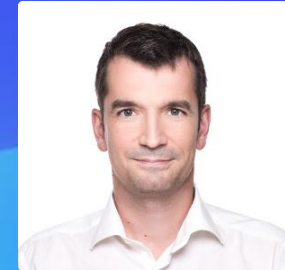


# Beyond the AI Hype

Agentic AI in Observability - A Reality Check

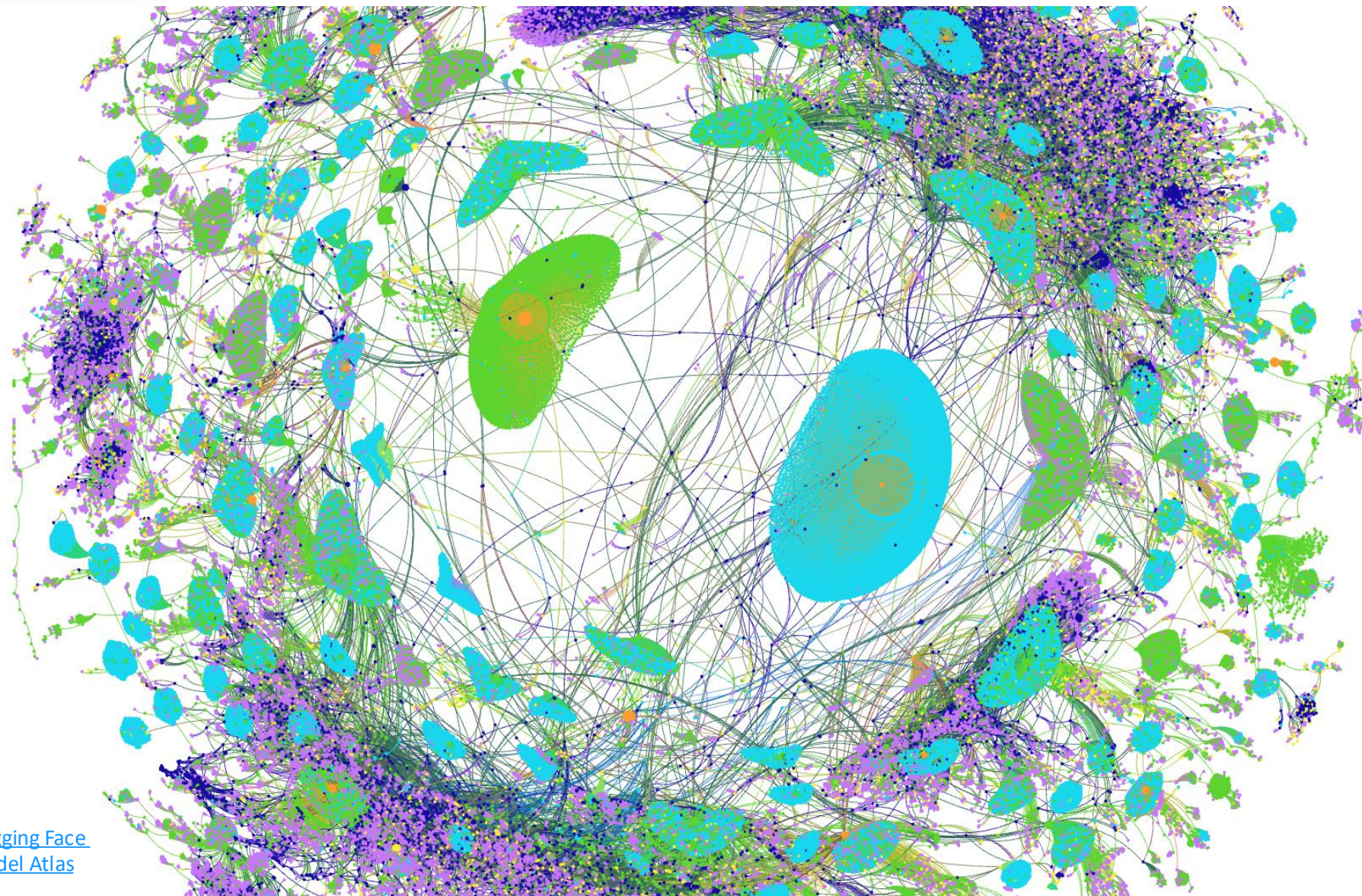


PRESENTER

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

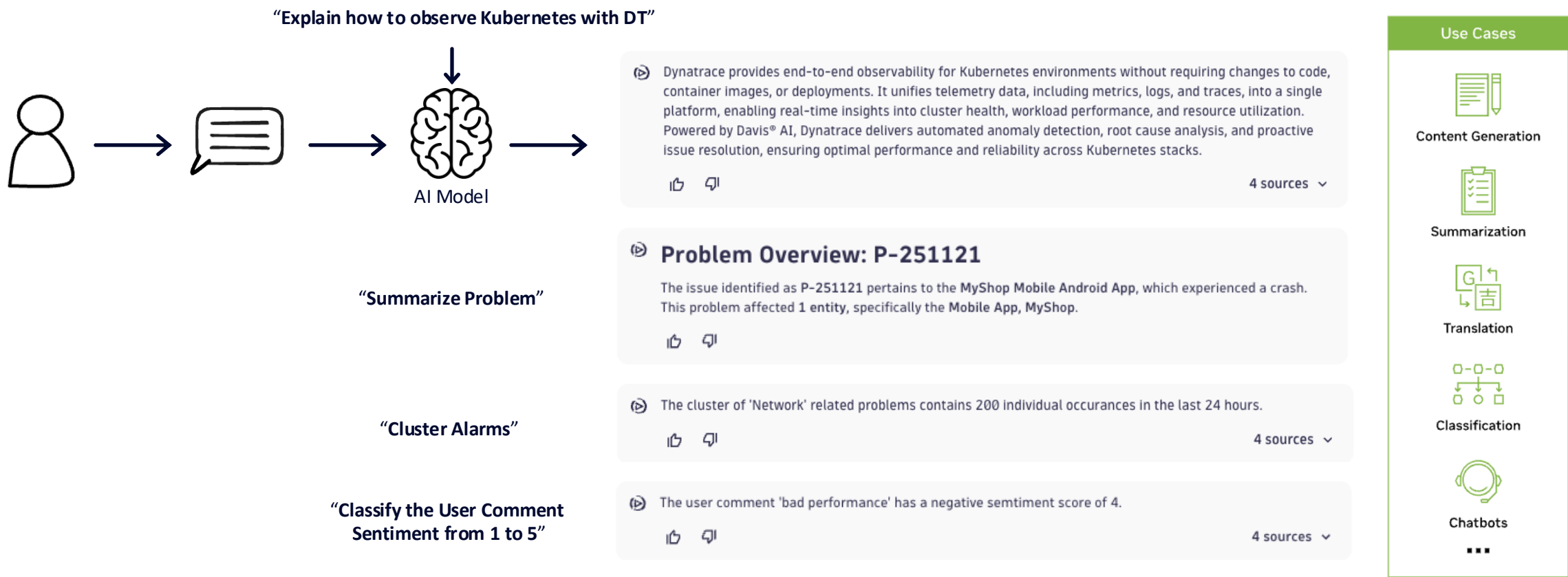


**500M**  
data sets to finetune  
