

Making Production Legible to Agents: Lessons from Building an Observability MCP

November 2026

Liz Fong-Jones

Technical Fellow
Honeycomb

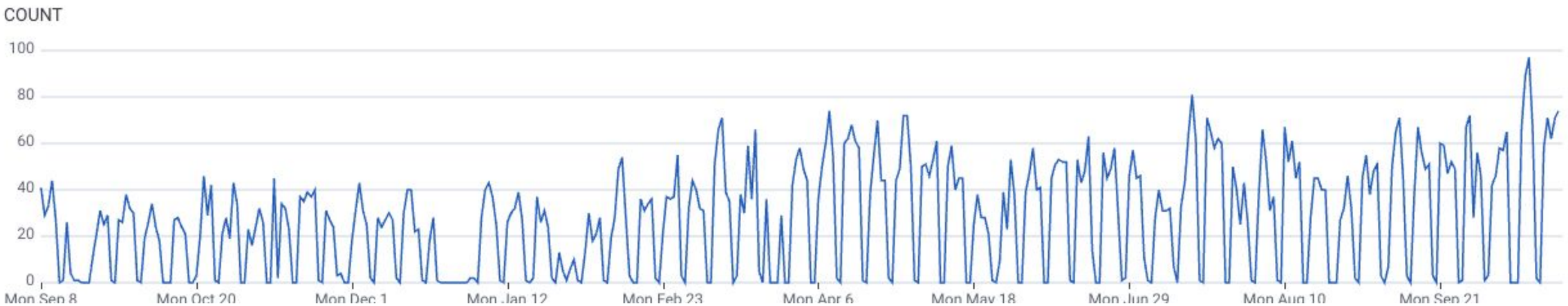
© 2026 Hound Technology, Inc. All rights reserved.

Our journey: PRs merged per day

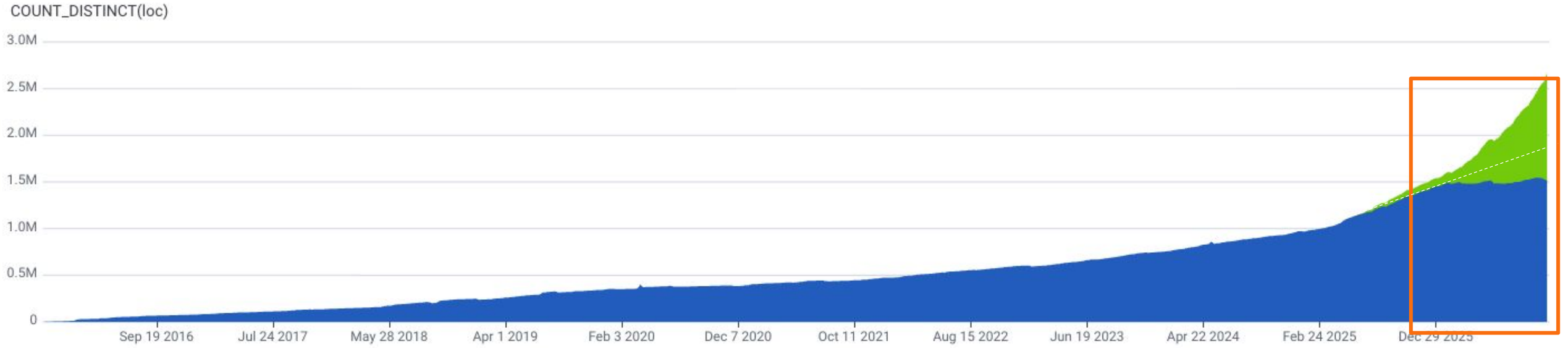
~30 to ~74

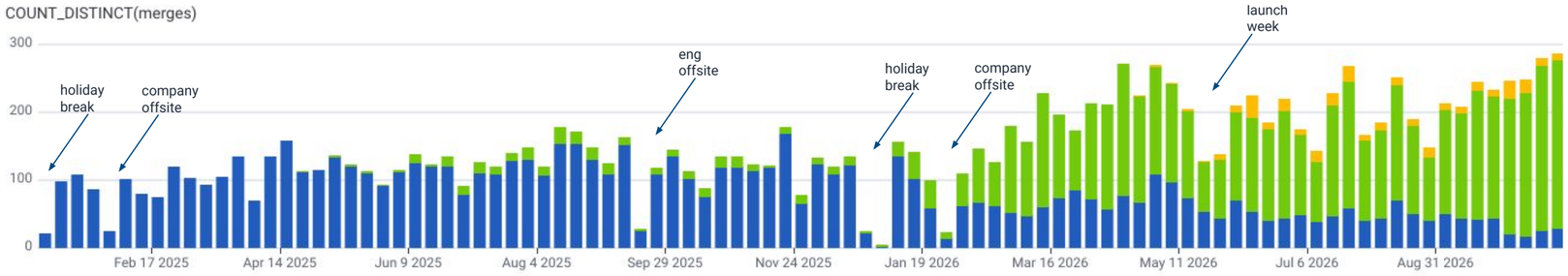
Validation is now the bottleneck.

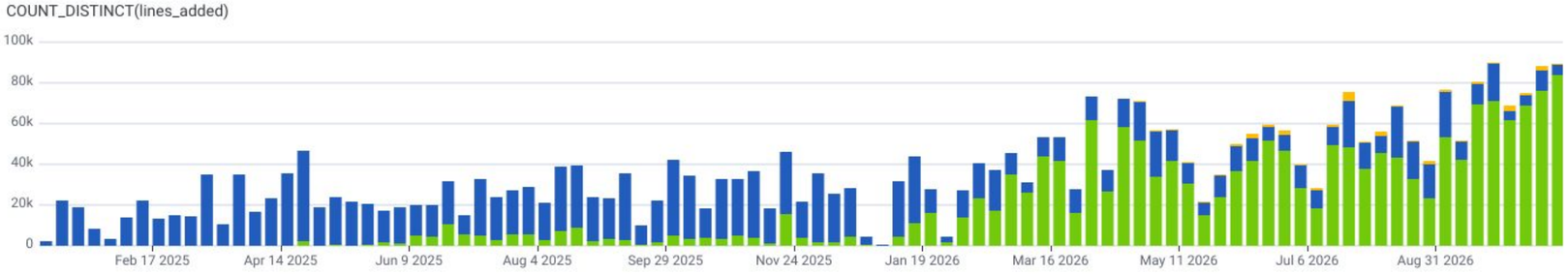
Observability helps us validate.



