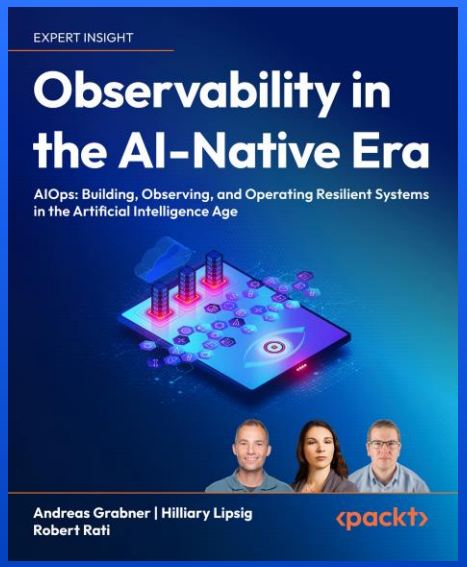
OBSERVABILITY IN THE AI-NATIVE AGE

HELPING THE HUMAN AND THE AI THROUGH OBSERVABILITY

Who am I?

Andi (Andreas) Grabner

- CNCF Ambassador
- Fellow DevRel @ Dynatrace
- PurePerformance Podcaster
- Occasional Book Author
- Salsa Dancer



When you hear “Observability and AI”

What is the first Use Case you think of?

1

Monitor and Optimize the AI/LLMs **we host**

2

Monitor usage of CoPilot, Claude, Cursor, Codex

3

Integrate Observability into IDE, e.g: via **MCP**

4

A different use case - or – **no thoughts**!

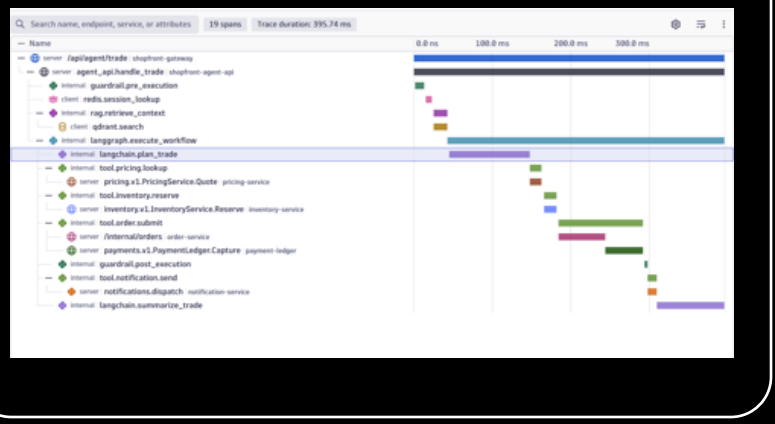
Based on what I see, I've prepared this!

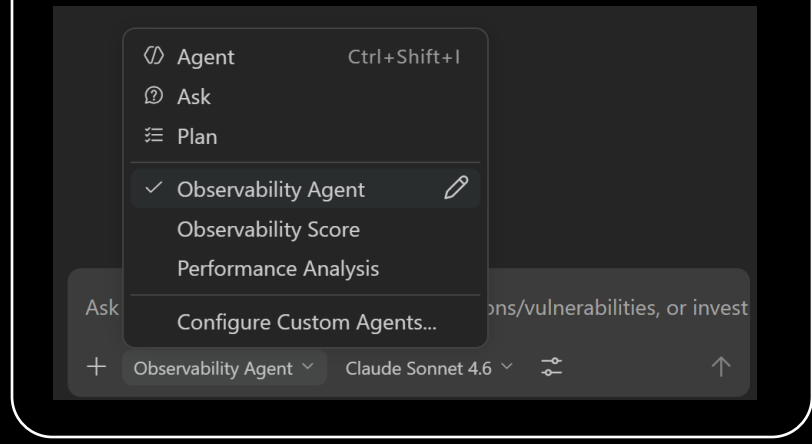
3 Things I hope you take away today + maybe a bonus point ☺

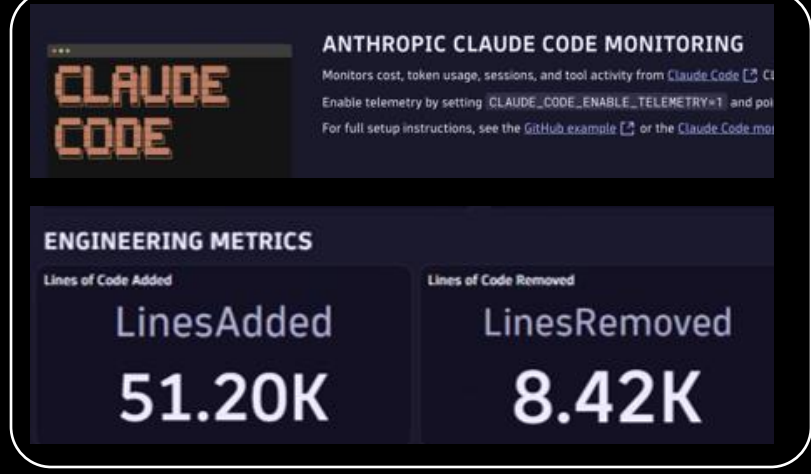
The Basics
Observability Standards

The AI Delivery Lifecycle
Observability for and with AI

Cost and Impact
Optimize usage and alternatives







"What's different in observing an app, an AI or a workflow?"

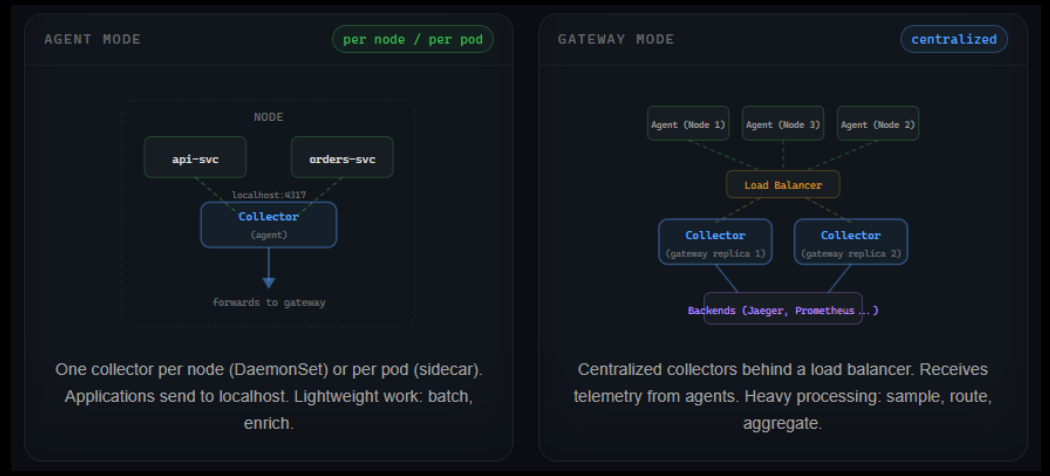
"How can my AI become better with close observability loops?"

"How to maximize the outcome while staying within budget?"

THE OBSERVABILITYY AND AI PRIMER: BASICS YOU NEED TO KNOW!

New to OTel? 12 Koans of OpenTelemetry!

A playful way to understand observability: https://otel.mreider.com/



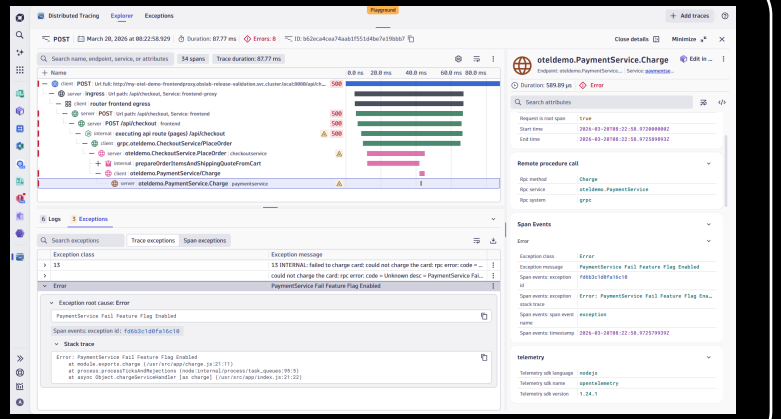
3 Types of Traced Systems: App, Agentic, GH Workflow

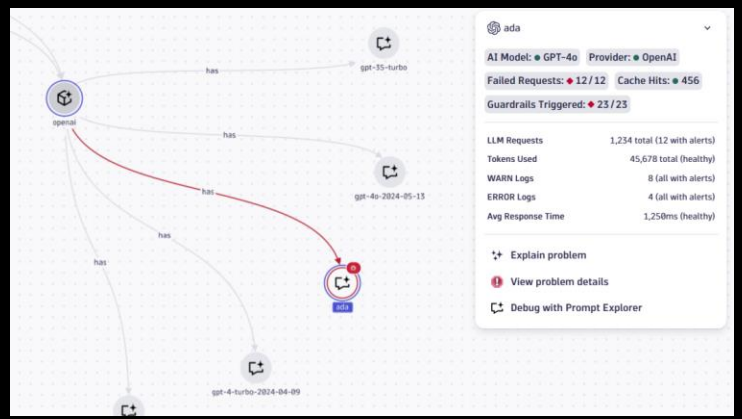
Its all Spans: but they tell a different story and enable different use cases

