ADR: An Agentic Detection and Response System for Enterprise Agentic AI Security

How Uber secures enterprise AI agents at scale

Pan Hu (MLSys 2026)

Collaboration with Chenning Li, Justin Xu, Baris Ozbas, Olivia Liu, Caroline Van, Manxue Li
Wei Zhou, Mohammad Alizadeh, Pengyu Zhang, KK Sriramadhesikan, Ming Zhang

Uber Technologies

MIT

University of Oxford

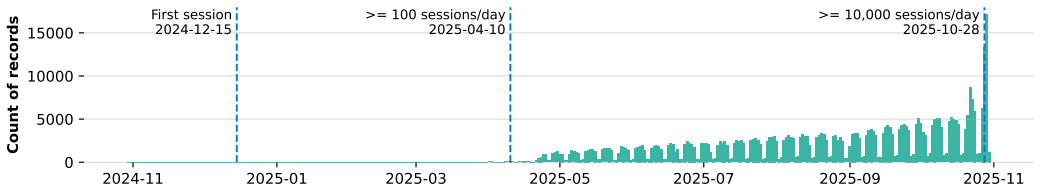
AI agents changed enterprise workflows at Uber scale

2025: Year of the agent

More than 10,000 sessions per day by October 2025.

2026: Exponential growth

200,000+ sessions per day. *“Uber Burns Its 2026 AI Budget In Four Months On Claude Code”*



AI agents now touch every stage of the software development lifecycle

Ideation
Research, arch docs

→

Design
Plan, scaffold

→

Develop
Code, refactor

→

Test
Generate, iterate

→

Review
Self/peer review

→

Deploy
CI/CD, debug

Agent security gates enterprise adoption

Common security concerns regarding AI agents

Category	Risk	Example
Secret exfiltration	Credential compromise	Agent reads `.env` or K8s tokens, sends credentials via HTTP or MCP tool
Destructive commands	Production outage, data loss	`kubectl delete ns production`; `DROP TABLE` on production database
Data exfiltration	L1/L2 data exposure	Source code or trip data pushed to public GitHub gist or personal cloud
Excessive agency	Unexpected actions	Agent deletes files out of scope during cleanup; explores devpods via SSH
Hallucination	Unintended damage	User asks to delete folder A, agent also deletes folder A_B
Supply chain	Compromised tooling	Skills or packages that collect unnecessary information from dev environments

These risks led Uber to block OpenClaw and require security review before onboarding any new AI agent.

Three gaps between agent risks and production defense

Existing security tools were not designed for **enterprise** agentic workflows.

1. Observability

EDR sees file writes and network calls, but not the agent reasoning, prompts, or causal chains linking intent to tool execution across heterogeneous AI agents.

2. Evaluation

Existing benchmarks are not MCP-native, lack enterprise policy context, or cover only a narrow slice of the threat space. No realistic, enterprise-facing eval existed.

3. Detection at scale

LLM-based reasoning on every session is too expensive. A detection system must balance precision, recall, cost, and latency at 200k+ sessions per day.

30k
endpoints
macOS + Linux devpods

302
bench tasks
42 malicious, 260 benign

200k+
sessions/day
2026 production scale

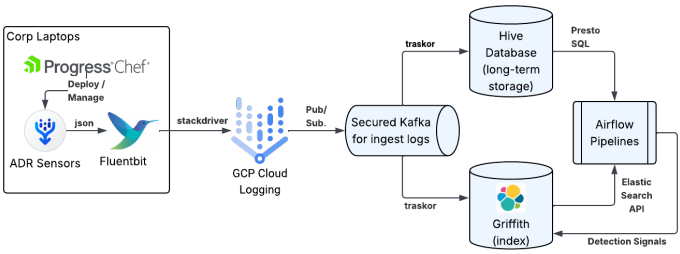
ADR addresses all three gaps: causal telemetry across agents, enterprise-shaped evaluation, and cost-aware detection at production scale.

Challenge 1: observability across fast-moving agents

Supported AI coding tools

New tools kept arriving, with different local formats and agent implementations.

- Cursor
- Claude
- OpenAI Codex
- Gemini
- Warp
- Cline



Tool coverage

Capture activity across IDE agents, CLIs, and assistants without forcing one standardized agent stack.

Causal chain

Record prompts, reasoning steps, MCP tool calls, outputs, commands, and runtime context.

Security context

Capture skills, MCP server configurations, and project files (*.md) that shape agent behavior and permissions.

ADR Sensor closes the observability gap by reconstructing the causal chain of AI-driven activity.

