



Measuring impact of Agents in Production using Honeycomb

Ken Rimple | Honeycomb

Financial Systems need to be fast and stable

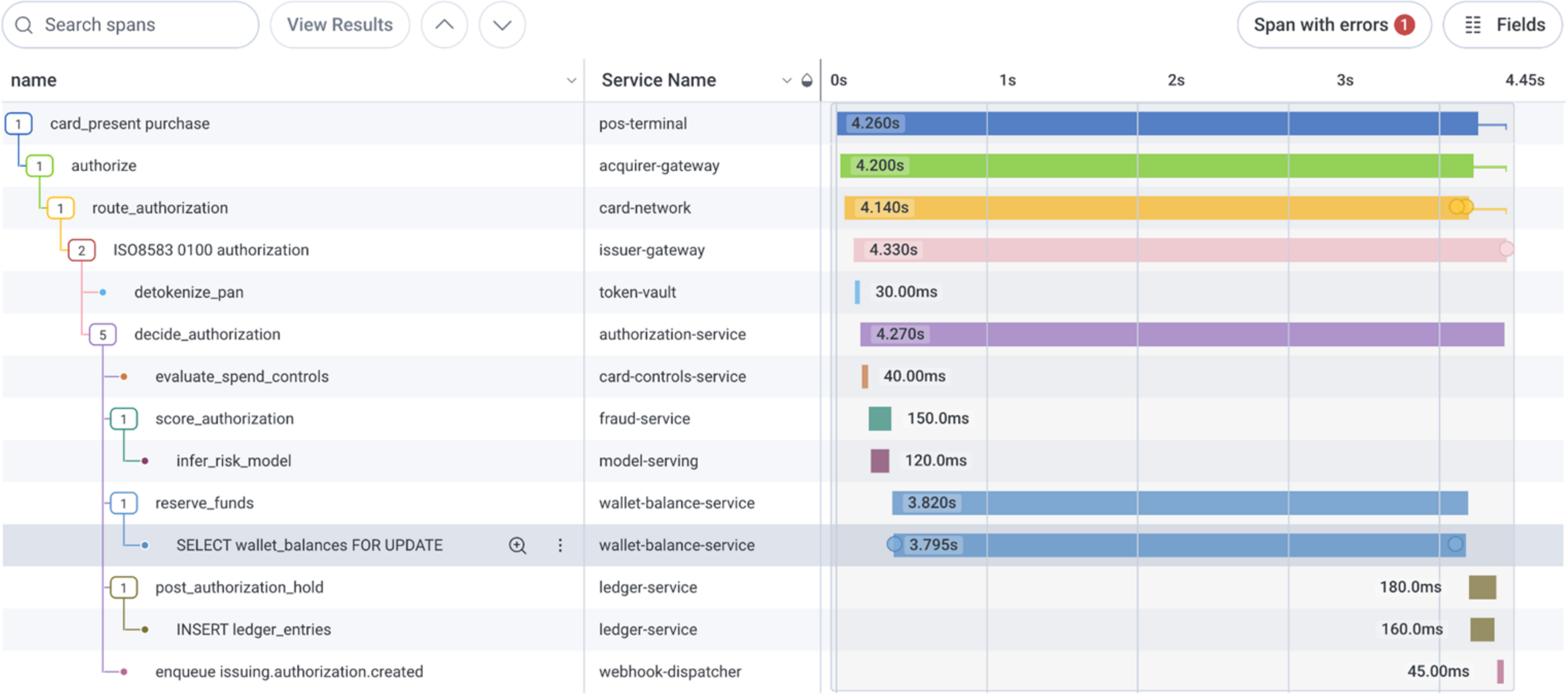
"Latency in payments is standing at the register embarrassed that the terminal hasn't shown approved yet, then Moov being fined for non-compliance of transaction response standards, making us and the brand look bad."



-- JJ Philipp, Principal Engineer



Honeycomb is the observability platform
for what comes next



Query in

All datasets in ai-in-finance

Share Save query

Define

SELECT P99(duration_ms)	WHERE None; include all events	GROUP BY None; don't segment
-----------------------------------	--	--