AI models



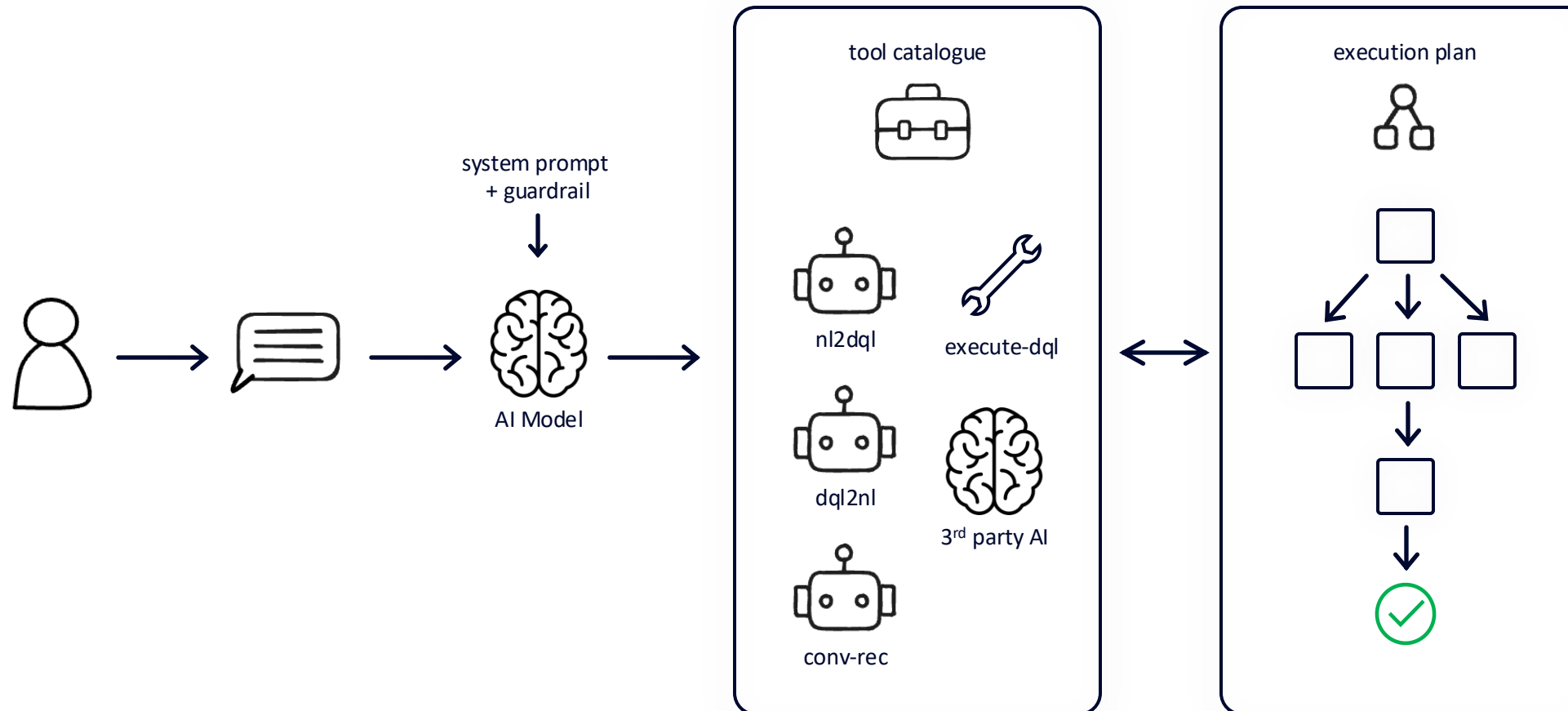
# What's a Generative LLM AI Model?

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



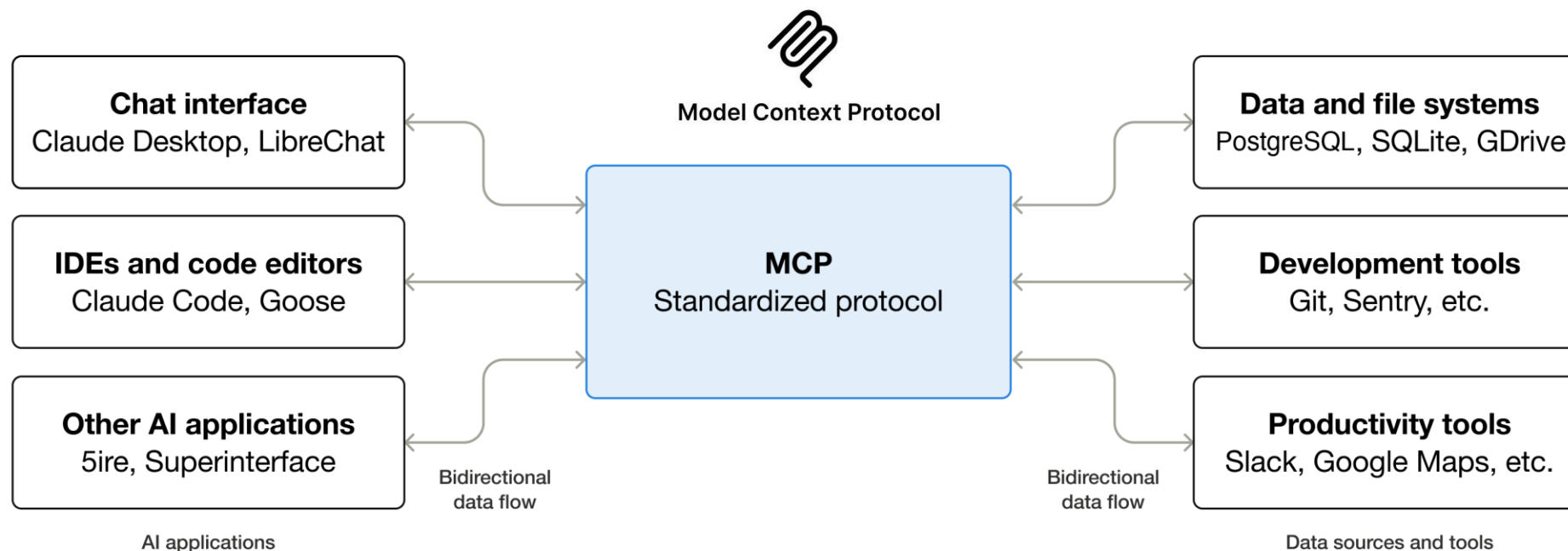
# What's an AI Agent?

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.

