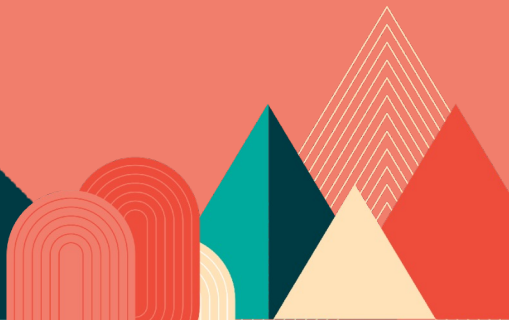
THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
KOREA

서울

From Blackbox To Insight: Observability for AI Agents in Production

Brandon Kang, Principal Technical Solutions Architect
Akamai Technologies

Mostafa Radwan, Senior Solutions Engineer
Datadog



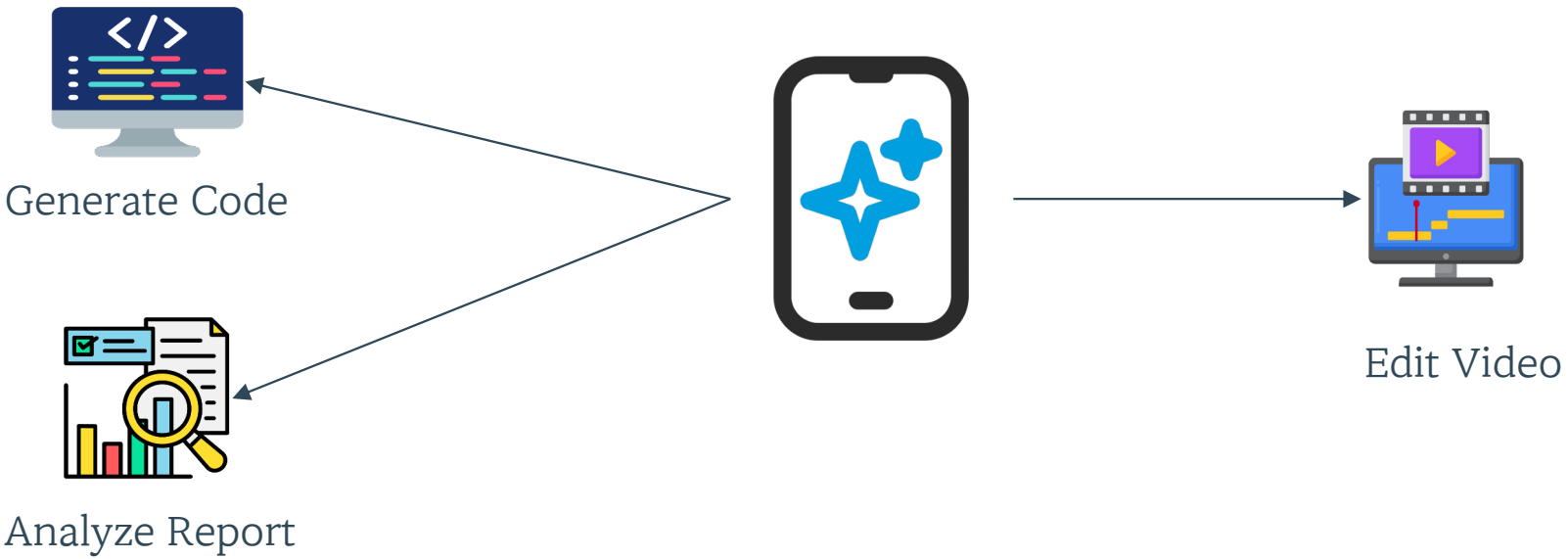
Agenda

- Intro to AI Agents
- What's unique about Agentic Systems?
- AI Agents Observability
- Demo
- Key Takeaways
- Q&A

