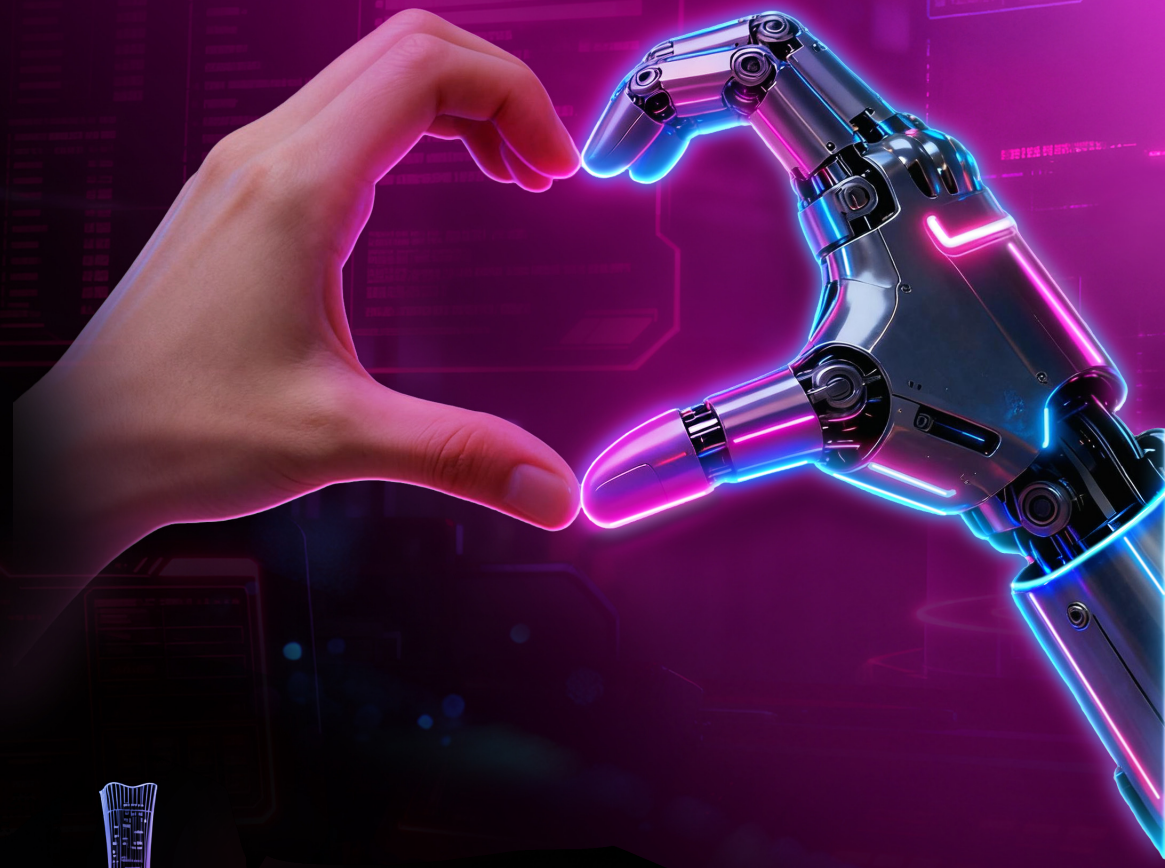
开源社
kaiyuanshe

COSCon'25
第十届中国开源年会

众智开源 | Open Source, Open Intelligence



2025可观测性行业观察和迭代思考

秦晓辉 / 快猫星云 · 联创&CMO

开源社
kaiyuanshe

COSCon'25
第十届中国开源年会
众智开源 | Open Source, Open Intelligence

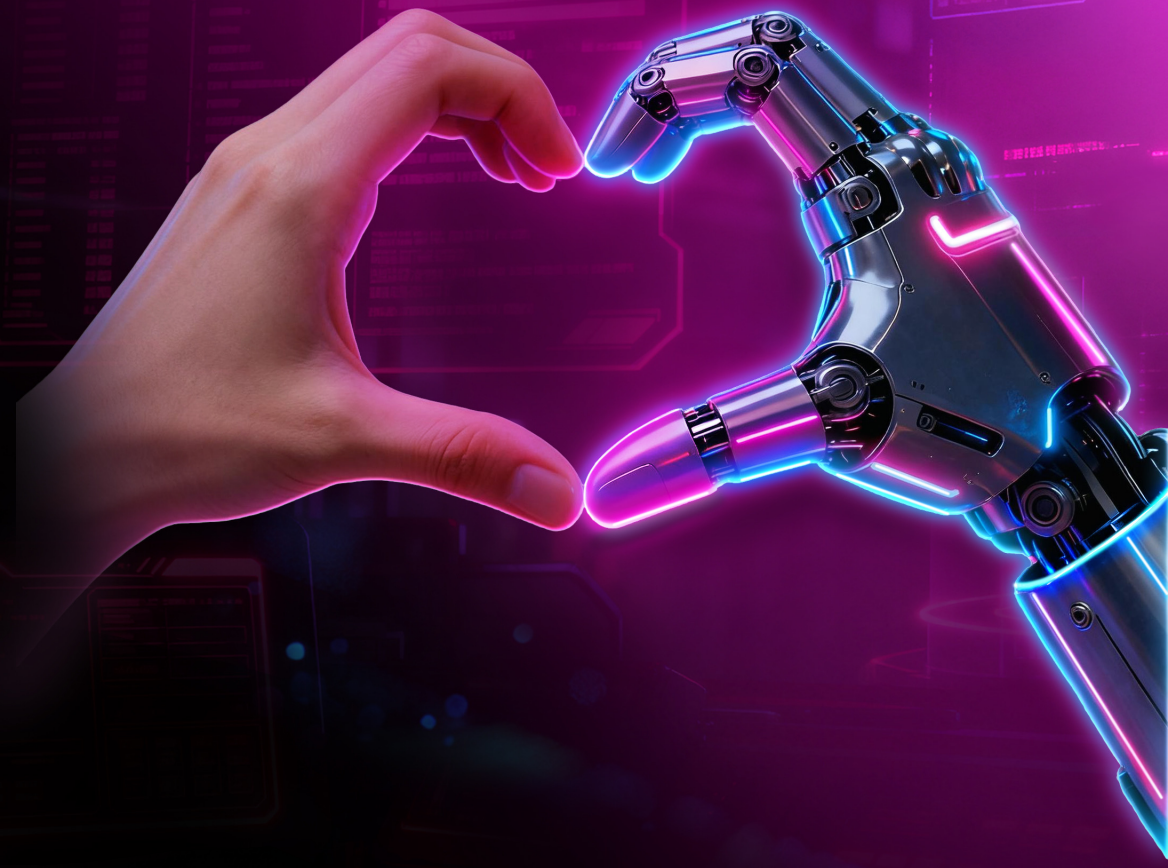
秦晓辉

Open-Falcon、Nightingale 等开源项目创始人和主程，极客时间专栏《运维监控系统实战笔记》作者，公众号 SRETALK 主理人，目前作为快猫星云的联创&CMO，创业中。

