

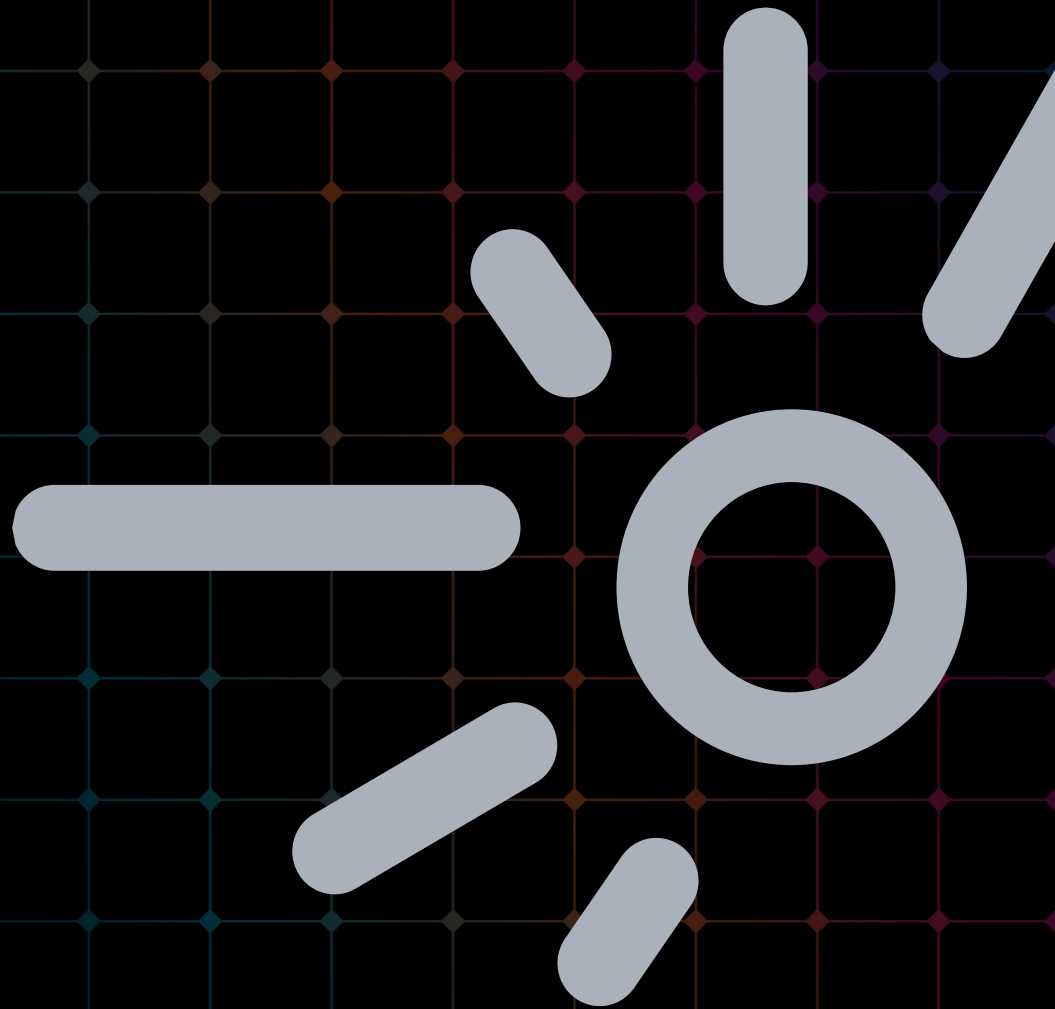


From Opaque to Observable: Tracing Multi-Agent Openclaw Workflows with OTel

Insight Module for Agentic Systems

Pavan Sudheendra
Technical Leader, Cisco Systems

AGNTCon, Europe
17-09-2026



Pavan Sudheendra

Technical Leader, Outshift by Cisco.

I lead technical efforts on distributed systems and emerging multi-agent architectures.

