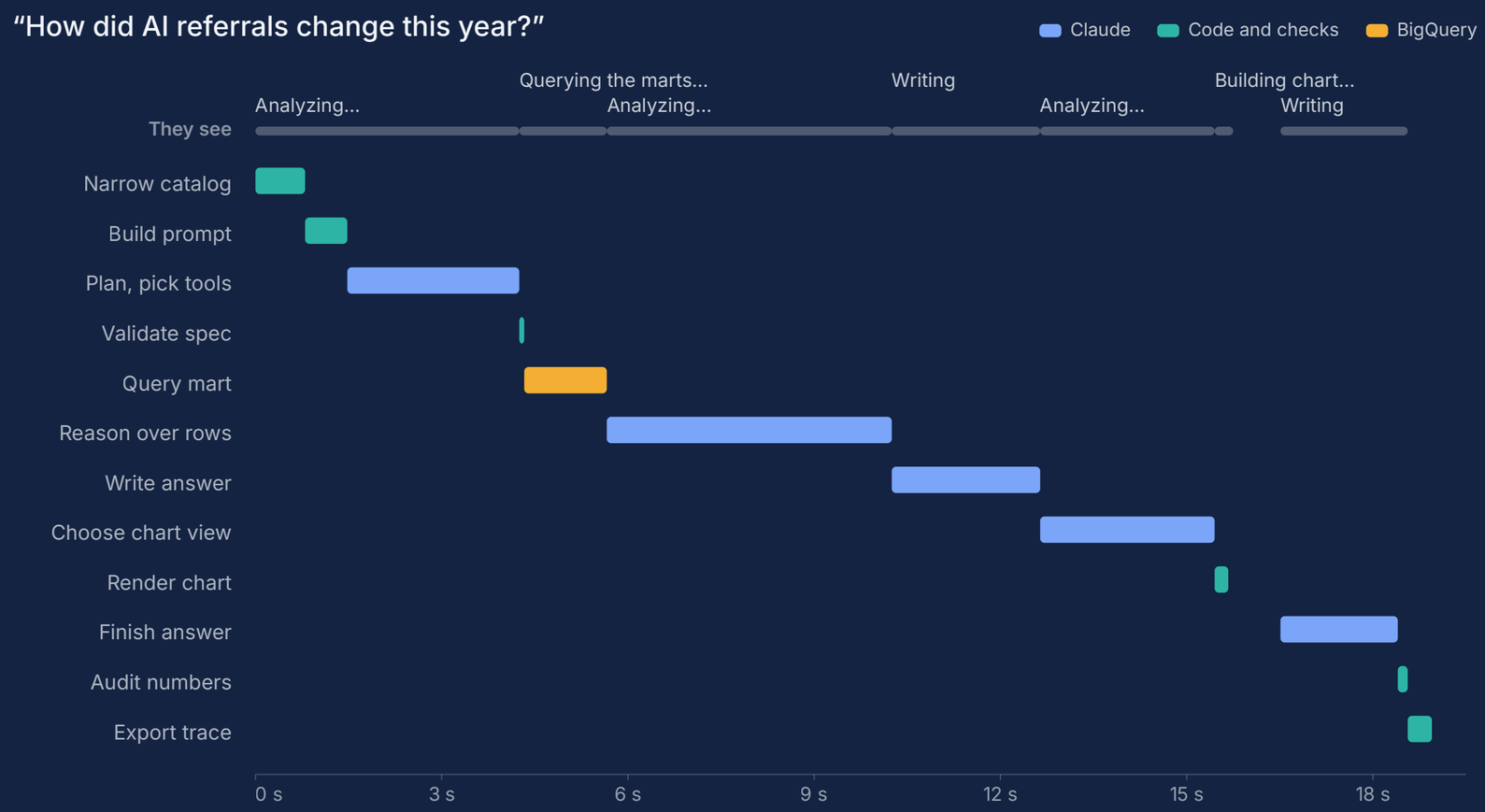
PROCESS BOOK · 2026

Data Lab

Process book



HERO-GODSWAY ZILEVU
UX design · Data & SQL · Applied AI
herogodsway.com

Contents

01 | The bottleneck

Overview 03
Who asks, who reviews 04
Where Data Lab sits 05
Agent architecture 06

02 | Database foundation

From dbt schema to live catalog 07
The agent block 08
The catalog 09
How the catalog stays true 10
Clarify, then refuse 11

03 | Verified math

A question becomes a number 12
How Claude answers 13
Specs that fail closed 14
Every number has an address 15
The audit 16
Year over year + CSV 17
Anatomy of a chart 18

04 | Visual workspace

A stage, not a sidebar 19
The workbench in motion 20
Eight chart types 21
Which view leads 22

05 | Visible progress

The state machine 23
The agent mark 24
Where the mark came from 25
The mark's states 26
One turn, close 27
Problem states 28

06 | Filters & permissions

Pins and shelves in your access 29
What Claude sees 30
See, then touch 31
Pinned, then outside access 32

07 | Sharing & testing

Share across the org 33
In-product feedback 34
Evals 35
Running in production 36

08 | Reflection

The agent team 37
Reflection 38
On every screen 39
Sources 40

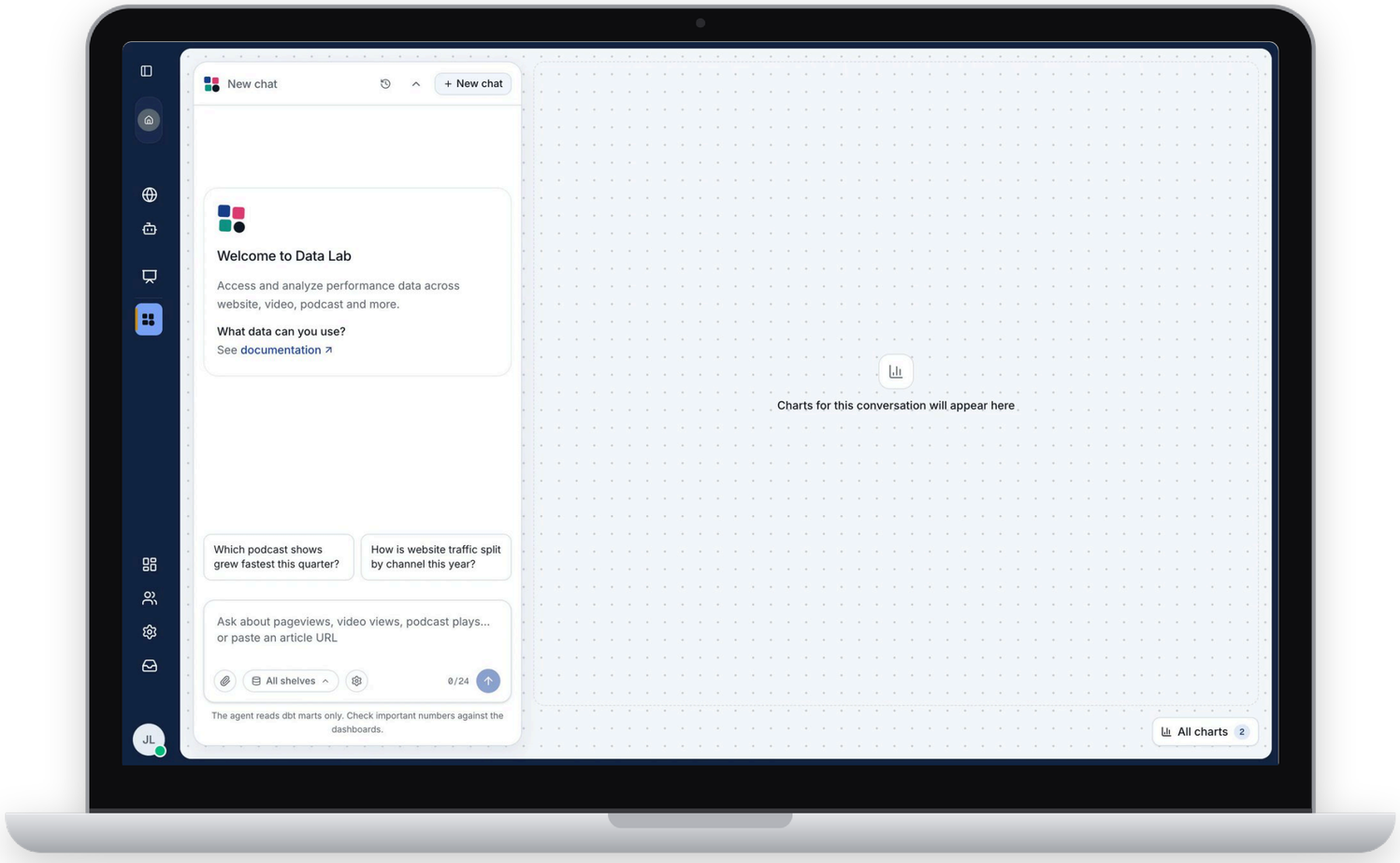
01 | The bottleneck

OVERVIEW

Our dashboards gave teams great daily headlines, but every good chart sparks a follow-up question. Getting that next answer meant waiting days for a custom SQL pull—yet plugging a generic AI chatbot into our database produced messy queries and unreliable numbers. I designed and built Data Lab: a self-serve analytics workspace where plain-English questions return verified BigQuery charts in seconds.

THE WELCOME PAGE · THE FINISHED PRODUCT WITH INVENTED DATA

COVER: THE TRACE OF ONE QUESTION, FROM THE CASE PAGE'S HEADER · TIMINGS INVENTED



Discovery & research

Overview

01 | The bottleneck

THE BOTTLENECK

Every chart sparks a next question.

Why static dashboards leave teams waiting.

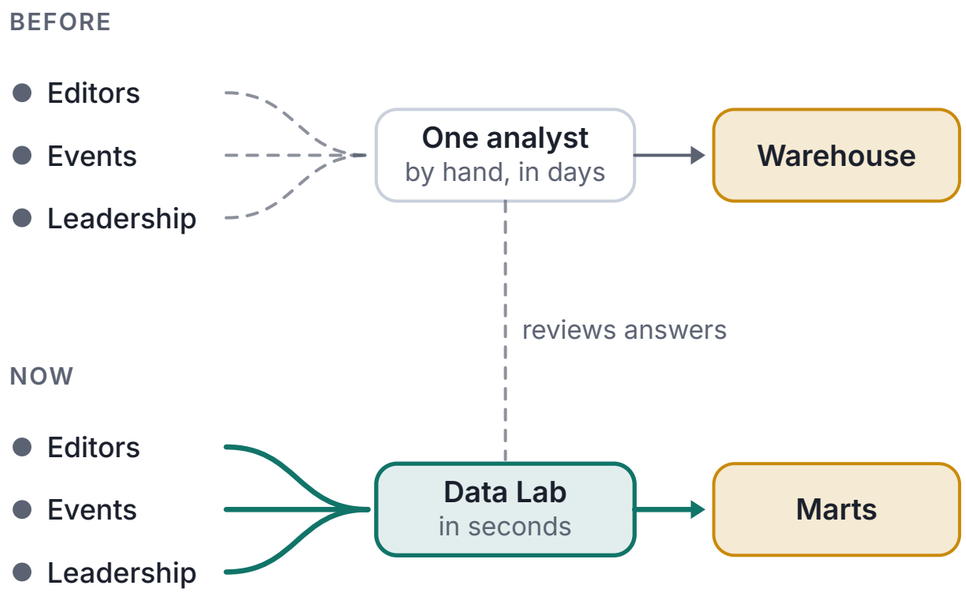
Process

Every dashboard answers the questions it was built for. The next one—this month against last year, one podcast on its own, which assistant sent the most visits—came back to me, asking for more nuance, a clarification or another chart.

