

Agentic Data Platform

Data management, done by agents. Governed by design.

One agentic platform for the whole data lifecycle
— understand, design, build, master and consume.
Running on your own infrastructure, where your
data already lives.



From our customers — numbers tracked per build by the platform, as of Q3 2026

2 hrs

First data product live after
deployment — eCommerce pilot

3 days

All 12 data products
delivered, against ~14 days
estimated by hand

2 days

Five governed, AI-ready
data products — public-
sector pilot

+35%

Revenue growth with AI agents
running on the governed data
products

Data management, redefined by agents.

Today, delivering data takes three budgets — integration, quality, governance — plus the engineers who stitch them together. And data leaders are still asked for two things at once: deliver data and AI faster, and prove more control over quality, lineage and privacy. Celvari replaces that stack with one agentic operating model: agents understand your estate, design domains, build and test data products, master golden records and answer questions — **on your database.**

This is not a copilot bolted onto a thirty-year-old engine. Copilots suggest; your engineers still build, test, document and ship. Celvari's agents do the work end to end and stop only for the decisions that matter — confirm the design, approve the deploy. Governance is produced by the build, your data never leaves your perimeter, and you deliver in days.

What your organization gets back

Lower TCO

Three tool budgets — ETL, quality, governance — collapse into one platform. Engineering billed by the hour becomes agent work.

Backlog cleared

Weeks of queue become same-day delivery. Engineers move from building every pipeline to reviewing them.

Models and master data in sessions

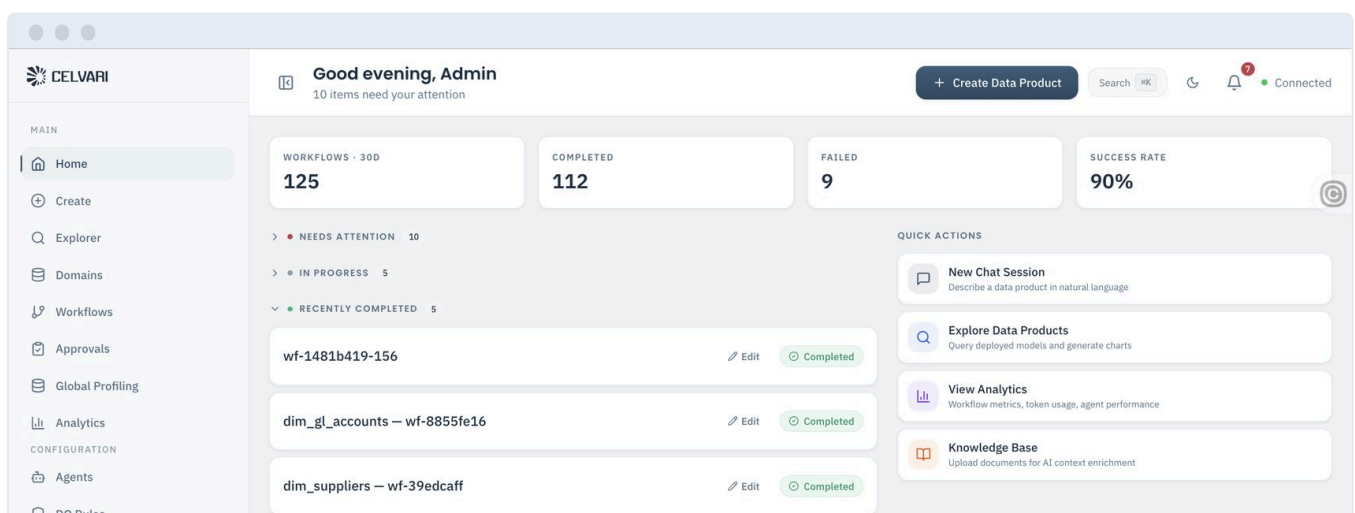
Domains, dimensional models and master data records built with agents in working sessions, your team deciding at every gate — not consulting quarters.

Governance without a program

Lineage, classification and approval evidence produced by the build, mapped to your data governance framework. Audit becomes an export.

