Red Hat Developer





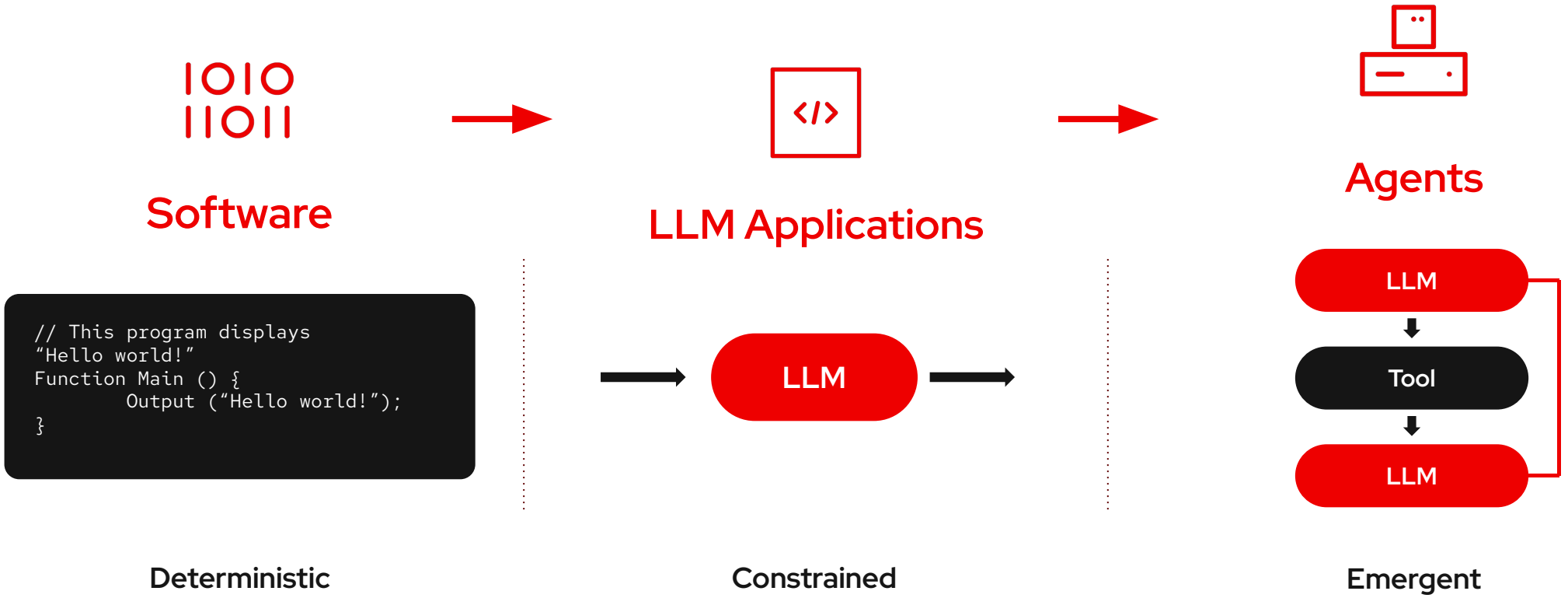
Observing AI Agents in Production

SREday Q4: 2026

Cedric Clyburn
Senior Developer Advocate
@cedricclyburn

Building with AI is fundamentally different

The typical AI adoption lifecycle for engineers



Do you actually know what your **agent is doing or can do?**

Red Hat

So how do we actually observe AI agents?

From debugging code to debugging reasoning

Software observability

- Deterministic
- The code is the source of truth
- A stack trace names the function
- You can write the tests first

Agent observability

- The path appears when it runs
- The trace is the source of truth
- What failed was the reasoning
- You learn the tests from production

Red Hat

The 3 pillars of agentic observability

Runs, traces, and threads

THREAD Claude Code session with a user

TRACE Build me a flappy bird clone... thx

RUN: llm

RUN: tool_Write

RUN: llm

TRACE Uhhhh can you make it a little easier

(this continues in the same thread, and most AI agents are this type of “conversation”)

Red Hat

Why agent evaluation differs from software evaluation

1. You're testing reasoning, not code paths

- **Single-step:** Did the agent make the right decision at this moment?
- **Full-turn:** Did the agent perform well in an end-to-end execution?
- **Multi-turn:** Did the agent maintain context across a conversation?