TASK ANALYSIS · GULF OF EXECUTION

Designing around the questions people actually asked, not hypothetical ones, kept the scope honest.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE



Discovery & research

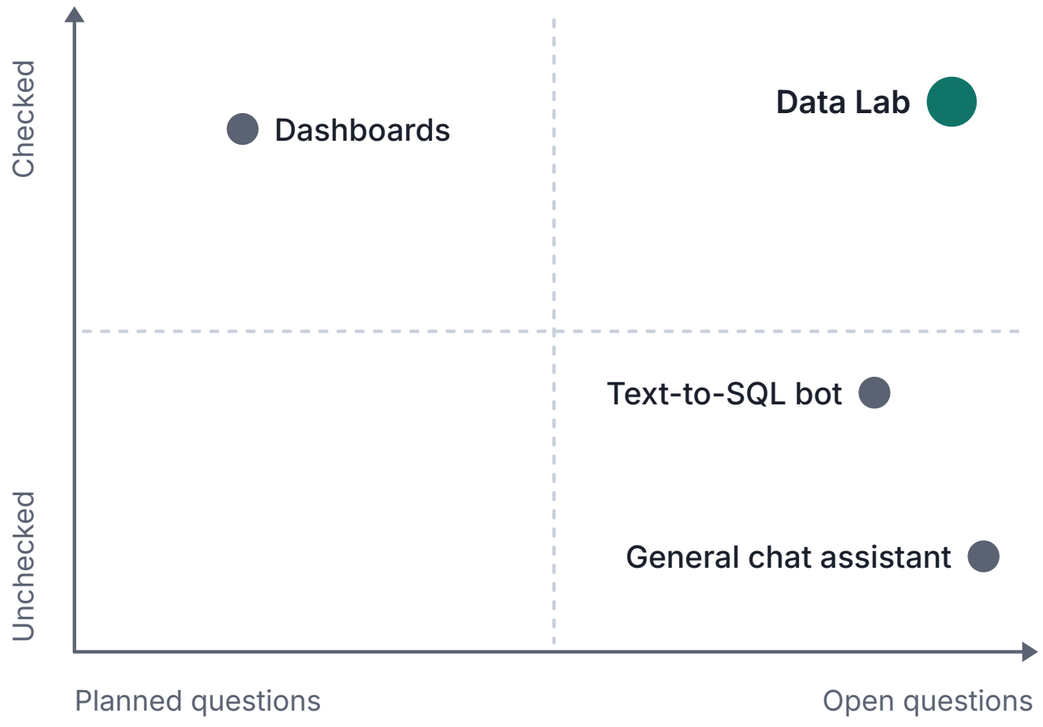
Who asks, who reviews

01 | The bottleneck

WHERE DATA LAB SITS

Process
A dashboard can’t anticipate every angle; a chatbot answers every angle but can’t be checked. Data Lab sits between the two.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE

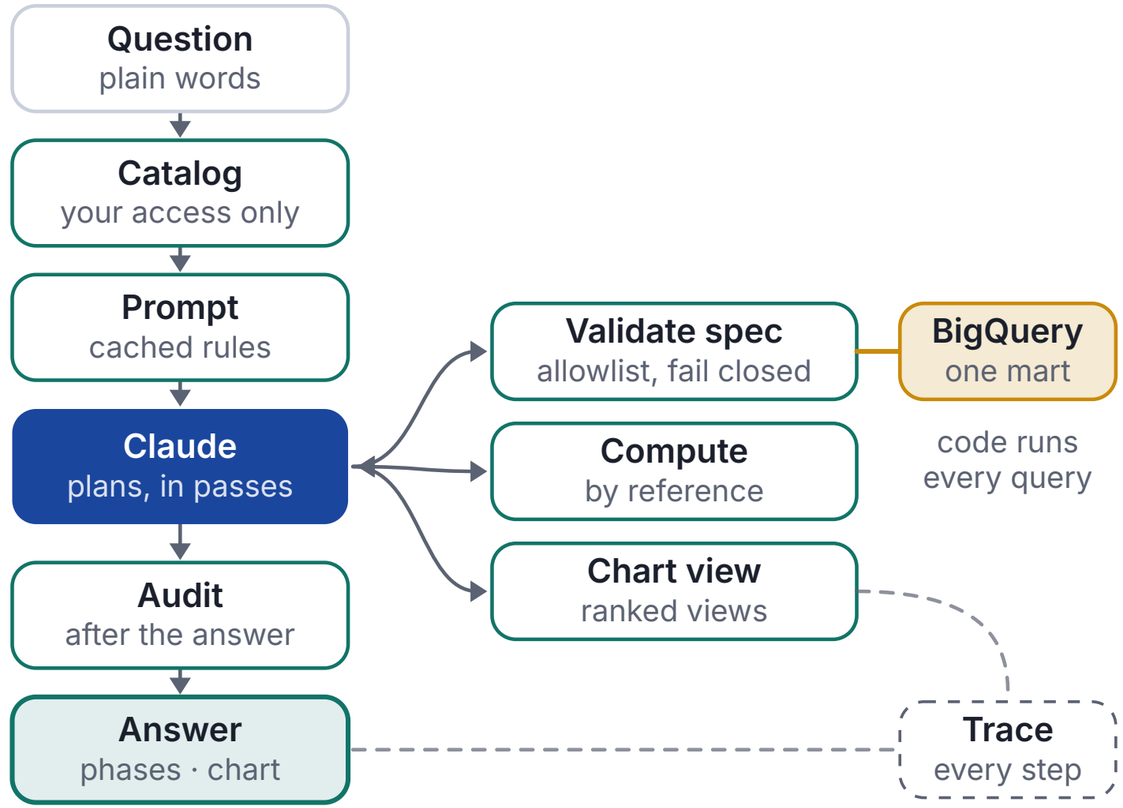


01 | The bottleneck

AGENT ARCHITECTURE

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE

Claude Code and checks BigQuery



02 | Database foundation

DATABASE FOUNDATION

Clean tables before clever prompts.

Structuring BigQuery and dbt for AI.

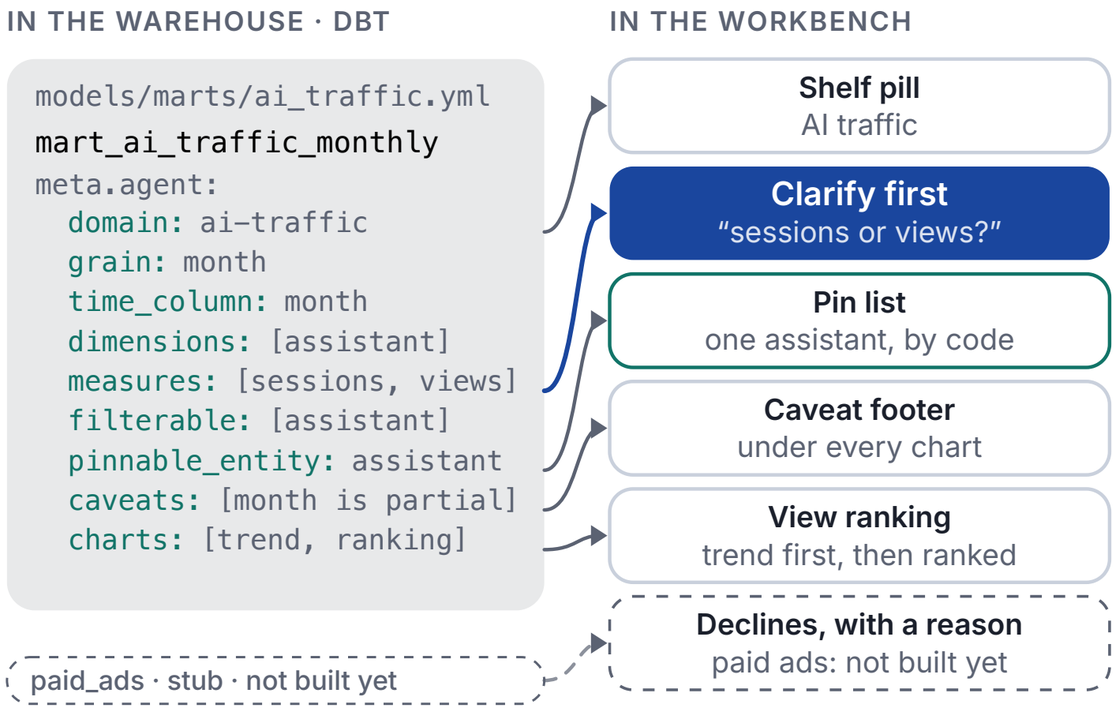
Process

Pointing a model at raw tables invites broken joins and double-counted rows. Before the interface, I modeled clean, pre-joined marts in BigQuery with dbt.

SINGLE SOURCE OF TRUTH · INFORMATION ARCHITECTURE

The schema is the one place a metric is defined, so dashboards and answers cannot drift apart.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE



The catalog is generated from these blocks at build time.
Nothing about a mart is written by hand.

02 | Database foundation

THE AGENT BLOCK

Process
Each mart's schema carries an agent block—grain, measures, caveats, pinnable entities—so the catalog Claude reads is generated from the warehouse itself.

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE · VALUES INVENTED

IN THE WAREHOUSE · DBT

```
models/marts/ai_traffic.yml
- name: mart_ai_traffic_monthly
  meta:
    agent:
      domain: ai-traffic
      dashboard_slug: ai-traffic
      tier: 2
      grain: month
      time_column: month
      dimensions: [assistant]
      measures: [sessions, views]
      filterable: [assistant]
      caveats: [month is partial]
      charts: [trend, ranking]
      pinnable_entity: assistant
```

WHAT EACH KEY DRIVES

- `domain: ai-traffic` — the shelf it sits on
- `dashboard_slug: ai-traffic` — access: seen only by people who can view this dashboard
- `tier: 2` — Claude may query it through a checked spec
- `grain: month` — what one row means
- `time_column: month` — the column a time window filters
- `dimensions: [assistant]` — the only columns a spec may group by
- `measures: [sessions, views]` — the only columns a spec may add up or average
- `filterable: [assistant]` — the only columns a filter may use
- `caveats: [month is partial]` — the footer under every chart from this mart
- `charts: [trend, ranking]` — declared views; the ranking picks among them
- `pinnable_entity: assistant` — the pin list in the composer

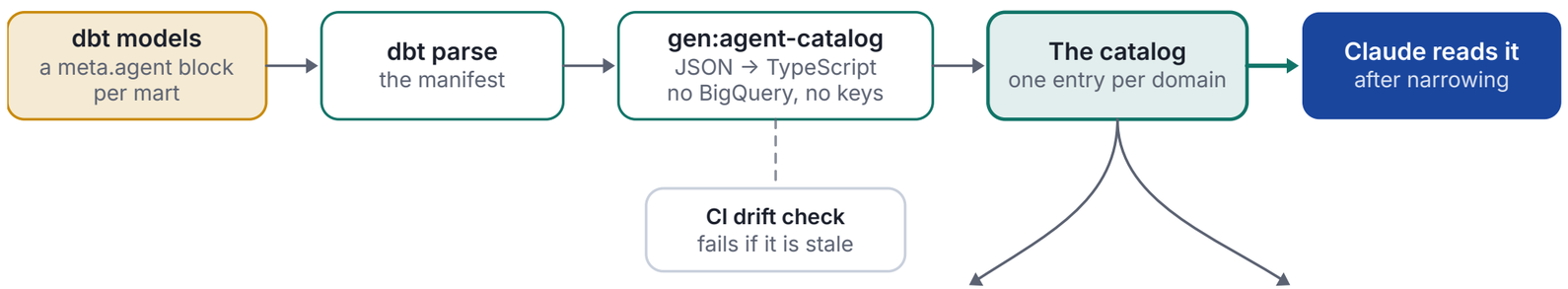
keys as in the product · values invented, as on the case page

02 | Database foundation

THE CATALOG

Process
When a definition changes, the agent learns it the same day, no prompt to rewrite.