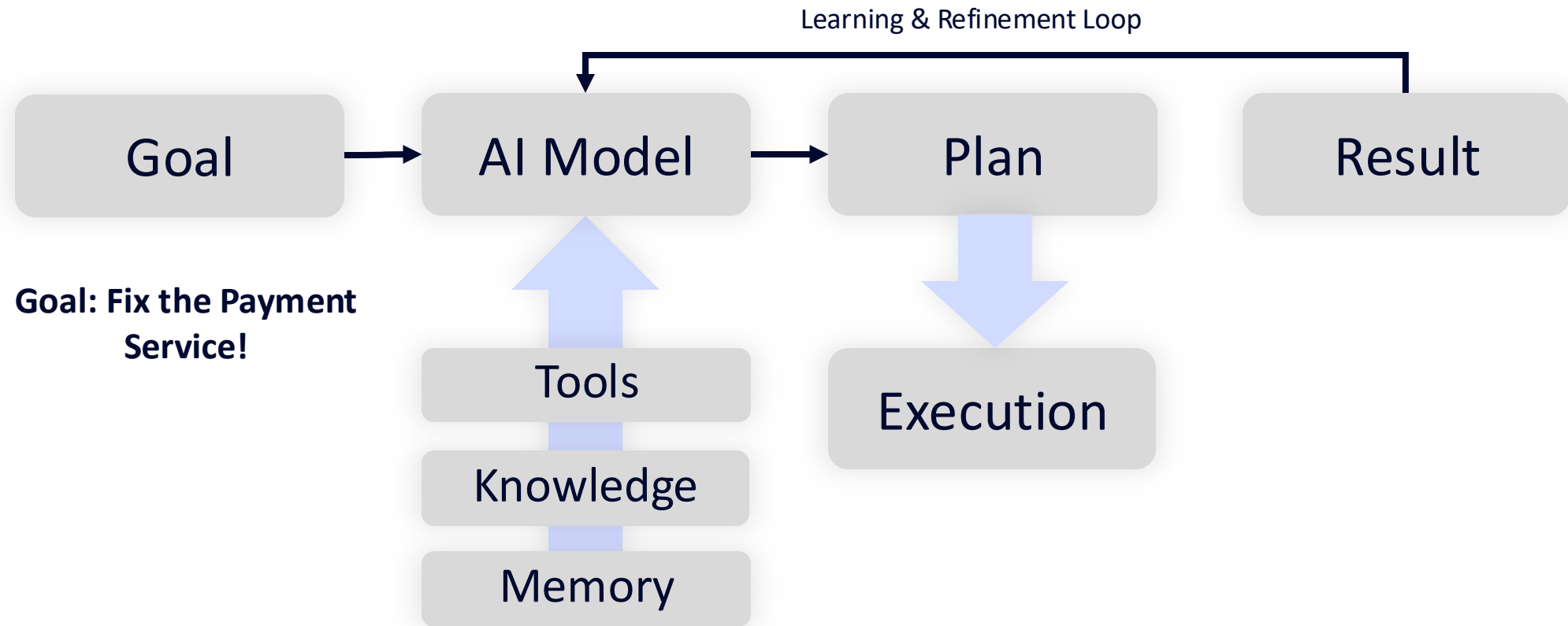


# Agentic AI – The Rise of Model Context Protocol

- 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

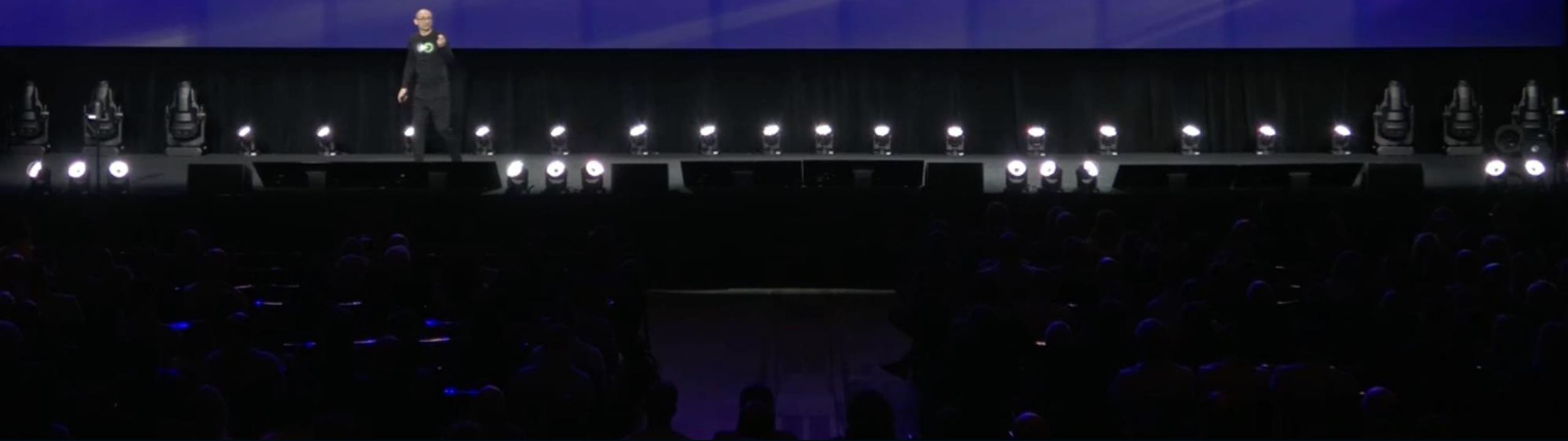
# Dynatrace is evolving into an agentic AI platform

Built for observability, security, and business

Orchestrating  
internal and  
external AI agents

Creating AI agents  
from app  
functionality

Collaborating with  
other external AI  
agents



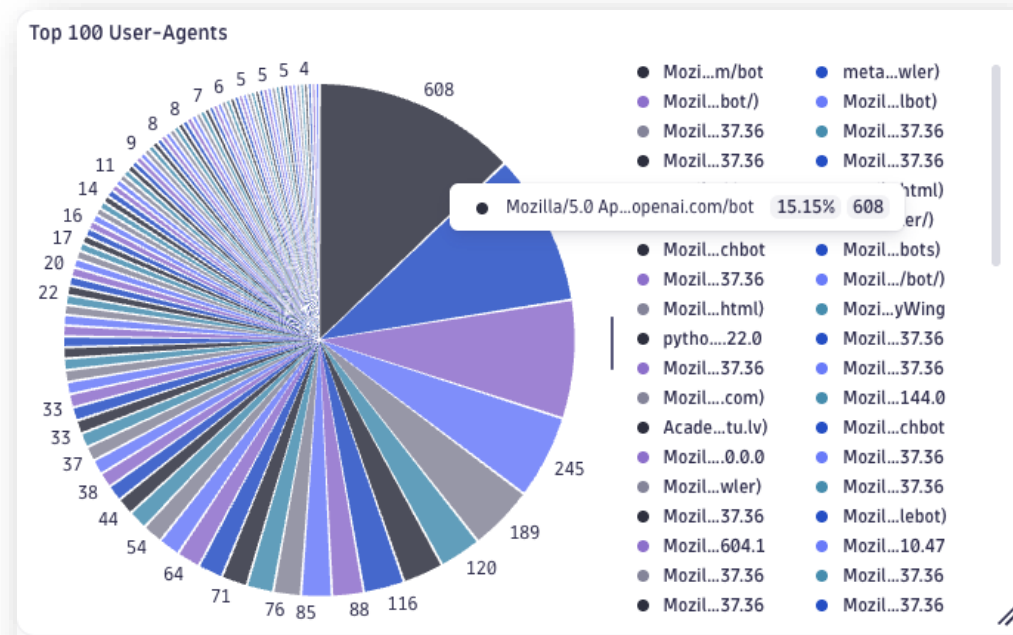
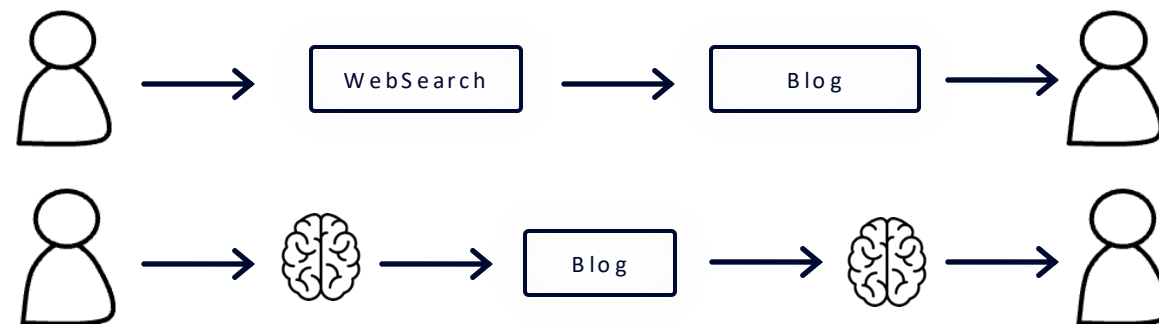