巴辉特
中国大陆

扫一扫上面的二维码图案，加我为朋友。

CONTENTS



PART 01

透彻理解可观测性的缘起和价值

何为可观测性？

开源社 kaiyuanshe

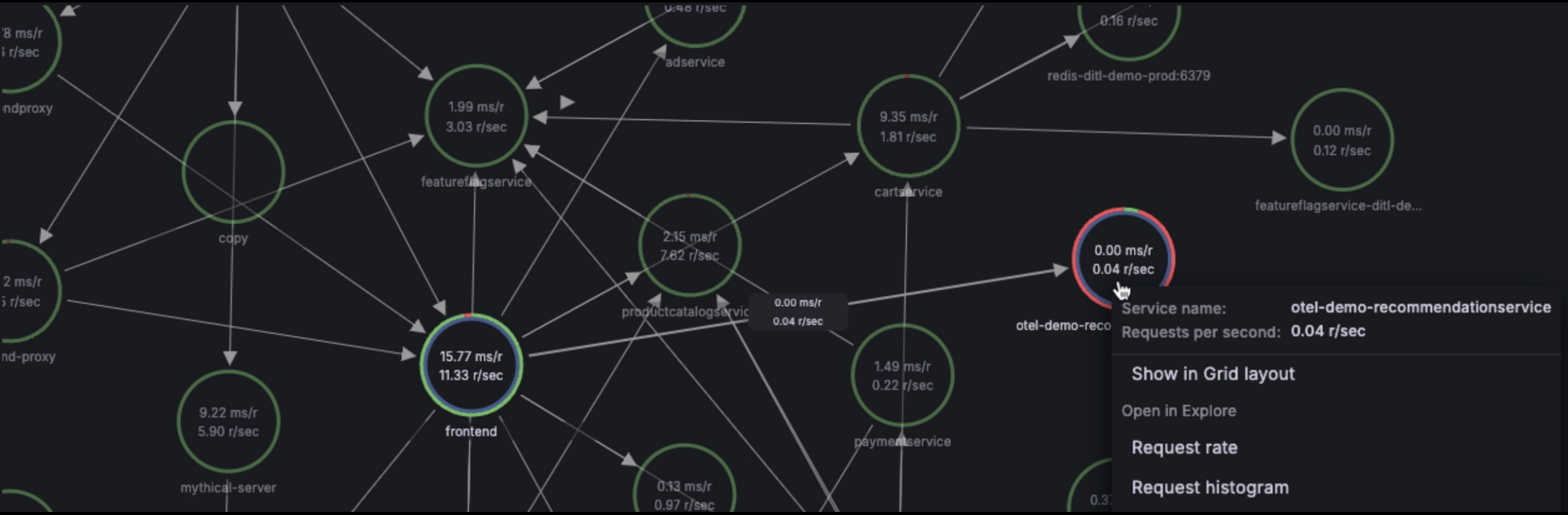
高可用性

可观测性

可观测性，是软件的一个特性，可以类比软件的高可用性。我们说某个软件具备好的可观测性，表示通过分析软件的对外暴露的数据可以较为容易理解其内部运行情况（比如其业务逻辑是否执行正常）。高可用性是要通过架构设计实现的，可观测性也需要有所设计。

可观测性解决什么问题？

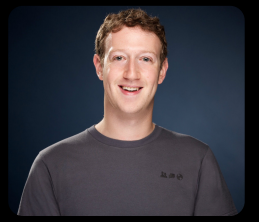
随着微服务的推广流行，软件问题越来越难排查，每次出故障，所有产生告警的微服务研发/运维都要参与排查，混乱不堪，缺少一个 high level 的视角，于是 Tracing 诞生。进而业界意识到，Metrics、Logs、Traces 都有助于排查故障理解软件，那引入一个新词吧，于是从控制论里偷师了“可观测性”，形成软件行业的一个细分领域。



单体架构不需要可观测性？

单体架构也要排查问题，显然也需要可观测性

单体架构也会出问题，也需要排查，所以也需要指标、日志等可观测性手段，如果单体架构逻辑复杂，内部分成很多 module，或者有很多层函数调用，引入链路追踪也是不错的手段，或者在日志里埋入用于串联的标识，也可以起到和链路追踪类似的效果。



世界级的顶级单体项目，会暴露各种数据辅助排查问题

MySQL 会暴露进程日志，也可以通过执行 SQL 获取指标（比如 show global status、show global variables）、慢查询等；Redis 也会暴露进程日志，也可以通过执行 INFO 等命令获取内部指标。

从理念之争来理解可观测性

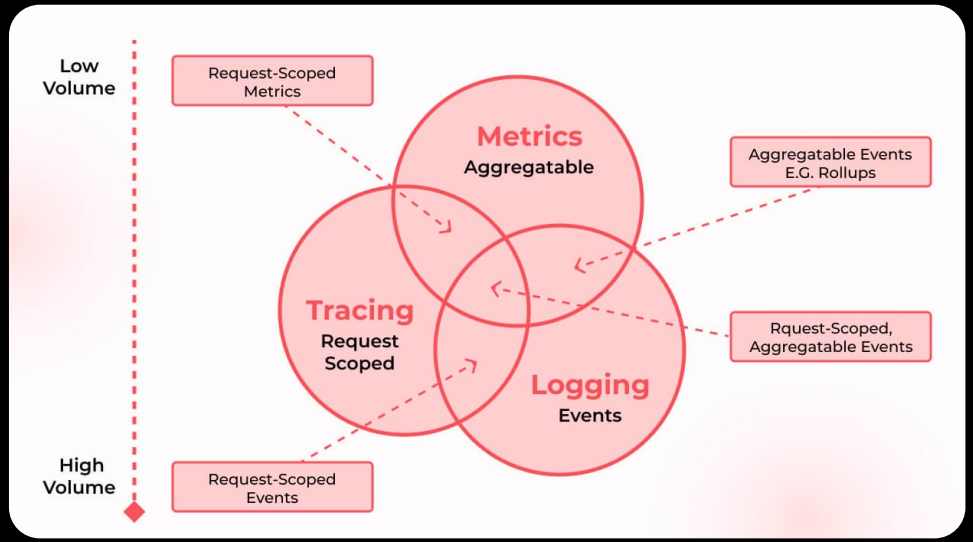
可观测性核心是数据收集、存储、分析，收集哪些数据？怎么存储怎么分析？一直争议不断。经典论调是“三大支柱”和“宽事件”。

三大支柱

Metrics、Logs、Traces，被看做支撑可观测性体系的三大支柱，当然，叫支柱其实不太合理，支柱是一种偏市场营销的话术，就简单理解为三种信号类型即可。可观测性领域炙手可热的 OpenTelemetry 项目，内部处理时就是把数据分成了这三种类型，当然，近期也开始引入 Profiles 数据。

宽事件（Wide Events）

三大支柱的数据分开处理，就容易割裂，难以串联分析。宽事件理念则比较粗暴极致，它认为可观测性数据只有一种，就是丰富上下文语义、具有很多维度的“宽事件”，宽事件聚合就是指标，宽事件串联就是链路。和 2016 年提出的 Canonical Log Lines 的理念很是相像。

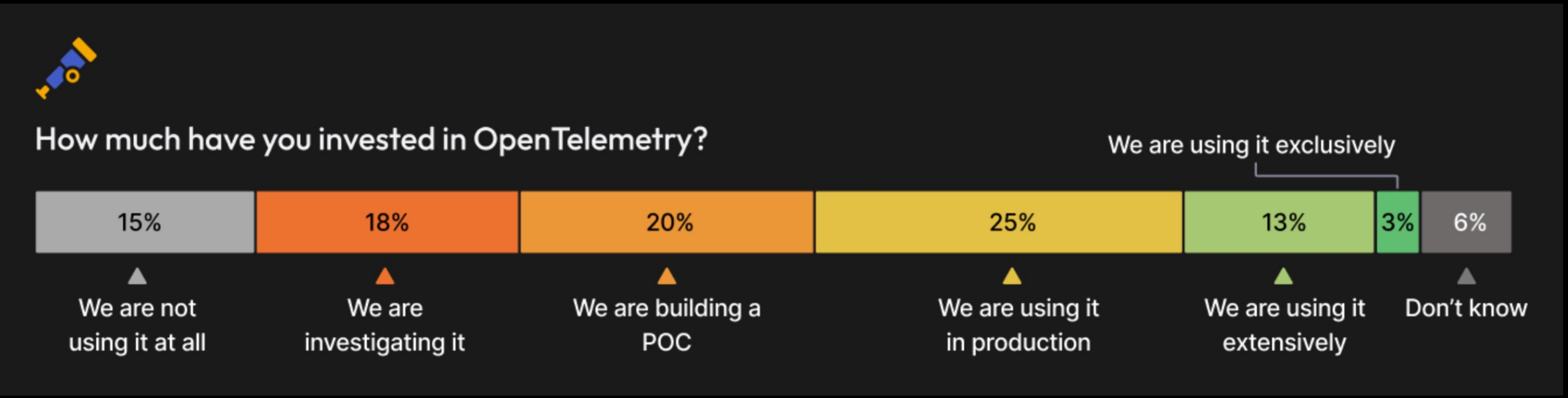


PART 02

可观测性发展现状和行业格局

可观测性认知度极高

OpenTelemetry 作为可观测性领域的旗帜项目，只有 6% 的企业不知道（是否直接采用 OpenTelemetry 每个企业情况各异），可见绝大部分企业都对可观测性有所了解，可观测性是一个充分被教育的市场，有着广泛的接受度。