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE



Run by hand after a dbt change:

```
cd dbt && dbt parse
cd portal && npm run gen:agent-catalog
```

Available
`status: available`

marts, grain, dimensions, measures, caveats, views

Unavailable · a stub
`status: unavailable`

social · pipeline not yet built

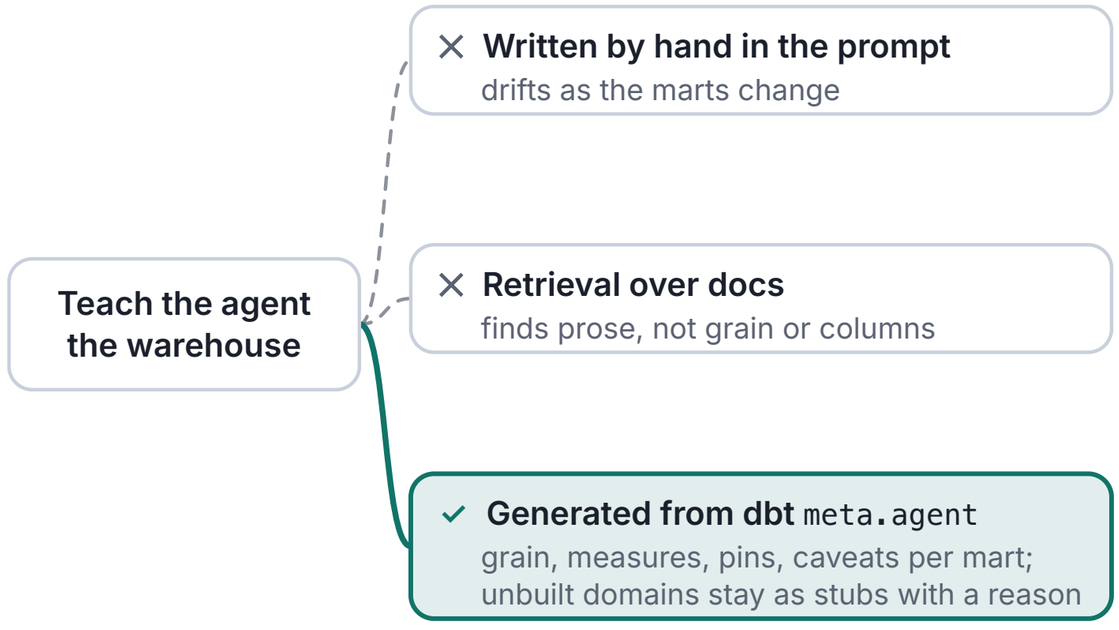
paid ads · not built yet

Claude declines with the reason

02 | Database foundation

HOW THE CATALOG STAYS TRUE

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE

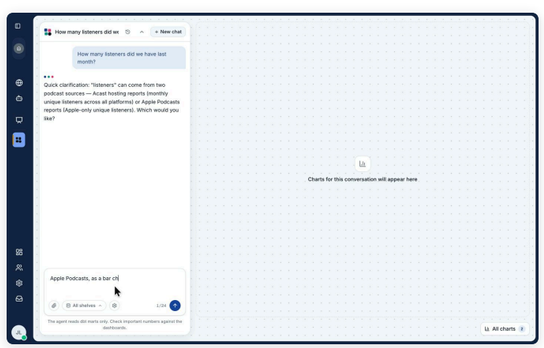


02 | Database foundation

CLARIFY, THEN REFUSE

FRAMES FROM THE CASE PAGE'S MOTION; BELOW, ITS X-RAY NOTES

Claude Code and checks BigQuery



Clarify 0 s

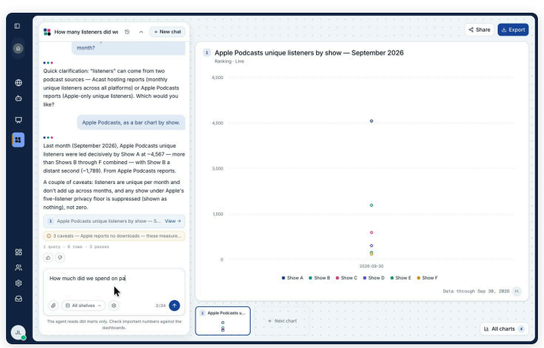
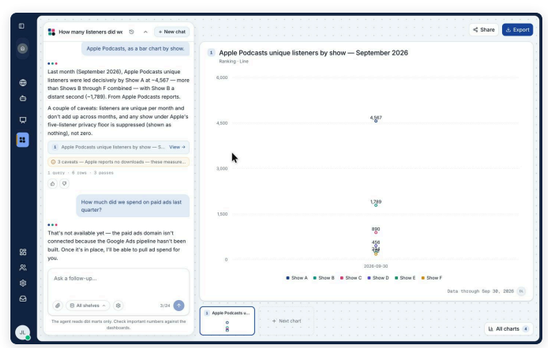


Chart 17.8 s



Refuse 29.7 s

TIME	CHAPTER	BESIDE THE CHAT	ON THE STAGE
0 s	**Clarify**	"listeners" matches two measures in the catalog · Claude asks one question instead of guessing	nothing runs until the person answers
8.3 s	**Reply**	the reply picks the measure: Apple Podcasts, by show	
12.7 s	**Query**	query_mart → Apple Podcasts listeners by show, last month · one mart, validated	phase: pulling
17.8 s	**Chart**	Claude answers from the rows · caveats from the mart's meta.agent block	render_chart · bar by show · source and date stamped
29.7 s	**Refuse**	paid ads: a stub in the catalog, no mart yet · Claude gives the stub's reason, exactly	no query runs

03 | Verified math

VERIFIED MATH

The AI translates; code calculates.

Zero raw SQL and zero mental math.

Process
Language models read intent well and do arithmetic badly, so the work is split.

ERROR PREVENTION · VERIFIABLE PROVENANCE
Making wrong numbers impossible to produce beats asking people to spot them.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE

1 The request

Claude writes a spec, never SQL.

```
query_mart({
mart: "ai_traffic_monthly",
select: ["month", "sessions"],
filters: [month >= "2026-01"] })
```

validated against the allowlist · fails closed

2 Query, compute

Code runs the query; arithmetic by reference.

```
SELECT month, sessions
FROM mart_ai_traffic_monthly
WHERE month >= @from
```

BigQuery · 10 rows

```
compute(pct_change, 29227, 17753) → 0.646
```

3 The audit

After the answer, each numeral is traced. Misses count toward a badge earned under 2%

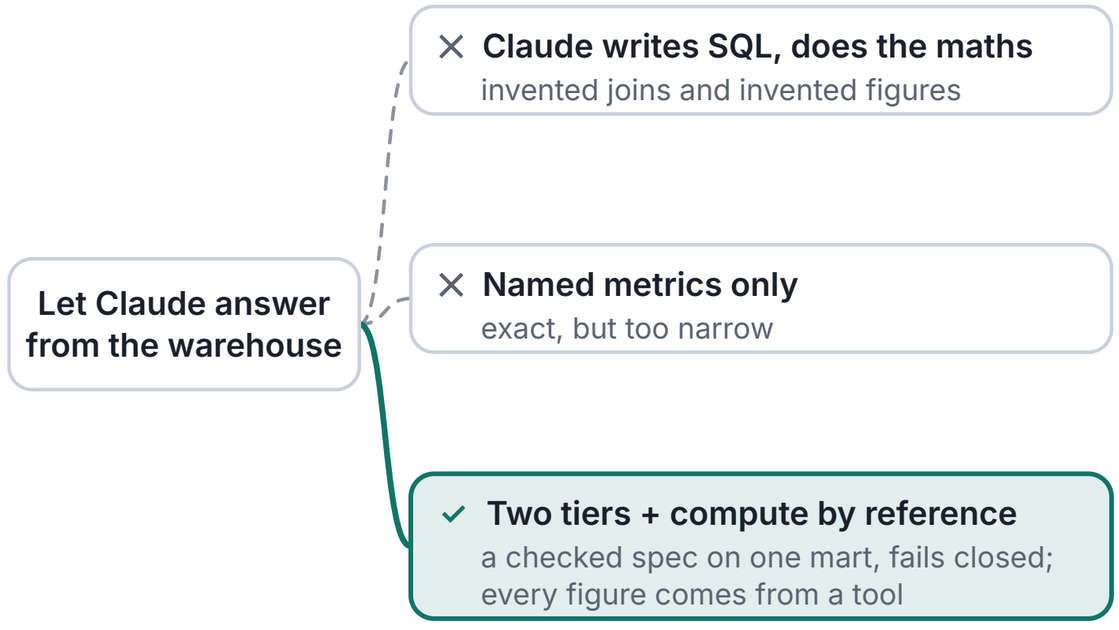
AI referrals are up 65% from January (17,753) to September (29,227).

```
17,753 → row 1 · 29,227 → row 9
65% → compute · 3 of 3 matched
```

03 | Verified math

HOW CLAUDE ANSWERS

A CASE-PAGE DRAWING KEPT FOR THIS BOOK



03 | Verified math

SPECS THAT FAIL CLOSED

Process
Claude turns a question into a structured request against one mart; code validates it, runs the query and computes every change and share by reference—the model never writes SQL and never adds up.

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

A SPEC THAT PASSES

```
query_mart({
  mart: "ai_traffic_monthly",
  select: ["month", "ai_sessions"],
  filters: [{ column: "month",
              op: ">=", value: "2026-01" }]
})
```

- ✓ one Tier-2 mart, in a domain you can see
- ✓ every column belongs to that mart
- ✓ filters use filterable columns and allowed operators
- ✓ values are bound parameters, never pasted into SQL
- ✓ no joins, no subqueries
- ✓ an unknown key is rejected, never stripped

TWO SPECS THAT FAIL CLOSED

```
select: ["month", "revenue"]
```

```
× select: 'revenue' is not a column
   of mart 'ai_traffic_monthly'
```

```
aggregations: [{ fn: "sum",
  column: "share_of_site_sessions" }]
```

```
× aggregation: 'share_of_site_sessions'
   is non-additive (a share/rate) and cannot
   be summed; use avg/min/max
```

the first failed rule wins · its reason names the field
nothing runs · Claude reads the reason

THE ALLOWLIST

Filter operators `=` `!=` `<` `<=` `>` `>=` `in` `between`

Aggregations `sum` `avg` `min` `max` `count`

Rows 100 by default · 1,000 at most

03 | Verified math

EVERY NUMBER HAS AN ADDRESS

DRAWN FOR THIS CASE STUDY FROM THE CASE PAGE'S INTERACTIVE FIGURE · INVENTED ROWS

AI referrals have grown sharply this year — up 65% from January (17,753 sessions) to September (29,227). October (30,455) is only a partial month.