# Agentic AI - Fundamentally Changes User Interaction

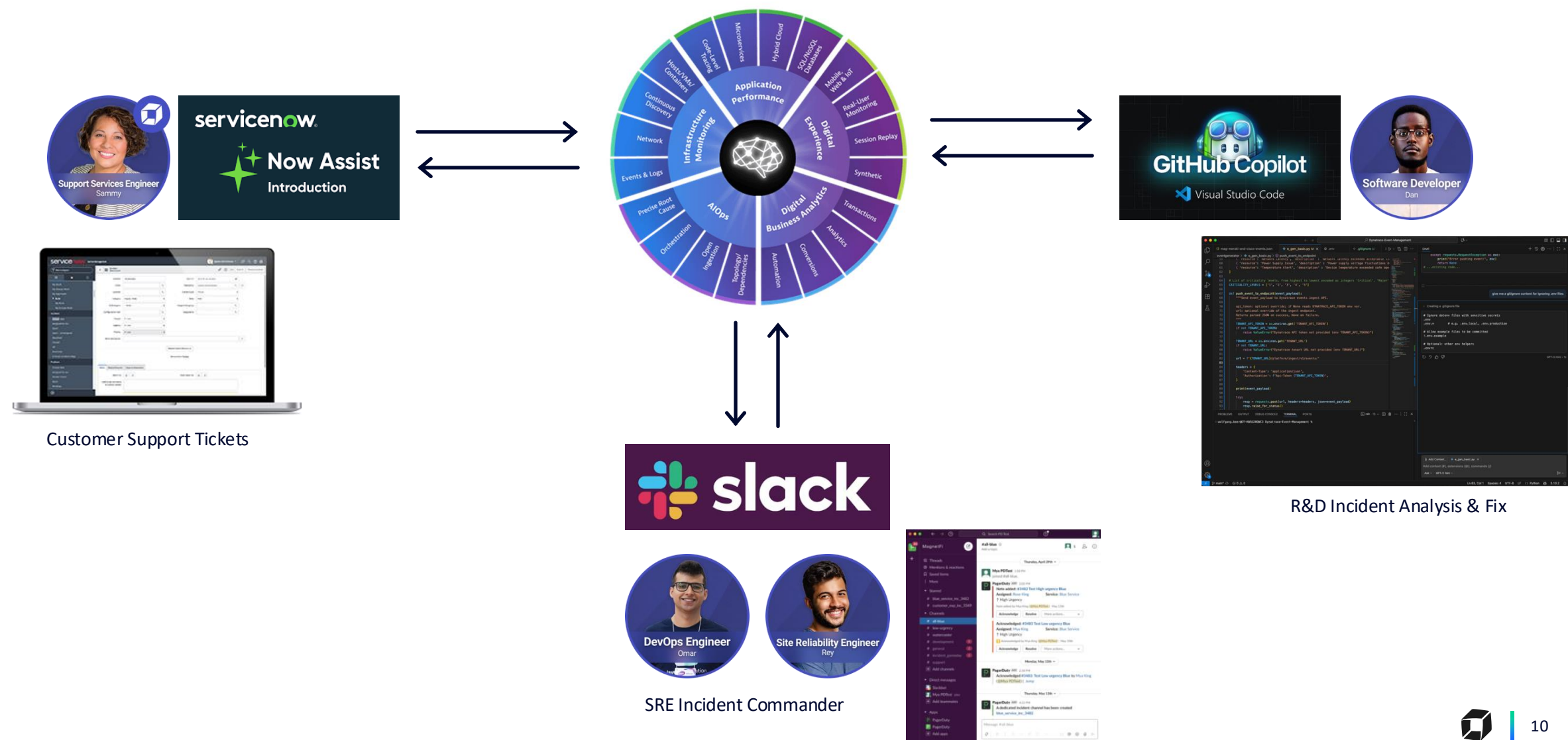
- Agentic AI breaks down silos!
- Democratizes access to information.
- Removes barriers for non-tech roles.
  - Ask data questions.
  - Help formulate and test hypothesis.
  - Let AI configure something complex.
  - Vibe your own app, based on a use-case.
- Etc.

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.

“

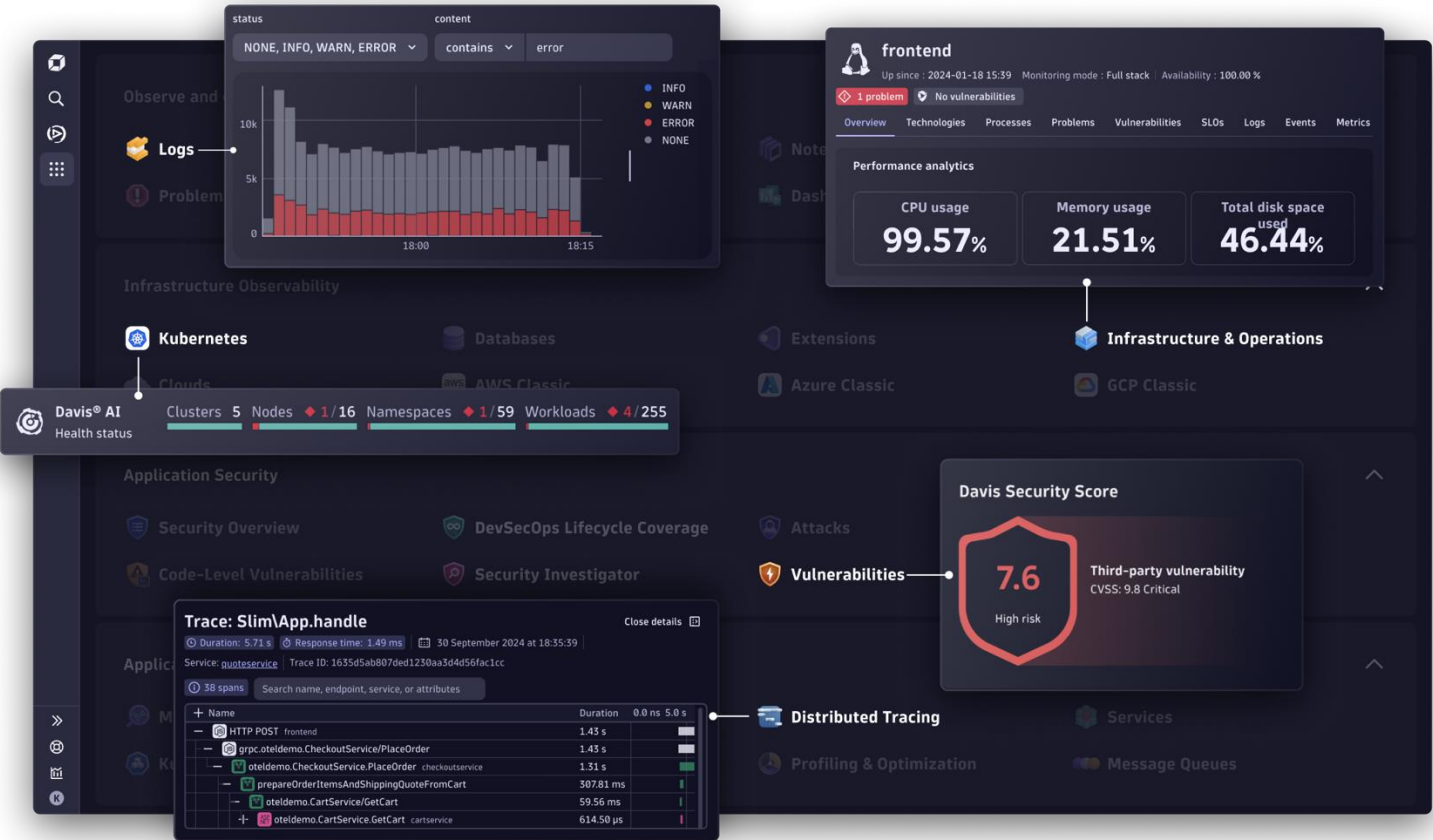
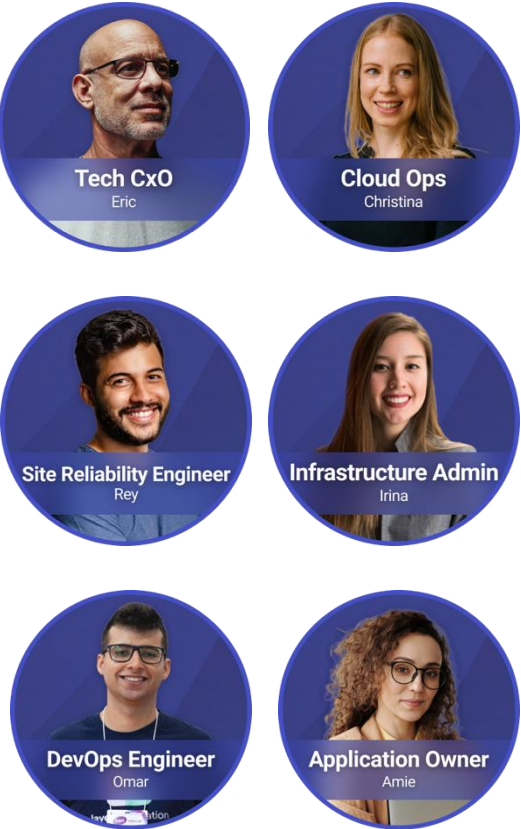
**You never know who's  
swimming naked until the  
tide goes out.**

