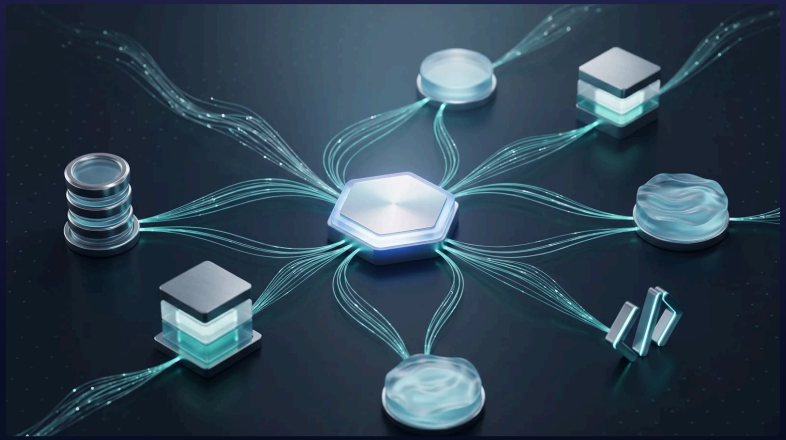
DATA FEDERATION WITH A GOVERNED SEMANTIC LAYER

SemanticFed

Ask one question of all your data, answered where that data already lives.

FEDERATED QUERY · SEMANTIC LAYER · CROSS-SOURCE POLICY · AGENT-READY · QUERY IN PLACE



Pre-launch. Launching 2026, on a waitlist today. Every capability in this document is marked shipped, in progress, or planned.

01 The short version

SemanticFed answers questions across every database, warehouse, lake and business application you own, without first copying all of it into one place.

It registers each of your systems once, then sends the question to the data instead of dragging the data to the question. A planner works out what each system can filter and total on its own, asks it to do exactly that, and joins only the small results that come back.

Above that sits a semantic layer: entities, dimensions and metrics defined once, versioned, and resolved the same way by every dashboard, notebook, script and AI agent that asks.

THE PROMISE

Define it once. Query it where it lives. Open the plan behind any answer you are asked to defend.

1

One definition per metric

Written down once in the published model, so every surface resolves the same thing.

0

Copies made

Queries run in place. A run record keeps a plan and statistics, never the rows.

40

Tools an agent can call

Sources, health, models, metrics, queries, runs and policy — under the caller's permissions.

2026

Launch year

Pre-launch and on a waitlist. No live contracts and no benchmarks to quote.

02 Who it is for

Data platform leads

One governed layer over the estate you already have, rather than a migration project you spend two years defending.

Analytics engineers and BI leads

Define entities, dimensions and metrics once, publish a version, and let every consumer resolve it.

Data engineers who own the sources

A source registry with a validated connectivity lifecycle and recorded health history, so drift is visible before an analyst reports it.

Governance and compliance leads

One policy model at row, column or entity level, plus a run trail you can read as evidence for an access review.

Business analysts

Ask across systems without filing a ticket. Each run keeps a readable plan you can point at when someone questions the number.

AI and platform engineers

An agent that reasons over documented business concepts, under exactly the permissions the calling person already holds.

Security architects, too: authorisation happens at one edge for people, scripts and agents alike, and the gaps that remain are listed rather than glossed.

03 What breaks today

A slow tax: copies that drift, numbers that disagree, governance that fragments at every hop, and an AI layer confidently answering from tables it does not understand.

The data lives in too many places to query together

Questions that would need three systems simply do not get asked.

- Register sources and join them in a single query, with no copy between question and answer.

Every dashboard reports a different number

Meetings become arguments about whose figure is right, and trust erodes one discrepancy at a time.

- Define measures once in a typed, versioned model, so the numbers agree by construction.

Governance re-fragments at every system

Rules drift apart, and an access review becomes archaeology across a dozen consoles.

- One policy model, attached where the meaning lives rather than where the bytes sit.

Nobody can prove where a number came from

A disputed figure is rebuilt by hand, badly, under time pressure.

- Every run keeps a structured plan: sources touched, filters pushed, join order, rows, duration.