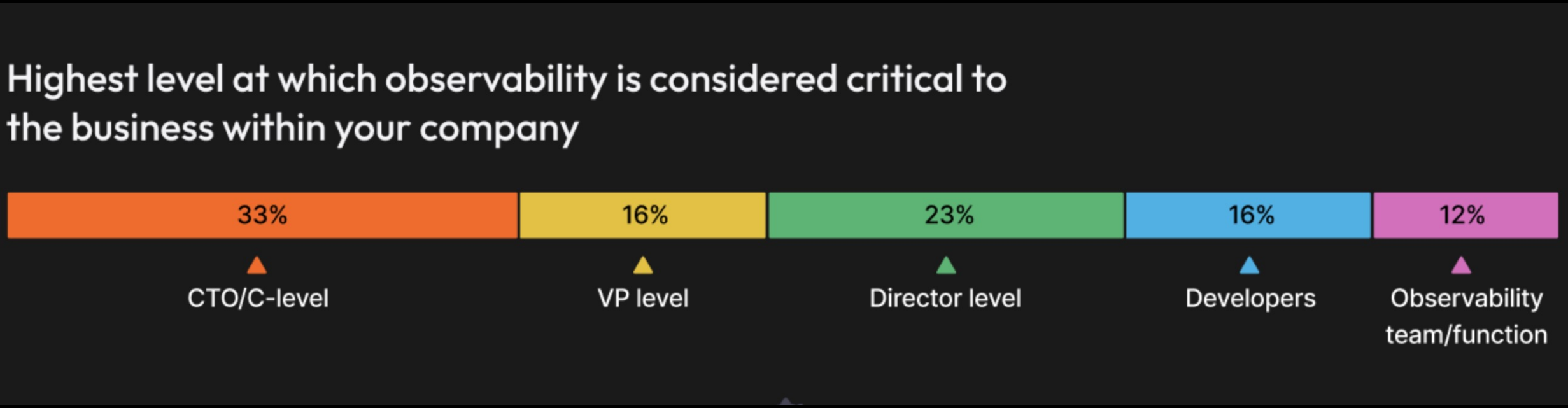


Traces 作为可观测性的第三大 Telemetry 采用率过半

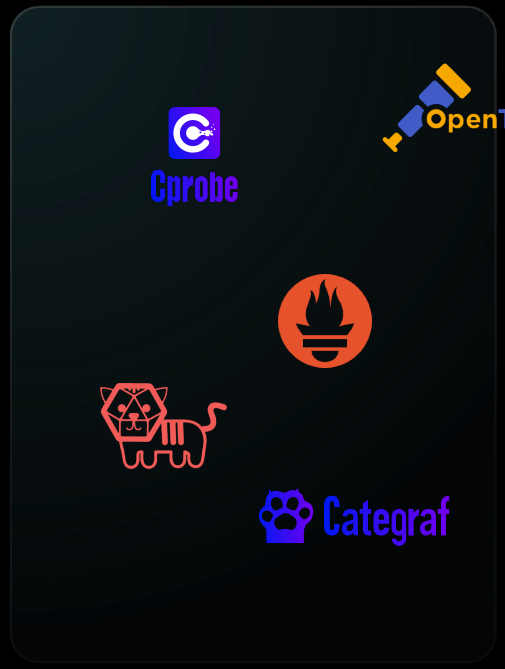
可观测性领域有“三大支柱”的说辞，Metrics 和 Logs 采用率极高，一定程度上是因为监控时代已经有所落地，而 Traces 作为第三大支柱，其采用率一定程度上可以反映可观测性实施的成熟度，在 Grafana 的调查中已经有过半企业落地了链路追踪。

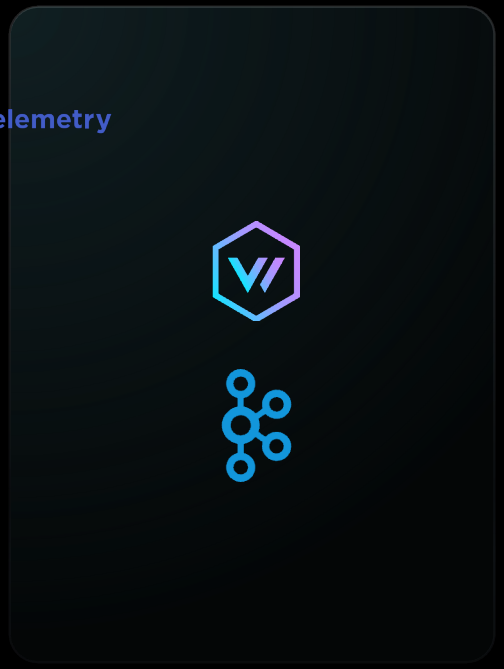
在 70%+ 的企业中，可观测性已进入高管视野

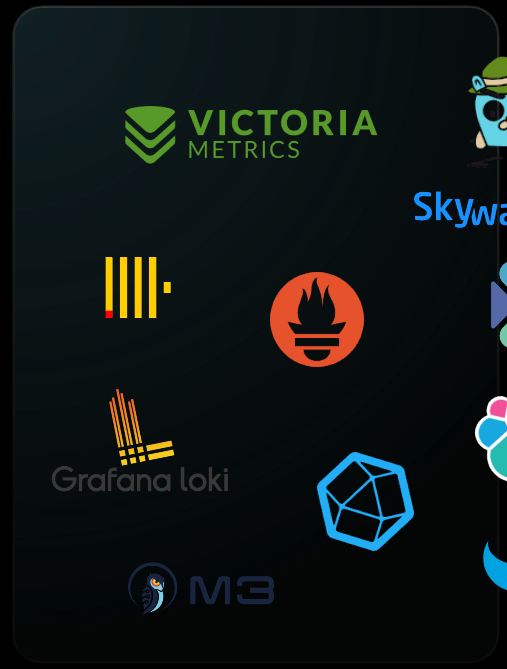
在 70%+ 的企业中，可观测性已进入高管视野，这意味着该领域会得到更多重视，得到人才和资源的倾斜。而认为可观测性的重要性未超出单个团队层面的比例不到三分之一（开发人员占16%，可观测性团队/贡献者占12%）。



核心生态位均有对应的开源项目







采集　　传输　　存储　　可视化分析

告警和事件二次处理是拼图的最后一块

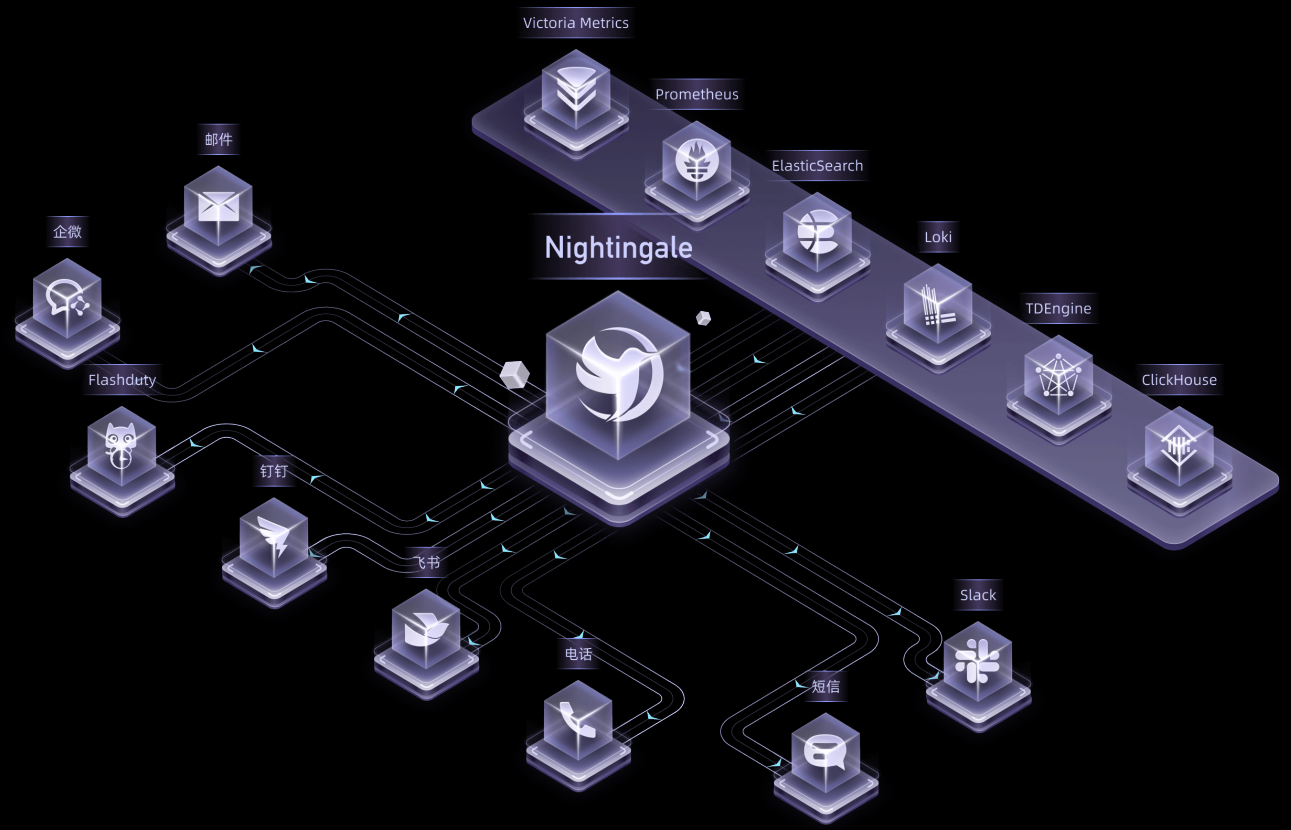
夜莺（Nightingale）开源项目卡位在多数据源统一告警和事件的二次处理，或许，这是可观测性开源项目拼图的最后一块了