2. Production is where you discover what to test for offline

Red Hat

The different levels of agent evaluations

Single-step evaluation: Unit tests for decisions

- Validates individual agent decisions in isolation
- Example: "*Did the booking agent check availability before scheduling?*"
- Fast to execute with clear pass/fail criteria

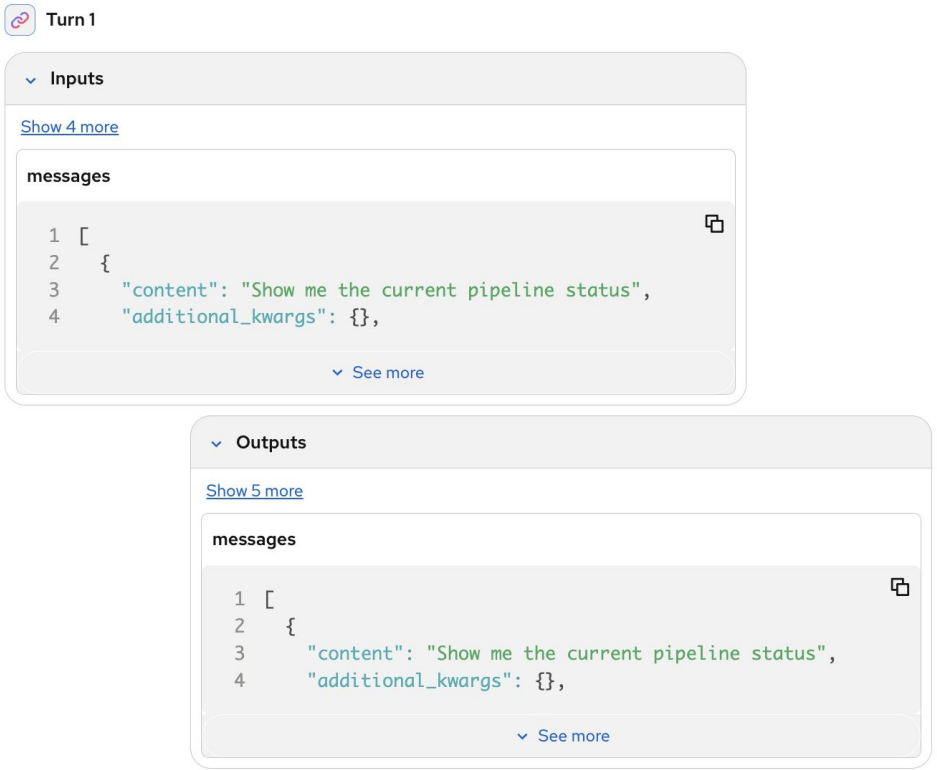


Red Hat

The different levels of agent evaluations

Full-turn evaluation: End-to-end trajectory assessment

- Validates complete agent runs from input to final output
- Example: "Did the coding agent produce working, tested code?"
- Tests real behavior including tool usage and state changes

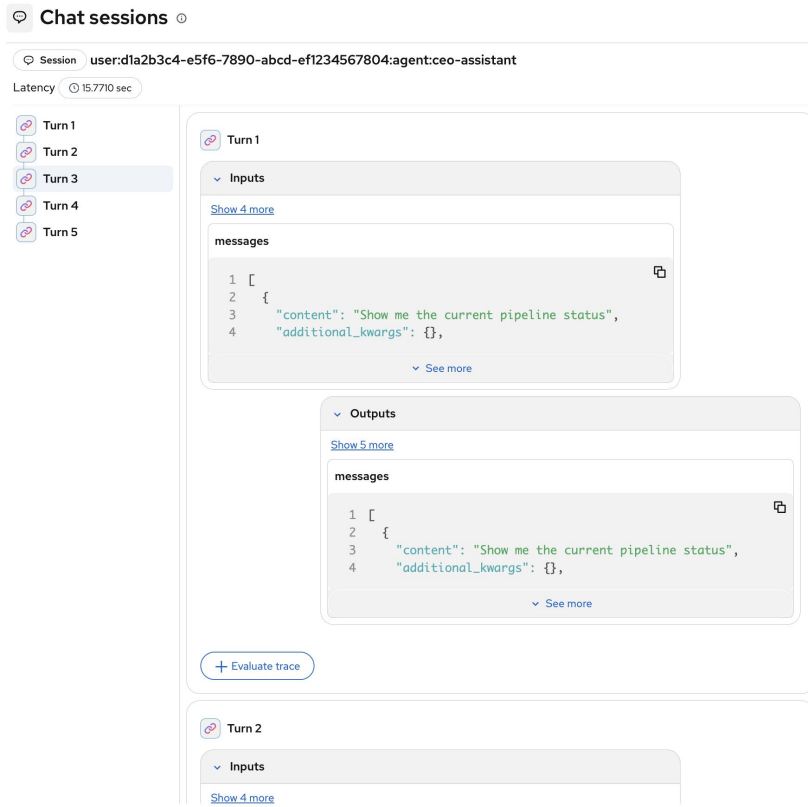


Red Hat

The different levels of agent evaluations

Multi-turn evaluation: Realistic conversation flows

- Validates threads across multiple conversation turns
- Example: "Does the agent remember user preferences from Turn 1 in Turn 3?"
- Most realistic test of production behavior and context persistence



