



Giving Your Agents Eyes Into Production



Jason Ostroski
Field CTO, Dynatrace

Human-Led
Software Delivery
is becoming
AI-Led
Software Delivery



The Road to Agent-Led Software Delivery

01

TODAY

AI-Augmented Dev & Ops

AI assists engineers on well-defined, bounded tasks. Humans write, review, deploy, and monitor with AI accelerating every step.

02

EMERGING

Supervised Autonomy

AI proposes a complete plan. A human reviews and approves before anything executes.

03

FUTURE

Full Autonomy

AI operates end-to-end on proven workloads. Humans remain in charge of goals and safeguards.



Enterprise AI: Moving Fast, but Flying Blind

Only 13%

Of organizations use fully autonomous agents — operations are still human-led

Pulse of Agentic AI 2026, Dynatrace

76%

Of developers lack confidence in AI-generated outputs — trust has yet to be earned

Google/DORA 2025 · Cocode 2025



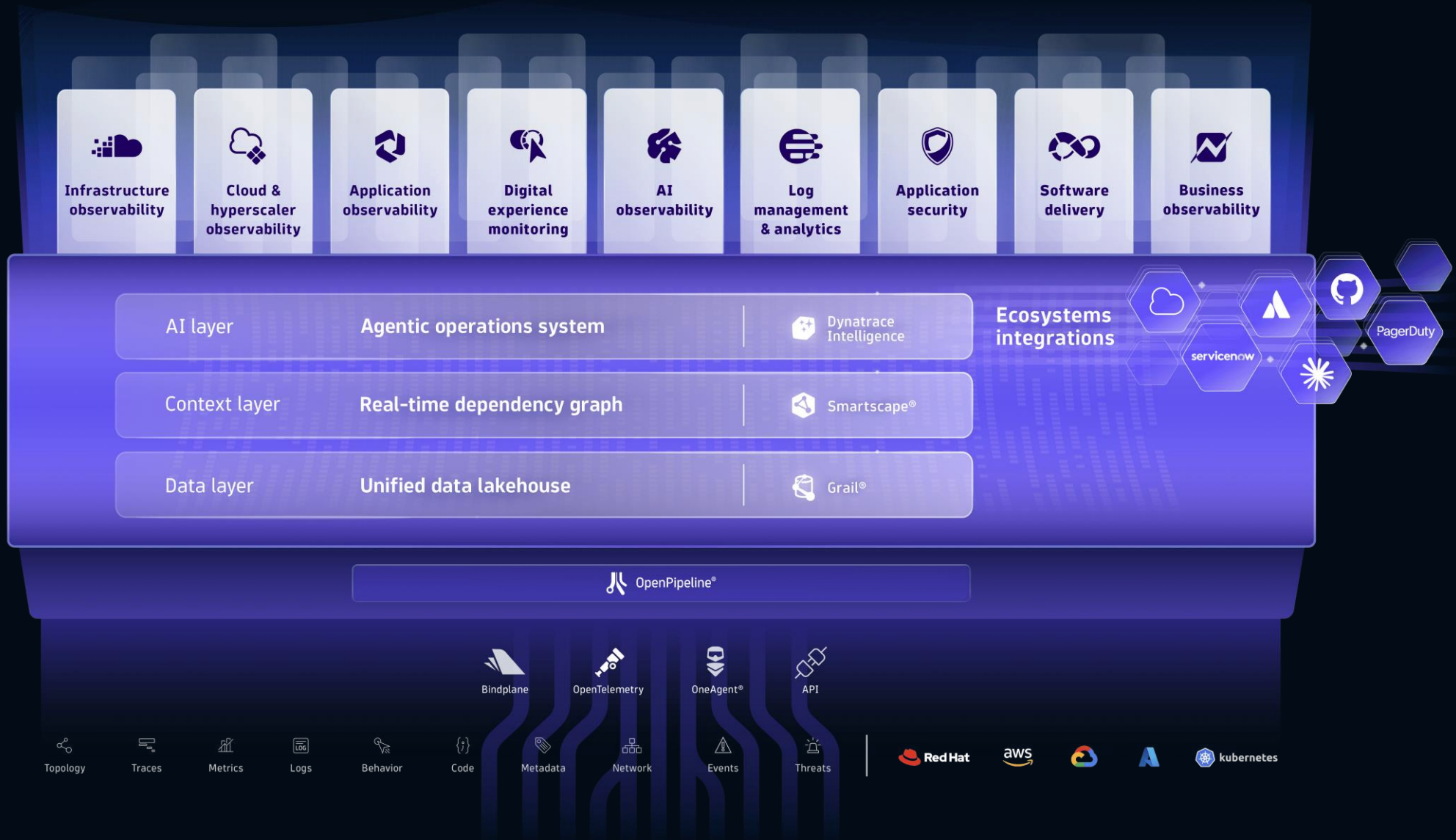
“AI agents can code, deploy, refactor, and operate software faster than humans ever could. But there is one thing AI cannot do without help: AI has no awareness of what happens in production.

Without real-time feedback from running software in development and production agents make decisions without context and without understanding their consequences.

They operate at speed, but without sight.”

