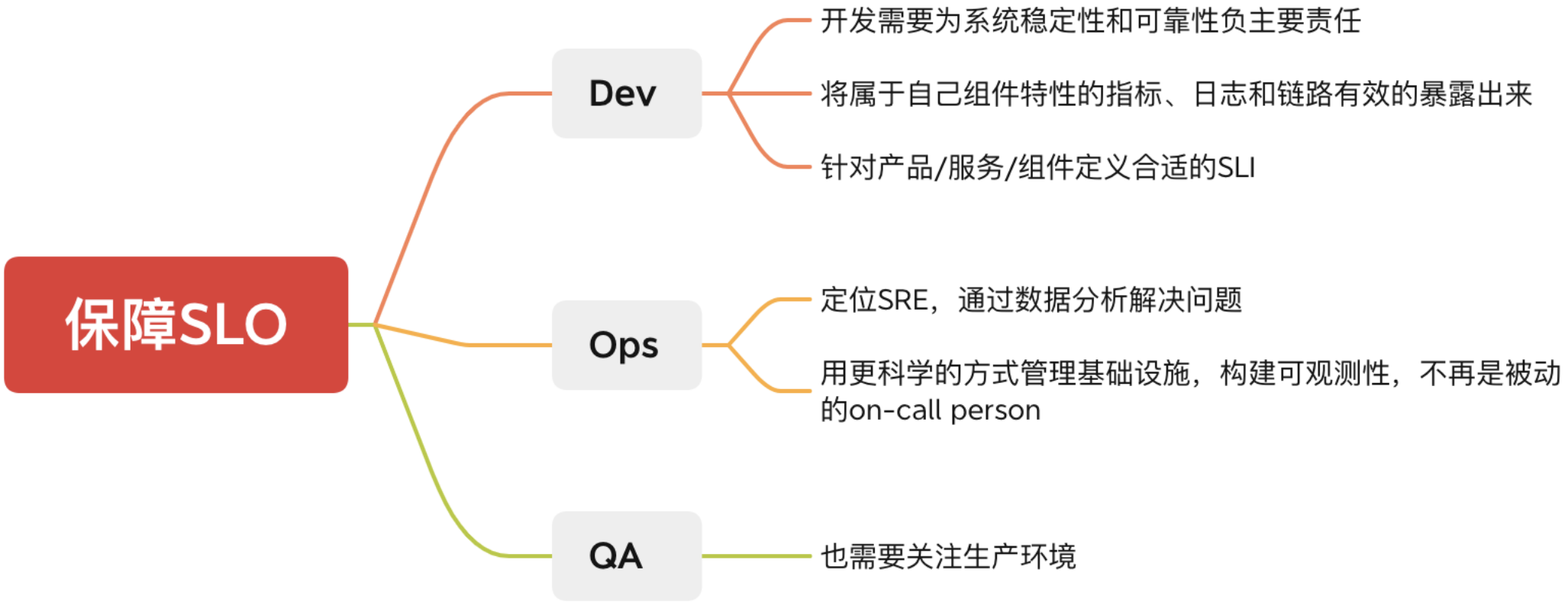
- 目标（30天）：99%
- 当期状态：99.98%
- 余额Budget：98%，xx小时xx分xx秒

协议 SLA

是对于自身业务可靠性向使用的承诺。

- 如果SLO没有达到时，有什么后果？
- 财务方面的（例如罚款）或其他类型

各个团队的责任


保障SLO
Dev
开发需要为系统稳定性和可靠性负主要责任
将属于自己组件特性的指标、日志和链路有效的暴露出来
针对产品/服务/组件定义合适的SLI
Ops
定位SRE，通过数据分析解决问题
用更科学的方式管理基础设施，构建可观测性，不再是被动的on-call person
QA
也需要关注生产环境