多数据源统一告警

指标、日志等通常存在不同的数据源，即便只是指标也可能会分成多个数据源，统一维护告警规则是一个刚需。

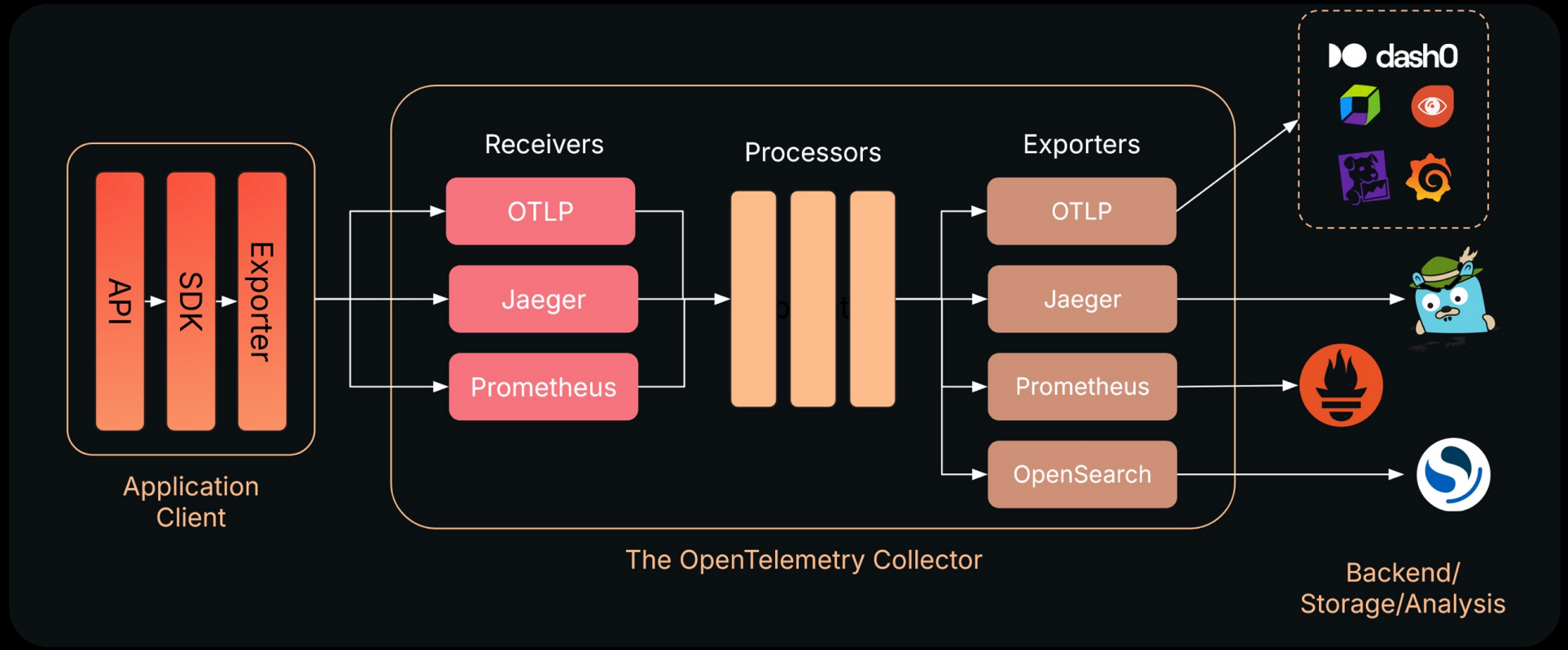
告警事件二次处理

告警事件产生之后，通常会有二次处理的需求，比如通过 Label Enrichment、联动 CMDB 等附加更多上下文信息，帮助 On-call 人员快速排障、止损。



OpenTelemetry 初长成，声量极高，不过并未大一统

由于 OpenTelemetry 的中立性、繁荣的社区生态，OTEL 已然成为新的观测厂商必然支持的选项。Jaeger 直接放弃了自己的采集器，Grafana 和 SigNoz 做了 OTEL 的发行版，Dash0 则提出了 OpenTelemetry-Native 的概念。



OpenTelemetry 在指标领域尚有争议，落地案例较少

开源社 kaiyuanshe

Prometheus 的作者之一 Julius Volz 在 2025.07 发表了一篇雄文，提醒用户如果想要把 Prometheus 和 OTEL 整合起来使用，需要慎重三思，陈述了 6 点考量。

01

丢失 P8s 的目标健康监控能力

P8s 是 PULL 模式，可以感知监控对象是否存活，而 OTEL 是 PUSH 模式，失去了感知能力

02

指标名称字符集和语义差异

OTEL 的指标名称限制较少，较少考虑 QL 场景的直观和语法结构冲突

03

资源属性和目标标签处理差异

P8s 采用目标 info 标签机制来唯一标识监控目标，和 OTEL 的资源属性理念各异，没法直接映射

04

P8s 需要额外配置才能支持 OTEL

OTEL 推送的指标可能是乱序的，使用 P8s 作为接收器需要额外配置

05

OTEL SDK 复杂且性能差

OTEL 要兼顾三大支柱数据，整体设计复杂，相比直接使用 P8s 的插桩 SDK，性能差

06

其实，P8s 也是开放的

大家选择 OTEL 的关键是因为 OTEL 是开放的中立的，其实，P8s 也是

https://promlabs.com/blog/2025/07/17/why-i-recommend-native-prometheus-instrumentation-over-opentelemetry/

OTEL 在不同领域成熟度不同

开源社 kaiyuanshe

Traces

OTEL 在链路方面最为成熟，落地案例最多，除了 Rust SDK 尚处于 Beta 状态，其他所有语言的 SDK 都是 Stable

Metrics

除了 P8s 作者陈述的那几点考量之外，OTEL 还有一个很大的问题，是仪表盘和告警规则生态的匮乏，这导致其短时间很难流行起来

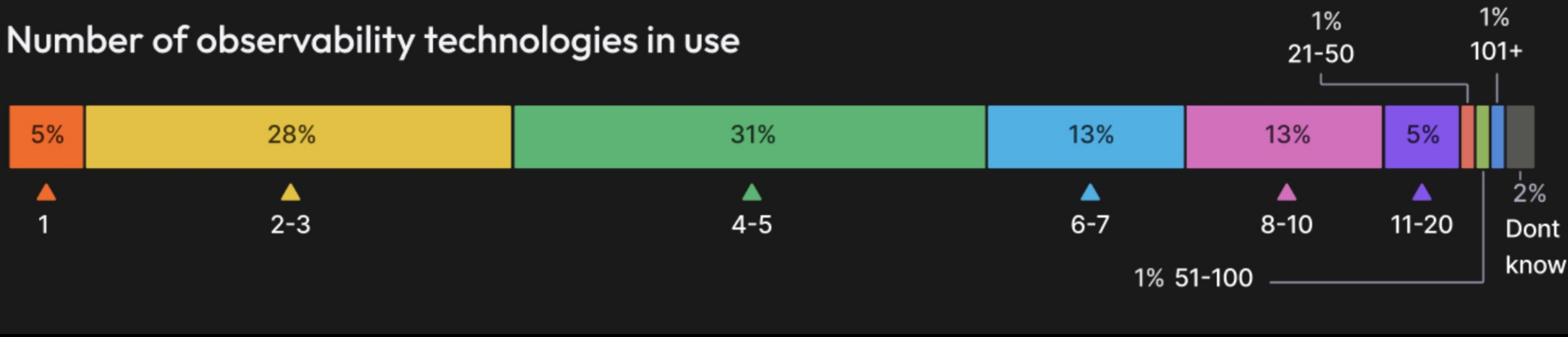
Logs

日志方面很多语言的 SDK 尚在开发中，但在社区中已看到有公司在生产采用，核心诉求来源是想统一链路和日志的标签语义规范，方便串联

Language	Traces	Metrics	Logs
C++	Stable	Stable	Stable
C#/.NET	Stable	Stable	Stable
Erlang/Elixir	Stable	Development	Development
Go	Stable	Stable	Beta
Java	Stable	Stable	Stable
JavaScript	Stable	Stable	Development
PHP	Stable	Stable	Stable
Python	Stable	Stable	Development
Ruby	Stable	Development	Development
Rust	Beta	Beta	Beta
Swift	Stable	Development	Development