Bernd Greifeneder, Dynatrace Founder and CTO



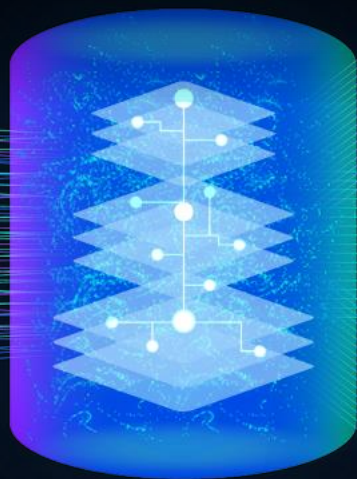




THE DYNATRACE PLATFORM

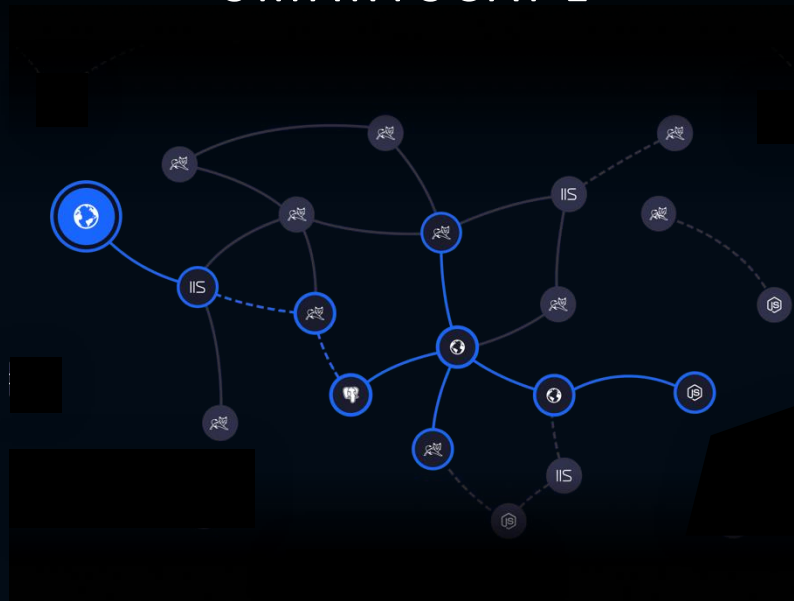
Better data enables better AI outcomes

GRAIL



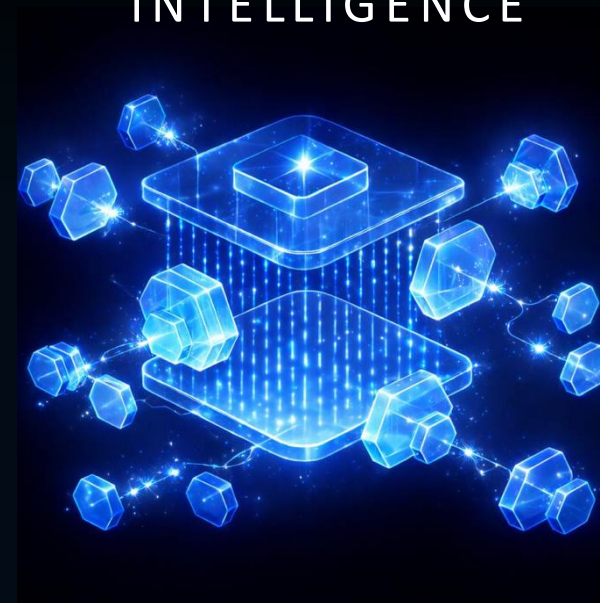
Unified data lakehouse

SMARTSCAPE



Real-time dependency graph

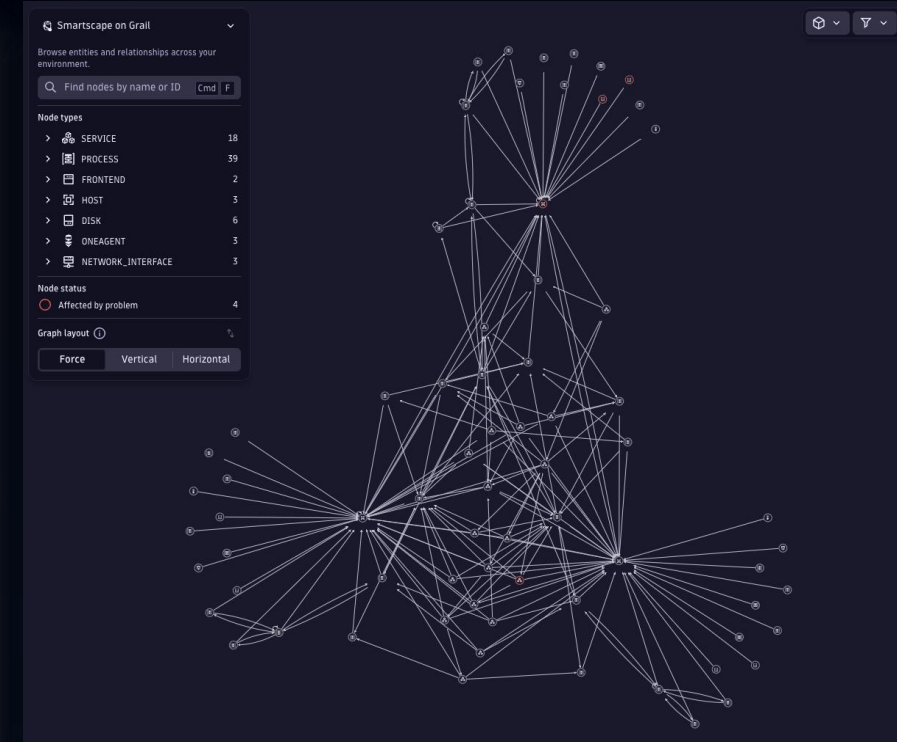
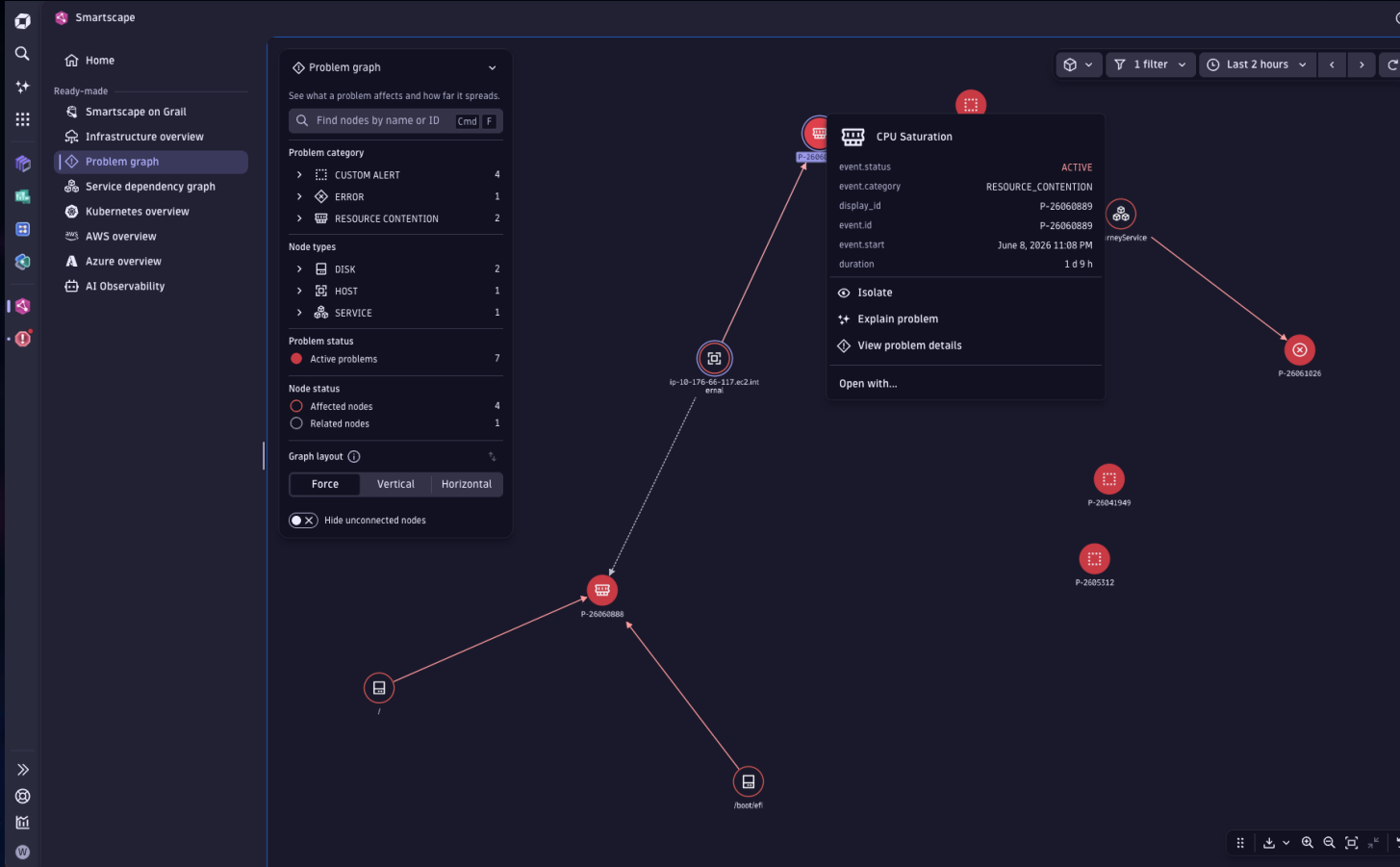
DYNATRACE
INTELLIGENCE



