

Beyond the AI Hype

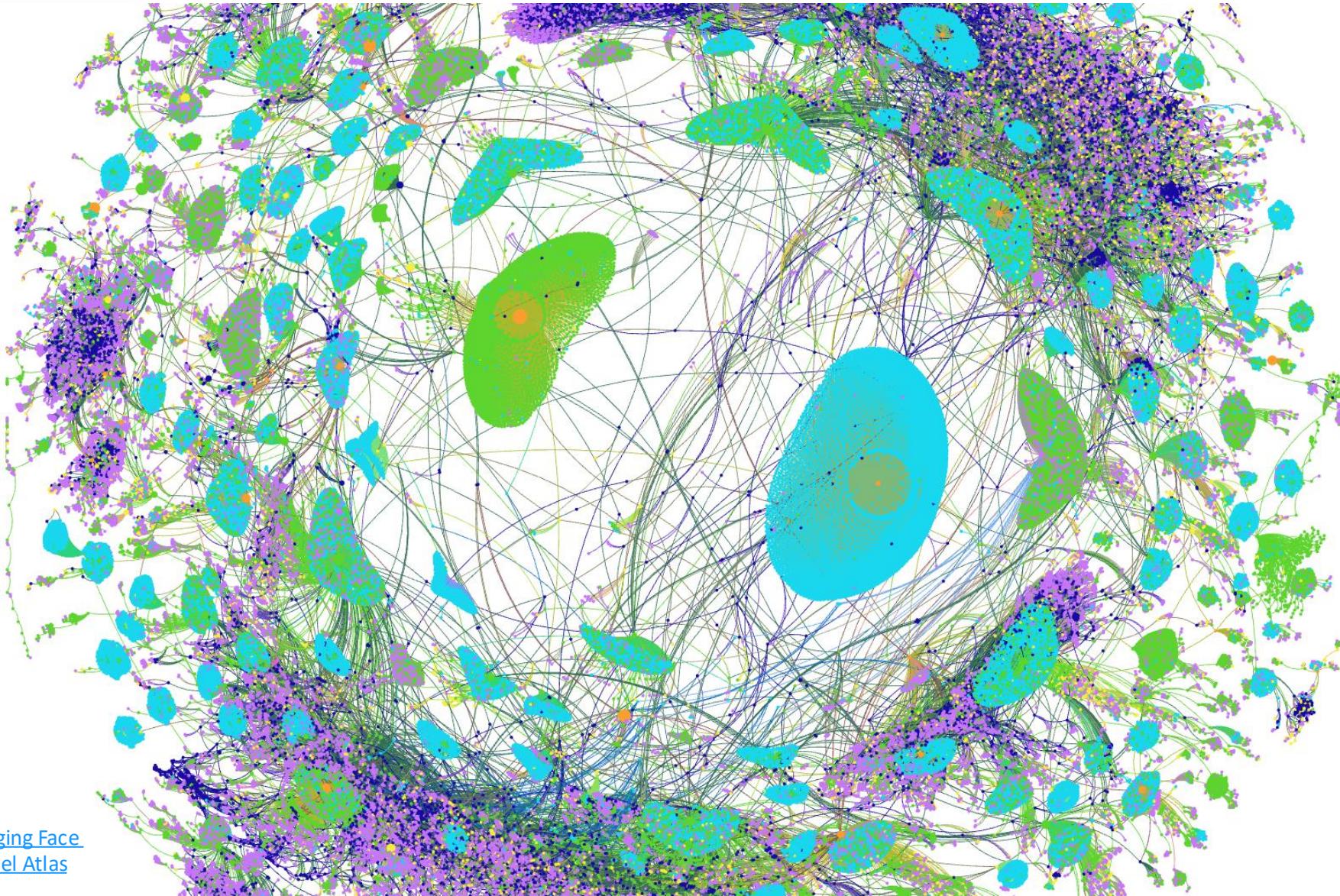
Agentic AI in Observability - A Reality Check



PRESENTER

Wolfgang Beer
Fellow Product Manager
Dynatrace

Decoding the AI Hype: Reality vs. Buzz



2.176.528
published AI models
on Hugging Face



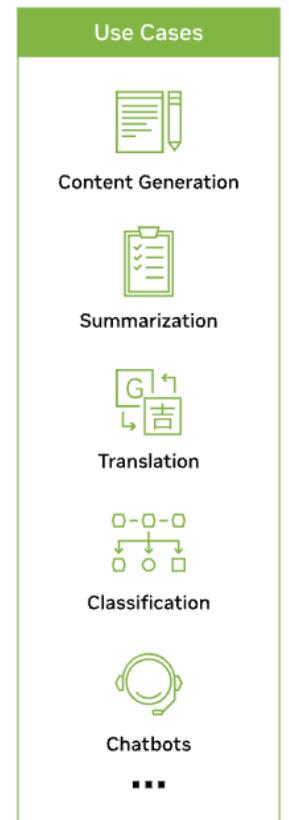
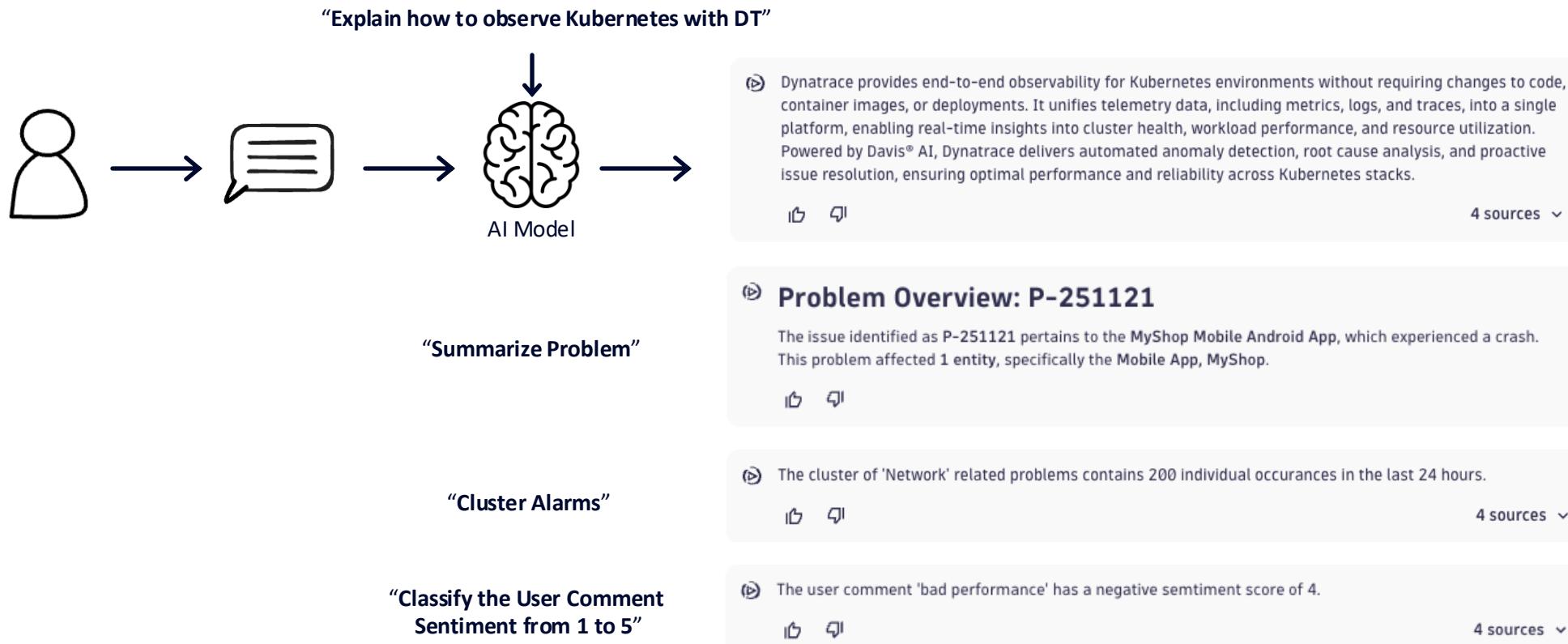
350M
downloads
of top 3 models



500M
data sets to finetune
AI models

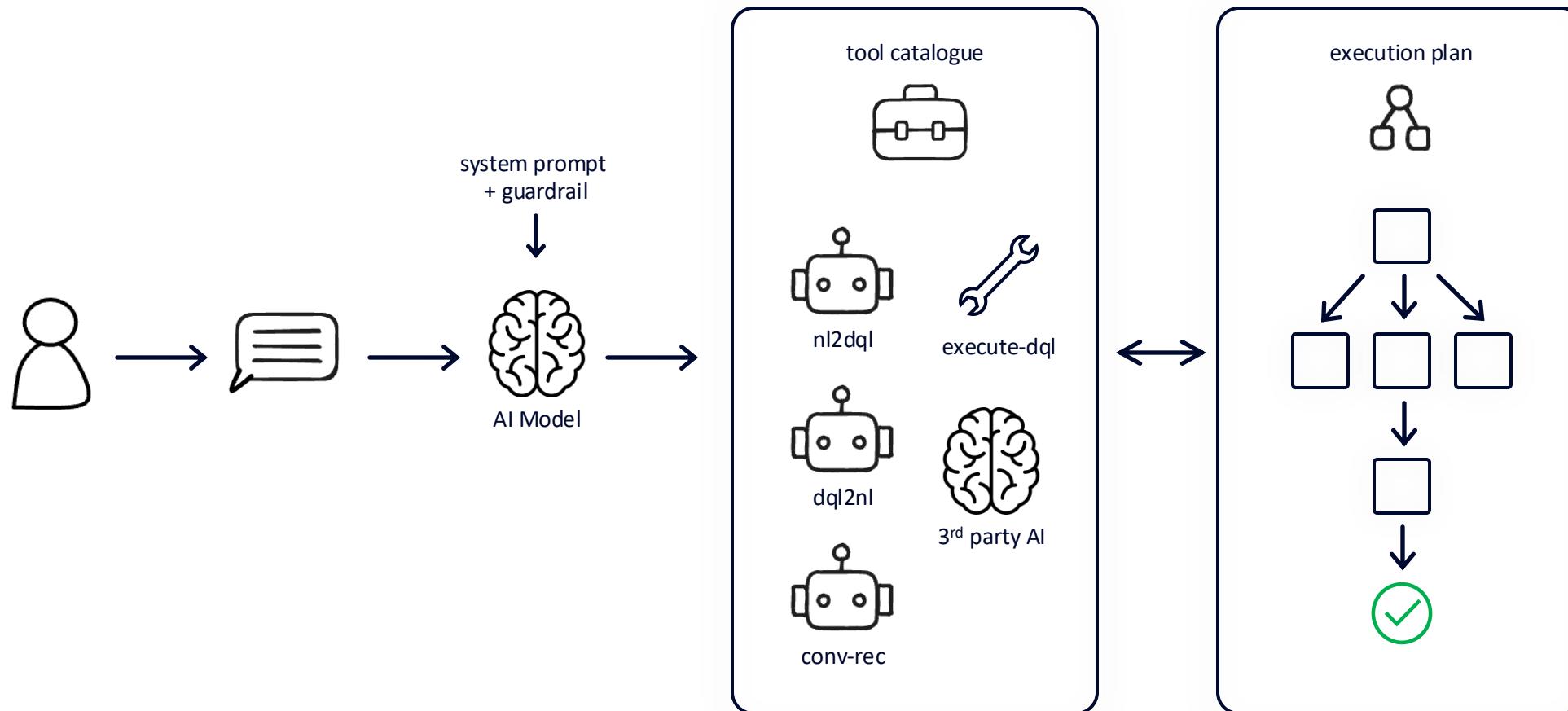
What's a Generative LLM AI Model?