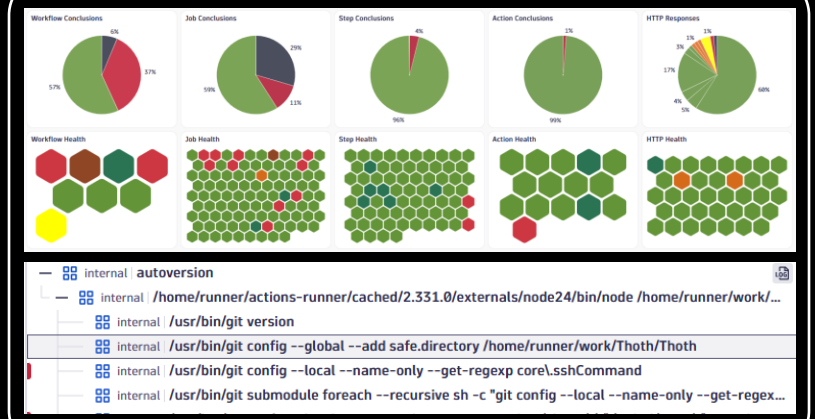
App Requests
User clicks on a link …

Agentic Workflow
Agent takes complex tasks

GitHub Workflow
Pull Request triggers CI/CD







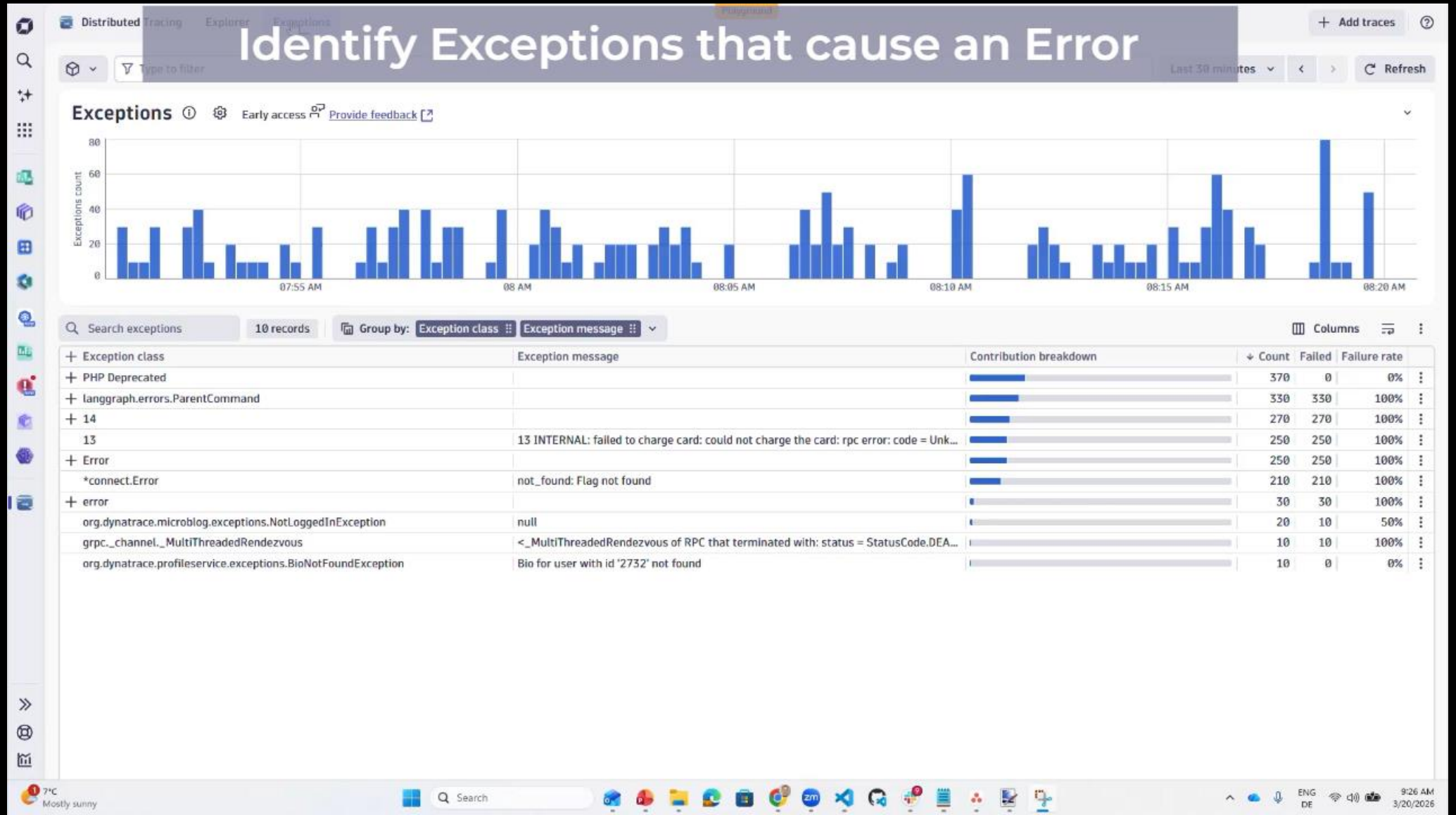
"Aah – the root CAUSE of my HTTP 500 is THIS exception!"

"Aah – THAT workflow called the SAME tool 10 times!"

"Aah – THAT GitHub action connects to the WRONG k8s!"

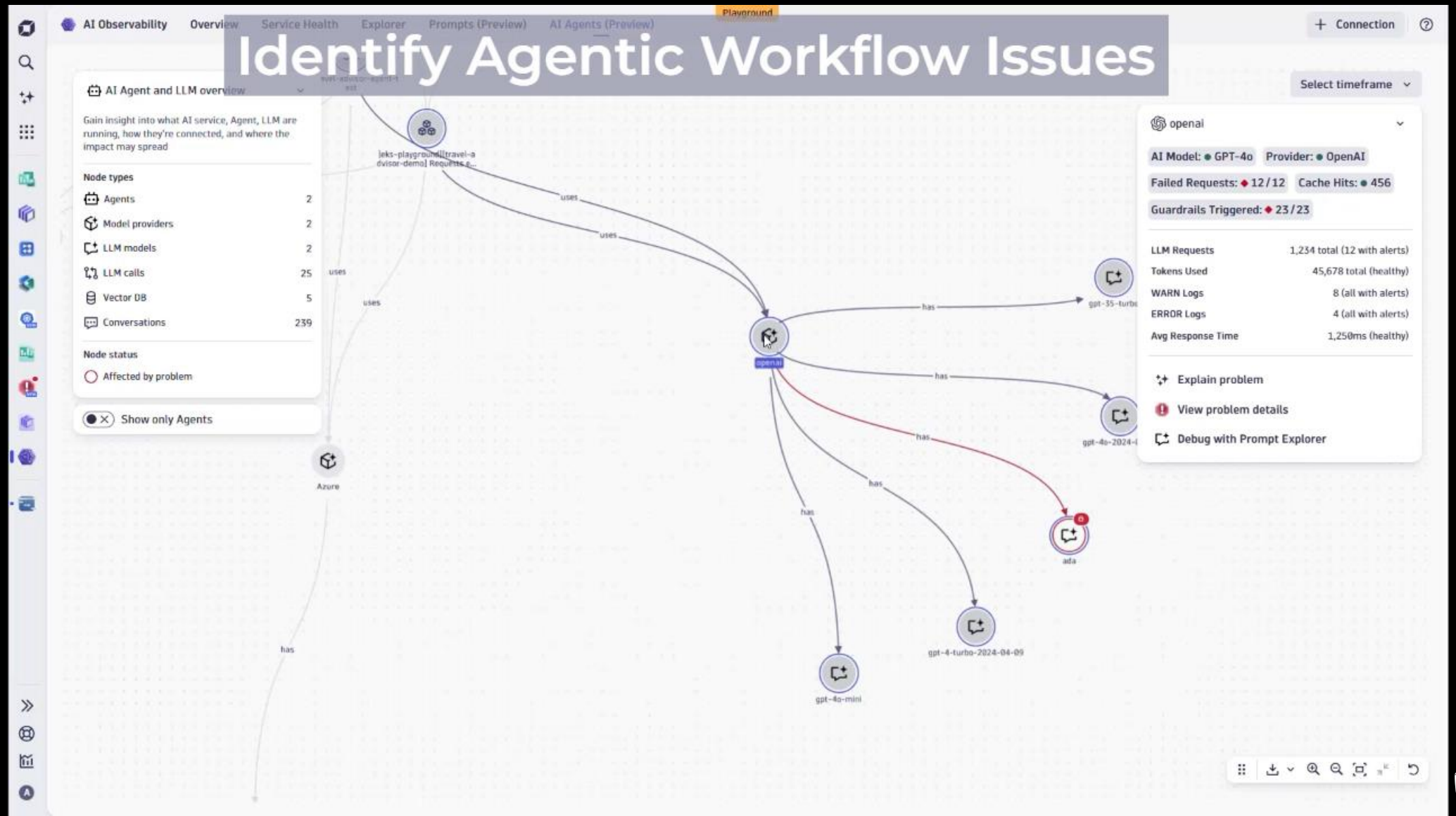
Type 1: Analyzing Root Cause of an App Request