+ ORDER BY + HAVING + LIMIT

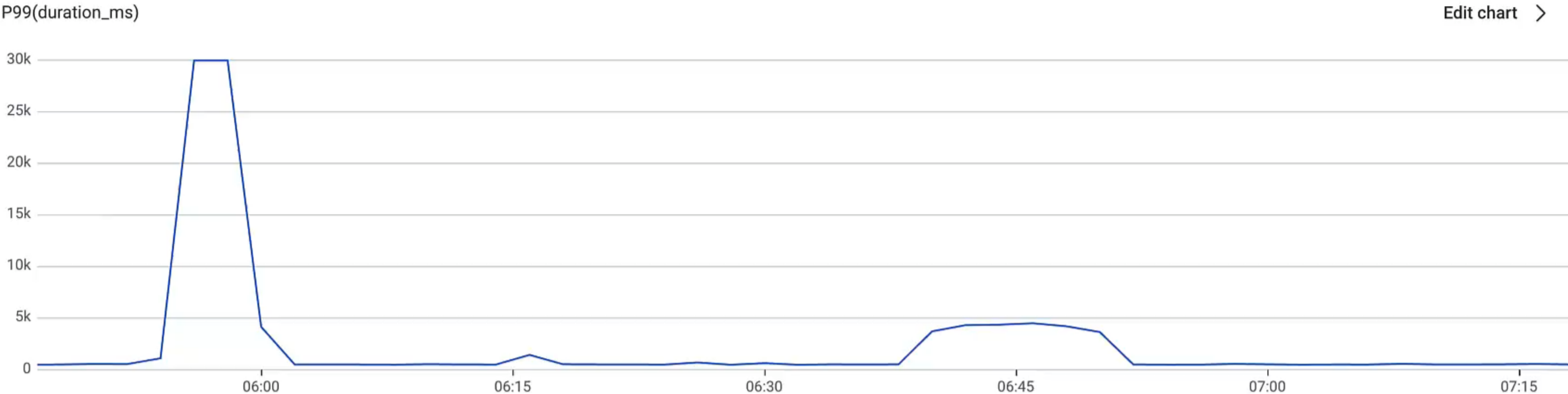
Edit Run

Query Results

Oct 16 05:46:39 – 07:18:23 UTC-04:00

Oct 16 2026 05:46:39 – 07:18:23 UTC-04:00 (Granularity: 2 min)

Download Settings



Trace 1ad3d56f08e84008a0892e3e4681851f [View events](#)

^

v

Span with errors 4

Fields

name	Service Name	0s	5s	10s	15s	20s	25s	31.86s
1 card_present purchase	pos-terminal	30.061s						
1 authorize	acquirer-gateway	30.000s						
1 route_authorization	card-network	29.965s						
5 ISO8583 0100 authorization	issuer-gateway	31.738s						
decode_iso8583	issuer-gateway	0.8693ms						
detokenize_pan	token-vault	23.73ms						
5 decide_authorization	authorization-service	360.8ms						
evaluate_spend_controls	card-controls-service	22.14ms						
1 score_authorization	fraud-service	216.5ms						
infer_risk_model	model-serving	187.2ms						
1 reserve_funds	wallet-balance-service	25.41ms						
SELECT wallet_bala...	wallet-balance-service	16.50ms						
1 post_authorization_...	ledger-service	66.36ms						
INSERT ledger_entries	ledger-service	56.93ms						
enqueue issuing.authorization.creat...	webhook-dispatcher	25.43ms						
mip.session.acquire	issuer-gateway	31.341s						
send_iso8583_0110	issuer-gateway						7.313ms	

card-network >
route_authorization



Fields

Span events (1)

Links (0)

bool error

true

str Timestamp

2026-10-16T09:57:57.77724541Z

Uniqueness

int network.issuer_timeout_ms

4000

str network.name

mastercard

str network.response_code

91

str network.response_source

acquirer_timeout

bool network.stand_in



All



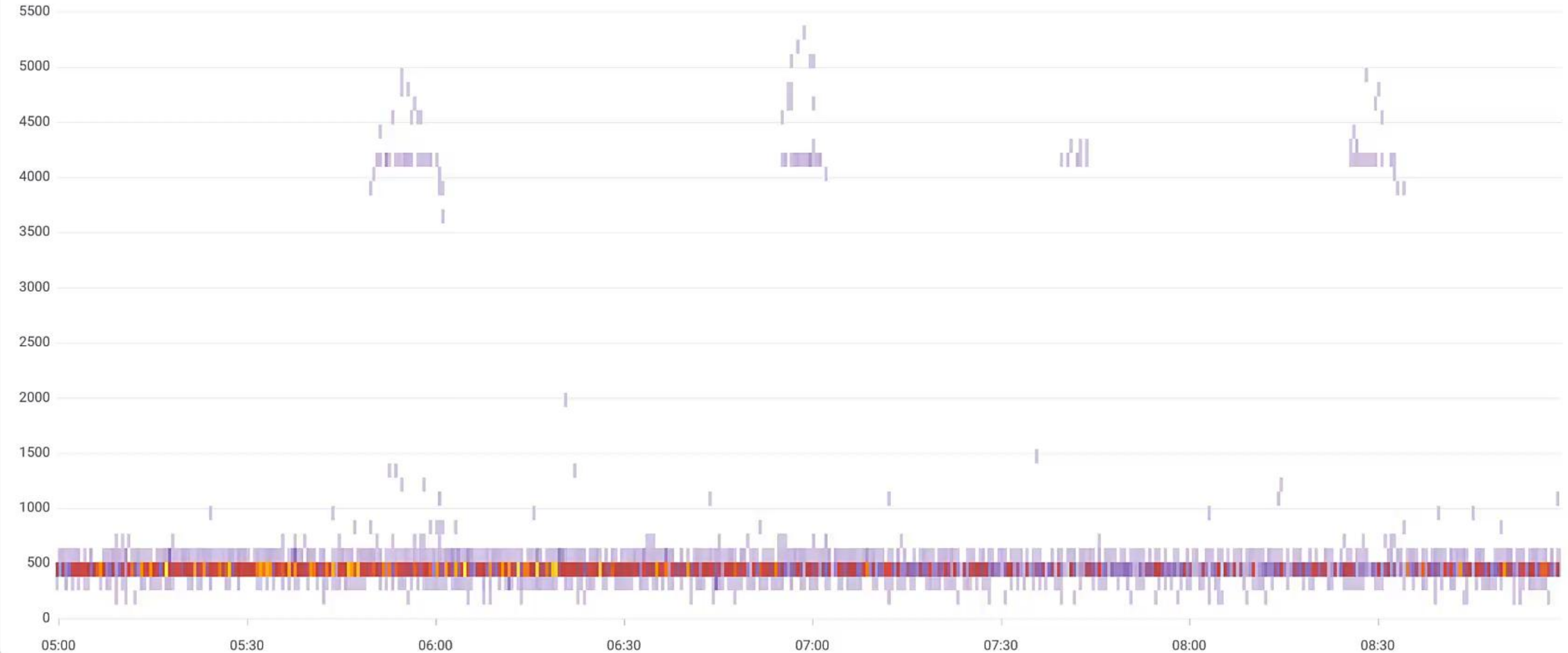
Original query: authorization latency heatmap