A Generative Large Language Model is a pretrained AI model that transforms the users text prompt into **the next likely text output**.



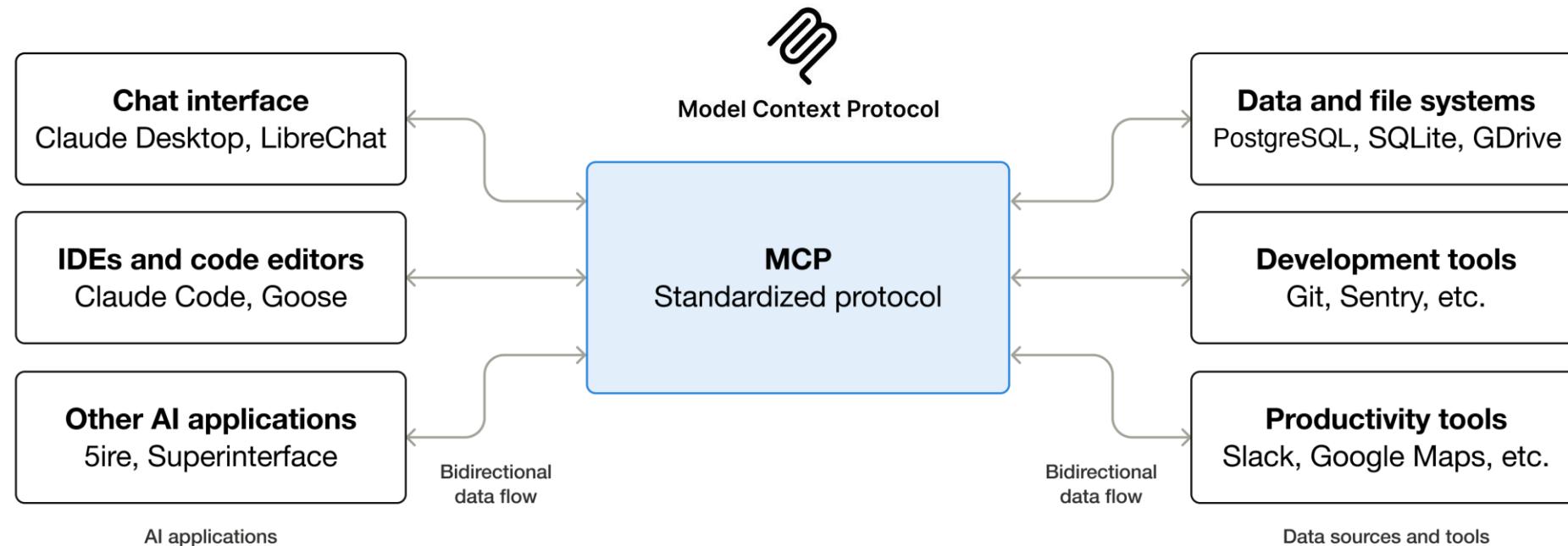
What's an AI Agent?

An AI agent is a software system that leverages AI to **reason**, **plan**, **act**, and **adapt**, with the goal of **completing tasks** on behalf of humans and other systems.

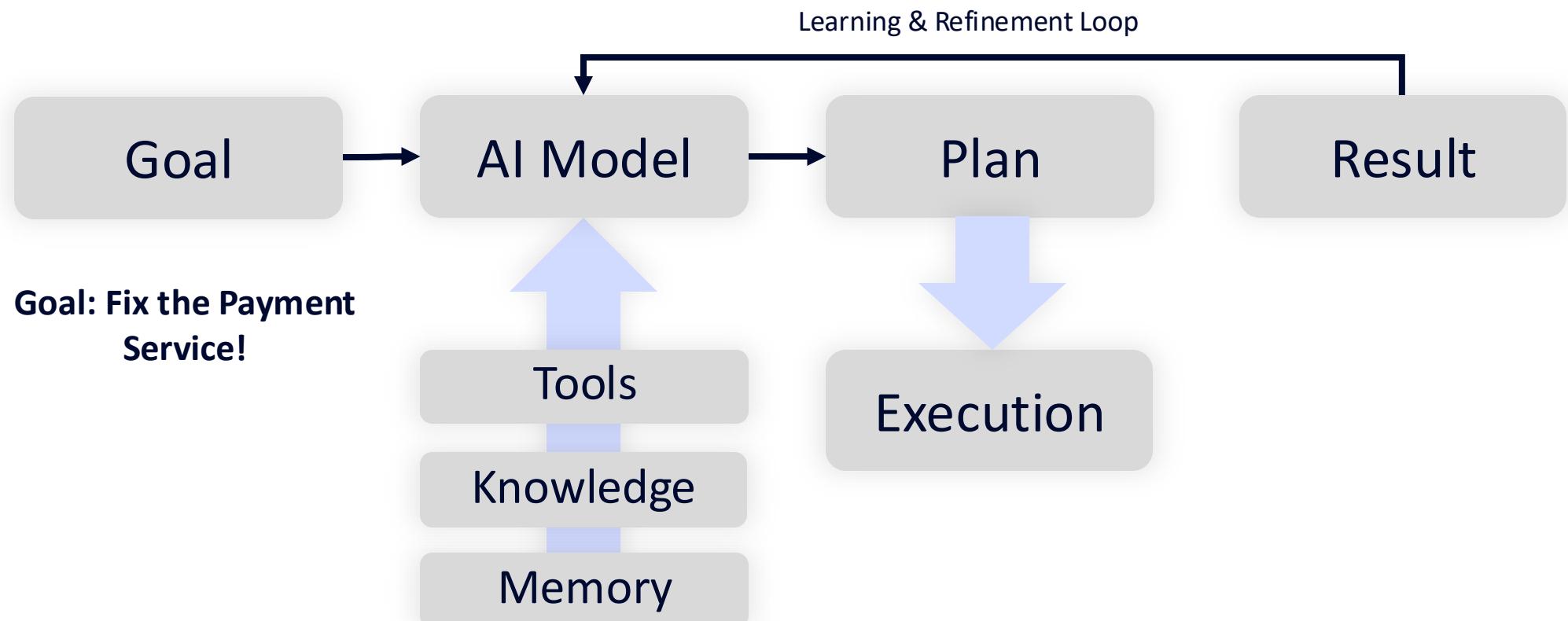


Agentic AI – The Rise of Model Context Protocol

- MCP (Model Context Protocol) is an open-source standard for connecting AI applications to external systems. It's the "**USB-C for AI**".



What's the Core of Agentic AI?



How Dynatrace is Delivering Value with AI?

AI Platform

OBSERVABILITY **WITH** AI
Causation & Automation

CONTEXT-DRIVEN, CAUSAL REASONING-BASED
REMEDIATION

AI Observability

OBSERVABILITY **OF** AI
Protecting AI Workloads

SUPPORT CUSTOMERS IN DELIVERING TRANSPARENT
& RESPONSIBLE AI SOLUTIONS

OUR FUTURE

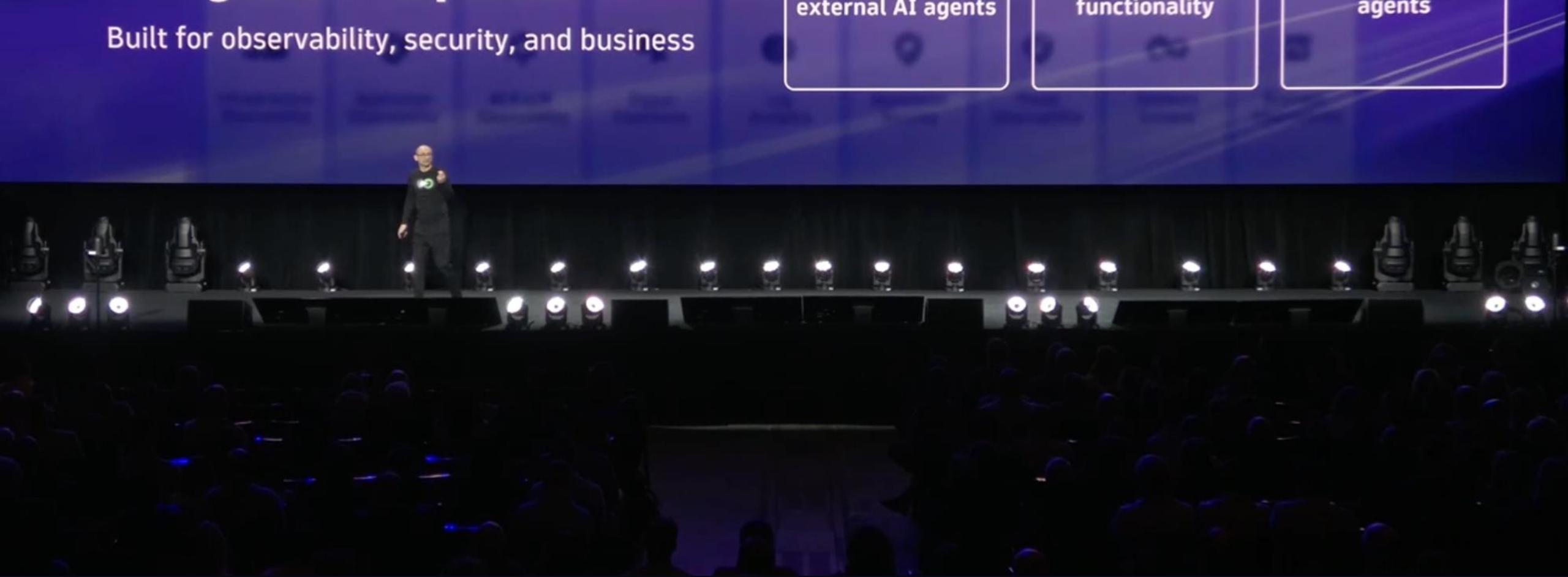
Dynatrace is evolving into an agentic AI platform