What caused a problem? Exception, Database, CPU, Memory …



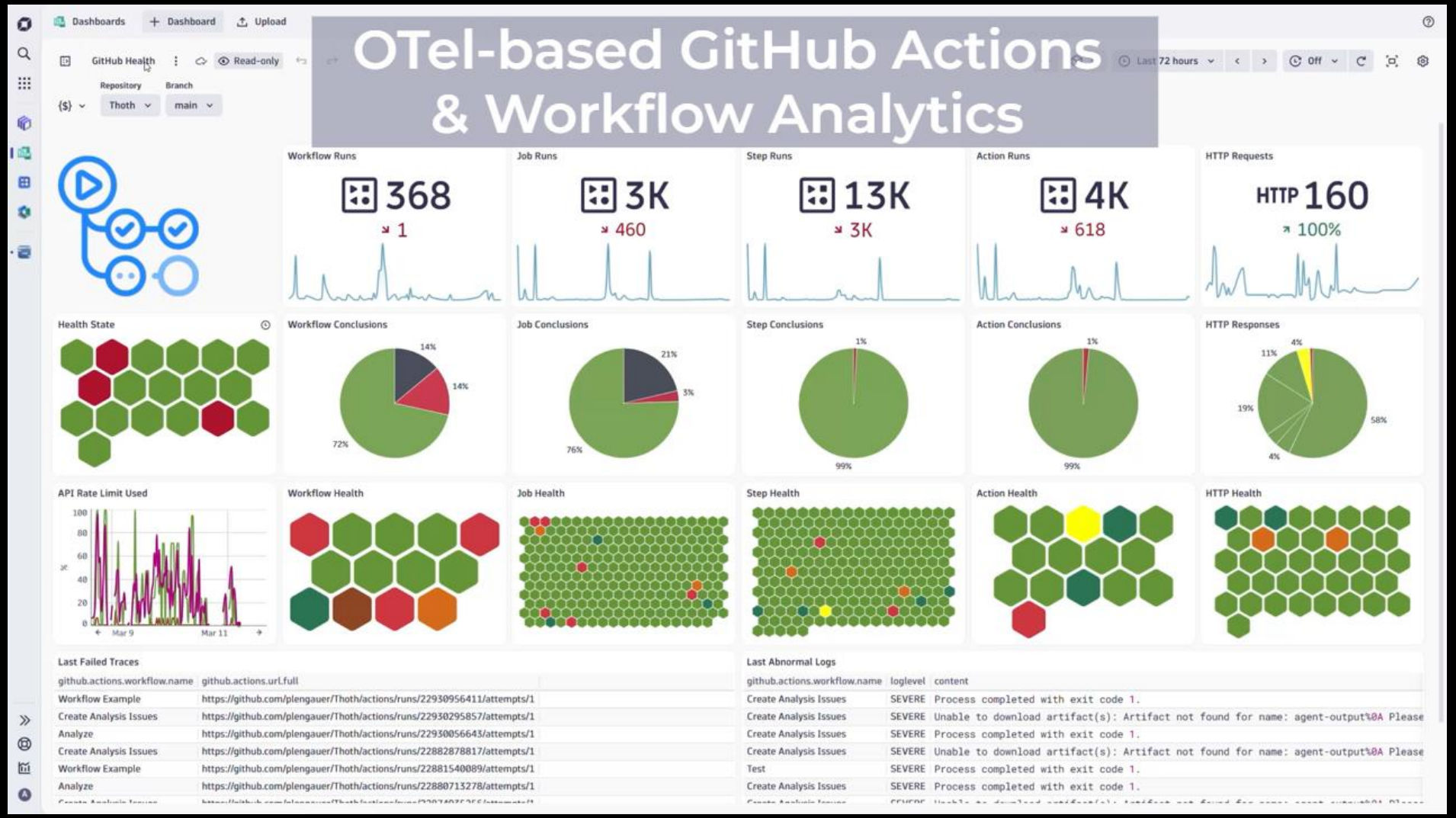
Type 2: Understand an Agentic Worklow

What tools are executed, which models used ...?



Type 3: GitHub Workflow Insights

Insights into your Workflows, Actions, Bash, Agents …

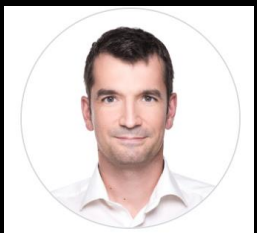


Philipp Lengauer
plengauer

https://dt-url.net/devrel-pl-github-thoth

Simulate your own OpenTelemetry

https://github.com/wolfgangB33r/otel-demo-service



Wolfgang Beer
wolfgangB33r

```
PS C:\GitHub\grabnerandi\otel-demo-service> python .\app.py
```

OTEL Demo Service Control

ai-agent-application

Purpose: simulate a LangGraph and LangChain RAG agent running an autonomous shopfront.
Topology: gateway, agent API, graph runtime, retrieval, tools, and commerce backends share one trace.
Signal shape: spans include GenAI model, token, cache, tool, and guardrail observability fields.
Failure modes: vector search slowness, LLM rate limits, tool failures, cache staleness, and checkout delays.
Best for: validating AI agent observability, RAG flow analysis, and autonomous commerce operations.

Path: scenarios/ai-agent-application.py
Status: Running
PID: 26296

Scheduled Problems

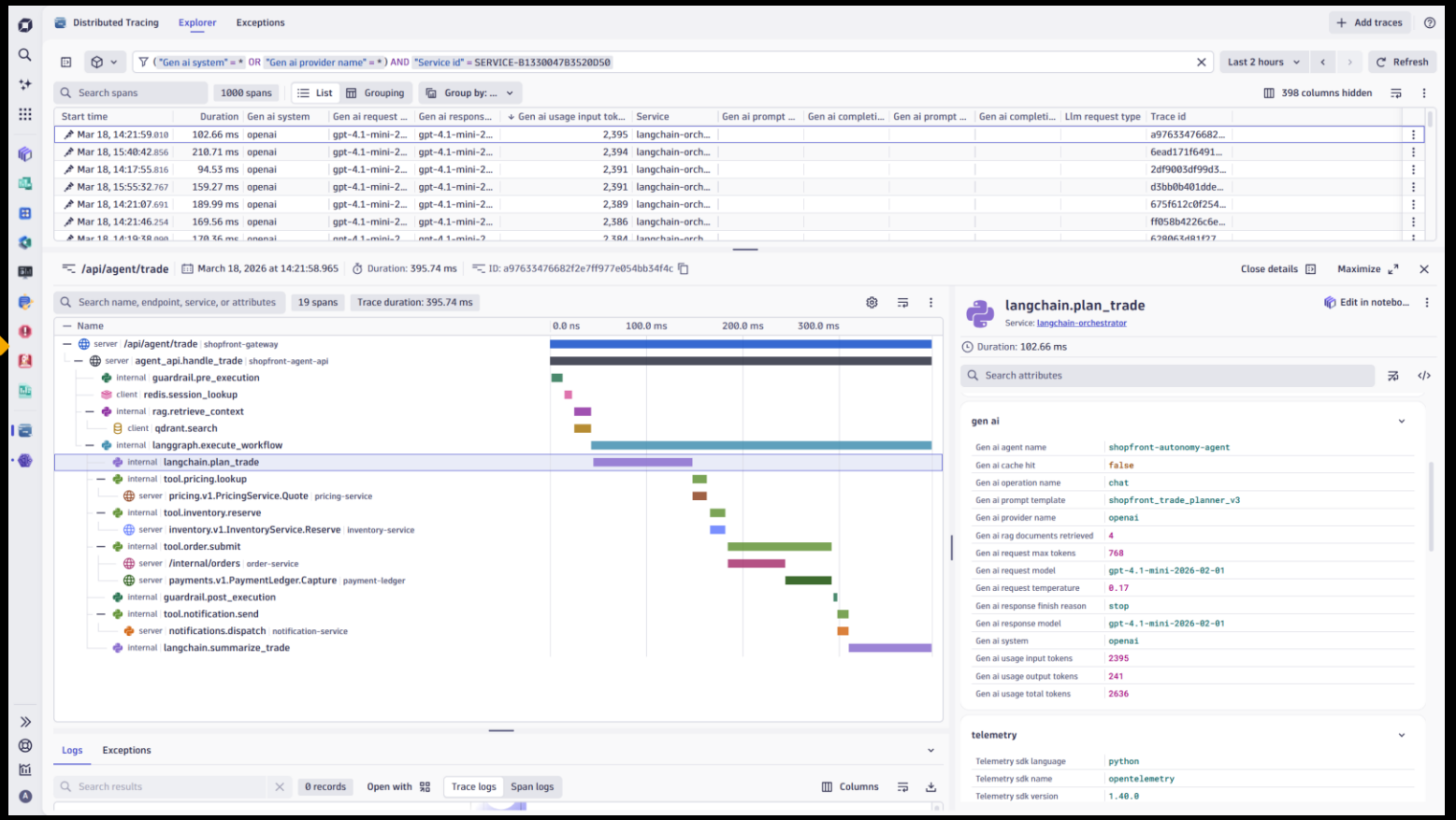
40 3 * * * → slow_vector_search for 60 min — Remove
40 3 * * * → llm_rate_limit for 60 min — Remove
40 3 * * * → tool_failures for 60 min — Remove
40 3 * * * → guardrail_blocks for 60 min — Remove
40 3 * * * → stale_inventory_cache for 60 min — Remove
40 3 * * * → checkout_latency for 60 min — Remove

