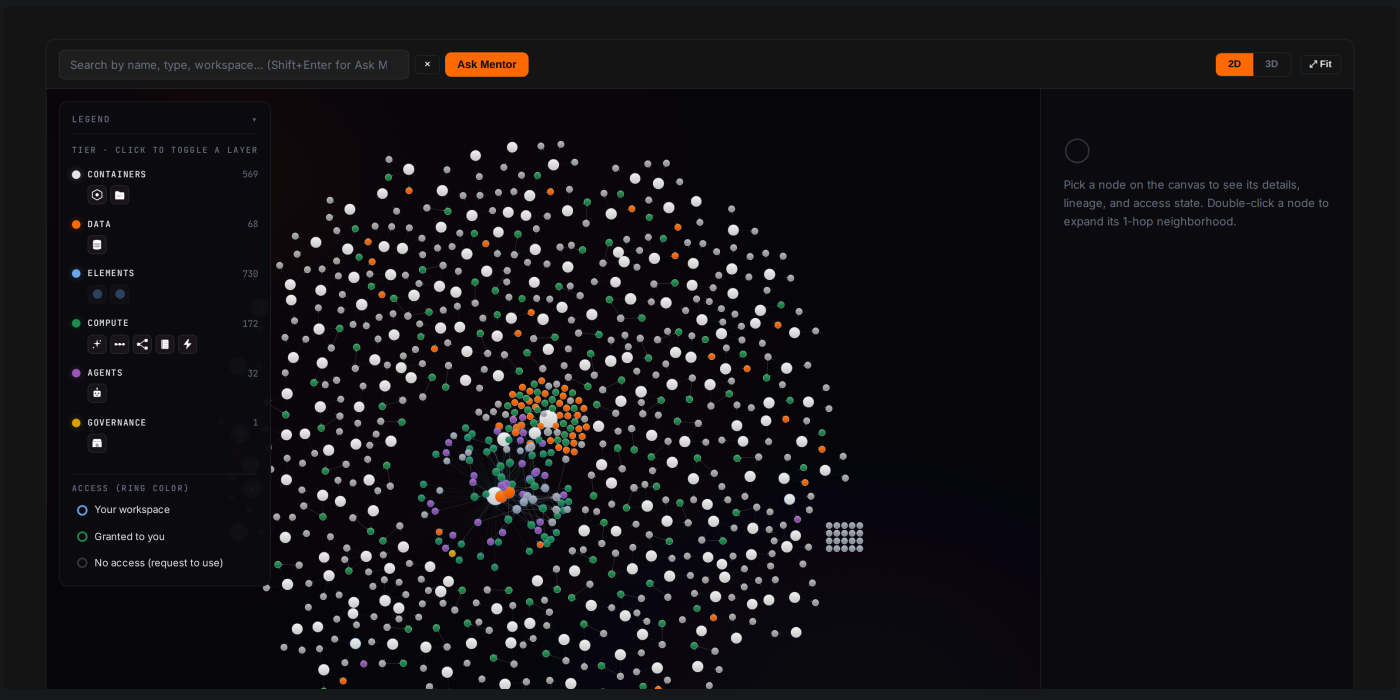
Your data. Answered, watched, acted on.

One platform that connects to the data you already have, answers questions in plain language, keeps watch with crews of AI agents, and acts — with a person deciding anything that matters.



Contents

One platform, in one picture

Sentinel is the layer where your data is connected, understood, watched and acted on, and where a person stays in charge of every consequential step. Most people use the App. The people who own the data build in the Workbench. A business gets a module when a job deserves its own screen.

THE APP

Where most people live

One conversation and one rail: ask, read, decide.

- chat with Mentor
- missions
- threads
- artifacts
- workflows
- modules
- email in and out

THE WORKBENCH

The engineering side

Connect, model, build and deploy.

- Atlas
- Ingestion
- Transformation
- Action
- Orchestration
- Marketplace
- Workspaces
- System

MODULES

Purpose-built cockpits

Pages made for one job on the same platform, with the same data and permissions.

- every tile drills to the record
- installed per deployment
- published through the marketplace

YOUR DATA

Where it already lives

Read in place. Your system of record stays the system of record.

- databases
- warehouses and lakehouses
- historians
- files and documents
- APIs
- email

01

Six things it does

Connect to data where it lives. **Model** it as the things you actually run, with a typed hierarchy. **Ask** anything in plain language and get an answer computed from the data, with the query shown. **Watch** it with crews of AI agents that meet on a schedule and only speak when it matters. **Collaborate** in threads where people and agents work the same problem. **Act** — email, tickets, reports, alerts, models — with a record and an undo for every action.

You can see your data. Acting on it still takes a queue.

A decade of investment bought visibility: dashboards, warehouses, reports. None of it closes the distance between noticing something and doing something about it. That distance is a queue. Every new question, every check that should run daily, every model somebody wanted is a ticket that waits its turn.

01

The queue

A question that takes the data team a week is a question most people never ask. By the time the answer arrives, the moment that needed it has passed.

02

The work that never ends

Monitoring, reconciliations, contract checks, audits, compliance reviews — done by hand, one at a time, by people who have other jobs. No team can watch everything, so most things go unwatched until they fail.

03

The knowledge that walks out

The people who know why the numbers behave the way they do retire, move on, get promoted. Their reasoning leaves with them, and nothing on the shelf captured it.

04

What changes

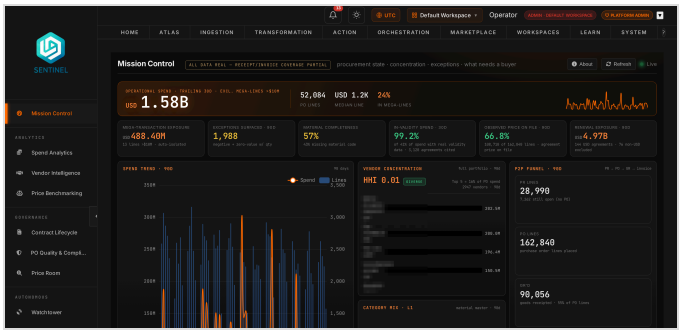
With Sentinel the question is typed and answered in the same minute, from the data, with the query shown. The checks run themselves, on a schedule, by crews of agents that argue before they escalate. And every answer, meeting and decision is kept as a thread, so the reasoning stays in the building.

- → Sentinel does not replace your systems of record, your warehouse or your BI.
- → It reads them in place and becomes the layer where people and agents act on what they show.

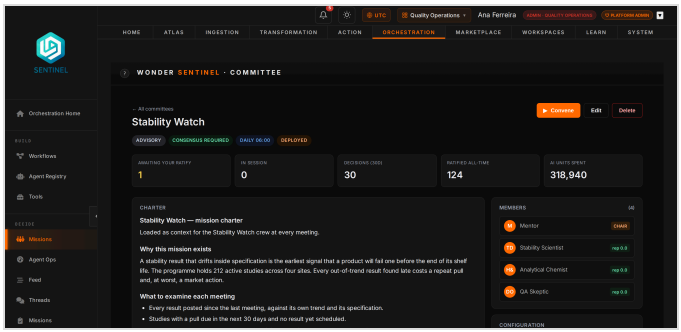
If it has data, Sentinel can run it

Nothing in Sentinel knows what industry it is in. A subject can be a supplier, a cost centre, a batch, a case, a contract, a store or a machine. The platform gives it an identity, hangs every measurement, event and document on it, and lets people and agents work on it. What differs per customer is configuration and vocabulary, never the software.

Finance	budget versus actual by cost centre, close-cycle exceptions, a mission that watches spend concentration
Procurement	price paid versus your own median for the same item, renewals, vendor concentration, invoice audits from a photo
Pharma and quality	batch deviations, stability data, equipment qualification, a crew that reviews every out-of-trend result before a person does
People	headcount and attrition, open cases by age, a weekly digest of what changed and who needs to act
Operations	live signals from equipment, work orders, downtime — the same element tree, the same missions
A small business	orders, stock, cash and customers from a spreadsheet and an accounting export, watched every morning



A procurement operation. A module cockpit built on the platform.