The shape of this curve changed.







© 2026 Hound Technology, Inc

Shipping at this rate only works if shipping is safe

© 2026 Hound Technology, Inc. All rights reserved.

An agent that can't see production guesses.

One that can is testing hypotheses.

© 2026 Hound Technology, Inc. All rights reserved.

Let the agent check what it shipped

- **Link running code to its change rationale**

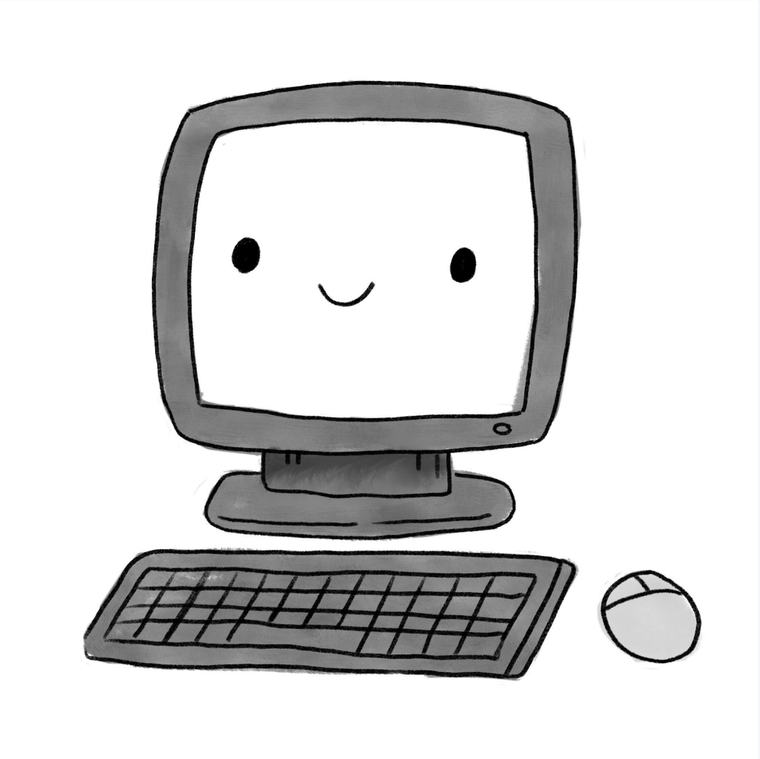
  Span attributes for the PR (commit sha) and link agent session to PR

- **Let the agent query its own change in production**

  Latency, errors and cost for real users, through the observability MCP

- **Close the loop before the next change**

  What it finds feeds the next PR or a review rule



© 2026 Hound Technology, Inc. All rights reserved.

Your telemetry quality sets the ceiling

- **Describe your data**

  A person with no docs would misread it too (ms? sec? cumulative?)

- **Tools can't rescue undescribed data**

  Descriptions help agents and humans alike

- **The cheapest MCP improvement**

  The agent won't do it for you



© 2026 Hound Technology, Inc. All rights reserved.

Roadmap

01 Why give agents production access

02 Design the tools; don't wrap the API

03 Spend tokens like they're money

04 Be consistent, so models can't misread you

05 Learn from agents you can't watch

© 2026 Hound Technology, Inc. All rights reserved.