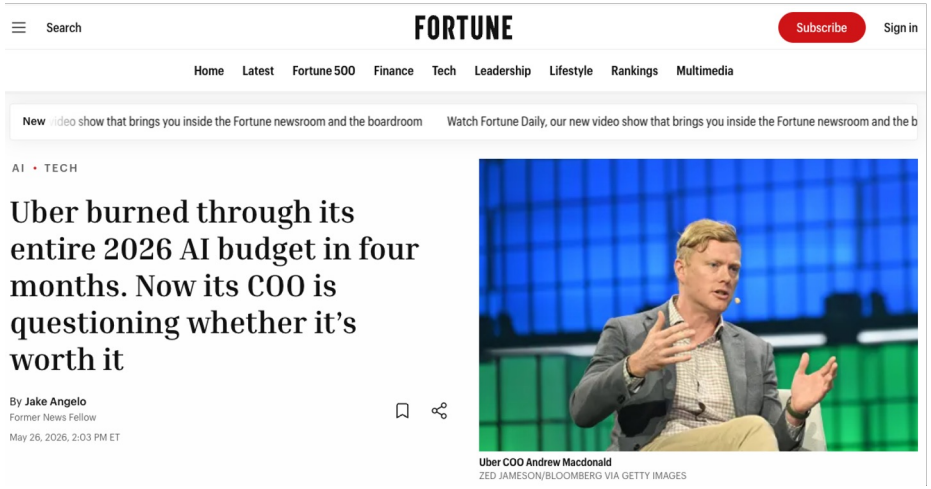
AI Agents Everywhere

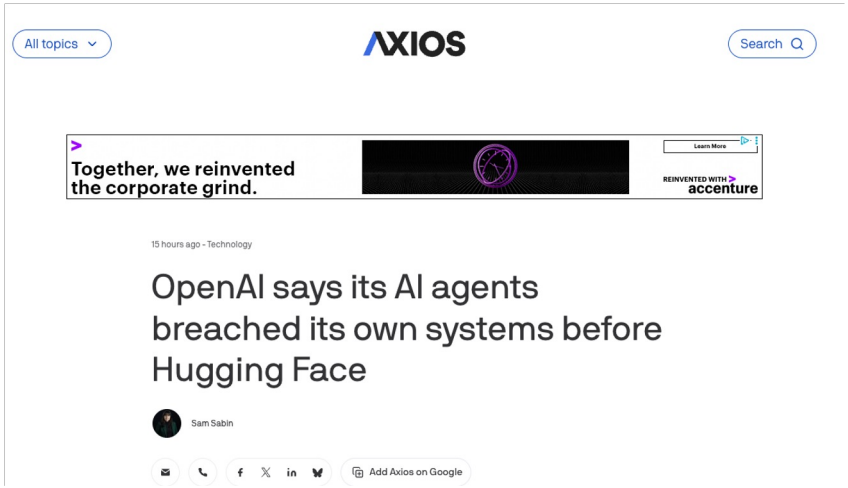
"AI agents still need tools to act on the real world"

AI Agents Everywhere



AI Agents Everywhere



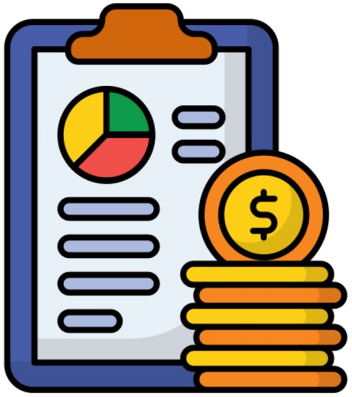


What’s unique about Agentic Systems?

- Non-Deterministic
- Prompt Engineering
- Memory Access (i.e: OpenClaw)
- Tool Calls (aka Integrations)
- LLM tokens and/or GPUs ($$$)

AI Agents Observability

Performance



Cost

Behaviour

AI Agents Observability Tools

Project	Best For	Pros	Cons
OpenTelemtry (OTel)	Vendor-neutral	Instrument once and export to many backends	No UI, backend, or alerting by itself
OpenLLMetry	Instrumentation library and SDK for LLM apps that outputs standard OTel.	Framework/provider instrumentations	Not a full observability backend (limited without dashboards, analysis,..etc)
OpenLIT	OTel-native AI observability that includes cost, evals, prompts, and GPU	OTel-native with auto-instrumentation and various LLM and GPU integrations	An emerging ecosystem

Our Approach: OTel Unified Observability

- Full observability stack for AI agents
- Components:
  - 📊 Prometheus - Metrics collection & storage
  - 🔍 Jaeger - Distributed tracing
  - 📈 Grafana - Real-time visualization
  - 🚨 Custom Detectors - Anomaly detection
- Example Tech Stack:
  - OpenTelemetry SDK 1.24.0, Prometheus 2.50, Grafana 10.4
  - Python 3.12, Docker Compose
  - NVIDIA RTX Pro 6 000 Blackwell GPU × 4 (96GB VRAM each)