SRE实施：构建系统的可观测性

改变对于监控系统的看法，分布式分层次构建整体系统的可观测性（Observability）

与传统的监控使□□式不同，构建系统可观测性有以下□个特点

- □标不同，传统监控的□标是监控资源或者估值，□可观测性□标是保障SLO，监控的是可能影响SLO的SLI。
- 开发者要深度参与到可观测构建中，开发者要主动上报数据（指标，□志，链路），并且基于模块特点构建相应的仪表盘
- 要具备强□的分析钻取能□，有能□分析到任何可能影响到SLO的问题细节
- 使□者为了避免触及SLO budget扣分，因此需要经常性主动观测分析整个系统，及时发现□产系统中的问题，□不是被动On Call

可观测性相关的基本概念

观测云
GUANCE.COM

- 指标（Metric）：指标是连续时间下的系统的值的记录，基础指标通常用于描述两种数据类型，一种是计数（Count），一种是计量（Gauge）。
- Tag：每个指标均需要标记不同的Tag，以构建不同维度的数据，方便相关性的分析所需要的聚合与分组。
- 时间线：不管指标上有多少Tag，唯一连续记录的基础时序数据就称为一条时间线。
- 日志（Log）：系统/应用输出的时间相关的记录，通常由系统/软件开发人员输出，方便定位系统的错误和状态。
- 日志数据提取：将日志中有关联性的数据提取为Tag或Field，Tag方便进行聚合和分组，Field方便基于计算和聚合。
- 链路（Tracing）：基于有向无环图构建的软件各个模块直接的调用关系。
- OpenTelemetry：开放的可观测性标准，目前包括了OpenMetrics和OpenTracing两个部分，Logging部分正在起草中。OpenMetrics使用得就是Prometheus exporter的标准协议，jaeger是OpenTelemetry最推荐的复合OpenTelemetry规范的分布式追踪系统。
- 指标化：将非指标形态的数据结构如日志，链路实现动态的生成指标的过程，指标化是为了从这些数据进行计算后有效的获得可以用于SLI定义的复合型指标。

复杂系统的可观测性拆解

SLO
Target 99%
Budget：99.98%

SLO
Target 99.5%
Budget：99.98%

- SLI:5分钟内的API延迟>2s
- SLI:5分钟内的三接失败率率>1%
- SLI:kafka消息队列累积任务速度超过了处理速度

应/中间件层

SLO
Target 99.5%
Budget：99.98%

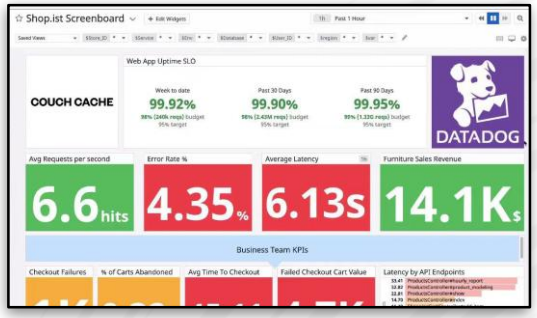
- SLI:集群平均负载超过了90%
- SLI:系统出现了可疑的账号
- SLI:某个核组件整体不可

基础设施层

SLO
Target 99.5%
Budget：99.98%

- SLI:5分钟内的户P99付平均延迟>2s
- SLI:5分钟内的接错误率>1%
- SLI:5分钟内的触发了服务不可

户体验层



复杂系统的可观测性体系设计原则

观测云
GUANCE.COM

尽可能的收集所有组件系统的所有相关□ 的基础数据：组件包括云，主机，容器，kubernates集群，应□ ，各种终端，相关□ 是指
（性能，□ 络，安全，容量），基础数据包括指标，□ 志，链路，实时收集数据成本并不□ ，但如果没有收集，□ 旦系统故障需要排 查分析的时候，就□ 法有效评估当是的状态。

明确系统可观测性构建的责任：谁是这个组件的构建者，谁负责定义这个组件的SLI，以及负责收集所有的相关基础数据，并且构建 相应的仪表盘以及为相关的组件的SLO负责。