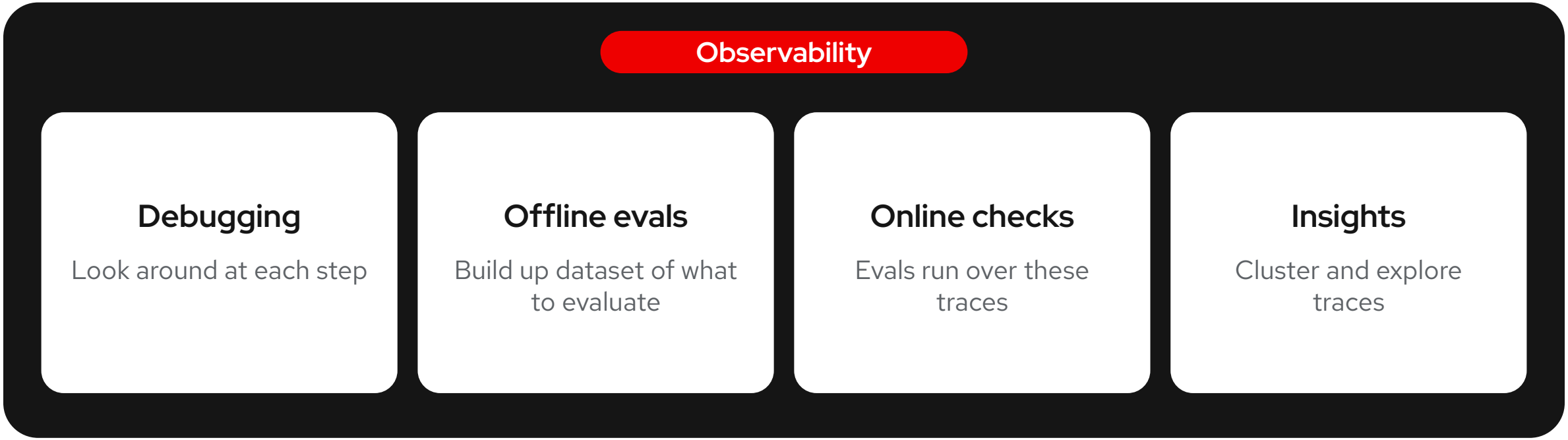
Red Hat

What’s the pros and cons of each?

	Pros	Harder
Run, single step	Easier to define metrics for	It goes stale when the agent changes.
Trace, full turn	Easier to define inputs for	Scoring the answer is the hard part.
Thread, multi-turn	Most realistic user behavior	The next line depends on the last.

Red Hat

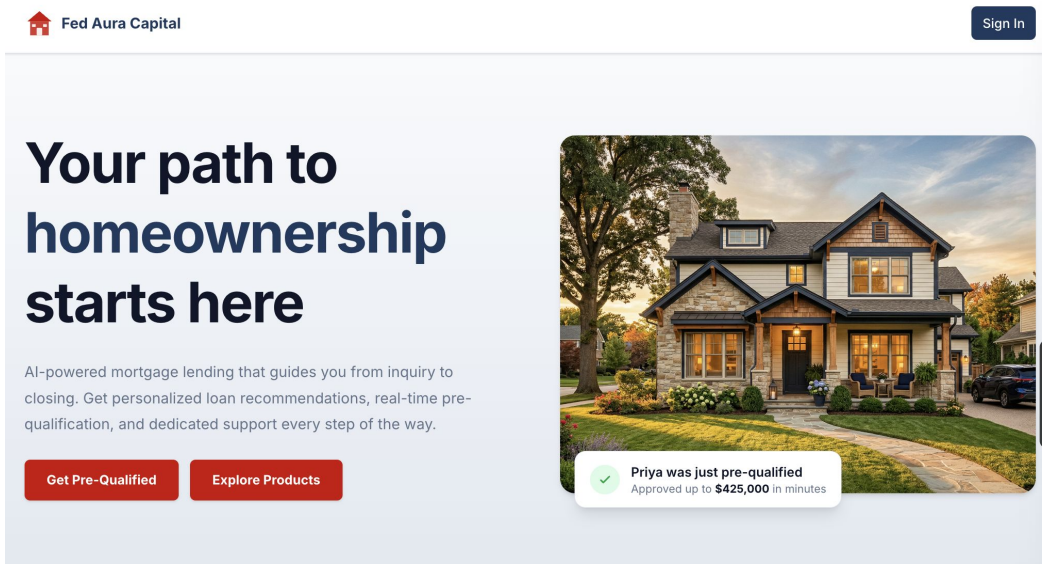
How agent observability powers evaluation



Red Hat

Enough yap...