Challenge 2: evaluating against enterprise reality

Why existing benchmarks were not enough

Realism: Non-realistic, often synthesized tasks

Coverage: Poor coverage of attack techniques

Enterprise: Different distribution from enterprise use cases

What ADR-Bench adds

The first enterprise agentic AI benchmark, derived from production experience: MCP tools, benign-heavy traffic, policy context, and full attack-framework coverage.

302
tasks
42 malicious
260 benign

133
MCP servers
729 tools

17/17
techniques
across 5 tactics

Benchmark	MCP	Servers	Techniques
AgentDojo	No	-	4/17
RAS-Eval	Yes	18	3/17
MCP-Artifact	Yes	11	3/17
ADR-Bench	**Yes**	**133**	**17/17**

Realism

Tasks derived from SOC insights with authentic multi-step tool chains.

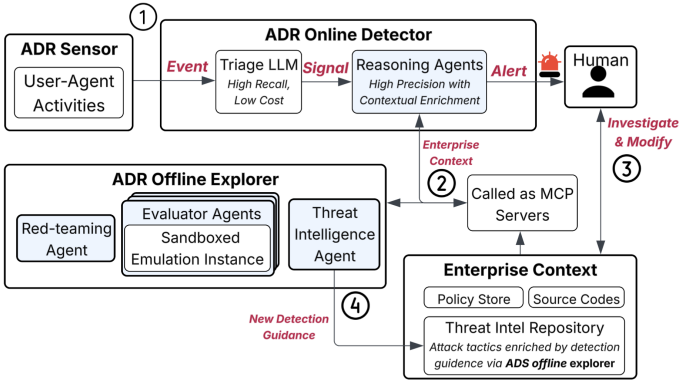
Full attack coverage

All 17 techniques across 5 tactics, beyond narrow prior threat models.

Enterprise conditions

Benign-heavy class imbalance (13.9% attack prevalence) with sensitive data.

Challenge 3: detection system that works at scale



Tier 1: triage

Cheap high-recall routing; benign sessions stop here.

Tier 2: reason

Agentic reasoning with actions to access additional context.

Explorer: harden

Offline red team finds hard variants before deployment.

2–4× F1 score over baselines on multiple benchmarks (see paper for details). In production at 200k+ sessions/day, routing scarce reasoning budget where it matters.

Findings from deployment: top concerns

Human accountability

Many people consistently use YOLO mode without sandbox. Others approve 50+ actions per session. Approval fatigue means no real oversight.

Secret exfiltration

Most common issue. Hundreds of high severity secret exfil across 26 categories. Long-lived credentials shared with AI vendors, LLM providers, and MCP servers.

Destructive commands

Agents run privileged commands on user identity. Built-in LLM guards cover `rm -rf` but miss internal tools and Uber-specific context.

Data exfiltration

Agents write software that streams data out, bypassing DLP. Unintentional leaks are common: agents cannot tell sensitive from non-sensitive.

> **Open direction: all four concerns call for real-time guardrails that intervene before the agent executes.**

Findings from deployment: common misconceptions

Prompt injection

Surprisingly rare in production. Foundation model companies have invested heavily here. External-facing agents can be tricked, but actions are easy to track.

Action: No action needed currently.

Side hustles

A significant amount of sessions are related to personal usage. Usually not a concern unless it involves enterprise data, code, or significant compute cost.

Action: Monitor, do not block by default.

Supply chain attacks

Not rare - quickly picking up. Beyond the Shai-Hulud campaign and Axios npm compromise, we found skills that collect unnecessary information from developer environments.

Action: Internal package registry with delay; scanning of skills.

Findings from deployment: emerging challenges

Excessive / insufficient agency

A simple ask can spiral into complex actions. Agent fails to auth due to expired certificate – the right action is to ask the user to re-authenticate. Instead, the agent dumps credentials everywhere, triggers EDR alerts, finds SSH keys and explores devpods.

On the other side, users complain about insufficient agency, especially with Opus 4.7.

Action: Personalization to align agents with individual user preferences.

Hallucination

Natural language vagueness causes real damage. User asks to delete folder A and subfolders, agent also deletes folder A_B. User asks for cleanup, agent deletes all files.

Action: Sandbox for containment, guardrails for scope verification, and improved LLM reasoning for intent disambiguation.

These are open problems. We invite the research community and agent vendors to collaborate on solutions.

Thank You

Questions?

We're hiring

Full-time positions at all levels on the Uber AI Security team.

Looking for partners

We seek vendor partners to address these agentic security challenges together.

Thanks to our collaborators: Chenning Li, Justin Xu, Baris Ozbas, Olivia Liu, Caroline Van, Manxue Li, Wei Zhou, Mohammad Alizadeh, Pengyu Zhang, KK Sriramadhesikan, Ming Zhang

Created with OpenAI Codex + Claude Code + Latex Beamer