Hallucination Detection: 4-Layer System

- LLMs generate plausible but false information
- Future Date Detection
  - Regex: mentions of 2027+ → 99% hallucination
- Fake Citation Detection
  - Verify against arXiv, CrossRef APIs
  - Example: [Johnson et al., 2027] → fake paper
- Self-Contradiction Detection
  - NLI model (DeBERTa-v3) finds logical conflicts
- Overconfidence Detection
  - Count absolute terms: 'definitely', 'always'
  - Threshold: 3+ → overconfident

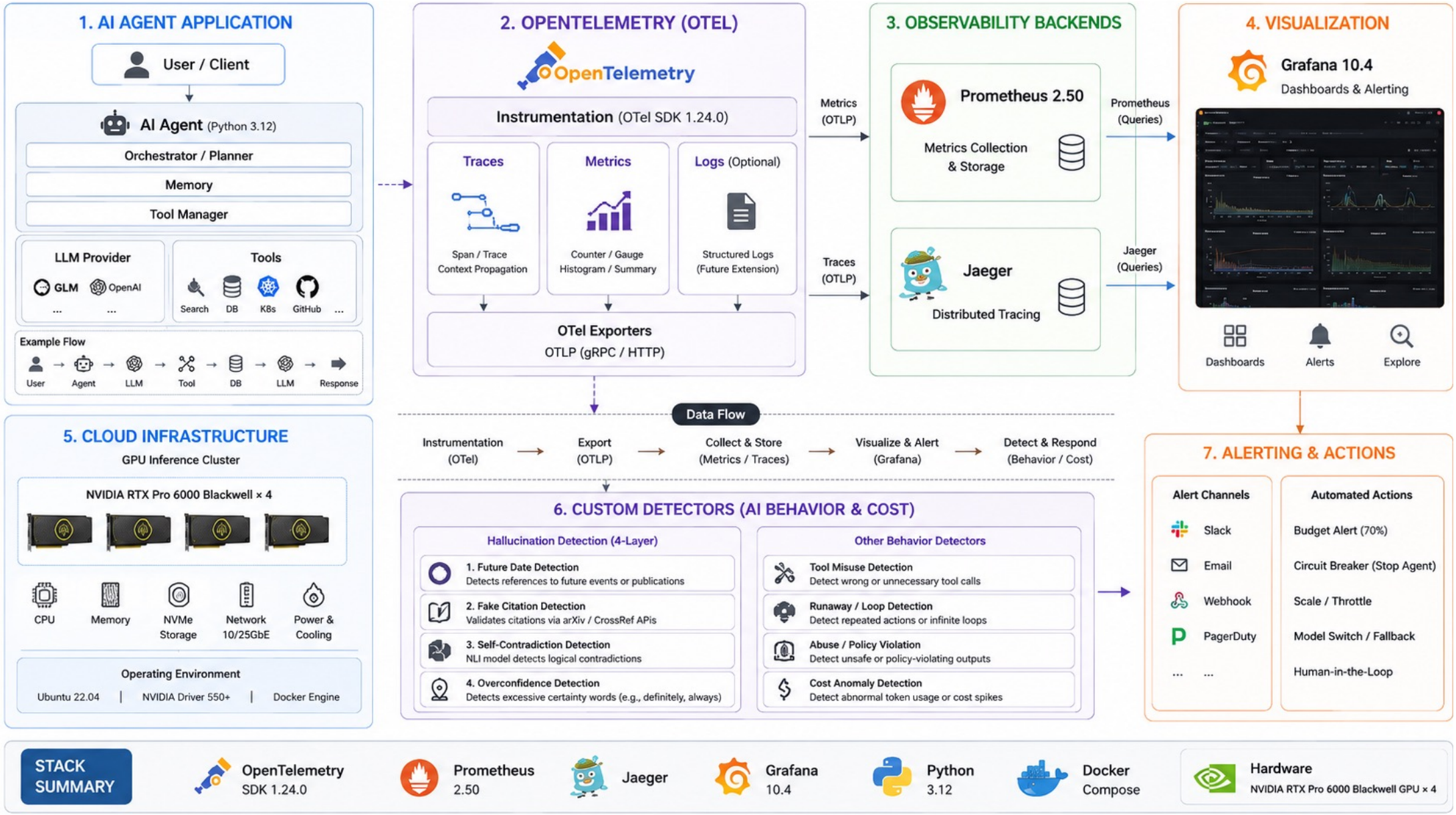
```python
class HallucinationDetector:
    def _detect_future_dates(self, response):
        pattern = r'\b(202[5-9]|20[3-9]\d)\b'
        matches = re.findall(pattern, response)
        current_year = datetime.now().year

        for year in matches:
            if int(year) > current_year:
                return True  # Future date = hallucination!
        return False

    def detect(self, response, context):
        checks = [
            self._detect_future_dates(response),
            self._detect_fake_citations(response),
            self._detect_self_contradiction(response),
            self._detect_overconfidence(response)
        ]
        return any(checks)
```