Built for observability, security, and business

Orchestrating
internal and
external AI agents

Creating AI agents
from app
functionality

Collaborating with
other external AI
agents

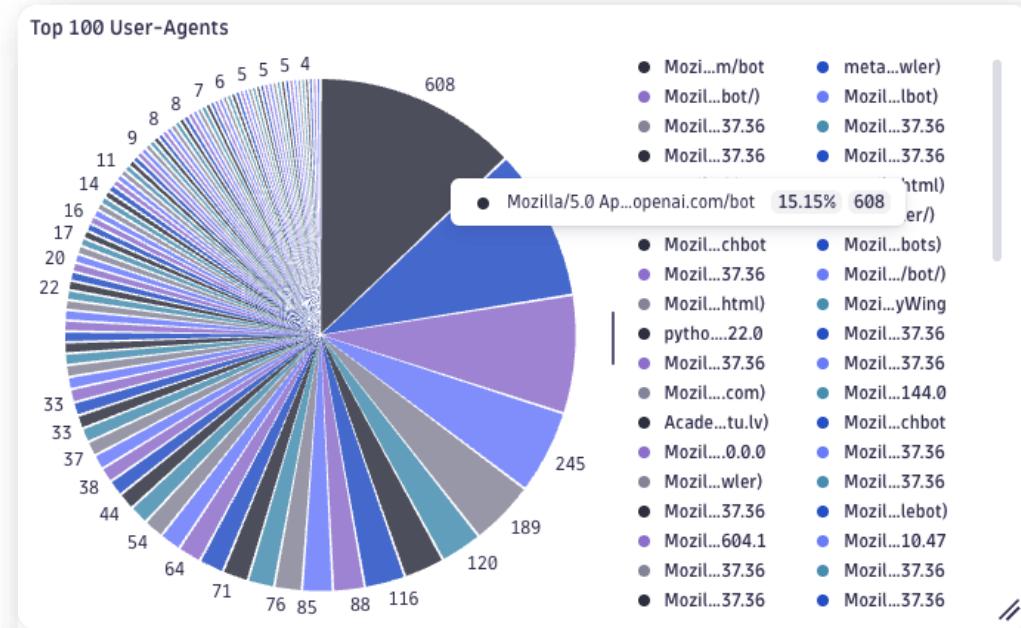
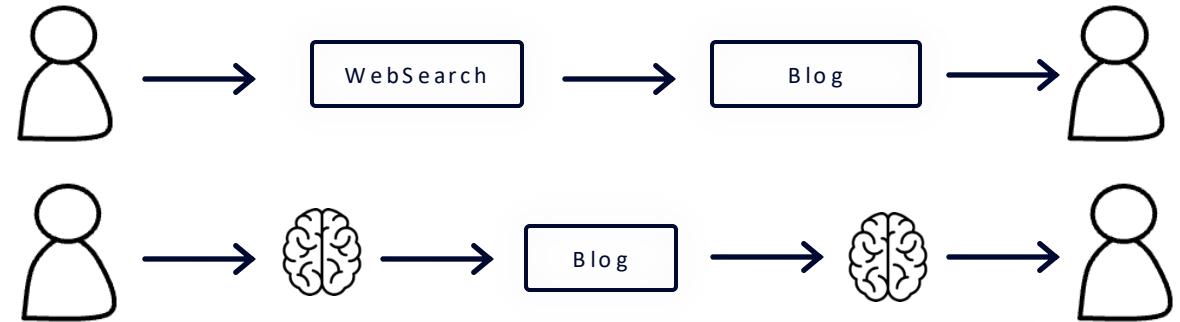


Agentic AI - Fundamentally Changes User Interaction

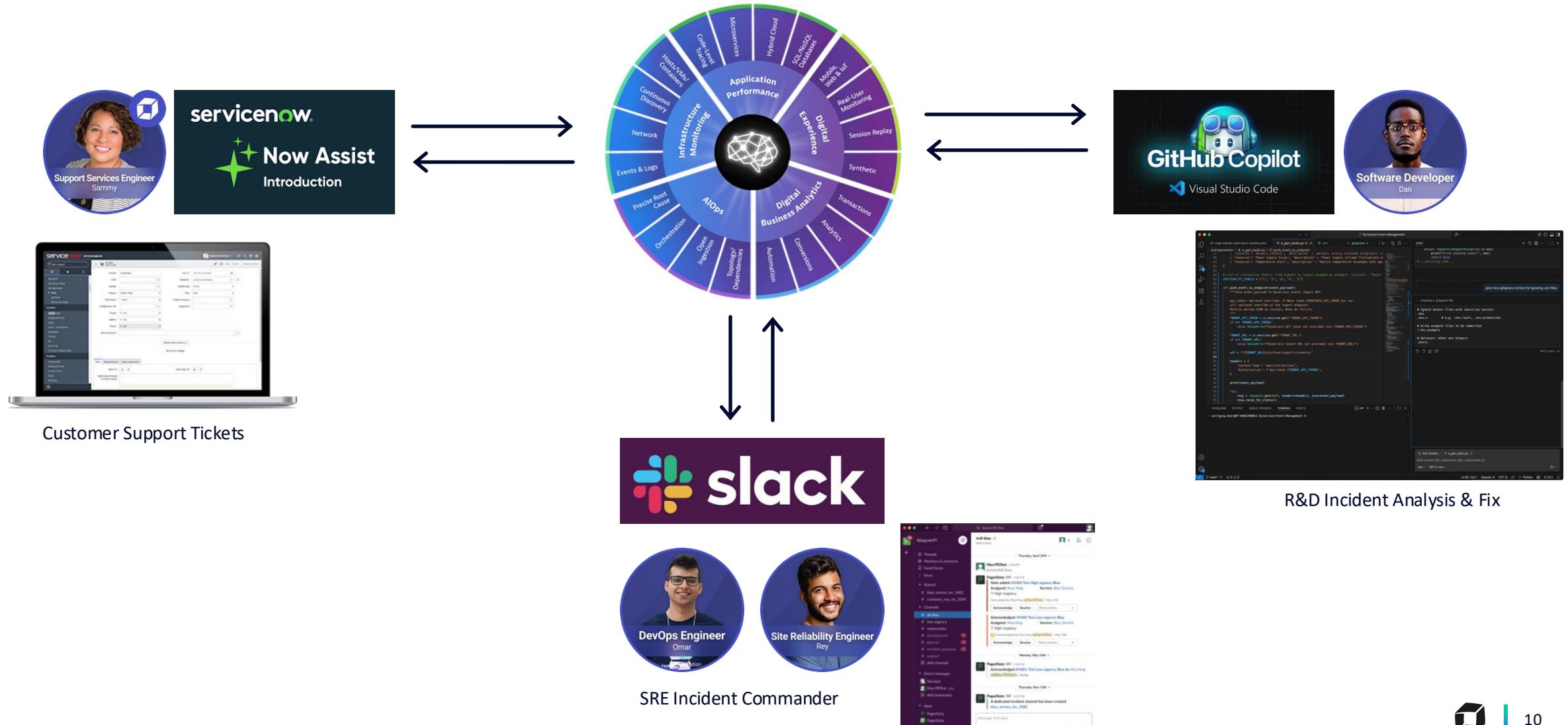
- Agentic AI breaks down silos!
- Democratizes access to information.
- Removes barriers for non-tech roles.
 - Ask data questions.
 - Help formulate and test hypothesis.
 - Let AI configure something complex.
 - Vibe your own app, based on a use-case.
 - Etc.

Example:

User Interaction with Information Search



Adopt AI to Deliver Value in the Users Own Context!



Adoption of AI in Observability – Hype vs. Reality

- **Provide value not just AI!**
- Evaluate the balance of value and risk.
- Respect the hard facts.
 - Generative models hallucinate.
 - Generative models are not the cheapest option.
- Consider risk, security, and legal early.
 - Risk class of use-cases
 - Security & Privacy
 - Copyright
 - Incident responsibility
 - etc.