How I ended up here

Active contributor to the LF ecosystem, including AGNTCY.org initiatives, and contributes to OTel GenAI standardization efforts. Chair of the AAIF Observability & Traceability WG

<https://github.com/outshift-open/insight-module-for-agentic-systems>

Apache-2.0



Scan for the repo



What's the current situation

Problem

OpenClaw's built-in observability mainly covers local runtime diagnostics (agent logs, tool calls, session state, basic traces) and is not designed for cross-agent, multi-instance, or lifecycle-level analysis.

Status

OpenClaw now ships a built-in OpenTelemetry exporter (*diagnostics-otel*) that emits basic traces, logs and metrics out of the box: e.g. llm/tool calls, cost, duration, gateway logs.

Gaps

Openclaw metric names do not map to Otel GenAI semantic conventions :
lack of Otel semantic convention compliant telemetry

OpenClaw's built-in telemetry is good for platform health and message processing, but thin for "agent forensics" — agent-to-agent interaction, sub-agent trees, memory changes, and end-to-end correlation across workflows. Control-flow and governance events are fragmented across hooks, logs, and diagnostics



Four blind spots exists today

■ Context assembly & sharing

- Sources: prompts, agent soul, memory, tool output
- Selection: what was included, cut or truncated
- Boundaries: what passes down to sub-agents

■ Routing & delegation

- Why this agent and not another?
- Why was Fermor skipped but Bourdain called?
- Task type, confidence, urgency — or randomness?
- Delegation errors fail silently

■ Memory lifecycle

- Why a memory was written, and by whom
- When it is retrieved
- How strongly it shaped a decision
- Stale, tentative, or authoritative?

■ Decision checkpoints

- Not private reasoning — decision points
- Alternative paths considered, high level
- Confidence levels
- Why a validation step was skipped



Insight Module for Agentic Systems



A new plugin

Today OpenClaw built-in diagnostic plugin provides basic observability support but it is found lacking in terms of comprehensive visibility and detailed diagnostics across multiple dimensions. Our plugin complements it by providing the following missing functionalities:

1 Connected request lifecycle

Inbound message → agent turn → tool calls → outbound response as one trace, instead of separate root spans.

2 Workflow session semantics

A session boundary that spans multiple turns and multiple agents, with explicit start and end spans.

3 Delegation topology

Spawn and handoff span links, fork/join annotation, and child-session lineage across runtime session keys.

4 Derived coordination metrics

Parallelisation, repetition, novelty and memory-fragmentation scores computed from the trace itself.