1 query · 10 rows · 5 passes · Data from Jan 2026 to Oct 2026

amber a row of the mart · **teal** a computation

SRC_1 · THE AI TRAFFIC MART BY MONTH

row	month	ai_sessions
r1	2026-01	17,753
r2	2026-02	18,827
r3	2026-03	20,343
r4	2026-04	22,041
r5	2026-05	23,126
r6	2026-06	22,595
r7	2026-07	24,262
r8	2026-08	25,935
r9	2026-09	29,227
r10	2026-10	30,455

r10 · a partial month → the caveat

COMPUTE, BY REFERENCE

```
compute({
  operation: "pct_change",
  operands: [
    { source_id: "src_1", column: "ai_sessions",
      row: { column: "month", equals: "2026-09" } },
    { source_id: "src_1", column: "ai_sessions",
      row: { column: "month", equals: "2026-01" } } ]
})
```

→ 0.646 a 0–1 fraction; the answer says 65%

Operations add · subtract · multiply · divide · pct_change · share

Aggregates sum · avg · min · max · count

operands: references to this turn's results, never typed numbers · a zero baseline is refused

03 | Verified math

THE AUDIT

Process
An audit then checks each numeral in the answer against the tool results.

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

AFTER THE ANSWER

1 Find every numeral the answer quotes

```
\$?\d{1,3}(?:,\d{3})+(?:\.\d+)?%?|\$?\d+(?:\.\d+)?%?
```

2 Match each to a number a tool returned this turn

65% → 0.646 as a percentage, at the answer's precision

17,753 → row r1, ai_sessions

29,227 → row r9, ai_sessions

30,455 → row r10, ai_sessions

3 Record the result

```
checked 4 · unmatched 0
```

an unmatched numeral → a warning on the turn's trace
shadow mode: nothing changes for the reader

What counts as a match

- rounded to the answer's precision
- a 0–1 fraction read as a percentage
- the sign dropped in “down 12%”
- date parts
- about 1% slack on round figures
- small whole numbers, 0 to 12, are skipped
- links are removed first

The gate

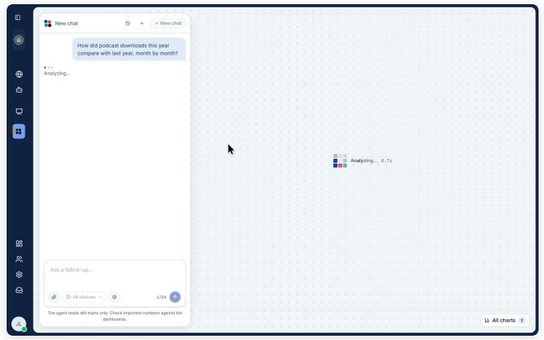
a visible “figures verified” badge ships only if false alarms stay under about 2%

03 | Verified math

YEAR OVER YEAR + CSV

FRAMES FROM THE CASE PAGE'S MOTION; BELOW, ITS X-RAY NOTES

Claude Code and checks BigQuery



Ask 0 s



Answer and chart 19.5 s



The rows 42.3 s

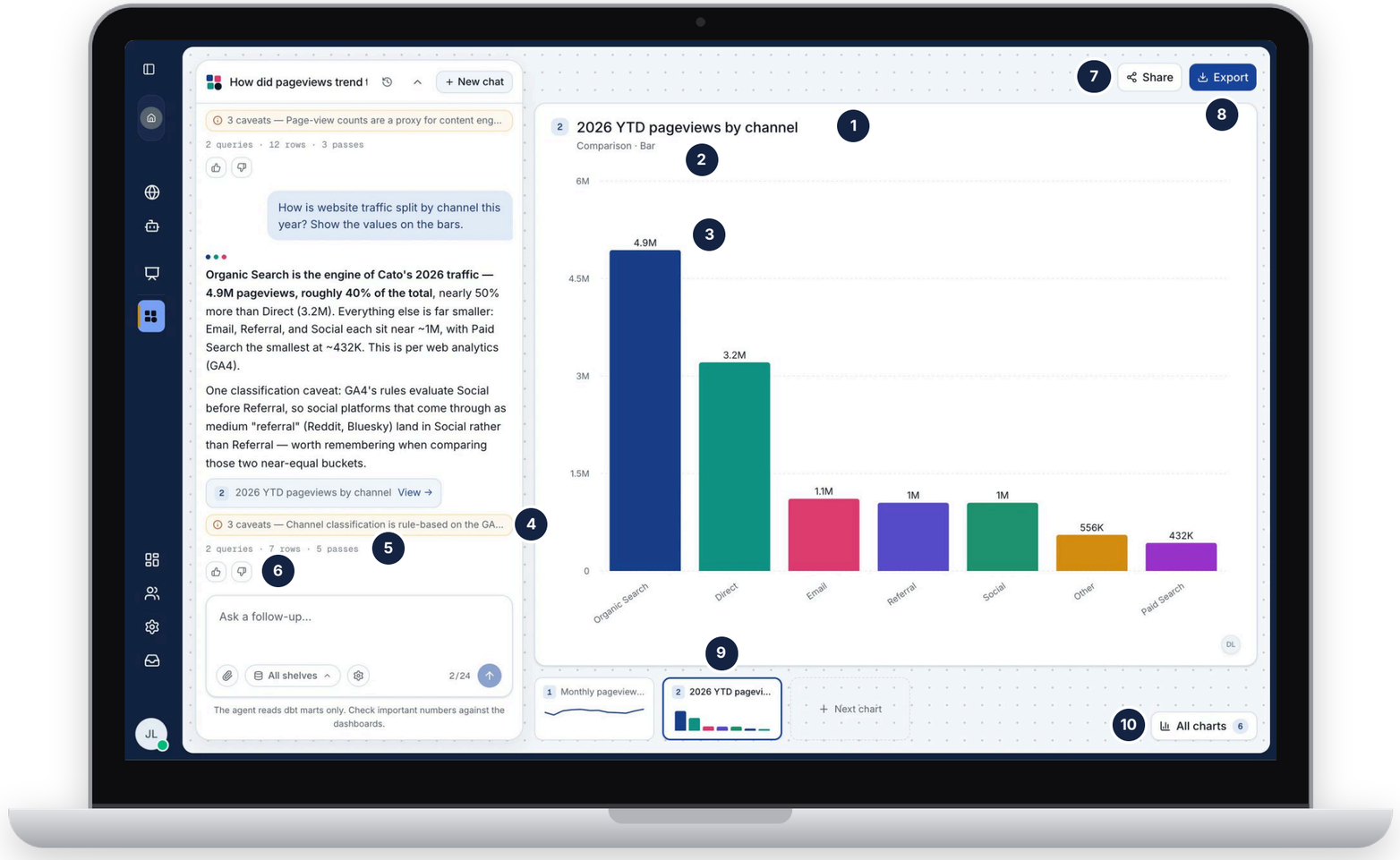
TIME	CHAPTER	BESIDE THE CHAT	ON THE STAGE
0 s	**Ask**	catalog: the podcast marts in your access · prompt = cached rules + your question	stage empty
8.1 s	**Query**	query_mart → podcast downloads by month, this year and last · spec validated, fail closed	phase: pulling
9.5 s	**Query**	Claude reasons over the rows · each change is compute(pct_change), by reference	phase: analyzing
19.5 s	**Answer and chart**	Claude writes from the rows · the audit traces every numeral to a row or a compute	render_chart · year over year · source and data date stamped by the server
27.7 s	**Ask for a CSV**	a download is asked for · an export takes only this turn's results	the chart stays
37.6 s	**Ask for a CSV**	query_mart again, this turn · the same mart, the same spec	
42.3 s	**The rows**	export_csv(source_id) → the rows behind the chart · a time-limited link	the CSV holds the rows the chart drew

03 | Verified math

ANATOMY OF A CHART

1. A title for this answer
2. The view it chose
3. Values on the bars
4. Caveats from the mart
5. Queries · rows · passes
6. Rate the answer
7. Share with the org
8. Export PNG or CSV
9. The thread's filmstrip
10. Every chart, kept

THE FINISHED PRODUCT WITH INVENTED DATA · MARKS AS ON THE CASE PAGE



04 | Visual workspace

VISUAL WORKSPACE

Room to see the story.

A split canvas designed for charts.

Process
Most AI tools squeeze answers into a narrow chat sidebar, which crushes axis labels and buries earlier charts as you scroll. Because data analysis is visual, I designed a wide split workspace: conversation on the left, and a dedicated chart stage on the right.

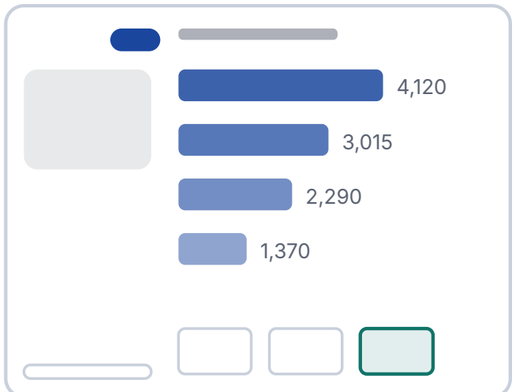
SPLIT-ATTENTION EFFECT · VISUAL COMPARISON
Words beside a persistent chart stage let people compare instead of scroll.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE · INVENTED VALUES

THE FIRST VERSION · JUNE



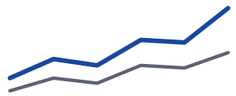
THE WORKBENCH · JULY



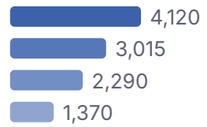
THE VIEW THE RANKING PICKS



Trend
line, year marks



Comparison
this year vs last



Ranking
bars with values



Share
donut, as shares

Spatial UX & data viz

A stage, not a sidebar

04 | Visual workspace

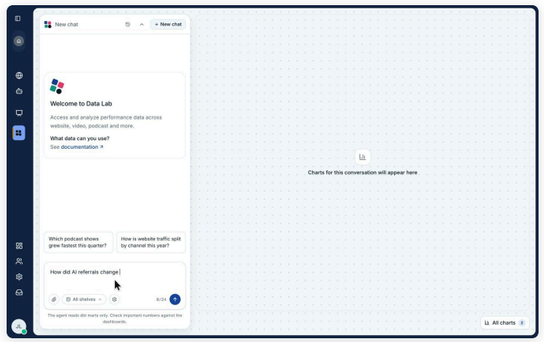
THE WORKBENCH IN MOTION

Process

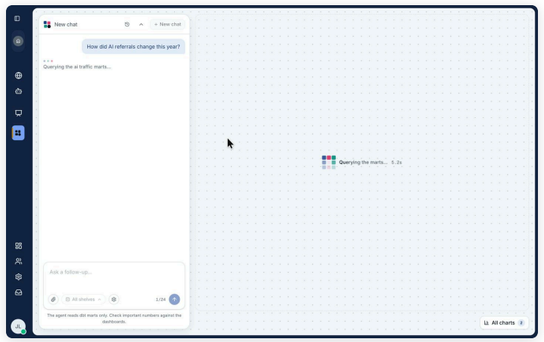
A bottom filmstrip keeps every chart from the session one click away.