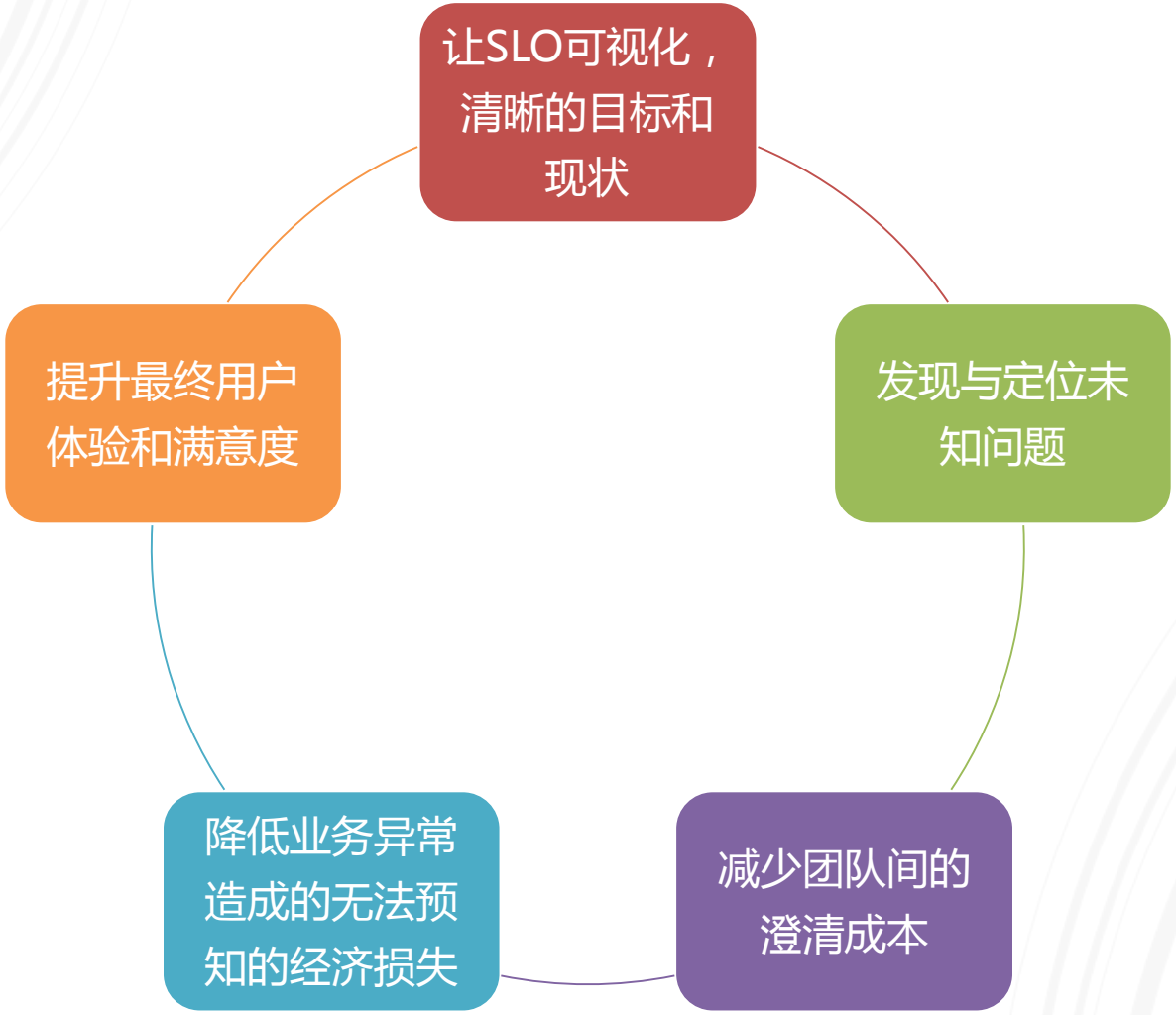
开发者为可观测性负责：开发者要将□ □ 开发的系统的可观测性数据暴露作为软件质量□ 程的□ 部分，如果说单元测试是为了保证最
□ 单元代码的可□ 性，那么开发者标准化暴露可观测性基础数据也将作为□ 产系统可靠性的必要条件。（设想□ 下你正在驾驶□ 架没 有仪表盘的□机）

