Activity and conversations

The audit trail, filters, risk findings, prompt capture.

- Overview
- Read the activity feed
- Read a conversation
- Filters and the four tiles
- The Signals column
- The other columns

Overview

Activity & conversations

Activity is the audit trail: one row per AI request, recorded by the Gateway after the response streams back.

What's recorded

Model, tool (Claude Code / Cursor / …), developer identity, device, connection mode (subscription pass-through vs managed key), outcome (ok / blocked / upstream error), token counts, latency, MCP servers declared — and any **risk findings** embedded on the row (detector, action taken, severity). Request and response **content is never stored** by default.

Prompt capture (opt-in)

If your organization enables prompt capture, transcripts are stored **redacted** — secrets and personal data are replaced with `[REDACTED:kind]` markers before anything is written. Capture is off by default, and audit rows deep-link to the captured conversation when it exists.

Blocked conversations

When the AI classifier issues a blocking verdict, the whole conversation is flagged: subsequent requests are rejected until an admin clears the block from the Policy page. This turns a finding into an enforced stop, not just a log line.

Compliance evidence

The Compliance screen exports an evidence pack (policies in force, detector actions, activity summaries) for auditors — see the Compliance & SIEM guide.

Read the activity feed

Activity is the audit trail: every AI request your developers send through the Gateway, with the policy decision attached. The small dot beside the title means the view is updating live.

The four tiles

Requests and **Blocked** for the selected window, plus **Top models** and **Top MCP servers** — a quick read on what your team leans on.

The columns

Column	What it tells you
When	Request time
Developer	Who, and which tool (Claude Code, Cursor, Copilot…)
Model	The model actually used
Signals	Risk findings and MCP servers involved
Mode	`Subscription` (their own plan) or `API key` (your managed key)
Outcome	`OK`, `Blocked`, or an upstream error
Tokens / Duration	Size and latency

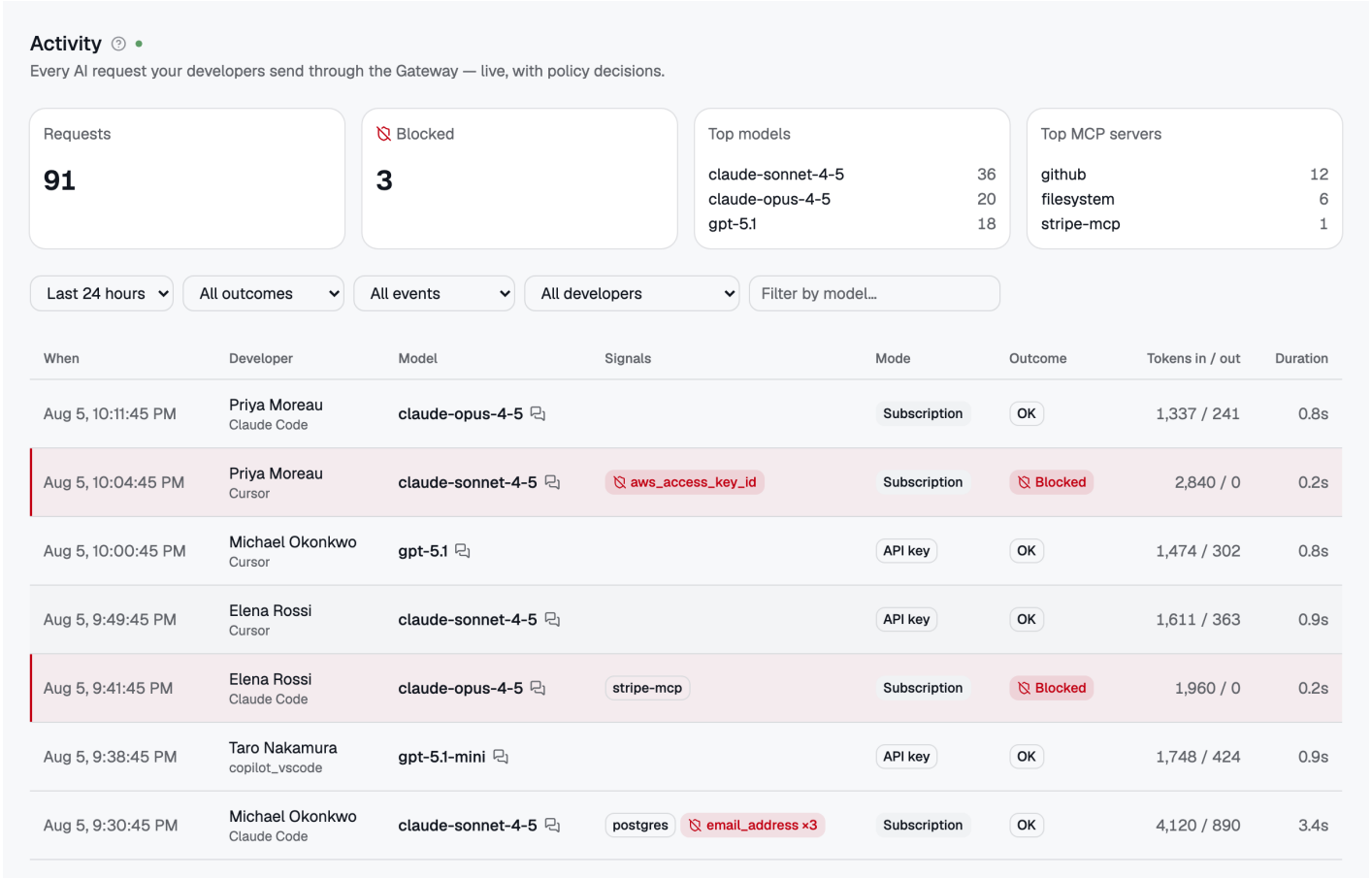
A **blocked** row is tinted and carries a red rail on the left, so it stands out when you're scanning rather than reading.