FRAMES FROM THE CASE PAGE'S MOTION; BELOW, ITS X-RAY NOTES

Claude Code and checks BigQuery



Ask 0 s



Query 10.5 s



Chart 21.9 s

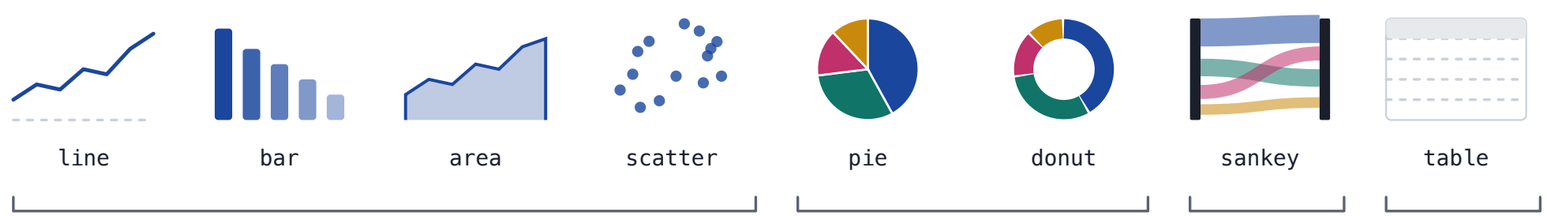
TIME	CHAPTER	BESIDE THE CHAT	ON THE STAGE
0 s	**Ask**	catalog narrowed to your access · shelf: all · 0/24 prompts	stage empty · this thread's charts land here
6.3 s	**Plan**	Claude plans, picks tools · prompt = cached rules + your question	phase: analyzing · pixel grid, timer
10.5 s	**Query**	query_mart → validate (allowlist, fail closed) → BigQuery · one mart, bound params	phase: pulling · ai-traffic
12 s	**Reason**	Claude reasons over 10 rows · arithmetic by compute(), by reference	phase: analyzing
16.5 s	**Answer**	stream: text deltas · every number comes from a tool result	render_chart · view: your request → views people liked → catalog default
21.9 s	**Chart**	done · caveats from the mart's dbt meta.agent block · audit traces every numeral · trace exported	chart · source and data date stamped by the server · filmstrip

04 | Visual workspace

EIGHT CHART TYPES

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

THE EIGHT CHART TYPES



in the code: `a learned “line” must never restyle a sankey`

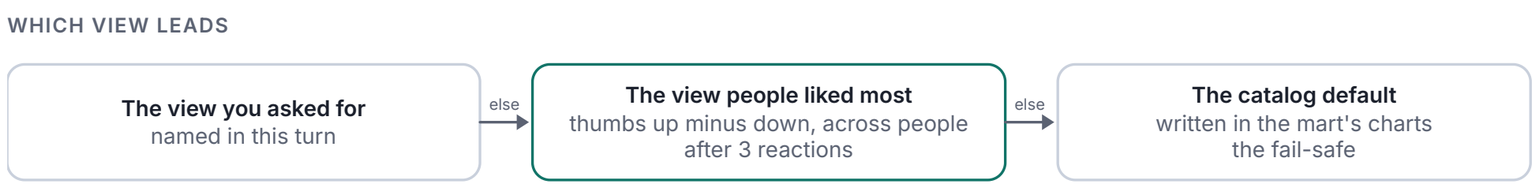
the first version, in June: line, bar, area · today: eight

04 | Visual workspace

WHICH VIEW LEADS

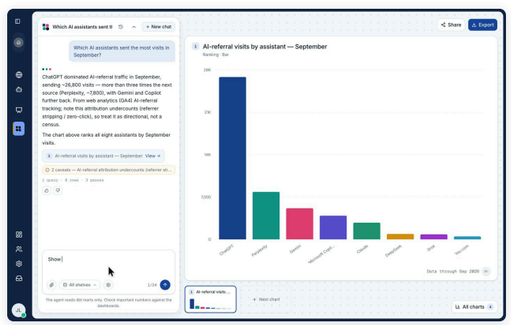
DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

BELOW: FRAMES FROM “RESTYLE, THEN RATE”

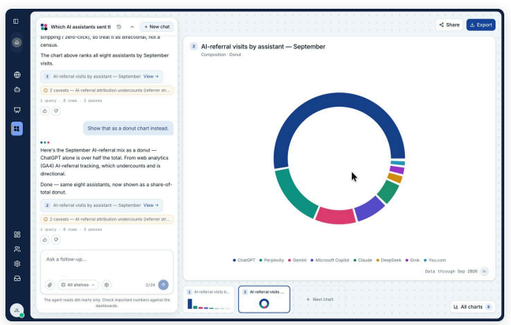


`INV-6:` personalize style, never substance.

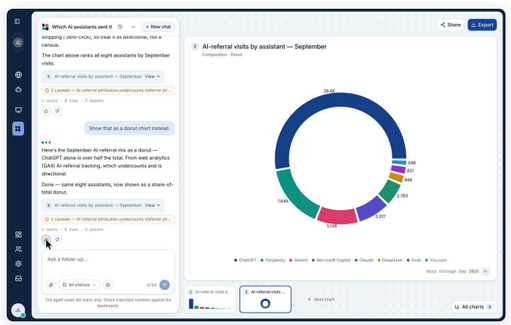
re-ranking changes which declared view leads · never the data, never the number quoted



The bar chart



Donut



Rate

05 | Visible progress

VISIBLE PROGRESS

Showing the work backstage.

Turning database wait time into trust.

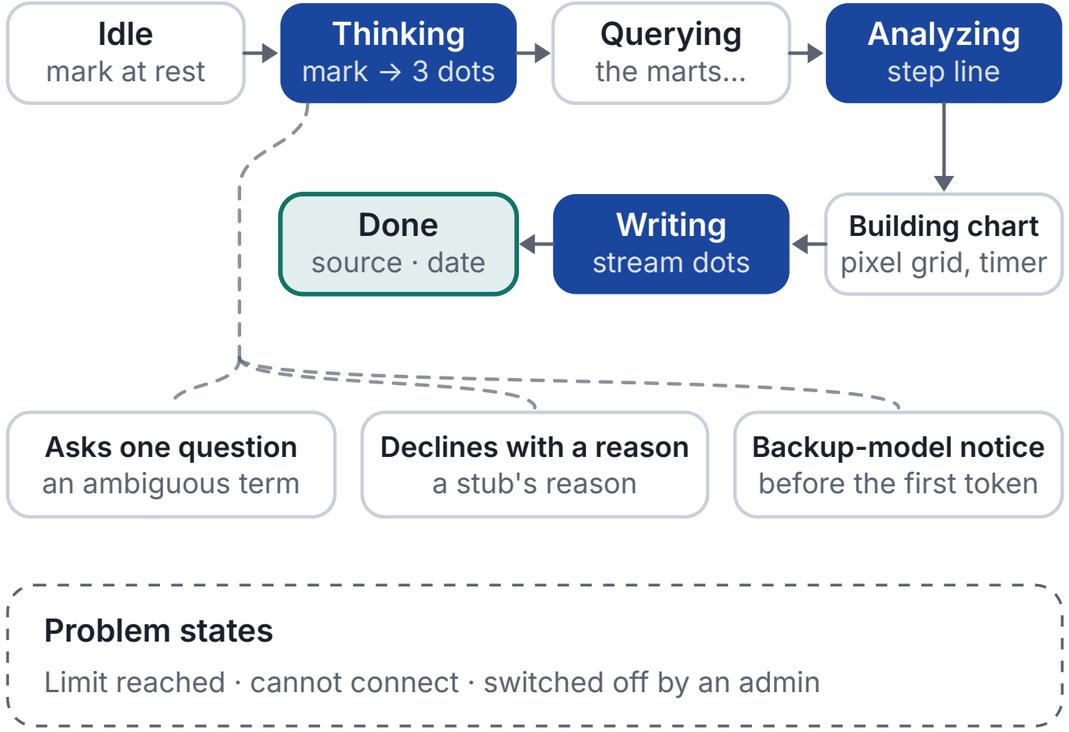
Process

Pulling live BigQuery data, running calculations, and verifying every figure takes 15 to 20 seconds. A generic loading spinner makes that pause feel like a freeze.

VISIBILITY OF SYSTEM STATUS · PERCEIVED PERFORMANCE

Clear, step-by-step progress cues reduce the felt wait and show that the answer is being checked.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE



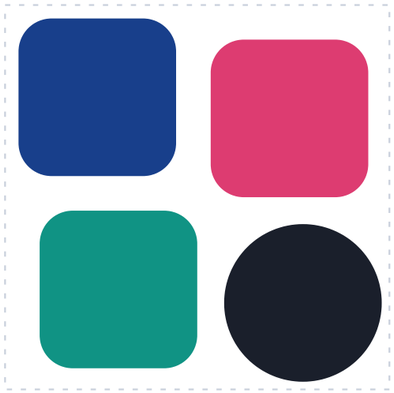
05 | Visible progress

THE AGENT MARK

Process
I designed an animated agent mark that narrates each live step in plain language—showing when the AI is planning, which mart is being queried, and when the chart is drawing.

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S BRAND CODE

TAKEN APART



q1 · blue
at 3.5, 3.5 on a 100 grid

q2 · rose
at 53.5, 9 on a 100 grid

q3 · teal
at 9, 53.5 on a 100 grid

q4 · ink, round
at 57, 57 on a 100 grid

each cell 41 wide, corners 21 %

SIZES



on dark: lighter cells, a white round cell

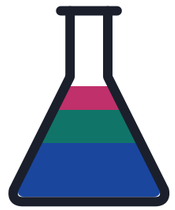


05 | Visible progress

WHERE THE MARK CAME FROM

PROCESS NOTE · HERO WRITES
Why the agent mark moved from a flask to a petri dish to four cells.

REDRAWN FROM THE PRODUCT'S HISTORY

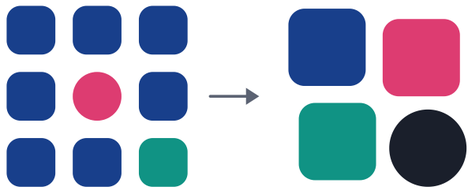


1 · A flask
three data colours as liquid



pulling analyzing

2 · A petri dish
samples scatter, then sort into three



3 · Four cells
the agent mark, grown from the product mark

05 | Visible progress

THE MARK'S STATES

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE



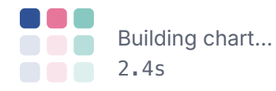
Rest
a slow orbit
`6.4 s a turn`



Thinking
folds into three dots
`0.45 s fold · 1.1 s bounce`



Writing
stream dots above the answer
`1.1 s, 0.15 s apart`



Building chart
pixel grid and a timer
`950 ms orbit`

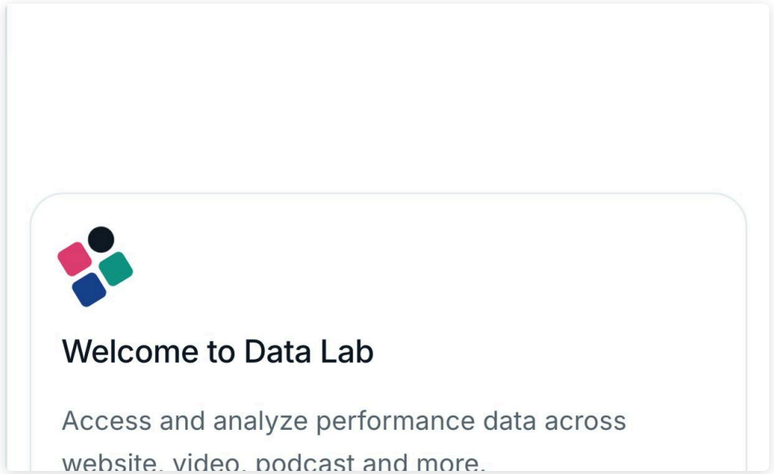
reduced motion: the mark breathes instead, once every 2.4 s