“ —

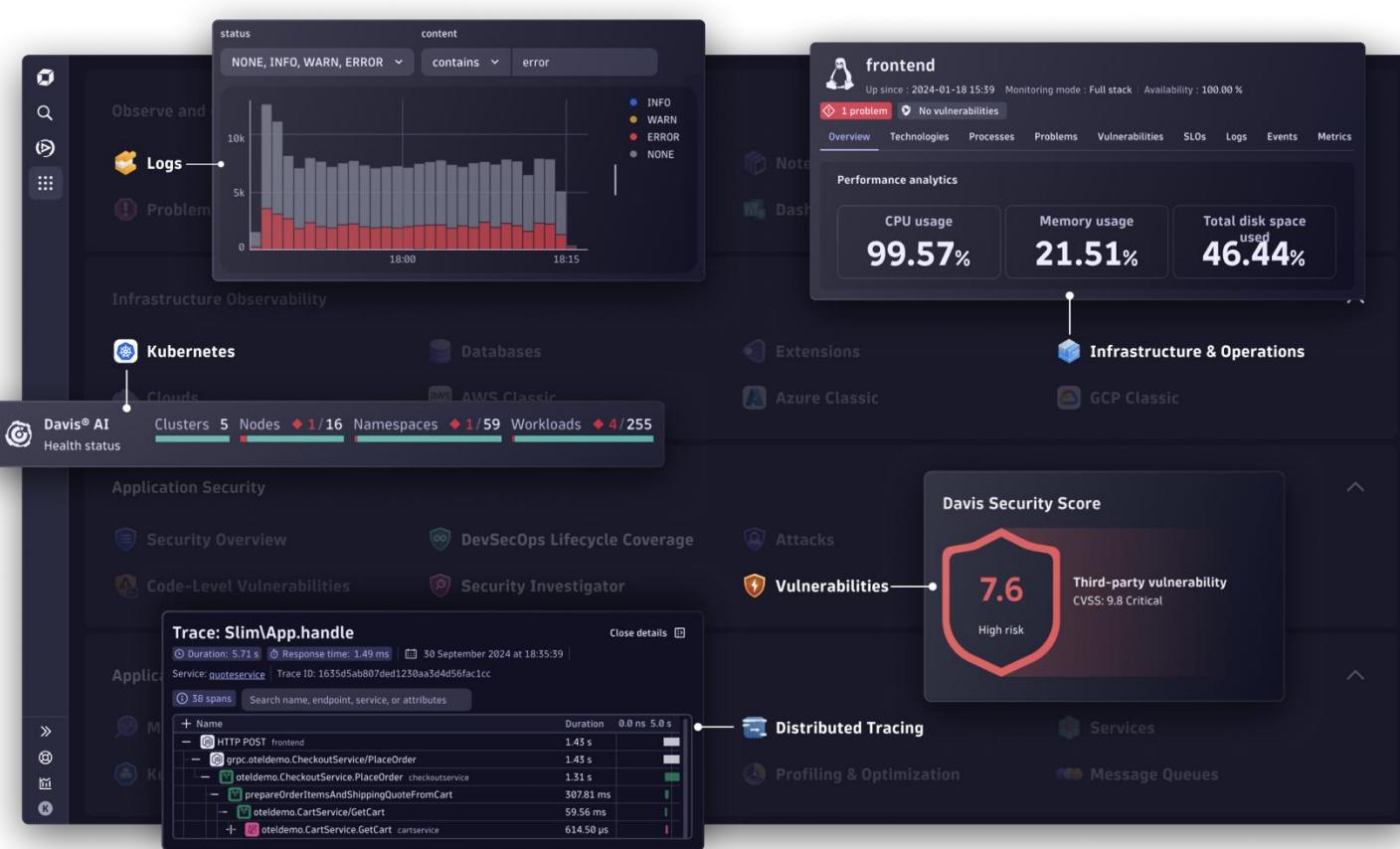
You never know who's swimming naked until the tide goes out.

— Warren Buffett

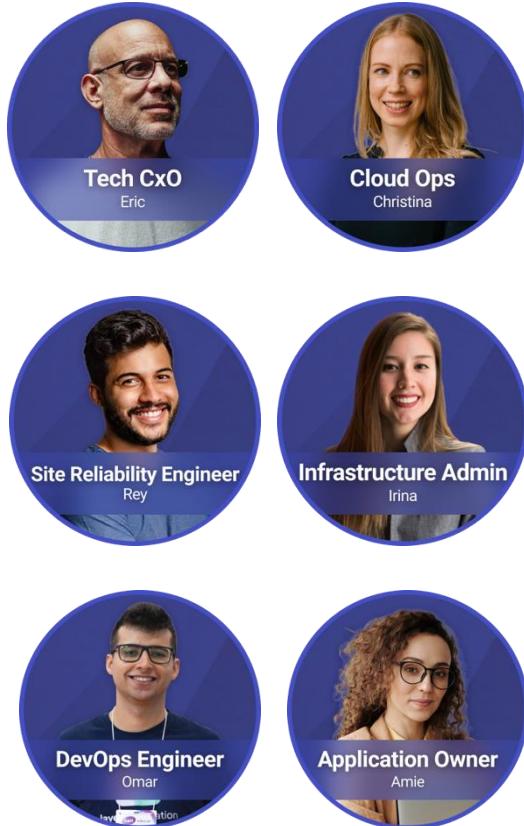


— ”

One AI Platform to Safeguard your Digital Systems



One AI Platform to Safeguard your Digital Systems



Build your own Agent

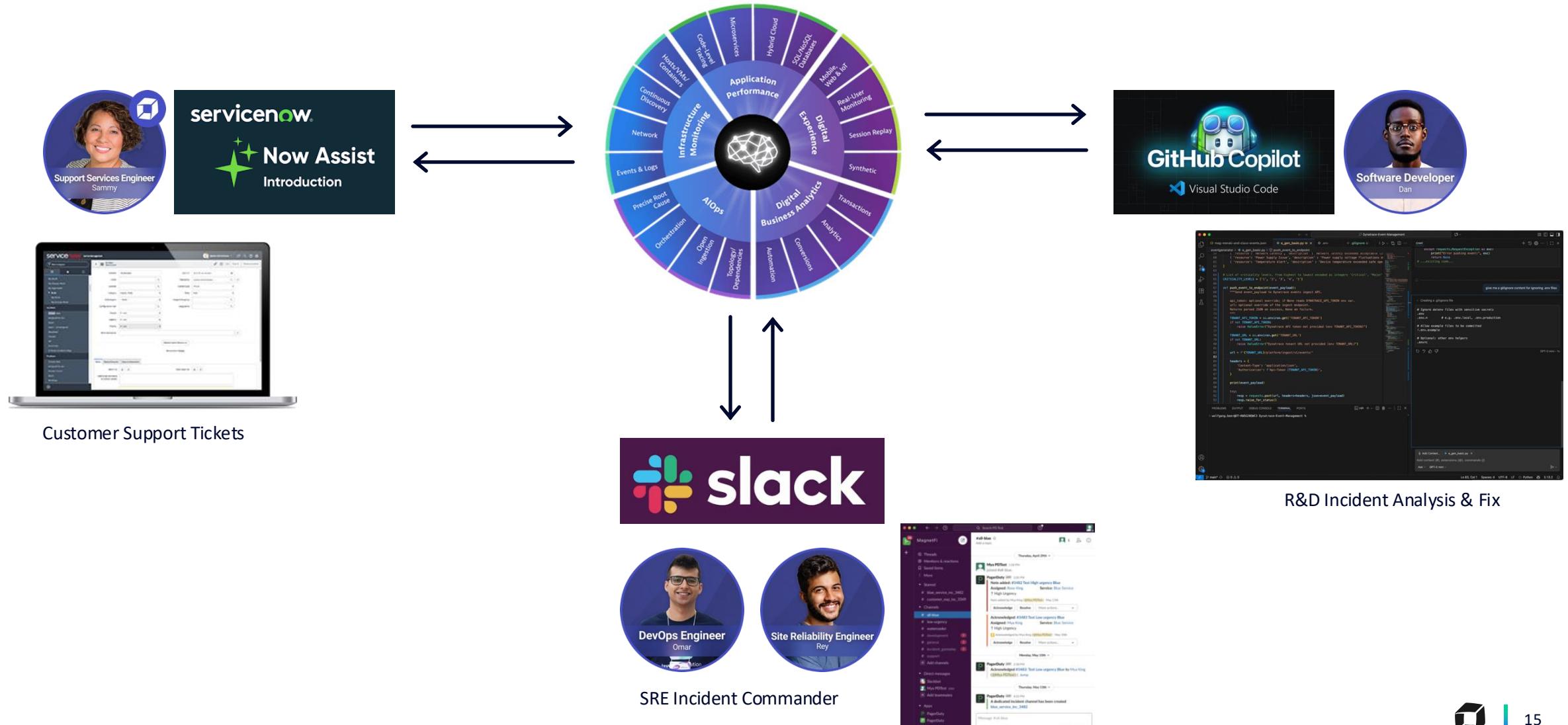
How Dynatrace AI enables domain-specific agentic
use-cases



PRESENTER

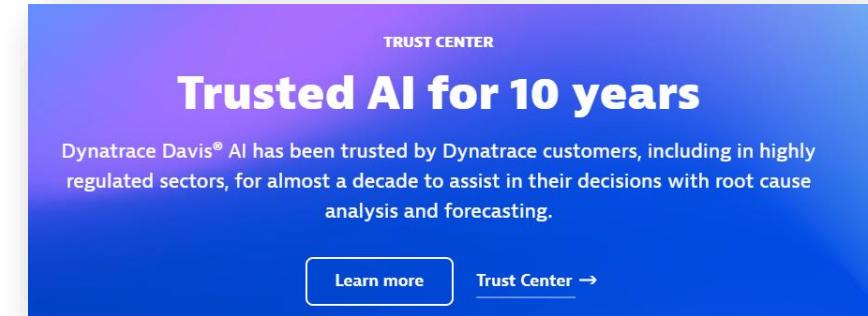
Wolfgang Beer
Fellow Product Manager
Dynatrace

Let's recap – AI Agents deliver Value in the Users Own Context!