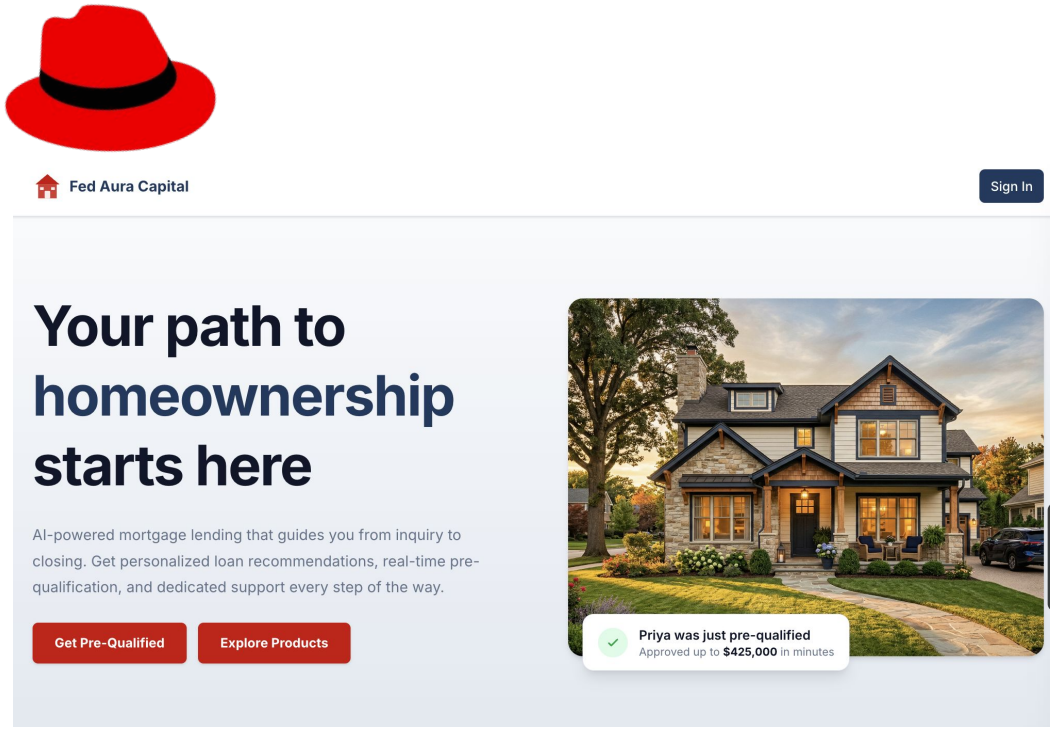
demo time!



Red Hat

Enough yap...