Sovereignty, no lock-in

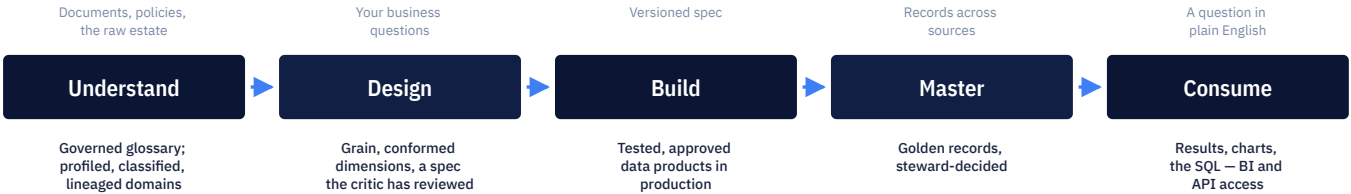
Data never leaves. Any model, any engine, an open catalog you own. Walk away tomorrow and keep all of it.



The Celvari workspace. Workflows, approvals, domains, global profiling, DQ and assertion rules, agents, the knowledge base and extracted terms in one place — with every build tracked and every decision queued for a named person.

Seven workflows. One governed platform.

Each stage takes what the previous one produced and hands the next something governed — gray is what goes in, dark is what comes out — with everything written to one open catalog you own.



01 Knowledge ingestion

Agents read your dictionaries, policies and specs; extract terms and synonyms with a confidence score and link them to your data catalog.

02 Profiling & classification

Every column profiled and classified; PII and sensitive data flagged automatically at column grain.

03 Domain building

Your estate organized into governed business domains — every table classified, owned, tagged, searchable, with column-level lineage.

04 Domain design

Design agents start from your business questions, declare grain in plain English, prove it against profiled data and lay out conformed dimensions. A critic agent reviews first.

05 Data product build

Intent → versioned spec → quality gate → generated, tested SQL → one human approval → production. Edit-by-chat re-runs the governed path.

06 Master data beta

Agents match records across sources, draft survivorship from your rules, route low-confidence pairs to a steward and publish golden records.

07 Explore & ask

Ask deployed products questions in natural language — results, charts and the SQL behind them. BI and API access.

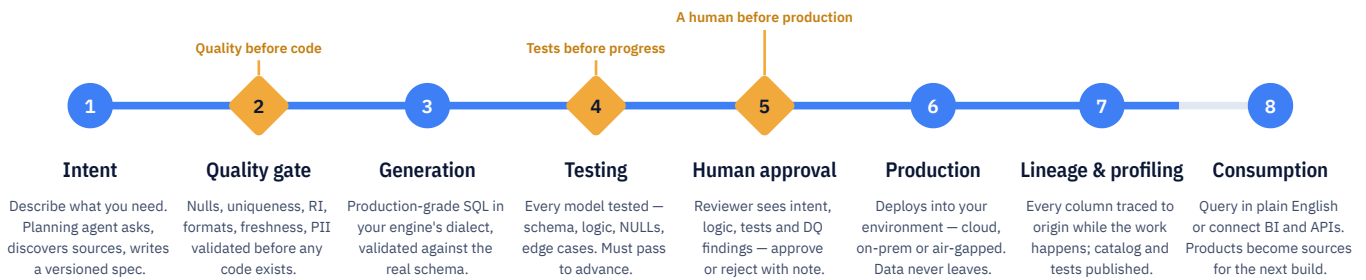
How this differs from what you run today

	Coding copilot in the IDE	ETL + quality + catalog stack	Celvari
Who builds	Your engineer, with suggestions	Your engineers, tool by tool	Agents end to end, with a human at every gate
Who tests and documents	Your engineer, manually	Separate tools, separate teams	Every build — tests, lineage and glossary are produced by the work
Where governance comes from	Not addressed	A program run beside delivery	Gates inside the path: quality before code, tests before progress, a human before production
Where your data goes	Stays put	Often copied into vendor runtimes	Stays put — the work comes to your engine
What you keep if you leave	Whatever was written	Metadata locked to each tool	Standard SQL models, pipeline engine, an open catalog

The governed path: building a data product.