slow_vector_search | Cron (e.g. 0 0 * * 1) | 60 | Add

Examples: Monday 00:00 = 0 0 * * 1, Tuesday 13:00 = 0 13 * * 2

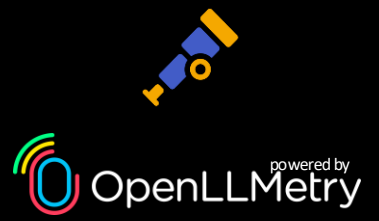
Request Rate — 10 req/min

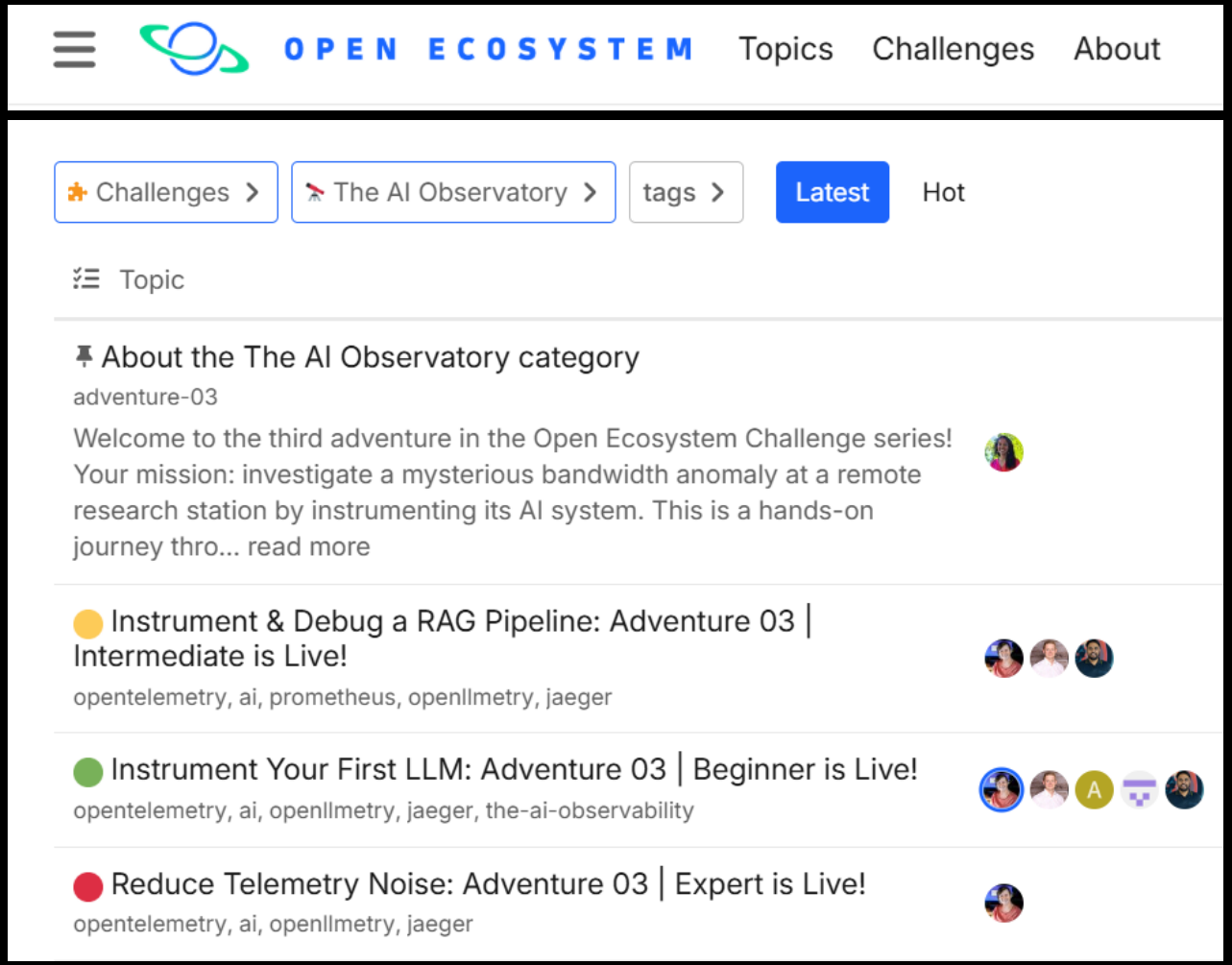
Start | Stop

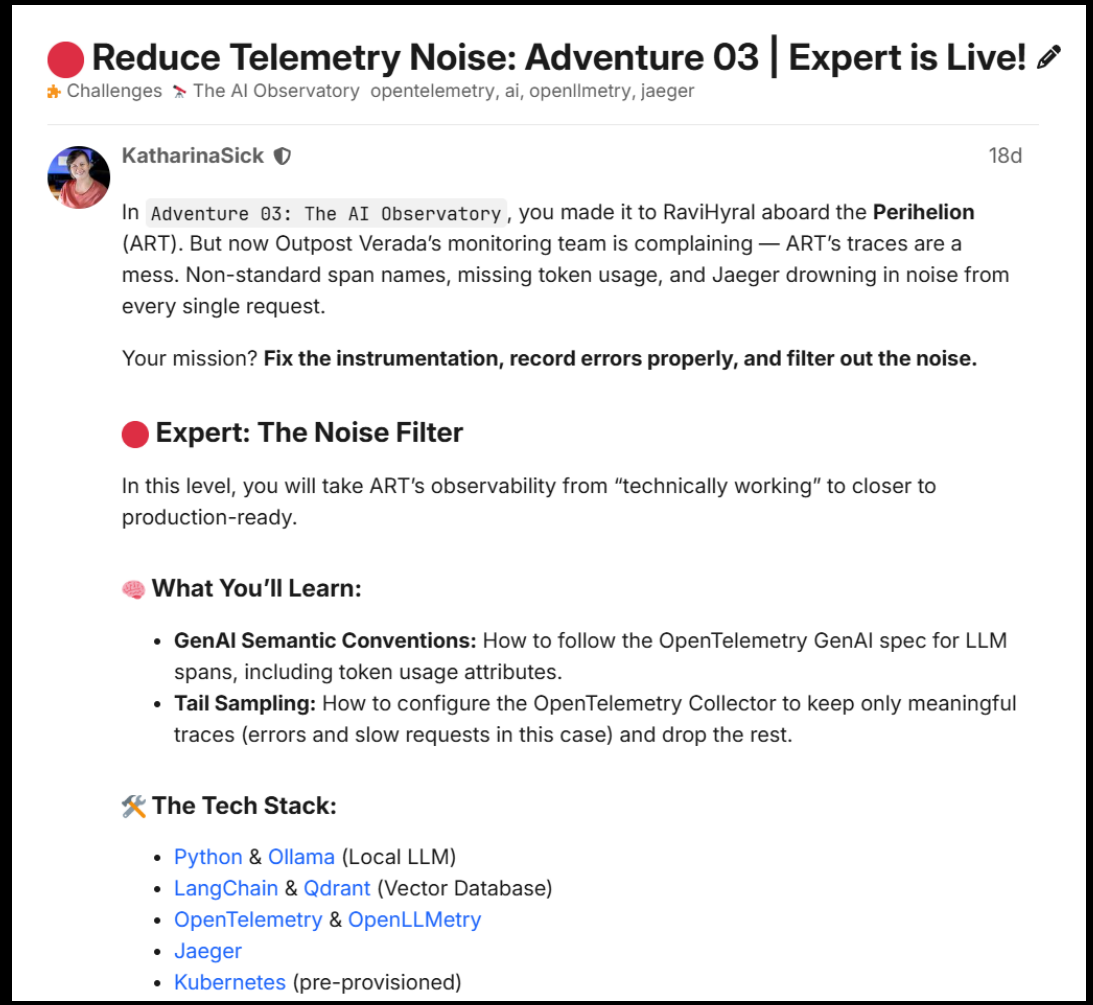


Instrument your own LLM Challenge!