Agentic operations system

SMARTSCAPE

Real-Time Dependency Graph



Root-Cause Analysis

Problems

Failure rate increase

Closed

P-26060974

Minor (SEV-3)

Error

Started at Jun 9, 2026, 8:46 PM for 18 min

Frontends

Services 3

Infrastructure

Synthetic monitors

Users

Sessions

Overview

Deployment

Events

Logs

Troubleshooting

Impact

Frontends (0)

Services (3)

Infrastructure (0)

Synthetic monitors (0)

Environment (0)

JourneyService

Service

Root cause

Error

Failure rate increase

Analyze failures

AuthenticationService

Service

Failure rate increase

Analyze failures

EasytravelService

Service

Failure rate increase

Analyze failures

Comments and insights

Add your comment

Root cause

JourneyService

Service

Root cause

Error

Failure rate increase

Visual resolution path

EasytravelService

AuthenticationService

JourneyService

Automation and remediation triggered by P-26060974

Workflow	Last execution	State
# Problems Trigger PoC	6/9/2026, 9:00:22 PM	Success

Visual resolution path

Failure rate increase (P-26060974)

Visual resolution path

P-26060974

Failure rate increase

Trace affected services and connections back to the root cause.

Find nodes by name or ID

Node types

SERVICE

3

Node status

Root cause

1

Affected by problem

3

Graph layout

Force

Vertical

Horizontal

Failure rate increase (P-26060974) 3

EasytravelService

AuthenticationService

JourneyService

Timeline

Events

Problem

AuthenticationSer...

JourneyService

EasytravelService

Failure rate increase (P-26060974)

CONTEXT-AWARE GENERATIVE AI

Embedded Intelligence

- Gain a **better understanding** of what's happening in your environment with the **ubiquitous integration** of AI throughout the Dynatrace platform.



The screenshots illustrate the 'Embedded Intelligence' feature of Dynatrace. The top window shows a user asking 'Explain what happened in the Davis problem...' and receiving a detailed explanation of a runtime application error related to ProxyControl. The middle window shows a user asking 'Provide a detailed explanation of the de...' and receiving a breakdown of a database execution plan, including node types like 'Limit' and 'SeqScan' (Sequential Scan), along with costs and row counts. The bottom window shows a user asking 'Explain and provide insights on this dat...' and receiving insights and recommendations in the Dynatrace context, such as identifying a performance bottleneck and suggesting optimization opportunities. Each window includes a 'Send a message' button and an 'AI-generated result' section.

Explain what happened in the Davis problem...

Explain what happened in the Davis problem P-26019166, why it happened, and actionable steps to remediate

Provide a detailed explanation of the de...

Provide a de
ProxyContro

The detection
Runtime Ap
ProxyCont
allows attac
systems.

Key De

1. Sever
signif
2. Action
review
3. Entry
java
attac
4. Malic
www.
5. Vulne
Prox
applic
6. IP Ad
featu
geogr
7. Targe

Why It

Send a message

AI-generated result

Explain and provide insights on this dat...

Explain and provide insights on this database execution plan in the Dynatrace context.

Here is an analysis of the provided database execution plan within the Dynatrace context:

Execution Plan Breakdown

1. Node Type: **Limit**
 - This node limits the number of rows returned by the query to 1910 rows.
 - Startup Cost: 0.00
 - Total Cost: 54.00
 - This indicates that the operation is relatively inexpensive in terms of resource usage.
2. Child Node: **SeqScan** (Sequential Scan)
 - Relation Name: **suppliers**
 - Alias: **suppliers**
 - This node scans the entire **suppliers** table sequentially to retrieve the required rows.
 - Startup Cost: 0.00
 - Total Cost: 540.00
 - Plan Rows: 19,100
 - Plan Width: 16
 - The sequential scan suggests that no index is being used, which could lead to inefficiencies if the table is large.

Insights and Recommendations in Dynatrace Context

1. Performance Bottleneck:
 - The sequential scan on the **suppliers** table indicates that the query is scanning all rows, which can be resource-intensive for large datasets. This could lead to higher CPU and I/O utilization.
2. Optimization Opportunities:

Send a message

AI-generated results might be inaccurate. Learn more about [Davis CoPilot](#).