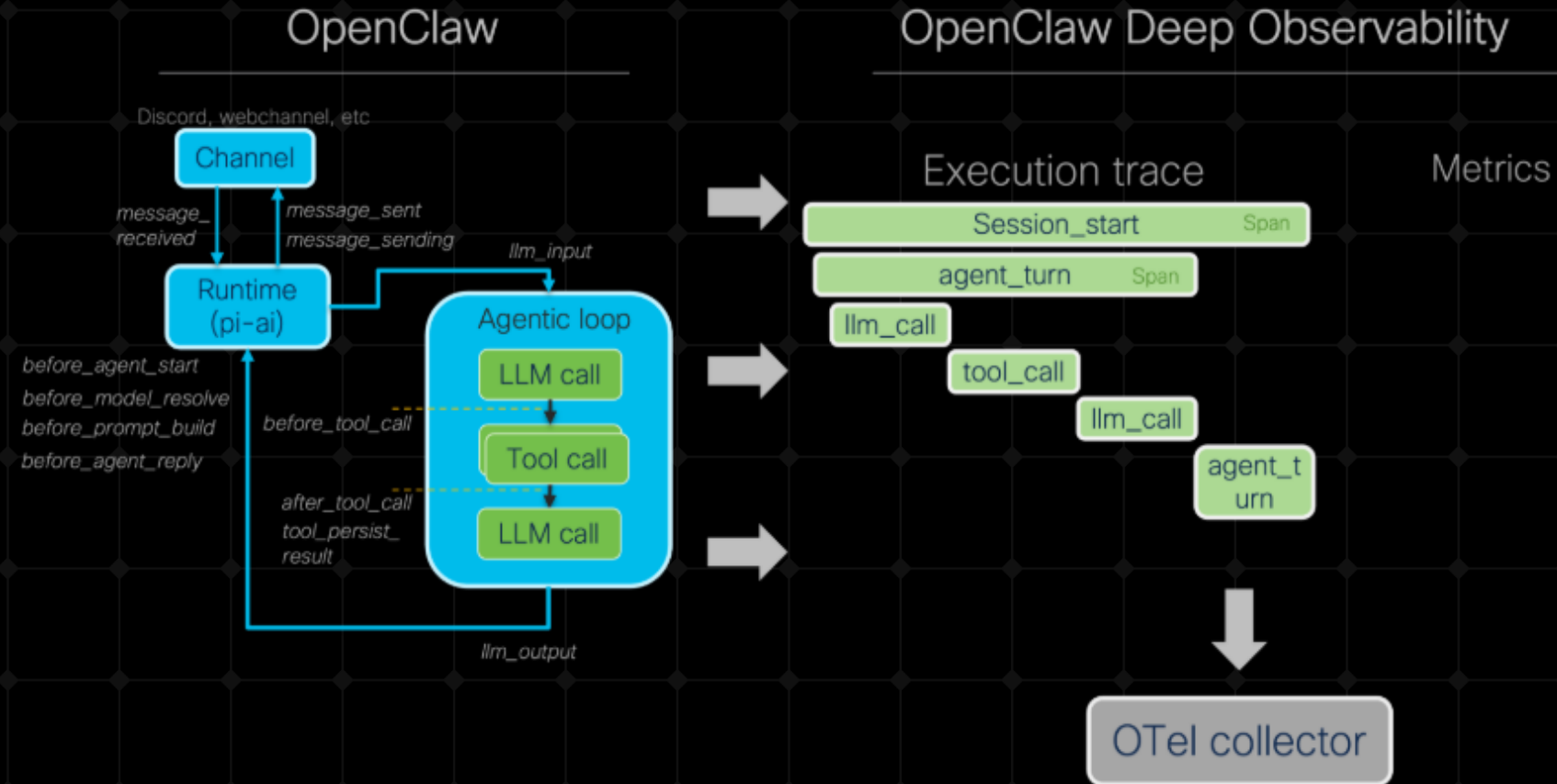
How it works

It combines three telemetry paths:

- **Typed OpenClaw lifecycle hooks** for request, agent, sub-agent, tool, and outbound message flow.
- **Event-stream hooks for control-plane events** such as session reset and gateway startup.
- **OpenClaw diagnostics events** for accurate model usage, cost, queue, webhook, and stuck-session signals.

It captures the full request journey from inbound message to agent turn, tool calls, model usage, and outbound response.

It also adds token, cost, latency, session, and gateway health visibility.



Observability Features

- Session lifecycle tracking with session start and idle-based end
- Tracing delegation topology
- detecting collaboration failure modes
- Gateway diagnostics for queue, webhook, stuck session, and tool-loop signals
- Auto-instrumentation support via OpenLLMetry
- In-plugin cache
- Derived metrics support.
- Exposing system-level coordination health



Enriching the notion of "session"

Core identifiers

openclaw.session.key

Runtime key — correlation across events, turns and tools.

session.id

Plugin-generated UUID — the lifecycle boundary.

Lifecycle

Start first activity on a runtime key — inbound request, agent start or LLM input.

End explicit reset, a 5-minute idle timeout checked every 30s, or graceful flush on shutdown.

Design philosophy: a best-effort boundary

session.id is a heuristic wrapper. It is not tied to a single request span — it persists across turns until an end condition fires. A boundary that is usually right beats no boundary at all.



Where the GenAI conventions stop

We ship on the Agntcy Observe schema, built on OTel. The GenAI semantic conventions do not yet cover agent handoff and delegation – among other things, so we align on what exists and propose the rest.