Oct 15, 2026 04:59:35 – 08:59:10 UTC-04:00 (Granularity: 30 sec)

4h

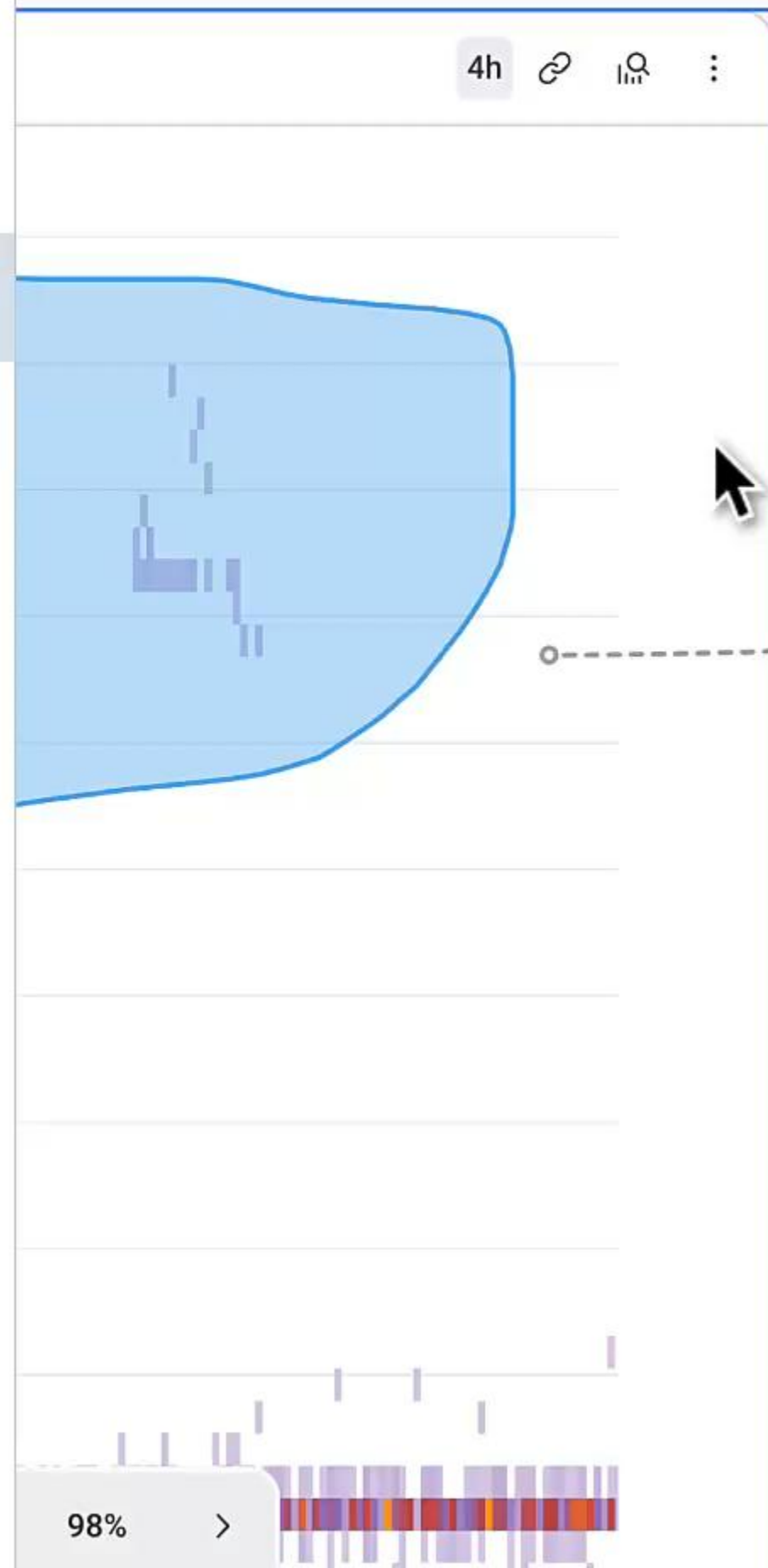


HEATMAP(duration_ms)