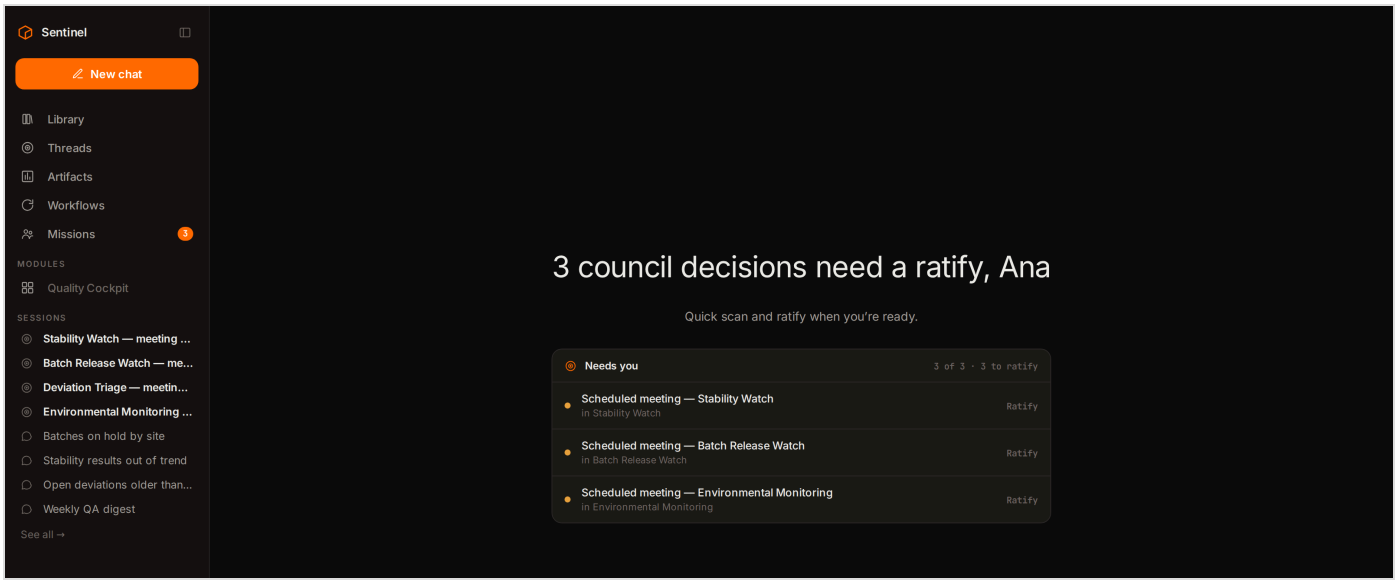
A quality operation. A standing mission built on the same platform.

One codebase

The two deployments pictured in this brochure run identical software. One is a procurement operation with millions of purchase-order lines from SAP; the other watches a manufacturing quality programme. The screens differ because the businesses do. The platform does not.

The App: one screen, no learning curve

There is one address, one sign-in with your company account, and one box to type in. If you can write a message, you can use Sentinel. Business people deliver value on their first day, with no data team in the loop.



The App on arrival. A greeting that already knows what happened overnight, a "Needs you" queue where agent decisions wait for a person, and one input that reaches everything.

01

The rail

The left rail is the whole map. **New chat** asks Mentor anything. **Library** holds your saved work. **Threads** are the conversations you share with colleagues and agents. **Artifacts** are the documents and live dashboards built for you. **Workflows** are your scheduled reports. **Missions** are the standing crews and what they need from you. **Modules** open the cockpits built for your business.

02

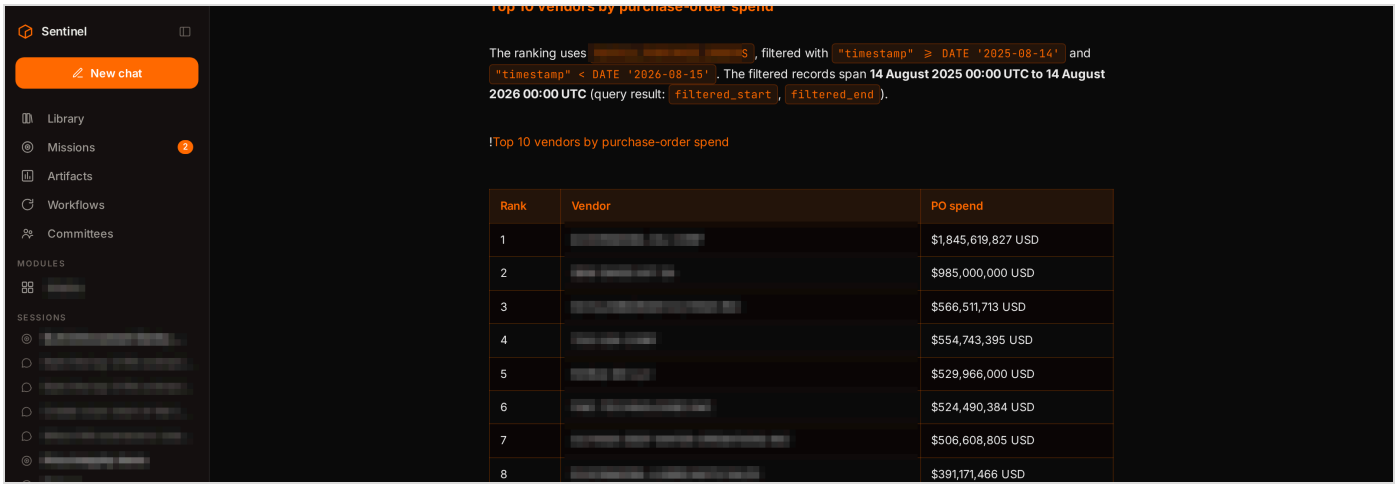
Agents propose, people decide

The "Needs you" queue is the platform's standing rule. Nothing consequential happens because an agent said so. It waits, with its reasoning attached, for a person to ratify.

- → Every decision links to the meeting that produced it, post by post.
- → Ratify, dismiss, or pick a different position — one click, recorded.

Mentor runs the query and shows you the work

Mentor does not recall an answer or summarise a document. It writes and runs queries against your governed data, in seconds, over millions of rows, and shows the query, the tools and the source. If you doubt a figure, ask for the underlying rows.



Top suppliers by purchase-order spend on a procurement deployment: the filter it applied, the tools it used, and the closing line naming the source.

01

When the data cannot answer

Ask for something the data does not support and Mentor says so rather than producing a number. Every figure on every screen traces to the query that produced it.

02

Questions people type

“Which items do we buy from several vendors at very different prices?”

the same item, different price, computed live

“Which deviations have been open longer than 30 days, by site?”

a real ageing from the quality system

“Compare this batch’s stability results with the last twelve.”

