ATSCALE

AtScale and Grok: The Comprehensive Guide

Everyone will have the same AI models. The advantage goes to the company with the best semantic model. Here is how Grok gets one.

Contents

Introduction 3

Why AtScale and SpaceX Super Intelligence are working together 3

- The models are converging 4
- Every enterprise already has semantics. That's the problem. 4
- What the BIRD-Interact benchmark shows 4

How it works: The path to 100% AI accuracy with Grok 5

- CoModel builds the semantic model 6
- Ava asks which metric you meant 7
- ACE writes the SQL and computes the metric 9
- Canary measures how often Grok is right 10

Conclusion 11

FAQs and resources 12

Introduction

SpaceXSI, OpenAI, Anthropic and Google are racing to build ever-smarter models, and enterprises are spending billions to keep up. Grok 4.3 is one of the strongest of them. Yet ask any frontier model a plain question about the business, what revenue did we book in the Northeast last quarter, and it will usually get the number wrong.

On BIRD-Interact, a public benchmark that runs business questions against real enterprise databases, every model AtScale tested answered more of them correctly once a semantic layer sat underneath it, some by several times.[4]

The gap keeps AI out of the decisions that matter most: financial reporting, forecasting and operations, where the number has to be right and has to be the same number every time. It's also the ceiling on what the model vendors can sell into the enterprise.

Everyone will have the same AI models. The advantage goes to the company with the best, most accurate, most cost-effective *semantic* model. Grok brings the reasoning. Your semantic model brings the definitions.

Why AtScale and SpaceX Super Intelligence are working together

AtScale has joined the SpaceX Super Intelligence Partner Network as a Marketplace Partner. Developers and business teams building on Grok now get the metric definitions their business has already approved, and the same number every time they ask.

Grok Build connects to AtScale through AtScale's Model Context Protocol (MCP) server to read metrics, dimensions and relationships, subject to the user's existing permissions. AtScale Ava clarifies an ambiguous business question before a calculation runs, and the AtScale AI Computation Engine (ACE) computes the answer on the customer's existing data platform.

> “Enterprise teams want agents that can work with their data and understand how their business operates. AtScale brings established business definitions and governed computation to Grok, helping customers build useful applications and extend agents into everyday business workflows.”
>
> **Toni Adams**, Head of Partnerships, SpaceXSI

The AtScale plugin for Grok Build is available now, and customers with an existing AtScale deployment can install it from the marketplace listing.

That's the news.

The models are converging

Grok 4.3 reasons over unfamiliar data well. On BIRD-Interact it posts the highest no-layer score of any model we tested, ahead of Claude Sonnet 5 and well ahead of GPT-5. If raw reasoning over an unfamiliar schema were the whole job, Grok would already be winning it.

It isn't the whole job, and the spread between frontier models on these questions is narrow and getting narrower. SpaceXSI, OpenAI, Anthropic and Google each shipped a stronger model this year, and every enterprise can buy any of them. A year from now the difference between two companies will be the definitions their model reads, not the model itself.

Every enterprise already has semantics. That's the problem.

Revenue is defined in the finance close, again in the Power BI model, again in a Databricks Unity Catalog metric view, again in a Snowflake semantic view and again in a spreadsheet someone owns, and they disagree. The ones that do agree were written for last year's business and nobody has touched them since. The best ones are locked inside a BI tool where an agent can't reach them.

So a model asks what we booked last quarter, finds five answers, picks one on the strength of a column name, and builds the rest of the report on top of it. Working SQL looks exactly the same whether the definition behind it is yours or improvised, so the reviewer has no way to tell.

None of the rules that decide the number live in the schema. A tenure rule says a customer counts as active only after 90 days. An exclusion list keeps intercompany transactions out of the total. The fiscal calendar starts in February. Finance reads the booked column and the data team reads the recognized one. None of that is a column type, a constraint or a foreign key. It lives in a close checklist, a BI measure and somebody's head.

Fixing it is a computation problem. A semantic layer that computes the metric gives every model and every agent one definition of revenue, margin and active customer, and computes it the same way every time it's asked. That claim is testable, and we tested it.