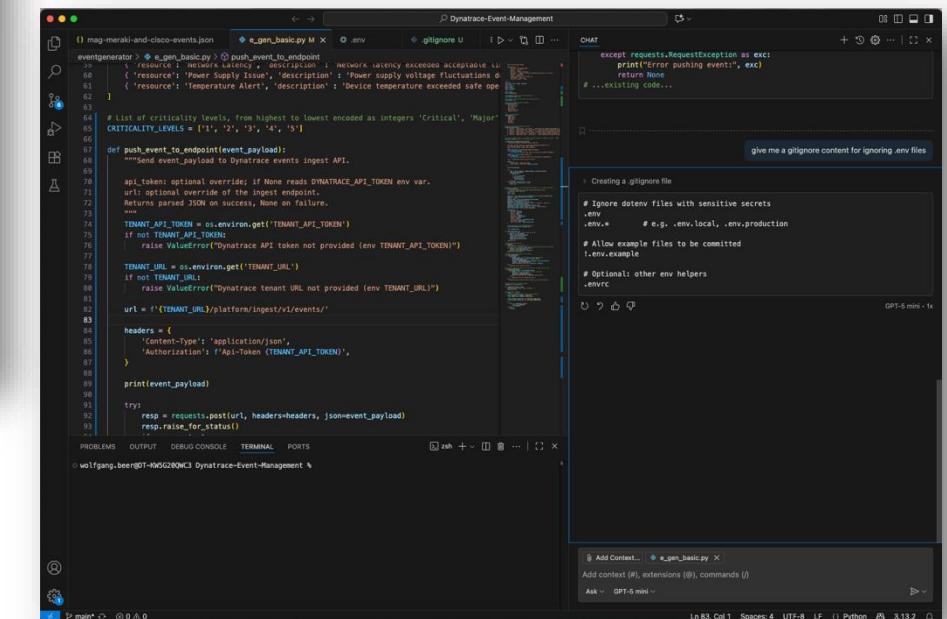
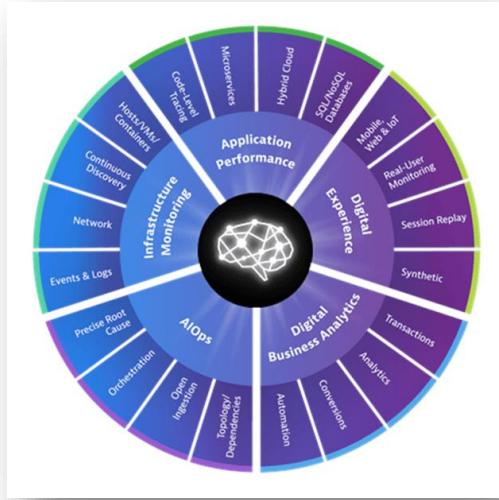


Dynatrace AI – Responsible & Transparent

- Data security and permissions (**Grail policies**)
- Control and transparency (**Grail query logs**)
- Reliable context (**Smartscape**)
- Scalable and flexible data pipeline (**OpenPipeline**)
- Dynamic data transformation (**Grail & DQL**)
- Real time data analysis (**Anomaly Detection**)
- Actionable automations (**Automation Engine**)
- Natural language (LLM) interface (**Generative AI**)



Dynatrace Agentic Interaction with Developers

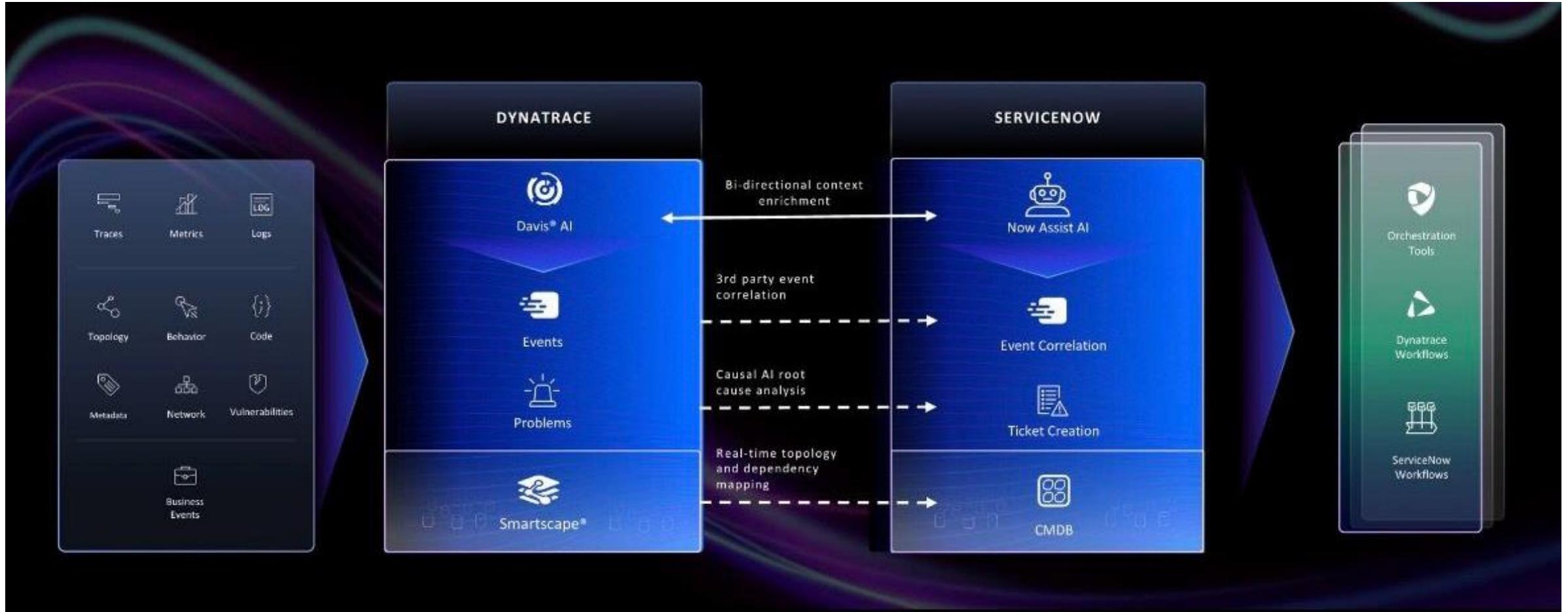


A screenshot of a terminal window titled "Dynatrace-Event-Management". The window shows a GitHub Copilot interaction. The user has typed "give me a gitignore content for ignoring .env files" and the Copilot has generated the following code:

```
#!/usr/bin/env python3
# Ignore dotenv files with sensitive secrets
# .env
# .env.local
# .env.production
# Allow example files to be committed
# .env.example
# Optional: other env helpers
# .envrc
```



Dynatrace Agentic Interaction with ServiceNow IT Operations



Express List +

Active Alerts 4

Search filtered alerts Close Last 24 hours Live Alerts Help Settings

<input type="checkbox"/>	Number	Description	Duration 	Severity 	Priority	Source	Impacted services	Configuration items	Node	Assigned to	Assignment...
<input type="checkbox"/>	Alert0011696	OPEN Problem P-25114226 in environment tacocorp Problem detected at: 09:11 (UTC)...	19 minutes ago	Major		SGO-Dynatrace				TradeManagement	
<input type="checkbox"/>	Alert0011694	OPEN Problem P-25114217 in environment tacocorp Problem detected at: 08:07 (UTC)...	1 hour ago	Major	High	SGO-Dynatrace	easyTradeManager ... easyTradeLoginService ... BrokerService ...	TradeManagement - SERVI	easyTradeM	easyTradeM	anager
<input type="checkbox"/>	Alert0011692 2	Group of alerts, OPEN Problem P-25114208 in environment tacocorp Problem detected a...	2 hours ago	Major	Moderate	SGO-Dynatrace	easyTradeManager ... BrokerService ... easyTradeLoginService ...	TradeManagement - SERVI	easyTradeManager - SERVI	easyTradeM	anager
<input type="checkbox"/>	Alert0011678	There are 3 alerts which were not updated for over 2 days. To remediate the issue follow...	5 hours ago	Minor	Low		ServiceNow Event...	Alert Processing			

Express List +

Active Alerts 4

	Number	Description	Duration	Severity	Priority	Source	Impacted services	Configuration items	Notes
	Alert0011698 2	Group of alerts, OPEN Problem P-25114226 in environment tacocorp Problem detected a...	21 minutes ago	● Major	High	SGO-Dynatrace	easyTradeManager ... engine-* (/api) ... BrokerService -...	TradeManagement - SERVICE ... engine-* (/api) - SERVICE-C...	TradeManagement - SERVICE ... BrokerService -...
	Alert0011694	OPEN Problem P-25114217 in environment tacocorp Problem detected at: 08:07 (UTC)...	1 hour ago	● Major	High	SGO-Dynatrace	easyTradeManager -... easyTradeLoginService ... BrokerService -...	TradeManagement - SERVICE ... easyTradeManager - SERVICE ...	TradeManagement - SERVICE ... easyTradeLoginService -...
	Alert0011692 2	Group of alerts, OPEN Problem P-25114208 in environment tacocorp Problem detected a...	2 hours ago	● Major	Moderate	SGO-Dynatrace	easyTradeManager -... BrokerService -... easyTradeLoginService ...	TradeManagement - SERVICE ... easyTradeManager - SERVICE ...	TradeManagement - SERVICE ... easyTradeLoginService -...
	Alert0011678	There are 3 alerts which were not updated for over 2 days. To remediate the issue follow...	5 hours ago	● Minor	Low	ServiceNow Event	ServiceNow Event...	Alert Processing	

1 tell me more details for alert0011698

Starting "Analyze alert impact Dynatrace Demo..."

View AI Agent Processing Steps

I am now fetching the Dynatrace problem report for your requested alert. You may refresh the report at any time if you wish to view the latest details.

View AI Agent Processing Steps

Dynatrace Problem Report for Alert0011698

- Problem Title: Failure rate increase
- Display ID: P-25114226
- Impact Level: SERVICES
- Severity Level: ERROR
- Status: OPEN
- Start Time: 2025-11-20 23:51:00 (converted from timestamp)
- Affected Entities:
easyTradeManager, TradeManagement
- Root Cause Entity:
TradeManagement
- Management Zone: Kendon Test
- Environment Tag: easytrade
- Problem URL: [View in Dynatrace](#)

Suggested Follow-up Questions

Reply to Now Assist...