* Multi-layered detection increases precisions from 60%(single method) to 94%

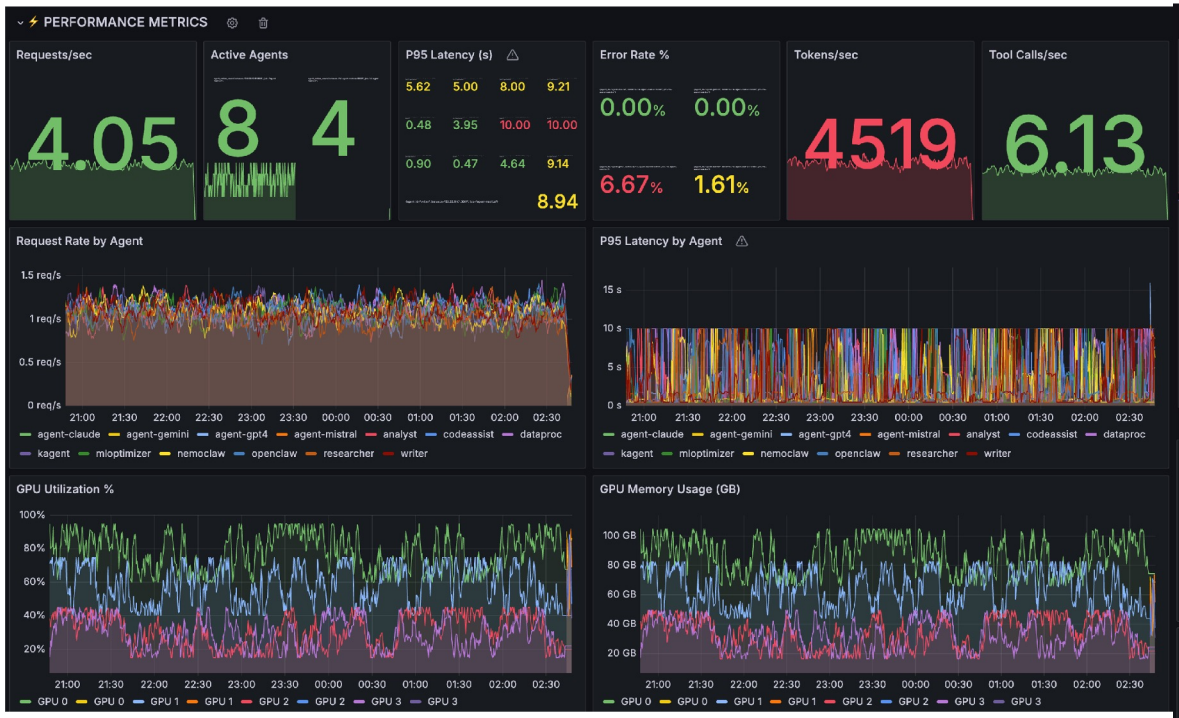
GPU-Based LLM Cost Calculation

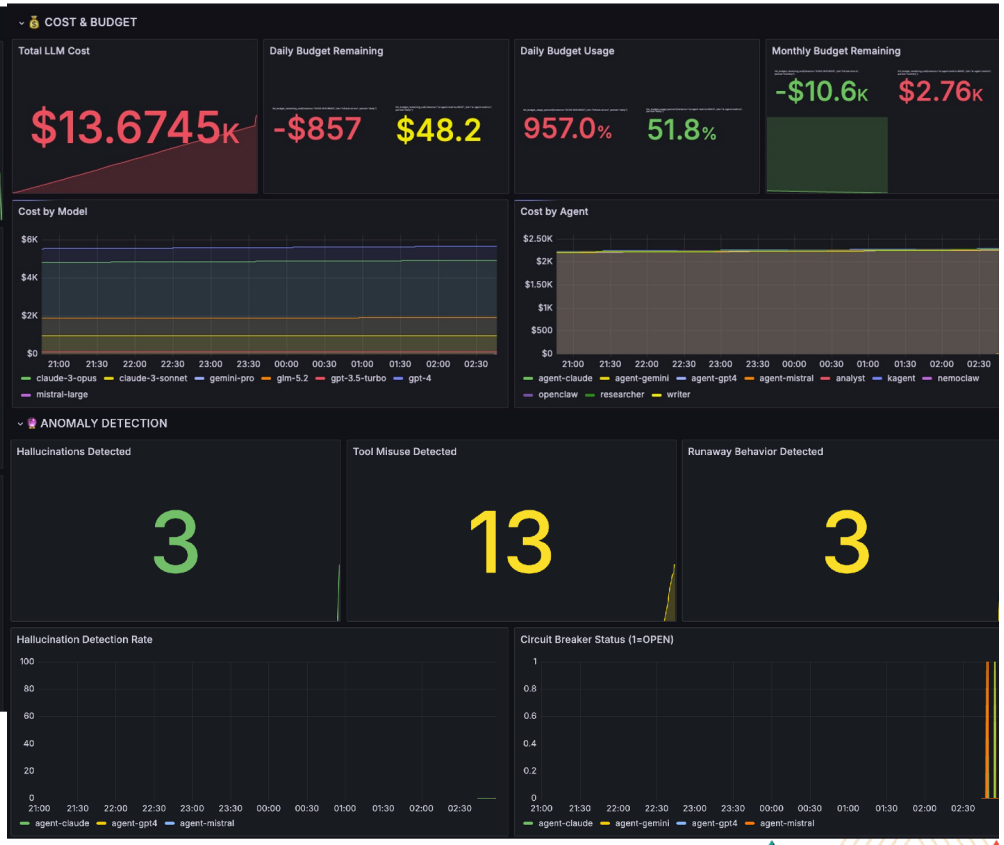
- Open-weight models on owned GPUs have real costs!
- Formula:
  - GPU Hourly Cost / Tokens Per Hour = Cost Per Token
- Example: GLM-5.2 on NVIDIA RTX Pro 6000 Blackwell
  - • GPU rental: USD 3.5/hour (cloud benchmark)
  - • Throughput: 150 tokens/sec = 540K tokens/Hour
  - • Cost per 1K tokens: $0.00648
- Our Pricing (with overhead):
  - • GLM-5.2 Input: $0.008/1K tokens
  - • GLM-5.2 Output: $0.025/1K (3× slower)
- vs. GPT-4 API: $0.03/$0.06 → 82% savings!