Filters

Time window, outcome, event type, developer, and a model filter. The common investigation — "show me everything blocked in the last 24 hours" — is two clicks.

Mode is worth understanding

`Subscription` means the developer's own AI subscription paid for that request and Sentilai only governed it. `API key` means it went through your organization's provider key and is billed to you. Both are audited identically; the difference is who pays.



The Activity feed: every AI request with its policy decision. Blocked rows carry a red rail.

Read a conversation

When prompt capture is on, **Conversations** holds the transcripts.

The list

Every captured conversation with the developer, the tool, an excerpt of the first message, how many messages it has, and when it was last active. Filter by developer.

The transcript

Messages in order, each labelled by role — **Developer**, **Assistant**, or **System prompt** — with timestamps. A "truncated" badge means the content exceeded what is stored for a single message.

Seeing the system prompt is often the most instructive part: it shows what the tool told the model before your developer typed anything.

Redaction markers

Amber shield badges naming a kind of secret mark where redaction happened. The transcript contains `[REDACTED:aws_access_key_id]` rather than the key, because redaction runs **before** storage — the value was never written down.

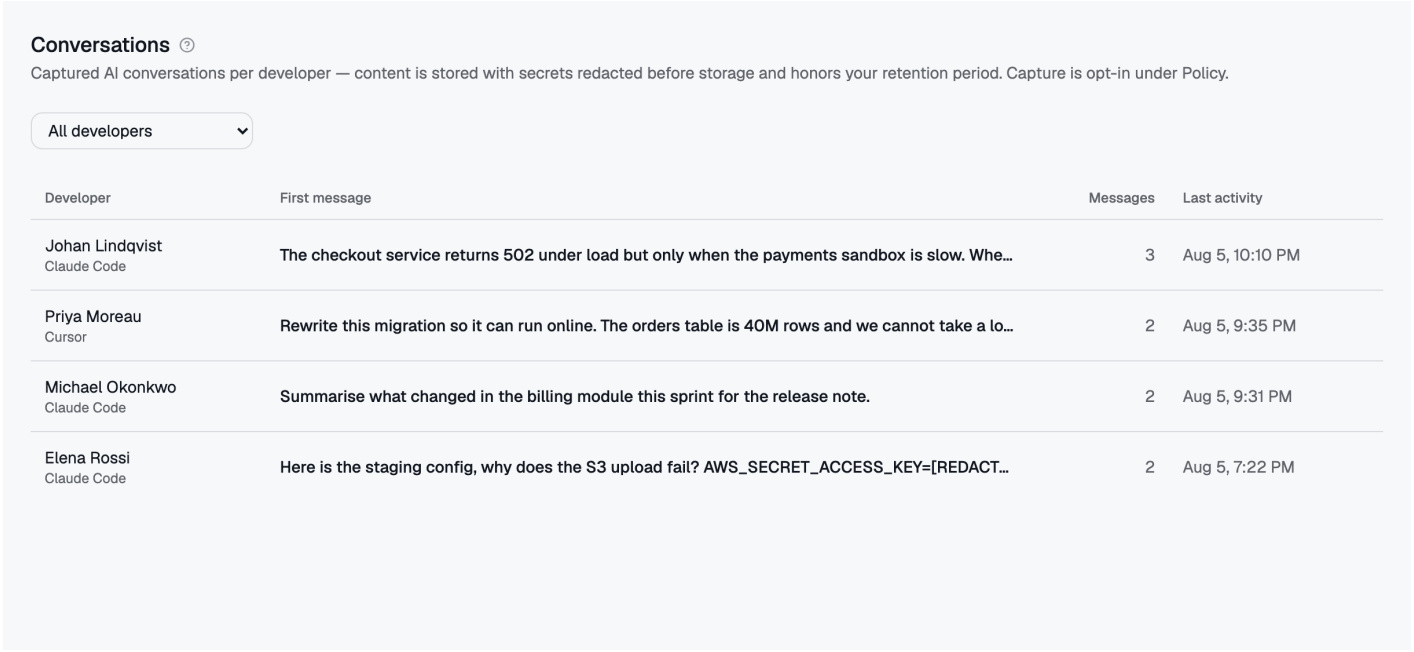
Two different empty states

- **"Prompt capture is off"**, with a link to Policy. Nothing is wrong; the feature is not on.
- **"No conversations captured yet"** — capture is on, nothing has come through.

Telling these apart is deliberate. An admin who sees a blank list usually assumes a bug.

What support can see

Nothing here. Sentilai support can see that a request happened and what the policy decided; conversation content is not readable by us.



Conversations — reconstructed sessions, grouped per developer. Secrets are redacted before storage, so a pasted key is already gone here.

Filters and the four tiles

Activity is a live feed — it refreshes every ten seconds, and the pulsing dot beside the title means what you are looking at is current.

The tiles

Over the selected window: **Requests**, **Blocked**, **Top models** (top three) and **Top MCP servers** (top three). Each shows "No data yet" rather than a zero when there is genuinely nothing, so you can tell "quiet" from "broken".