建□ 统□ 的指标，□ 志，链路规范,统□ 团队的□ 具链：相同的指标命名规范，相同的□ 志格式，相同的链路系统（即使都遵循
opentelemetry标准，仍有不同），定义串联整个系统的统□ TAG规范，如所有错误都是state:error。

持续优化改进整体可观测性：针对整个系统的可观测，包括数据收集，视图构建，TAG体系建□ 均需要时间，不能因为覆盖度或者构 建的仪表盘未能在某次事故中发挥作□ ，□ 回到了继续□ 挫事的□ 式处理问题。每次未被观测的故障都是进□ 步提升可观测范围的绝 佳机会。

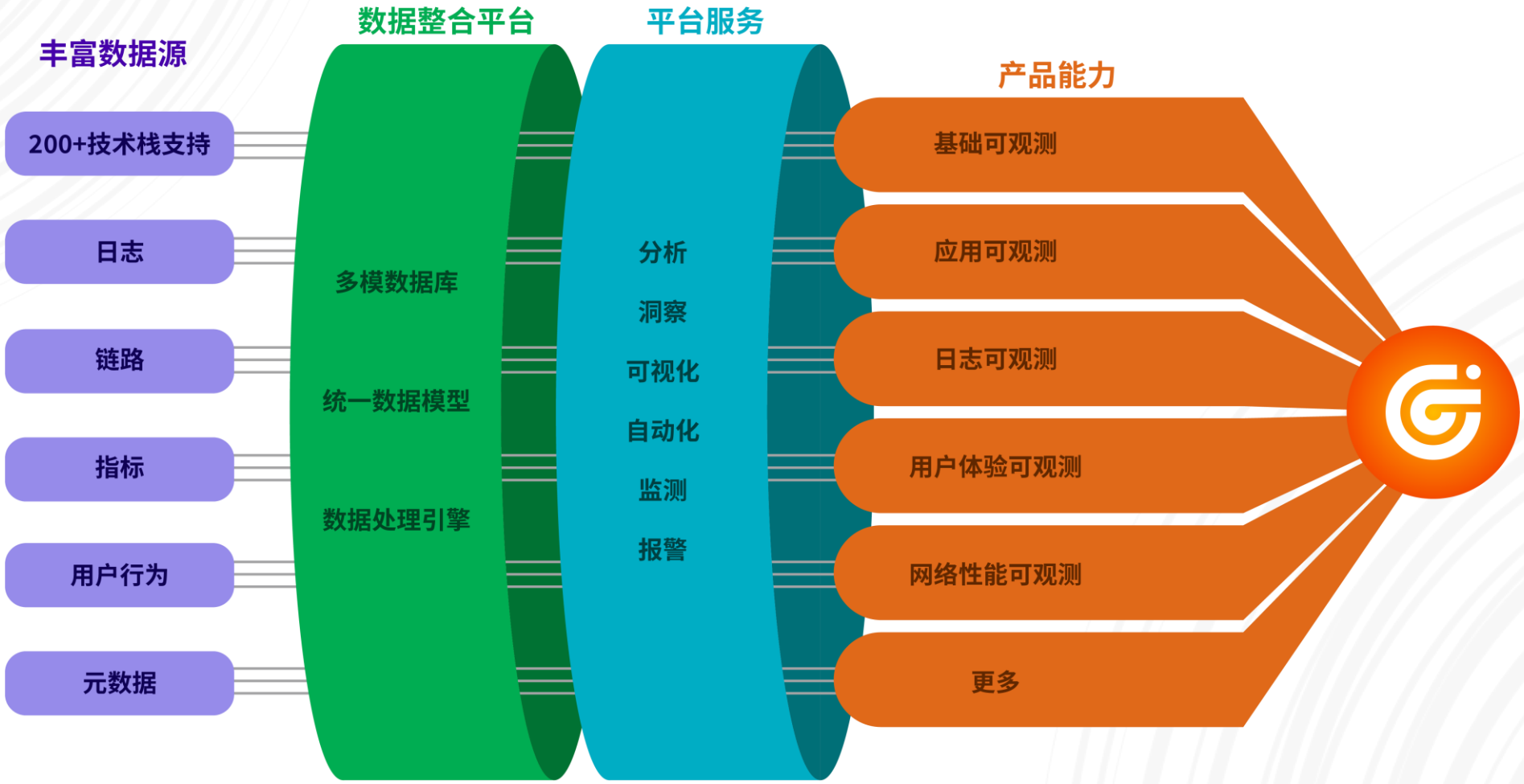
可观测的价值

观测云
GUANCE.COM



可观测的实现路径

观测云
GUANCE.COM



观测云
GUANCE.COM

观测云
GUANCE.COM

云 时 代 的 系 统 可 观 测 平 台

One Agent

观测云
GUANCE.COM

- 一个DataKit收集所有需要观测的数据
- 配置简单，兼容云/容器/K8S
- 跨平台与操作系统，支持X86/ARM，Windows/macOS/Linux
- 可控的性能占用，不超过5%的CPU和300MB内存
- 整合开源的工具，Telegraf/Prometheus/Skywalking

DQL 统一查询语法

观测云
GUANCE.COM

统一查询语法，无需关心底层数据存储，简单易上手。

```
M::cpu:(usage_idle) {region="cn-Hangzhou"} [5m] BY host
```

说明：查询时序数据指标集 cpu 最近 5分钟的字段 usage_idle（CPU使用率），以 region 来过滤，同时以 host 来分组显示结果。