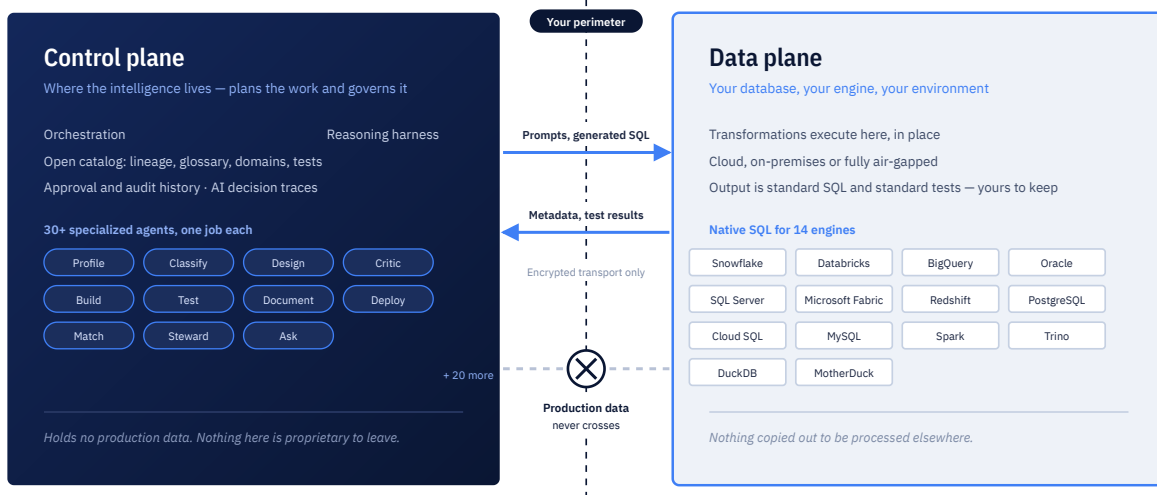
Workflow 05 shows the model at full stretch: eight steps, three of them hard gates — quality before code, tests before progress, a human before production. One call, start to finish.

● Agents do the work ● Gate — nothing passes without it



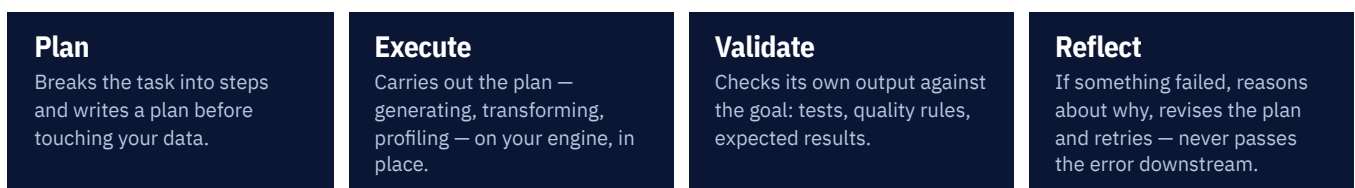
Your data stays. The work comes to it.

Most platforms make you ship your data to where the processing lives. Celvari does the opposite. A control plane holds the agents, orchestration, reasoning and catalog; the data plane is your own database, where transformations execute in place. Only metadata, generated SQL and prompts cross the line, over encrypted transport. Production data never does. Celvari runs alongside the ETL/ELT you already own — an on-ramp, not a forklift.



One reasoning harness behind every agent

30+ specialized agents each own one job. Every one runs the same loop, none can skip a quality or testing gate, and every step — prompt, model, reasoning, result — is logged and replayable. Models are assigned per agent and switchable: cloud, private or local.



Controls you don't have to build.

Governance is not a program beside delivery — it is produced by delivery. Lineage, classification, quality evidence, approval history and AI traces exist the moment a product exists, mapped to whichever data governance framework you are measured against.

Your data stays in your environment and is never used to train any model. Access is governed by your identity provider and the same role-based controls as the rest of the platform, so people and agents see only what they are entitled to.

Data quality gate

Centrally defined, severity-classified rules — nulls, duplicates, referential integrity, formats, freshness, PII patterns — enforced before code generation. Audit trail per build.

Human approval

Explicit approval before every production deployment, recorded with reviewer name and comments in a durable, replayable history.

Classification & privacy

Automatic column classification and PII tagging; glossary terms and ownership published to the open catalog.

Enterprise SSO

SAML 2.0 and OIDC (Okta and others) with automatic user creation and role mapping from your identity provider.

Column-grain lineage

Generated automatically as the work happens; every deployed column traced end to end to its source.