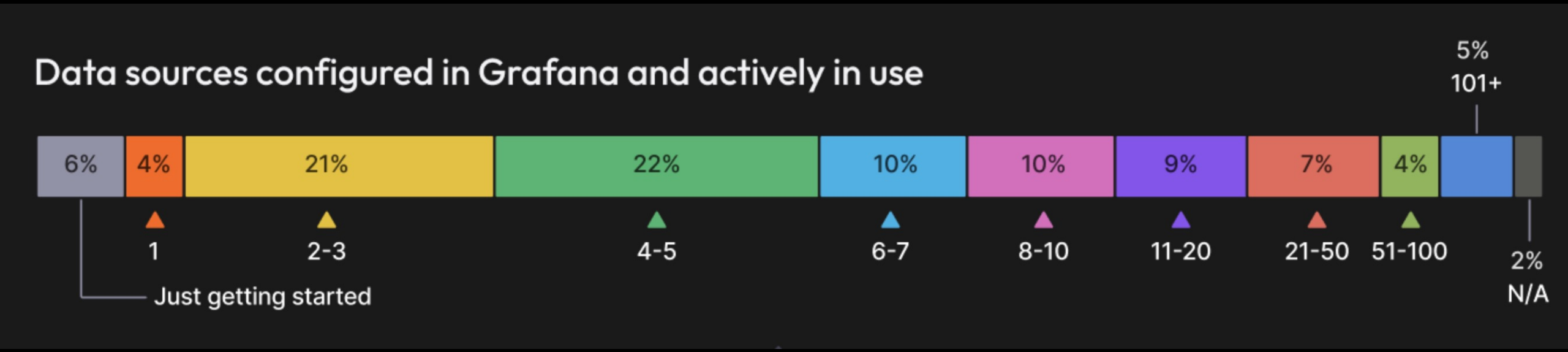
企业在用的可观测性工具平均多达 8 个

可观测性领域既老又新，从古老的 IBM Tivoli 到时髦的 OpenTelemetry，不同的细分需求容易安放不同的单一解决方案（确实也很难找到一款尽善尽美的方案通吃一切场景），导致企业真实在用的可观测性相关工具数量众多，平均多达 8 个。



Grafana 用户在平台上配置的数据源数量平均为 16 个

Grafana 用户在平台上配置的数据源数量平均为 16 个，不同的数据存在不同类型的数据源，即便是相同类型的数据源，也有数据隔离切分的需求。我们在夜莺社区用户中看到过接入了 200 多个数据源的用户。



据说，可观测性工具的数量在“收敛”

New relic 调研数据

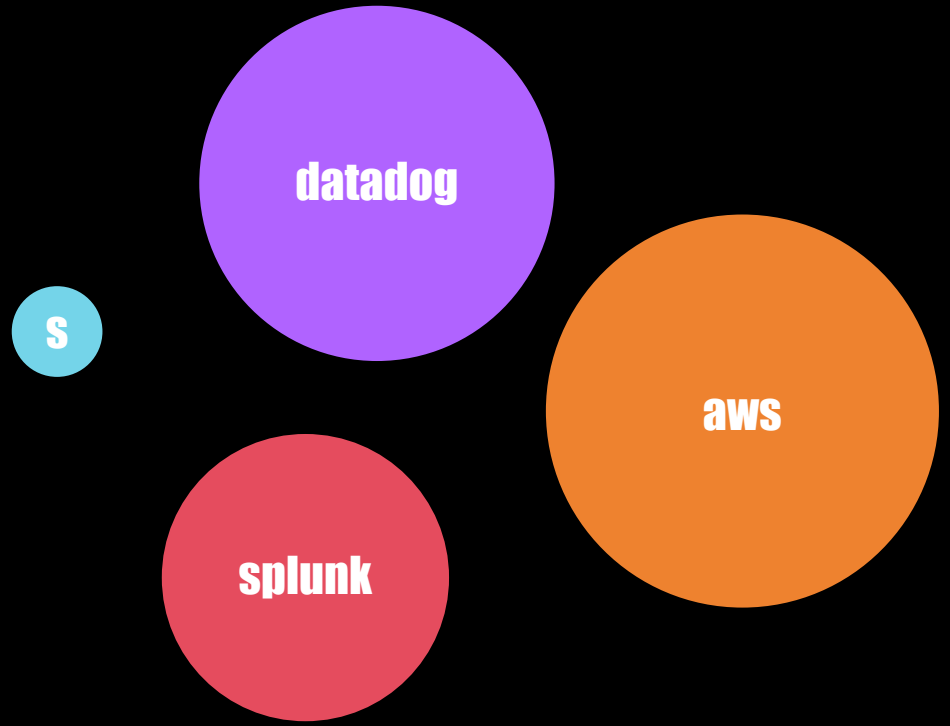
New relic 调研显示，过去两年企业使用的可观测性工具呈现收敛趋势，下降了 27%，目前平均使用的数量是 4.4 个

注意 SaaS 厂商调研的用户群体有一定的共性，未必能代表整个行业的情况，仅供参考。

所有数据都接入 SaaS 很省事，但可能也很贵

- reddit 上一篇帖子广受关注 Our observability costs are now higher than our AWS bill：$47k for datadog. $38k for splunk. $12k for sentry. our actual aws infrastructure costs $52k.

- SaaS 很省心，但是 SaaS 太贵。如果减少上报的数据则可以降低成本，但是效果就打了折扣。可观测性领域曾经的暴论“应采尽采”，为了探索“未知的未知”，现在已经少有人提及。



好在，SaaS 也开始兼顾 On-Premise 数据，比如 Datadog

Datadog 推出了 CloudPrem

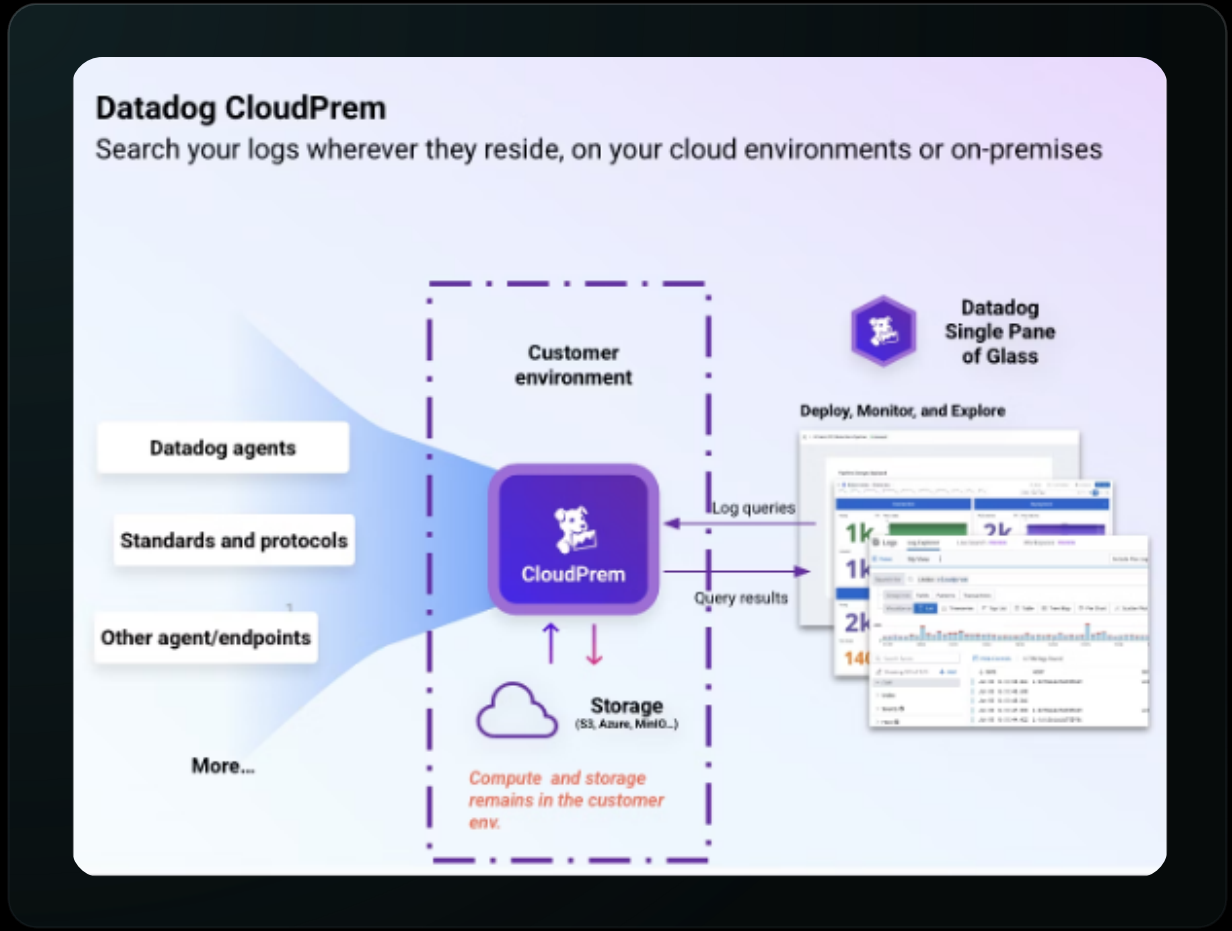
或许是因为 Datadog 的成本太高，所以把数据存储放到客户自己的环境里降低成本？

开源生态已经非常完备

指标、链路、日志都有相应的开源方案，不买产品，用开源来攒，我来维护，我就是产品...