SaaS模式

- 免运维：只管使用即可
- 高性价比：无需额外投入硬件资源
- 高可靠性：永不停机的守护者

安全

观测云
GUANCE.COM

- 放心的代码：用户侧代码永久开源
- 数据采集安全：单向通讯，不会也不能向客户环境下发指令
- 数据脱敏：所有的数据收集默认脱敏，也可以让用户控制
- 安全承诺：可信云、等保以及合同保证

产品价值

观测云
GUANCE.COM

场景	开源自建产品	使用观测云
构建云时代监测体系	专业技术团队至少超过3个月的投入，并且仅仅是开始	30分钟开箱即用
相关费用投入	一个简单开源监控产品的硬件投入也需要超过2万/年，如果是云时代可观测平台则至少10万/年固定投入（以云硬件估计）	按需付费，根据实际的业务情况，费用弹性，整体费用比使用开源产品的综合投入低50%以上
系统维护管理	需要专业的技术工程师长期关注与投入，多种开源产品混合使用也增加了管理的复杂度	无需关注，将精力放在业务问题上
需要在服务器上安装的探针数量	每个开源软件都需要一个探针，服务器大量性能被探针占用	一个探针，完全基于二进制运行，极低的CPU和内存占用
带来的价值	仅仅取决于公司自身工程师的能力，以及其钻研开源产品的能力	全方位数据化平台，全面可观测性，让工程师们利用数据解决问题
性能与故障的根因分析	仅仅靠团队自身的能力	基于数据分析快速定位
安全性	各种混杂开源软件，考验技术工程师的综合能力	全面的安全扫描和测试，客户侧代码向用户开源，并且产品及时迭代更新，确保安全
可扩展性与服务	需要自己构建SRE工程师团队	提供专业的服务，相当于配置一个外部的SRE支持团队
培训与支持	聘请外部老师	长期的在线培训支持