04 What you get

A small number of governed objects, each workspace-scoped, permission-checked, and reachable from every surface alike.

CONNECT

Connectors

The reusable driver for a class of system, declaring which filtering and aggregation it can hand down.

REGISTER

Data sources

One concrete system: endpoint details, a credential held by reference, and a five-state lifecycle.

WATCH

Connection health

An append-only trail of verdict, latency and message per probe, reconciling the source's status.

DEFINE

Semantic models

The versioned container for your vocabulary: draft, published, deprecated, archived.

SHAPE

Entities and dimensions

The things you count, bound to real tables, and the ways you slice them, with declared types.

MEASURE

Metrics

A measure declares how it aggregates and how it displays, then resolves identically everywhere.

JOIN

Relationships

Typed joins between entities, including across two different systems.

ASK

Saved queries and runs

Governed query text with parameters and opt-in caching; each run records plan, rows and who ran it.

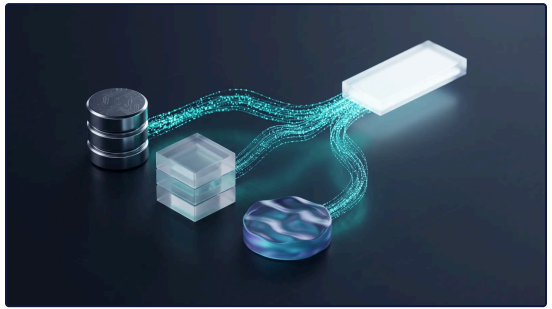
GOVERN

Access policies

Allow or deny at row, column or entity scope. Rules are immutable; the update is the active toggle.

05 Put to work

Illustrative scenarios. SemanticFed is pre-launch, so these describe the experience the platform is built to deliver.

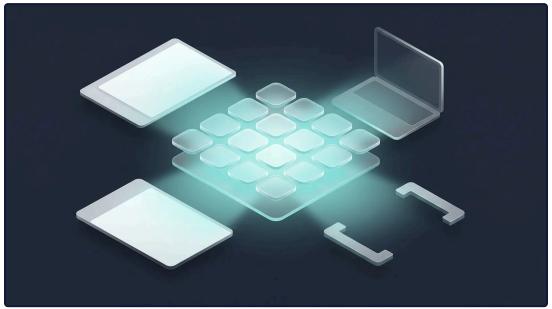


ANALYTICS ENGINEER

Join a live table to warehouse facts and lake files in one query

Register each system once, model the join, and write a single query. The planner pushes down what each system can take.

- **One statement, live operational data, and no nightly job to babysit.**

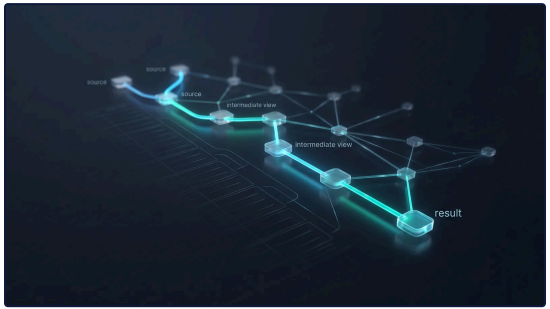


BI LEAD

Define a measure once and have every tool agree

Metrics become typed, documented objects in a published model rather than copied query fragments.

- **Change a definition in one place and every surface updates with the model version.**



COMPLIANCE OFFICER

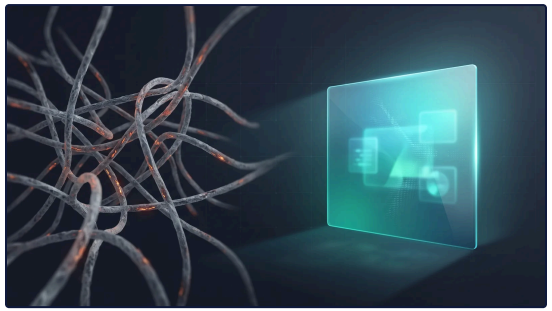
Answer who queried what from a single record

Every run is attributable and recorded with the sources it touched, the plan it followed, rows and duration.