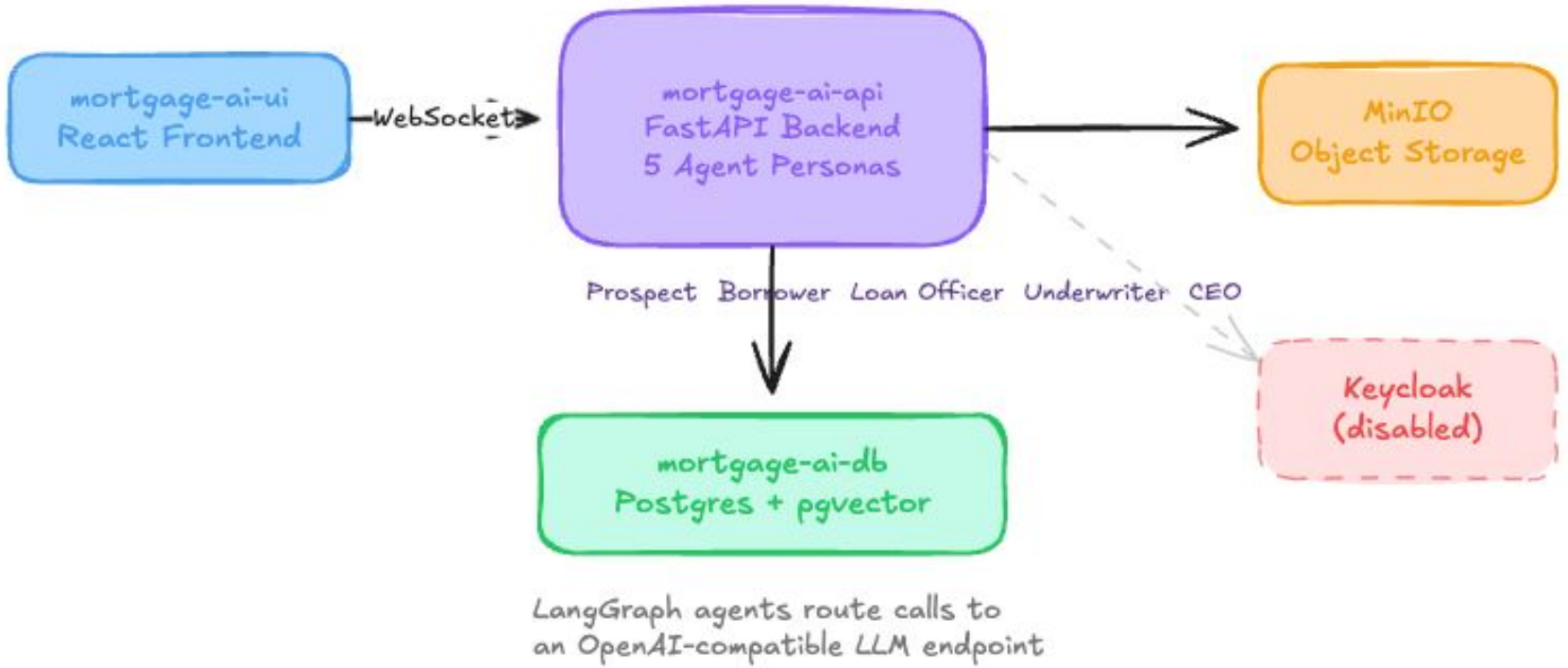
demo time!