分布式微服务应用开发 可观测性实践

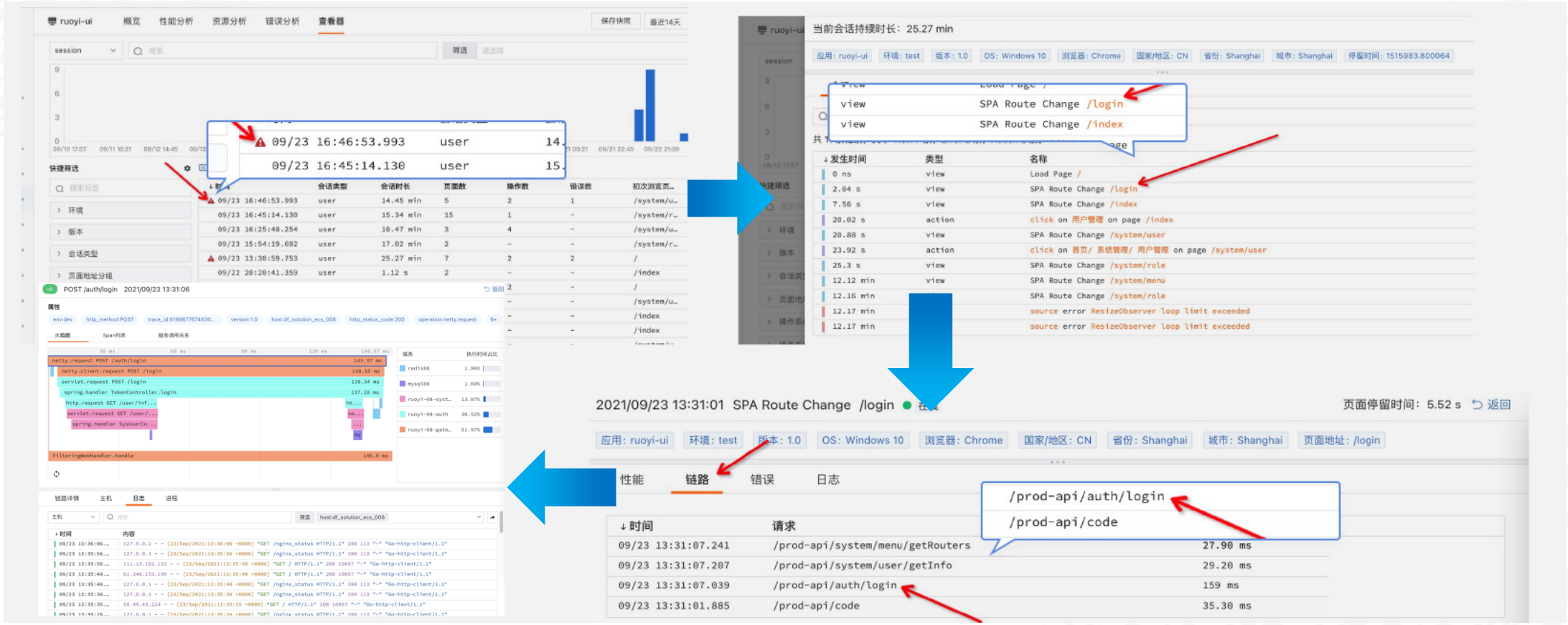
问题排查实践（示例：某用户登录加载失败）

用户体验监测
session查看器，通过时间或者userId找到该用户session下所有操作

查看该行为下的和后端打通的链路

点击具体的链路进入链路查看器

查看相应链路的日志



四条总结

1. 是否构建可靠性工程与观测系统，取决于两点：
    1. 系统的重要程度与复杂程度
    2. 是否关注用户体验

2. 选择什么样的观测系统，取决于两点
    1. 不能影响原有的系统
    2. 实施可观测性的成本(人力+软件+硬件)不能超过高可靠性带来的额外收益

3. 为了达到真正的可观测，取决于两点
    1. 数据收集的全面性(指标,日志,链路)
    2. 数据模型的统一性(多模数据库,统一查询)

4. 实现可观测的路径有两条：
    1. 基于多种开源各种组件，配置多种采集,打通数据模型,掌握多种查询，个性化开发和重度运维，这本身就是一个复杂系统
    2. 使用观测云：经验共享，软件免费,运维免费，以及更便宜的硬件资源投入

联系我们

观测云
GUANCE.COM

热线电话：400-882-3320（7*24h 技术支持）

咨询邮箱：sales@jiagouyun.com

官方网址：www.guance.com

上海（总公司）

电话：400-882-3320

地址：上海市浦东新区科苑路399号张江创新园7号楼

成都：

电话：028-65779858

地址：四川省成都市武侯区西部智谷B区4栋2单元802

扫码添加，加入观测云社群

第一时间掌握观测云最新资讯