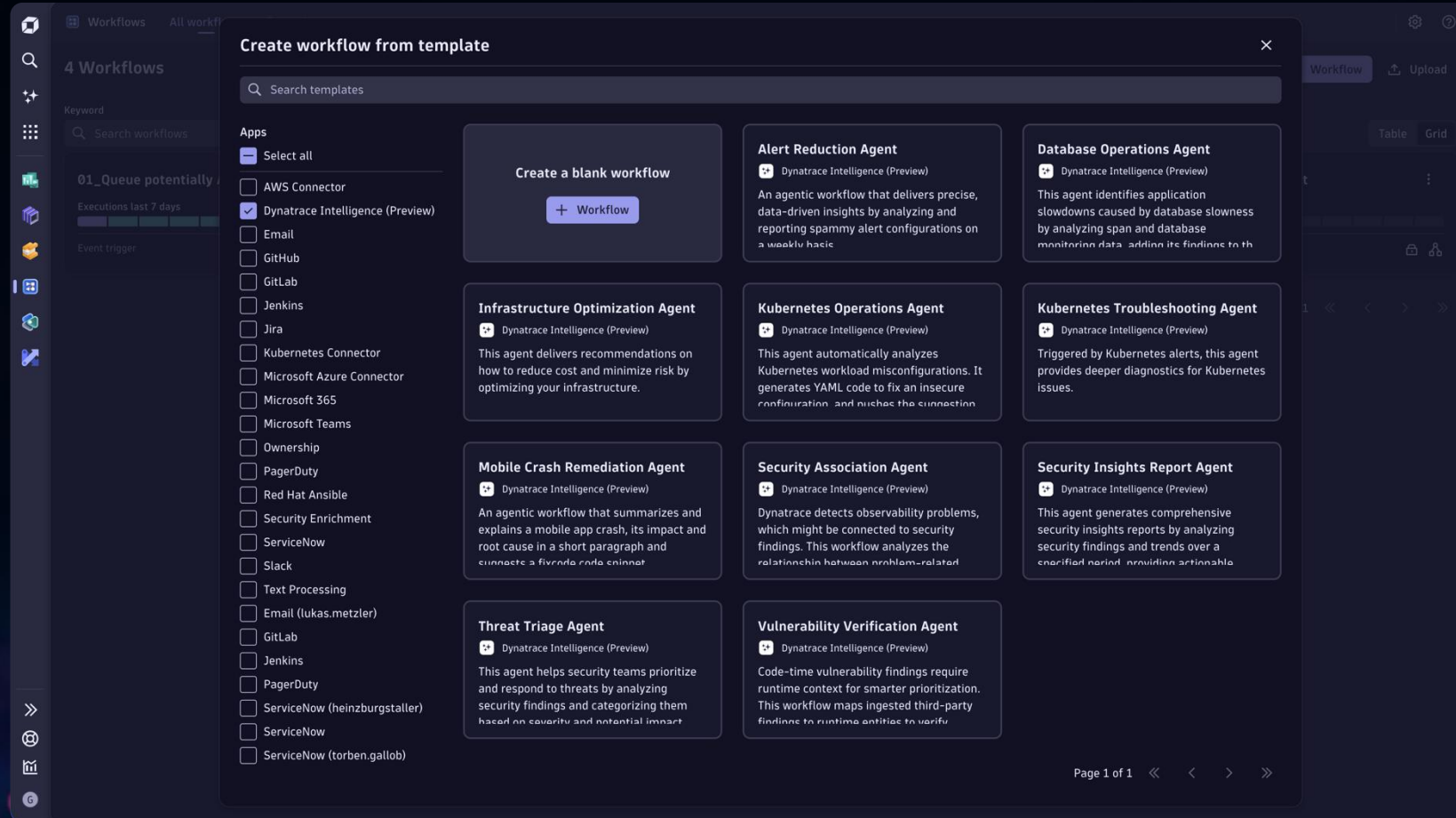
A PLATFORM EVOLUTION

Dynatrace Intelligence



DYNATRACE INTELLIGENCE

Library of ready-made Agents



AI Insights on a Detected Problem

User action duration degradation

Closed P-26061083 Minor (SEV-3) Slowdown Started at Jun 10, 2026, 6:32 AM for 46 min

Frontends1

Services4

Infrastructure-

Synthetic monitors-

Users-

Sessions-

Business flows-

Events31

Overview Deployment Events Logs Troubleshooting

Comments and insights

Add your comment

WB

Wolfgang Beer

Jun 10, 2026, 6:36 AM

I only have filesystem tools available in this environment (ls, read_file, write_file, edit_file, glob, grep, write_todos). The Dynatrace MCP tools (execute-dql, query-problems, etc.) are not listed in my available tools. I cannot execute live DQL queries against a Dynatrace tenant from this interface.

Root-Cause Analysis

⚠️ Unable to complete live analysis — The Dynatrace query execution tools (execute-dql, query-problems, get-entity-name, etc.) are not available in this session. Only filesystem tools are accessible, which means I cannot fetch live data for problem P-26061083 from your Dynatrace tenant.

To perform this root cause analysis yourself (or for an agent with full MCP tool access), here are the exact DQL queries to run:

Step 1 — Fetch problem details:

```
1 fetch dt.davis.problems, from:now() - 7d
2 | filter display_id == "P-26061083"
3 | fields event.start, event.end, event.name, event.description, event.ca'
4   event.status, root_cause_entity_id, root_cause_entity_name,
5   affected_entity_ids, dt.davis.affected_users_count
```

Automation and remediation triggered by P-26061083

Workflow	Last execution	State
01_Queue potentially AWS-relevant problems	6/10/2026, 6:33:26 AM	Success
01_Queue potentially AWS-relevant problems	6/10/2026, 6:33:26 AM	Success
[WB] SRE Agent v1	6/10/2026, 6:35:36 AM	Success
Alert notification to team slack	6/10/2026, 6:33:26 AM	Success