Red Hat

How our multi-agent setup looks

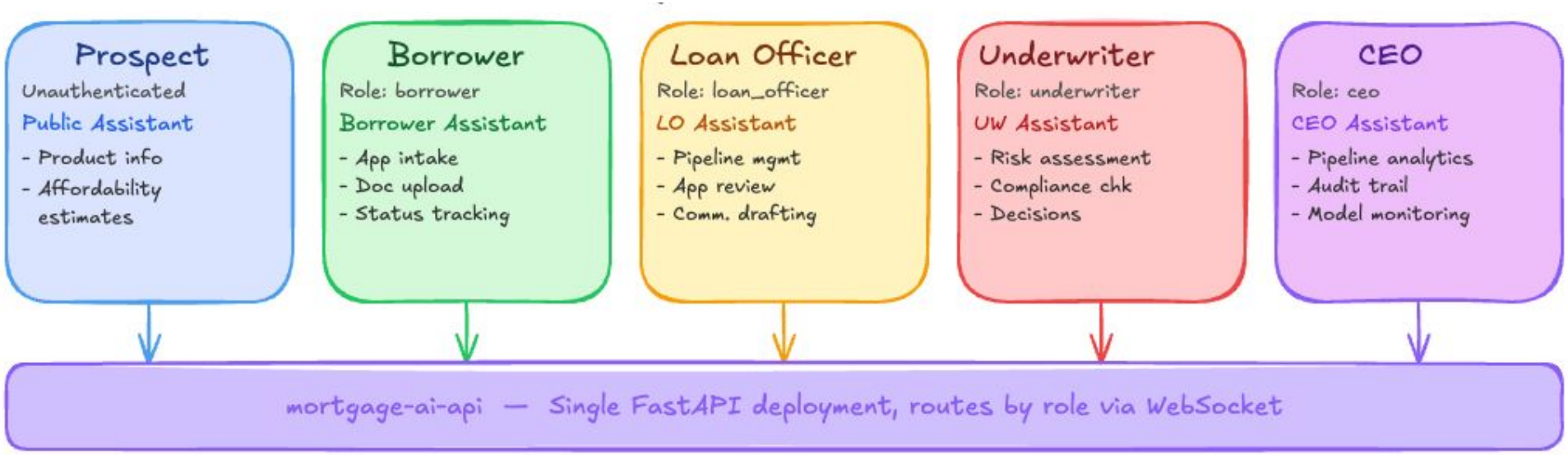
High level architecture



Red Hat

Each of these agents has a specific role/persona

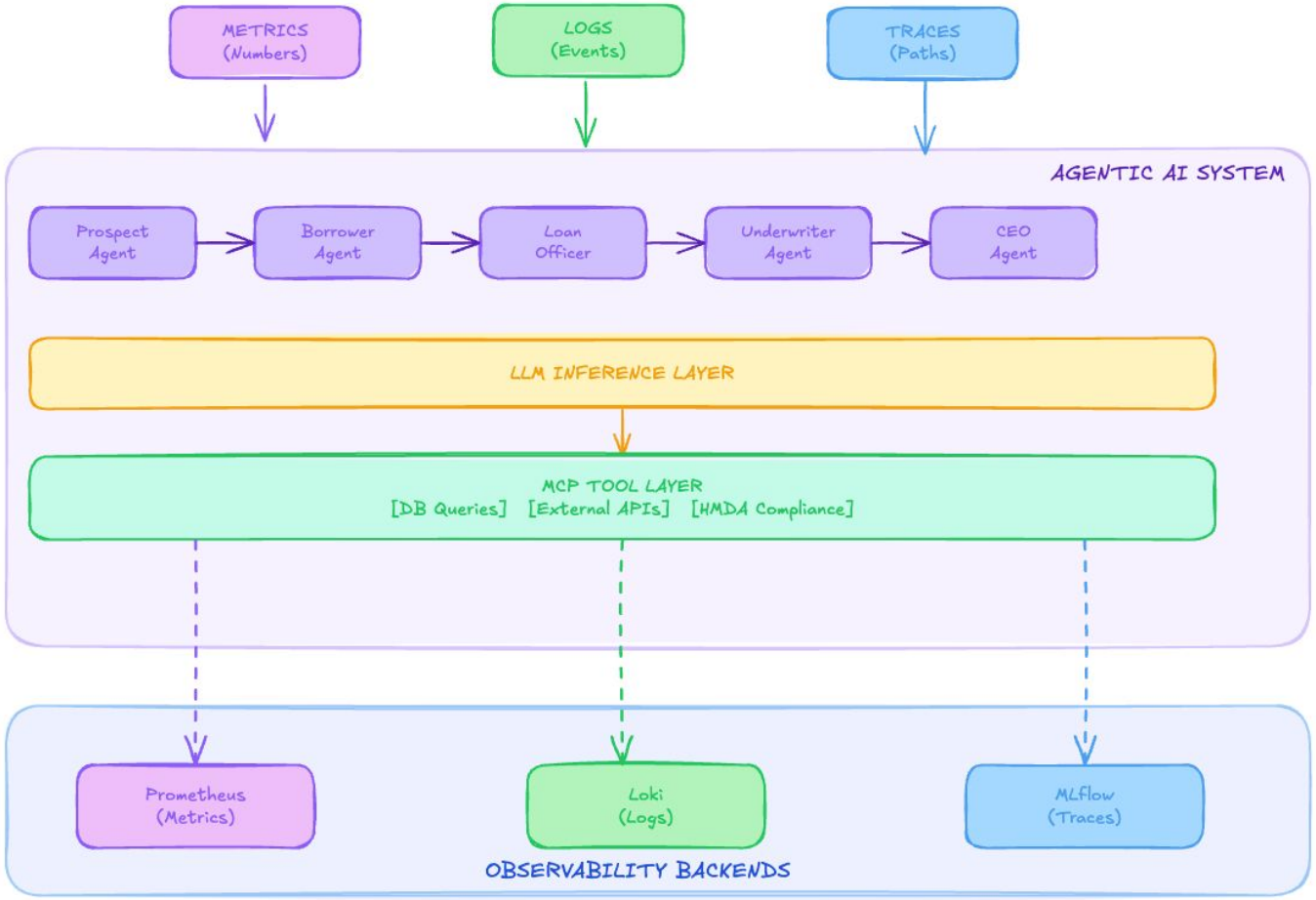
Typical example of enterprise tasks



Red Hat

What type of observability do we need to have?

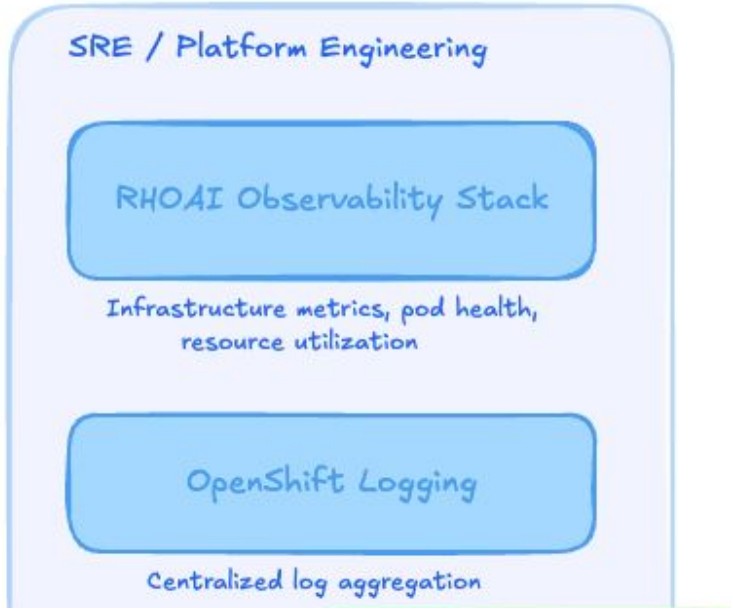
For measuring token consumption, model reasoning, etc