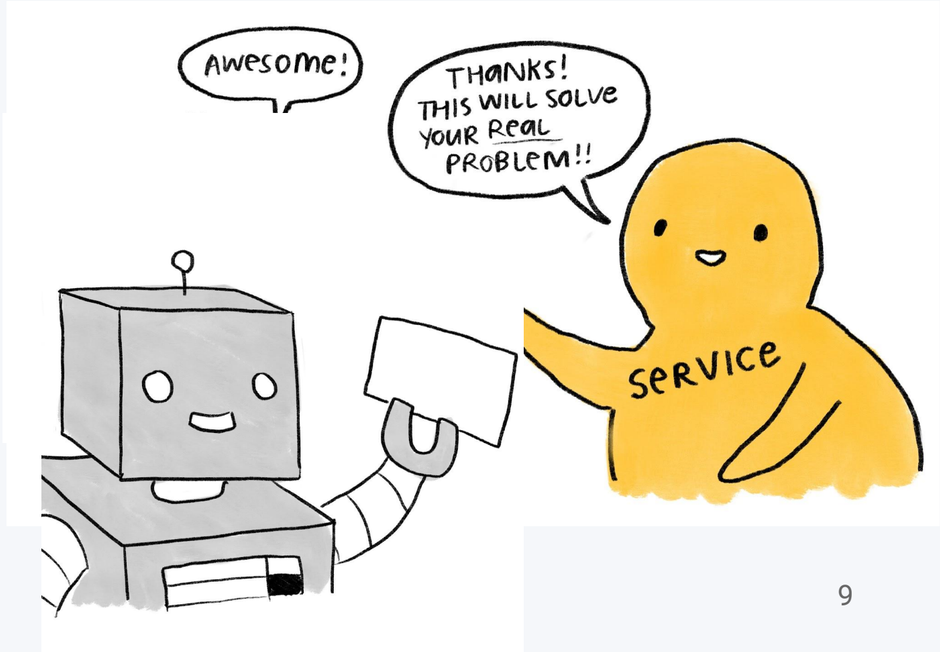
MCP: the tools the human had

- **Ticket → fix → PR in one session**

  Intent capture cost went to near zero
- **Runtime feedback and intent**

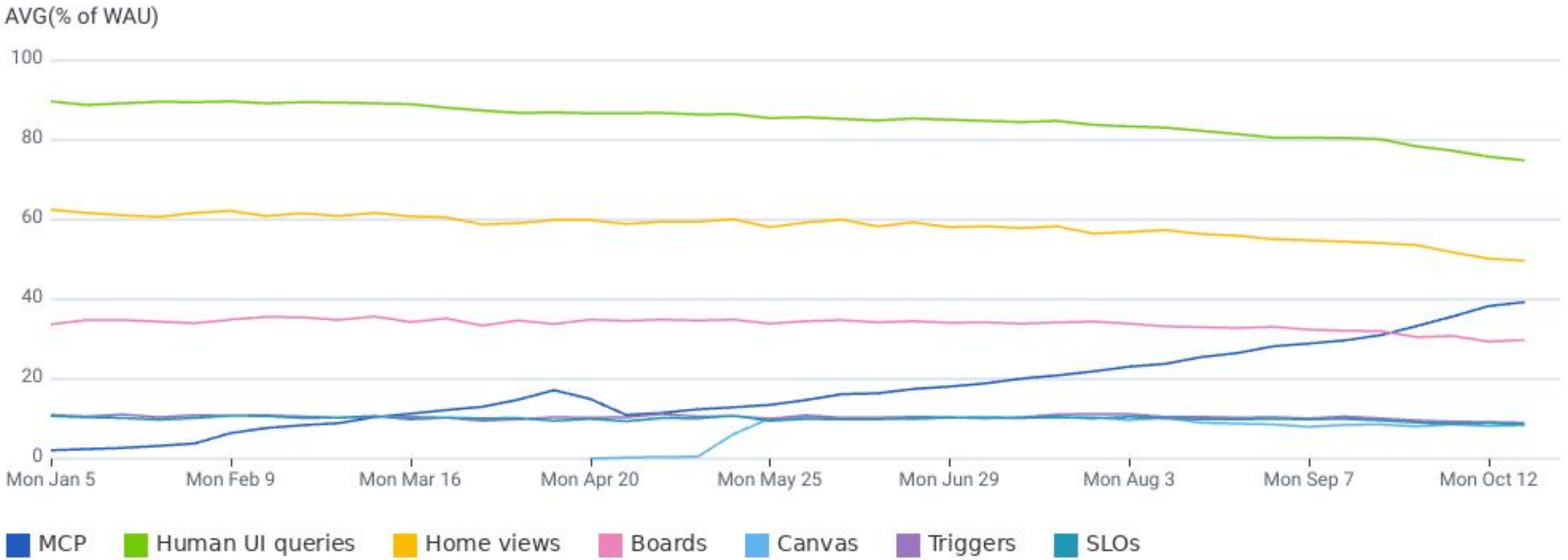
  Honeycomb, Playwright, Notion and Linear MCPs
  (observability, browser, docs, and task data)
- **Without these, an agent flies blind**

  A junior engineer wouldn't



© 2026 Hound Technology, Inc. All rights reserved.

~43% of weekly active users use MCP, up from ~2% in January

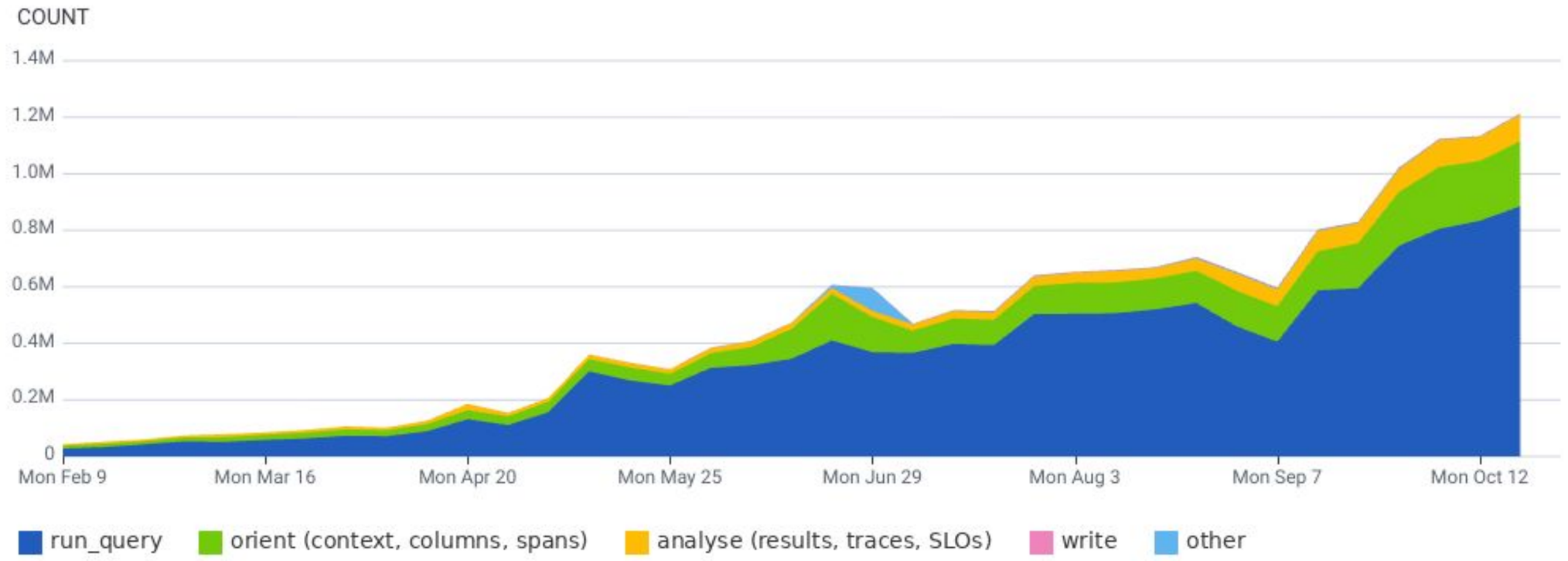


Share of weekly active users using each part of Honeycomb, Monday to Sunday weeks, 5 January to 25 October, customers only

© 2026 Hound Technology, Inc. All rights reserved.

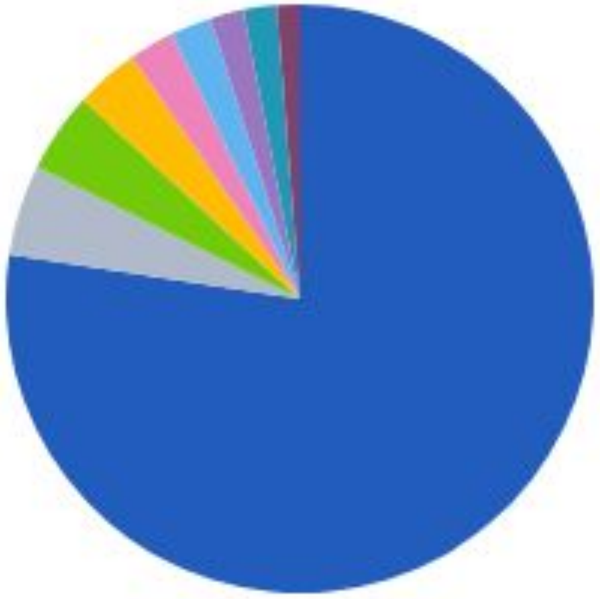
MCP calls per week are up thirty-fold

Since February; three quarters are still run_query



© 2026 Hound Technology, Inc. All rights reserved.