All



BubbleUp Analysis



130 selection / 984 baseline

Top Differences

Fields

Insights

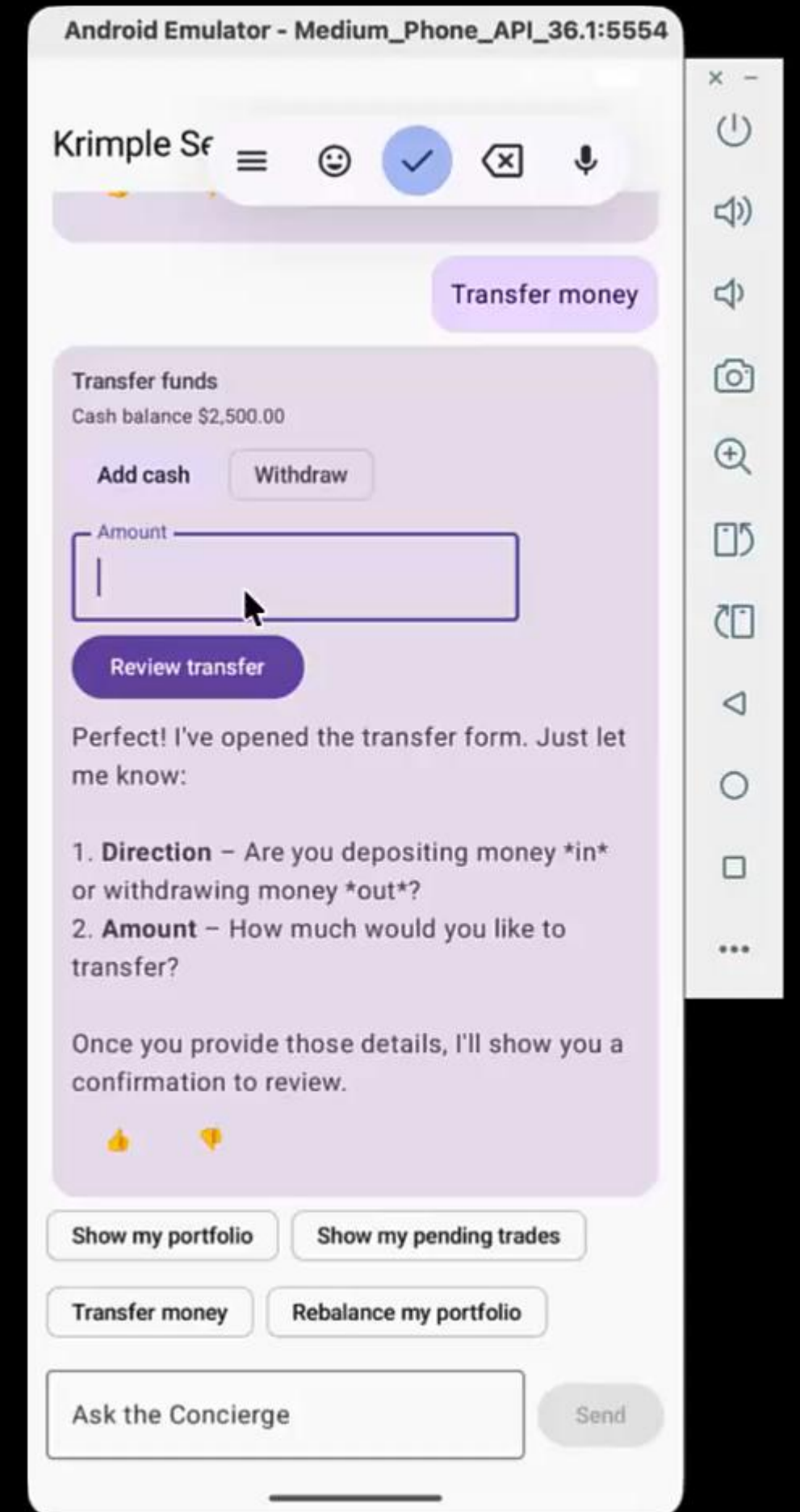
Field	Value	Selection	Baseline	Diff
card.bin_range	5310-5319	100%	19%	High
card.issuer_bank	Summit Community Bank	100%	19%	High
customer.tier	business	91%	16%	High
customer.segment	fleet	91%	16%	High
card.product_tier	fleet	91%	16%	High