- **Governance becomes evidence you can produce rather than a story you reconstruct.**

05

Put to work, continued

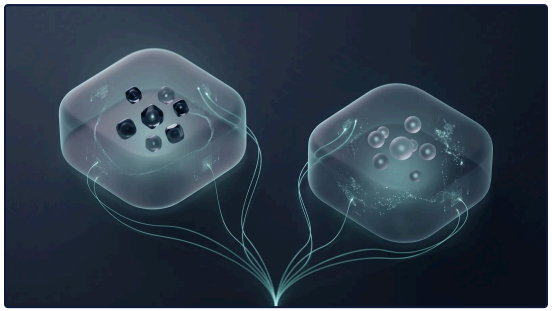


DATA ENGINEER

Retire a pipeline you only ever built to move data

Define the same shape as a virtual view over federated sources. Joins happen at query time, with no materialised copy underneath.

- **Keep pre-computation only where it earns its keep.**

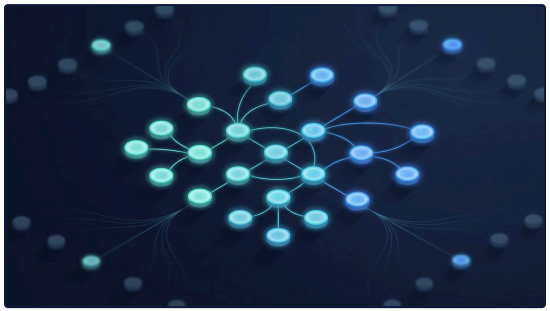


SECURITY ARCHITECT

Keep regulated data inside the border it must stay in

Because work goes to the source instead of rows being exported, a model can span regions and clouds while data stays put.

- **One view of the business without relocating regulated data to get it.**



DATA ARCHITECT

See how everything connects before you change any of it

Modelled entities and typed relationships render as an explorable graph, so you can check blast radius before editing.

- **Impact analysis becomes something you look at rather than something you estimate.**

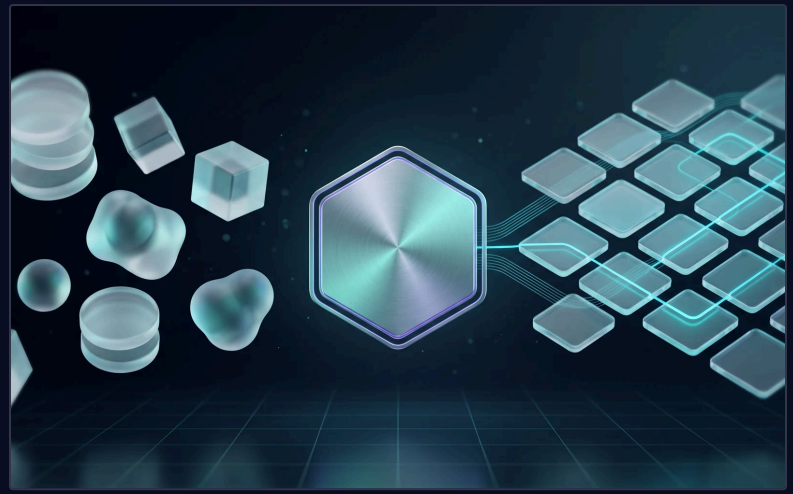
06

Built AI-native

AI that answers from your governed model — and shows its work

Point a model at raw tables and it has to guess: which column is the measure, in which currency, for which population, and which rows this person may see. That is a data-architecture problem, not a prompting one.

Your published model gives the agent something to bind to. A query naming a field the model does not contain fails before any source is contacted, and authorisation happens at one edge — so adding AI adds a caller rather than an exemption.



Answers bind to your published model

Queries name fields that really exist, not ones that merely sound plausible.

Agents get your permissions, not new ones

An agent is another signed-in caller, with no private door and no parallel permission system.

Every answer opens into its plan

Sources touched, filters pushed, join order, and estimate against actual.

An interface agents can already drive

One permission-checked API behind the agent surface, the command line and the SDKs.