Three quarters of MCP calls are run_query



mcp.tool.name	%
run_query	77.40%
119 other categories	5.02%
get_workspace_context	4.39%
find_columns	3.50%
get_query_results	2.51%
get_trace	2.24%
get_span_details	1.85%
list_spans	1.84%
get_environment	1.25%

© 2026 Hound Technology, Inc. All rights reserved.

Design the tools, not the endpoints

© 2026 Hound Technology, Inc. All rights reserved.

The thing most people miss in their MCP is the *design* part.

– Austin Parker (@aparker.io)
prev. Director AI Strategy, Honeycomb

With thanks: much of this talk builds on his work

© 2026 Hound Technology, Inc. All rights reserved.

Tools should be heavy: one tool per workflow

- **A tool is a workflow**

  The steps an agent would otherwise chain itself

- **Weaker models get lost**

  Every extra step is a chance to misread a parameter

- **BubbleUp in one call**

  Agents used to query, download and diff in Python



© 2026 Hound Technology, Inc. All rights reserved.

Agents pave desire paths; build them

- **Watch what agents improvise**

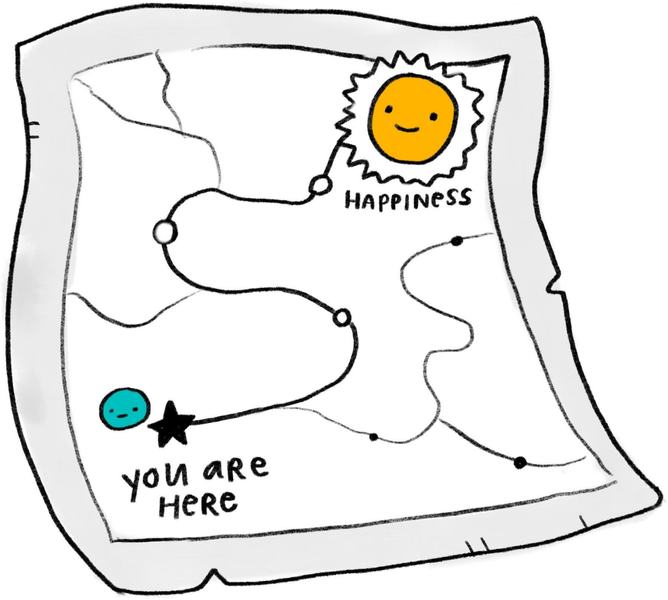
  They built BubbleUp from queries and Python

- **Repetition is your roadmap**

  Look for it in your MCP telemetry

- **Meet agents where they are**

  Our query spec is already in their training data



© 2026 Hound Technology, Inc. All rights reserved.