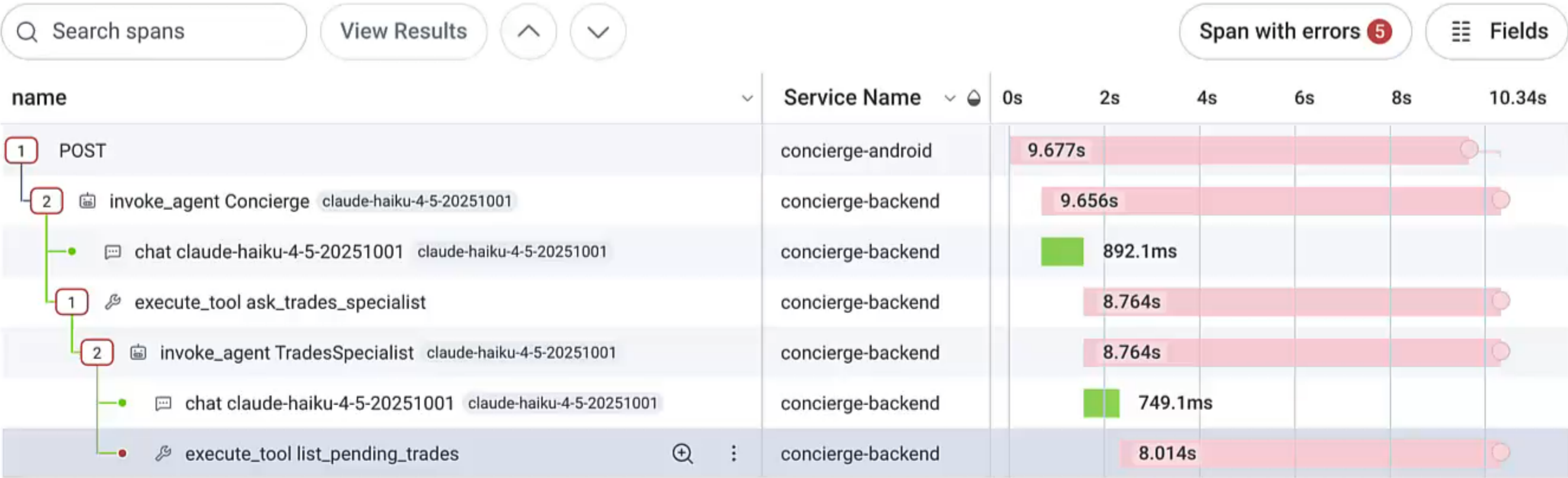


Honeycomb is Observability for What's Next

- Tracing shows you what is slow
- BubbleUp shows you what is different
- Everything your software knows you can measure

Agents are applications, too...





concierge-backend >

execute_tool list_pending_trades

Gen AI

Fields

Span events (1)

Links (0)

Identity

Operation

execute_tool

Conversation ID

1b6ee7a0-cd0e-4632-a414-f052559ae2f6

Agent

TradesSpecialist

Provider

anthropic

Errors (2)

Error Type

ToolTimeoutError

Exception Message

tool list_pending_trades did not finish within 8000 ms

Tool

Tool name

list_pending_trades

Call ID

toolu_01QKPWSZSYW4sJdKvA5coJ9Z

Arguments 2 B

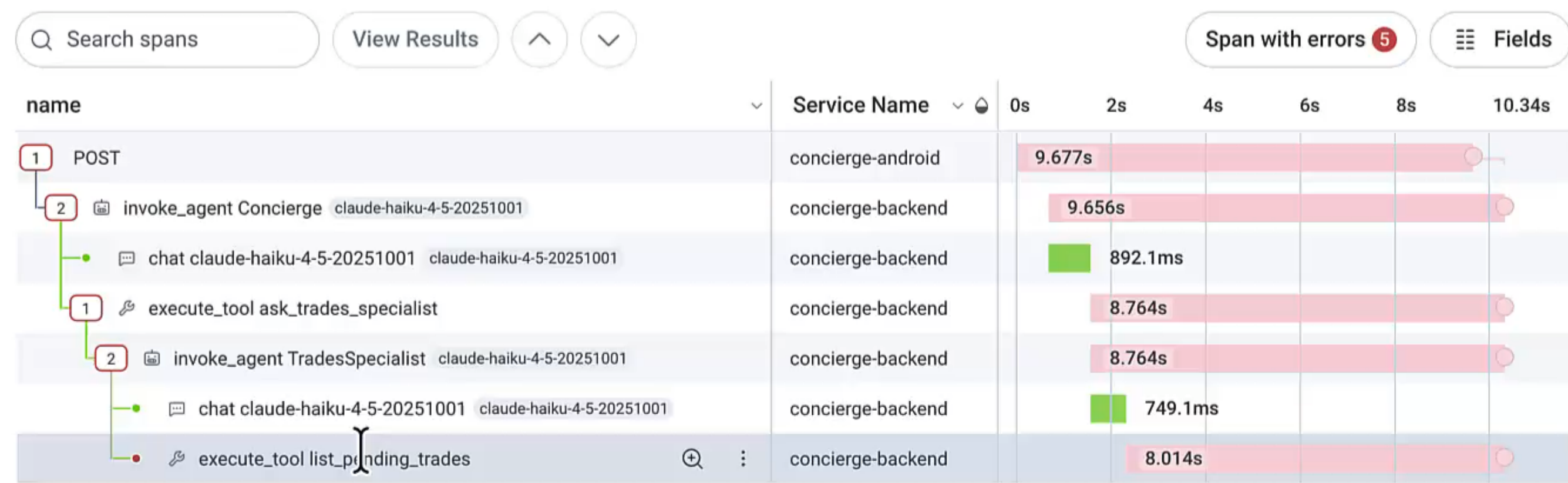
{}





trad >





concierge-backend >

execute_tool list_pending_trades

Gen AI

Fields

Span events (1)