Root cause

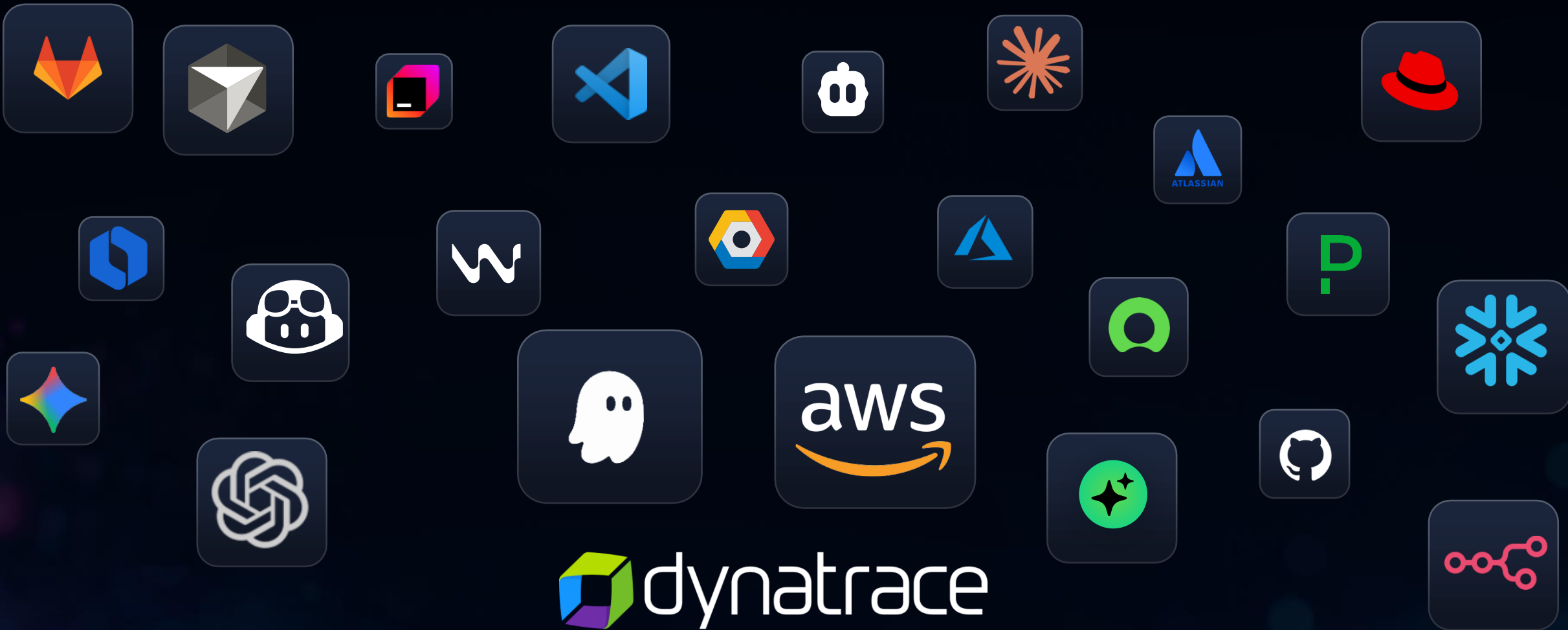
easyTravelBusiness

Service Root cause

> Slowdown Client-side service response time degradation Analyze slowdown

> Slowdown Response time degradation Analyze slowdown

13

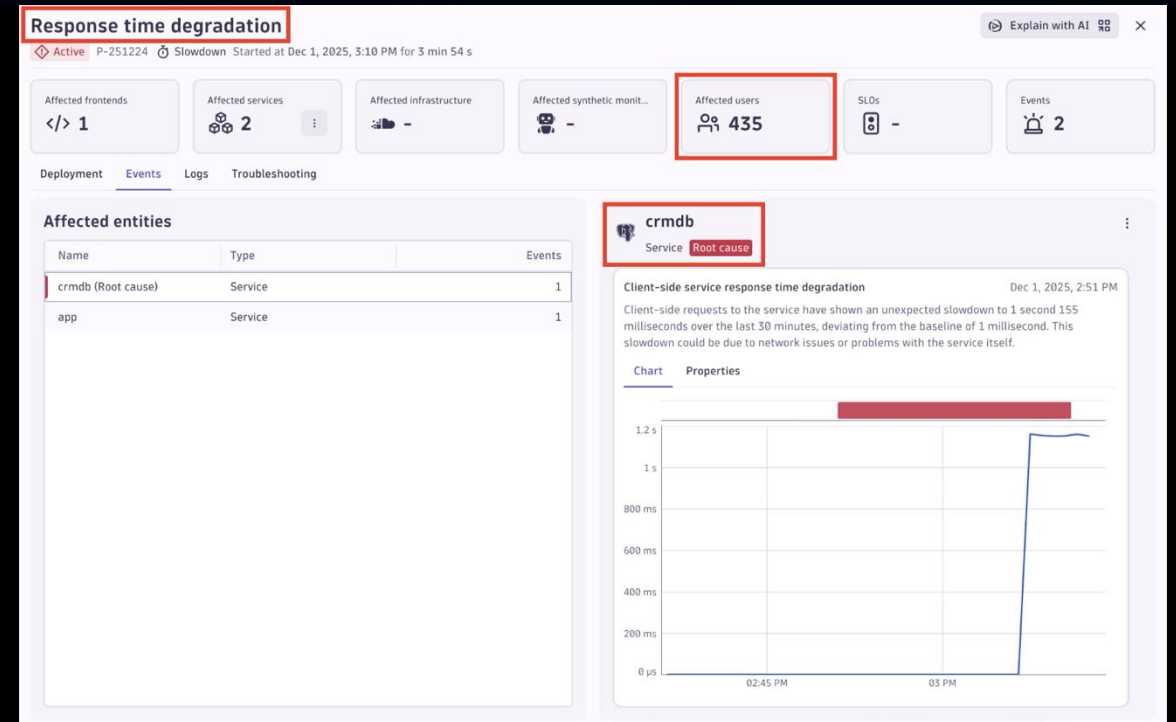


A collage of various technology and cloud service logos, including AWS, GitHub, Docker, and others, arranged in a grid-like pattern. The Dynatrace logo is prominently displayed at the bottom center.

A collage of various technology and cloud service logos, including AWS, GitHub, Docker, and others, arranged in a grid-like pattern. The Dynatrace logo is prominently displayed at the bottom center.