Tool output is a prompt

- **The whole result goes into context**

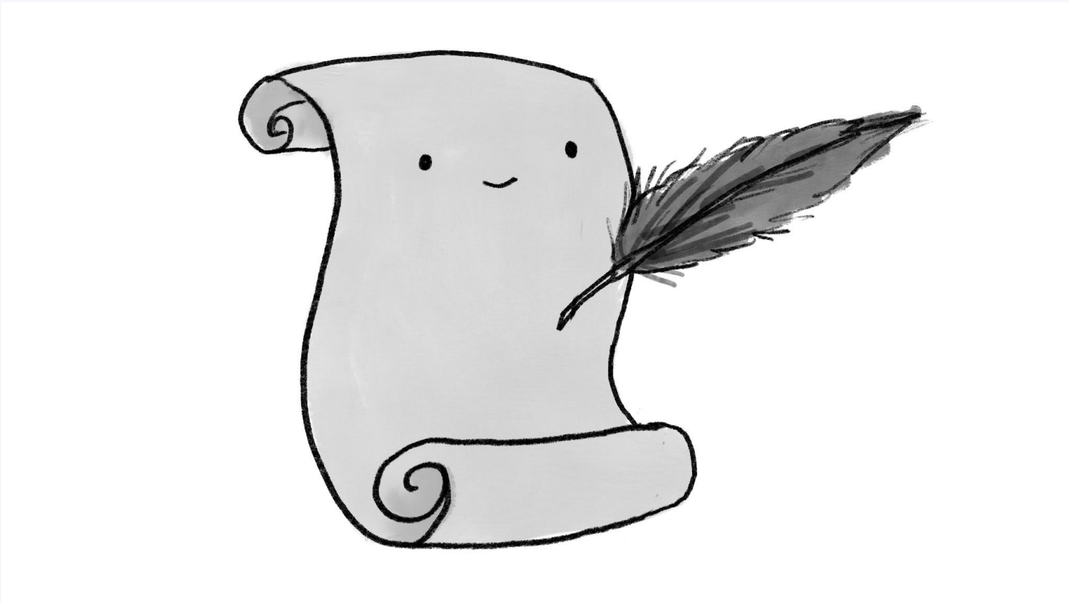
  Write tool output like a prompt

- **Lead with what matters**

  Error count and slowest span, then a trimmed CSV

- **Say where to look next**

  Hints save the agent a guess



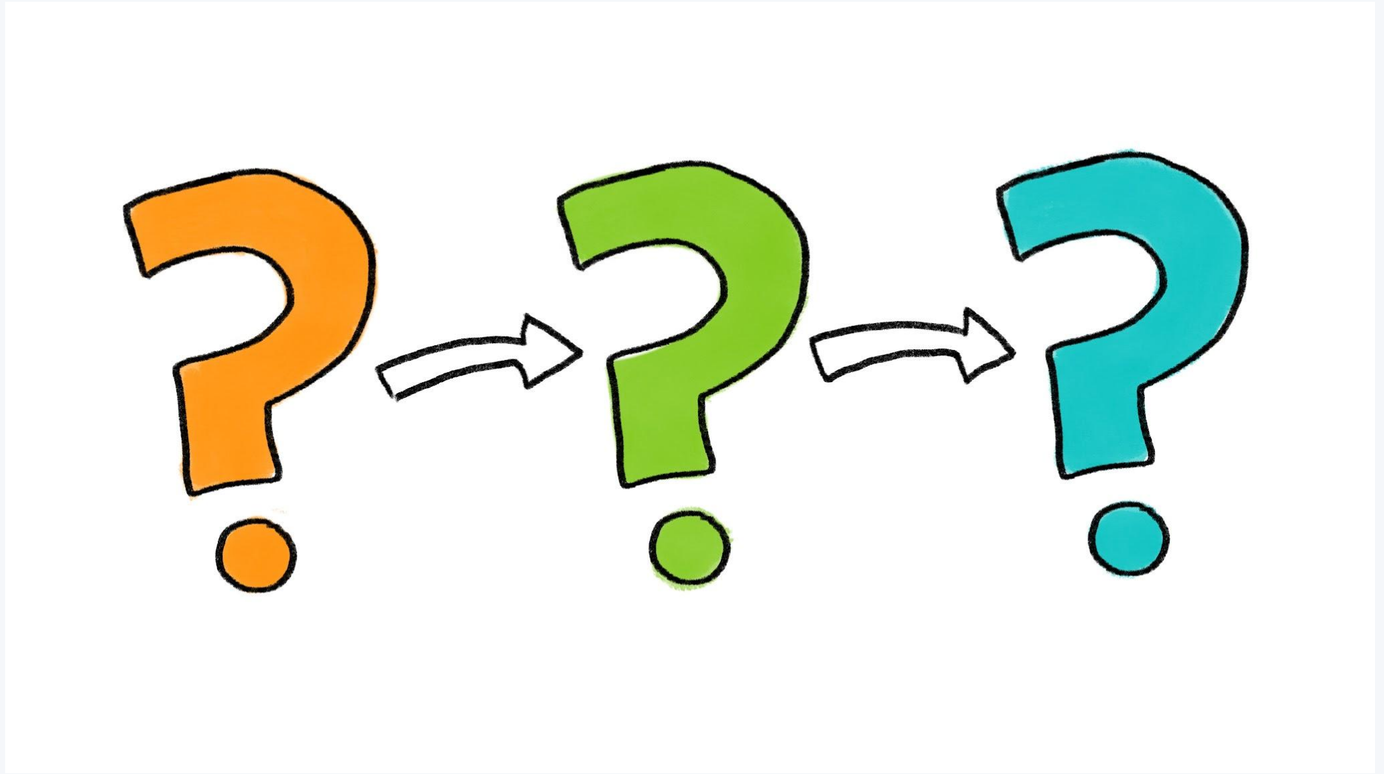
© 2026 Hound Technology, Inc. All rights reserved.

The three things that matter

Capabilities: what are we letting the agent do?

Context: what are we giving it to do those things?

Feedback: how do we learn from both?



© 2026 Hound Technology, Inc. All rights reserved.

Spend tokens like they're money (they are your customers')

© 2026 Hound Technology, Inc. All rights reserved.

The right format uses 5 to 11 times fewer tokens

A query result

~35,000
tokens as JSON

~21,000
tokens as CSV, about 40% fewer

~7,000
tokens as ASCII tables

A service map

~3,400
tokens as SVG

~1,700
tokens as ASCII art

~300
tokens as Mermaid

© 2026 Hound Technology, Inc. All rights reserved.

Bigger windows aren't cheap or accurate

- **1M-token windows**

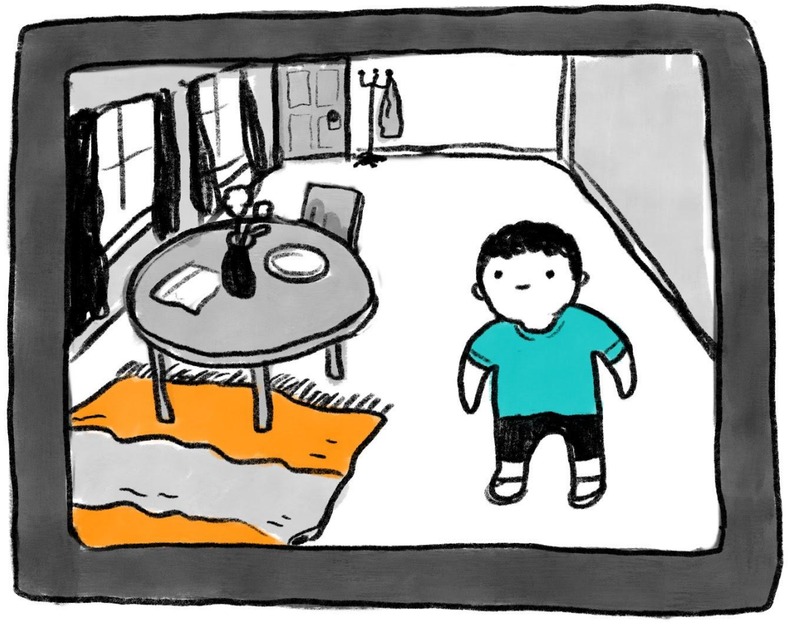
  Retrieval accuracy still around 70 to 80%

- **Irrelevant context is noise**

  It makes the model worse at the task

- **Every wasted token costs your customer**

  And brings them back more often



© 2026 Hound Technology, Inc. All rights reserved.

Compress by default; let the agent drill in

- **Outliers over digits**

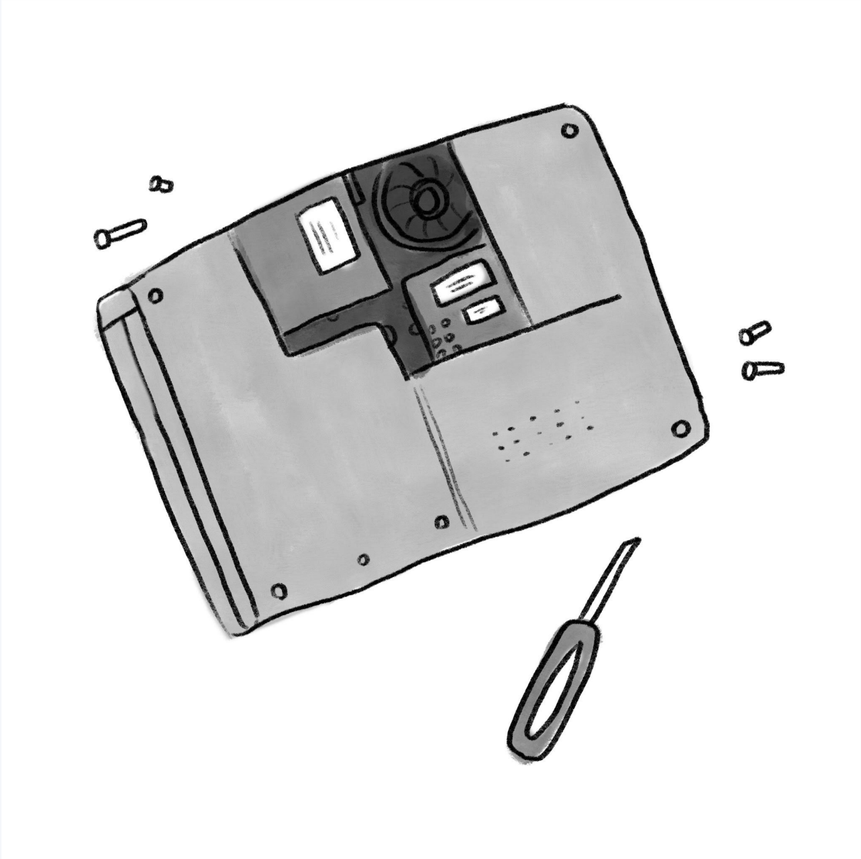
  In an incident, the agent needs what's different

- **ASCII charts just worked**

  Before we understood why

- **Progressive disclosure**

  Summary inline, plus a link to the full result



© 2026 Hound Technology, Inc. All rights reserved.