Links (0)

Identity

Operation

execute_tool

Conversation ID

1b6ee7a0-cd0e-4632-a414-f052559ae2f6

Agent

TradesSpecialist

Provider

anthropic

Errors (2)

Error Type

ToolTimeoutError

Exception Message

tool list_pending_trades did not finish within 8000 ms

Tool

Tool name

list_pending_trades

Call ID

toolu_01QKPWSZSYW4sJdKvA5coJ9Z

Arguments

2 B

{}



Search spans

View Results

Span with errors 5

Fields

name	Service Name	0s	2s	4s	6s	8s	10.34s
1 POST	concierge-android	9.677s					
2 invoke_agent Concierge claude-haiku-4-5-20251001	concierge-backend	9.656s					
chat claude-haiku-4-5-20251001 claude-haiku-4-5-20251001	concierge-backend	892.1ms					
1 execute_tool ask_trades_specialist	concierge-backend	8.764s					
2 invoke_agent TradesSpecialist claude-haiku-4-5-20251001	concierge-backend	8.764s					
chat claude-haiku-4-5-202510... claude-haiku-4-5-20251001	concierge-backend	749.1ms					
execute_tool list_pending_trades	concierge-backend	8.014s					

Messages (3)

SYSTEM You are the Trades s... 478 B

You are the Trades specialist at Krimple Securities (a pretend broker). Use list_pending_trades to show pending trades (the app shows the list, so summarize in one sentence). To change a trade, call propose_trade_change; the app shows a confirm card, so tell the customer to tap Confirm. Call apply_trade_change only when the customer says they confirmed. Never claim a change happened unless apply_trade_change succeeded.

USER What are my pending tra... 81 B

What are my pending trades?

ASSISTANT Tool call: list_pending... 160 B

Tool call: list_... (toolu_01QKPWSZS...)

{}

Finish reason: tool use

Conversation

1b6ee7a0-cd0e-4632-a414-f052559ae2f6



Duration
26.2sec

Traces
2

LLM Calls
6

Tool Calls
4

GenAI Failures
4

Total Tokens
5,923

Est. Total Cost
\$0.008

Timeline

Agent Invocation LLM Operation Tool Call Other Failure Show failures only



- > Concierge
- > PortfolioSpecialist
- > TradesSpecialist

chat claude-haiku... Hide Full Trace

Gen AI	Fields	Span Events	(0)	Links (0)
--------	--------	-------------	-----	-----------

Identity	
Operation	chat
Conversation ID	1b6ee7a0-cd0e-4632-a414-f052559ae2f6
Agent	TradesSpecialist
Provider	anthropic
Model	claude-haiku-4-5-20251001
Response Model	claude-haiku-4-5-20251001
Response ID	msg_011Cg6UoDb5gXRMvdoRXuUCe

Parameters	
finish_reasons	["tool_use"]

Performance

INPUT TOKENS
854

OUTPUT TOKENS
39

CACHE WRITE TOKENS
0

CACHE READ TOKENS
0

Token usage 893 total

Input (854) Output (39)

Estimated Cost



All



Token spend, 11:55–12:00 UT...

- 2 of 19 conversations account for ~95% of tokens (~1.94M of ~2.03M) and ~91% of estimated cost (\$1.83 of \$2.01): [1f13fece031a5bbffa404](#) and [1d6e98c4177079f22b524](#).
- Nearly all of it is spent inside account-portfolio-agent on Opus 5.5, in an N+1 tool loop: ~140 `get_position_detail` calls spread over 21 LLM turns. The staircases in the heatmap are those turns.
- Prompt caching absorbs most of it (90% / 80% of input is cache reads). `1d6e...` cached less and cost more despite using fewer tokens.
- `1d6e...`'s final Opus call failed with `ValidationException: "Input is too long for the requested model"` at 94,664 input tokens. `1f13...` finished at 95,272, so these runs are sitting right at the context ceiling.

Next steps: a batch detail tool (or `get_positions` returning detail when `detail=full`) would collapse ~21 turns into ~2. Trimming tool results between turns would flatten the staircase.

[Source query](#) · [Failed call](#)

GenAI spans by operation, agent, conversation (11:55–12:00 UTC)



Agent note
Single ~\$0.41 call near 11:55 UTC
cache-read drop at the end of the call
`1d6e98c4177079f22b524`

Team Your chat

☒ Hide tool calls

series the whole input of the turn before it, plus about 3–6K tokens of new position details. The input grows from 285 tokens on turn 1 to 95K on turn 21.