— Warren Buffett



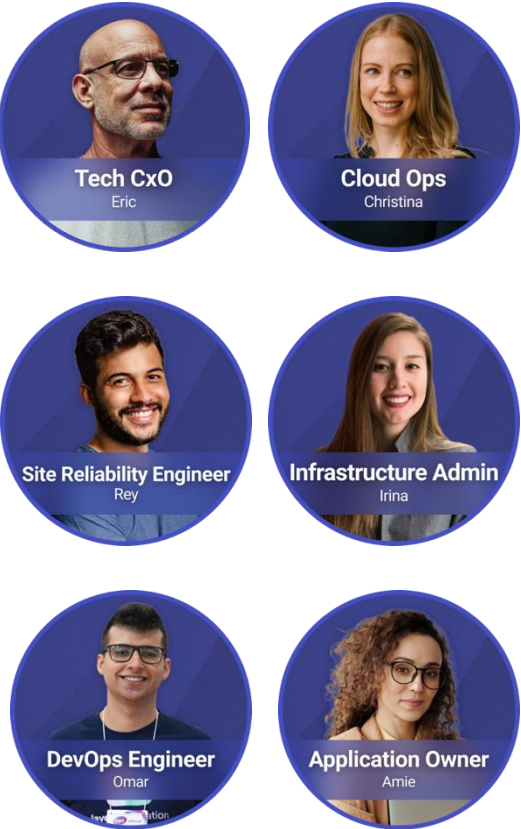
”

# One AI Platform to Safeguard your Digital Systems





# One AI Platform to Safeguard your Digital Systems



# Build your own Agent

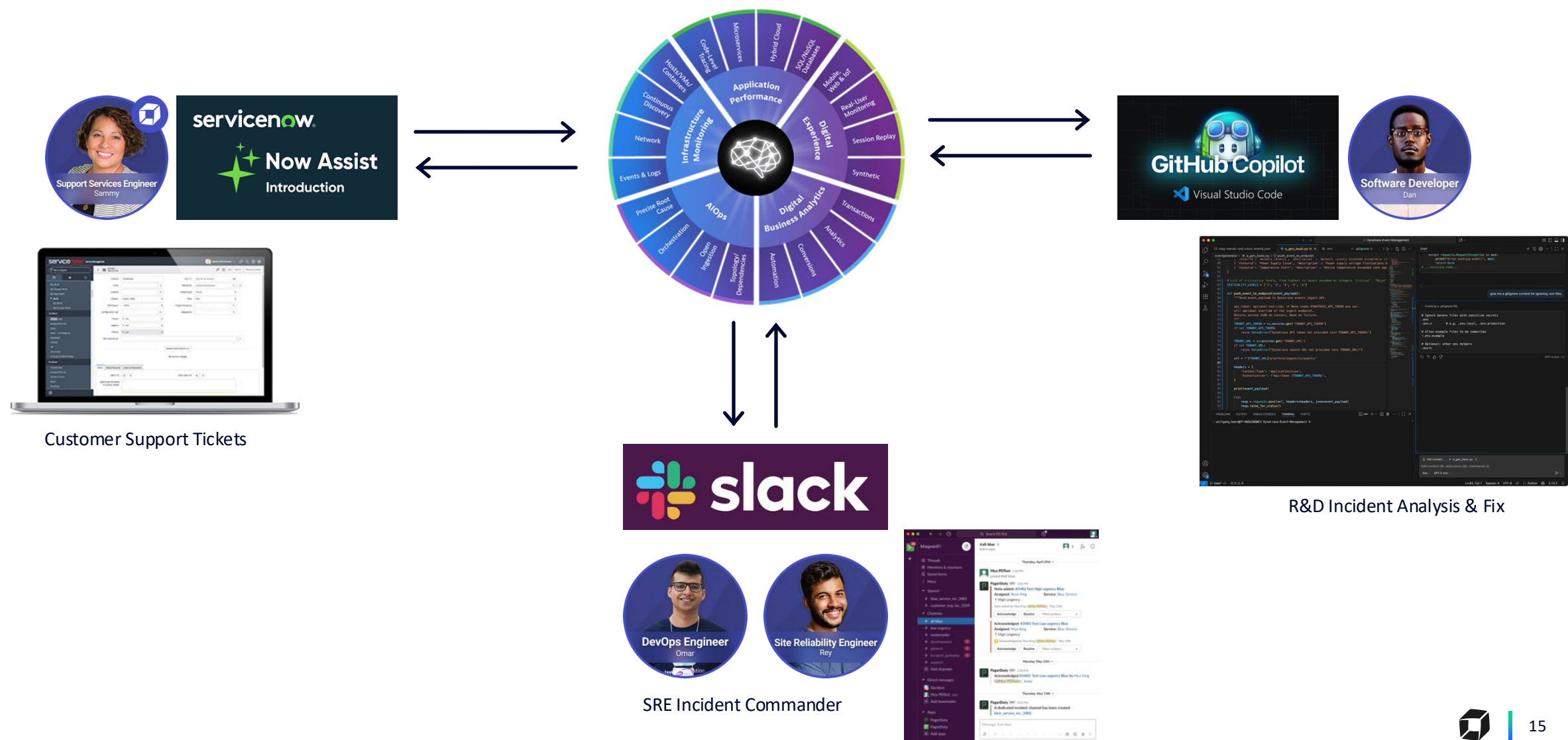
How Dynatrace AI enables domain-specific agentic use-cases



PRESENTER

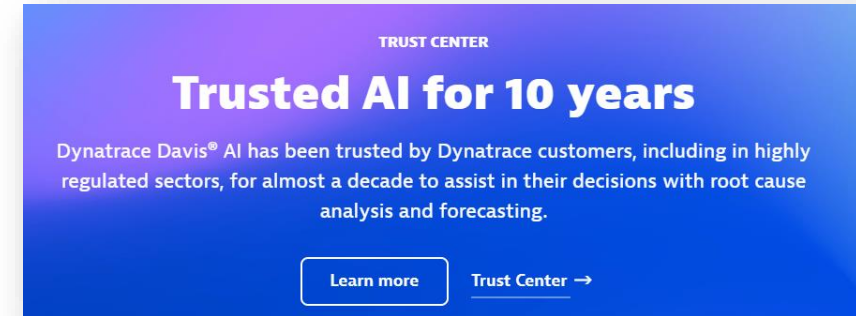
Wolfgang Beer  
Fellow Product Manager  
Dynatrace

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



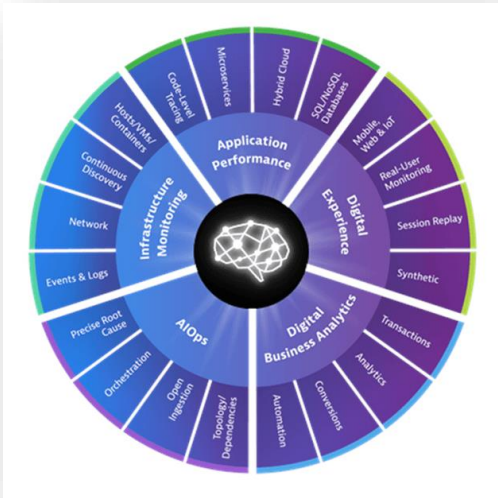
# Dynatrace AI – Responsible & Transparent

- 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 code editor window titled "Dynatrace-Event-Management". The code is in Python and defines a function "push\_event\_to\_endpoint" that interacts with the Dynatrace API. The code includes comments in Chinese and English, and uses the "requests" library. On the right side of the editor, there is a "CHAT" panel with a text input field and a "give me a .gitignore content for ignoring .env files" button. Below the chat panel, there is a "PROBLEMS" panel showing a list of errors.