What the BIRD-Interact benchmark shows

In the BIRD-Interact benchmark, 317 questions have a right answer. Across every model AtScale tested, adding a semantic layer raised how often the model got that answer, by as much as 345%. For Grok, an AI-generated AtScale semantic model took it to **62.5% accuracy**.[4]

That's half the story. The real question is how Grok and AtScale work together to reach 100% attested accuracy on the questions an enterprise actually asks.

The benchmark shows where the remaining gap is. The misses cluster in four technical areas:

1. A prompt that can mean more than one thing: “show me revenue” against five revenue measures in the model.
2. A term the data never defines: “active customer,” where no column marks one.
3. The same value spelled differently across tables: CA, Calif. and California in three dimensions.
4. A metric that takes several passes to compute: a ratio of two aggregates at different grains, or a balance that can't be summed across time.

Faced with any of these, Grok writes more SQL and guesses, and a guess that runs looks exactly like an answer.

The AtScale and Grok flywheel closes all four.

How it works: The path to 100% AI accuracy with Grok

Four components close the rest of the gap, and each one keeps a person who knows the business inside the loop:

1. **The semantic model.** AI reads the documentation, ontologies, dashboards and spreadsheets a company already has and drafts them into one semantic model, which a data engineer approves before it ships.
2. **Prompt disambiguation.** A resolution step settles every request before anything runs, both for a person talking to their data and for an agent calling a tool.
3. **A computation engine.** Something has to compute the metrics that take several passes, and compute them the same way every time.
4. **Accuracy attestation.** A layer that measures how often the whole system is right, so each miss becomes a change to the model.

That loop is the AtScale AI Accuracy Flywheel, and each turn leaves it more accurate than the last.

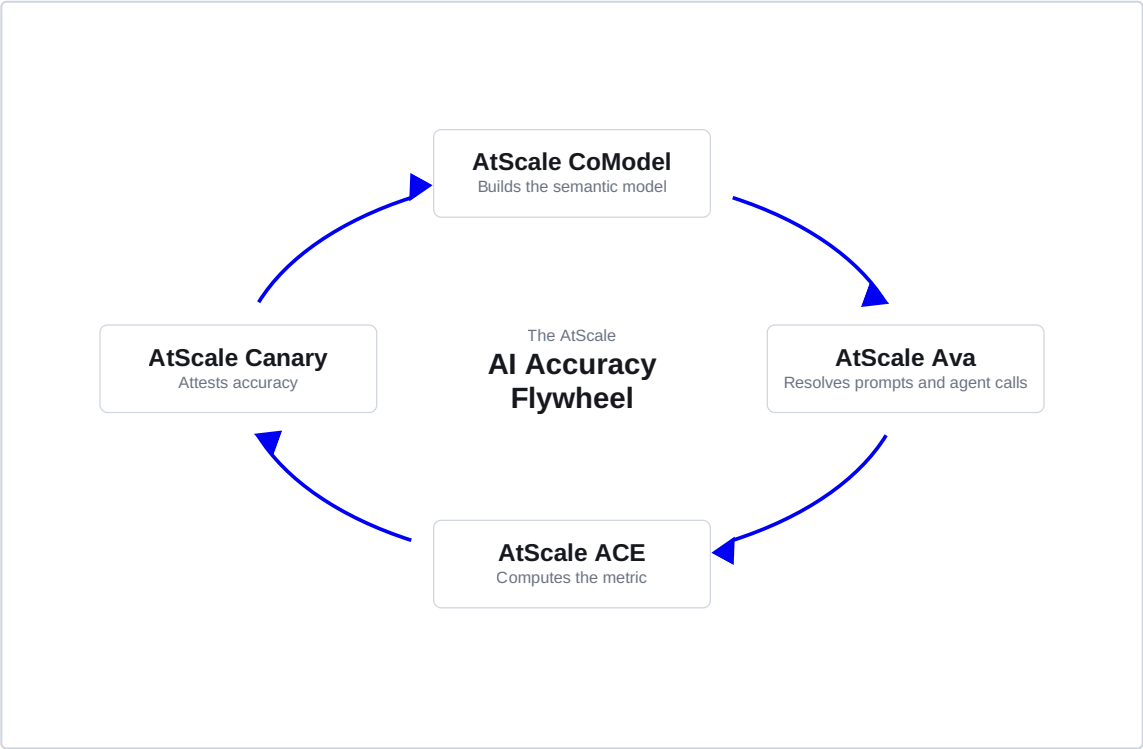


Figure 1. The AI Accuracy Flywheel: model, measure, compute, repeat. CoModel builds the semantic model, Ava resolves the request, ACE computes the metric and Canary measures the error rate, feeding every miss back as a fix.