07 What the AI can do, and what it cannot yet

SHIPPED

40 agent tools across the whole product

Sources, health, models, entities, metrics, relationships, queries, runs and policy toggles.

Guided recipes, not just a pile of tools

Two sequenced procedures encode the known-good order instead of leaving an agent to improvise one.

Validation before anything is contacted

A hallucinated field is rejected up front rather than silently answered.

A stored plan behind every run

Provenance is a queryable record, not a log line to grep.

IN PROGRESS AND ROADMAP

Confirmation before irreversible agent actions

In progress. Until it lands, grant agent access the way you would grant a write-capable service account.

Row and column policy applied during a run

In progress. Permission decides who may run something; it does not yet decide which rows come back.

Plain-language questions into governed queries

Roadmap. You would see which of your own definitions were used, and edit before running.

Model drafting from a connected source

Roadmap. Publishing stays a human act by design, because it changes what a measure means downstream.

> No model provider is wired into the product today, so bring-your-own-key is neither offered nor promised. The in-app AI page is a placeholder, and demonstration data on some screens is not evidence of AI capability.

08 What keeps it in bounds

One place where permission is decided

Every operation declares the action, resource and scope it needs, enforced at one edge for people, scripts and agents alike.

No agent-only route to data

A restricted person gets identical results whether a query was written by them or generated for them.

Workspace isolation on every write

Writes must match exactly one record in the calling workspace, so an identifier from another tenant resolves to nothing.

Publishing stays a human act

Proposals arrive as drafts. Publishing bumps a version every downstream consumer inherits.

Policy activation is human-only, permanently

Switching a deny policy on could break every legitimate consumer; switching it off could expose what it protects.

Propose by default, act only where reversible

Of three planned loops, only source-health reconciliation may act. The other two propose and stop.

A per-workspace off switch

An autonomous agent without an off switch is a defect. Every planned action is metered and stoppable.

Source metadata is untrusted input

A column comment is data, never an instruction to follow.

Nothing becomes AI-only

Modelling, publishing, querying and policy changes stay doable by hand. Automation is an accelerator, never the only route.

09 Who does what

Six roles are seeded automatically into every workspace. They are deliberately sparse: a role lists only the resources it grants, and omitting one is the access control rather than an oversight.

Federation Owner	The only role with full administrative reach across all eleven resource types: the source estate, the model, saved queries and access policy.
Governance Lead	Writes and toggles row, column and entity rules, and reads the run trail as evidence. Read-only on saved queries, so the person governing access cannot query the data.
Semantic Modeler	Builds and publishes the business vocabulary. Can publish a model version, but cannot archive one, register a source, or touch access policy at all.
Source Engineer	Registers and operates the federated systems and records health probes. Reads the model to judge blast radius without being able to edit it.
Query Analyst	The day-to-day consumer: writes and runs saved queries against the published model, and inspects plan, timing and cache status of every run.
Auditor	Read-only on all eleven resource types, including the record of who ran which query when. Changes nothing.

10 Under the hood

METADATA PLANE

Connectors, registered sources, health history, semantic models with their entities, dimensions, metrics and relationships, saved queries, run records and access policies — each versioned, permission-checked and workspace-scoped.

Shipped, and complete enough to build on today.

RUNTIME PLANE

The engine parses a saved query into a typed request, resolves every field against the published model, then plans one step per source carrying exactly the predicates, projections and aggregations that source can accept — most selective first.

Built and working. Cost estimates are rough, execution is in-process, and live drivers cover relational and REST sources today.

Push-down execution

Queries are parameterised against each system's own server; only results that survive filtering cross the network.

Typed relationships

Joins are declared, not inferred — including between entities bound to two different systems.

Caching by choice

An opt-in freshness window per saved query, and every run records whether it was served from cache.

Runs as provenance

Join order, per-step log, estimate against actual, rows, duration and calling identity — stored structurally.