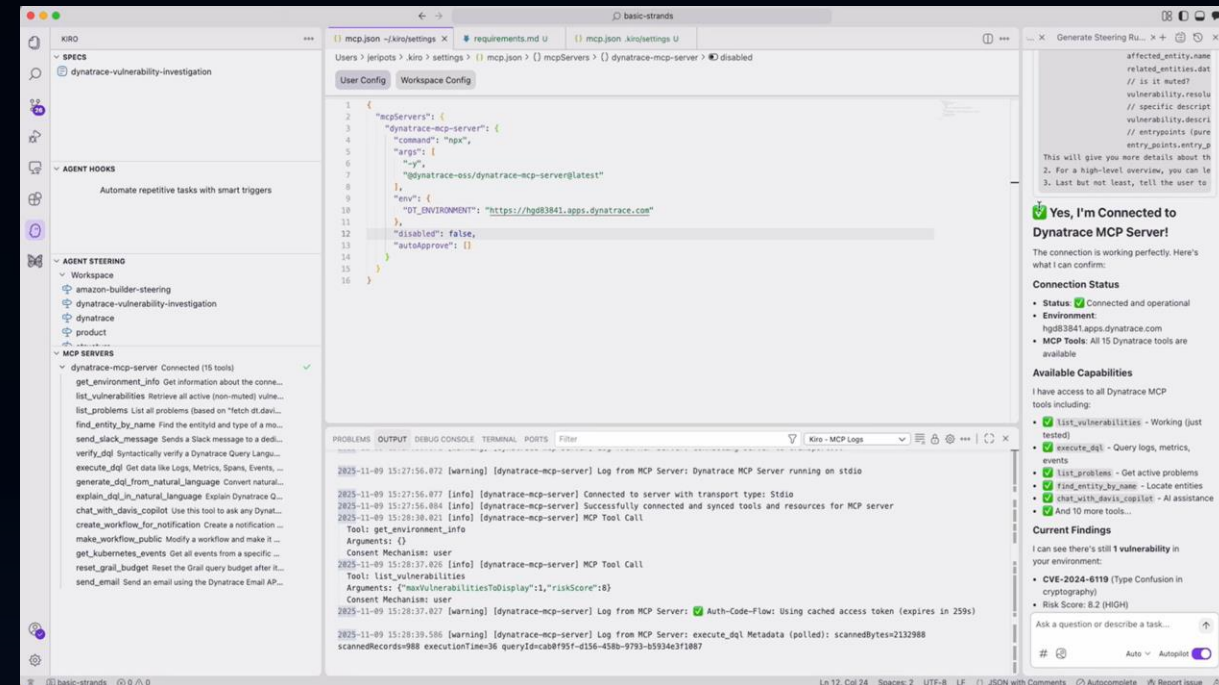
AWS DevOps Agent

- Pairs Dynatrace's AI-powered root cause analysis and real-time production context with AWS's frontier agent capabilities.
- Delivers reduced MTTR through topology-aware root cause analysis, and autonomous operations that keep applications healthy without constant intervention.



Coding Agent Integrations

- Boost Productivity: Real-time Dynatrace context and insights where your Developers live
- Automatically fix bugs and vulnerabilities: Dynatrace collaborates with AI Coding Agents
- Direct access to Dynatrace AI Agents and Tools in natural language



Move fast,
~~break things~~
break nothing.

Observability - Built for coding agents



Built for developers who ship with coding agents.

Your coding agent understands your code. Bluebox is what makes it understand what that code does in the real world — so you can merge with confidence.

Trusted and open standard.

TRUSTED

Dynatrace powered insights

Deterministic AI under the hood. Not pattern-matching, not guesswork. Real root cause, every time.

GITHUB NATIVE

GitHub Issues as output

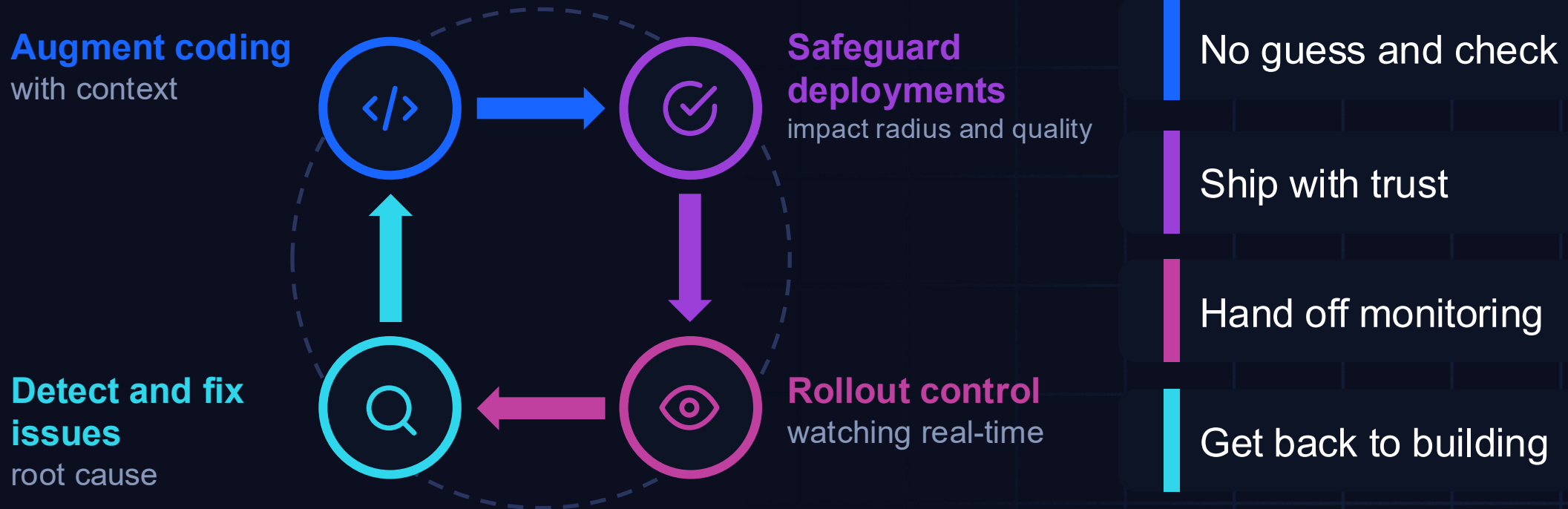
Every fix lands as a GitHub Issue with all the evidence your agent or you needs to act. No new tools to learn.