CoModel, Ava, ACE and Canary are those four stages, and a person approves every change they propose.

CoModel builds the semantic model

Every answer Grok returns depends on the semantic model underneath it, and writing that model by hand is why semantic layers stall. AtScale CoModel drafts it instead, from the business context a company already has: warehouse schemas, BI dashboards and the measures inside them, spreadsheets, documentation, catalogs and ontologies.

CoModel's agents gather that context, draft metrics, dimensions and relationships in SML, test the draft with Canary, and propose every change as a reviewable diff. A data engineer approves each one. Nothing ships unreviewed, and approved changes land in Git, where Ava reads them and ACE computes from them.

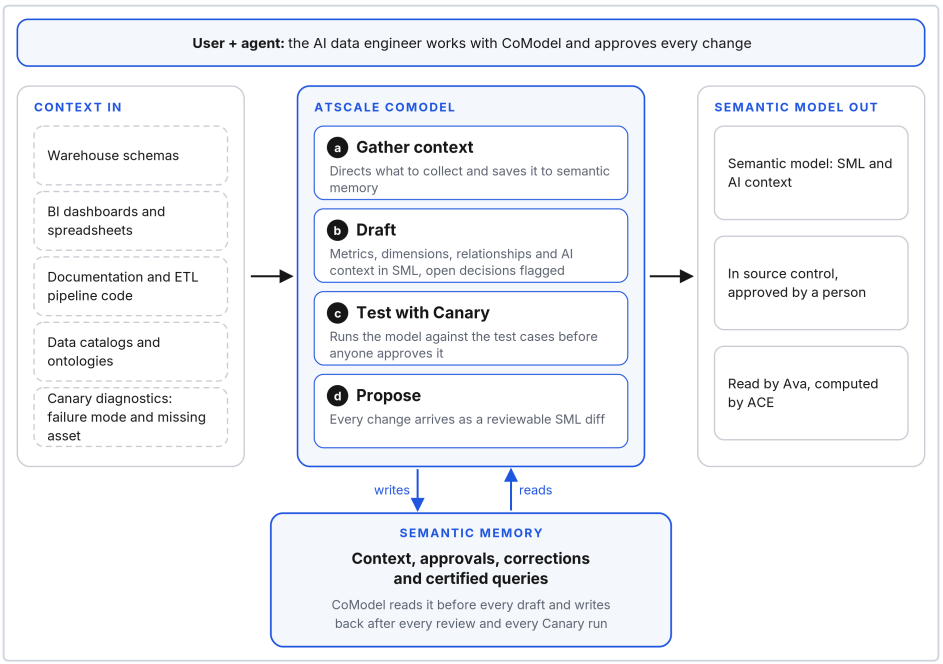


Figure 2. AtScale CoModel. AI agents gather, draft, test and propose; the data engineer approves every change; semantic memory records what each pass learns.

For the BIRD-Interact runs, a CoModel preview modeled all 22 databases in a couple of hours, with no hand tuning.

A model only helps if the question reaching it is clear.

Ava asks which metric you meant

Most wrong answers start at the prompt. “Show me revenue” has five right answers in most enterprises, and a model that picks one on the strength of a column name has already decided the number before any SQL runs. LLM accuracy drops 39% on average when a request arrives underspecified, and models rarely recover from the first wrong reading.[6]

AtScale Ava (Ask, Verify, Answer) resolves the request before anything computes. For a person typing in Grok, it asks which metric they meant and saves the answer as a rule, so the next person isn't asked again. For a Grok Build agent, it skips the question and points the call at the right certified metric, because an agent has no one to ask.