# Dynatrace Agentic Interaction with ServiceNow IT Operations



Express List +

Active Alerts 4

Search filtered alertsCloseLast 24 hoursLive

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






Active Alerts 4

Search filtered alerts

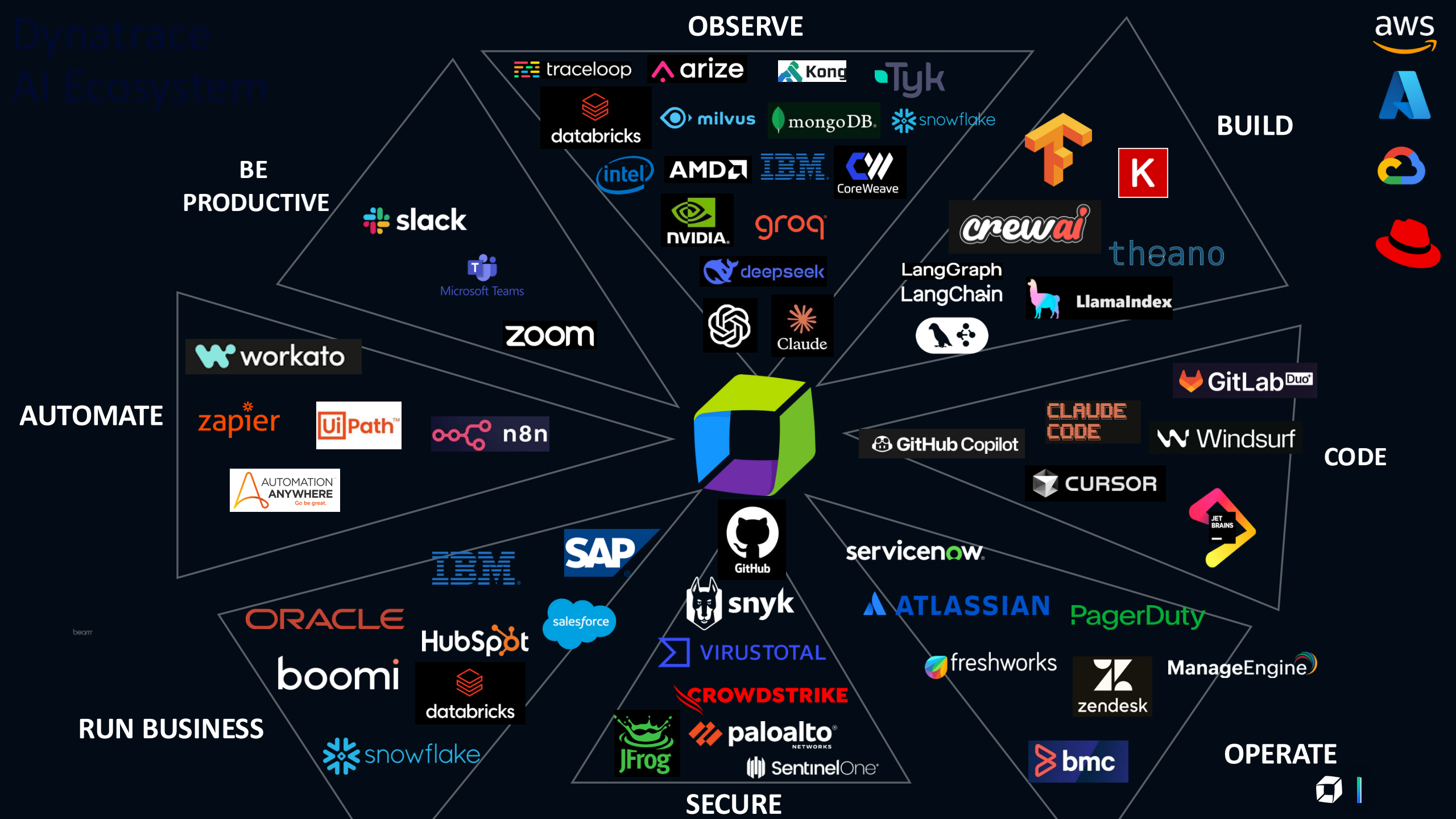
Close

Last 24 hours

Live

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

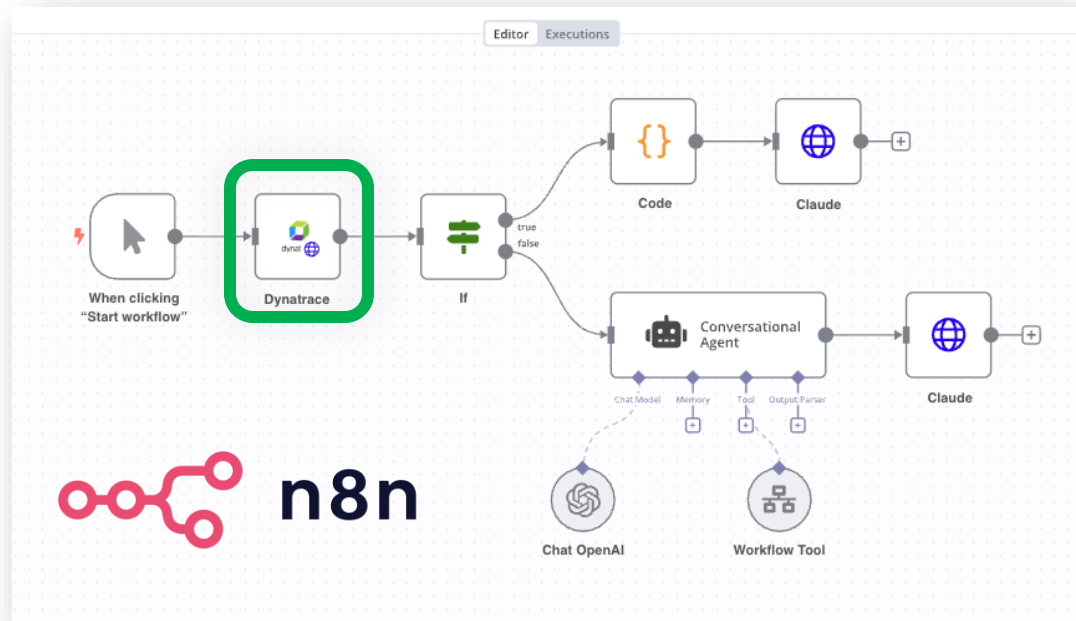
# Dynatrace AI Ecosystem



# 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



```
helsinki > agent.py > ...
13 def get_current_date_time_utc() -> dict:
22
23     now_utc = datetime.now()
24     report = {
25         'The current time is {now_utc.strftime("%Y-%m-%d %H:%M:%S")}
26     }
27     return {"status": "success", "report": report}
28
29
30 root_agent = Agent(
31     name="HelsinkiAgent",
32     model="gemini-2.0-flash",
33     description=(
34         "Agent that is trigge
35     ),
36     instruction=(
37         "You are a helpful ag
38         "Always reply with cl
39     ),
40     before_tool_callback=lamb
41     print(f"Executing too
42     ),
43     tools=[
44         get_current_date_time
45         MCPToolset(
46             connection_params
47             url='https://
48             headers={
49                 'Authoriz
50             }
51         ),
52         tool_filter={}
53     ),
54     MCPToolset(
55         connection_params
56     )
57 )
```