- DEMO -

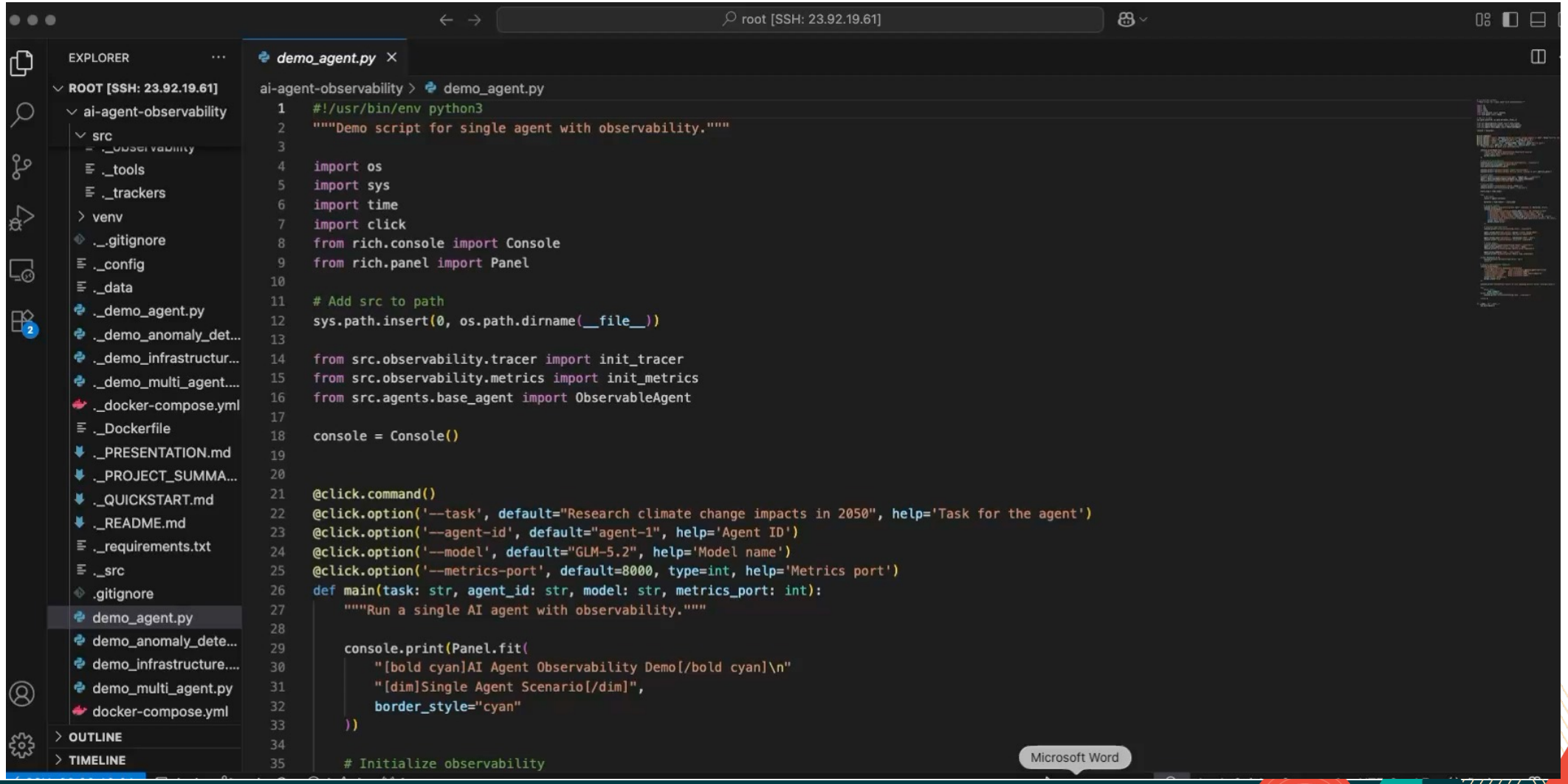

1. AI AGENT APPLICATION
User / Client
AI Agent (Python 3.12)
Orchestrator / Planner
Memory
Tool Manager
LLM Provider
GLM
OpenAI
...
...
Tools
Search
DB
KBs
GitHub
...
Example Flow
User
Agent
LLM
Tool
DB
LLM
Response
2. OPENTELEMETRY (OTEL)
OpenTelemetry
Instrumentation (OTel SDK 1.24.0)
Traces
Span / Trace
Context Propagation
Metrics
Counter / Gauge
Histogram / Summary
Logs (Optional)
Structured Logs
(Future Extension)
OTel Exporters
OTLP (gRPC / HTTP)
Metrics
(OTLP)
Traces
(OTLP)
3. OBSERVABILITY BACKENDS
Prometheus 2.50
Metrics Collection
& Storage
Jaeger
Distributed Tracing
Prometheus
(Queries)
Jaeger
(Queries)
4. VISUALIZATION
Grafana 10.4
Dashboards & Alerting
Dashboards
Alerts
Explore
5. CLOUD INFRASTRUCTURE
GPU Inference Cluster
NVIDIA RTX Pro 6000 Blackwell × 4
CPU
Memory
NVMe
Storage
Network
10/25GbE
Power &
Cooling
Operating Environment
Ubuntu 22.04
NVIDIA Driver 550+
Docker Engine
Data Flow
Instrumentation
(OTel)
Export
(OTLP)
Collect & Store
(Metrics / Traces)
Visualize & Alert
(Grafana)