Your best guidance may get cut off

- **Descriptions cut at ~2,048 characters**

  Server instructions too, with no warning

- **Our guidance sat after the cut**

  Agents never saw it

- **Put the rules that matter first**

  Enforce the costliest on the server



© 2026 Hound Technology, Inc. All rights reserved.

Be consistent, so models can't misread you

© 2026 Hound Technology, Inc. All rights reserved.

One time format: from and to, ISO 8601

- **Three time formats, one confused model**

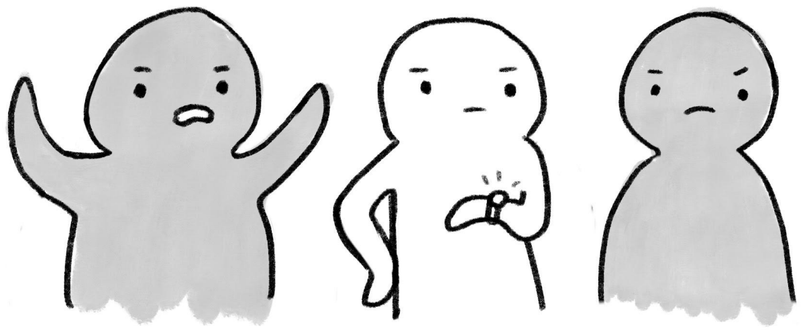
  Go durations, ISO strings, Unix timestamps

- **Failures clustered on time parameters**

  Models are bad at epoch maths

- **From and to, ISO 8601, everywhere**

  Failures dropped off a cliff



© 2026 Hound Technology, Inc. All rights reserved.

Changing the schema exposed who was scripting our MCP

~8%

tool-call errors before the change

~12%

after the change

~4%

extra errors were scripts, not agents

© 2026 Hound Technology, Inc. All rights reserved.

Every client parses schemas differently

- **One schema, many parsers**

  A oneOf broke Claude and Cursor

- **Renames bite too**

  A renamed tool vanished from one client

- **Test where your customers are**

  In the clients they actually use



© 2026 Hound Technology, Inc. All rights reserved.

API-key queries: simple and repetitive

- **API-key queries were simple**

  Many were identical

- **Some wanted a query data API**

  Another desire path

- **Rate limits keep it in bounds**

  Scope huge breakdown queries narrowly, too



© 2026 Hound Technology, Inc. All rights reserved.

Agents will spend your money, too

- **Long-range log searches via coding agents**

  Substring scans over months of data

- **Thousands of dollars a day**

  Until S3 throttled us

- **Know who is running what, and from where**

  Guidance helped; cost-based rate limits & batching are next



© 2026 Hound Technology, Inc. All rights reserved.

Learn from agents you can't watch

© 2026 Hound Technology, Inc. All rights reserved.

No session replay for third-party agents

- **You see what the agent asks for**

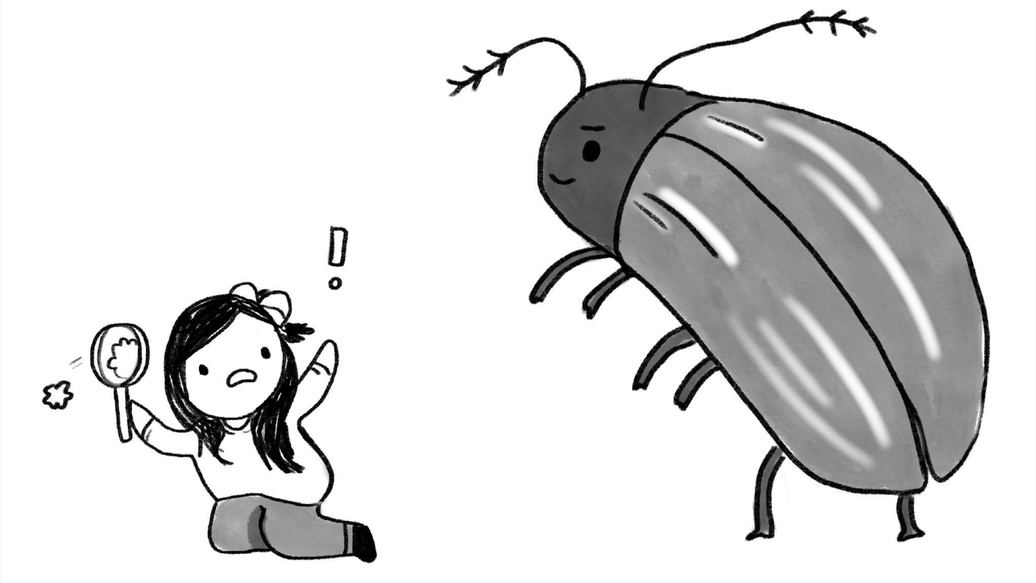
  What led to it stays hidden

- **Our own agent is an open book**

  We can analyse Canvas trajectories end to end

- **Customer agents stay opaque**

  So we learn some other way



© 2026 Hound Technology, Inc. All rights reserved.