https://community.open-ecosystem.com/





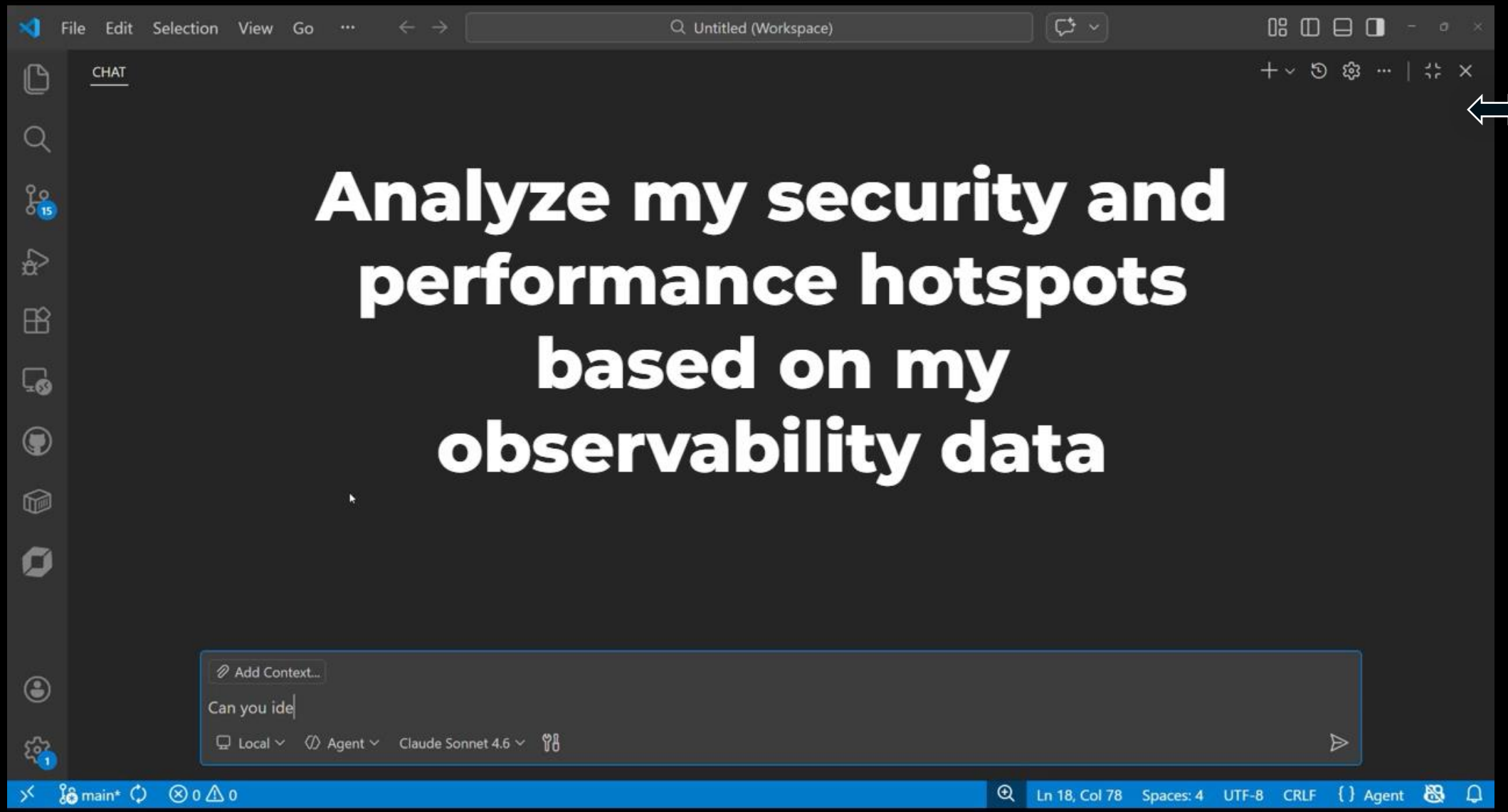


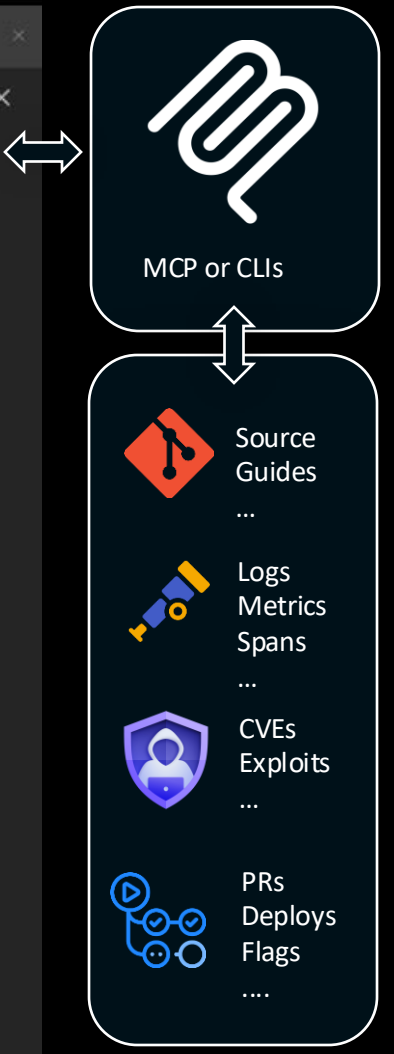
THE AI DELIVERY LIFECYCLE

OBSERVABILITY FOR AND WITH AI

Observability in the AI-Delivery Lifecycle

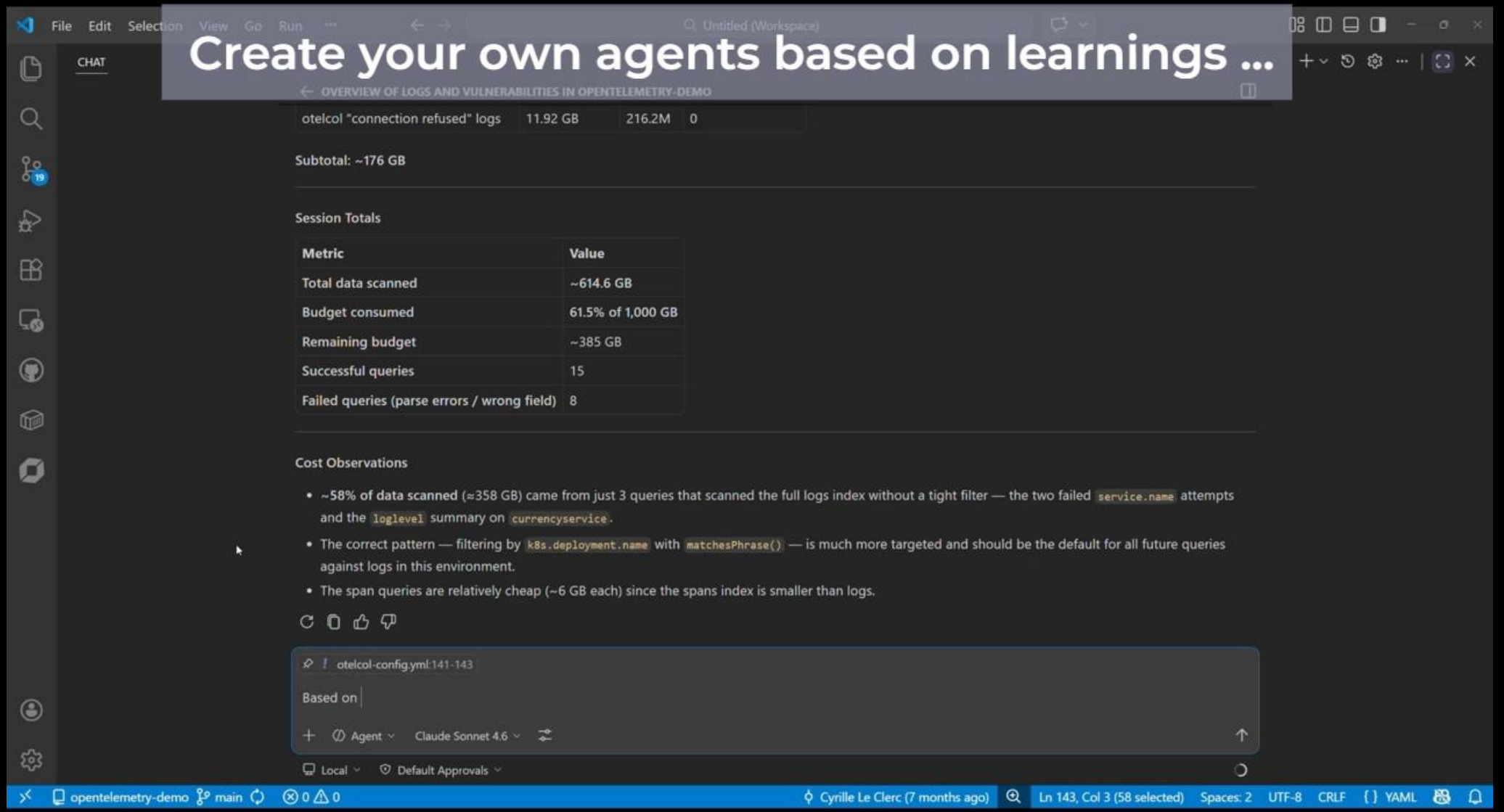
Provide code suggestions based on insights, change it, validate it …