商业方案和开源混用也是一个思路

我只做我擅长的，简单的，ROI 高的，我不擅长的则交给厂商



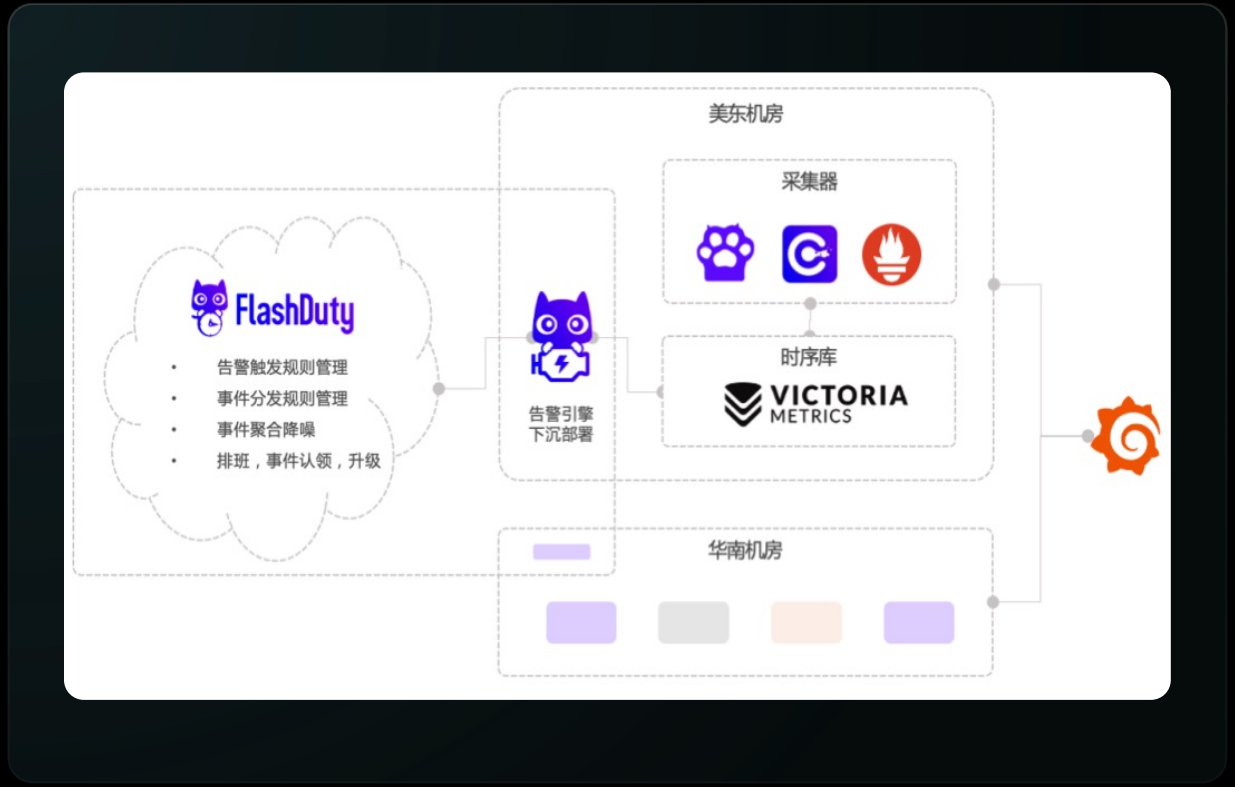
好在，SaaS 也开始兼顾 On-Premise 数据，比如 Flashduty

Flashduty monitors 产品逻辑

Flashduty SaaS 仅管理告警规则，在客户的数据中心部署告警引擎（monitedge），引擎从云端同步告警规则，做判定，生成事件之后推给云端，云上云下打通

Flashduty monitors 产品特点

支持对接常见数据源，比如 Prometheus、Victoria、ElasticSearch、MySQL、ClickHouse、SLS 等；
告警规则极为灵活，支持阈值判定、数据存在、数据缺失等多种告警模式，支持通过额外的查询语句判定是否恢复，支持附加查询；



注册体验：https://console.flashcat.cloud/

未来十年，多种数据源相互集成的 “插线板” 模式可能就是主流和现实

开源社 kaiyuanshe

阵营A 插线板

Grafana Flashcat

- 目前没有一款产品可以通吃所有功能，不同的产品定位不同，擅长的方向不同
- 企业同时使用多款可观测性工具是现实和常态，有强烈的利旧需求，那就把各个工具整合起来一同发挥价值

阵营B 一体化

- SaaS 模式没法访问客户私有数据源，所有数据都要进入云端
- 投入大量研发人力做透每个细分方向，以全球老大哥 Datadog 为例，员工有上千人，资本市场估值几百亿美金

eBPF 是一种有用的采集方式，但是尚未成为主流

Grafana Beyla 捐给了 OpenTelemetry 社区

Grafana Beyla 发布了较长时间但是迭代缓慢，亟需社区贡献力量，后来捐赠给了 OpenTelemetry 社区，变得中立，于是有了更多厂商参与贡献。

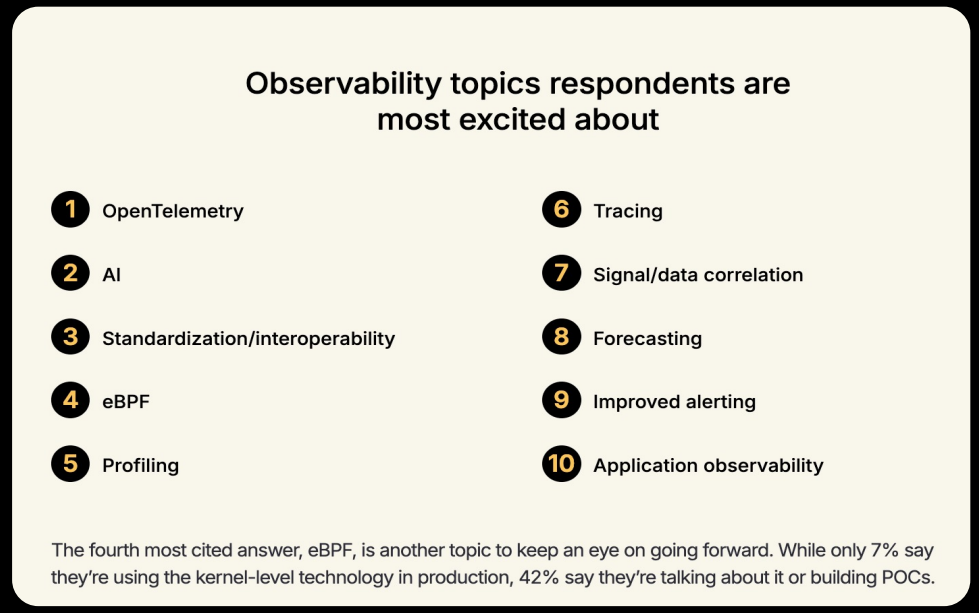
OpenTelemetry eBPF Instrumentation 于 2025.11.3 发布第一个 alpha 版本

OpenTelemetry eBPF Instrumentation Marks the First Release 侧重在 RED 指标的收集，对 Tracing 也是一个补充，但后续还需要投入巨大的精力做语言和框架协议的兼容。