Aligning with semconv

- `gen_ai.operation.name=invoke_workflow`
on request spans
- `gen_ai.workflow.name`
on workflow spans
- `gen_ai.provider.name`
replaces `gen_ai.system`
- current cache-token field names
- OTel payload fields on workflow, agent, LLM and tool spans when `captureContent=true`

Not covered yet — our proposal

Fork / join semantics

Handoff and spawn span links

Child-session lineage

Derived coordination scores

These live under `ioa_observe.*` behind the `emitIoaObserveAttributes` flag (default true) until the conventions catch up.



Metrics and Monitoring



~46 metrics

Core

12

Request flow, tool activity, LLM usage, cost

openclaw.llm.tokens.total
openclaw.cost.usd

Gateway diagnostics

14

Queue health, webhooks, session health

openclaw.queue.depth
openclaw.session.stuck_age_ms

Memory events

8

Read, write and edit operations

openclaw.memory.search_hit
openclaw.memory.write_duration

Context assembly

8

Context composition and propagation

openclaw.context.prompt_size
openclaw.context.preparation_duration

Memory lifecycle

1

Fragmentation across searches

openclaw.memory.search_fragmentation

Routing & delegation

3

Coordination quality scores

openclaw.session.parallelisation_score
openclaw.agent.novelty_score



What monitoring enables

- **End-to-end system visibility** - unified view across agent execution, infrastructure, and control into a single observable execution trace.
- **Root-cause diagnosis of failures** - identification of bottlenecks (queueing, latency, tool errors, stuck sessions, retries)
- **Separation of semantic vs operational issues** - differentiate whether failures arise from agent reasoning/routing (e.g., tool loops, poor context) or from external constraints (e.g., queue delays, webhook failures, policy limits).
- **Evaluation of context/memory use and reconstruction of information flow** - provide insight into how memory is accessed/updated/impacts execution. Reveal how context is assembled and propagated, helping detect over/under-sharing and context composition.



Demo



Sample MAS



SRE Lead

Supervisor – decomposes tasks, routes work, decides actions



Telemetry Analyst

Queries MELT and summarizes evidence.



Backend Eng

Reasons about services, proposes rollback/fix.



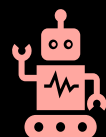
DB specialist

Diagnoses DB symptoms, proposes DB mitigations.