PHASE		WHAT THE PERSON READS	WHEN
`analyzing`	→	**Analyzing...**	the first phase, and every compute
`pulling`	→	**Querying the marts...**	named when one shelf is in scope: “Querying the AI traffic marts...”
`rendering`	→	**Building chart...**	on the stage, with the pixel grid
`text streams`	→	● ● ●	stream dots mark the turn; no words

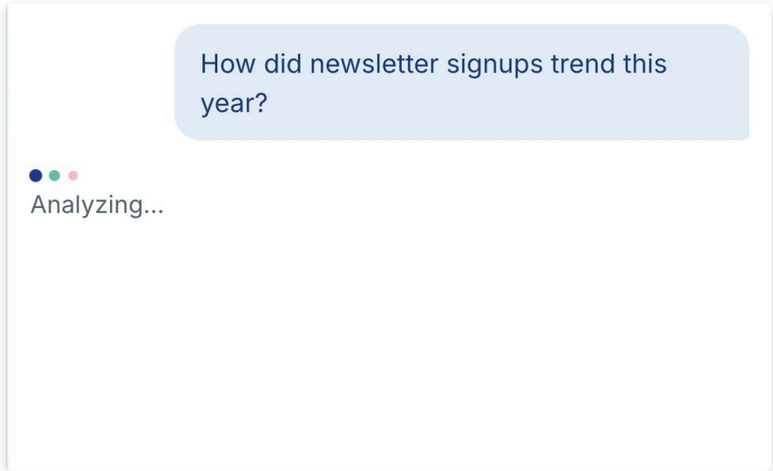
05 | Visible progress

ONE TURN, CLOSE

FRAMES FROM THE CASE PAGE'S MOTION, CLOSE CROP



Mark



Analyzing



Querying



Writing

HCI & system feedback

One turn, close

05 | Visible progress

PROBLEM STATES

THE PRODUCT'S OWN WORDS, DRAWN FOR THIS CASE STUDY

LIMIT REACHED

This conversation reached its 24-prompt limit.
Start a new conversation to keep going. New chat

TOO MANY AT ONCE

That's a lot of questions at once — try again in ⟨n⟩s.

TODAY'S BUDGET

You've reached today's assistant usage limit.
It resets at midnight (Eastern).

CANNOT CONNECT

Couldn't reach the assistant.
Check your connection and try again.

SIGNED OUT

Your session has expired — please sign in again.

ONE DOMAIN SWITCHED OFF

temporarily disabled for maintenance —
check back shortly

A BACKUP MODEL ANSWERED

Answered by a backup model while the primary
was briefly unavailable.

NOTHING CAME BACK

No answer came back for that. Try asking again.

06 | Filters & permissions

FILTERS & PERMISSIONS

Guided by clicks, not prompts.

One-click filters and smart boundaries.

Process

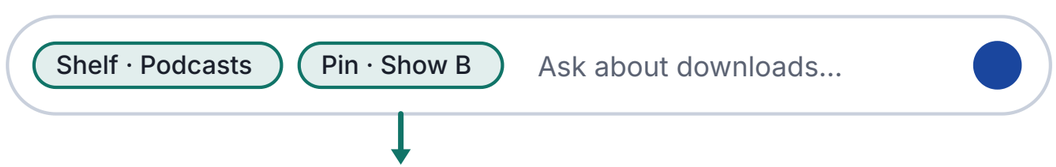
Teammates shouldn’t have to write long, technical prompts just to filter a chart. I added one-click filter pills that lock a specific topic or show onto every query in the thread.

RECOGNITION RATHER THAN RECALL · LEAST PRIVILEGE

Clickable pins replace memory-heavy typing, and a catalog narrowed to each person’s access stops a question before it runs.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE

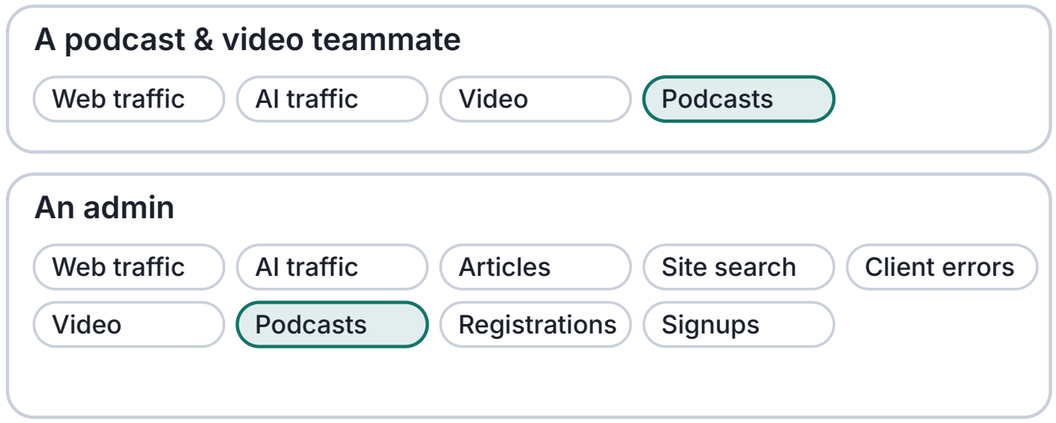
WHAT A PIN BECOMES




```
filters += { column: "display_name", value: "Show B" }
```

added by code to every turn in the thread · never by prose

WHAT THE SHELF PICKER LISTS



Shelves outside your access are not greyed out. They are not listed.

Interaction & governance

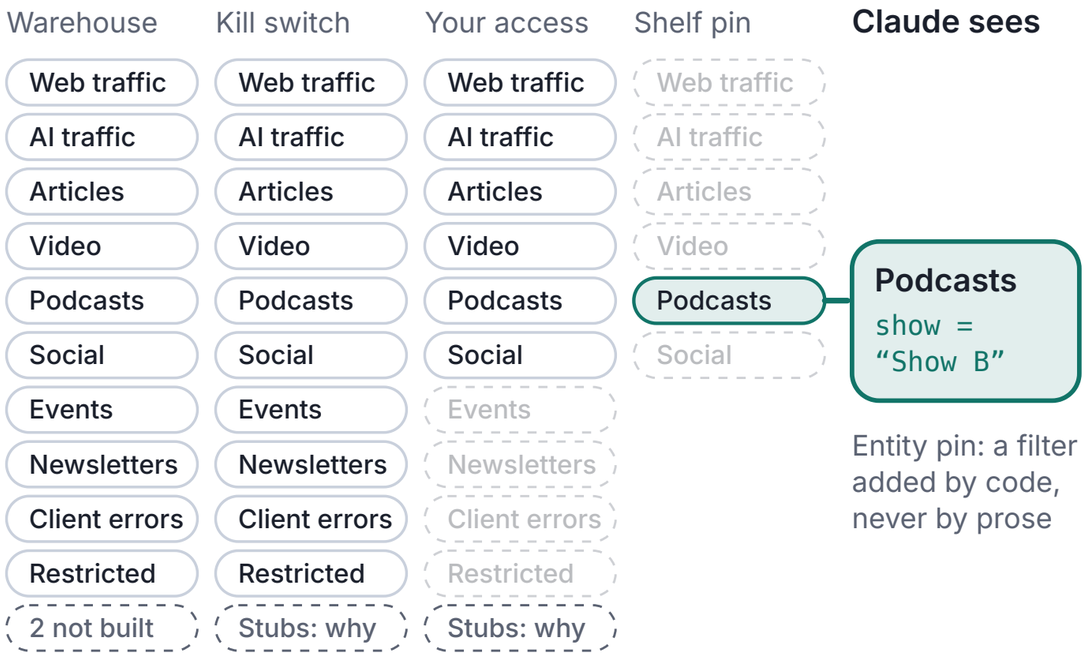
Pins and shelves in your access

06 | Filters & permissions

WHAT CLAUDE SEES

Process
When a term matches two metrics, Data Lab asks one quick clarifying question—and because each person's catalog matches their login permissions, restricted tables stay invisible.

THE CASE PAGE'S OWN DRAWING, EXPORTED FROM THE PAGE



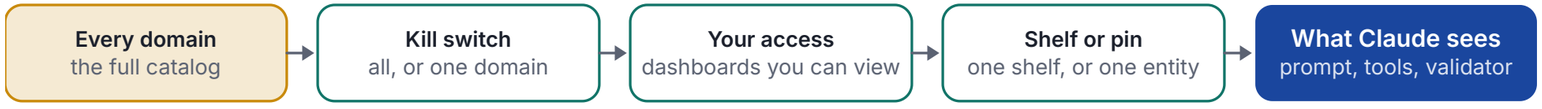
Every tool call checks access again, with the session's user id.

06 | Filters & permissions

SEE, THEN TOUCH

DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

WHAT CLAUDE CAN SEE



outside your access: omitted · not stubbed · not mentioned

the prompt, the tool list and the validator: all built from what is left

WHAT EACH CALL CAN TOUCH

Session	the user id comes from the sign-in, never the prompt
Every query	access checked again → 403, forbidden
A pin	added by code before validation → a 400 that names it if it cannot apply

HOW THE PROMPT IS LAYERED

Rules + the scoped catalog
cached · the same bytes for everyone who shares that scope

cache breakpoint

Entity pin	the pinned show or page
Uploaded context	files added to the thread
Personal context	settings the person chose
The question	this turn

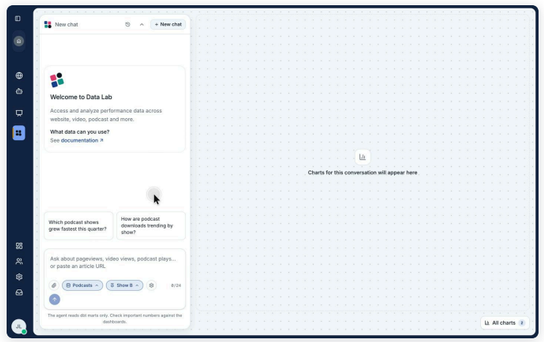
anything a person can influence: after the breakpoint, never in the cached prefix

06 | Filters & permissions

PINNED, THEN OUTSIDE ACCESS

FRAMES FROM THE CASE PAGE'S MOTION; BELOW, ITS X-RAY NOTES

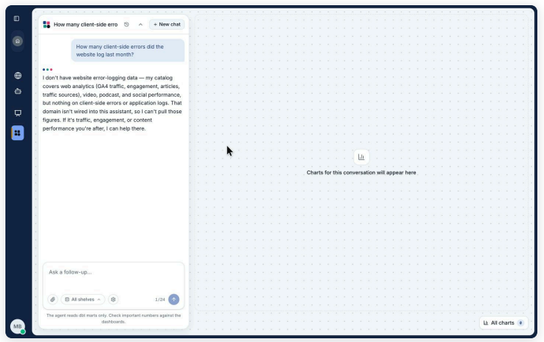
Claude Code and checks BigQuery



Pin a show 0 s



Answer 14.9 s



Outside access 31.2 s

TIME	CHAPTER	BESIDE THE CHAT	ON THE STAGE
0 s	**Pin a show**	shelf: Podcasts, from the shelves in your access · the show list comes from the mart	
1.5 s	**Ask**	entity pin: “Show B” becomes a query filter, added by code, never by prose	
10.7 s	**Query**	query_mart → podcast downloads this year · display_name = “Show B”, added by code	phase: pulling
14.9 s	**Answer**	Claude answers for Show B only · the pin stays on the thread	render_chart · one show
23.8 s	**Another person**	another person, on the podcast and video team · their catalog is narrowed to their access	stage empty
31.2 s	**Outside access**	client errors are not in their catalog, so Claude cannot ask for them · it declines	no query runs

Interaction & governance

Pinned, then outside access

07 | Sharing & testing

SHARING & TESTING

Built to check, share, and learn.