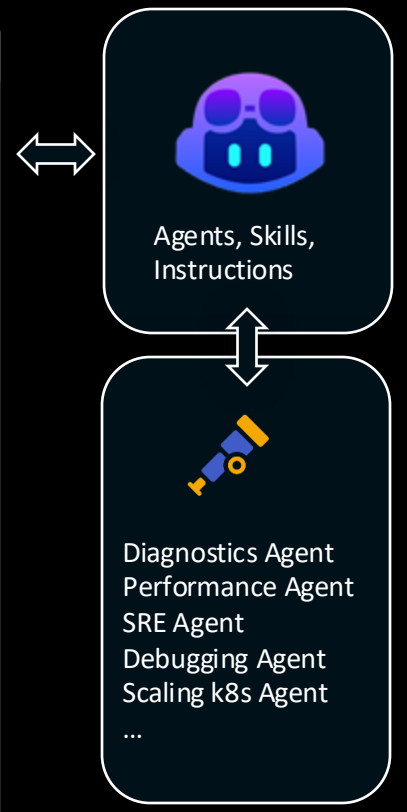




Observability in the AI-Delivery Lifecycle

Provide code suggestions based on insights, change it, validate it …



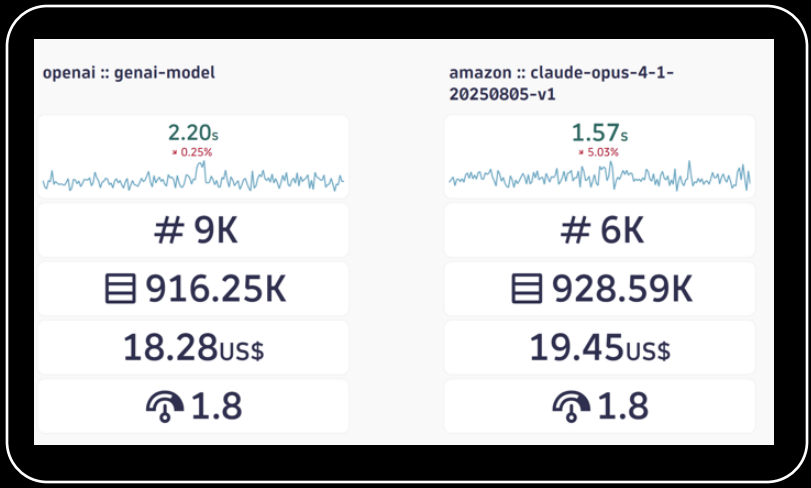


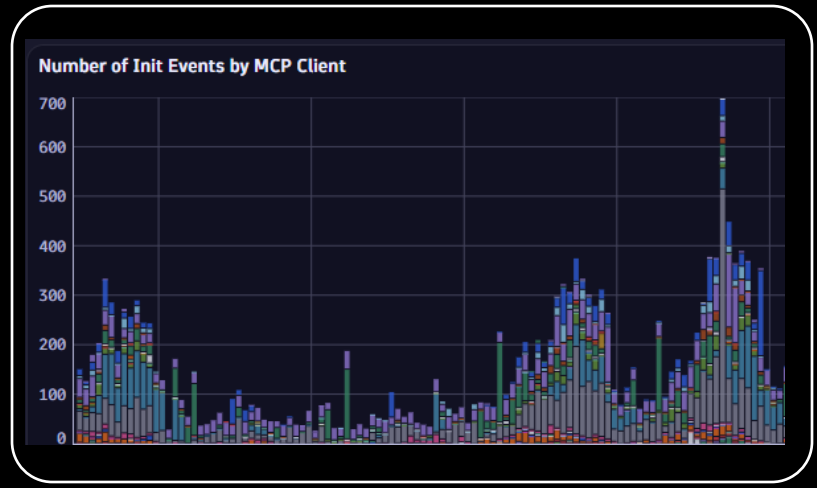
UNDERSTANDING EFFICIENCY, USAGE, GUARDRAILS, AND COSTS OF AGENTS

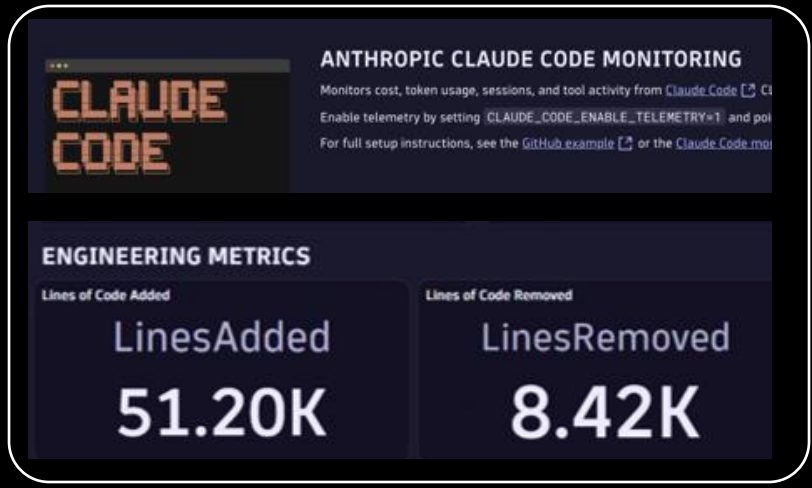
3 Lenses on AI Usage, Cost and Guardrails

Models, MCPs, Agents

Models	MCPs	Agents
A/B Testing for Use Cases	Adoption, Performance	Impact vs Cost



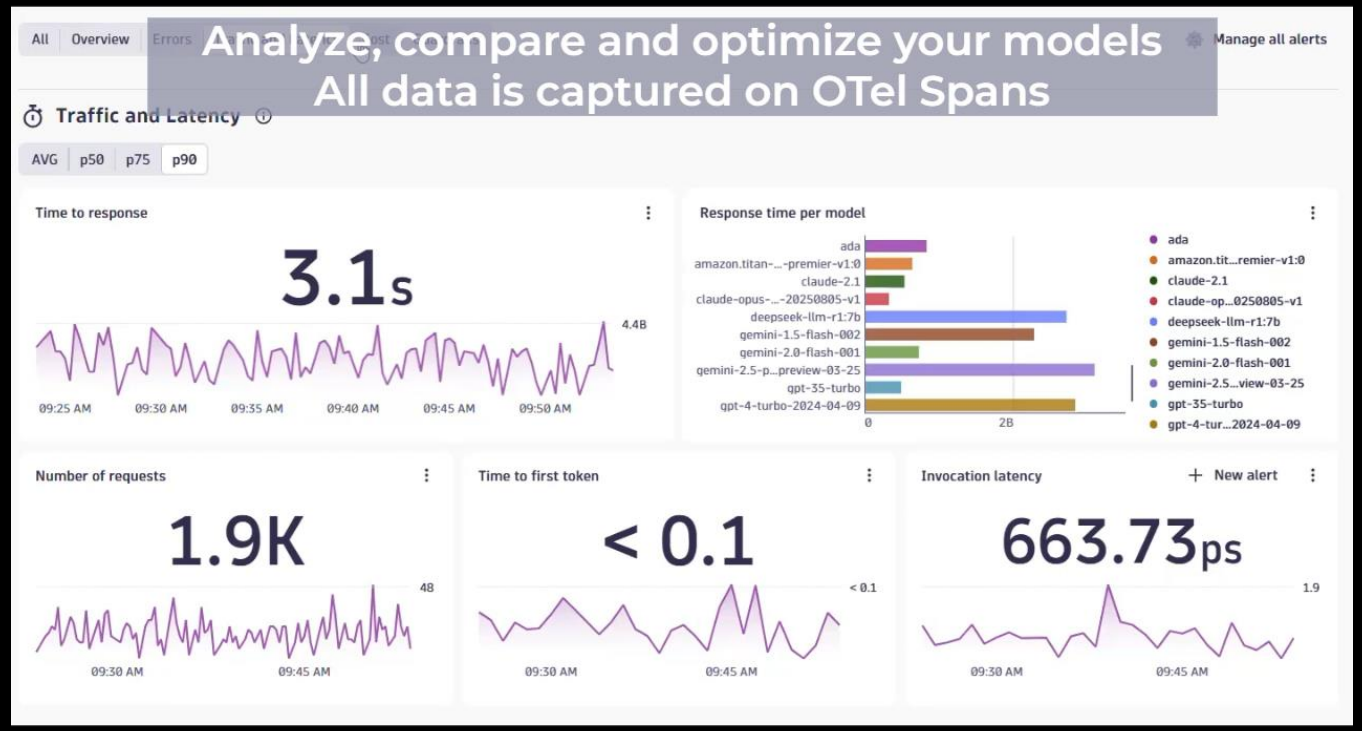


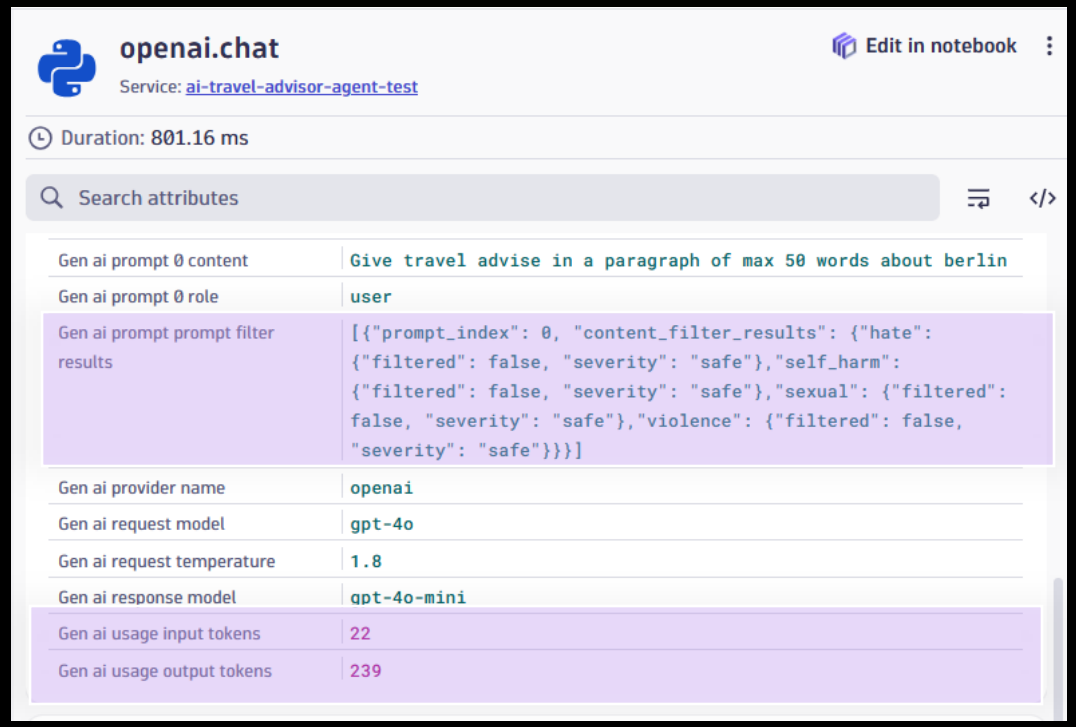


"Which model is best for every use case?"	*"Which tools are used? Which tools aren't? And Why?"*	*"Who is using this? And what's the ROI?"*