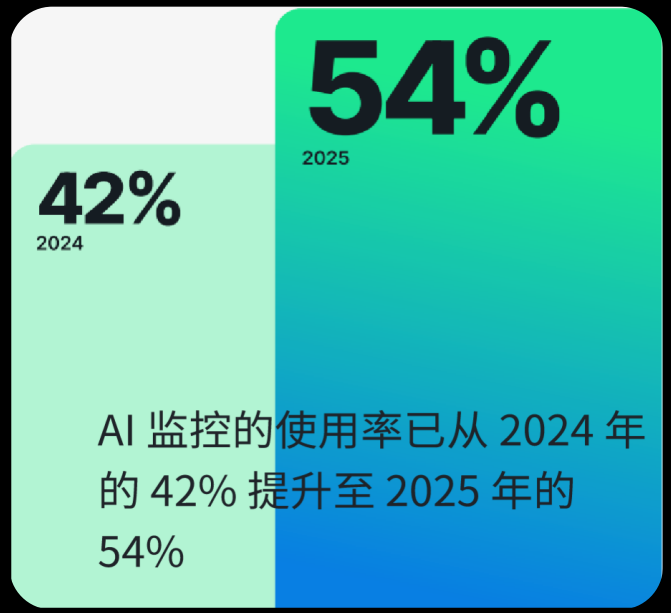
eBPF 尚未成为主流

可能的原因：内核版本升不上去、支持的语言和框架还不够丰富、已经手工（更灵活、颗粒度更细）埋过点了、出现问题担心搞不定等等

AI 驱动的可观测性技术在企业内快速导入



在可观测性领域，调查者反馈：最让人兴奋的是 OpenTelemetry，第二让人兴奋的就是 AI，AI 正在企业内快速导入。



调查显示，AI 监控的使用率已经从 2024 年的 42% 提升至 2025 年的 54%，未来已来。

因为，AI 确实可以加速 O11y

所有厂商都在积极拥抱 AI，这是趋势无疑

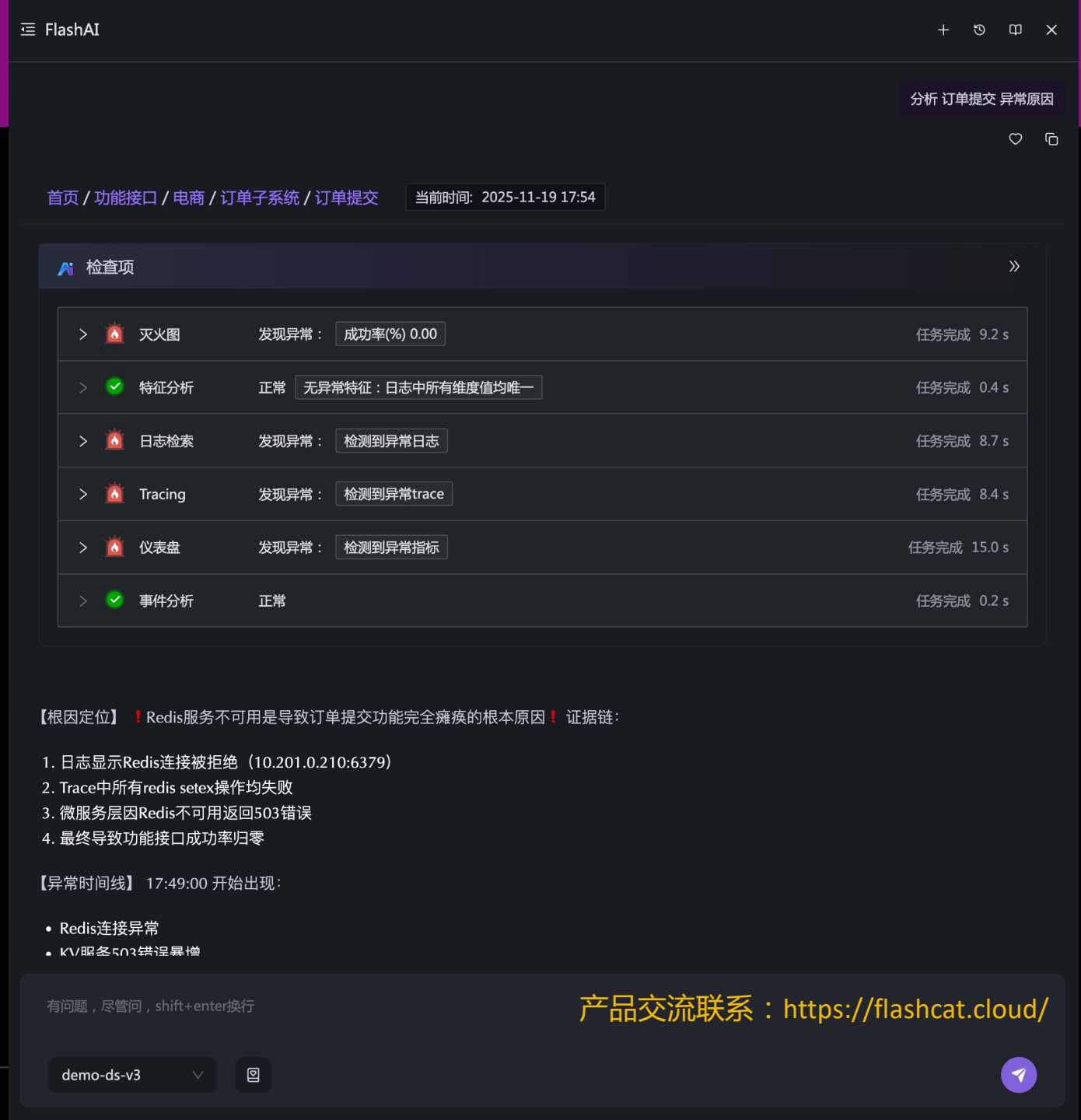
所有可观测性厂商都在尝试把 AI 能力引入自己的产品，比如 Datadog 的 Bits、New Relic AI、快猫星云的 Flash AI

数据底座是 AI 良好效果的基石

没有好的数据，那就是 Garbage in, Garbage out。怎么算好数据：完备、维度丰富、格式统一、语义统一、有关联字段

把故障定位路径总结好，让 AI 帮我们走这个路径

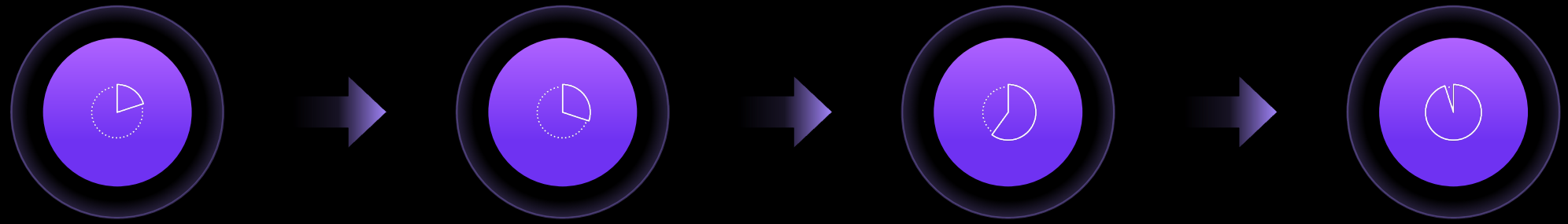
只是针对单个告警事件分析，意义不大。把故障定位路径提前准备好，让 AI 帮我们走这个路径可以大幅提升工作效率。故障定位路径举例：全局业务指标 -> 某个系统的 SLO -> 某个模块的 SLO -> 日志聚类 -> 链路追踪。