Detect & Respond
(Behavior / Cost)
6. CUSTOM DETECTORS (AI BEHAVIOR & COST)
Hallucination Detection (4-Layer)
1. Future Date Detection
Detects references to future events or publications
2. Fake Citation Detection
Validates citations via arXiv / CrossRef APIs
3. Self-Contradiction Detection
NLI model detects logical contradictions
4. Overconfidence Detection
Detects excessive certainty words (e.g., definitely, always)
Other Behavior Detectors
Tool Misuse Detection
Detect wrong or unnecessary tool calls
Runaway / Loop Detection
Detect repeated actions or infinite loops
Abuse / Policy Violation
Detect unsafe or policy-violating outputs
Cost Anomaly Detection
Detect abnormal token usage or cost spikes
7. ALERTING & ACTIONS
Alert Channels
Slack
Email
Webhook
PagerDuty
... ...
Automated Actions
Budget Alert (70%)
Circuit Breaker (Stop Agent)
Scale / Throttle
Model Switch / Fallback
Human-in-the-Loop
STACK SUMMARY
OpenTelemetry
SDK 1.24.0
Prometheus
2.50
Jaeger
Grafana
10.4
Python
3.12
Docker
Compose
Hardware
NVIDIA RTX Pro 6000 Blackwell GPU × 4