Analyzing different Models – A/B Testing

All data is on the spans or comes in through metrics

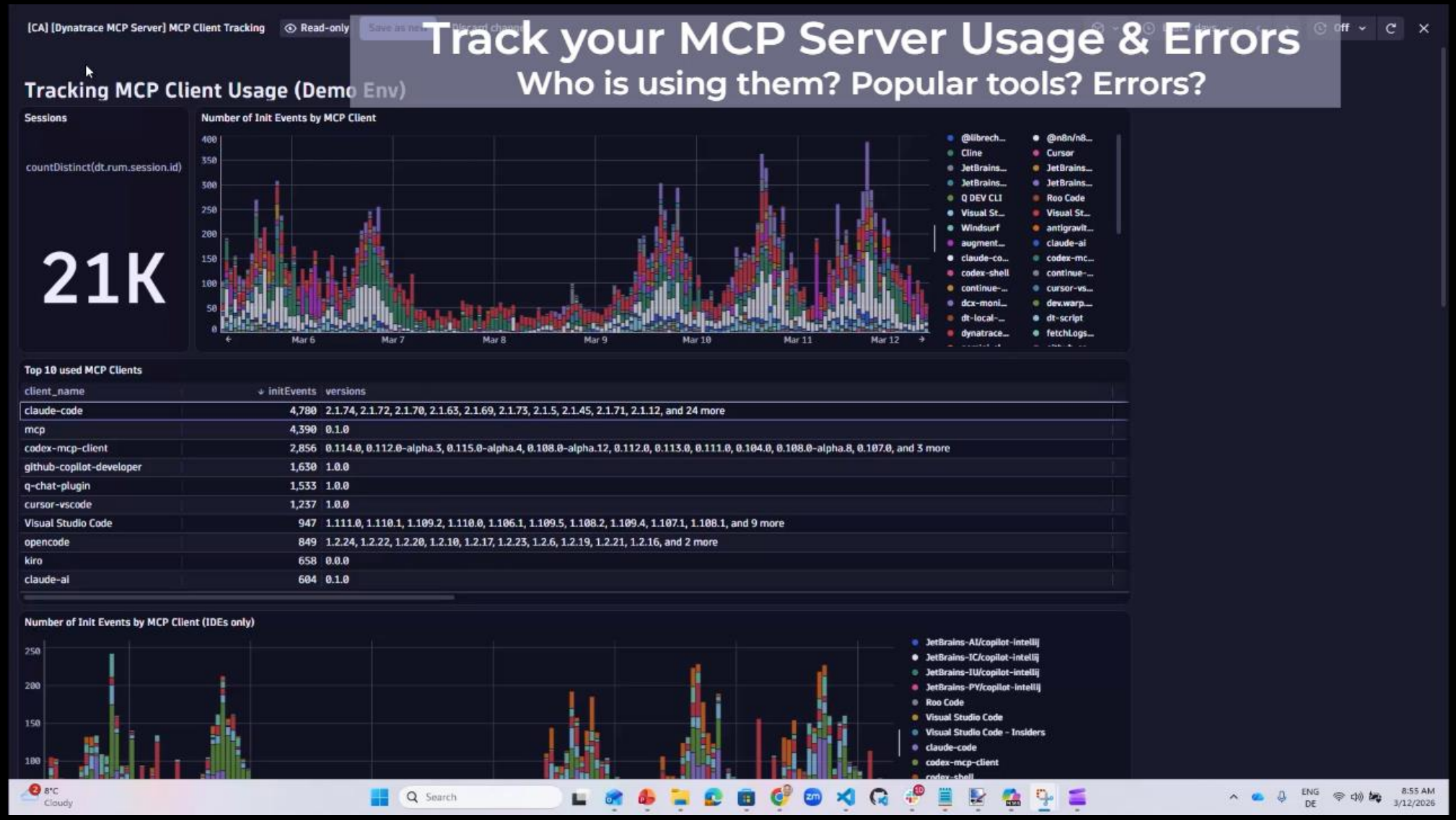




MCP Usage Observability

Understand how often your MCP is used by which agent, which tools …

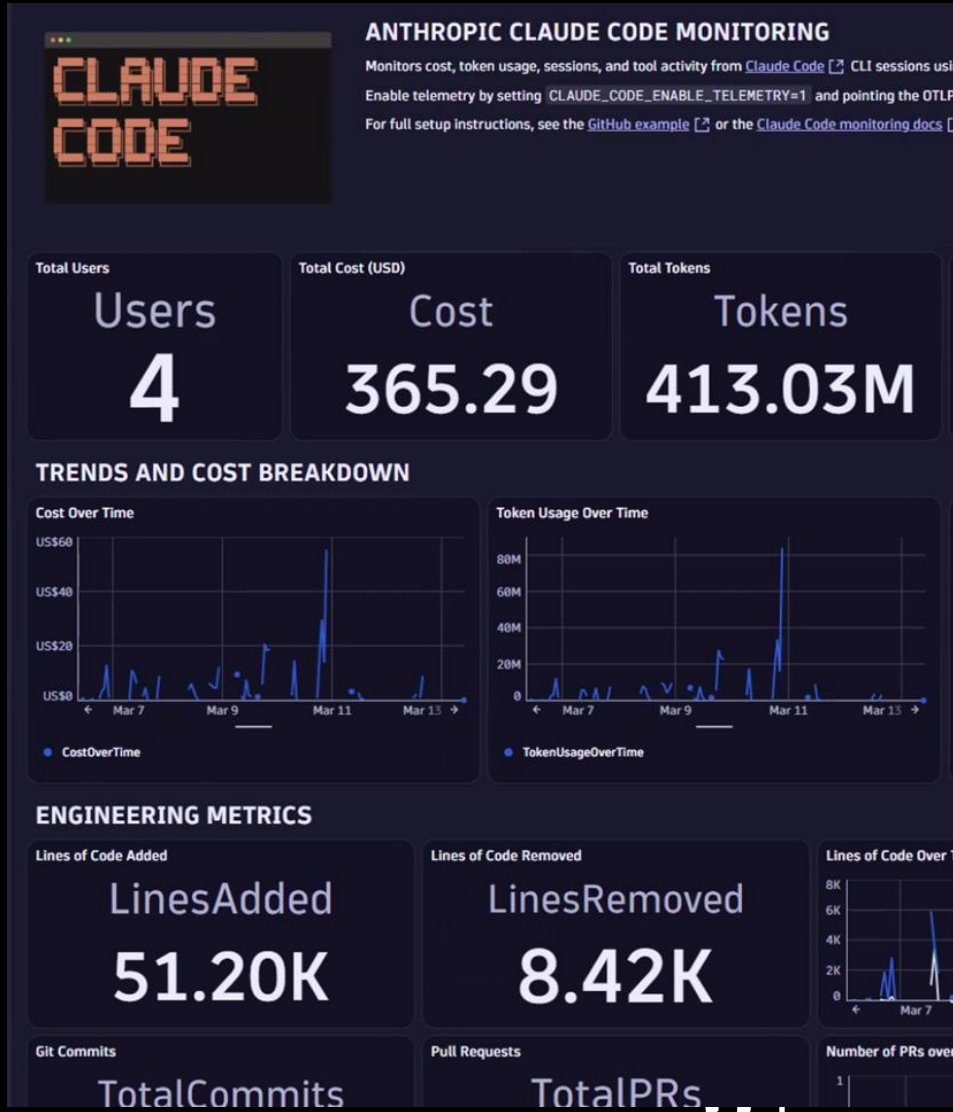
AI/LLM



MCP Client / Coding Agent Insights:

Coding Agents exposing OTel Metrics, Logs or provide an API to pull metrics

```
# 1. Enable telemetry
export CLAUDE_CODE_ENABLE_TELEMETRY=1

# 2. Choose OTLP exporters (Dynatrace ingests metrics and logs)
export OTEL_METRICS_EXPORTER=otlp
export OTEL_LOGS_EXPORTER=otlp

# 3. Use HTTP/protobuf – the protocol Dynatrace expects
export OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf

# 4. Point to your OTLP Endpoint
export OTEL_EXPORTER_OTLP_ENDPOINT=https://...

# 5. Add Authentication Header
export OTEL_EXPORTER_OTLP_HEADERS="Authorization=xxx"

# 6. Specify metric export cumulative/delta
export OTEL_EXPORTER_OTLP_METRICS_TEMPORALITY_PREFERENCE=delta
```



Adoption and Impact Analytics at Scale

See trends, tool usage, identify hotspots, …

GET /orgs/{org}/copilot/metrics


```
{
  "date": "2024-06-24",
  "total_active_users": 24,
  "total_engaged_users": 20,
  "copilot_ide_code_completions": {
    "total_engaged_users": 20,
    "languages": [
      {
        "name": "python",
        "total_engaged_users": 10
      },
      {
        "name": "ruby",
        "total_engaged_users": 10
      }
    ],
    "editors": [
      {
        "name": "vscode",
        "total_engaged_users": 13,
        "models": [
          {
            "name": "default",
            "is_custom_model": false,
            "custom_model_training_date": null,
            "total_engaged_users": 13,
            "languages": [
              {
                "name": "python",
                "total_engaged_users": 6,
```

DO WE NEED AGENTS FOR EVERYTHING?

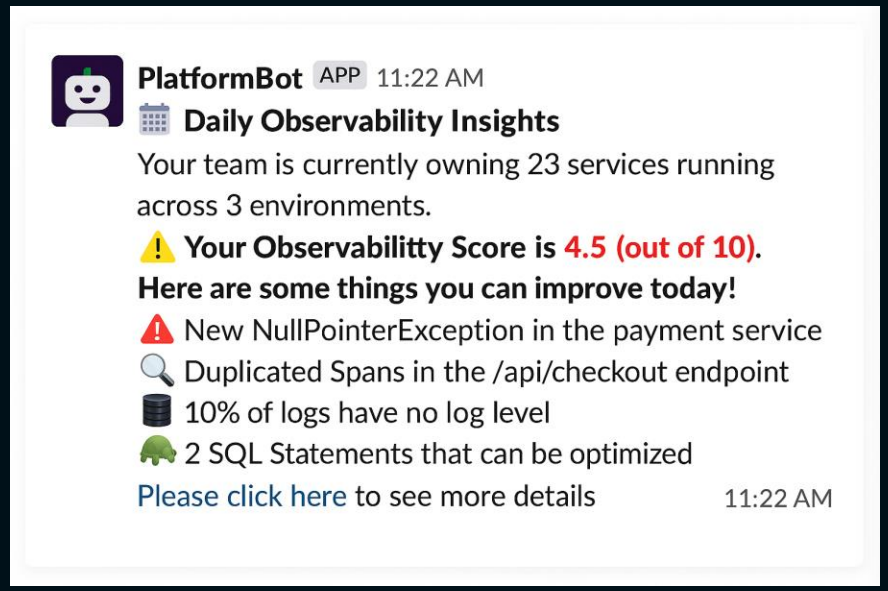
Detecting bad patterns in observability data is simple!