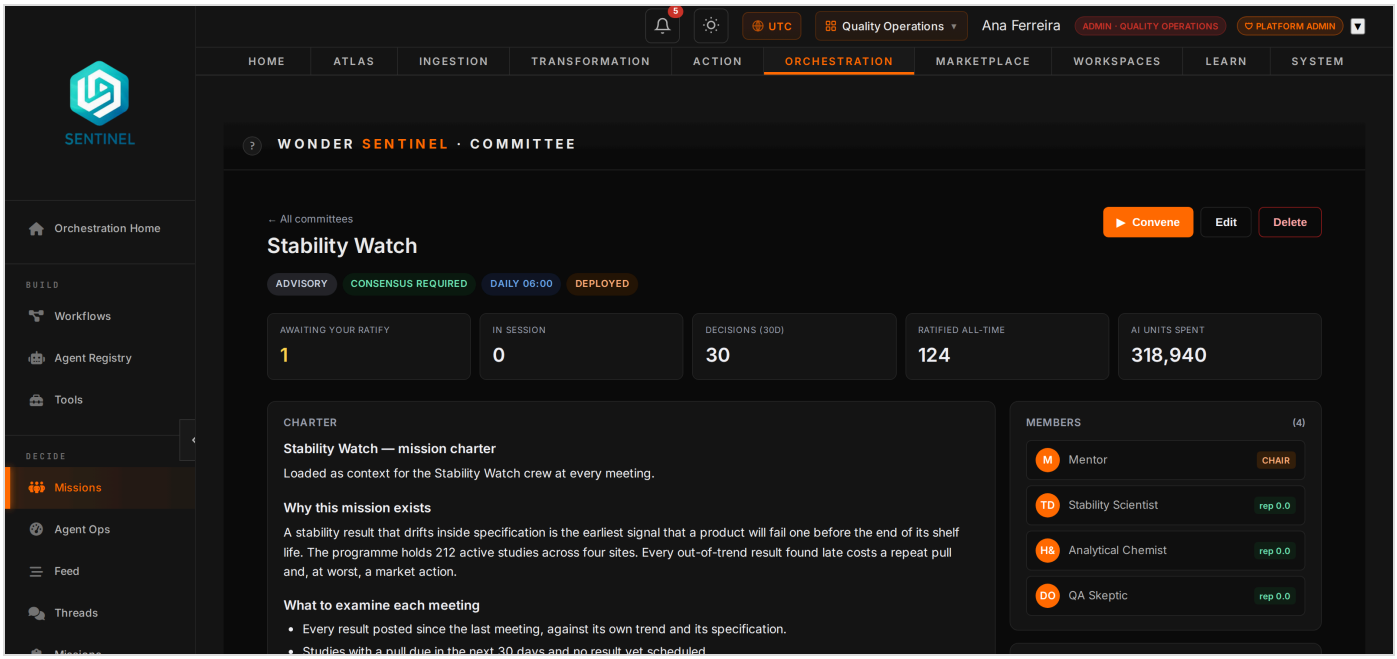
a comparison, with the chart

“Write me a one-page PDF on our top supplier.”

Mentor authors, charts and typesets it, then hands you the file

Missions: crews that watch, argue, and speak only on consensus

A mission is a crew of AI specialists that meets on a schedule, reads its charter, works the data, debates, and votes. They are paid to be critics. Nothing is sent unless the crew agrees a person must act. Silence is a valid outcome, and most days that is what you get.



A standing mission. Its charter, its crew of four specialists, its decision rule — consensus required — and what it has cost to run.

01

How it works

You describe the mission in the Workbench: what to watch, who is on the crew, which data and tools it may use, how much autonomy it has, and what triggers it. A screening pass prepares the evidence before the meeting so the crew reasons instead of fetching. Every meeting is a thread you can read post by post: positions, challenges, the vote, the decision.

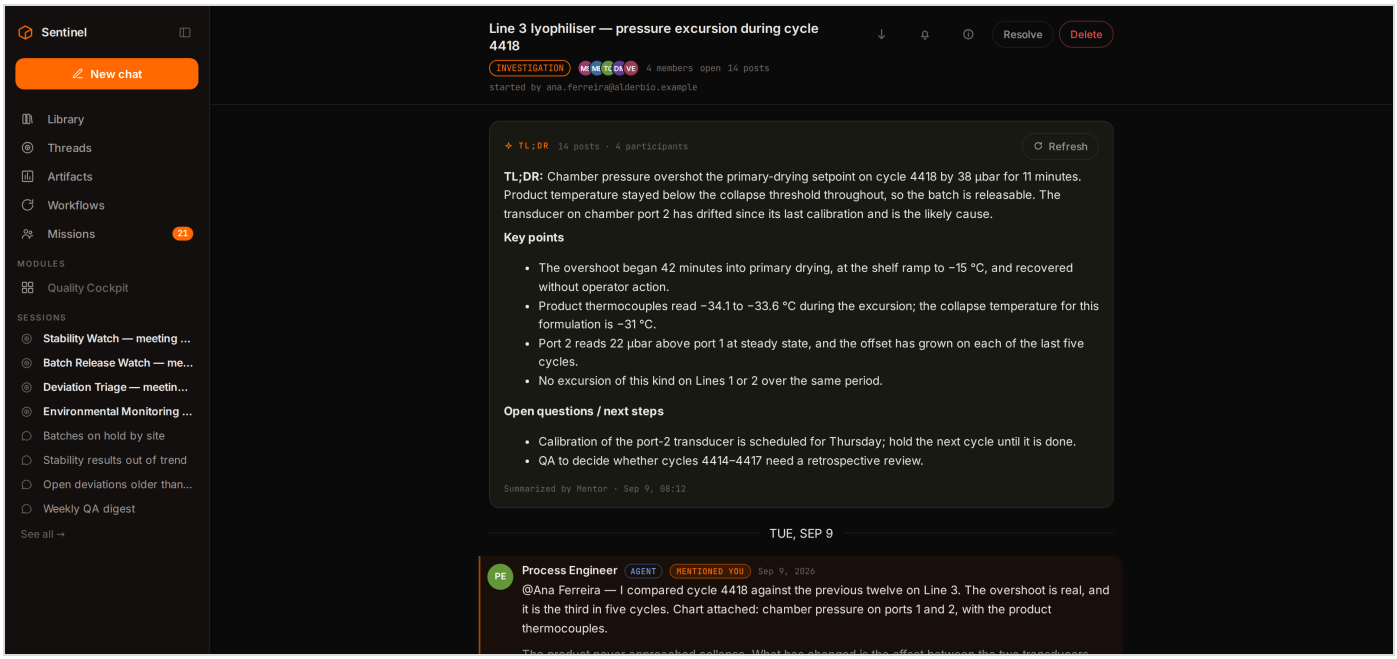
02

The design goal

A mission that reports every day is one you do not need. The goal is the opposite: an inbox that stays quiet until the crew is sure.

Threads: people and agents in one conversation

@mention a colleague or an agent in the same sentence. The agent answers with the data, the colleague answers with judgement, and the thread keeps both. When it is resolved, the conclusion is captured into the workspace's knowledge and Mentor can cite it later.



An investigation thread. An engineer, a QA lead and two agents work one excursion. Mentor keeps a running summary at the top; the full record sits below.

01

What a thread carries

Files, charts and references to the elements it is about. A mailbox with inbox, mentions, watching, muted, flagged and private views. Folders and rules that file new threads for you. A briefing at the top that says what happened since you last looked.