Comms agent

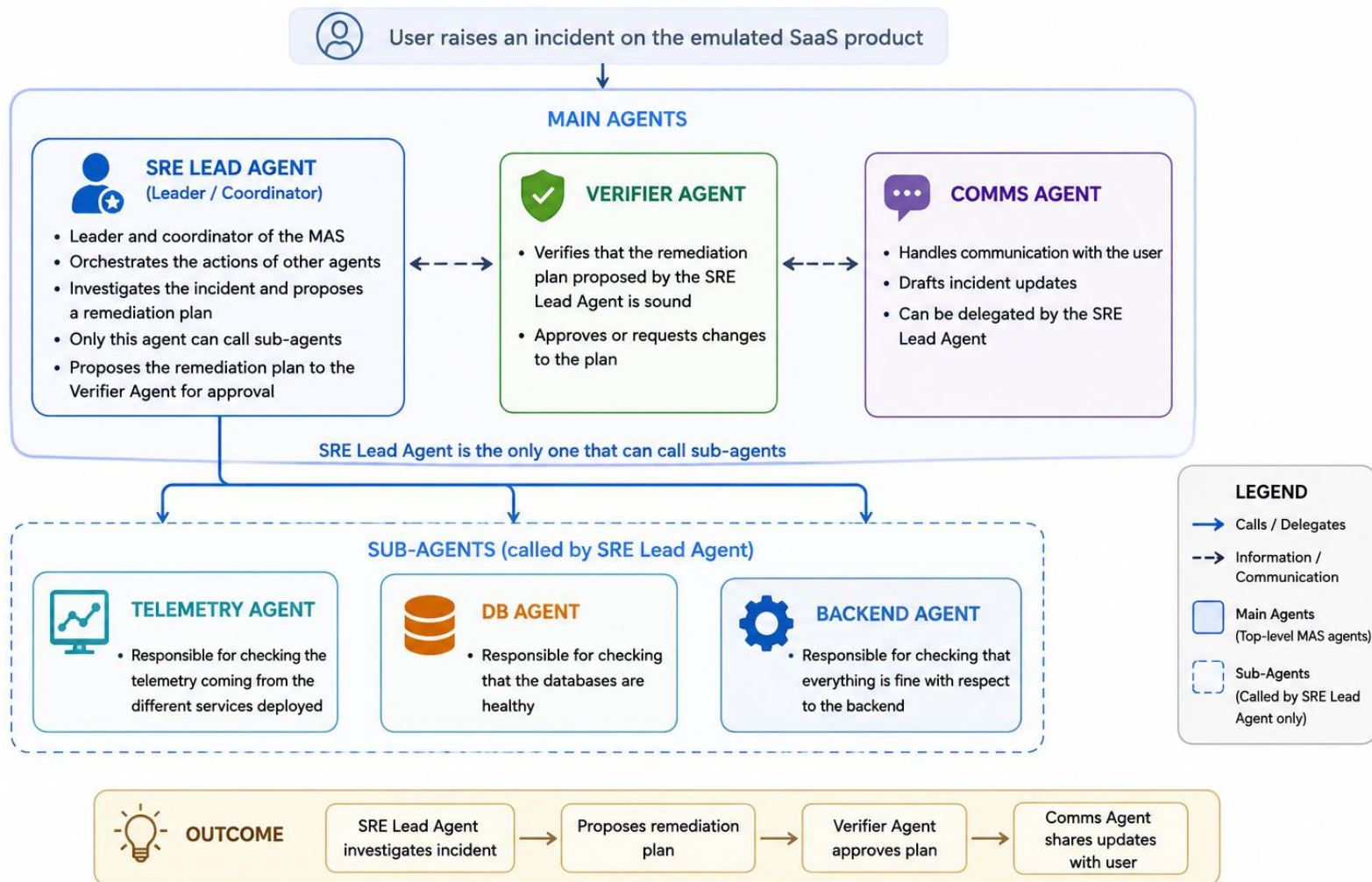
Drafts stakeholder/customer updates.



Verifier/Auditor

Checks correctness, constraints, termination criteria.

SRE Triage Multi-Agentic System (MAS)



Demo scenario 0 – Normal behavior

- We show a normal SRE Triage session and user interaction via Discord
- provides a trace waterfall for a given session
- A glimpse at the monitoring dashboard (Splunk for internal integration, Grafana equivalent for OSS)





SRE-Claw

S-C

Events

O-t-0

Salons textuels

O-t

général

nemoclaw

+

Salons vocaux

Général

général

Giulio Grassi

@SRE lead

who are you?

1

SRE lead

APP

09:38

The investigation is stalled until this specialist agent is functioning correctly.

I am the SRE lead for this workspace, acting as an incident triage orchestrator. My primary role is to receive incident reports, delegate investigation to specialist sub-agents (telemetry, backend, and db), and then synthesize their findings into a coherent root-cause analysis and a safe, verified remediation plan.

I operate exclusively through a remote API and do not have direct access to the local filesystem or machine. My goal is to restore service quickly, control risk, and ensure all actions are based on clear evidence.

Giulio Grassi

@SRE lead

which agents can you interact with?

1

SRE lead

APP

09:39

I can interact with several specialist and coordination agents to manage incidents:

- **Sub-agents for Investigation:**
 - @telemetry : I delegate to this agent to get quantified evidence like latency, error rates, and resource utilization.
 - @backend : Engaged for issues related to application logic, deployments, or runtime configurations.
 - @db : Focused on the database layer, investigating slow queries, lock contention, or connection pool problems.
- **Agents for Coordination:**
 - @verifier : Before any remediation is applied, I am required to send the plan to this agent for approval to ensure it's safe.
 - @comms : After a fix is verified, I contact this agent to prepare communication for stakeholders.

I use `sessions_spawn` to create new investigation tasks for my sub-agents and `sessions_send` to coordinate with the verifier and comms agents.