OPEN STANDARD

Built on OpenTelemetry

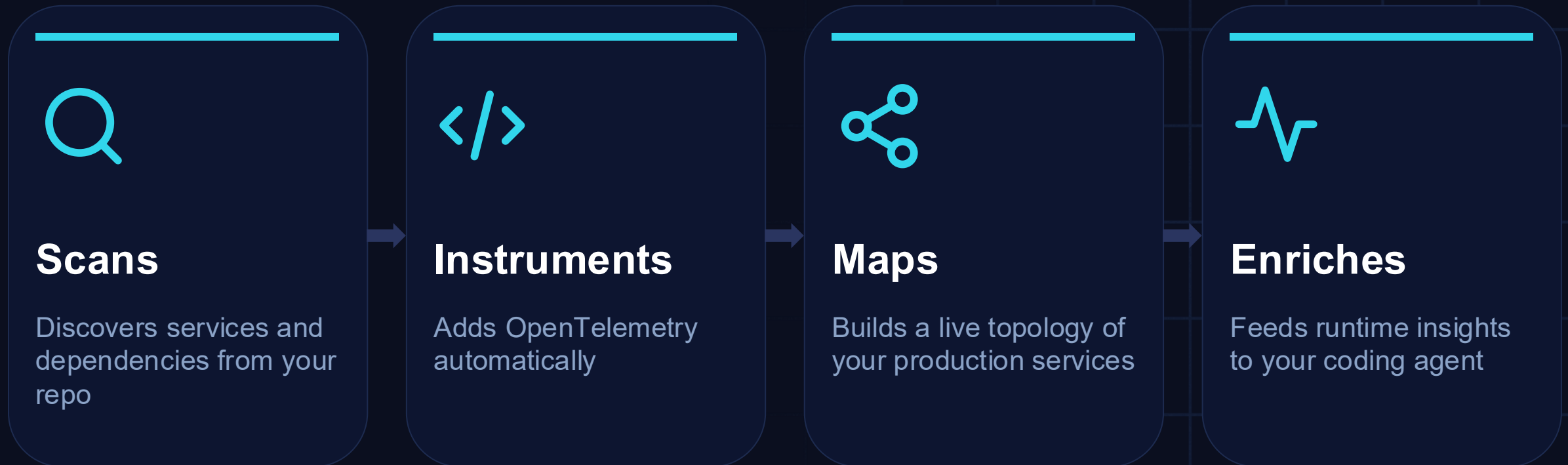
No vendor lock-in. Works with your existing OTel setup, or guides you through setting one up.

Observability driving the **agentic dev loop**



HOW DOES IT WORK?

Gives your agents **eyes into prod**



Briefs your agents before building



Plan before you prompt



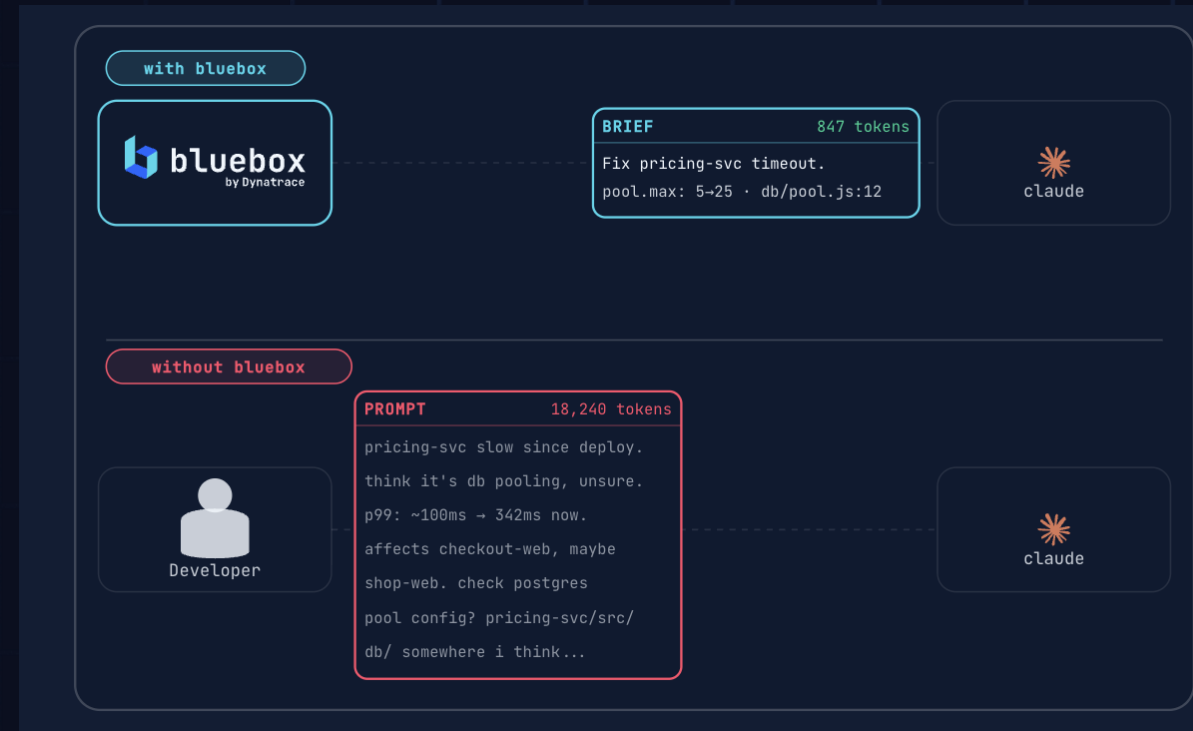
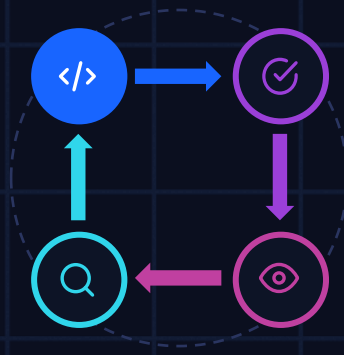
Build with non-frontier models