AI decision tracing

Every agent step captured and replayable; Langfuse and OpenTelemetry-compatible traces for your LLM-governance stack.

Role-based access

Admin, Operator and Viewer roles enforced across the platform, for people and for agents.

Encrypted credentials

API keys and connection secrets encrypted at rest and held in your secret store; every connection change tested before going live.

What your risk and compliance team will ask

Can an agent ship wrong or destructive logic to production?

No. Controls compound: the quality gate stops bad data before generation, every model must pass its tests, transformations run as deterministic, versioned SQL on your engine, and nothing deploys without a named human approval.

Will a regulator accept AI-generated lineage and evidence?

The evidence is recorded from the work, not generated by a model: every column's lineage, every quality check, every approval with reviewer and comments, every agent decision — captured as it happens and replayable on demand.

Where does our data go, and is it used to train anything?

Nowhere, and never. Production data stays in your database; only metadata, generated SQL and prompts cross to the control plane, over encrypted transport. Your knowledge base is retrieval context, not training data.

What happens if we stop using Celvari?

You keep everything. Data products are standard SQL on your engine, tests are standard test files, and lineage, glossary and domains live in an open catalog you own. There is no proprietary runtime to migrate off.

Who is accountable when generated SQL is wrong?

The named approver — the same person who would sign off on an engineer's work today. The difference is what they sign with: the intent, the generated logic, the test results and the quality findings side by side, and a replayable trace of every agent decision behind them. Accountability stays where your policy already puts it; the evidence gets better.

Open foundations. Your infrastructure.

Built on tools your engineers already trust. Where Celvari runs and how you procure it are separate decisions — deployment mode is a configuration choice, not a different product or a different price.

Layer	Today	Direction
Catalog	Open catalog on an open standard — you own it: lineage, glossary, domains, tests, metrics	Catalog-neutral; more catalogs over time
Source engines	Native SQL for 14 engines: Snowflake, Databricks, BigQuery, Oracle, SQL Server, Microsoft Fabric, Redshift, PostgreSQL, Cloud SQL for Postgres, MySQL, Apache Spark, Trino, DuckDB, MotherDuck	More engines on the roadmap
AI models	Open-weight, OpenAI, Anthropic, Google or any compatible endpoint — switchable per agent; cloud, private or local	Model-neutral, always
Identity	SAML 2.0 / OIDC SSO (Okta and others); Admin, Operator, Viewer roles	
Orchestration	Agentic, durable and event-driven workflow execution	
AI observability	Langfuse and OpenTelemetry-compatible tracing, per decision	More LLM-governance integrations
Deployment	Container-scaled on your infrastructure: on-premises, hybrid, cloud or fully air-gapped	Marketplace availability expanding

Direct license

On your own tenancy

License Celvari directly and run it on-premises, in your private cloud or fully air-gapped. Neither your data nor the platform leaves your estate.

Your cloud

In the cloud you already run

Deploy into your own cloud account, next to the warehouse or lakehouse you use today. Same governance, same in-place execution on your engine.

Marketplace

Through a cloud marketplace

Procure through a marketplace listing and draw down against your existing cloud commitment. Available today on sccc by stc Marketplace; availability expanding.

Licensed once, supported annually

Celvari is sold as a one-time platform license with an annual support agreement covering updates, patches and assistance — not a consumption or per-build subscription. The license is the same whichever deployment path you choose; the marketplace route lets you procure it against existing cloud commitments.

How a pilot runs

Connect a scoped service account to your engine.
Upload the dictionaries and policies you already have.
Build your worst backlog item live. Run the pilot on your backlog with your team approving every deploy.
Compare the numbers and decide — expand, or walk away and keep everything built.

A typical live build — monthly churn by region, from CRM and billing

3 models	~412 lines of SQL models	31/31 tests passed	14 DQ checks	Hours end to end, request to production
--------------------	------------------------------------	------------------------------	------------------------	---

Bring your worst backlog item.

In a 60–90 minute working session we take it from plain-language request to a governed, tested, approved data product — live, on your data, not a canned script — then map what you saw against the rest of your backlog and your governance obligations.

Book a working session

Gamal Badr · Founding Partner

gamal.badr@celvari.ai

www.celvari.ai

Riyadh · Cairo