G...
On...

+

Message #général



Home



Dashboards



Alerts



APM



Infrastructure



Digital Experience



Logs



Metrics



Data Management



Settings

← APM

Trace Analyzer

🕒 -15m

Environment : unknown

Transaction : All

Services

🔍 Add filters

🔄 Refresh

Trace & error count

Duration
(ms)⬆️ ⬆️
⬆️ ⬆️Sample
Ratio

1:1

🔴

Errors
only

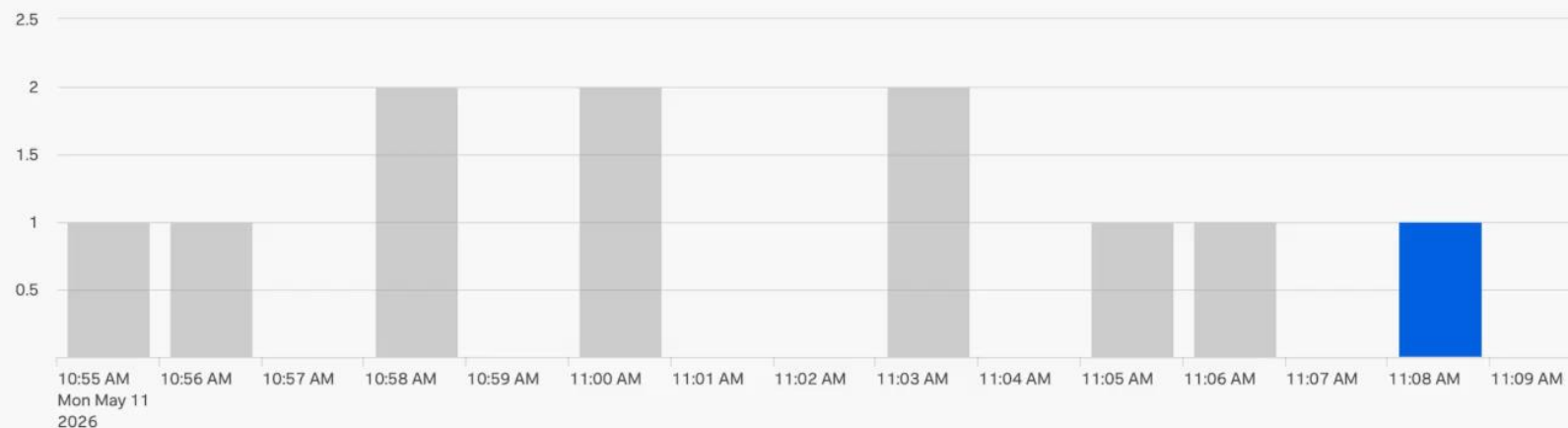
🔴

Call graphs
only

View Trace ID

Select...