Duplicated Spans, excessive logs, unexpected dependencies …

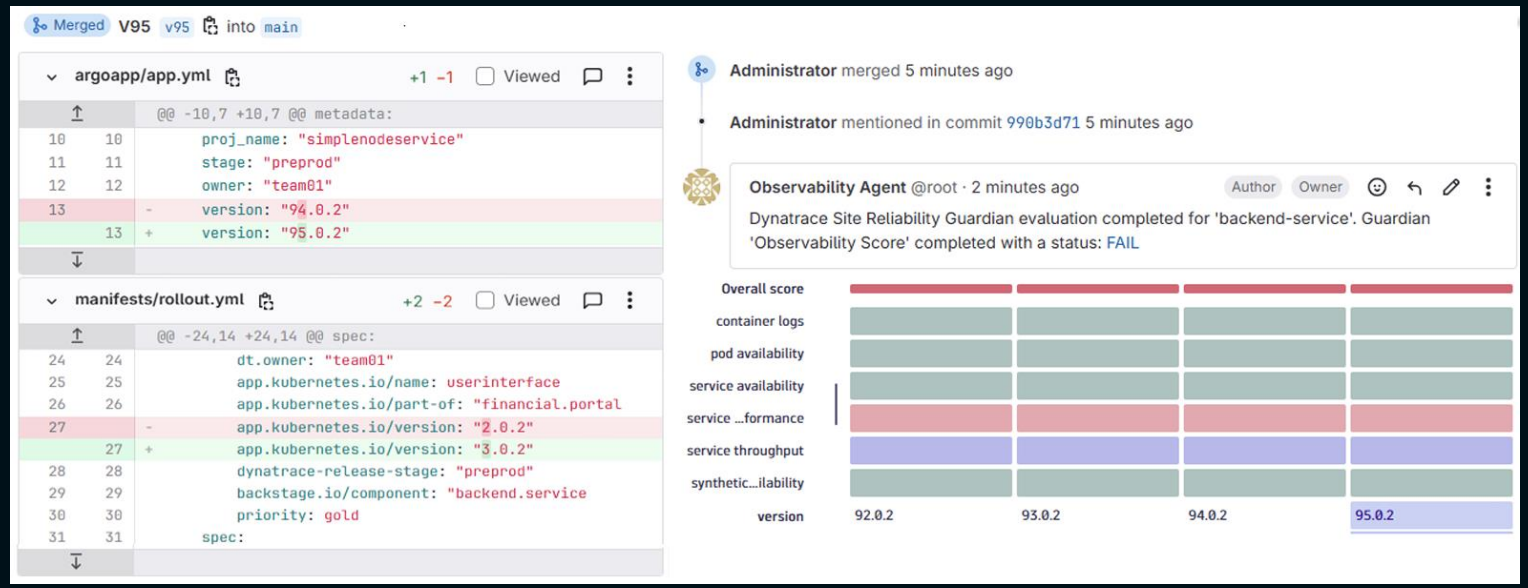
Duplicated Spans

Excessive Logs

Automate through a Daily Standup Reminder

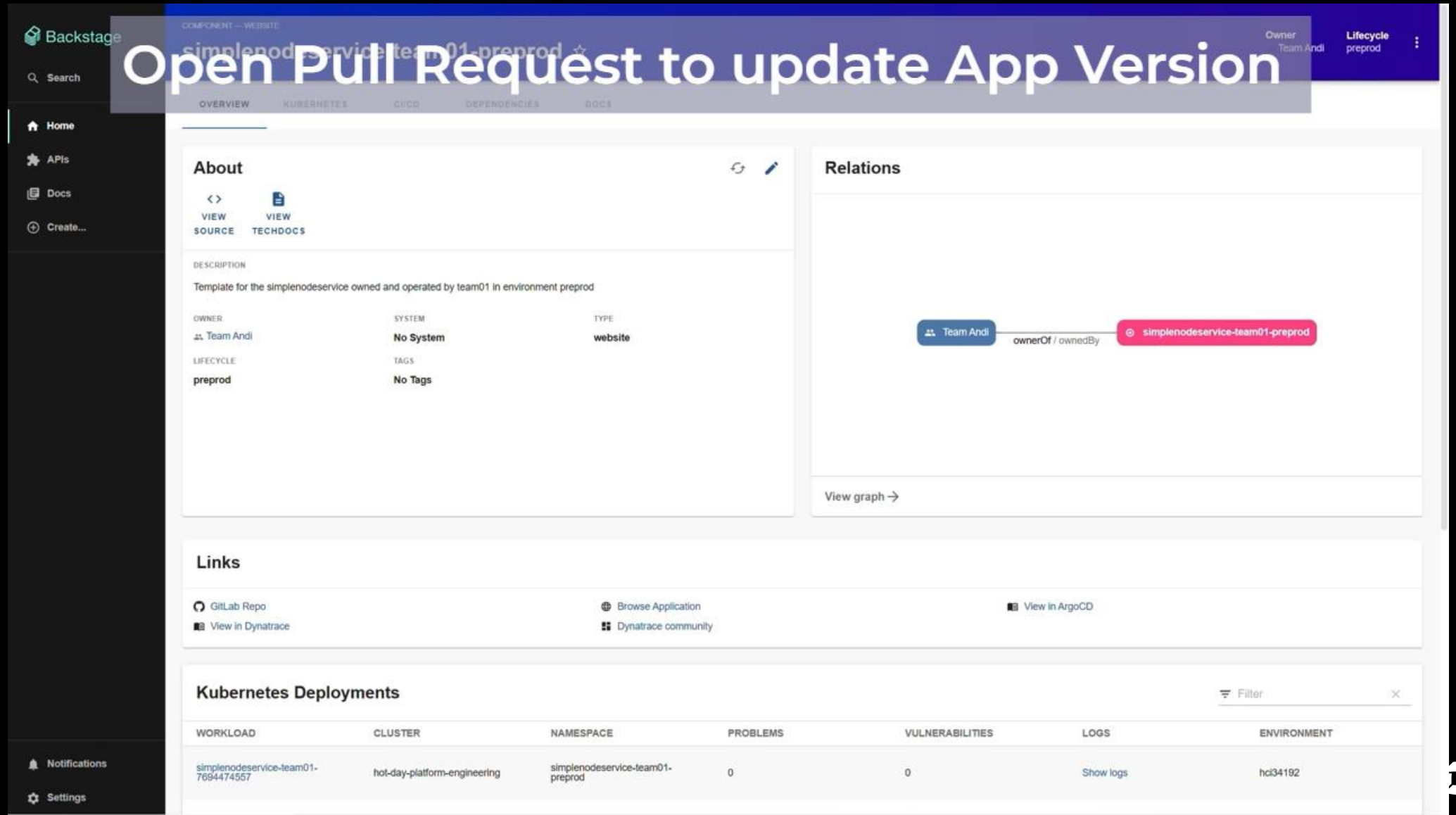


Automate and Push the results back to your Pull Request!



Demo: Observability in your PR-Flow

We don’t need to prompt an agent to ask for feedback



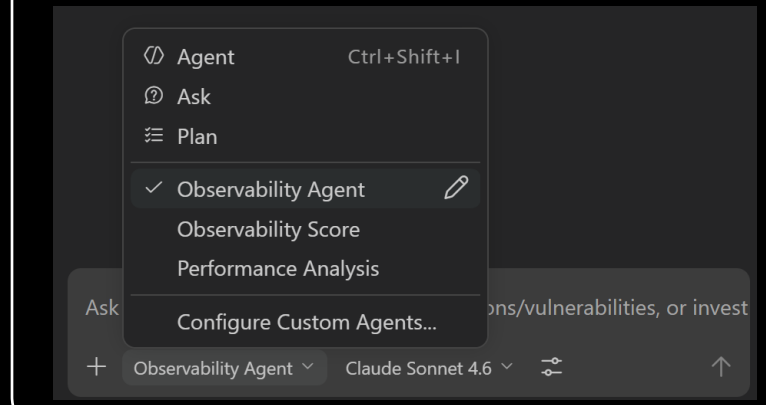
I am done now – so …

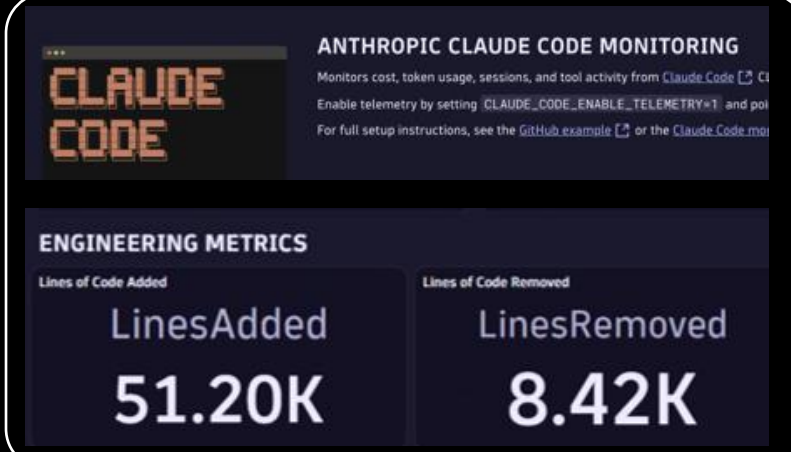
3+1 Things I hope you took away from this today

The Basics
Observability Standards

The AI Delivery Lifecycle
Observability for and with AI

Cost and Impact
Optimize usage and alternatives





"Observability beyond apps: AI, Workflows, CI/CD …"

"Integrate Observability Loops into your AI Development"

"Observability helps you optimize usage and impact!"

Remember

"Some problems can be solved with simple automation!"

Thank You!

Andi (Andreas) Grabner

- CNCF Ambassador
- Fellow DevRel @ Dynatrace
- PurePerformance Podcaster
- Occasional Book Author
- Salsa Dancer