- 95K tokens of actual content were sent 994K times over, about 10x more than the content itself. Output was only 5.4K tokens across all 21 turns.
- Prompt caching made it cheaper but not smaller. The context still keeps growing toward the model limit, which is where `1d6e98c4177079f22b524` failed.
- A second row in the diagram shows the same account without `detail`: "full" : 2 turns and 4.7K tokens for the whole conversation.

The markdown also estimates the effect of a batched detail call: about 3 turns and roughly 100K tokens instead of about 1M. That assumes the position details stay the same size.



What do you want to dig into next?



Analysis

Fields Insights

Showing 1-10 of 60

Top Value	Selection %	Distinct Values
arn:aws:bedrock:us-east-1:123456789012:inference-profile/us.anthropic.claude-opus-5-5-v1	100%	4
us.anthropic.claude-opus-5-5-v1:0		

BubbleUp Analysis

2 selection / 766 baseline

Top Differences Fields Insights

The anomalous selection is entirely associated with a single authenticated session (account AC-100234, conversation `1f13fece031a5bbffa404`) making an unusually complex multi-step portfolio detail request that retrieves all 144 position concierge agent, a pattern distinct from the broader baseline of simpler, varied agent interactions.

Aha! It's one account... in the fintech agent...

66%

AI SRE Agents That Join Your Team as Coworkers

Teams on Honeycomb Intelligence can now add Agent Teammates to their observability and SRE teams: more gets done, and MTTR comes down.



Ten years of expertise

Trained on Honeycomb's decade of SRE and observability practice.



They learn your systems

How your services behave and what your team knows, remembered.



They work on their own

Autonomous and proactive: they improve systems and resolve incidents.



ONE FOR EVERY PERSON

Personal Agent

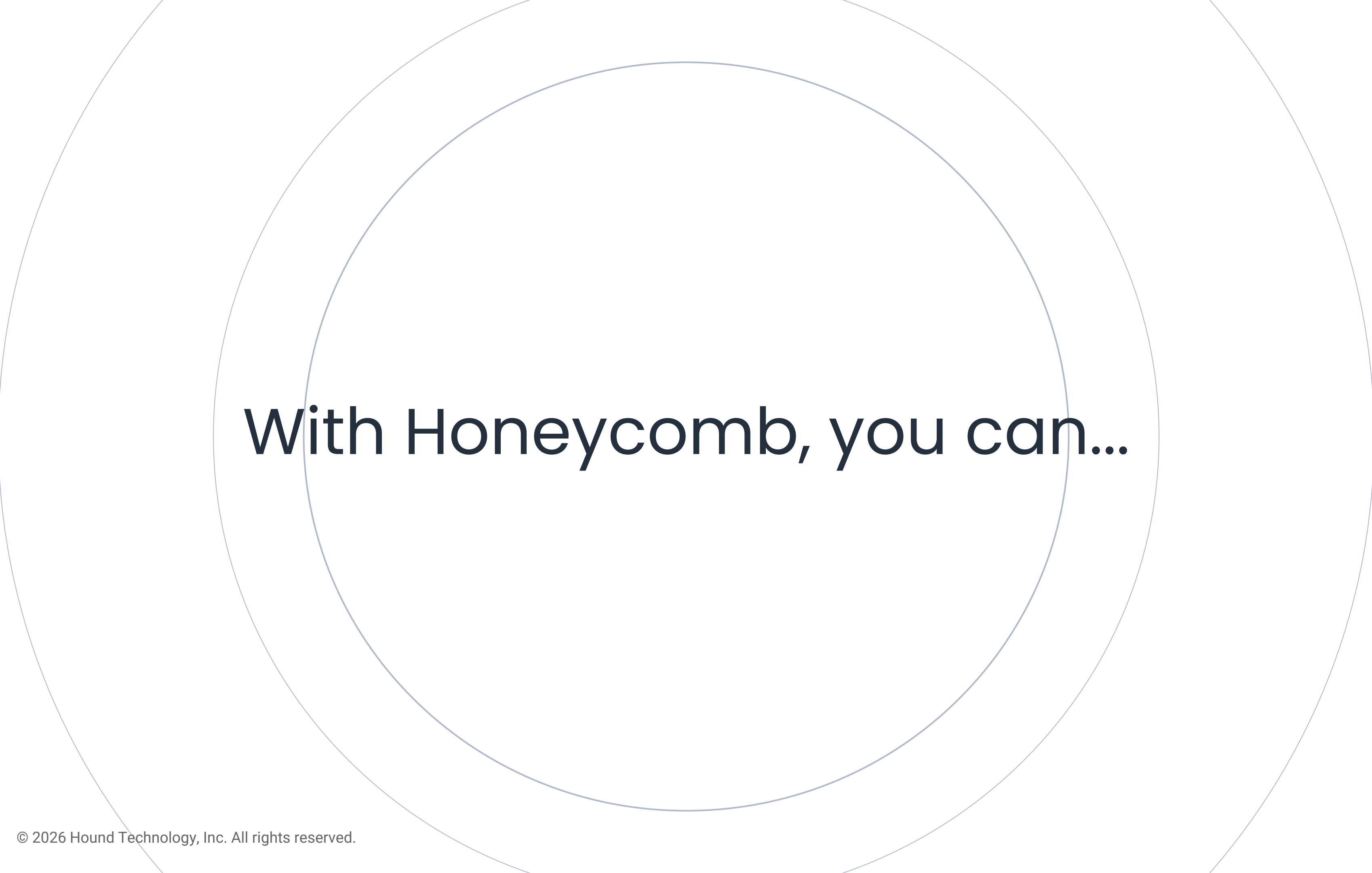
A collaborator and assistant for each human teammate. Ask it anything about production, and it learns how you work.