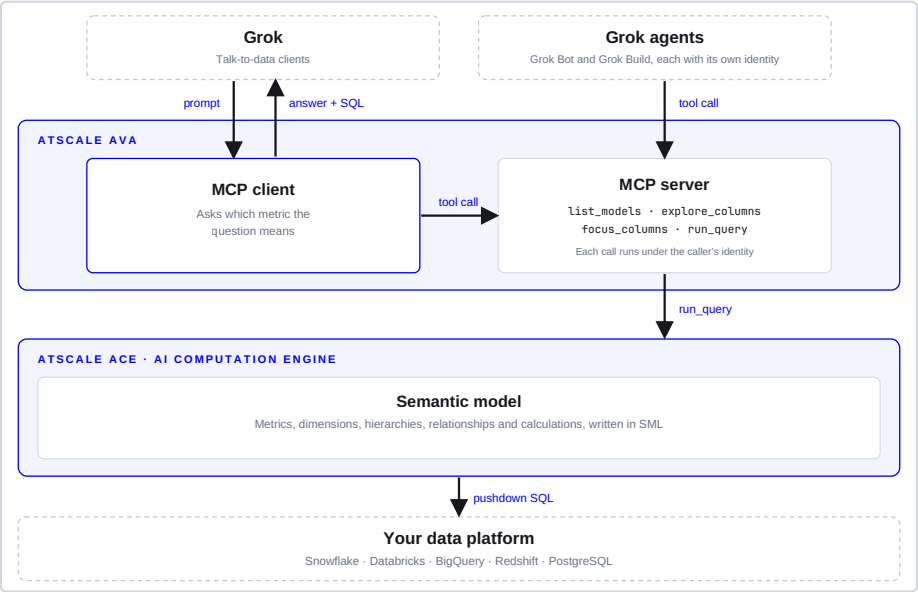


Figure 3. The two runtime paths. A person talking to their data reaches the MCP client inside Ava, which settles the question and then calls the MCP server. Grok Bot and Grok Build are MCP clients themselves and call that server directly. Both arrive at ACE as one resolved metric request.

That one step separates an answer from a guess, and no model can take it alone: the model can't see that five definitions of revenue exist, let alone which one you meant.

Here’s one from the benchmark. A BIRD-Interact task asks for a ranked list of stars in a galaxy. The schema defines three measures that could do the ranking, and the question names none of them.

Without a semantic model the LLM guesses: alphabetical first, then by radius. Each attempt spends from a fixed budget of BIRD coins, and the budget runs out before it tries brightness. Ava reads the three measures off the semantic model and asks which one, so the first query is the right one.