Red Hat

What each team needs to actually care about

SRE's & Platform Engineers



System health, SLO's, capacity, etc.

- "P99 latency for the **x** agent exceeded SLO"
- "Kubernetes pods for the MCP server keep restarting"
- "Agent tracking server is running out of disk space"

Red Hat

What each team needs to actually care about

AI Engineers & Developers



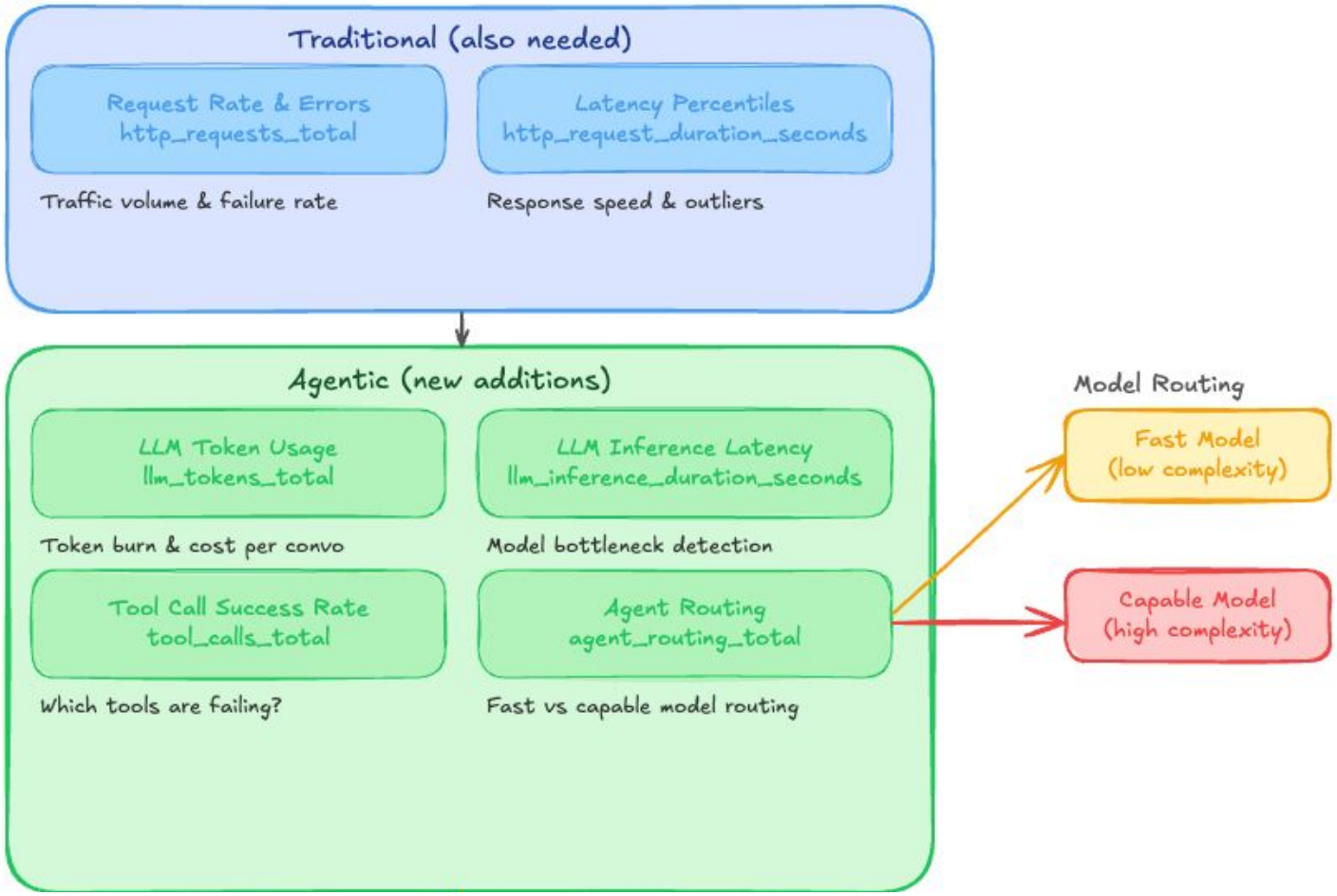
Model behavior, decision quality, agent performance