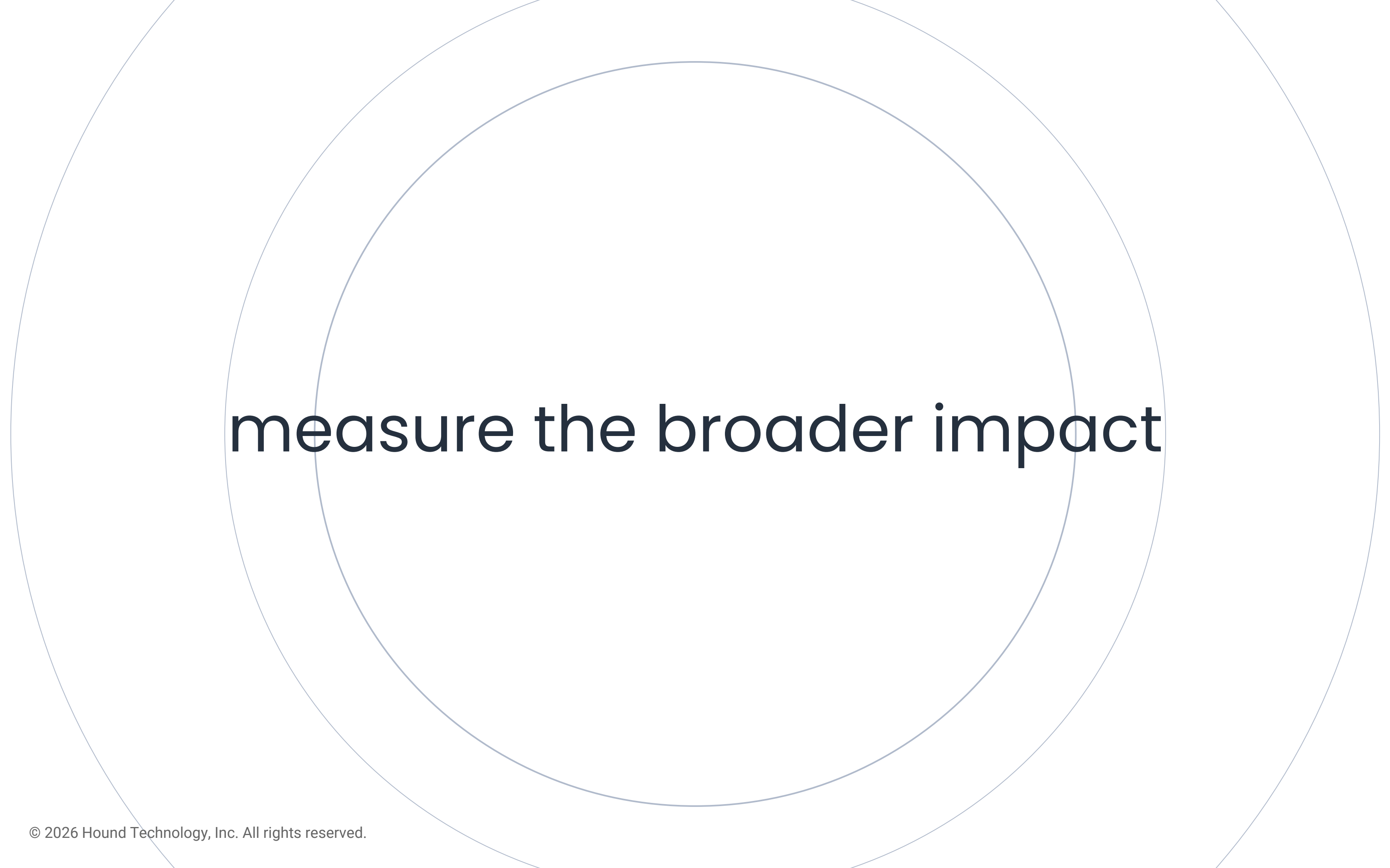
ONE OR MORE PER TEAM

Team Agents

Backing for your whole team or org, working in the background on goals you define and collaborating with everyone's personal agents.



With Honeycomb, you can...



measure the broader impact



...and see every detail



...in financial applications



...as well as agents.



See everything. Solve **anything.**

honeycomb.io