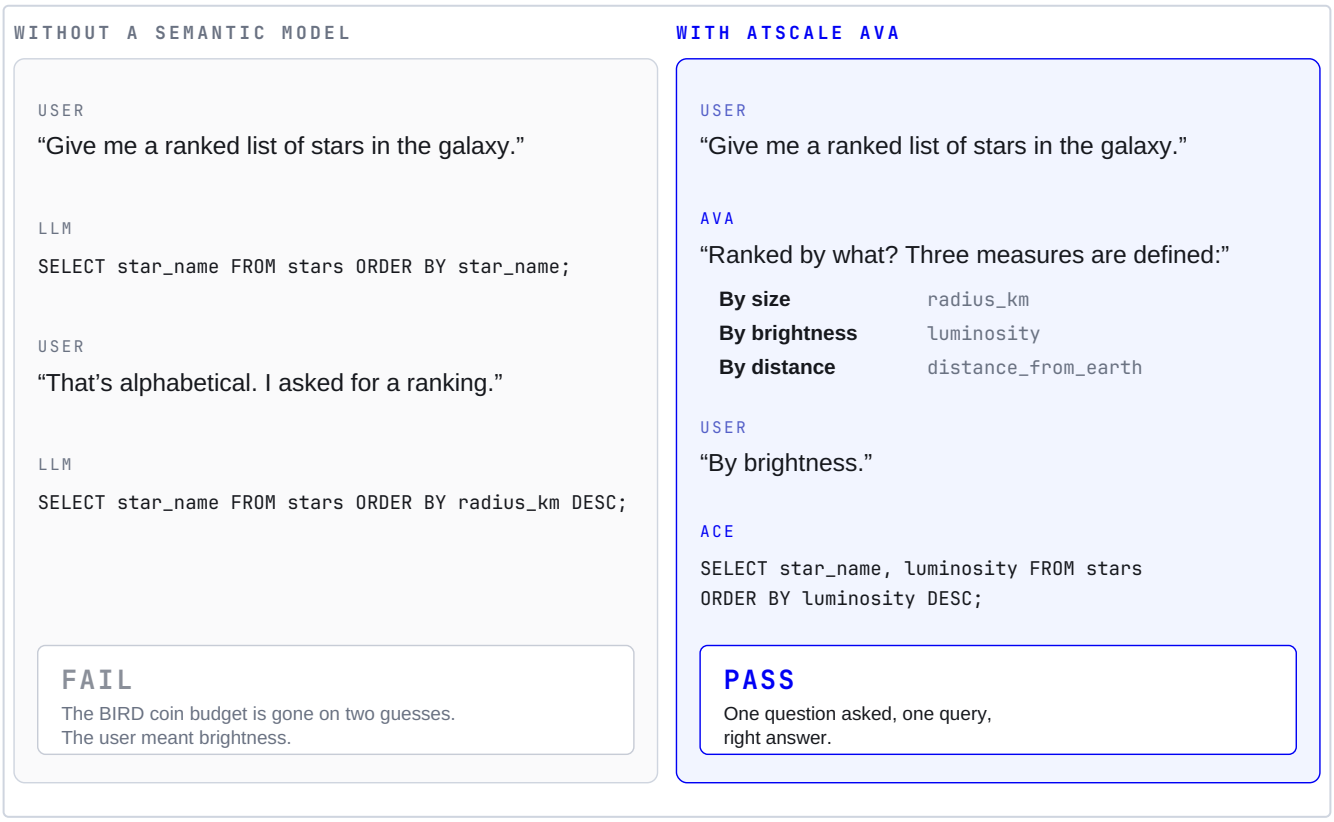


Figure 4. One BIRD-Interact task, run twice. Without a semantic model the LLM spends its budget on two wrong rankings. Ava offers the three measures the model defines, and the first query passes. Adapted from AtScale’s BIRD-Interact results.

Once the question is settled, something has to do the arithmetic.

ACE writes the SQL and computes the metric

When Grok asks for a metric, AtScale ACE (the AI Computation Engine) computes it. ACE compiles the semantic model into a graph, plans every join, grain and pass over that graph, and writes the SQL your platform runs, in your platform's own dialect. Grok never writes it.

That matters most where generated SQL fails hardest: distinct counts, ratios, semi-additive measures like balances, and multi-pass calculations that filter, take a base total and then recompute at every level. ACE also builds and refreshes aggregates in your platform and serves repeat questions from cache, which is why cost per right answer fell about 21% across the seven models we tested.