Express List

+

Active Alerts

4

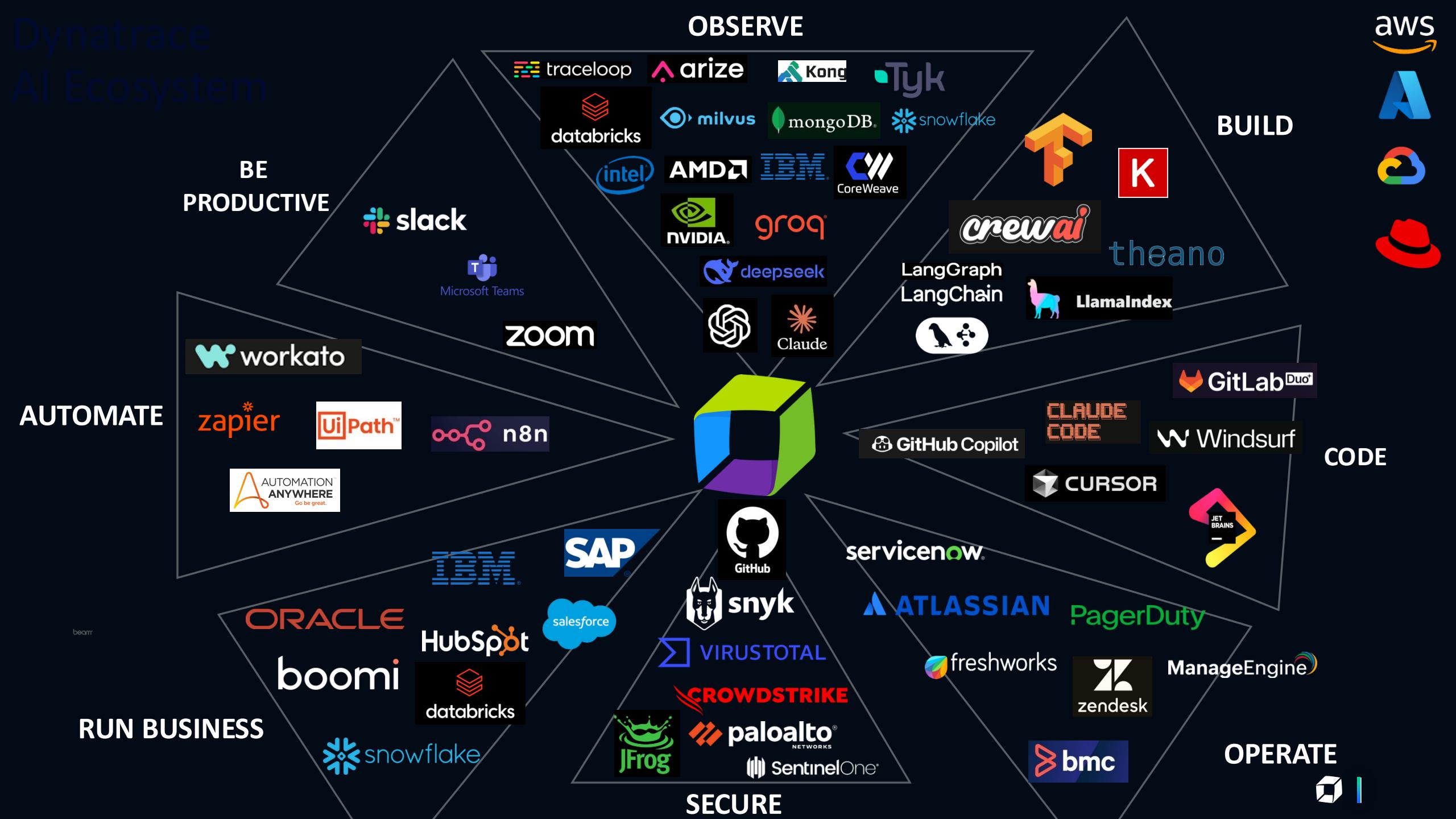
Q Search filtered alerts

Clos

Last 24 hours

1 live

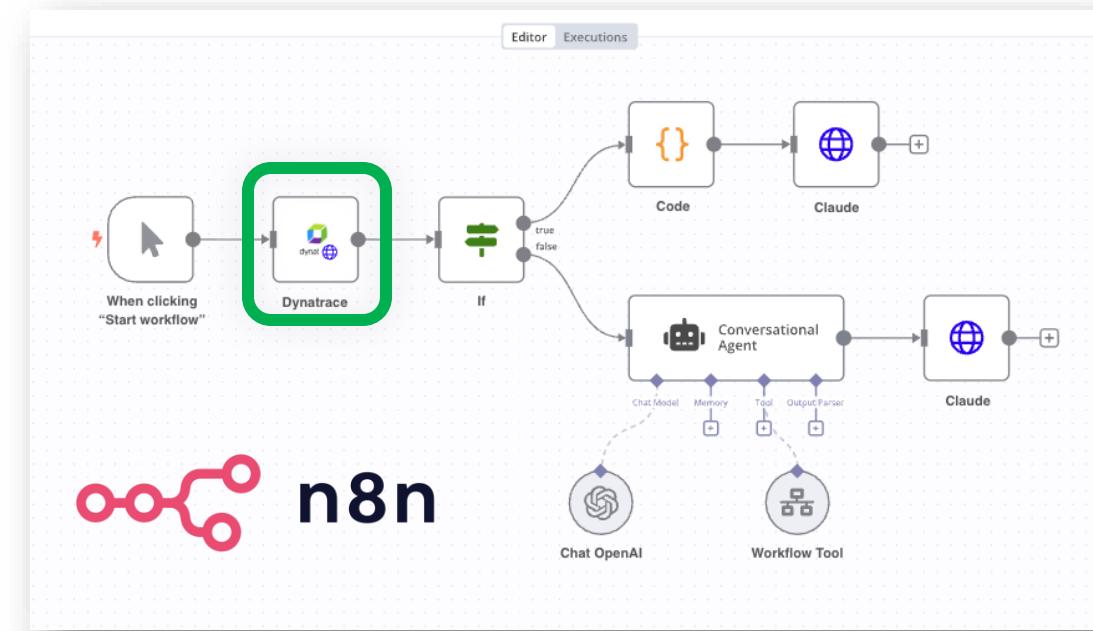
Number	Description	Duration	Severity	Priority	Source	Impacted services	Configuration items	Node	Assigned to	Assignment
Alert0011698 	Group of alerts, OPEN Problem P-25114226 in environment tacocorp Problem detected a...	29 minutes ago	Major	High	SGO-Dynatrace	easyTradeManager ... engine-* (/api) ... BrokerService ... 	TradeManagement - SERVI engine-* (/api) - SERVICE-C	easyTradeManager		
Alert0011694	OPEN Problem P-25114217 in environment tacocorp Problem detected at: 08:07 (UTC)...	1 hour ago	Major	High	SGO-Dynatrace	easyTradeManager ... easyTradeLoginService ... BrokerService ...	TradeManagement - SERVI	easyTradeManager		
Alert0011692 	Group of alerts, OPEN Problem P-25114208 in environment tacocorp Problem detected a...	2 hours ago	Major	Moderate	SGO-Dynatrace	easyTradeManager ... BrokerService ... easyTradeLoginService ...	TradeManagement - SERVI easyTradeManager - SERVI	easyTradeManager		
Alert0011678	There are 3 alerts which were not updated for over 2 days. To remediate the issue follow...	5 hours ago	Minor	Low		ServiceNow Event...	Alert Processing			



Dynatrace AI in Custom Agents

Build your GitHub, LangGraph, OpenAI, Google ADK and n8n Agents

- Automating and speeding up repetitive tasks.
- Automate on top of companies' business and domain logic.
- Giving an automated process more flexibility to handle unexpected situations.



N8N configured AI Agent interacting with Dynatrace AI

Dynatrace Remote MCP Server in GitHub CoPilot



Configure Tools

52 Selected OK

Select tools that are available to chat:

- Built-In
 - changes Get diffs of changed files
 - edit Edit files in your workspace
 - extensions Search for VS Code extensions
 - fetch Fetch the main content from a web page. You should include the URL of the page ...
 - githubRepo Searches a GitHub repository for relevant source code snippets. You can s...
 - new Scaffold a new workspace with VS Code-specific configurations to compile, debug ...

CHAT

dt-agentic-ai

how many problems do I have in my dynatrace environment?

You currently have 21 active problems in your Dynatrace environment. If you need a breakdown or details about these problems, let me know!

can you give me a summary of the most critical ones?

Based on the available data, most of your active problems are categorized as "CUSTOM_ALERT" events, with some related to security, availability, and performance. However, the problem records do not include explicit severity or criticality levels.

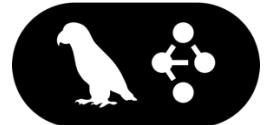
To identify the most critical problems, I recommend focusing on:

- Problems affecting core services (e.g., frontend, adservice, productcatalogservice)
- Problems with categories like AVAILABILITY, ERROR, or SLOWDOWN
- Security-related events (ACTIVE SECURITY_EVENT)

Would you like a detailed list of these higher-priority problems, or a summary grouped by type and affected entity?

GPT-4.1 • 1x

Demo Build an SREAgent with Dynatrace and LangGraph



- Let me introduce you, my 'Helsinki' AI Agent
- <https://github.com/wolfgangB33r/dt-example-agent>

The screenshot shows a GitHub repository page for 'dt-example-agent' (https://github.com/wolfgangB33r/dt-example-agent). The repository is public and has 0 stars, 0 forks, and 0 releases. The README file contains instructions for setting up the Dynatrace AI agent, requiring environment variables like OPENAI_API_KEY and DT_TENANT. The repository has 1 branch and 0 tags.

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

dt-example-agent (Public)

main | 1 Branch | 0 Tags | Go to file | Add file | Code | About

README | MIT license

Helsinki - An Example Dynatrace Agent

AI agent for analyzing and optimizing alert configurations in a Dynatrace tenant.

How to set up?