### Configure Tools

Select tools that are available to chat. **52 Selected** **OK**

- ✓ 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

### PROBLEMS

wolfgang.beer@DT-KW5G28QWC3 dt-agenti

```
>>> Tool: dynatrace-mcp-experimental
>>> Tool: dynatrace-mcp-experimental
>>> Tool: dynatrace-mcp-experimental
>>> Calling tool 'dynatrace-mcp_exce
Session termination failed: 400
Session termination failed: 400
Session termination failed: 400
Session termination failed: 400
<<< Result: Query metadata:
{
  "canonicalQuery": "fetch events\n|
  "timezone": "Z",
  "query": "fetch events | limit 50",
  "scannedRecords": 316,
  "dqVersion": "V1.0",
  "scannedBytes": 14585,
  "scannedDataPoints": 0,
  "analysisTimeframe": {
    "start": "2025-11-19T12:17:32.51",
    "end": "2025-11-19T14:17:32.51",
  },
  "locale": "und",
  "executionTimeInMilliseconds": 20,
  "notifications": [],
  "queryId": "0ae84d02-47e6-48ff-b7",
  "sampled": false
}
```

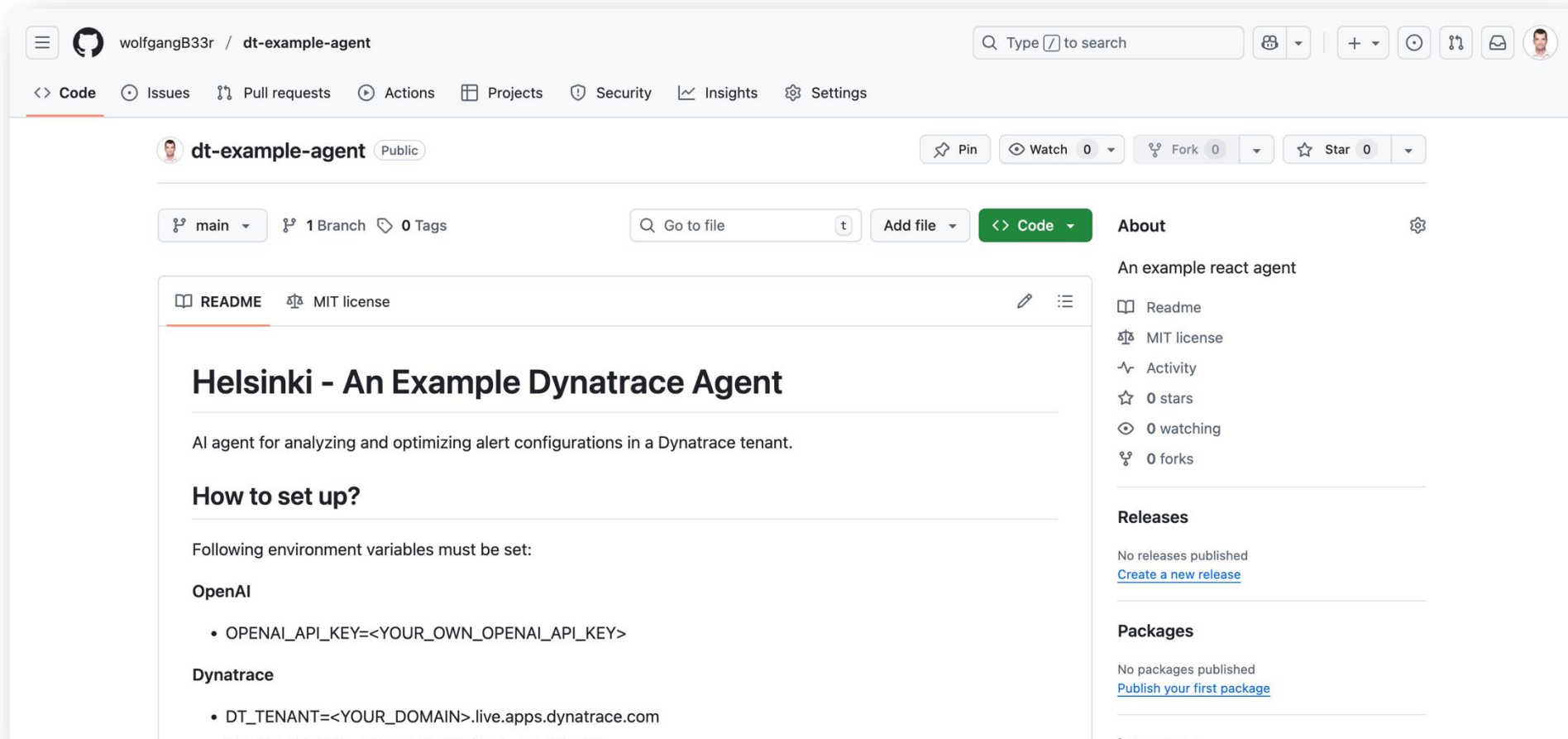




# Demo Build an SREAgent with Dynatrace and LangGraph



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



# Demo Build an SREAgent with Dynatrace and LangGraph



```
# Initialize the MultiServerMCPClient with the client
client = MultiServerMCPClient(
    {
        "dynatrace-mcp": {
            "transport": "streamable_http",
            "url": 'https://{tenant}/platform/api/v2/mcp',
            "headers": {
                "Authorization": "Bearer {token}"
            }
        }
    }
)
```

```
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.ainvoke({"messages": input_messages}, config=config)
```

```
tenant=DT_TENANT),
```