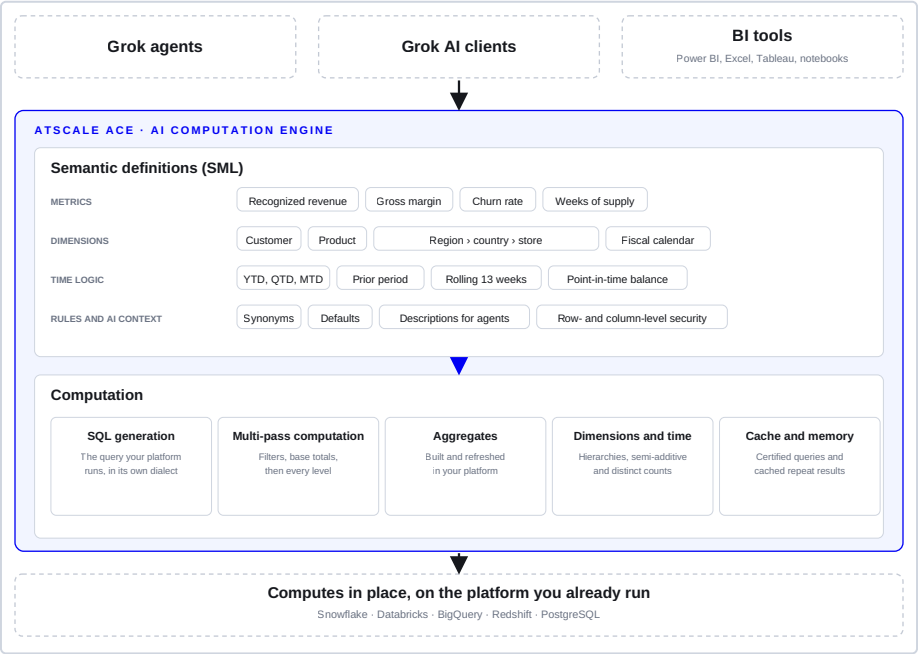


Figure 5. Inside ACE. Definitions on top, computation underneath, and ACE computing in place on the platform you already run.

The result is deterministic: the same question gets the same SQL and the same number, so a Grok-built report can go in front of a board. Whether that number is correct is a separate question, and only a measurement answers it.

Canary measures how often Grok is right

Most teams can't say how often their AI is wrong, which is why a person still reviews everything an agent produces. AtScale Canary gives you a measured error rate on your own Grok prompts, rerun whenever anything changes.

It works the way BIRD-Interact does, against your data. Each test case pairs a prompt, written the way your people and your agents send it, with gold-standard SQL and the result it returns. Canary sends every question down the same path production traffic takes, through Ava and into ACE, scores each one pass or fail, and recommends a fix for every miss. Each miss goes back to CoModel as a proposed change; each query that passes becomes a certified query in semantic memory.

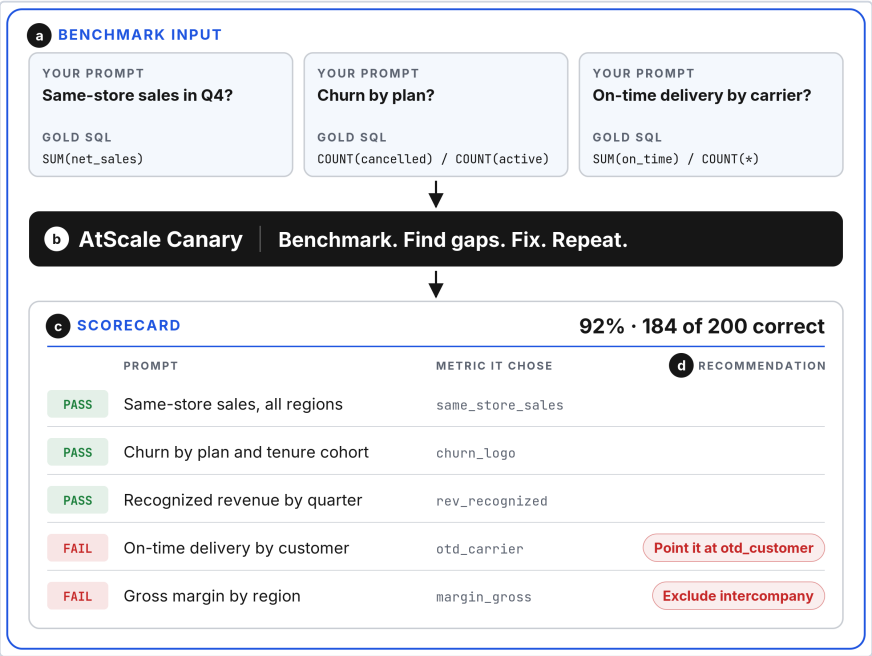


Figure 6. AtScale Canary: questions and gold SQL go in, Canary runs them, a scorecard comes out, and each miss gets a recommended fix. Scores are illustrative.

That closes the loop. Grok asks, Ava resolves, ACE computes, Canary measures, CoModel fixes, and the next turn starts from a better model than the last.

Conclusion

Every company will license the same models. A semantic model of your own business has to be built: from your documentation, your dashboards and your spreadsheets, kept current as the business changes, and measured against the questions your people actually ask. AtScale and SpaceXSI are making that routine for Grok.