Go



Traces

Group Metrics

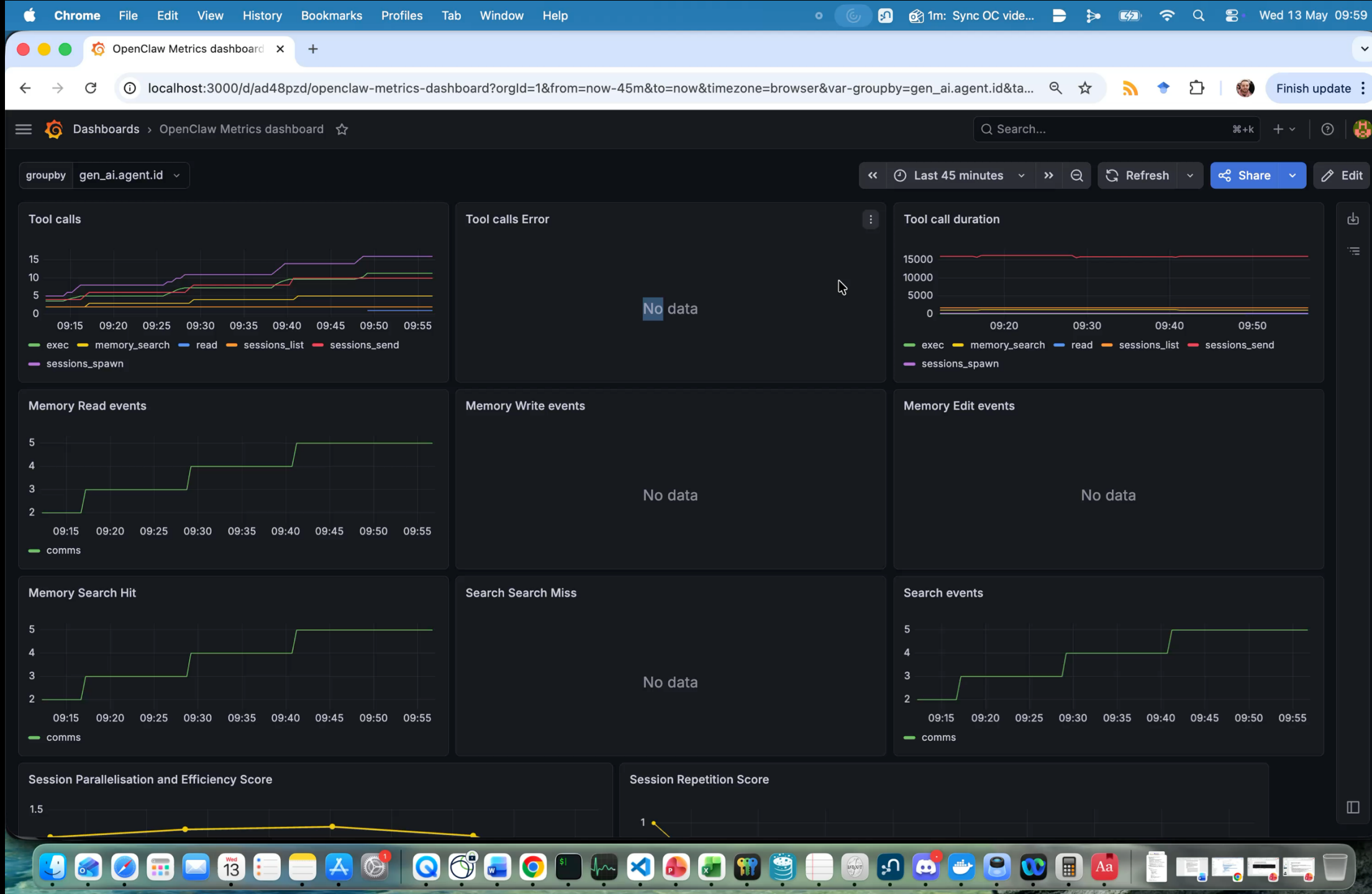
Group traces by

Select...