Ask the agent what went wrong

- **Give models a feedback tool**

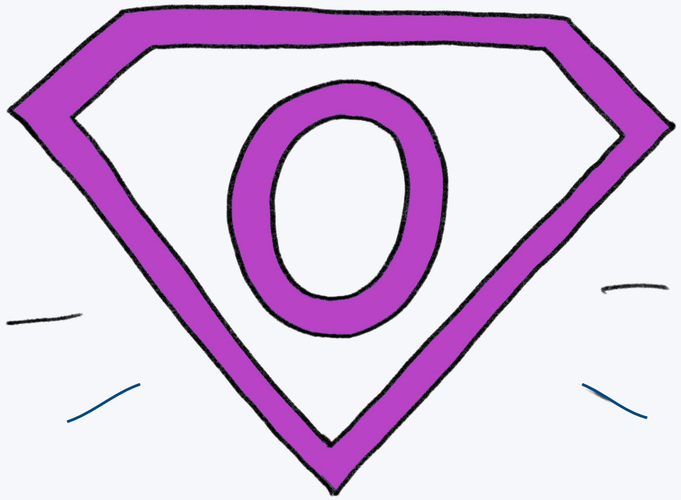
  Reports and suggested fixes land on a board we read

- **Mostly misconfigurations**

  Things the agent couldn't correct itself

- **One agent sent customer data, unprompted**

  Treat agent writing as untrusted input



© 2026 Hound Technology, Inc. All rights reserved.

Trace every request, score every query

- **Record every tool call**

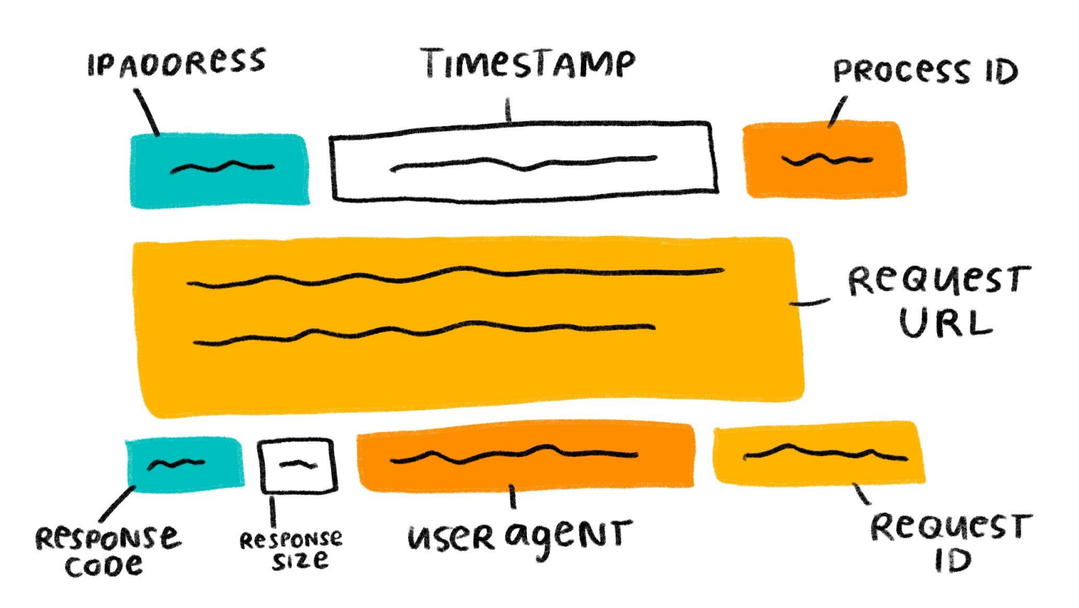
  The query and its metadata, on the span

- **Score queries as they arrive**

  Ours counts filters, breakdowns and calculations

- **Then do observability on it**

  Who is simple, what repeats, what fails



© 2026 Hound Technology, Inc. All rights reserved.

We couldn't tell who queried what

- **Missing environment and dataset**

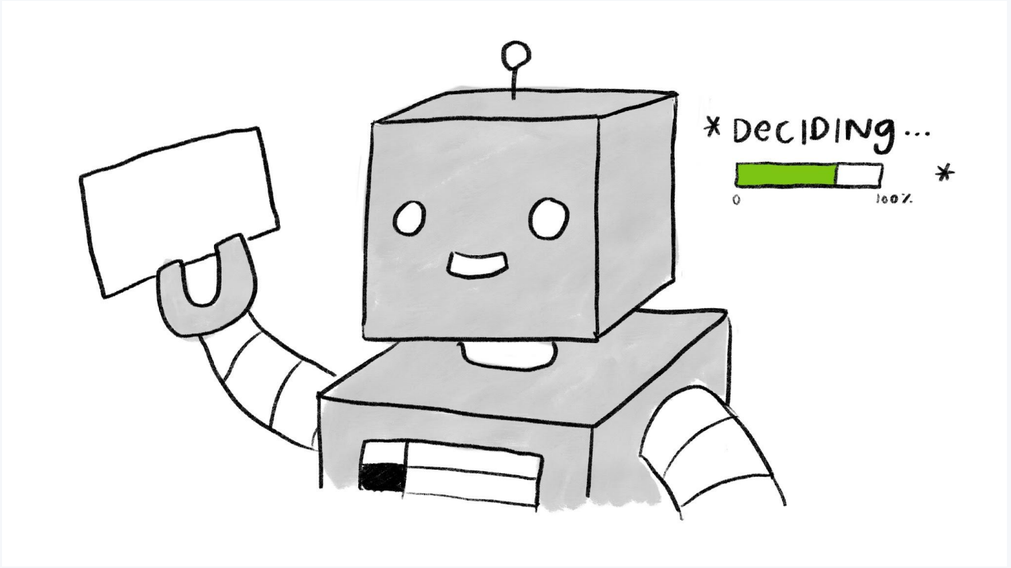
  A Canvas investigation flagged it for us

- **Some gaps are by design**

  No EU prompt data, for data residency

- **Design MCP telemetry early**

  Before you need to debug with it



© 2026 Hound Technology, Inc. All rights reserved.

Evals are unit tests; rerun on new models

- **Evals are unit tests**

  Fixed data, pass or fail judge, humans for taste

- **Track what it took**

  Tokens, tool calls, latency; fast retries are fine

- **Rerun on every new model**

  Dynamic schemas let us A/B test tools



© 2026 Hound Technology, Inc. All rights reserved.

Let humans cross-check the agent

- **Show your work**

  The queries and results the agent used

- **Or it may be confabulating at you**

  Evidence beats confident prose

- **Readable by people, too**

  The agent accentuates you; it doesn't replace you



© 2026 Hound Technology, Inc. All rights reserved.

Silent failures look like answers

- **Missing tools look like no tools**

  A customer noticed before we did

- **Wrong answers look like answers**

  run_query once ignored filters

- **Return explicit errors**

  And test the tool list, as well as the tools



© 2026 Hound Technology, Inc. All rights reserved.