PART 03

可观测性成熟度模型和落地路径

可观测性成熟度模型和落地路径



1. 零散组件监控

收集服务器、网络设备、数据库、中间件的基础指标和日志，配置基础监控

2. 建立观测数仓

建立应用白盒监控，采集更多类型的观测数据，建立较为完备的数仓，导入仪表盘做基础分析

3. 人工提炼洞见

从目标和价值出发，懂得分析数据特征，可观测性数据在根因定位、容量管理方面发挥价值，重视 SLO、RED 等方法论

4. AI 产生洞见

数据越发完备、规范，引入 AI，原本由人来做的数据分析路径交给 AI 分析，提升效率、灵感创意

PART 04

可观测性建设的经验和教训

可观测性落地，组织推动形式是第一位的

优先自顶向下推动

深度落地可观测性是需要所有技术团队参与的，如果能够统筹设计，并自顶向下推动，那是最容易的。

次之先做标杆出来

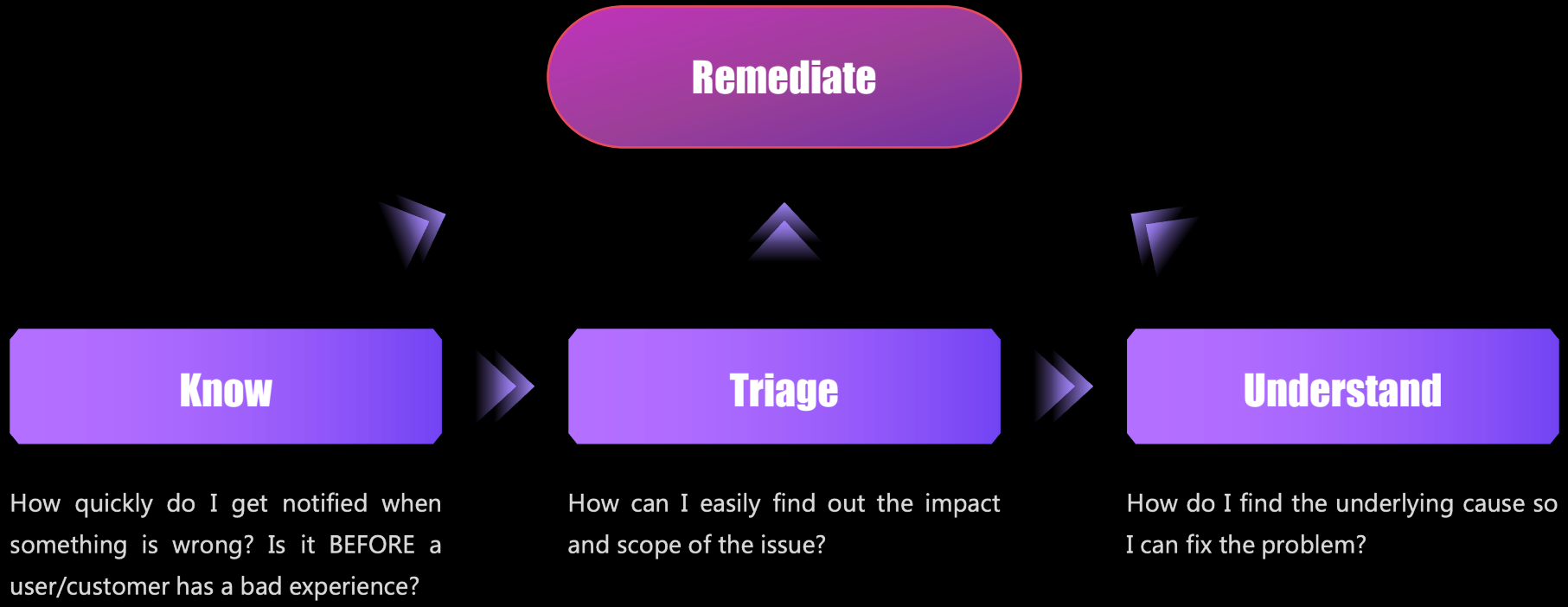
寻找有代表性（公司重点业务？故障定位时效性诉求很强？）的业务做出标杆效果，其他业务效仿之。

平台团队输出：平台工具+最佳实践+样例+运营手段

其中运营手段容易被忽视，需要把可观测性的价值讲清楚，如果有定量对比那是最好的，否则就把过程指标和成果呈现出来。

可观测性要目标驱动，最大的应用场景是定位止损

不要为了数据而数据，为了平台而平台，可观测性最大的场景是故障定界、定位、止损



https://thenewstack.io/beyond-the-3-pillars-of-observability/

可观测性建设要规范先行

输出格式要有规范

比如日志全部要求使用 JSON 格式输出，方便后续的 Pipeline 处理；指标全部在 /metrics 端点输出，统一使用 OpenTelemetry 或 Prometheus 格式

输出内容要有规范

比如直接采用 OpenTelemetry 的语义规范，内容可以参考 Wide Events 把尽可能多的维度信息塞进日志

该有的内容必须有

比如应用程序必须要有 RED 指标，系统边界必须要有调用成功与否、延迟如何的指标，从进程外（调用其他 API 的数据或者从 OS 拿到的一些关键数据）获取的所有数据都要打印日志

Semantic Conventions

Common names for different kinds of operations and data.

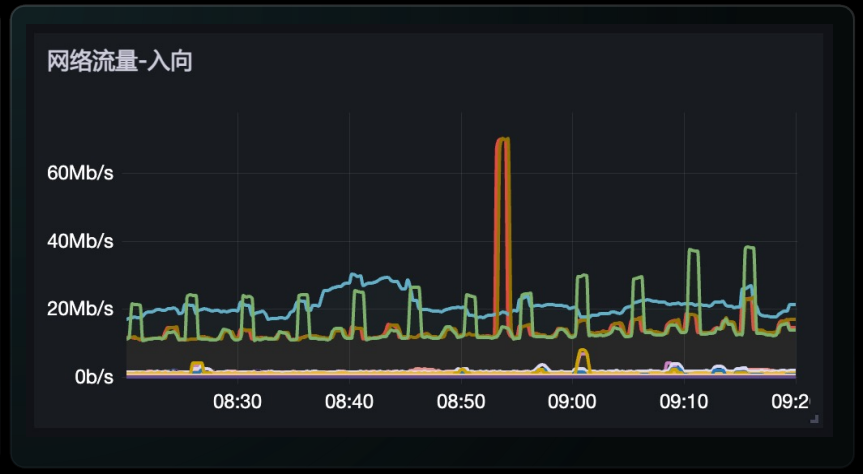
OpenTelemetry defines Semantic Conventions, sometimes called Semantic attributes, that specify common names for different kinds of operations and data. The benefit of using Semantic conventions is in following a common naming scheme that can be standardized across a codebase, libraries, and platforms.

Semantic conventions are available for traces, metrics, logs, profiles and resources:

- Trace semantic conventions
- Metric semantic conventions
- Log semantic conventions
- Profiles semantic conventions
- Resource semantic conventions

要重视利用特征分析快速下钻的能力，而非图表炫酷性

开源社 kaiyuanshe



关于告警的一些建议

针对衡量用户体验和业务目标的指标告警

在微服务和 Kubernetes 盛行的今天，某个单机的故障很可能不会影响服务，只针对重要指标告警，可以大大减少告警疲劳，告警疲劳是导致故障的重要因素之一

但凡告警必有 SOP

如果告警把人呼起来了，那就应该有 SOP，这基本已经是业内共识，统计所有的告警规则，看看有多少告警规则预置了 SOP，这个比例可以作为告警治理的量化数据

普通指标可以产生告警事件，但通常无需通知到人

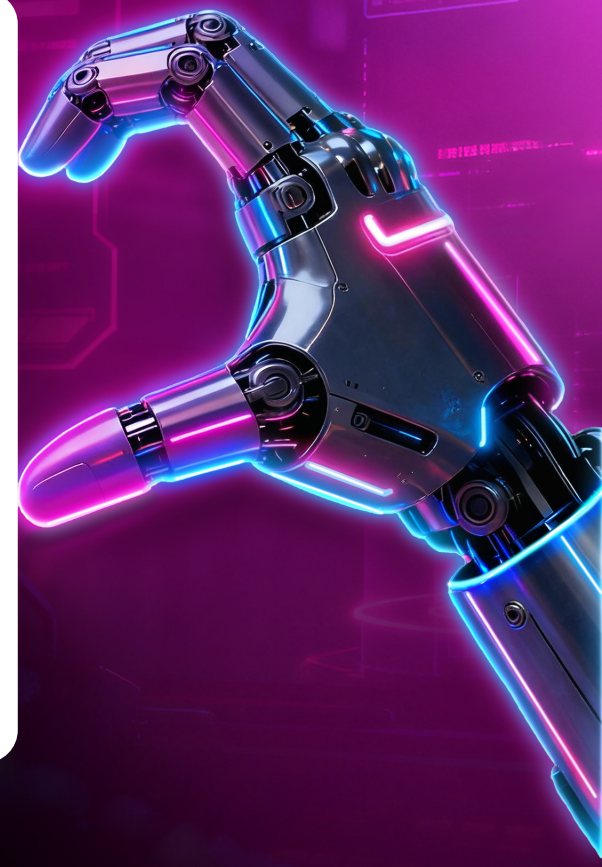
跟最终体验无关的告警也可以有，作为一种海量观测数据的 insight，但这类告警事件不用通知到人，On-call 人员查问题的时候可以参考即可

开源社
kaiyuanshe

COSCon'25
第十届中国开源年会

众智开源 | Open Source, Open Intelligence

Thanks



巴辉特
中国大陆

扫一扫上面的二维码图案，加我为朋友。

秦晓辉 / 快猫星云 · CMO