Following environment variables must be set:

OpenAI

- OPENAI_API_KEY=<YOUR_OWN_OPENAI_API_KEY>

Dynatrace

- DT_TENANT=<YOUR_DOMAIN>.live.apps.dynatrace.com
- DT_API_TOKEN=<YOUR_PLATFORM_API_TOKEN>

An example react agent

Readme | MIT license | Activity | 0 stars | 0 watching | 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Demo Build an SREAgent with Dynatrace and LangGraph



```
# Initialize the MultiServerMCPClient with the configuration
client = MultiServerMCPClient(
    {
        "dynatrace-mcp": {
            "transport": "streamable_http",
            "url": 'https://{{tenant}}/playground',
            "headers": {
                "Authorization": "Bearer {{token}}"
            }
        }
    }
)

# Define the run_agent function
async def run_agent(msg, thread_id):
    total_input_tokens = 0
    total_output_tokens = 0
    try:
        # Initiate the agent model and tools
        #model = init_chat_model("gemini-2.0-flash", model_provider="google_genai")
        #model = init_chat_model("gemini-3-pro-preview", model_provider="google_genai")
        model = init_chat_model("gpt-4o", model_provider="openai")
        #
        mcp_tools = await client.get_tools()

        tools=[
            get_current_time,
            chat_response
        ]
        tools.extend(mcp_tools)
        agent_executor = create_agent(model, tools)

        config = {
            "configurable": {
                "thread_id": thread_id,
                "agent_name": AGENT_NAME,
                "agent_description": f"{{AGENT_NAME}}, a helpful assistant.",
            }
        }

        # Append newest user message
        input_messages.append({ "role": "user", "content": msg })

        # invoke the agent and pass callbacks
        response = await agent_executor.invoke({ "messages": input_messages}, config=config)
    except Exception as e:
        print(f"An error occurred: {e}")
        return None
    finally:
        # Clean up
        pass
```



Build an SREAgent with Dynatrace & Google Agent Toolkit



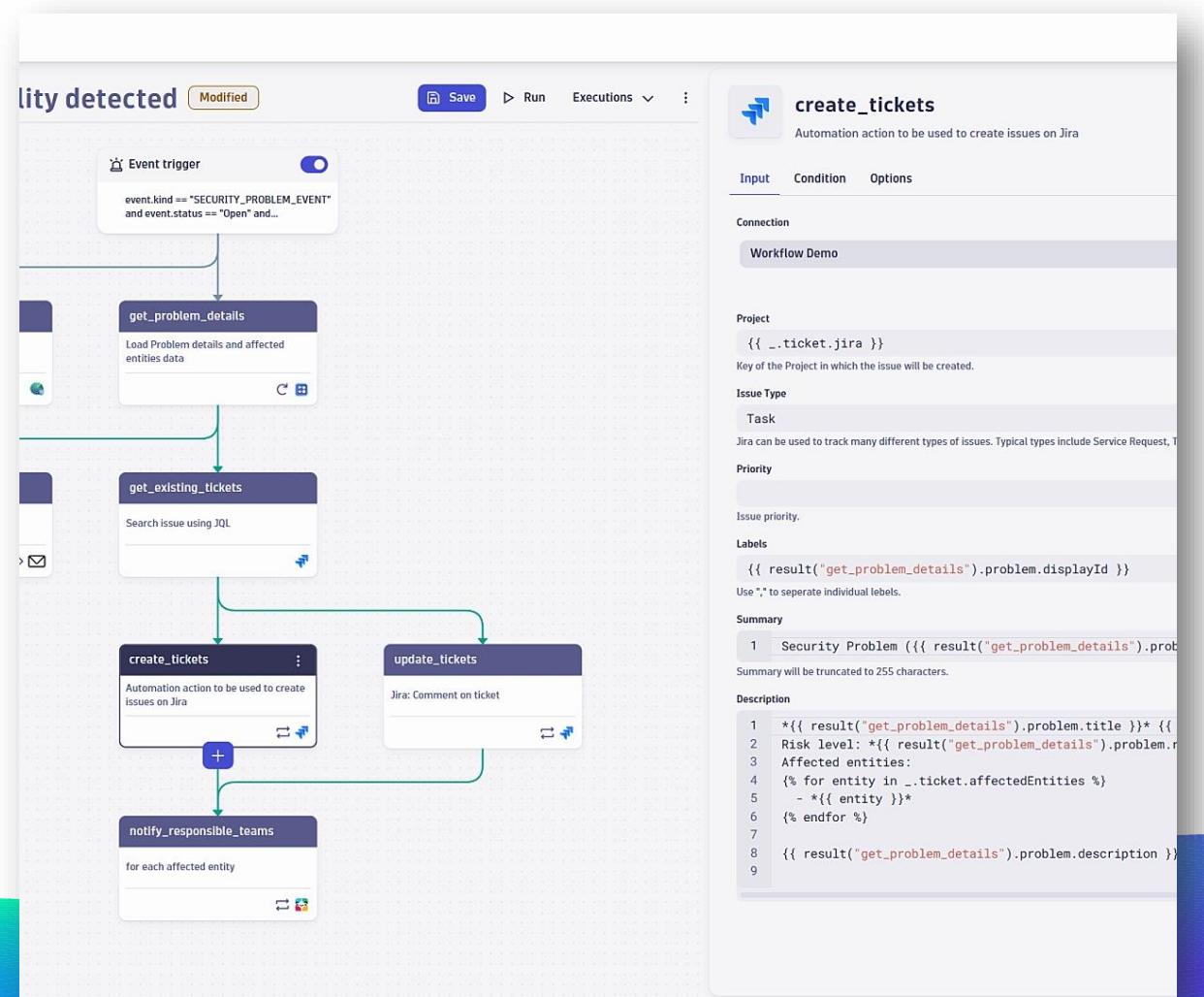
```
root_agent = Agent(
    name="Helsinki Agent",
    model="gemini-2.0-flash",
    description=(
        "Agent that is triggered by operations teams to investigate and remediate a critical incident."
    ),
    instruction=(
        "You are a helpful agent for observing digital systems. Autonomously execute tools as needed to answer questions. " +
        "Always reply with clear, human-readable summaries of your findings."
    ),
    before_tool_callback=lambda tool, args, tool_context: (
        print(f"Executing tool: {tool.name}")
    ),
    tools=[
        get_current_date_time_utc,
        MCPToolset(
            connection_params=StreamableHTTPServerParams(
                url='https://{{tenant}}/platform-reserved/mcp-gateway/v0.1/dynatrace-mcp'.format(tenant=DT_TENANT),
                headers={
                    'Authorization' : 'Bearer ' + DT_API_TOKEN
                }
            ),
            tool_filter=[]
        ),
        MCPToolset(
            connection_params=StreamableHTTPServerParams(
                url='https://{{tenant}}/platform-reserved/mcp-gateway/v0.1/dynatrace-mcp-experimental'.format(tenant=DT_TENANT),
                headers={
                    'Authorization' : 'Bearer ' + DT_API_TOKEN
                }
            )
        )
    ],
)
```

Show & Tell # 2

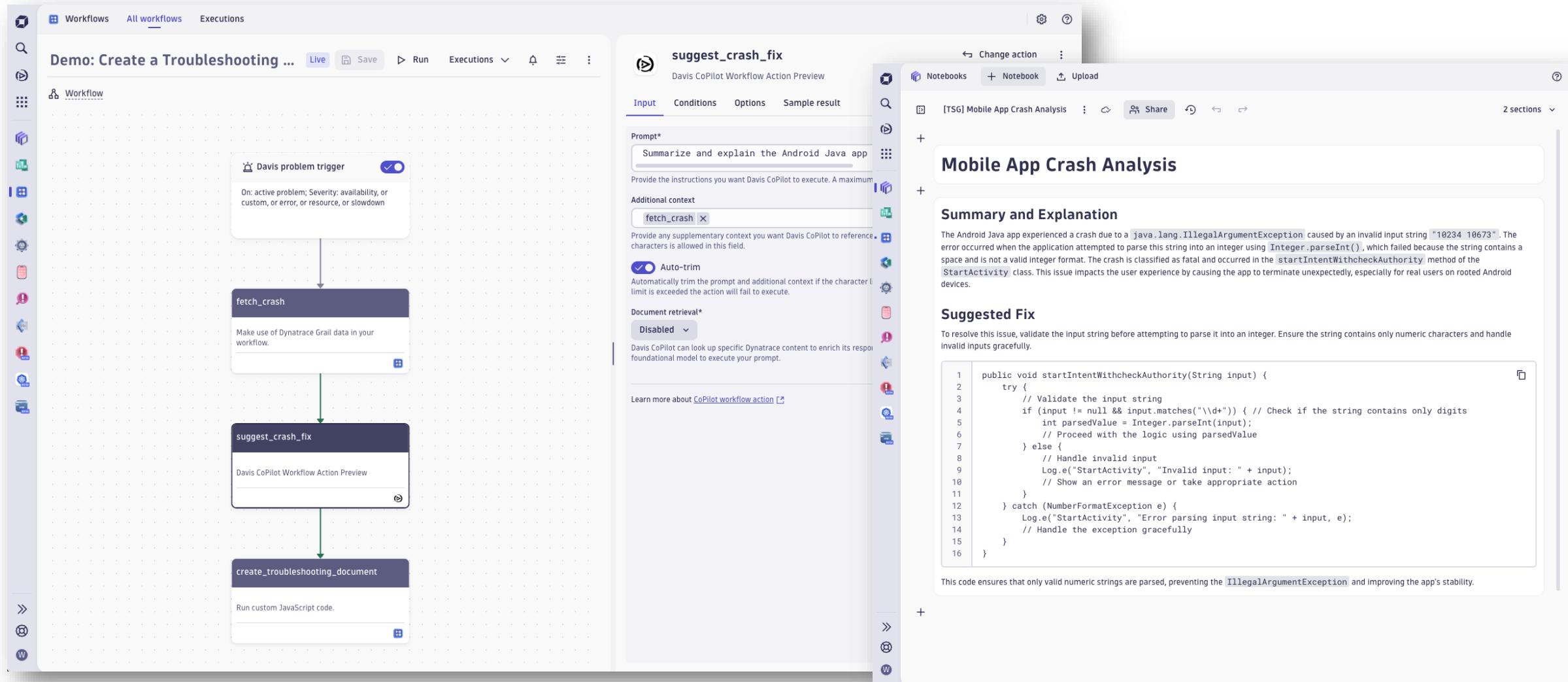
Dynatrace AI - Demo

Dynatrace AI Workflows

- Lowest possible entry barrier, as automation directly runs inside the Dynatrace platform.
- AI leverages the Dynatrace data access permission policies, with oAuth and service users as actor.
- Maximum flexibility and configurability.
- Visual workflow builder as an integral part of your Dynatrace tenant.



Demo Dynatrace Workflows unlocks AI Automation



The image shows a composite screenshot of the Dynatrace Workflows interface and the Davis CoPilot Workflow Action Preview.

Left Panel (Workflows): Displays a workflow titled "Demo: Create a Troubleshooting ...". The workflow consists of the following steps:

- Davis problem trigger:** Triggered on: active problem; Severity: availability, or custom, or error, or resource, or slowdown.
- fetch_crash:** Make use of Dynatrace Grail data in your workflow.
- suggest_crash_fix:** Davis CoPilot Workflow Action Preview. This step is highlighted, indicating it is the current focus.
- create_troubleshooting_document:** Run custom JavaScript code.