What to do on Monday

- **Design the tools; don't wrap the API**

  Design tools around workflows, not endpoints

- **Spend tokens like they're money**

  Return the cheapest representation that keeps the signal; disclose progressively

- **Be consistent, so models can't misread you**

  Pick one format for time, IDs and units

- **Learn from agents you can't watch**

  Trace and score your own MCP traffic; let humans check the agent's work



© 2026 Hound Technology, Inc. All rights reserved.

Thank you.

danielmiessler.com/blog/ai-influence-level-ail

Usage analysis queries: AIL-4

Slide text & speaker notes: AIL-3

Images: AIL-0 (by bbghost.bsky.social)

Art should be made by artists, not machines.





TRANSPARENCY STANDARD

AI INFLUENCE LEVEL (AIL)

A system for rating the amount of AI contained within a creative work

VERSION 2.0
LEVELS 0 – 6
REVISED 2026-09

CLASS	LEVEL	DESIGNATION AND EXAMPLE	IDEA FROM	STARTED BY
HUMAN CREATED	AIL 0	**Human Created, No AI Involved** A handwritten letter, a painting created from an independent idea, a typed essay done without any AI-based tooling.	HUMAN	HUMAN
	AIL 1	**Human Created, Minor AI Assistance** An essay written by hand, but grammar and/or sentence structure was fixed by an AI.	HUMAN	HUMAN
	AIL 2	**Human Created, Major AI Augmentation** An article was written by a human, but it was significantly modified or expanded upon using AI tools.	HUMAN	HUMAN
AI CREATED	AIL 3	**AI Created, Human Full Structure** A human fully described a story, including giving extensive structure to an AI, and the AI filled it in.	HUMAN	HUMAN
	AIL 4	**AI Created, Human Basic Idea** A human had a basic idea for a story and gave it to an AI for implementation.	HUMAN	HUMAN
	AIL 5	**AI Created, Human Request** An AI writing tool has an API, and when invoked it produces full stories, including the basic idea all the way through the finished product.	AI	HUMAN
AI INITIATED	AIL 6	**AI Initiated, No Human Involvement** An AI agent decides on its own that a story should exist, comes up with the idea, then writes and publishes it.	AI	AI

DANIEL MIESSLER AIL v2.0 danielmiessler.com/blog/ai-influence-level-ail

© 2026 Hound Technology, Inc. All rights reserved.

honeycomb.io



Liz Fong-Jones

Technical Fellow | Honeycomb

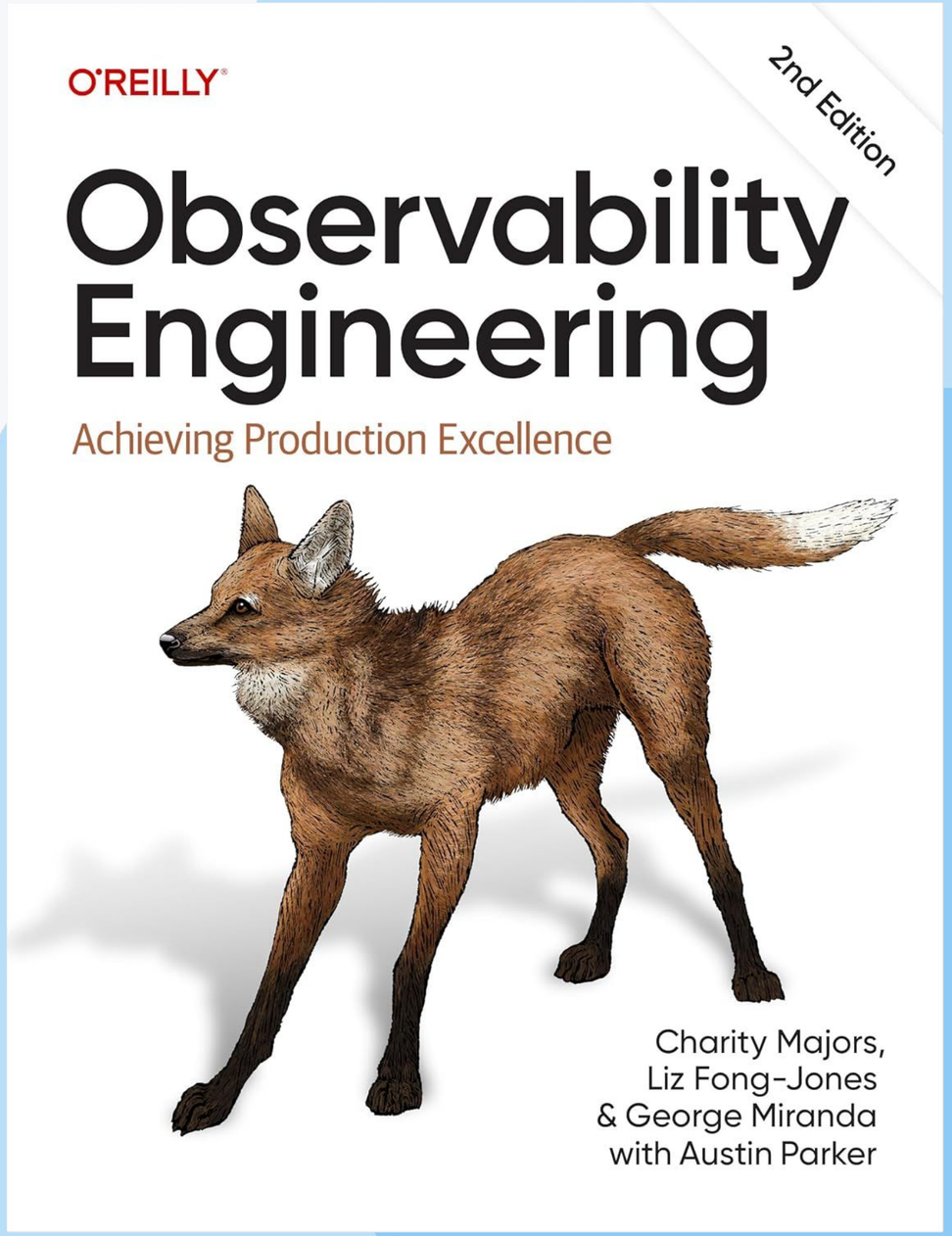
Sydney, NSW | Vancouver, BC

MACS (Snr) CP

linkedin.com/in/efong

lizf@honeycomb.io

@lizthegrey.com



© 2026 Hound Technology, Inc. All rights reserved.



© 2026 Hound Technology, Inc. All rights reserved.