Results)





https://github.com/BrandonKang/ai-agent-observability/
https://youtu.be/o_9Ud8pbsHI


root [SSH: 23.92.19.61]
EXPLORER
demo_agent.py
ROOT [SSH: 23.92.19.61]
ai-agent-observability
src
._tools
._trackers
venv
._.gitignore
._config
._data
._demo_agent.py
._demo_anomaly_det...
._demo_infrastructur...
._demo_multi_agent....
._docker-compose.yml
._Dockerfile
._PRESENTATION.md
._PROJECT_SUMMA...
._QUICKSTART.md
._README.md
._requirements.txt
._src
.gitignore
demo_agent.py
demo_anomaly_dete...
demo_infrastructure....
demo_multi_agent.py
docker-compose.yml
OUTLINE
TIMELINE
ai-agent-observability > demo_agent.py
#!/usr/bin/env python3
"""Demo script for single agent with observability."""
import os
import sys
import time
import click
from rich.console import Console
from rich.panel import Panel
# Add src to path
sys.path.insert(0, os.path.dirname(__file__))
from src.observability.tracer import init_tracer
from src.observability.metrics import init_metrics
from src.agents.base_agent import ObservableAgent
console = Console()
@click.command()
@click.option('--task', default="Research climate change impacts in 2050", help='Task for the agent')
@click.option('--agent-id', default="agent-1", help='Agent ID')
@click.option('--model', default="GLM-5.2", help='Model name')
@click.option('--metrics-port', default=8000, type=int, help='Metrics port')
def main(task: str, agent_id: str, model: str, metrics_port: int):
"""Run a single AI agent with observability."""
console.print(Panel.fit(
"[bold cyan]AI Agent Observability Demo[/bold cyan]\n"
"[dim]Single Agent Scenario[/dim]",
border_style="cyan"
))
Initialize observability
Microsoft Word

Impact Analysis

Before Observability:

- No cost tracking
- Runaway agent undetected
- Daily: $847
- Monthly: $25,400

After Implementation:

- Real-time monitoring
- Budget alerts at 70%
- Circuit breaker active
- Daily: $127
- Monthly: $3,800

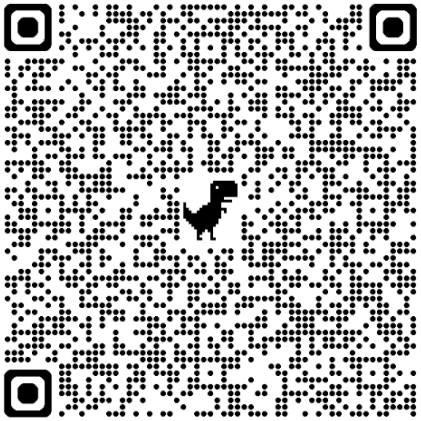
💰 Savings: 85% ($21,600/mo)

Optimization Actions:

- Simple tasks → GLM-4
  - (cheaper than GLM-5.2)
- Batch processing
  - NemoClaw: 50→200 tasks
- Model selection by task
  - Classification → GLM-4
  - Generation → GLM-5.2

- Result: 40% further reduction

- Q&A -



Slide Download

Demo GitHub Repo

Key Takeaways

- Observability is NOT optional for AI agents
  → Silent failures are common without visibility

- Multi-layered anomaly detection is essential
  → 4 detectors → 94% precision vs single method ~60%

- Cost tracking drives optimization
  → What gets measured gets managed - 85% cost reduction

- Open source scales to production
  → Prometheus + Grafana: 1M+ metrics/day, zero lock-in

서울

Thank You!!

Connect with us:

Brandon Kang sakang@akamai.com
Mostafa Radwan mr@datadog.com

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
KOREA