11 traces matched

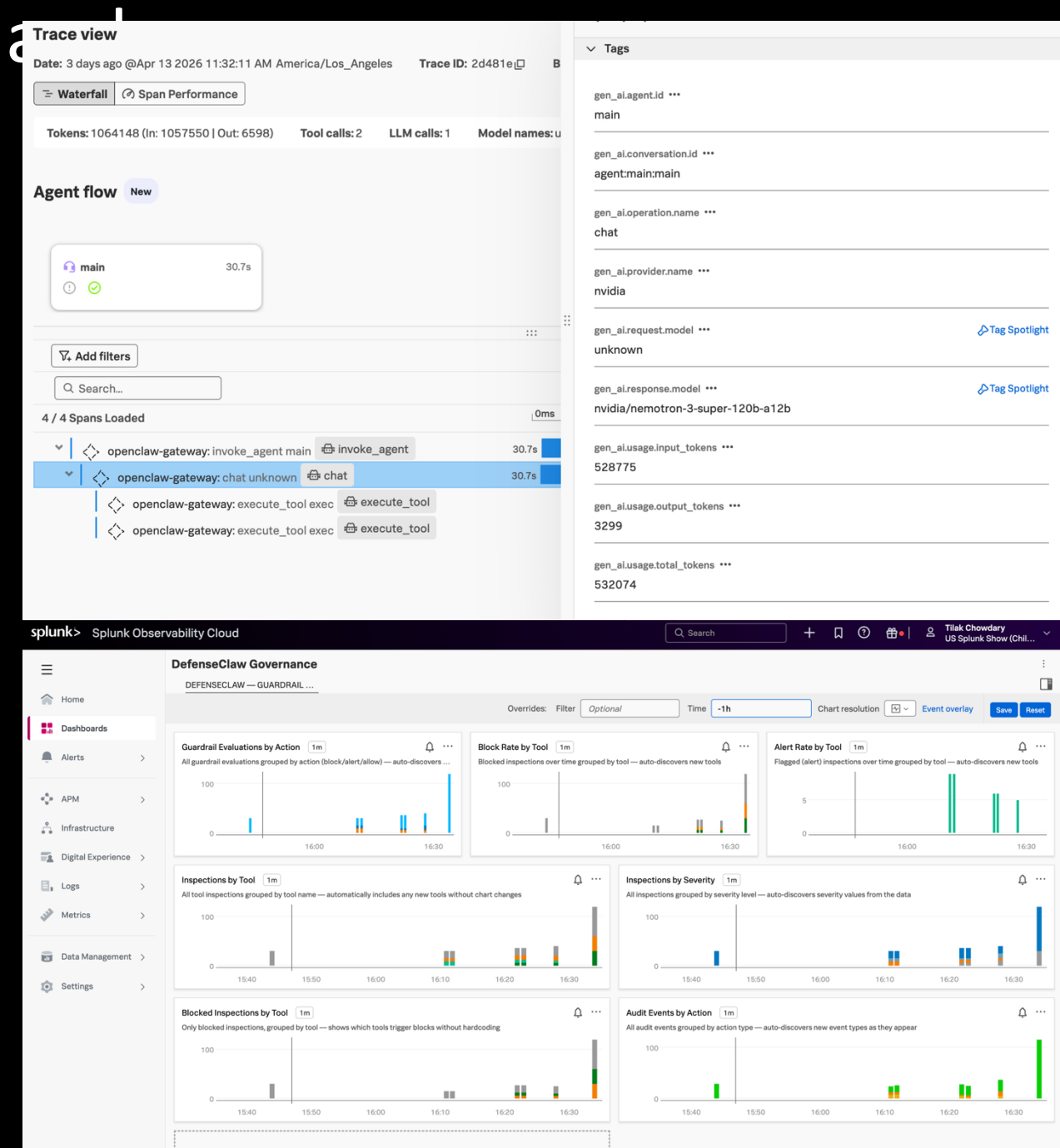
Trace ID	Timestamp	Duration	Initiating Operation	Services
05a4e0d1bc4a31927f9	Mon, May 11 2026 11:06:29 AM CEST	2.0m	openclaw-gateway: openclaw.request	openclaw-gateway (57)
9f309b8f81321916e4f	Mon, May 11 2026 11:06:15 AM CEST	29.06s	openclaw-gateway: session.start	openclaw-gateway (6)

Obs-normal01



Support for NemoClaw DefenseClaw

- Consistent Architecture: The OpenClaw plugin is natively compatible with NemoClaw and DefenseClaw.
- Shared Instrumentation
 - We can leverage the same OTLP-based telemetry pipeline across all platforms.
- Platform-Specific Benefits:
 - NemoClaw: Enhanced visibility into high-throughput agent workflows and resource utilization.
 - DefenseClaw: Audit-ready tracing for security-sensitive tool calls and decision-making logic.



Call to Action

github.com/outshift-open/insight-module-for-agentic-systems

Issues and PRs welcome.

Try it

Point it at your own OpenClaw deployment.

Tell us what broke

Especially session boundaries under long pauses.

Help with extensions

To harnesses or other types of MAS



Pavan Sudheendra · outshift.com · [@outshiftbycisco](https://twitter.com/outshiftbycisco)



outshift.com ↗

@ outshiftbycisco