CSV proof, team snapshots, and weekly tests.

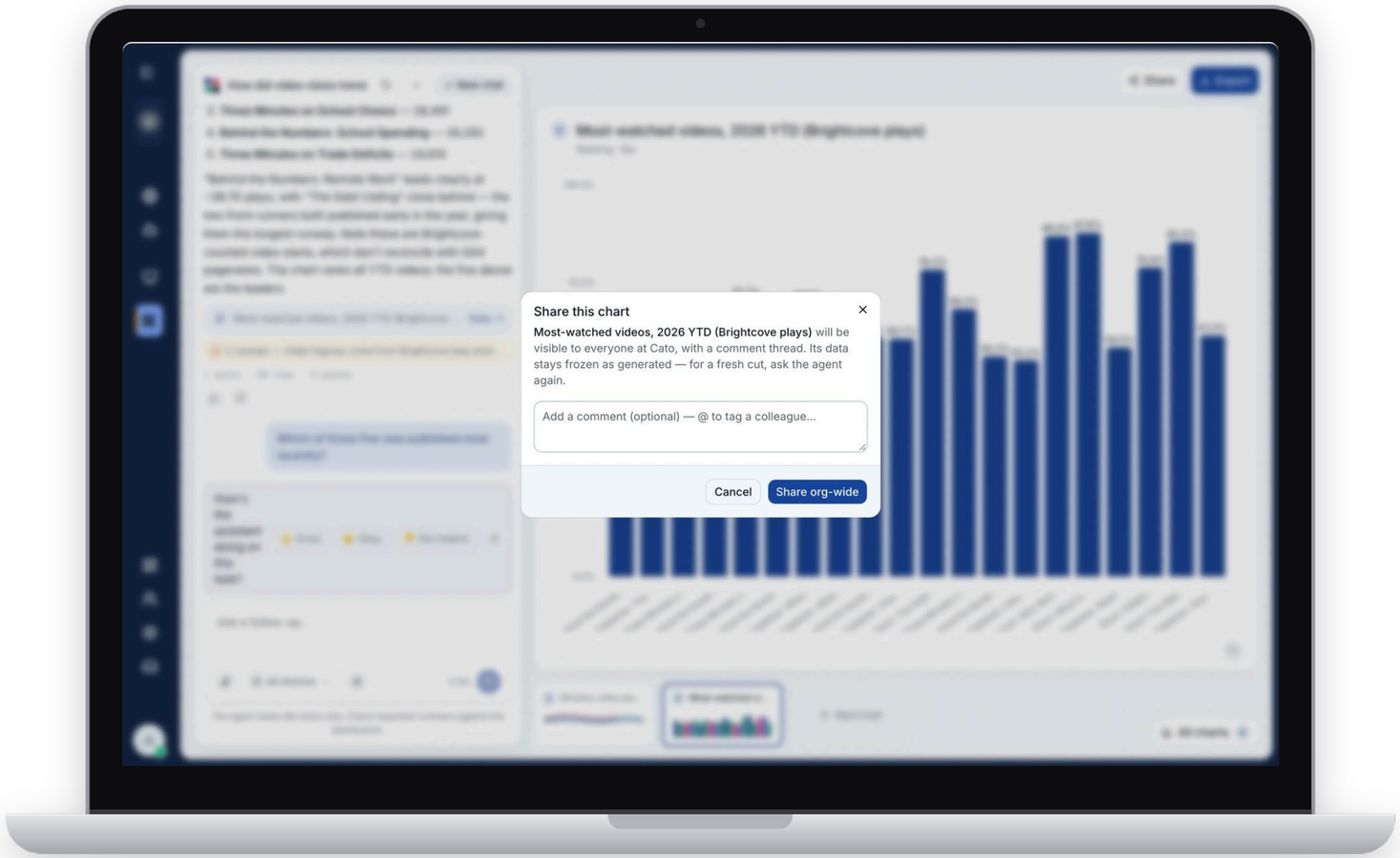
Process

An insight only matters when a team can act on it. Anyone can download the rows behind a chart as a CSV, or freeze a snapshot to share across the organization with a comment thread.

USER CONTROL AND FREEDOM · CONTINUOUS QUALITY ASSURANCE

One-click CSV downloads give analysts the rows, while keeping chart-style feedback apart from the math keeps the numbers safe.

THE FINISHED PRODUCT WITH INVENTED DATA



Collaboration & quality

Share across the org

07 | Sharing & testing

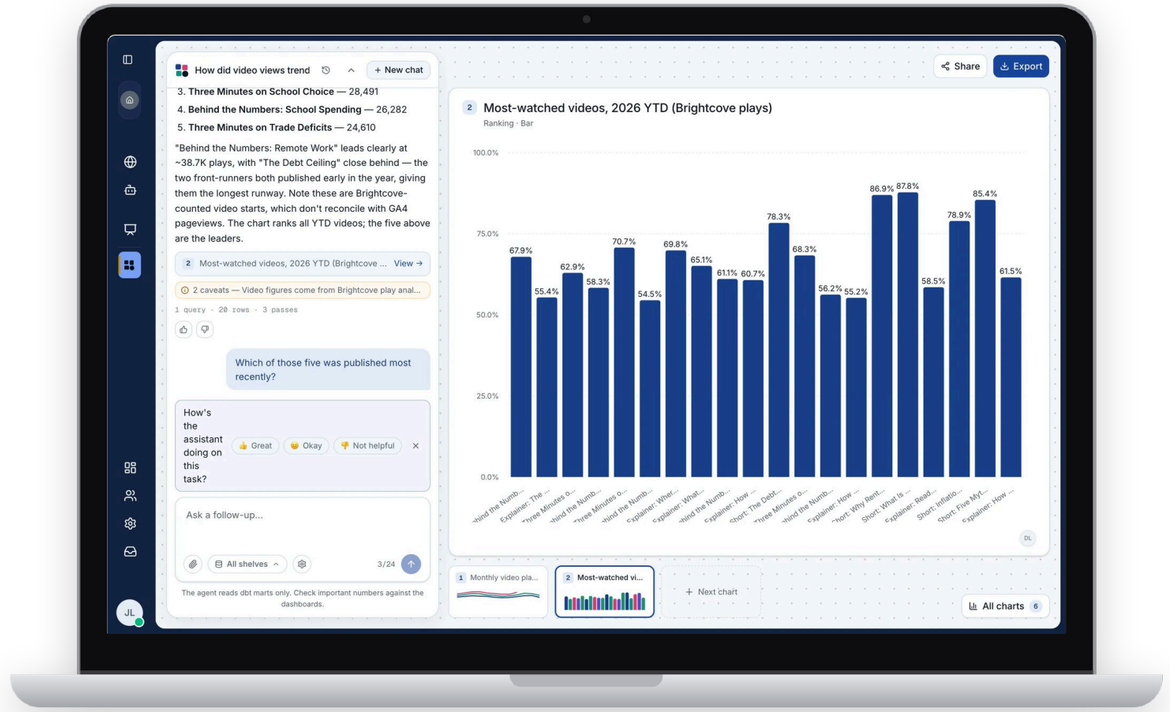
IN-PRODUCT FEEDBACK

Process

Thumbs and a one-line survey teach Data Lab which chart views people prefer—never the numbers—while a weekly golden set in CI checks answers by category.

THE FINISHED PRODUCT WITH INVENTED DATA

BELOW: THE SURVEY'S THREE WORDINGS, DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE



ONE QUESTION, THREE WORDINGS

How's the assistant doing on this task?	Was this helpful?	Did this answer what you needed?
Great · Okay · Not helpful	Yes · No	Fully · Partly · Not really

one wording at random each time it shows · after the third prompt

thumbs on a chart → the view ranking · the survey → the feedback log

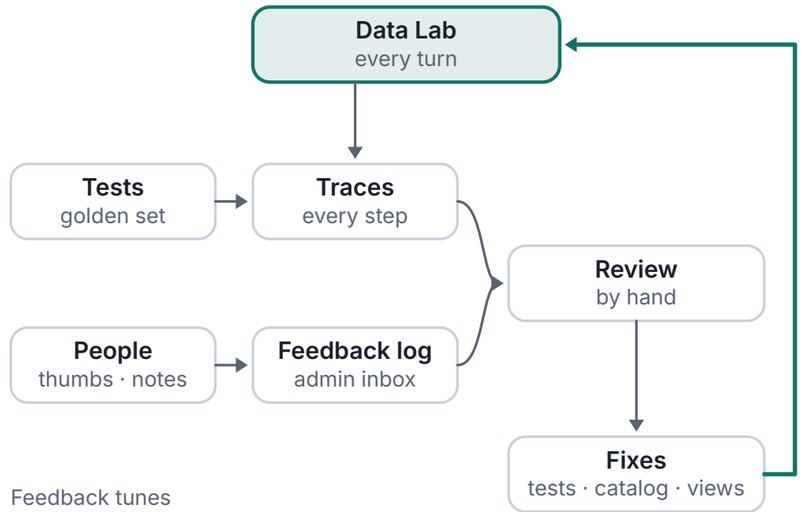
07 | Sharing & testing

EVALS

PROCESS NOTE · HERO WRITES

How the weekly golden set and the review loop catch a wrong answer.

A CASE-PAGE DRAWING KEPT FOR THIS BOOK; BELOW, DRAWN FOR THIS CASE STUDY FROM THE PRODUCT'S CODE

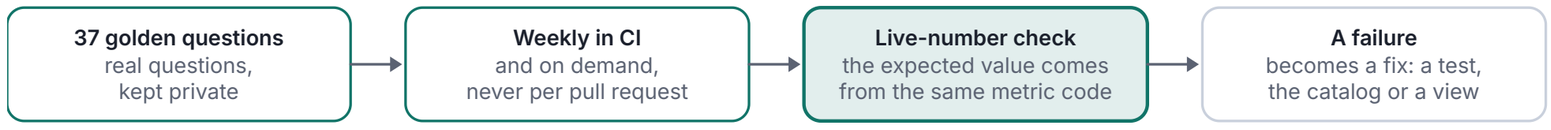


Feedback tunes chart style only, never numbers.

WHAT A GOLDEN QUESTION CHECKS

- **decline** — it should refuse, and why
- **calls** — the tools it should call
- **maxDataCalls** — at most this many data calls
- **forbidMarts** — marts it must not touch
- **chart** — the chart it should draw
- **answerMatches** — what the answer must say

THE RUN



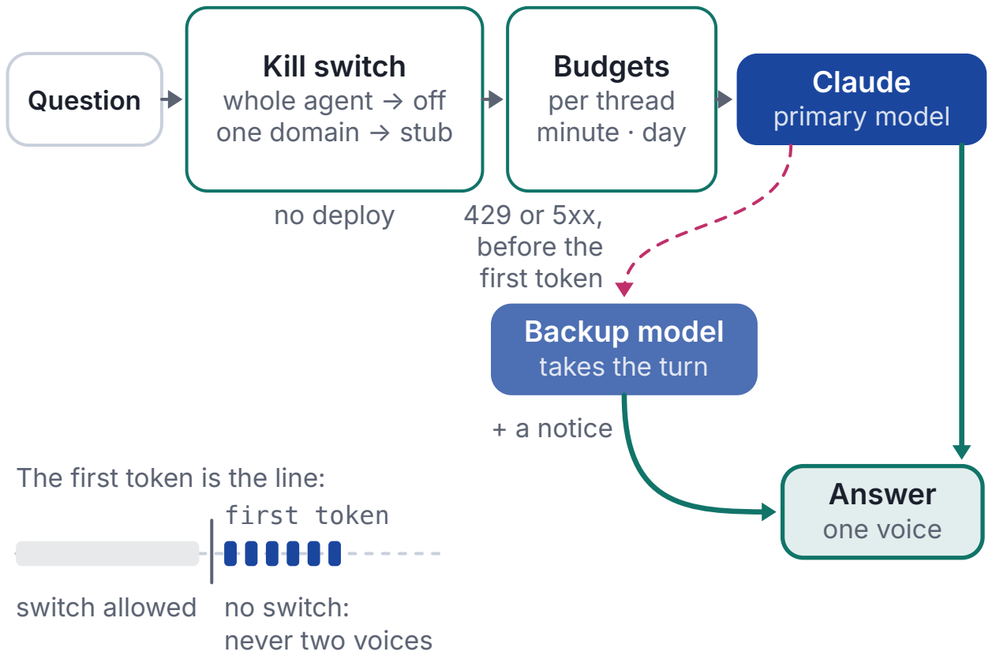
07 | Sharing & testing

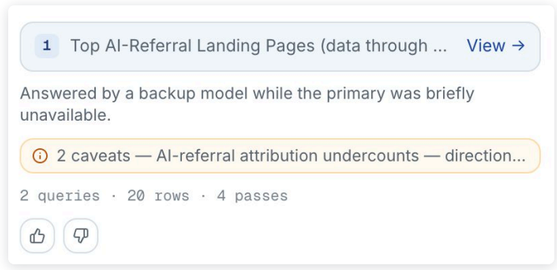
RUNNING IN PRODUCTION

PROCESS NOTE · HERO WRITES

What happens when the model provider has a bad minute, and who can switch Data Lab off.

A CASE-PAGE DRAWING KEPT FOR THIS BOOK





The backup-model notice under an answer · invented data

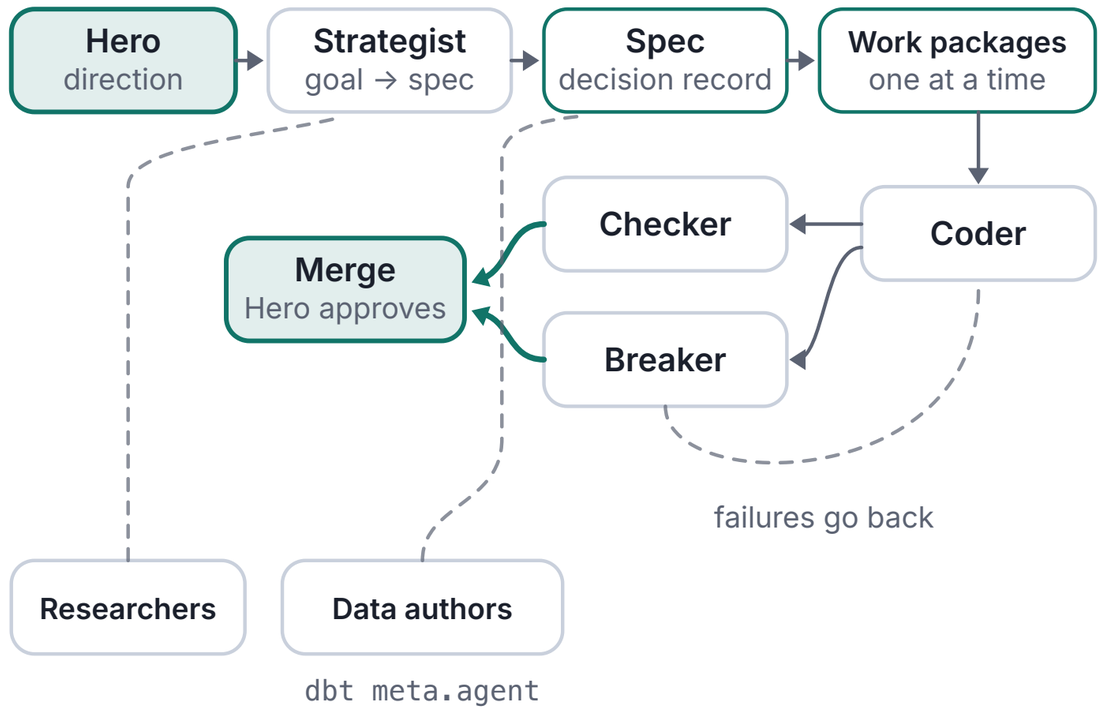
08 | Reflection

THE AGENT TEAM

PROCESS NOTE · HERO WRITES

How one person designed, built and tested Data Lab with a Claude Code agent team.

A CASE-PAGE DRAWING KEPT FOR THIS BOOK



08 | Reflection

REFLECTION

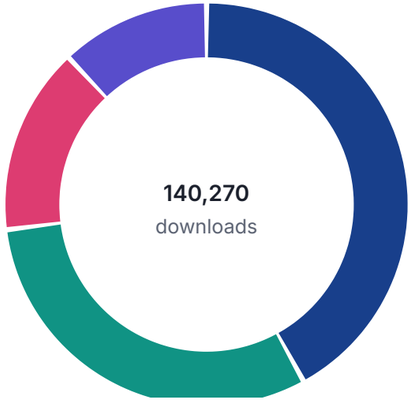
Designing and building Data Lab alone taught me that the hardest UX problems in AI start in the database. Because I owned the dbt models, the verification code and the interface together, I could build accuracy into the foundation instead of hiding flaws behind disclaimers—and turn a multi-day request queue into self-serve answers.

THE REFLECTION'S SAMPLE ANSWER, DRAWN · INVENTED DATA

Where did podcast downloads come from in October?

Podcast downloads by platform, October 2026

Share · Donut



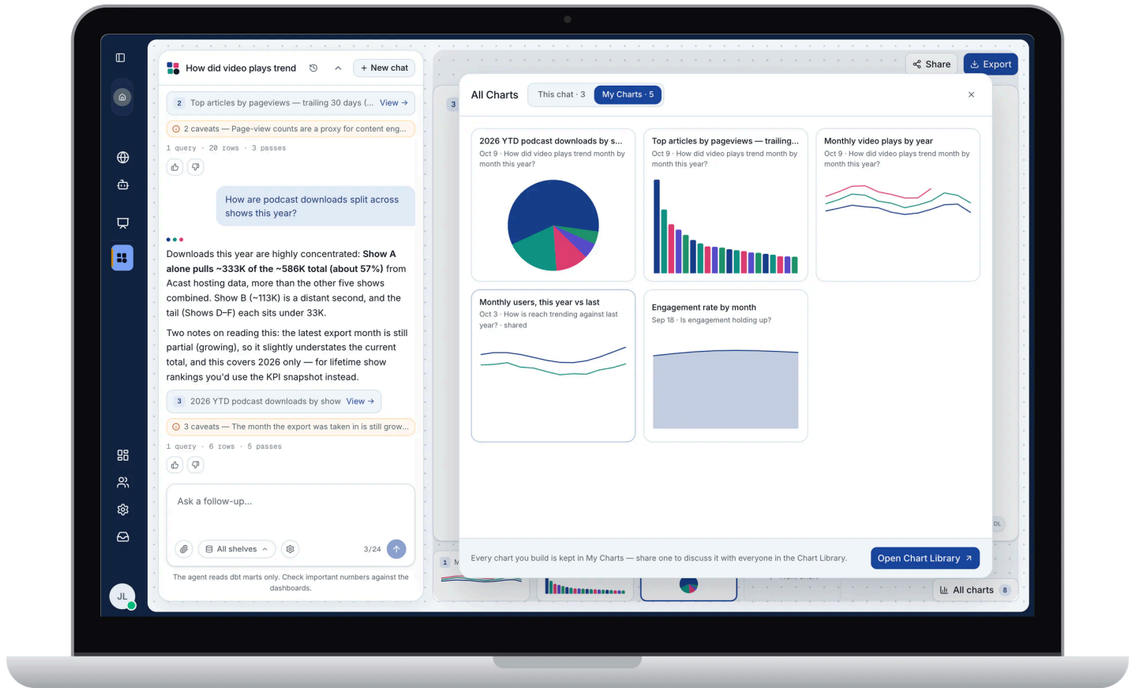
Apple Podcasts	58,920	42 %
Spotify	43,480	31 %
Web player	21,040	15 %
Other apps	16,830	12 %

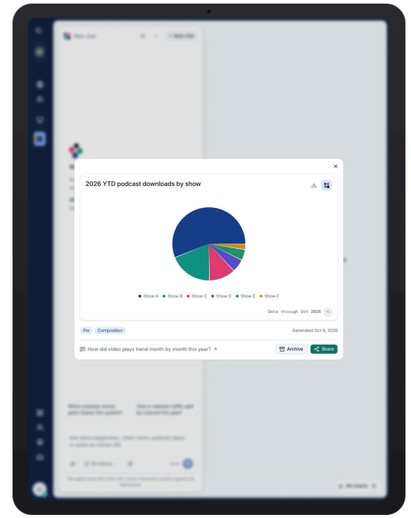
invented sample data, as on the case page

08 | Reflection

ON EVERY SCREEN

THE FINISHED PRODUCT WITH INVENTED DATA





Reflection

On every screen

08 | Reflection

SOURCES

Words

The Data Lab case page's copy: the overview, the eight section titles and clauses, the section texts and the heuristic notes, word for word.

The case page's drawings

Exported from herogodsway.com/work/cato-data-lab: where Data Lab sits, who asks, agent architecture, from dbt schema to live catalog, how the catalog stays true, a question becomes a number, a stage not a sidebar, the state machine, pins and shelves in your access, what Claude sees.

Kept for this book: running in production, the review loop, how Claude answers, the agent team.

Motions and screens

Screen recordings of the demo copy with invented rows, the same takes the case page plays (M1 to M6), and the stills for anatomy, share, feedback, the backup-model notice and the line-up.

The product's code

`lib/agent/query-spec.ts`, `validate-query.ts`, `run-query.ts`, `compute.ts`, `provenance.ts`, `ask.ts`, `phases.ts`, `chart-spec.ts`, `chart-view-ranking.ts`, `chart-preferences.ts`, `catalog/types.ts`, `catalog/meta-schema.ts`, `scripts/gen-agent-catalog.ts`, `kill-switch.ts`, `limits.ts`, `llm.ts`, `feedback/variants.ts`; `components/brand/` (agent loader, stream dots, pixel grid, signal grid).

History

The first version's layout and chart list, the flask mark, the petri loader and the four-cell mark, read from the product's commit history.

Evals

The weekly eval workflow and the golden-set matcher (keys only; the questions stay private).



© 2026 Hero-Godsway Zilevu · herogodsway.com · All rights reserved.