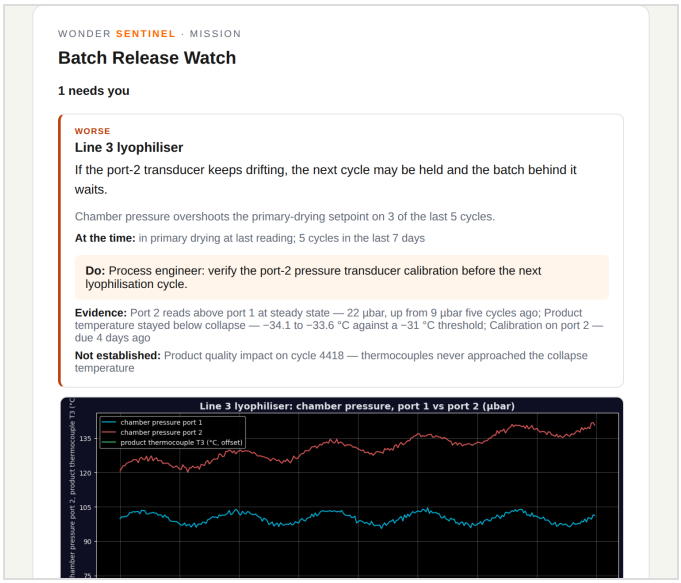
02

Why it matters

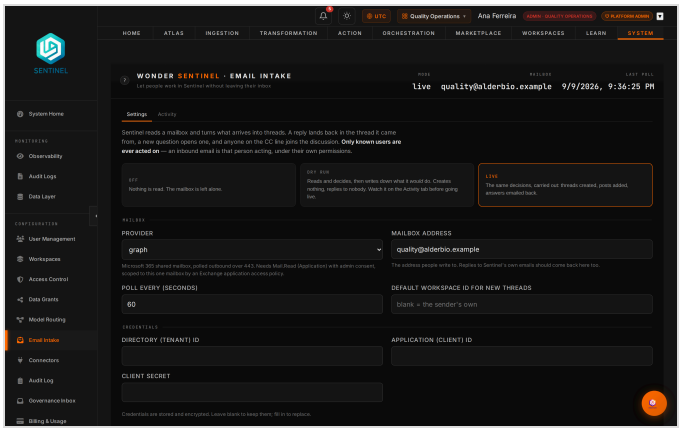
The reasoning behind a decision no longer lives in someone's head or in a chat app nobody can search. It lives next to the data it was about.

Email is a client

Sentinel works from email like a colleague. Forward a question, a supplier email or an attachment to its mailbox; a thread opens, the analysis runs, and the reply lands back in your inbox. Reply to the reply and the thread continues. Anyone on the CC line joins.



A mission briefing as it lands. The consequence first, then the ask, the evidence and the chart, with a one-click “was this worth your time?”.



Email Intake. Sentinel reads a shared mailbox and turns what arrives into threads. A dry-run mode lets you watch it for a day before it acts.

01

Inbound

Only known users are ever acted on. An email is that person acting, under their own permissions. A new question opens a thread; a reply lands in the thread it came from.

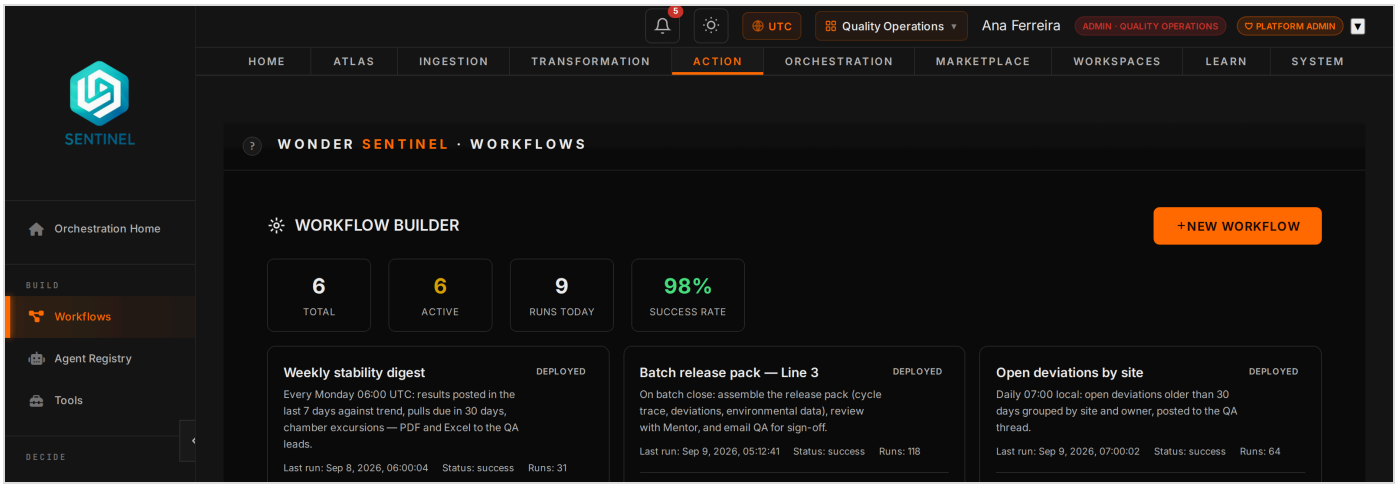
02

Outbound

Mission briefings, weekly digests, scheduled reports with PDF and Excel attached, and approval requests all arrive by email, charts embedded, with links back to the thread. A person who never opens the App still gets full value: ask by email, receive by email, approve by email.

What arrives without asking

The reports your team assembles by hand every week are a workflow. Build it once, in the canvas or by asking Mentor to build it, and it arrives every Monday with the numbers recomputed from the data.

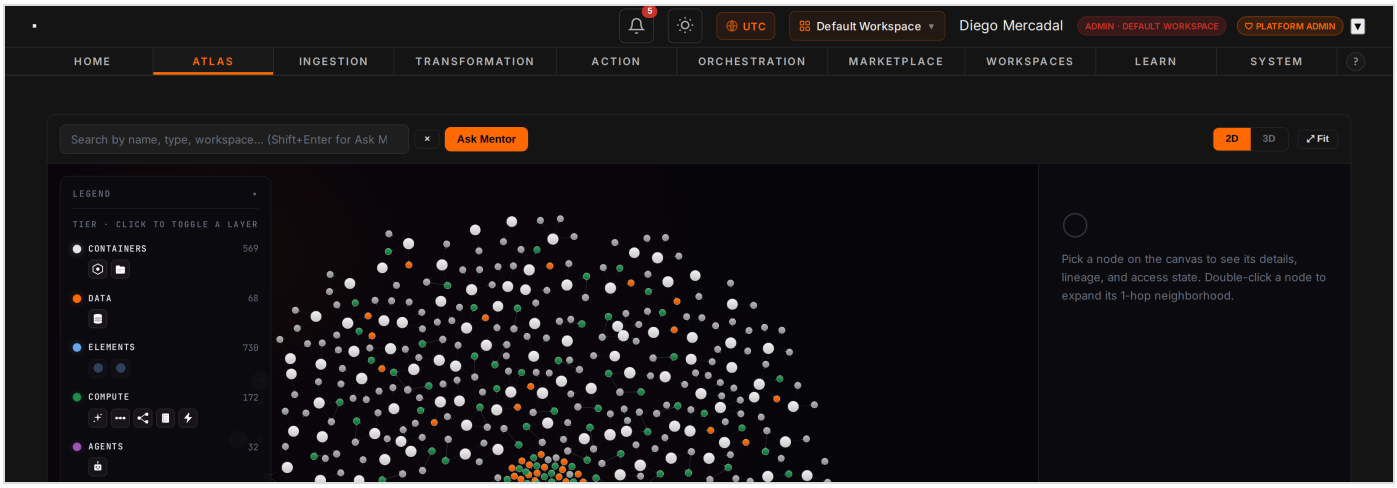


The workflow builder. Scheduled reports that query, author, chart, typeset and send, with a run history and a success rate you can see.

Mission briefings	sent only when a crew reaches consensus that a person must act; the consequence first, then the evidence
Weekly digest	what changed in your workspace this week: decisions, threads, mentions, what needs you
Scheduled workflows	a visual canvas — query, Mentor, chart, PDF, email — that runs on a schedule and attaches the files
A reviewer	an AI reviewer checks a generated report before it is allowed to send; a flawed one is blocked, not delivered
Alerts and events	rules on any measurement create alerts; notification rules decide who hears, how, and when

The Workbench

Everything the App shows was built here, by the people who own the data, and everything built here is immediately available to Mentor, to missions and to the App.