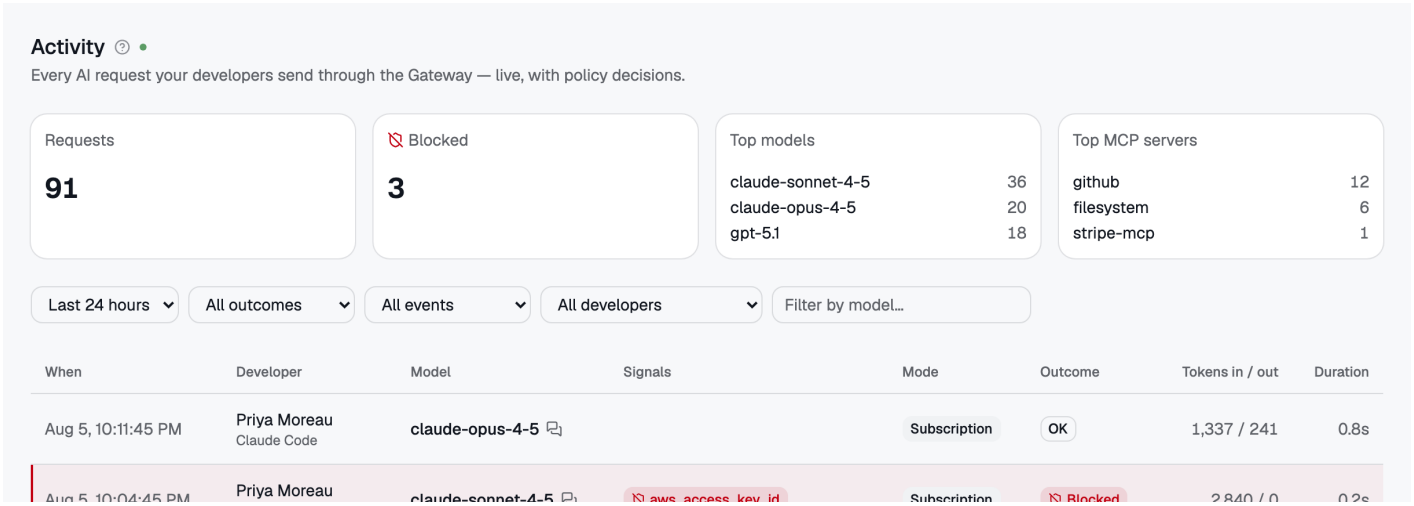
Top MCP servers is the one people find most surprising. It is a ranking of what your AI assistants are actually reaching for.

The filters

- **Time window** — last 24 hours, 7 days or 30 days.
- **Outcome** — all, OK, blocked, or upstream error.
- **Event type** — all, requests, or risk events.
- **Developer** — a specific person. The list includes admins as well as developers, because admins can have governed tools too.
- **Model** — free text, so `opus` narrows to everything with that in the model name.

Combinations worth knowing

- *Blocked, last 7 days* — the review you should do weekly. It tells you whether your policy is doing something useful or just being annoying.
- *Risk events, last 30 days* — what the detectors are finding, independently of whether anything was stopped.
- *One developer, 24 hours* — what you look at when somebody says "it stopped working".



The four tiles and the filter row. "No data yet" rather than a zero, so quiet reads differently from broken.

The Signals column

The densest column in the table, and the one that repays learning.

MCP server badges

Which MCP servers this request declared. This is how you find out that a request to summarize a pull request also had a database server and an HTTP client available.

Risk finding badges

One per detector that fired, in a warning colour, with a count. `secrets ×2` means the secret scanner found two.

The badge never shows the value. Findings carry the kind and the count only — the detectors are built so that the matched text is never stored. For the dependency detector, the tooltip does show the offending **package names**, because a package name is not a secret and knowing which one was proposed is the entire point.

The classifier badge

A millisecond figure — how long the prompt-injection classifier took — with the score in its tooltip. It is shown because the classifier runs in parallel with the upstream call and you should be able to confirm that it is not costing your developers latency.

Reading a row quickly

A row with no badges is ordinary traffic. A row with MCP badges and no risk badges is an agent doing agentic things safely. A row with several MCP badges *and* a risk finding is the one to open.

The other columns

When

Request time. The feed is newest-first.

Developer

Who sent it, and which tool. A dash with a tooltip means the request arrived with a device token issued before attribution was in place — the developer should reconnect the tool in the Endpoint Suite, after which their name appears.

Model

The model actually used, which is not always the one the developer thinks they selected.

Mode

Subscription or **API key** — whose money paid. See Connection modes.

Outcome

OK, **Blocked**, or an upstream error passed through from the provider. Blocked rows are tinted and carry a red rail down the left edge, so a scan of the screen finds them without reading.

Tokens and duration

Size and latency. Useful for spotting the agent loop that is quietly burning ten times more tokens than anybody expected.

Opening a row

Clicking a row opens its captured conversation — but only when there is one. That requires **prompt capture** to have been on at the time. Without it, the row is the record.