- "Loan approval rate dropped 15% after a model update"
- "Agent is providing inconsistent advice to similar inquiries"
- "Compliance check agent is flagging too many false positives"

Red Hat

Monitoring the metrics that matter

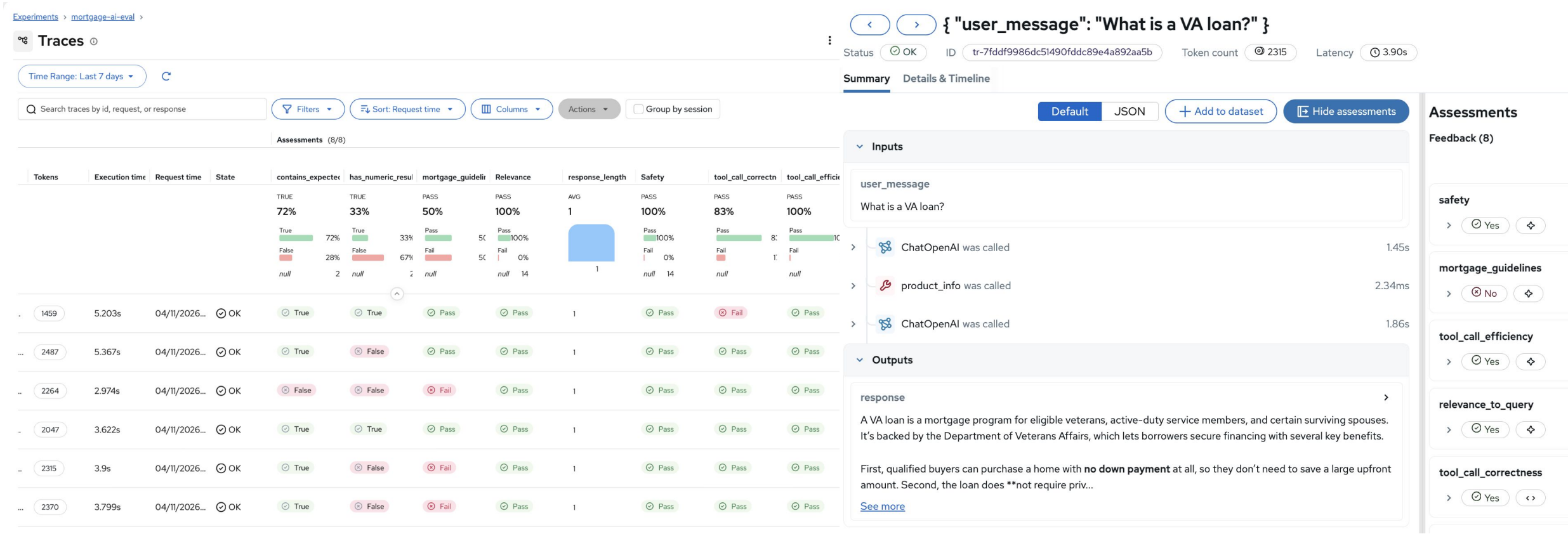
For both traditional & (now) agentic workloads



Red Hat

Automated evaluations?!

Using LLM-as-a-Judge & capabilities to go from vibes to real data



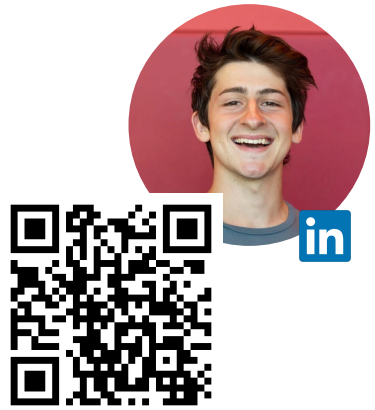
Red Hat

Thank you! You're awesome!

- The **3 pillars of agentic observability**
- Turning metrics into **evaluations**
- **Demo**: 5-Agent Enterprise Setup

Session Slides

red.ht/observing-agents



Feel free to connect on LinkedIn!

Red Hat

Thank you

Join the DevNation

Red Hat Developer serves the builders. The problem solvers who create careers with code. Let's keep in touch!

- Join Red Hat Developer at **developers.redhat.com/register**
- Follow us on any of our social channels
- Visit **dn.dev/upcoming** for a schedule of our upcoming events

Red Hat Developer

Build here. Go anywhere.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat