CASE FILE 002 · ASTRATO, BY VIZLIB · THREE YEARS · TWO PODS

Astrato, three years inside a data platform

WHAT	the workbook builder, join canvas and AI assistant of a Snowflake-native analytics platform
FOR	data teams turning their own warehouse into models and dashboards
PROBLEM	the canvas where every model begins had been filed as a bug

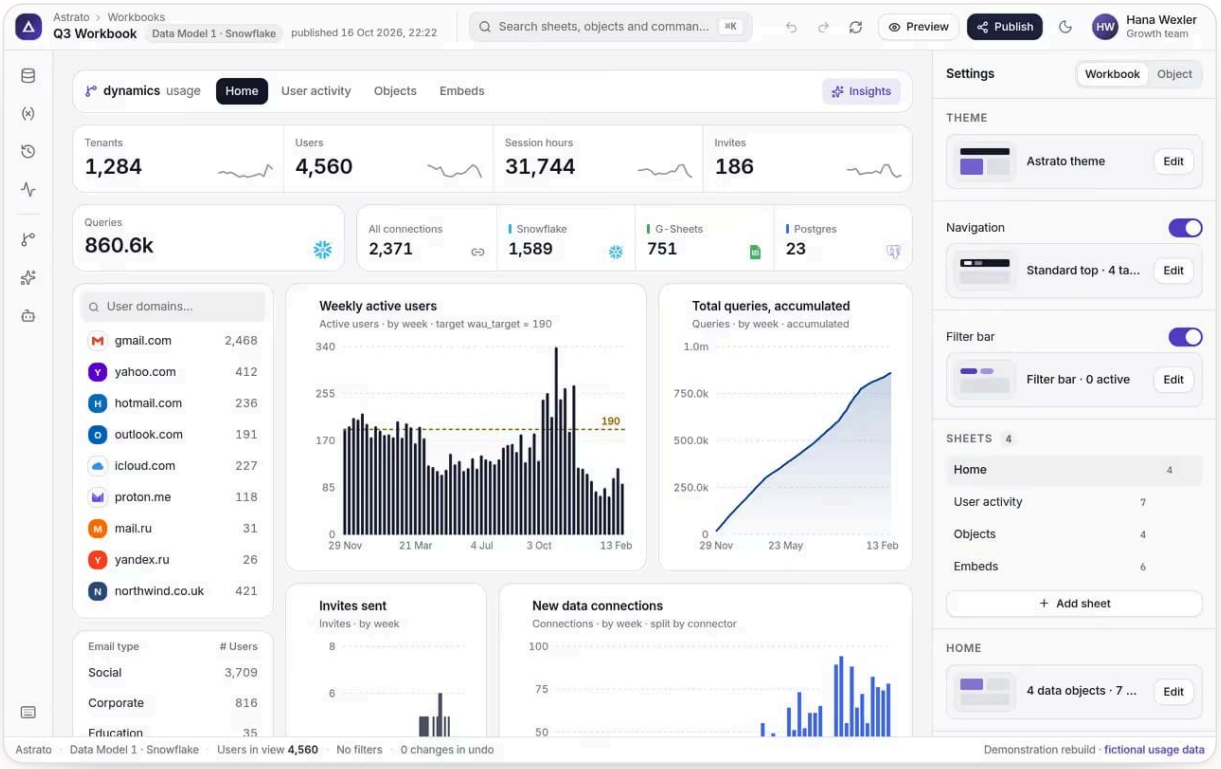
Lead product designer across **both delivery pods** of a Snowflake-native analytics platform, for over three years, from the join canvas to an AI assistant that keeps its SQL in view.

Astrato · the workbook builder

Accessibility: The workbook builder passes axe-core's WCAG 2.1 AA rules at desk and phone, in both settings, and in the 10 states this file links into. Not yet tried with a screen reader. The statement

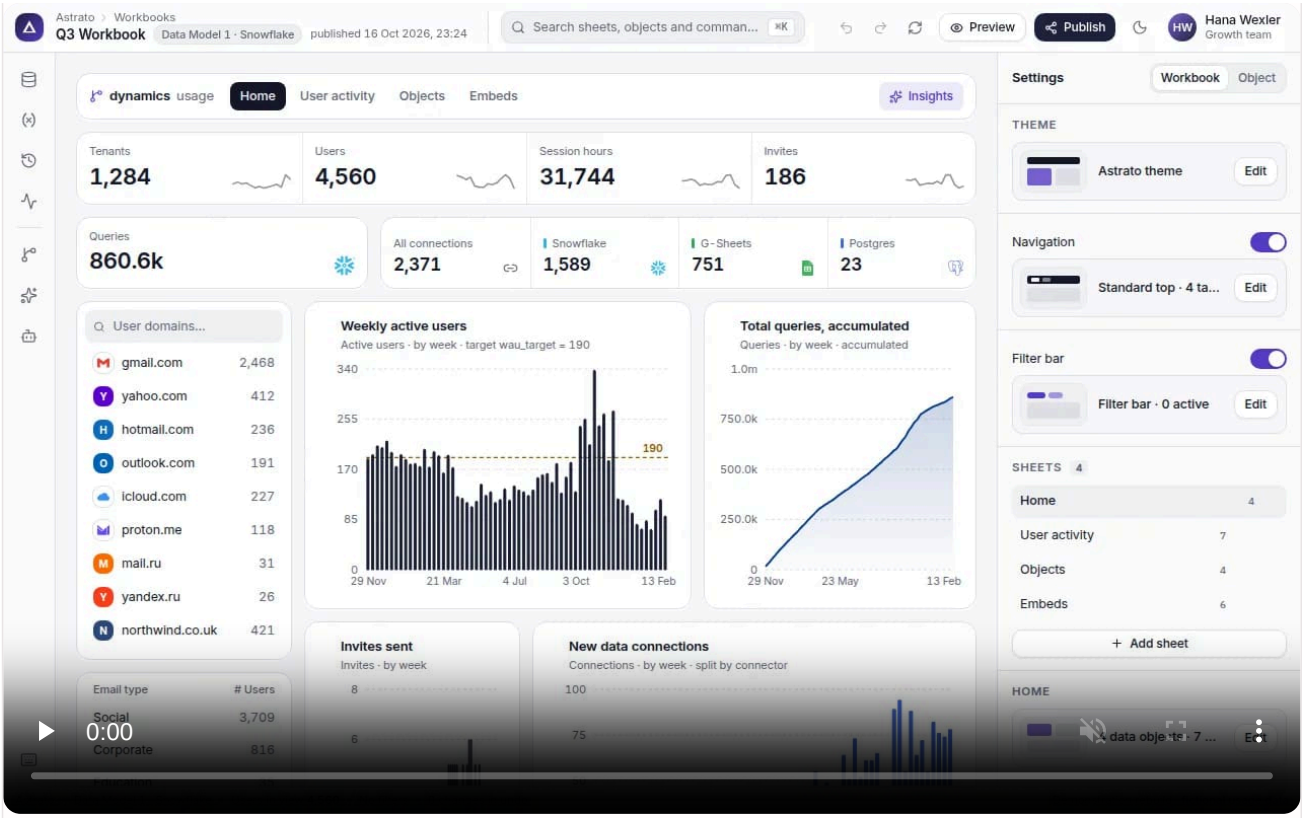
Weight on a first visit: 343 KB of script and 13 KB of style, compressed, against budgets of 360 and 15 KB. The build fails if a product passes its budget. React Flow loads only when the data model is opened.

The states around it, each one a link: the warehouse not answering, a search that finds nothing and a sheet with nothing on it yet. Each says what is true, names the next step and offers one action.



The workbook builder works on a phone, with the inspector and the data panel as sheets. Open it full screen for the whole workspace.

Open the workbook ↗



A walkthrough, 28 seconds, recorded from the rebuild. The workbook, step by step: a filter, the data model, Insights answering with its SQL, the answer kept, the event log and the preview. Each step is captioned as it happens, and the transcript has the same words.

Watch while you read

TRANSCRIPT

ROLE	CLIENT	DURATION	SCOPE
Lead Product Designer	Astrato, by Vizlib	Over three years	Both pods · three years

I OWNED	Design across both delivery pods for over three years: the join canvas and its research, Astrato Insights, and the component libraries
WITH	Both delivery pods, 29 people: Preptors, data preparation, and VIZ-Beans, data visualisation, as the one designer across both
THEIRS	The code, written by eighteen front-end engineers across the two pods, with two data scientists, four QA, a PM and the pods' heads of product and engineering
CONSTRAINTS	The customer's own Snowflake warehouse, paid for by the second per question, with nothing extracted or stored and governance left to the warehouse; and one designer for two pods
FRICTION	The case for the canvas was the query bill, and the research measured how easy the task was; nobody instrumented the bill

THE OUTCOME

4 of 5 rated the task easy

Moderated sessions with attention heatmaps, one thread of three years across both pods.

THE PROBLEM

A canvas filed as a bug

Jira VK-4532: messy overlapping join lines, one-direction connections and no zoom, on the screen where every model begins.

THE MOVE

Fix the geometry, then prove it

Joins route around tables, connect from any side, snap to a grid. Four clicks to create one.

00

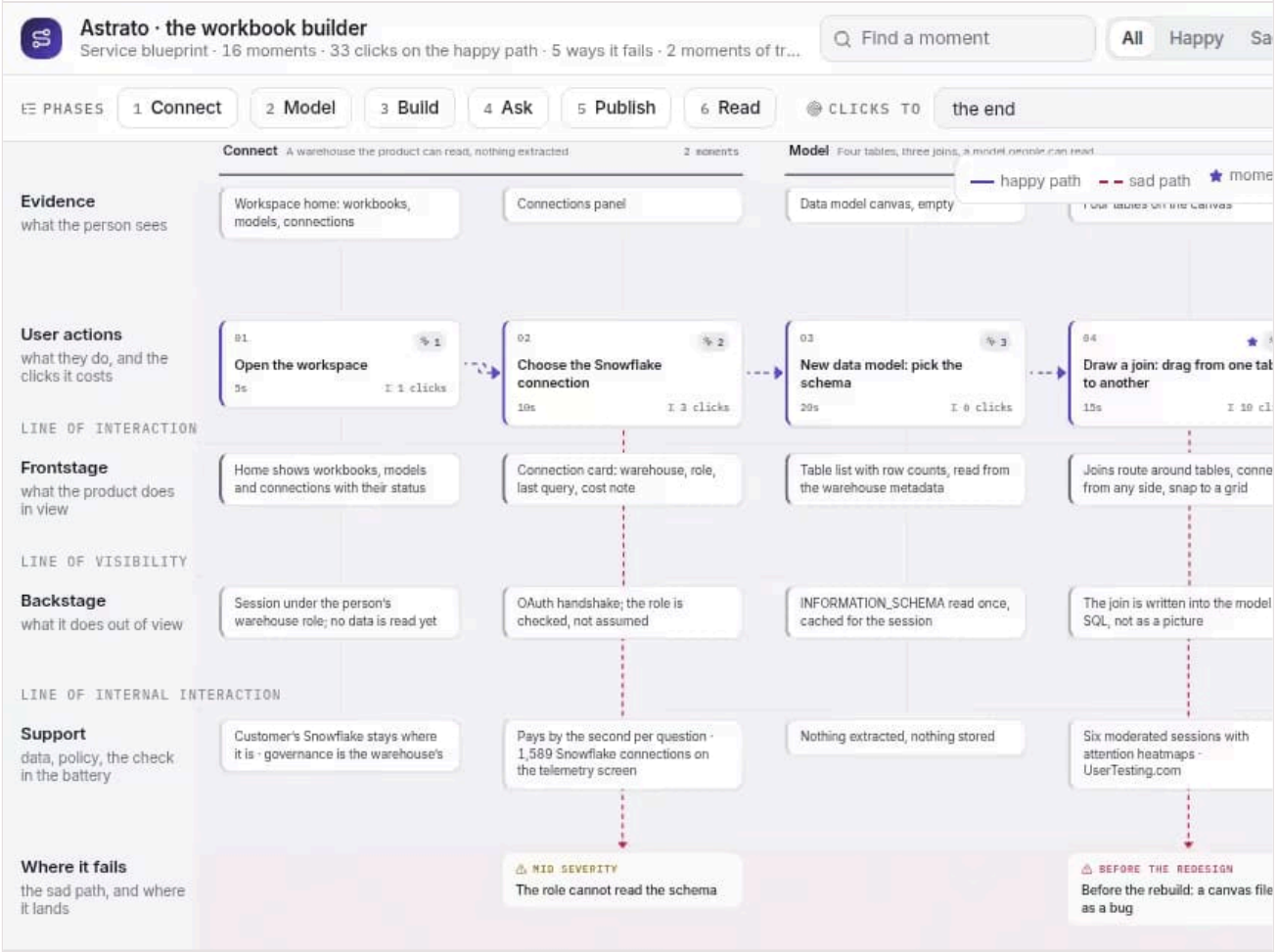
THE BLUEPRINT

The platform as a system, and the spec the rebuild is held to

Before the workbook builder: the platform end to end from the customer's warehouse to a viewer's screen, the numbers the join work was measured on, the evidence under each decision, and the spec the rebuilt editor passes on every release.

SERVICE BLUEPRINT

The workbook builder as a service: sixteen moments from a connection to a published dashboard, with the join canvas as the moment of truth. Pan, zoom, pick a moment, count the clicks to any point, switch the sad paths on.



Wheel to pan, pinch to zoom, click a moment for its detail. ← → walk the moments, f fits the map.

Open the blueprint full screen →

FIG. 001 · THE SYSTEM

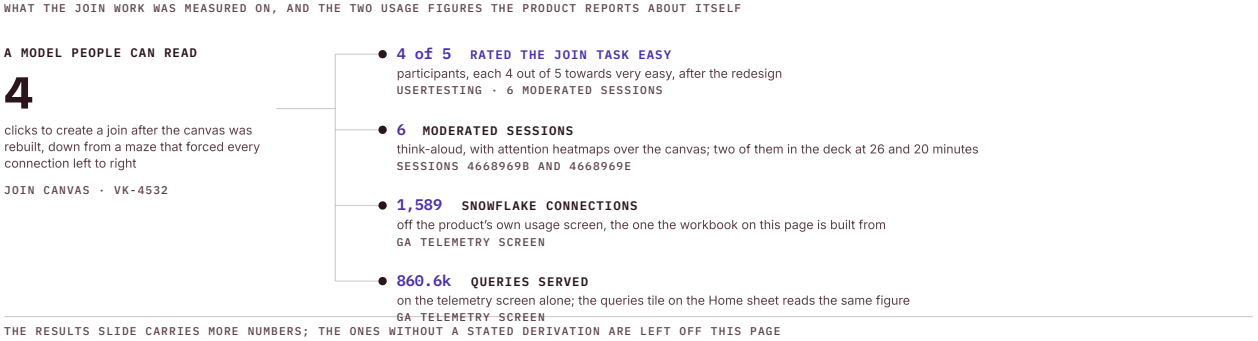


The warehouse never moves. The model sits above it, the workbook reads through the model, and nothing reaches a viewer that was not published on purpose.

The join canvas is the second station, and the reason the three-year file leads with it: everything to its right inherits the model. A join drawn wrong is wrong in every workbook, every published version and every answer the assistant gives, which is why the ticket that landed on the canvas was a platform ticket wearing a UI label.

The assistant sits off the main line on purpose. Insights and Ana read what the sheet is showing, filters included, and nothing else, so a reader can hold what they say against what is on screen. That is the design position from the AI section further down, drawn as a return leg.

FIG. 002 · THE KPI TREE



Two of the four are research measures and two are the product reporting on itself. The outcome section says which numbers from the results slide were dropped for having no derivation.

FIG. 003 · THE EVIDENCE



A ticket, a research programme, a scope declaration and three years. Each one forced a decision that is visible in the running product above.

The spec below holds the rebuilt editor to six statements. Each was written as an acceptance test before the screens were rebuilt, and each is verified by a check that runs before every release of this site, named in the right-hand column. Spec-driven in the plain sense: the sentence came first, the editor had to satisfy it, and something other than my eye confirms that it still does.

01

THE SPEC

AS-01

A join type change rewrites the SQL preview and the row estimate.

ACCEPTANCE

Given the joins editor, when a join is switched from left to inner, then the generated query says INNER JOIN for that table and the row estimate moves.

VERIFIED BY

The checks “the joins editor draws the four tables” and “a join type change rewrites the SQL”.

THE INTERACTION SUITE, RUN BEFORE EVERY RELEASE OF THIS SITE

02

THE SPEC

AS-02

A filter applies to every object on the sheet and says how much of the data is left.

ACCEPTANCE

Given the Home sheet, when a domain is clicked in the filter list, then every object restates and the filter bar reads the users in scope against the total.

VERIFIED BY

The checks “a domain click filters every object on the sheet” and “Clear all removes the pills”.

THE INTERACTION SUITE, RUN BEFORE EVERY RELEASE OF THIS SITE

03

THE SPEC

AS-03

Every edit to the workbook is undoable.

ACCEPTANCE

Given any change, when ⌘Z is pressed, then the workbook returns to the state before it and the status bar counts the changes held.

VERIFIED BY

The checks “the title edit shows on the canvas” and “⌘Z undoes it”.

THE INTERACTION SUITE, RUN BEFORE EVERY RELEASE OF THIS SITE

04

THE SPEC

AS-04

Insights and Ana read only what the sheet shows.

ACCEPTANCE

Given a sheet with filters, when Insights opens or Ana is asked, then every figure in the answer is one the sheet is showing under those filters.

VERIFIED BY

The checks “Insights reads the sheet” and “Ana answers from the data”.

THE INTERACTION SUITE, RUN BEFORE EVERY RELEASE OF THIS SITE

05

THE SPEC

AS-05

Publishing snapshots the workbook, and any version can be restored.

ACCEPTANCE

Given Publish, when a version is named and sent to an audience, then version history lists it and Restore loads it back onto the canvas.

VERIFIED BY

The check “Publish writes v3”, and the Restore button in version history.

THE INTERACTION SUITE, RUN BEFORE EVERY RELEASE OF THIS SITE

06

THE SPEC

AS-06

No join label is covered by a table or crossed by another join.

ACCEPTANCE

Given the data model, when it is drawn at any size, then every join label is clear of every table and every other join.

VERIFIED BY

The canvas check over the joins editor, the same one that guards the two other canvases on this site.

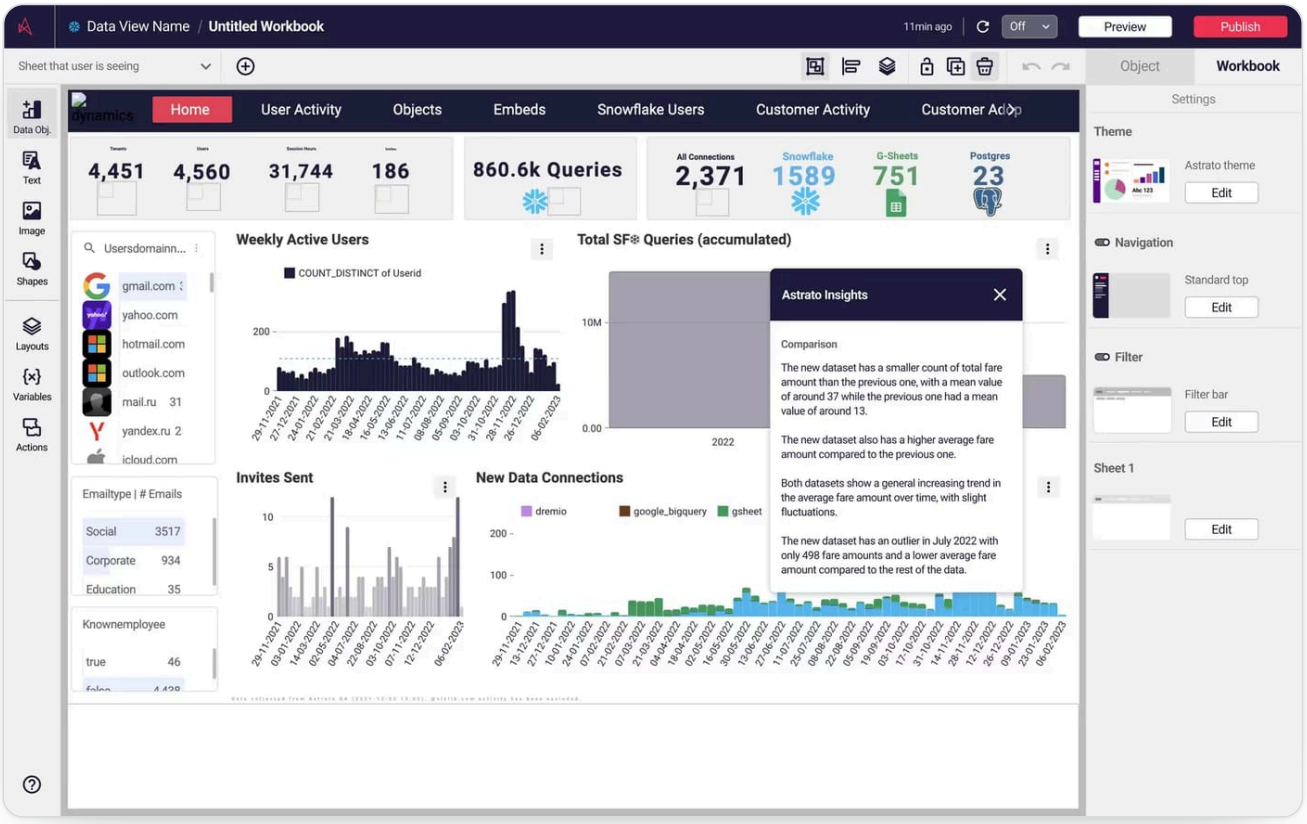
THE CANVAS CHECK, RUN BEFORE EVERY RELEASE OF THIS SITE

01

THE PLATFORM

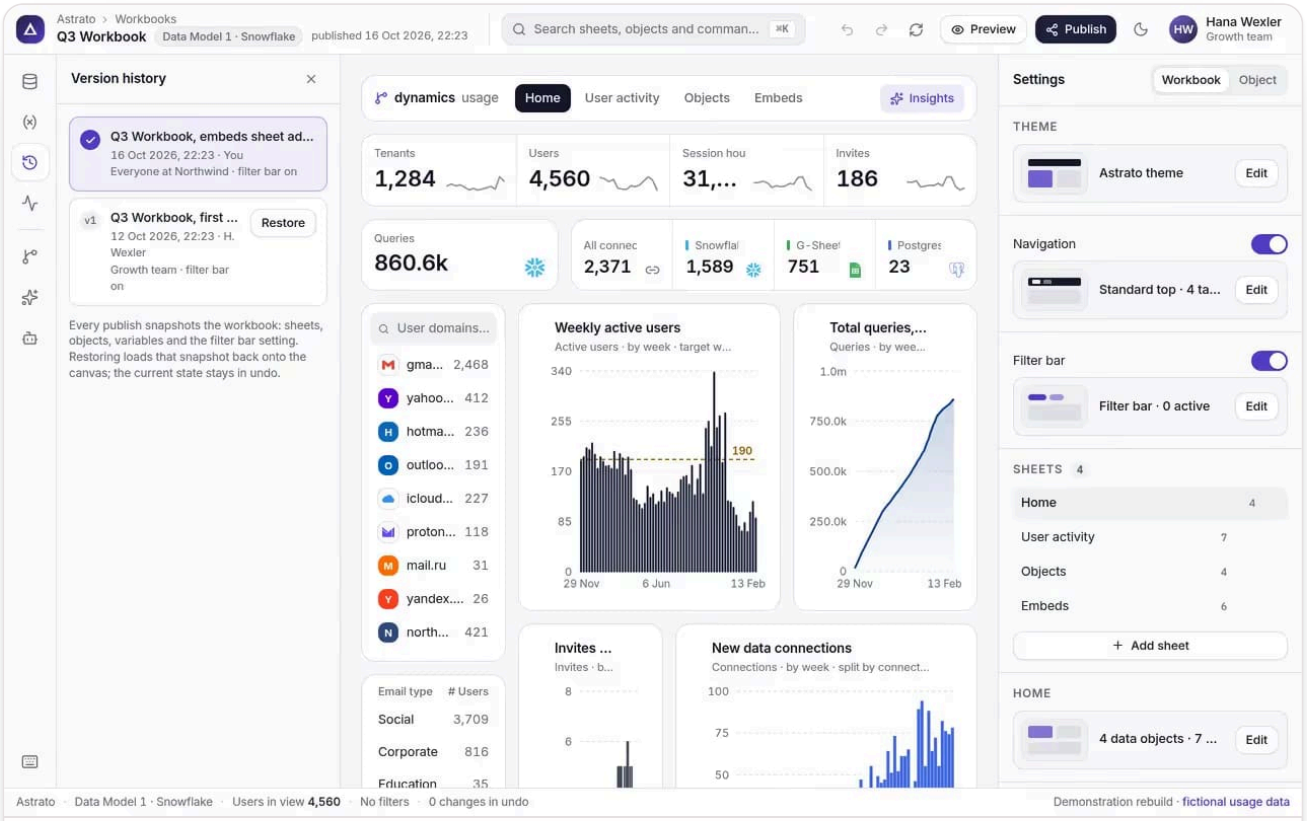
1,589 warehouses. 860.6k queries. One designer.

Astrato sits on the customer's own Snowflake warehouse, the database where a company keeps everything it has and pays by the second for every question asked of it, and turns that into dashboards and answers, 1,589 Snowflake connections and 860.6k queries served on the GA telemetry screen alone. I designed across all of it for over three years; what follows is the join canvas and the AI assistant, two threads out of three years of it.



The workbook editor with the Insights panel open.

Two pods, 29 people, one designer between them



Two pods, one product: the history panel lists what shipped and when, across both delivery pods.

FIG. 004 · BOTH PODS

9 + 20 = 29 PEOPLE

ONE DESIGNER

Counted in both blocks, which is the whole job

NOTHING CROSSES THE GAP

A preparation decision still arrives on the other side as a chart somebody has to read

A decision made on the preparation side turns up on the visualisation side as a chart someone has to read. Being the only designer on both pods meant carrying that context across: three years of making two halves feel like one product.

REBUILT FROM THE DECK Two pods, one designer across both.

Preptors

DATA PREPARATION

Head of Product · Head of Engineering · 6 front-end devs · me · the designer

9 people

VIZ-Beans

DATA VISUALISATION

1 PM · 12 front-end engineers · 2 data scientists · 4 QA · me · the designer

20 people

03

THE JOIN CANVAS

The problem arrived as a ticket. The fix was geometry.

The canvas is where analysts join their tables, telling the system that a row here and a row there describe the same customer, so a single chart can draw on both. By 2023 its problems were literally on file, in Jira.

"Messy overlapping join lines"

JIRA VK-4532 · THE PROBLEM AS FILED

FIG. 005 · WHAT A TANGLE COSTS

A DRAWING PROBLEM, FOLLOWED TO WHERE IT IS PAID FOR

01	02	03	04	05
A MODEL NOBODY CAN READ	**SO IT IS NOT TRUSTED**	**SO THE TEAM BUILDS ITS OWN**	**REDUNDANT DATA VIEWS**	**AND EVERY VIEW IS QUERIED**
join lines cross tables they have nothing to do with	a model you cannot follow is a model you cannot check	faster than arguing about somebody else's	the same question modelled several times over	against a warehouse billed by the second

THE BUSINESS CASE WAS NEVER THE CANVAS. IT WAS WHAT THE CANVAS COST WHEN IT DID NOT WORK.

THE FIX WAS THE GEOMETRY, NOT A BIGGER WAREHOUSE

FOUR CLICKS TO A JOIN · 4 OF 5 RATED IT EASY

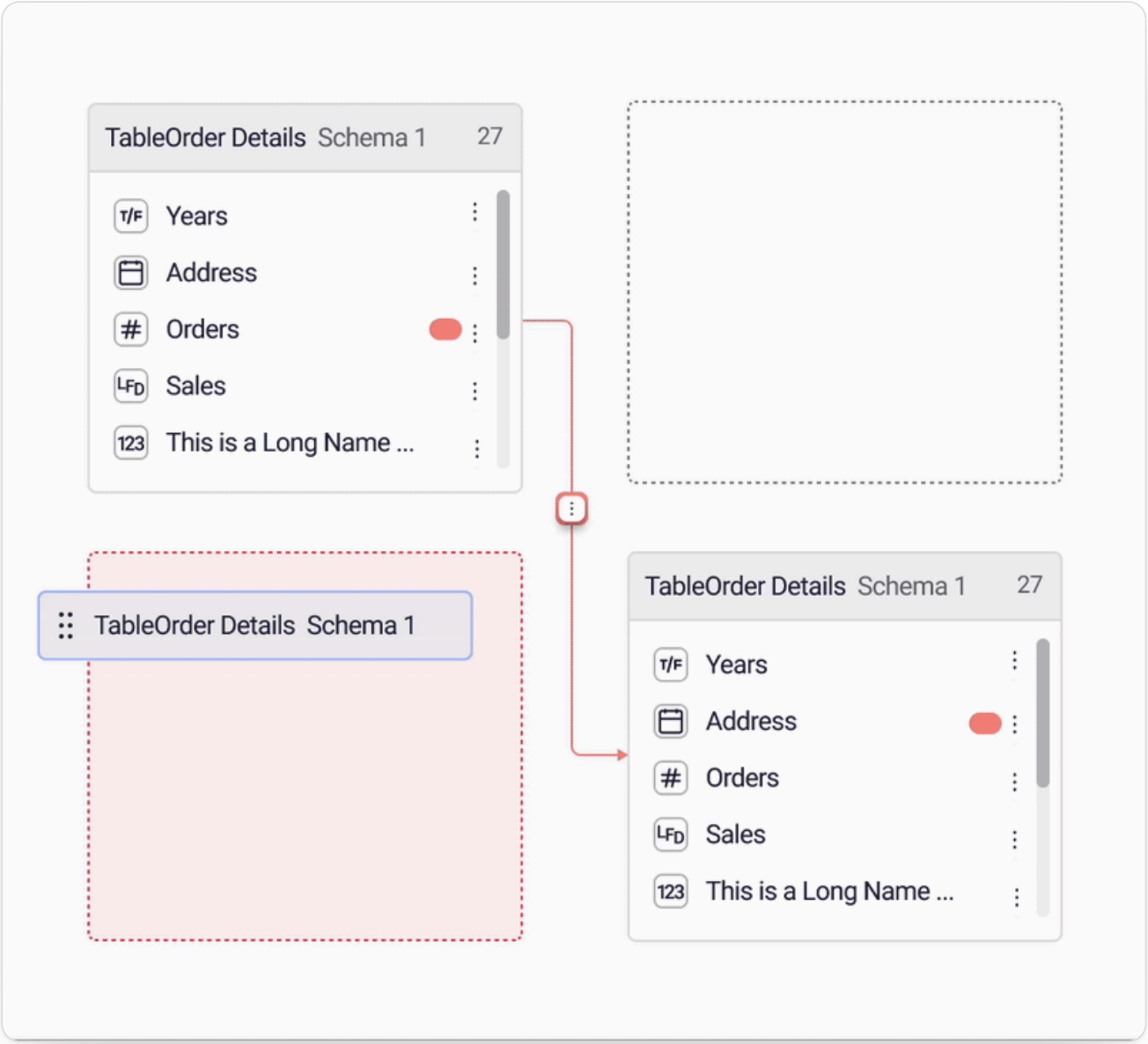
An analyst who cannot read a model does not file a bug about it; they quietly build a second one. That is what the join canvas was costing, and it is why the redesign started with geometry rather than styling.

The redesign fixed the geometry first: joins **route around tables**, connect from **any side**, **snap to a grid**, and creating one takes **4 clicks**. Flip the rebuild:

REBUILT FROM THE BACKLOG The problem as filed.



- Messy join lines, overlapping diagrams
- One-directional left-to-right line connections only
- No expand / collapse on tables
- No zoom on the canvas
- No indication that a table exists out of view



The real canvas in motion: a screen recording of the shipped product.

04

Four of five called it easy. I watched their

eyes to be sure.

Think-aloud sessions on UserTesting.com, with attention heatmaps over the canvas, because where people actually look while building a join is not always where the interface assumes.

REBUILT FROM THE DECK The research programme, and where it landed.

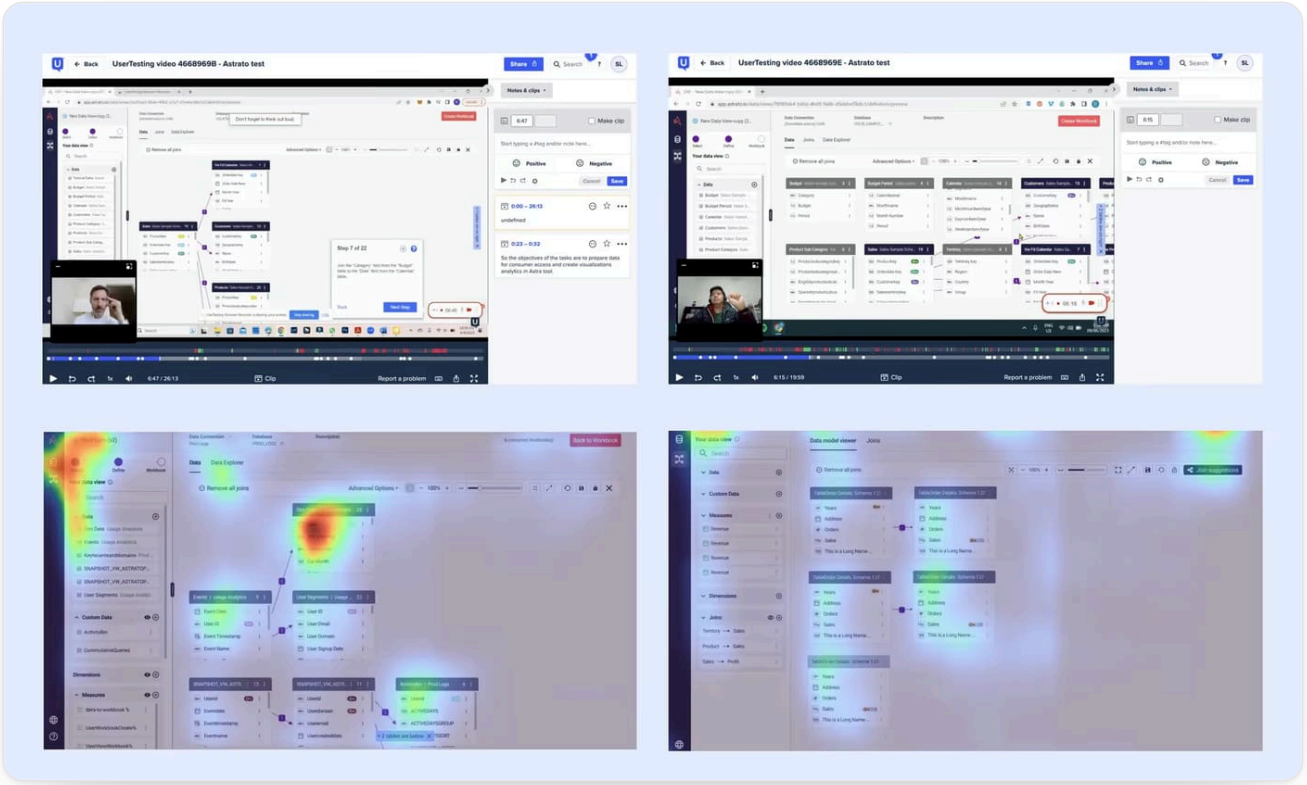
SESSIONS
6
moderated, on UserTesting.com

PROTOCOL
Think-aloud
watching where the canvas surprised people

RECORDINGS
2 in the deck
sessions 4668969B and 4668969E, 26 and 20 minutes

HEATMAPS
Attention
where people actually looked while building a join

Task difficulty after the redesign — 4 of 5 participants rated it 4 / 5 towards "very easy"



Attention gathered on the data list at the left and on each table's header; the join lines, the thing the canvas exists to show, drew far less. So the rebuilt canvas names each table's joins under its header, where people were already looking.

THE TASK, THE QUESTION AND THE HEATMAPS, AND WHAT THEY FOUND

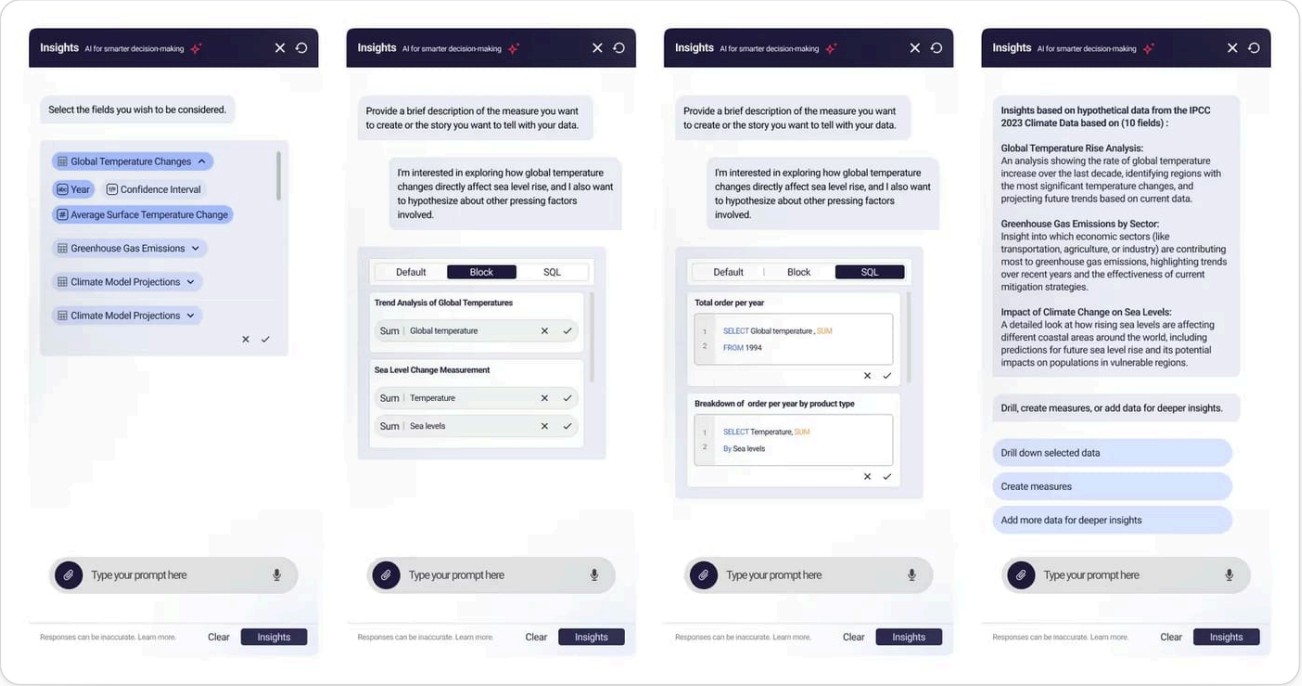
ASKED OR WATCHED	WITH	WHAT IT FOUND
The task, as a session note records its objective: “prepare data for consumer access and create visualisations”	5 participants	Set on the redesigned canvas, the one the product runs, where a join takes four clicks; one task and one question were the whole evidence base
How difficult was it, from 1, very difficult, to 5, very easy?	5 participants	4 of 5 answered 4, one answered 3, nobody answered 5 95% interval for 4 of 5: 36% to 98% A strong signal and a weak statistic, which is why the page never turns it into a percentage
Where people looked while building a join, from the attention heatmaps	The moderated sessions	On the data list and the table headers, much less on the lines; each table now names its joins under its header, and hovering one lights its line

05

ASTRATO INSIGHTS

An AI assistant that shows its SQL

Pick the fields it may draw on ("based on 144 fields"), type a prompt, get measures, dimensions and SQL blocks back. The design position that mattered: the SQL stays **visible and editable at every step.** For people whose job is querying data, hiding the query would have broken trust.



Default, Block and SQL views of the same suggestion.

FIG. 006 · INSIGHTS, THE QUERY STAYS VISIBLE

01
SCOPE THE FIELDS
144
fields the analyst chose,
and the ask is bounded to them

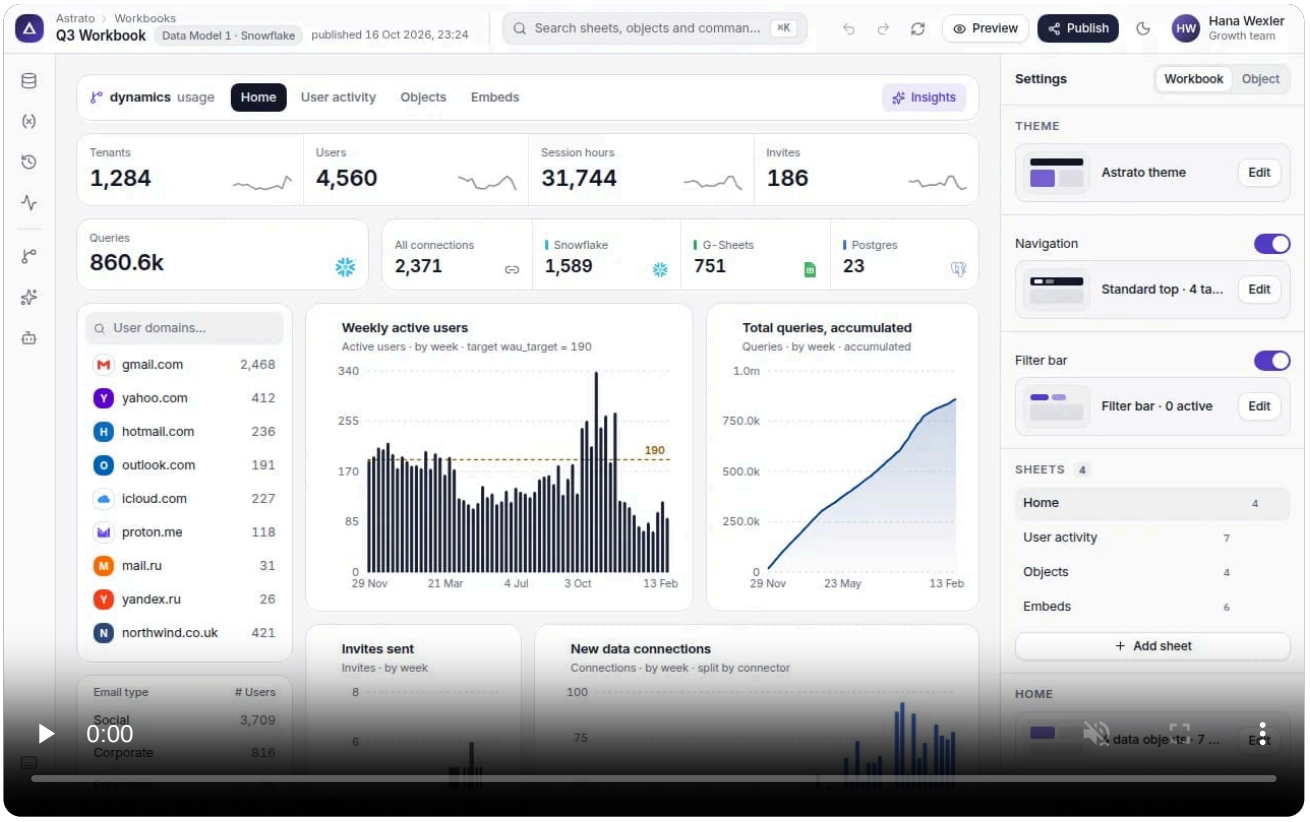
02
THE PROMPT
1
plain-language question,
no query language required

03
THE SUGGESTION
3
parts returned: measures,
dimensions and the SQL

04
THREE VIEWS
3
the answer, its parts,
and the SQL, all editable

05
INTO THE SHEET
1
object on the canvas.
The query stays yours

The SQL is not a disclosure panel opened after the fact; it is present at every step, and editable.



Seven seconds, recorded from the rebuild. Insights is opened and asked for MRR by region. It answers in its default view, four regions ranked, then shows the SQL that produced the answer, open to edit, with Add to sheet beneath it.

Watch while you read

TRANSCRIPT

How others show their working

Every assistant that writes queries for analysts has to decide how much of the query to show. What each of these shows, from its own documentation:

SHOWING THE WORKING, ASSISTANT BY ASSISTANT

ASSISTANT	WHAT IT SHOWS	WHAT THE ANALYST CAN DO WITH IT
Snowflake Cortex Analyst	The SQL it generated for the question	Edit it, rerun it, and save the pair to a repository of verified queries the assistant draws on next time
ThoughtSpot Sage	The search tokens the question became: the measures, attributes and filters	Edit the tokens; the documentation asks users to check them before sharing an answer
Power BI Copilot, in DAX query view	A DAX query, with comments	Run it, keep or discard it, and ask for it to be explained step by step
Astrato Insights	The fields in scope, then the answer, its blocks and its SQL, as three views of one thing	Change the scope, edit the SQL at any step, and add the answer to the sheet as an object they own

06

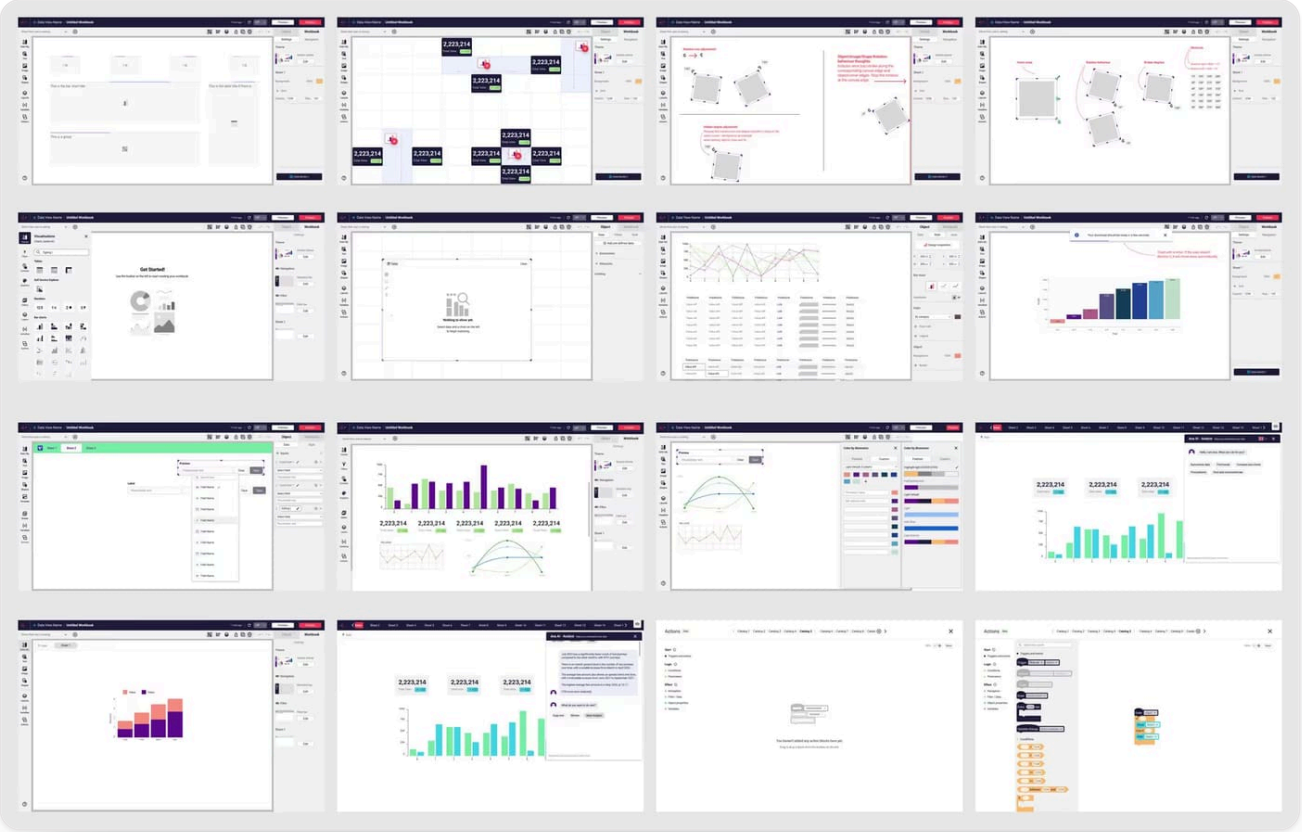
SYSTEM & HANDOFF

Specs shipped as designed

Every new component is a maintenance cost someone pays later, so new ones appeared only when an existing one genuinely couldn't stretch. With eighteen engineers across the pods, precision in the spec was cheaper than conversations after the build.

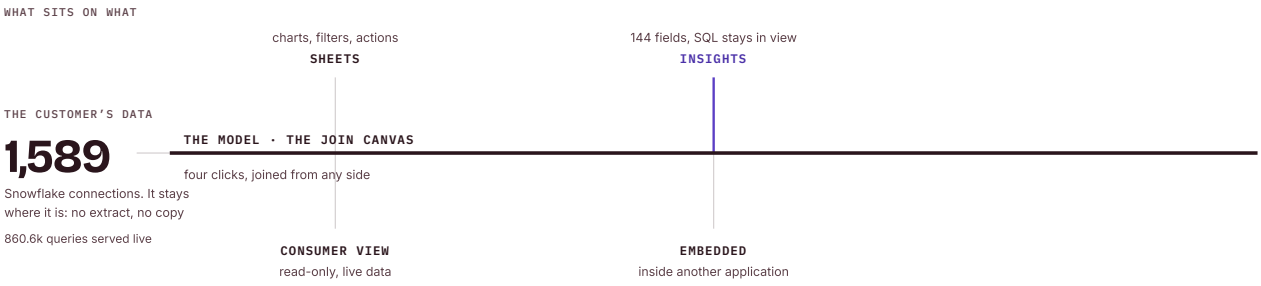


A component spec sheet as it went to engineering.



Sixteen screens of the workbook editor, one system underneath.

FIG. 007 · WAREHOUSE TO CONSUMER



The model is the load-bearing layer: every sheet, every answer and both exits sit on it.

THE FLOW · END TO END

07

Every dashboard in the product starts in the join canvas

The path from a Snowflake connection to something a colleague can open, and the decision that used to end the journey early.

FIG. 008 · CONNECT TO CONSUMER

01	02	03	04	05	06
CONNECT	THE CANVAS	THE SHEET	ASK ANA	PREVIEW	PUBLISH
1,589	**4**	**1**	**144**	**1**	**2**
connections to warehouses that never move	clicks to join, from any side, and it snaps	model underneath charts, filters and actions	fields in scope, with the SQL in view throughout	look at exactly what a consumer will get	exits: read-only, or embedded

THE CANVAS AS FOUND
joins ran one direction, lines crossed the tables, there was no zoom and the model that everything rests on never got built at all

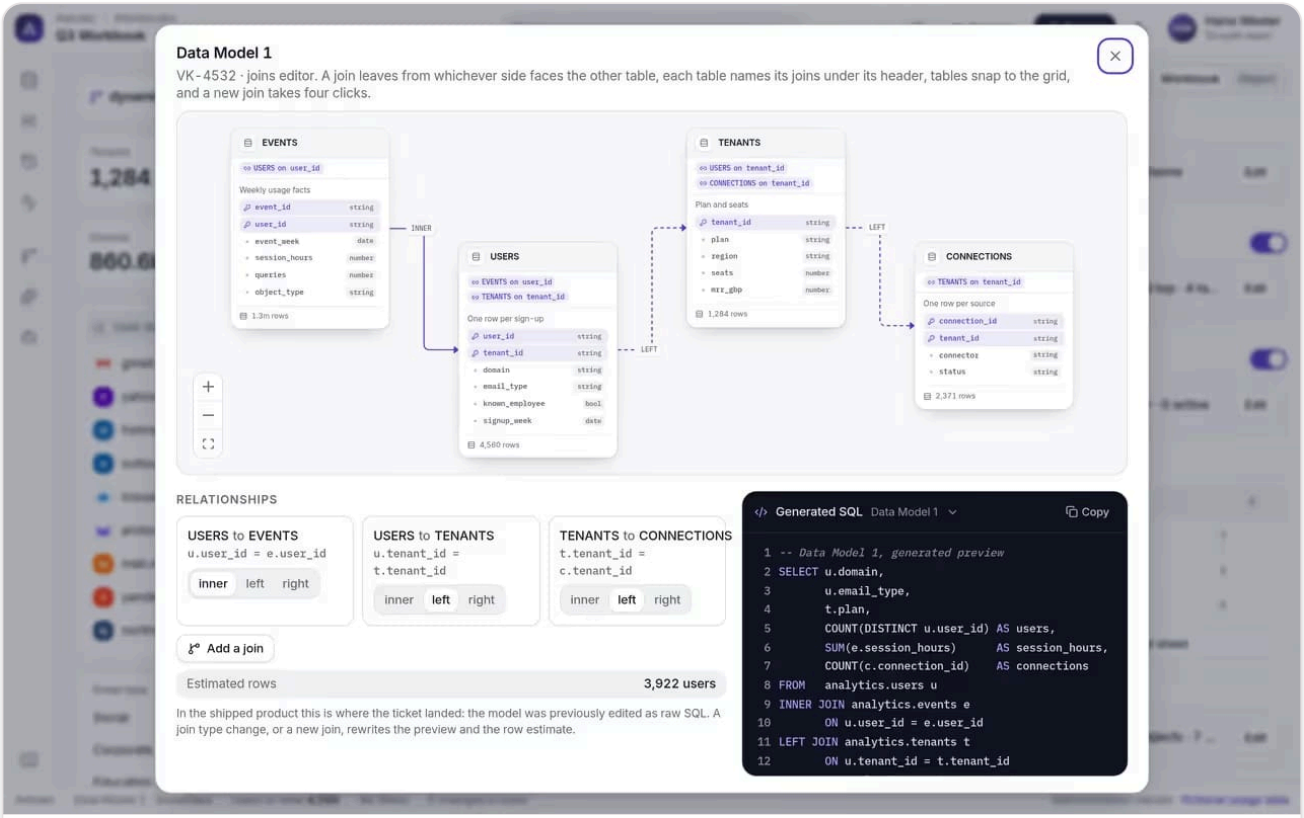
The whole model lives or dies at one decision: can an analyst read the joins they just made.

WHAT WAS BROKEN ·
WHAT CHANGED

08

What the ticket said, and what shipped against it

Six problems from three years on the product, the mechanism used on each, and what changed afterwards.



What moved, in the product: the joins editor as it runs today.

01

WAS BROKEN

Join lines crossed tables they had nothing to do with, every connection was forced left to right, and reading a model meant untangling it first. It was on file as VK-4532, affecting BI developers and data analysts.

WHAT I CHANGED

Rebuilt the canvas geometry rather than restyling it: joins route around tables, connect from any side, and snap to a grid, and creating one takes four clicks.

WHAT IT MOVED

4 of 5

participants rated the join task 4 out of 5 towards “very easy” after the redesign.

USERTESTING, 6 MODERATED SESSIONS

02

WAS BROKEN

The canvas complaints were opinion, argued release after release, so nothing was prioritised.

WHAT I CHANGED

Ran think-aloud sessions on UserTesting.com with attention heatmaps over the canvas, so what people looked at while building a join became evidence rather than a claim.

WHAT IT MOVED

6

moderated sessions, two of them in the deck at 26 and 20 minutes. The clutter stopped being an opinion and became measurable.

SESSIONS 4668969B & 4668969E

03

WAS BROKEN

An assistant that answers a data question with a chart asks people whose whole job is querying data to trust a black box.

WHAT I CHANGED

Kept the SQL visible and editable at every step (Default, Block and SQL views of the same suggestion), and made the field scope explicit up front: “based on 144 fields”.

WHAT IT MOVED

Insights shipped as something its users can check: nothing is returned that cannot be opened, read and rewritten by the analyst it was generated for.

ASTRATO INSIGHTS, AS SHIPPED

04

WAS BROKEN

A decision made on the data preparation side arrived weeks later on the visualisation side as a chart somebody had to read, because the two pods planned separately.

WHAT I CHANGED

Being the only designer across both pods, I carried that context deliberately: one component system underneath both halves, and specs precise enough for eighteen engineers to build from without a follow-up meeting.

WHAT IT MOVED

3 yrs

of delivery across both pods, on one system rather than two.

SCOPE FROM THE PROJECT DECK

05

WAS BROKEN

Nothing told an analyst that a table existed off-screen, tables could not be collapsed, and there was no zoom, so large models were built half blind.

WHAT I CHANGED

Zoom and expand/collapse on the canvas, plus an indication that something exists out of view, taken straight from the five problems listed in the ticket.

WHAT IT MOVED

Reading the model stopped being a prerequisite for building one, which is what made four-click joins usable on a real warehouse rather than a demo.

FIVE ITEMS ON VK-4532

06

WAS BROKEN

Every new requirement arrived with a new component attached, and each one is maintenance somebody pays for later.

WHAT I CHANGED

Owned the component libraries for both products and applied one test: a new component only when an existing one genuinely cannot stretch.

WHAT IT MOVED

Specs shipped as designed. Three years on, the dashboards have been restyled by other people, and the join canvas, the structural work, is still the way the product connects data.

HANDOFF SPECS, SHOWN ABOVE

09

OUTCOME

Three years, one

coherent product

MEASURED AGAINST THE KPI TREE IN THE BLUEPRINT

IT WORKED IF	RESULT	STATUS
A join takes a handful of clicks	4 clicks, down from a maze that forced every connection left to right	MET
Analysts find the modelling task easy	4 of 5 participants rated it easy, across 6 moderated sessions; a strong signal from a small sample	MET
The assistant's answers can be checked	The SQL stays visible and editable at every step of Insights	MET
The bill for duplicated dashboards falls	Not measured; the hindsight below says what I would change, and the plan after it how	NOT MEASURED

29
people across two pods
9 PREPTORS · 20 VIZ-BEANS · ONE DESIGNER ON BOTH

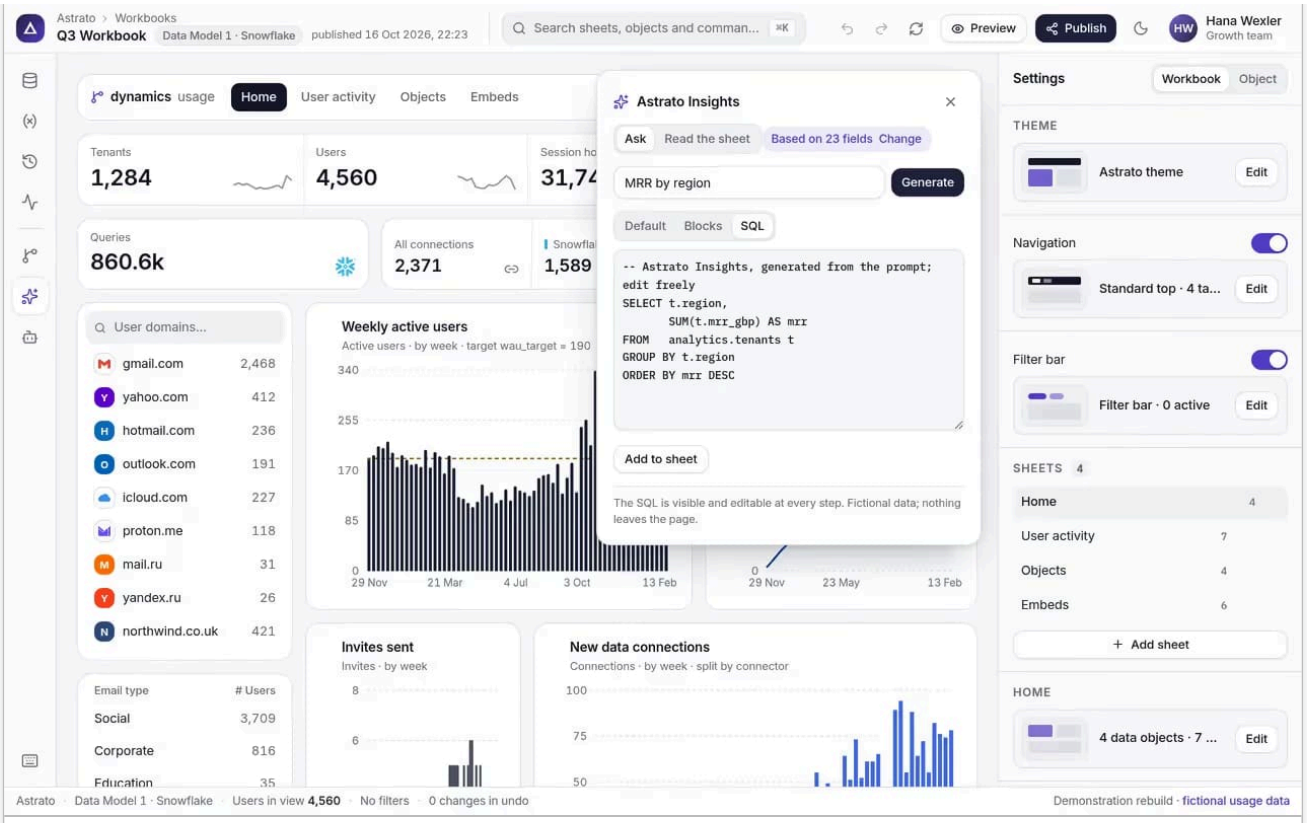
1,589
Snowflake connections
OFF THE PRODUCT'S OWN USAGE SCREEN

4
clicks to create a join
DOWN FROM A MAZE

4 of 5
participants rated the join task easy
ON TAPE · USERTESTING

The business case was never the canvas, it was what the canvas cost when it did not work: teams were independently replicating dashboards to meet needs the model could not express, which meant redundant data views and a query bill to match. Centralising that into the Data Explorer is the outcome the results slide led with, and it is the one a finance director understood without a design vocabulary. The deck carries the results slide, and says which of its numbers I have dropped for having no stated derivation.

Three years in one product teaches you something a portfolio of short engagements can't: features age. Dashboards get restyled, panels get renamed, and most of what I shipped in those three years has been touched by someone else since. The work that compounded was structural: the join canvas fixed how the product thinks about connecting data, and it outlived every dashboard trend it shipped alongside.



The outcome, running: Insights scoped to its fields, asked for MRR by region, with the SQL it wrote open to edit.

What I would do differently

01 **Spend more of the years on what does not age.** Most of what I shipped in three years has been touched by someone else since; the work that compounded was structural. I would have moved to the model and the canvas sooner.

02 **Measure the bill, not only the task.** The case for the canvas was the duplicated dashboards and the query bill they ran up, and the research measured how easy the task was. I would instrument the bill from the first sprint.

03 **Run more than five sessions before quoting a percentage.** Four of five participants rating the task easy is a strong signal and a weak statistic.

The plan that line asks for

Written for this rebuild, after the fact, in the order Amplitude's taxonomy playbook gives: what the work is for, the metrics that would show it, then the events those metrics need, named as Segment's Track spec asks, an object and a past-tense verb. The rebuild runs it: every call is checked against the plan as it is made and shown in the product's Events panel, and nothing is sent.

The objective is the one the business case named: fewer duplicated dashboards and a smaller query bill, without making the assistant's answers any harder to check.

THE METRICS, AND WHERE EACH IS READ

METRIC	HOW IT IS WORKED OUT	READ FROM
Query credits per sheet, weekly	The credits each query a sheet runs is charged, summed by the tag the query carries, which names its workbook, sheet and object	The warehouse itself: Snowflake's query attribution history, grouped by query tag. It runs up to eight hours behind, so the metric is daily, not live
Objects repeating a pair	Objects added that show a measure and dimension the workbook already shows, of all objects added	`Object Added`, its `repeats_pair`
Answers kept	Insights answers added to a sheet, of the answers given	`Insight Added to Sheet` over `Insight Answered`
Kept answers checked first	Kept answers whose SQL was opened before they were kept; the guard on the other three, since the assistant exists to be checked	`Insight Added to Sheet`, its `sql_viewed`

THE EVENTS, AND WHAT EACH ONE CARRIES

EVENT	FIRES WHEN	PROPERTIES
`Insight Asked`	A question is sent, typed, picked from the suggestions or arriving in a link	`source` typed, suggestion or link; `scope_fields` a number
`Insight Answered`	Insights returns its measures, dimensions and SQL	`measures`, `dimensions`, and `left_out`, the measures asked for but out of scope, each a number
`Insight View Changed`	The answer is switched between its views	`view` default, blocks or sql
`Insight SQL Edited`	The generated SQL is first changed by hand, once per answer	None
`Insight Added to Sheet`	An answer becomes an object on the sheet	`sql_viewed` and `sql_edited`, each true or false
`Experiment Viewed`	Someone in a workspace assigned to the test below first sees an answer, once per page	`experiment_id`, `experiment_name`, `variation_id` and `variation_name`, as Segment's A/B testing spec names them, each from a closed list
`Object Added`	An object lands on a sheet, inserted, copied or added from Insights	`kind` one of the seven object types; `source` insert, copy or insights; `repeats_pair` true or false

What is not collected matters as much. A question can carry a customer's own figures, so **its words never leave the page**: a call records where the question came from and how many fields it could draw on, and no property anywhere holds free text, a field's values or a row of data.

The test it makes possible

Insights opens each answer on its chart, with the SQL one click away. Whether opening on the SQL instead would make people check more without keeping less is a question a test can settle and an argument cannot. This design follows Kohavi, Tang and Xu's *Trustworthy Online Controlled Experiments*. It was written for the rebuild, which has no users, so it has not run; both arms are in the product, as the chart first and the SQL first.

THE DESIGN, FIXED BEFORE ANY DATA

PART	DECISION
Hypothesis	Opening answers on their SQL raises the share of kept answers that were checked first, and does not lower the share of answers kept
Primary metric	Kept answers checked first, from the plan above
Guardrails	Answers kept must not fall, query credits per sheet must not rise, and the split itself must hold
Unit	The workspace, not the person. A workbook is shared, so one analyst's object sits on a colleague's sheet, and splitting people inside a workspace would put both arms on one screen
Analysis	Per workspace, the unit that was split; counting each answer as if it were independent would make the result look surer than it is
Exposure	`Experiment Viewed`, once, when someone in a test workspace first sees an answer, so only people who met an arm are compared
Length	Whole weeks, at least two, so each arm sees every day of the working week
Size	Set from the baseline the first weeks of the plan would measure, not guessed here

The split, checked first. If the arms come out a different size from the even split the design asks for, the assignment or the logging is broken, and nothing else in the result can be trusted until the cause is found. The book calls this a sample ratio mismatch. Type the workspaces counted in each arm.

CHART FIRST

SQL FIRST

Waiting for a count in each arm.

SEE ALSO

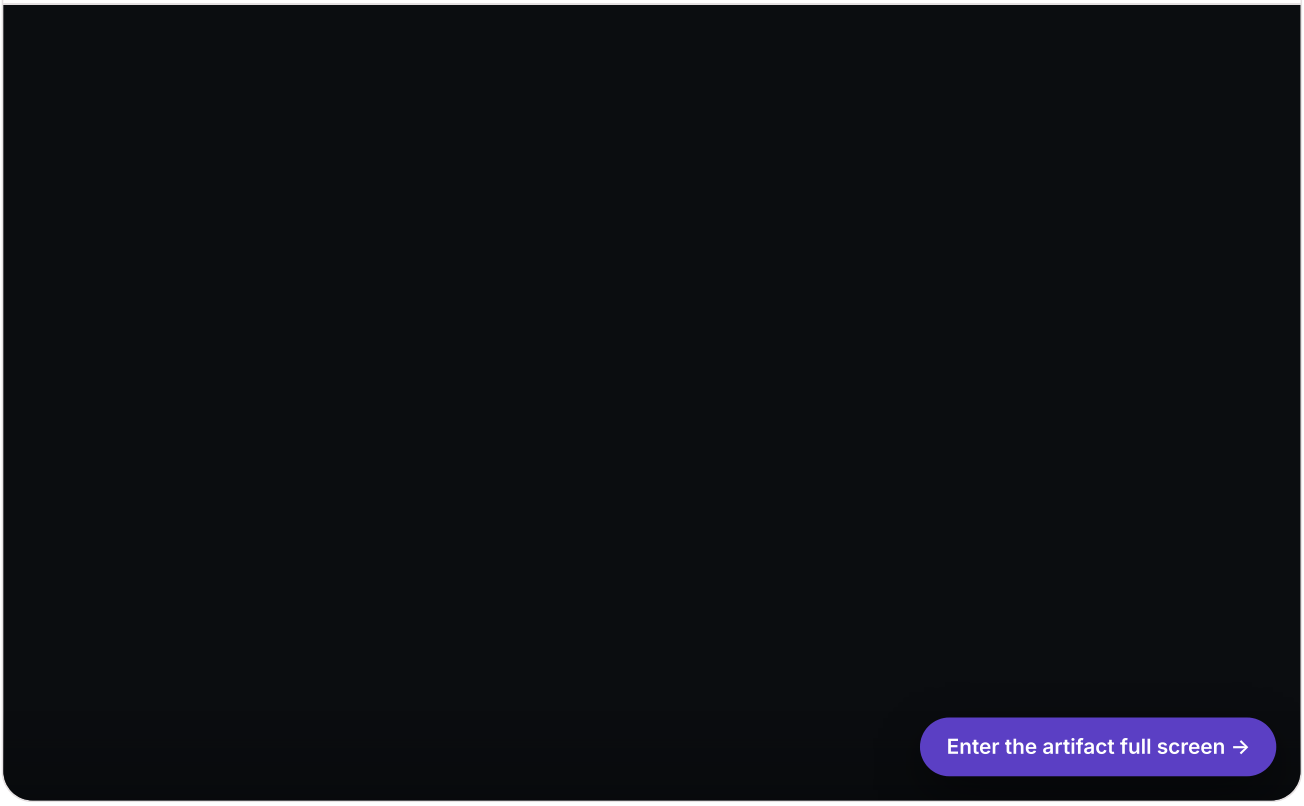
001 · OI Design System
The same rule at platform scale: one source of truth, and an AI that builds only inside it.

004 · Novo Nordisk

Built on Astrato: every number opens to the SQL that produced it, as Insights does.

LIVE ARTIFACT

The full Astrato case study: the interactive site, with the join-canvas demo and UserTesting evidence inside it.



SOURCE NOTES

Join canvas problems and affected users from Jira ticket VK-4532, last updated 3/8/2023, shown above.

Research material from moderated UserTesting.com sessions, videos 4668969B (26 min) and 4668969E (20 min), plus the six-session grid, heatmaps and results board, all from the project deck.

Connection and query counts from product telemetry visible in the GA screenshot; data collected from Astrato GA, Dec 2021.

Feature count, pod compositions, click count and the "based on 144 fields" panel are from the project deck.

Full interactive case study: withcontexts.com/artifacts/astrato

← PREVIOUS

OI Design System · a design system that remembers why

NEXT →

Safe Space · youth mental health for the NHS

Shad Abbaslou
Product Design Lead · London
withcontexts.com
Last updated 11 Nov 2026

LinkedIn ↗ Read the CV