# 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
                }
            )
        )
    ],
)
```

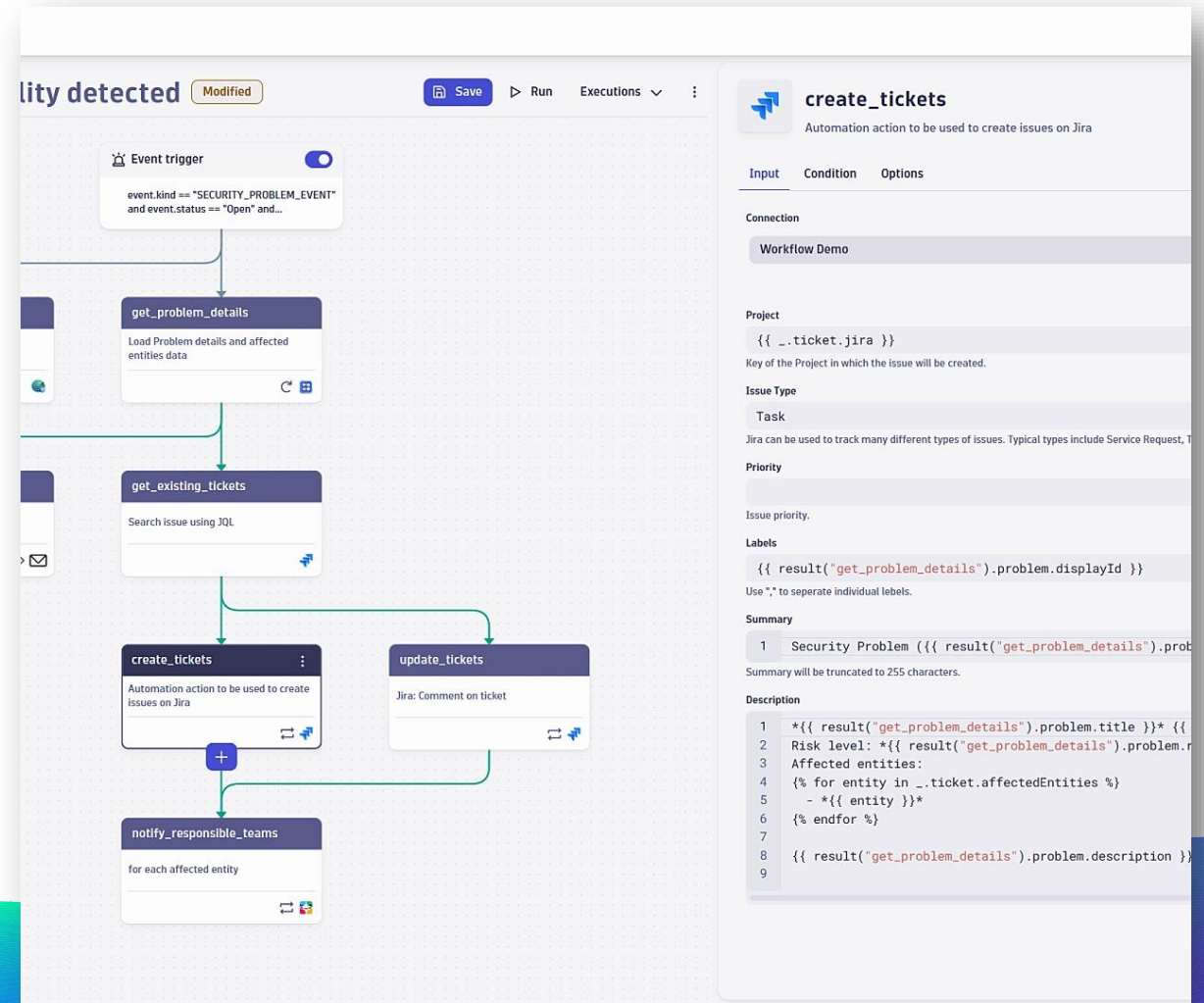
# Show & Tell # 2

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

**Workflow Editor:** The workflow titled "Demo: Create a Troubleshooting ..." is configured with the following steps:

- Davis problem trigger:** 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.
- create\_troubleshooting\_document:** Run custom JavaScript code.

**suggest\_crash\_fix Action Configuration:**

- Prompt:** Summarize and explain the Android Java app
- Additional context:** fetch\_crash
- Auto-trim:** Enabled (Automatically trim the prompt and additional context if the character limit is exceeded the action will fail to execute).
- Document retrieval:** Disabled

**Notebooks View:** The notebook titled "[TSG] Mobile App Crash Analysis" contains the following sections:

- Mobile App Crash Analysis**
- Summary and Explanation**

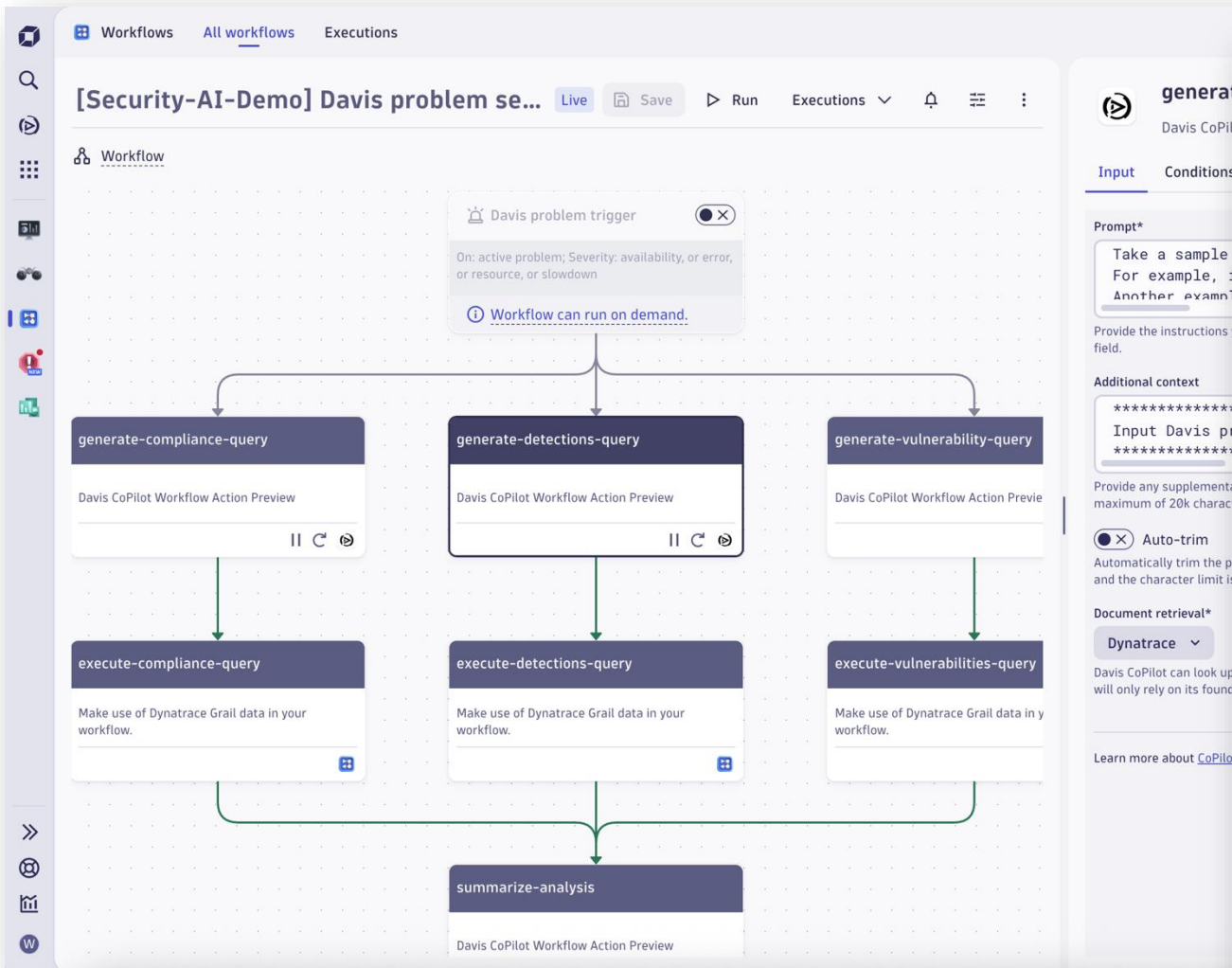
The Android Java app experienced a crash due to a `java.lang.IllegalArgumentException` caused by an invalid input string `"10234 10673"`. The error occurred when the application attempted to parse this string into an integer using `Integer.parseInt()`, which failed because the string contains a space 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.

```
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 }
```

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

### 1. Detection 1:

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



# Demo Dynatrace AI unlocks Explainable Root-Cause Analysis

The screenshot displays the Dynatrace AI interface for a problem titled "User action duration degradation". The problem status is "Closed" with ID "P-2511187093", categorized as a "Slowdown" that started on Nov 19, 2025, at 6:32 AM and lasted for 36 minutes. The interface includes a top navigation bar with tabs for Overview, Deployment, Events, Logs, and Troubleshooting. Below the problem title, there are several summary cards: Affected frontends (1), Affected services (5), Affected infrastructure (1), Affected synthetic monitors (-), Affected users (-), Affected sessions (-), Affected business flows (-), and Events (207). The main content area is divided into three sections: Impact, Root cause, and Visual resolution path. The Impact section shows the affected frontend "www.angular.easytravel.com" and the specific issue "Slowdown: User action duration degradation". The Root cause section identifies the service "easyTravelBusiness" as the root cause, with a "Slowdown" in "Response time degradation". The Visual resolution path section displays a complex dependency graph showing the flow of data and services involved in the problem, with a "View Smartscape" button. The bottom section, Automation and remediation, shows a workflow "Alert notification to team slack" that was successfully executed on 11/19/2025 at 6:32:50 AM.

**Problems**

## User action duration degradation

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

⚙ Explain problem 📧 Send ✕

Affected frontends  
1

Affected services  
5

Affected infrastructure  
1

Affected synthetic monit...  
-

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

### Visual resolution path

View Smartscape

### Automation and remediation

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



# 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. Here are the 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