11 Standards, integrations and deployment

OPEN STANDARDS IT SPEAKS

- GraphQL, as the governed contract for the whole product
- Model Context Protocol, for the published agent surface
- OpenID Connect sign-on and OAuth 2.0 device-code flow
- SQL, as the language saved federated queries are written in
- Parquet and columnar files in object storage, as a first-class source shape
- A SQL wire endpoint for external tools, specified and on the roadmap

PLUGS INTO

- Relational databases — live drivers ship for these today
- Cloud data warehouses, through the connector library
- Object-storage lakes holding columnar files
- REST and GraphQL business applications — REST drivers ship today
- Notebooks, pipelines and build systems, via the SDKs and command line
- Business intelligence tools, once the dedicated SQL endpoint lands

Managed multi-tenant service

The default. Isolation, permissions and multi-tenancy are structural rather than configured.

Dedicated or self-hosted

Enterprise shapes for regulated environments. Pre-launch, so treat availability as a conversation.

Regional hosting

Hosted in India, with selected services in Mumbai. Queries run in place, so regulated data stays inside its boundary.

12 Security and trust

Mostly architectural rather than procedural: one enforcement edge, credentials held by reference, results never persisted, and a run trail that reads as evidence.

Authorisation at a single edge

Every query and mutation carries an explicit action, resource and scope requirement, enforced before the request reaches the service.

Credentials by reference only

A source stores an opaque reference to a managed secret, never an inline one, so reading its configuration discloses nothing.

Federated results are never stored

A run record keeps statistics and a plan. The rows returned are not persisted, so the audit trail cannot become a second copy.

Sparse roles, deliberately

Only the Federation Owner and Governance Lead hold any write level on access policy, so no other role can weaken a mask.

Immutable policy rules

Effect, scope, principal and rule are fixed once written; correcting one means writing a replacement, which leaves the original in the record.

Metadata only, verified

A sensitivity review of the eleven enforced resource types found only federation metadata: configuration, definitions, query text, statistics and rules.

STATED PLAINLY

SemanticFed holds no third-party certifications. SOC 2 and ISO 27001 are on the roadmap, and no compliance badge should be inferred from anything here. Row and column policies are stored and toggled but not yet applied while a query runs; irreversible agent operations are not confirmation-gated. Both are tracked and active work.

13

One product, many front doors

One governed platform reached from several places. Because it is pre-launch, most of these are listed as planned rather than described as if you could install them today.

SHIPPED

Agent surface

A published Model Context Protocol server: 40 tools, two guided recipes and your live catalogue as readable resources.

SHIPPED

Command line

Drive connectors, sources, models, queries and policies from a terminal or a build script.

SHIPPED

Governed API

One typed contract over the whole product, where every read and write carries its own permission requirement.

PLANNED

Web console

Discovery, the semantic modeller, a query workbench, policy and lineage authoring, and the audit trail in one application.

PLANNED

Node and Python SDKs

Typed clients for pipelines, notebooks and scheduled automation, with token refresh and retry with backoff.

PLANNED

Editor, browser and desktop

Extensions bringing models, lineage and policy where the work happens, plus a local connector runtime for unreachable systems.

Embeddable components, mobile applications and documentation at docs.semanticfed.com complete the surface.

14

Plans

Transparent flat fees with a permanently free entry point. The bands below are the published direction, not a contract: regional prices are being settled ahead of the 2026 launch.

FREE	STARTER	GROWTH	BUSINESS	ENTERPRISE
$0	**$99/mo**	**$399/mo**	**$999/mo**	**~$9 per source**
Individuals and small teams exploring federation	Small data teams running their first federated models	Scaling teams and platform pilots	Larger teams standardising on one federation layer	Large or regulated organisations, billed annually
3 sources, 1 semantic model, and the full product surface. Permanently free rather than a trial.	Higher limits, scheduled caching for hot paths, the SDKs and the agent surface.	Up to 250 sources, the full governance suite, priority support. Predictable cost regardless of query load.	Up to 1,000 sources, higher caching capacity, support with a service-level commitment.	Custom quotas, single sign-on, dedicated tenancy or self-hosting, negotiated service levels.

The same platform on every band. What scales is source and query volume, caching capacity, governance depth and support tier.