Next, AtScale and SpaceXSI take this to shared customers and cover more of the development loop with definitions the business has already approved. The design-partner program for CoModel and Canary is open.

We'd like every enterprise to use Grok on its most important numbers, and to be able to prove they're right.

Bring us your most controversial metric

Bring the business questions your team and your agents ask about it every week. We benchmark your AI on them, with AtScale and without it. You keep the semantic model we built to run the test. Book a workshop.

FAQs and resources

How does Grok Build connect to AtScale?

Through AtScale's MCP server. You add it to Grok Build's MCP configuration and sign in as yourself with OAuth, so your access rules apply to the agent the same way they apply to you. The MCP tools are read-only. The setup guide has the exact steps.

Does it work in Cursor too?

Yes. The same MCP server works in Cursor, so teams moving to Grok Build keep the same setup.

Does the agent still write SQL?

Not for metrics. ACE computes each metric from the definition in your model and runs the SQL itself. You can read the query it used.

What if my prompt could mean more than one metric?

Ava checks which one you meant. If your model has five kinds of revenue, Ava finds all five and confirms the one you need, like recognized revenue for the quarterly report, before ACE computes it.

Which data platforms does it work with?

Snowflake, Databricks, Google BigQuery, Amazon Redshift, PostgreSQL and InterSystems IRIS. ACE computes on the platform you already run, so your data doesn't move. The current list for self-hosted deployments is in the AtScale product documentation.

Which other AI tools can connect?

Any client that supports remote MCP servers works, including Claude, ChatGPT, Google Gemini and Databricks agents alongside Grok Build. They all request the same metric definitions, and the full list is on the AtScale integrations page.

Do I need a public hostname?

Only for Grok on the web and for Grok Bot, because SpaceXSI's servers call your MCP server directly. Grok CLI and Grok Build run on your machine and can reach an instance on localhost. The setup guide covers the hostname, TLS and redirect-URI requirements.

Resources

These cover the install, the benchmark in full, and the architecture beyond Grok. Numbered references in the text point here.

1. **Connect Grok to AtScale**, the setup guide. This is the install, end to end and about ten minutes: turning on the MCP server, finding the client secret, the `config.toml` entry for

Grok CLI, and the custom connector for Grok on the web, with the redirect URI and the TLS requirement spelled out. Read it when you're ready to connect something.

2. **Grok on the AtScale Semantic Layer**, the recorded demo. Grok works through a solar panel business built on BIRD-Interact data, with the semantic layer as its only path to the data, and answers one open question about where the business can improve efficiency. Every query came back in under five milliseconds. Watch it to see Ava's clarifying question and ACE's generated SQL run in a live session.

3. **AtScale for Grok**, the integration page. The shortest statement of what the plugin does, plus the patterns developers build on top of it: models as code, metrics inside your own application, generated SQL that matches hand-written SQL, Grok Bot agents reading the same definitions, and an audit trail that is just Git.

4. **How AI scores on BIRD-Interact**, the full results. Every model we tested, with and without a semantic layer, and the reasoning behind choosing a benchmark whose ceiling is about 53%. Read it if you want the numbers in What the BIRD-Interact benchmark shows in their full context. The runs behind every figure in this guide are AtScale's own, from October 2026: seven models from six vendors, each run with and without an AtScale semantic layer, and Grok 4.3 scored as the mean of three runs on the business questions that have a valid answer key. The numbers are preliminary, the BIRD team has not verified them, and they don't compare with BIRD leaderboard results.

5. **Inside BIRD-Interact**, the methodology. How the benchmark scores an answer, what the corrected regime changes and why, and the 93 questions no correct query can pass. Read it before you quote any figure from this guide to someone who will push back on it.

6. **AtScale for AI: Technical Overview**, the architecture in full. The model, measure, compute, repeat loop beyond Grok: the seven failure points it closes, how the semantic model is written in SML, and where governance sits. This guide covers the Grok path through it. It's also the source of the 39% underspecification figure.

7. **The product pages**: CoModel, Ava, ACE, Canary and the AtScale MCP Server. One page each, for the component you want in more depth than this guide gives it.