Middle Panel (Davis CoPilot Workflow Action Preview): The preview for the "suggest_crash_fix" step is shown. It includes:

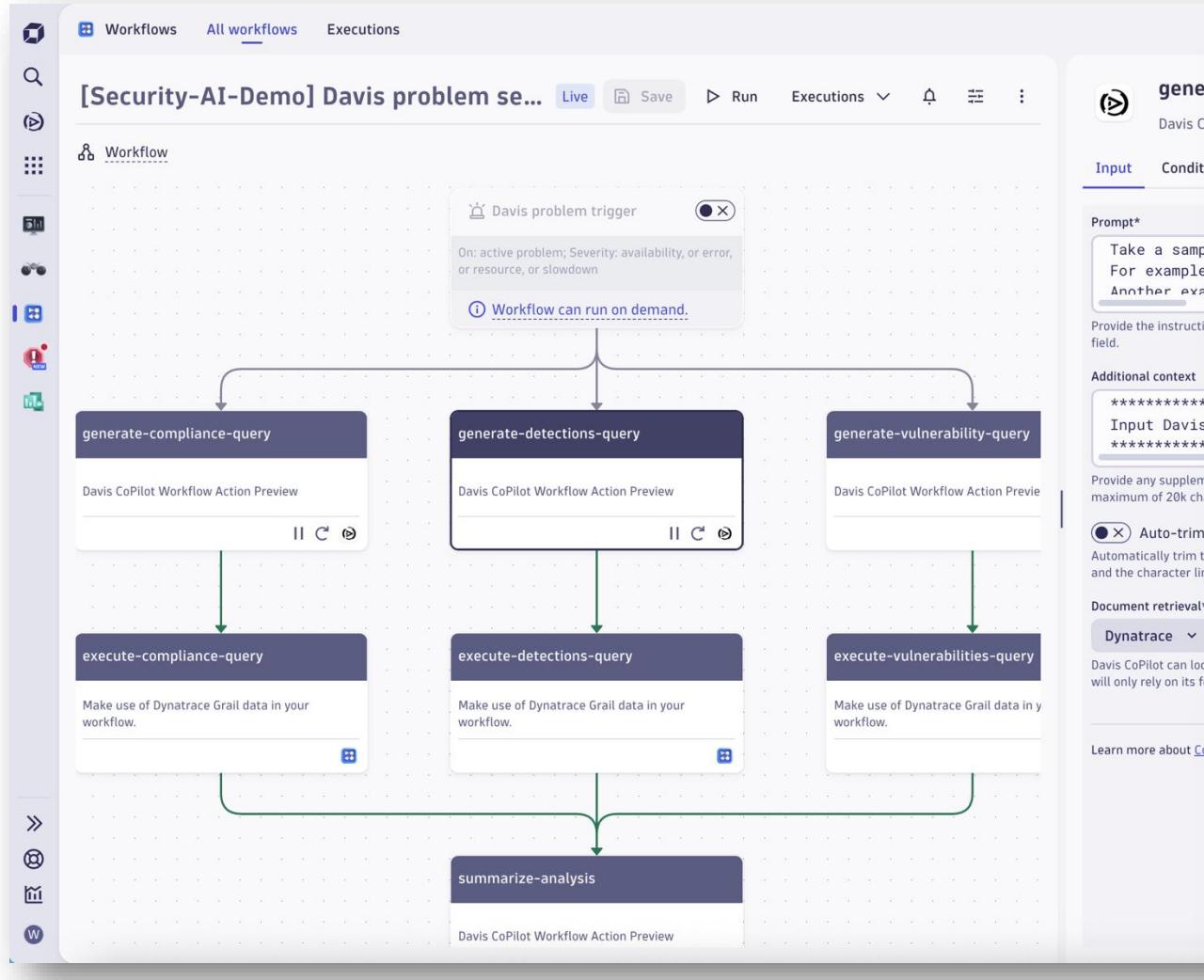
- Prompt:** Summarize and explain the Android Java app
- Additional context:** fetch_crash
- Auto-trim:** Checked
- Document retrieval:** Disabled

Right Panel (Mobile App Crash Analysis): A notebook titled "[TSG] Mobile App Crash Analysis" is open, showing the "Mobile App Crash Analysis" section. It includes:

- Summary and Explanation:** The Android Java app experienced a crash due to a `java.lang.IllegalArgumentException` caused by an invalid input string "10234 10673". The error occurred when the application attempted to parse this string into an integer using `Integer.parseInt()`, which failed because the string contains a space and is not a valid integer format. The crash is classified as fatal and occurred in the `startIntentWithCheckAuthority` method of the `StartActivity` class. This issue impacts the user experience by causing the app to terminate unexpectedly, especially for real users on rooted Android devices.
- Suggested Fix:** To resolve this issue, validate the input string before attempting to parse it into an integer. Ensure the string contains only numeric characters and handle invalid inputs gracefully.
- Code Snippet:**

```
1  public void startIntentWithCheckAuthority(String input) {  
2      try {  
3          // Validate the input string  
4          if (input != null && input.matches("\\d+")) { // Check if the string contains only digits  
5              int parsedValue = Integer.parseInt(input);  
6              // Proceed with the logic using parsedValue  
7          } else {  
8              // Handle invalid input  
9              Log.e("StartActivity", "Invalid input: " + input);  
10             // Show an error message or take appropriate action  
11         }  
12     } catch (NumberFormatException e) {  
13         Log.e("StartActivity", "Error parsing input string: " + input, e);  
14         // Handle the exception gracefully  
15     }  
16 }
```
- Note:** This code ensures that only valid numeric strings are parsed, preventing the `IllegalArgumentException` and improving the app's stability.

Demo Dynatrace Workflows unlocks Custom AI Security Checks



AI-Confidence Score: 75%

The AI-confidence score of 75% indicates a moderate to high likelihood that the detection findings contributed to the Davis problem (P-25104231). This score is based on the alignment of affected entities, Kubernetes clusters, and namespaces between the Davis problem and the detection findings.

Problem Description

The Davis problem (P-25104231) is categorized as a **Failure rate increase** affecting two services:

1. Service [eks-live] [easytrade-live-debugger] OrderController (SERVICE-8A5087303379DBEA)
2. Service [eks-live] [easytrade-live-debugger] :80 (SERVICE-EC21CDC3560BB0B4)

The problem occurred in the Kubernetes namespace `easytrade-live-debugger` within the cluster `eks-live` (UID: 8d179bf0-7549-46cf-99e5-1b903448e210). The root cause was identified as a failure rate increase in the `OrderController` service.

Detection Findings

- Detection 1:**
 - **Risk Level:** Critical
 - **Detection:** Blocked CMD injection attempt at `ProxyController.proxyUrlWithCurl() :163` and JNDI injection attempt at `JndiManager.lookup() :128`.
 - **Affected Entities:** `PROCESS_GROUP_INSTANCE-6FDE50F67864C2C1` (SpringBoot application `unguard-proxy-service` in namespace `unguard`).
 - **Kubernetes Cluster:** `eks-live` (UID: 8d179bf0-7549-46cf-99e5-1b903448e210).

Demo Dynatrace AI unlocks Explainable Root-Cause Analysis

Problems

User action duration degradation

Closed P-2511187093 Slowdown Started at Nov 19, 2025, 6:32 AM for 36 min

Affected frontends </> 1 Affected services 5 Affected infrastructure 1 Affected synthetic monitors - Affected users - Affected sessions - Affected business flows - Events 207

Overview Deployment Events Logs Troubleshooting

Impact

Frontends Services Infrastructure Synthetic monitors Environment

www.angular.easytravel.com Web application

Slowdown User action duration degradation View frontend

Root cause

easyTravelBusiness Service Root cause

Slowdown Response time degradation

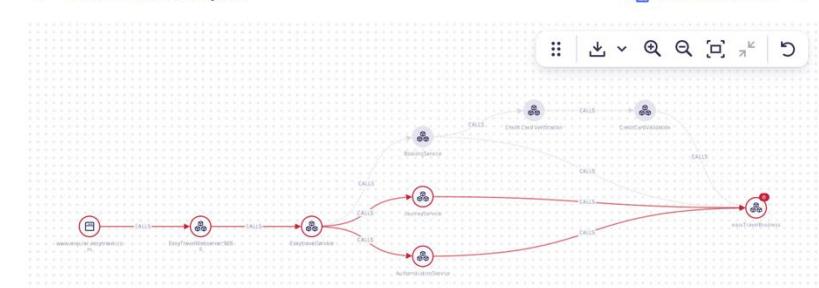
View service

Automation and remediation

Workflow Alert notification to team slack Last execution 11/19/2025, 6:32:50 AM State Success

Visual resolution path

View Smartscape



Demo Dynatrace AI democratizes Data Analytics

Helsinki Demo ▾

W give me the CPU usage average over the last 7 days

> Create DQL query

> Run DQL query

⌚ The average CPU usage over the last 7 days is approximately 24.47%.

↳

W give me a forecast of the next 12 hours for CPU usage

> Create DQL query

> Timeseries forecast

⌚ The forecast for CPU usage over the next 12 hours has been successfully generated. Average CPU utilization values:

1. Start Time: Now
2. Forecast Horizon: Next 12 hours
3. Predicted Values (Average CPU Utilization):
 - Hour 1: 24.62%
 - Hour 2: 24.71%
 - Hour 3: 24.72%

> Timeseries Characteristic Analyzer

⌚ The metric characteristics of the service load (`dt.service.request.count`) for the `frontend` service (`SERVICE-612D78ED86A2A09B`) over the last 2 weeks are summarized below:

Statistical Analysis:

- Maximum Load: 76,099 requests
- Minimum Load: 55,914 requests
- Mean Load: 73,050.87 requests
- Median Load: 73,199 requests
- Variance: 3,575,258.10 (indicating moderate variability)
- Mean Absolute Deviation (MAD): 652 requests
- Number of Distinct Values: 112
- Most Frequent Value Ratio: 1.77% (indicating no significant repetition of values)

Trends:

- Regression Line Slope: -10.87 (indicating a slight downward trend in load over time).
- P-Value for Slope: 0.044 (statistically significant downward trend).

Quantiles:

- 5th Percentile: 71,580 requests
- Median (50th Percentile): 73,199 requests
- 95th Percentile: 74,951.6 requests

Missing Data:

- Number of Gaps: 0 (no missing data)
- Longest Gap: 0 (data is complete)

Periodicity:

- No significant periodic patterns were detected.

This analysis provides insights into the load behavior and trends for the service. Let me know if you'd like further details or visualization!



CLOUD DONE RIGHT