Four regional bands including rupee pricing are planned. At the bottom of a band the effective per-source cost is higher: a 60-source estate on Growth pays the same as a 250-source one.

15

Why teams choose it

The combination, not any single piece

Federation engines lack a typed semantic layer; semantic layers lean on a warehouse you bring and pay for. SemanticFed sits at the intersection, with one bill.

A typed, versioned model

Entities, dimensions, metrics and relationships are explicit objects with lifecycle states and a version that bumps on publish, not a folder of query fragments.

The plan is a stored artefact

Most tools make provenance something you reconstruct. Here it is a record that already exists before anyone asks.

An agent surface that is not a bolt-on

40 tools, two guided recipes and readable catalogue resources, published and callable today over the same permission checks a person meets.

Transparent flat fees

Published bands from free to enterprise, rather than compute-metered credits that spike with query load or six-figure quote-only pricing.

Honest status labels on every capability

Shipped, in progress or roadmap, in this deck, on the website and in the product. The alternative is discovering the gaps during your pilot.

Honest answers to hard questions

You are pre-launch. Why would I bet on this?	You should not bet a critical production path on it yet. What is real is the built metadata plane, the working query engine with its planner and run records, and the published agent surface. The sensible engagement is an evaluation against a couple of real systems, not a migration.
Policies are cross-source, but not enforced at run time. Which is it?	Both. The policy model is shipped: allow and deny at row, column or entity scope, individually toggled. Applying those rules while a query executes is not built, so today permission decides who may run a query rather than which rows come back.
Federated queries are slow. I have been burned before.	You were probably burned by an engine that pulled whole tables across the network. The planner separates the work each source can take from the work that must happen after, and runs the most selective source first. The caveats: cost estimates are rough, execution is in-process, and live drivers cover relational and REST sources.
What happens to my metrics if we leave?	Your data never moved, so there is nothing to migrate back. Your sources remain yours, connected with scoped credentials you issued and can revoke. What you would lose is the governed model, which is why it is defined as explicit typed objects reachable through a documented interface.

17

Where it sits against the alternatives

VS ENTERPRISE DATA VIRTUALISATION

Incumbents have broader connectors, decades of optimiser work and multi-tier caching we do not match. They typically lack a typed semantic layer, an open agent surface, and any published price. We trade connector breadth for time-to-value and a number you can read on a page.

VS DISTRIBUTED SQL ENGINES

They win decisively on raw performance at very large scale, and if that is your binding constraint you should buy one. They are compute-metered and leave the semantic layer to you. Our answer is a flat fee and one governed model, not faster execution.

VS STANDALONE SEMANTIC LAYERS

Several are excellent at modelling and free for technical teams, which we cannot beat on price. They execute through a warehouse you bring, so federation is not theirs to solve. We include the engine, at the cost of a younger modelling ecosystem.

VS WAREHOUSE AND LAKEHOUSE PLATFORMS

If your data genuinely all belongs in one platform, consolidating there is coherent and its governance is mature. It is the opposite bet: it assumes the copies are worth it. We are engine-agnostic and visibly younger in operational tooling.

VS HAND-BUILT PIPELINES

Pipelines you already run are paid for and understood. They are also what generates the drift and the operational load. Our claim is that a large share exist only to move data that never needed moving, and that is where to start.

VS AI COPILOTS ON A DASHBOARD

They ship a chat window today, which we do not. What they usually cannot do is tell you why a number is what it is, or guarantee the agent met the same permission checks you would.

START TODAY

See whether federation actually fits your estate

SemanticFed launches in 2026 and is in waitlist mode today. The most useful next step is a short evaluation against two real systems rather than a procurement conversation.

- Join the waitlist and tell us which systems you would connect first
- Read the capability list with its shipped, in-progress and roadmap markers
- If the agent surface is what interests you, it is published and callable now
- Bring your hardest question: the one that needs three systems and a week

WAITLIST
semanticfed.com/waitlist

PRODUCT
semanticfed.com

AI CAPABILITIES, WITH HONEST STATUS
semanticfed.com/ai

DOCUMENTATION
docs.semanticfed.com