Less iteration and token chewing



No more prompt and pray



Safeguard deployments before you ship



Map blast radius before you push



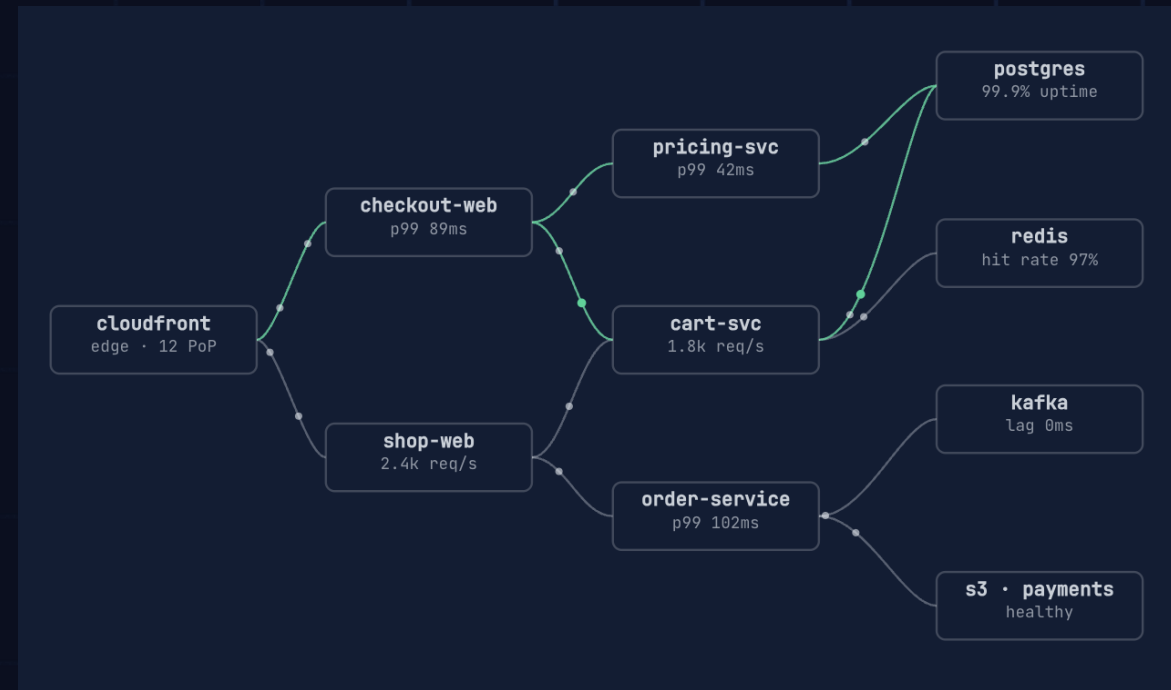
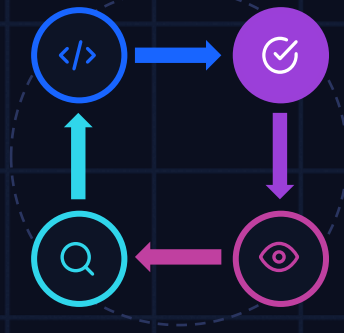
Verify environment readiness



Block bad releases with quality gates



Detect regressions before rollout



Finds issues before they're issues



Monitors production signals continuously



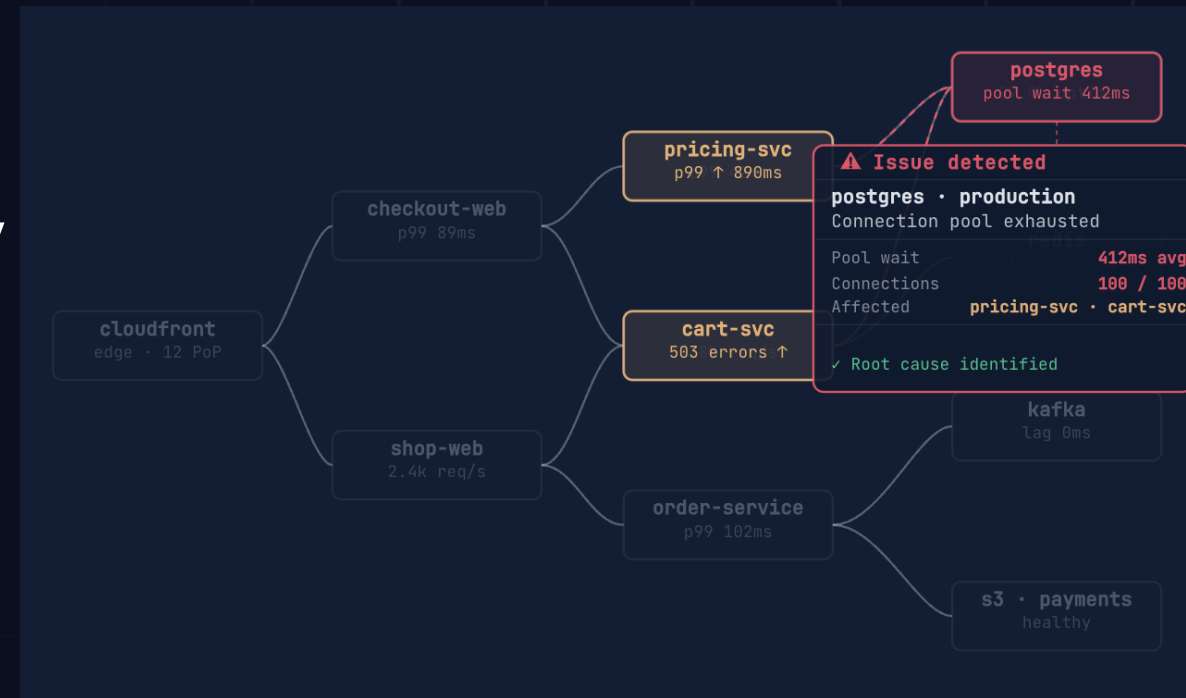
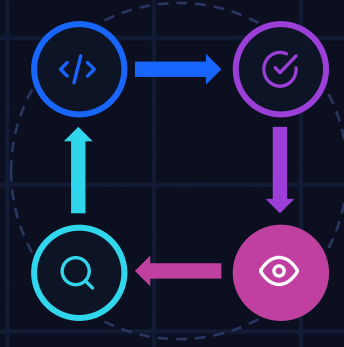
Surfaces incidents the moment they appear



Investigates with full runtime understanding



Detects root-cause in your code



Hands the **fix PR Plan** to you & your agents



Evidence-backed plans attached to every PR



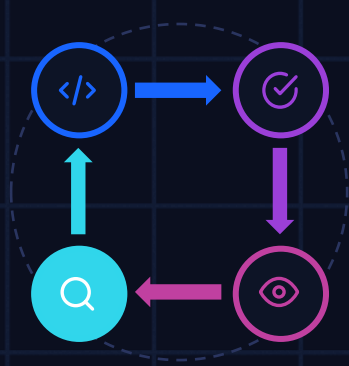
Blast radius shared before you merge



You stay in control – you review and merge



Bluebox confirms the fix held



 LIVE

 Analyzing pricing-svc timeout error...

 **claude**

Generating fix for timeout issue...

└ Patch ready · 3 files changed · +12 -4

 **github**

Creating pull request...

└ PR #142 opened: Fix pricing-svc timeout

 **Fix delivered · PR #142 ready for review**

 Developer assigned · 0 comments

Act on answers, not guesses