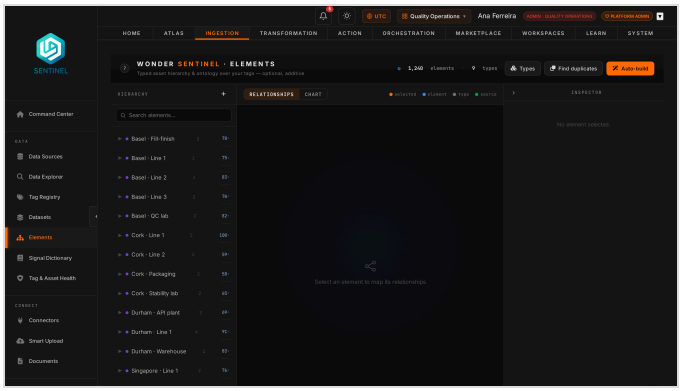


Atlas. A living map of every source, element, model, workflow and agent in the deployment, and how they connect. Pick a node to see its lineage and who may use it.

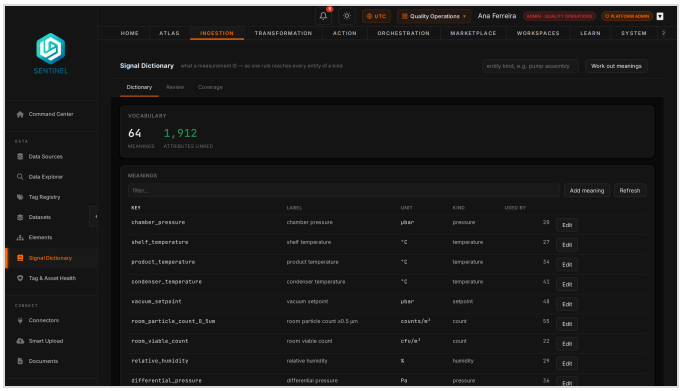
Atlas	walk upstream to see what feeds a number, downstream to see what breaks if it changes
Ingestion	sources, connectors, upload, the tag registry, the element tree and the signal dictionary
Transformation	Studio notebooks, models, the registry, experiments, pipelines and visualisation
Action	alerts, notifications, events, classifiers, artifacts and workflows
Orchestration	agents, tools, missions, threads, the live feed and agent operations
Marketplace	publish a workflow, agent, model or live dashboard once; install it anywhere as a fresh copy
Workspaces · Learn · System	who sees what; a step-by-step guide to every surface; the admin console

The element layer: one typed model of what you run

Raw data arrives as columns and tags with cryptic names. The element layer turns them into the nouns your business uses — a site, a line, a cost centre, a product, a contract — and hangs everything about each one in one place. Mentor, missions and modules all reason about elements, not columns.



Elements. A typed tree of the things the business runs, each with its own attributes, measurements, events and documents.



The signal dictionary. What a measurement is, defined once, so one rule reaches every entity of a kind.

01

How it works

Types carry attributes. Relationships connect elements to each other and to the tags that measure them. A dictionary says what a measurement means, so one idea under two names on two lines is one idea. Templates fan a mission or a model out across every element of a type.

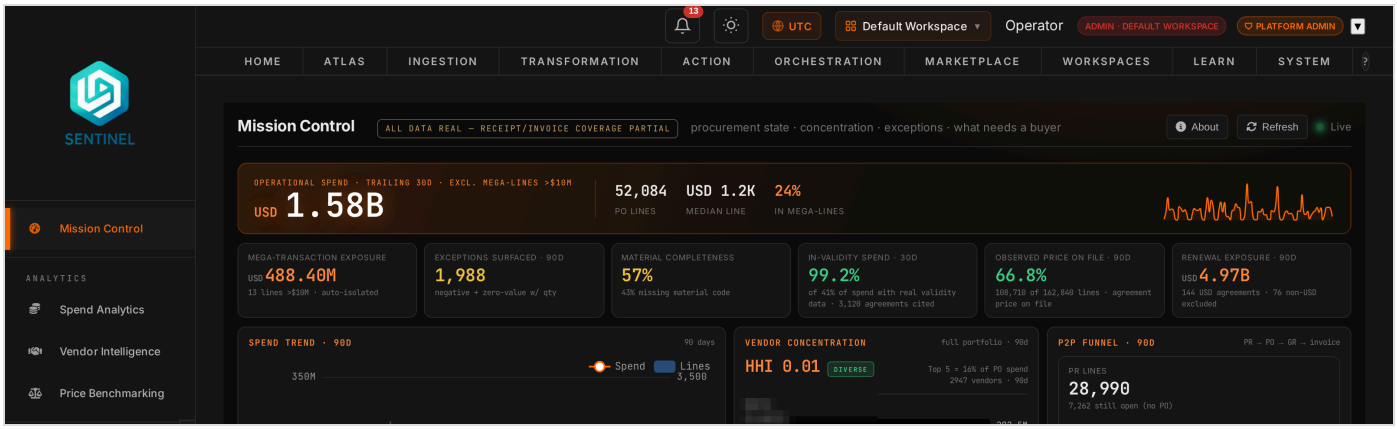
02

Why it is the hard part

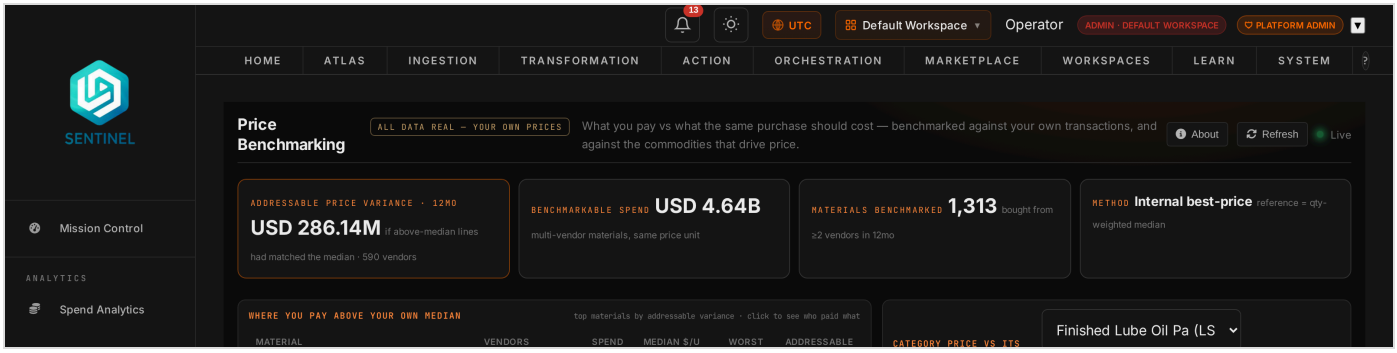
Without this layer an AI is guessing which column means what. With it, the model reads the same map your engineers use, which is why a small model performs like a large one here.

Modules: when a job deserves its own screen

A module is a set of purpose-built pages for one job, running on the same platform: the same data, the same permissions, the same Mentor button on every screen. Ask a question on any tile and the answer comes from the same governed query the tile used.



A procurement module's Mission Control. Trailing spend, exceptions, agreement coverage, vendor concentration and the procure-to-pay funnel. Every tile drills to the underlying lines.



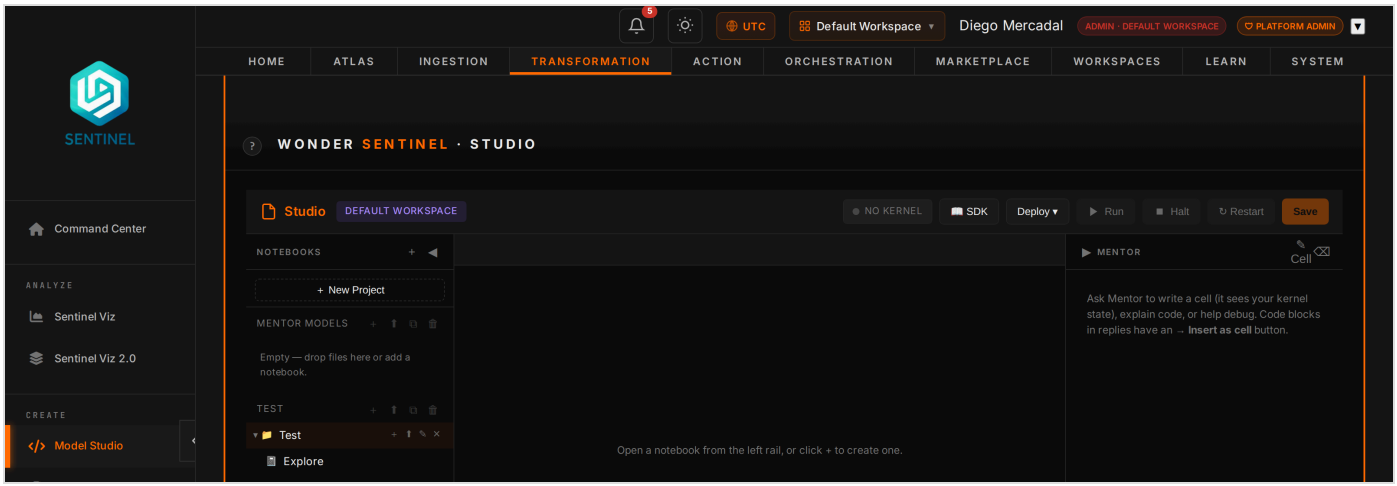
Price benchmarking. What you pay versus your own median for the same item, overlaid on a live public price index.

How modules are made

Built per deployment, in code or with Mentor, and installed per workspace. Anything reusable — a dashboard, a workflow, an agent, a model — is published once through the Marketplace and installed elsewhere as a fresh copy.

Three ways to build: talk, code, or your own IDE

The business user and the data scientist use the same platform and land in the same place. A model built in a notebook is what Mentor answers with; a workflow built from chat is what the engineer sees on the canvas.



Studio. Real notebooks with a kernel, a project per workspace, Mentor beside the code, one-click deploy to a pipeline.

Talk	ask Mentor to build it: a workflow, an alert rule, a live dashboard, a mission, a model. It authors, validates and deploys, and shows you what it made.
Code	open Studio: Python notebooks in the project's own environment, `import sentinel` to reach every source directly, scheduled runs, a model registry and pipelines to deploy into.
Your IDE	connect Visual Studio Code or the editor of your choice and work on the same projects, the same data and the same deploy path, from where you already work.
Mentor in Studio	Mentor is also a coding partner: it sees your kernel state, writes and explains cells, and inserts them for you.

Integrations: data in, actions out, agents both ways

Examples, not a list. Connectors and tools are added per deployment.

DATA IN
read in place

- SQL Server
- PostgreSQL
- MySQL
- other SQL databases
- Microsoft Fabric / OneLake
- OSIsoft PI
- InfluxDB
- DynamoDB
- REST APIs
- CSV, Excel, Parquet
- PDF, Word, images
- email and attachments

SENTINEL
understands, watches, acts

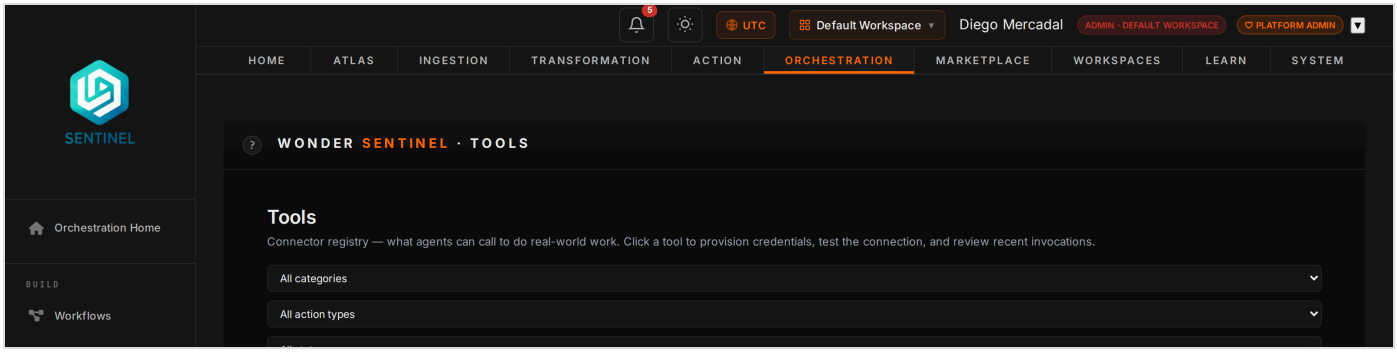
- elements and dictionary
- Mentor
- missions and crews
- threads
- workflows
- models and pipelines
- artifacts
- ledger and audit

ACTIONS OUT
with a record and an undo

- Email
- Microsoft Teams
- Slack
- Jira
- ServiceNow
- SharePoint / OneDrive
- Confluence
- Microsoft 365 Calendar
- webhooks
- S3 / SFTP
- PDF and Excel reports
- your own API

Agents, both ways

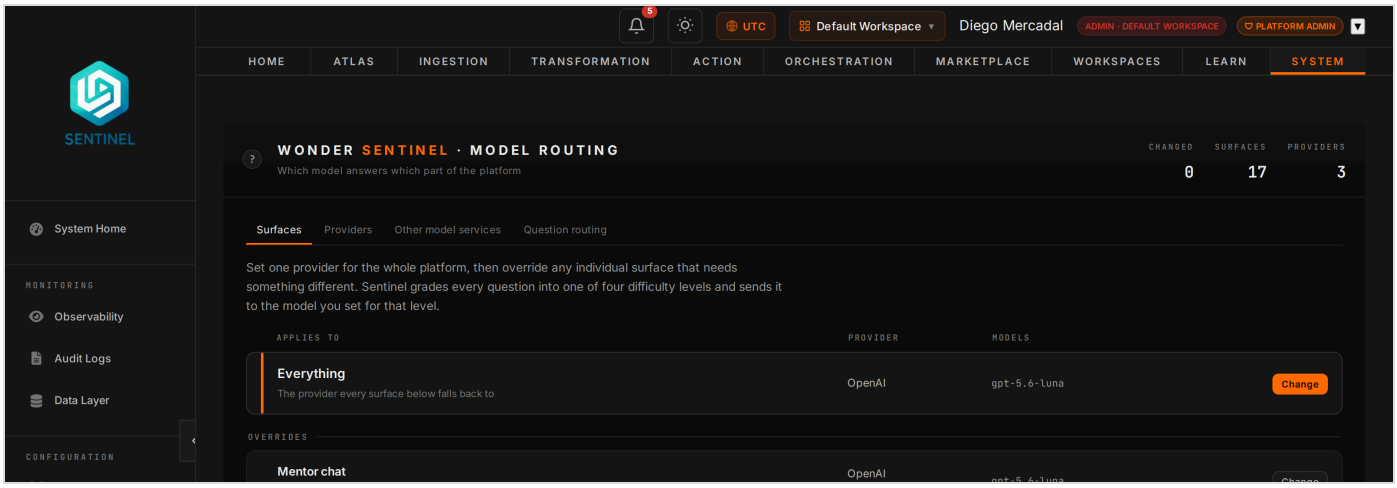
Your own agents reach in: Claude, Copilot Studio or GPT connect over MCP as the signed-in user and use Sentinel's tools with that person's permissions. Sentinel's agents reach out: every tool above is something a mission may call, simulated first, ledgered, and gated by the autonomy you set. A Python SDK and a REST API cover everything else.



The Tools registry. What agents may call, with credentials, a dry-run switch and a reversibility flag on each.

Your copilot asks. Sentinel is what it asks.

If your people already live in Claude, Copilot or ChatGPT, they can keep it. Add Sentinel as a connector and their assistant gains every Sentinel tool — discover sources, query, run recipes, build a dashboard — as that user, with that user's real workspace permissions.



Model Routing. Which model answers which part of the platform: one provider for everything, then overrides per surface.

01

How it works

The connector signs the user in with your identity provider, never a shared key. Every call is logged as that person. There are no destructive or admin operations on the connector. The same door is the one Copilot Studio agents use to reach tools, so a Teams agent can be built on it.

02

And the reverse

Sentinel's own Mentor runs on whichever model you choose, per surface. Because the grounding, the tools and the element layer do the heavy lifting, the platform is not hostage to any one provider, and a smaller model performs at a level it never could alone.

On-premises or cloud

Sentinel is one product with one codebase. It runs inside your own network, in a private cloud, or hosted by Wonder DataLabs. Nothing about the features depends on where it sits, only on where your data is allowed to be.

ON-PREMISES

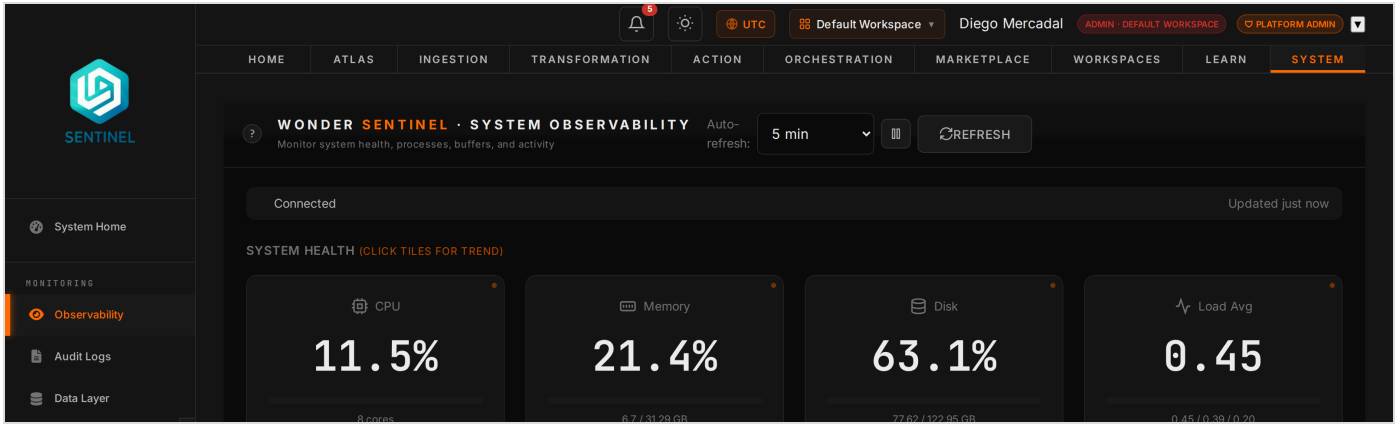
Behind your firewall

- runs on your own servers or private cloud
- signs in with your identity provider
- reads data that never leaves the network
- models private, or from a provider you choose
- updates delivered as releases

CLOUD

Hosted for you, or in your cloud account

- live in days, not quarters
- same software, same features
- your own tenant, nothing shared
- connects to cloud warehouses in place
- scales with the workload



System observability. The platform's own health page, on the customer's own server.

What stays where

Data stays inside the deployment. Models can be called from a provider of your choice or hosted privately; that choice is per surface and can change without touching anything else. The audit log, the ledger and every thread live with the deployment.

Sentinel is not a copilot

Copilots are language layers over your documents, mail and meetings, and they are good at it. An AI assistant and an AI operations platform are different species. The difference is not the chat box; it is everything behind it.

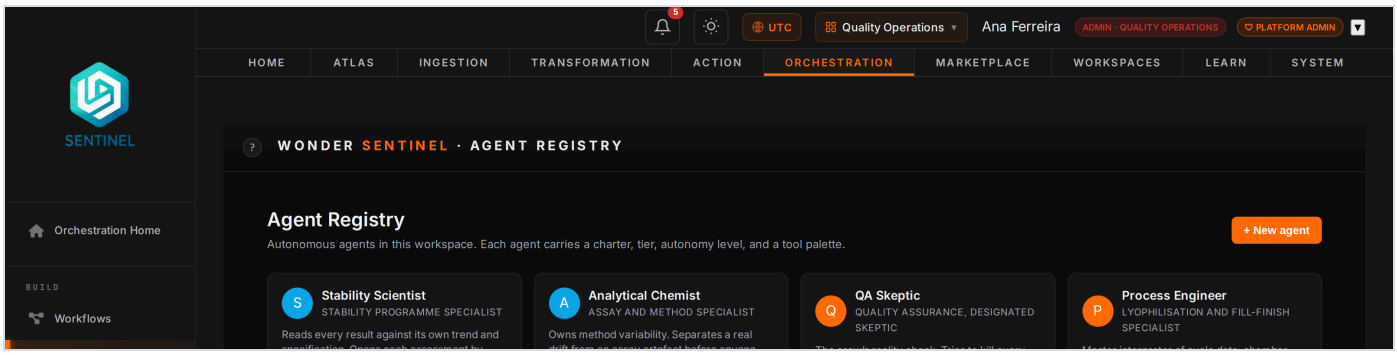
	A COPILOT	SENTINEL
Where the answer comes from	Retrieves and summarises what is written: documents, mail, meetings.	Runs a query on your governed data, shows the query, and agrees with the same number on every screen.
Between prompts	Nothing happens until someone asks.	Crews meet on a schedule, work the data, vote, and file findings. Workflows deliver reports. A reviewer blocks a flawed one.
Where it lives	In your productivity suite.	In your operation: the things you run, the systems that record them, the people who act on them, and the live external data federated onto your own.
The model	The vendor's model is the product.	A replaceable part, chosen per surface. The quality lives in the platform: the grounding, the element layer, the tools.
When it acts	Suggests. A person copies the suggestion somewhere else.	Acts — email, ticket, report, alert, model — through a ledger with cost, outcome and undo, behind gates a person sets.

> Keep your copilot. Connect it to Sentinel and let it ask Sentinel the questions it cannot answer alone. Your copilot asks; Sentinel is what it asks.

Why not build it: the parts list you would assemble

Foundry, Bedrock and their neighbours are excellent foundations: model hosting, agent frameworks, vector stores. Sentinel can run on them. The question is whether you want to build, and keep building, the layer above the foundation. Here is that layer, as it exists in Sentinel today.

Connectors	to databases, warehouses, historians, files, documents, APIs and email; read in place, kept current
A semantic layer	typed elements, relationships, a signal dictionary: the map the AI reasons over
Permissions	accounts, workspaces, roles, feature gates, data grants, enforced on every query, including the AI's
An analyst	a chat that writes and runs queries, shows them, cites the source and does not invent
Crews and missions	agents with charters that screen, meet, debate, vote, and escalate only on consensus
Approval gates	autonomy ceilings, simulate before execute, a ledger row with cost and undo for every action
Collaboration	threads shared by people and agents, mailboxes, mentions, digests, email in and out
Delivery	workflows that author, chart, typeset, review and send; modules; a marketplace
Operations	audit log, observability, model routing, break-glass, releases

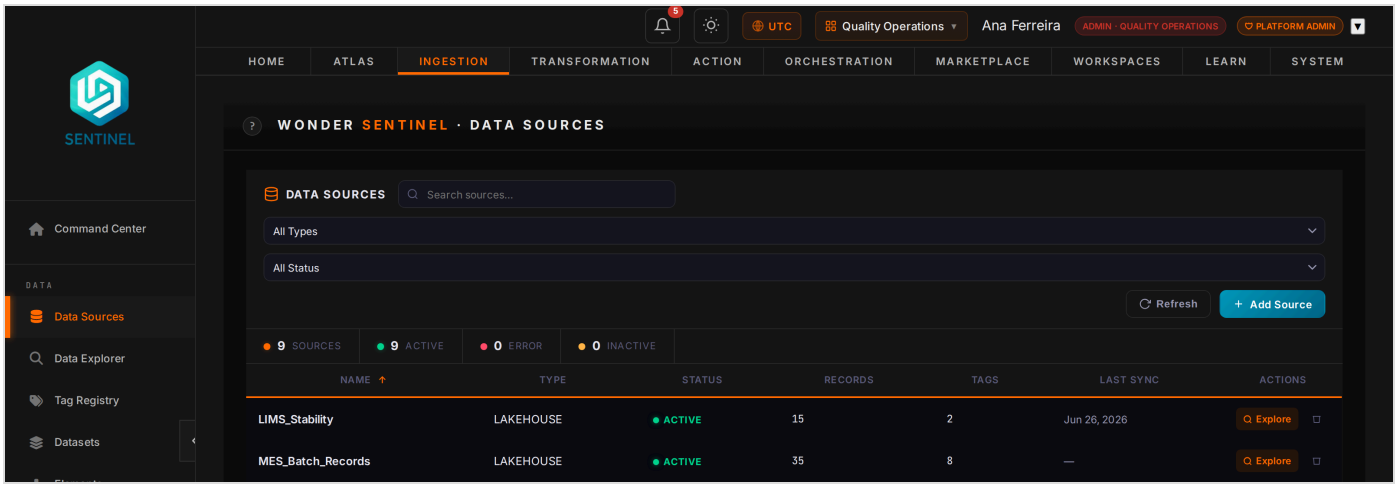


One line of the list, as it exists. The agent registry: each agent with a charter, a tier, an autonomy level and a tool palette.

> Each line is a project with an owner, a backlog and a maintenance bill. A custom build is frozen the day it ships. Sentinel is all of it, already running, maintained as one product, and open where it matters: your data, your models, your agents, your API.

Why not a warehouse: they prepare, Sentinel acts

A warehouse or lakehouse makes data clean, joined and queryable, superbly. Then the data goes back out to a dashboard, and the noticing, deciding and doing happens somewhere else, by people. Sentinel is that somewhere else, made into a platform.



Data sources. Lakehouse tables read through in place beside a historian and uploaded files: one list, one query language, nothing copied.

01

Keep your stack

Sentinel reads warehouses and lakehouses in place; one of the deployments pictured here reads a Fabric lakehouse directly. Your warehouse stays the system of record and the engineering investment stays intact. Sentinel adds the layer where a question is asked in plain language, a crew keeps watch, a thread records the decision, and an action is taken.

02

What storage does not have

A typed model of the things you run rather than tables. An analyst that shows its query. Agents that meet and vote. Threads where people and agents work together. Email as a client. Approval gates and a ledger. Modules for a business.

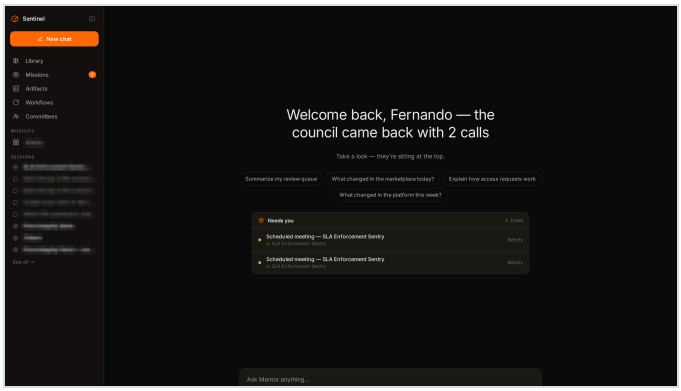
03

If you have no warehouse

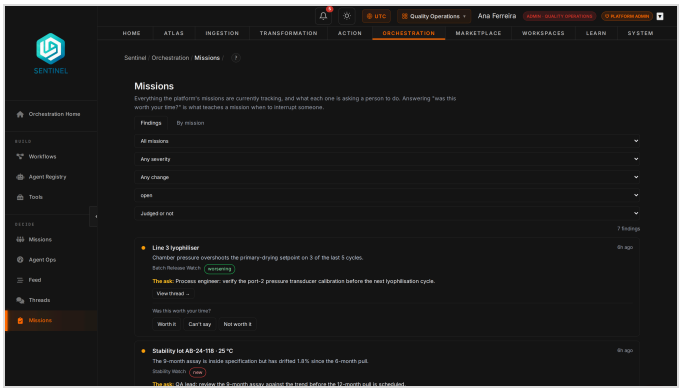
You do not need one to start. Sentinel connects to databases, files and exports directly.

Two industries, one platform

The same release serves a procurement team and a manufacturing quality programme on the same day. Your business would be a third configuration, not a third product.



Procurement. The App on arrival: a council of missions reporting back, and a procurement module on the rail.



Quality. The Missions page: what each crew is asking a person to do, and whether it was worth their time.

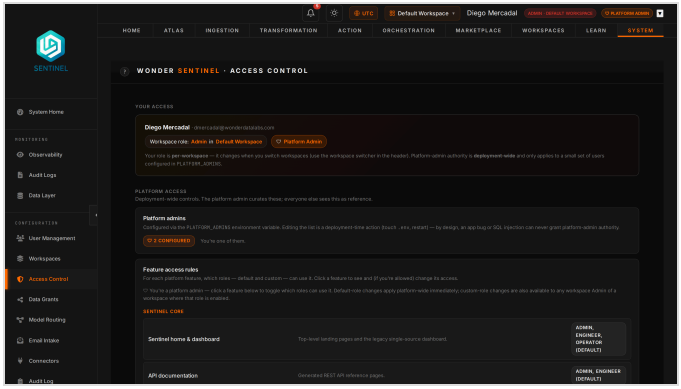
Procurement	millions of purchase-order lines, receipts, invoices and agreements from several SAP systems; price-integrity and SLA missions; a module with eight cockpits; a live public price index on real prices
Quality	batch records, stability studies, deviations and environmental monitoring across four sites; standing missions per programme; a release pack workflow per line; email intake for the QA leads

What is identical

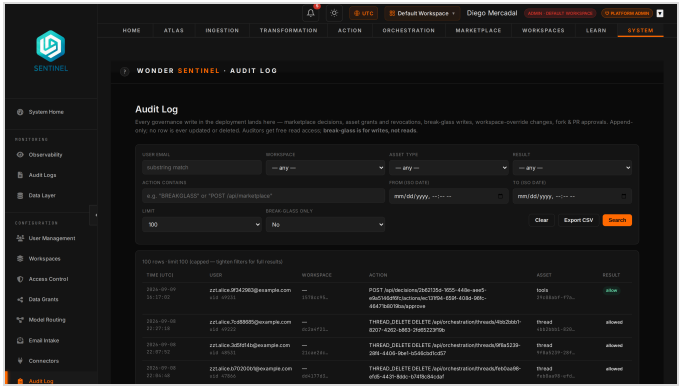
The App, the Workbench, Mentor, missions, threads, email, workflows, the element layer, the marketplace, the permissions, the ledger. What differs is configuration: sources, element types, the missions' charters, the module.

Governance: safe enough to let it act

Autonomy is earned in layers, and every layer is visible. Your people stay on the decisions; the platform keeps the receipts.



Access control. Per-workspace roles, platform-admin authority, and feature gates by role.



The audit log. Every governance write, append-only. No row is ever updated or deleted.

Accounts	the level above workspaces: one customer, many workspaces, one place to see and bill them
Workspaces	data, people, agents and work are scoped to a workspace; nothing crosses unless it is shared on purpose
Roles	per workspace, per person — Viewer, Operator, Engineer, Admin — with feature gates deciding which role may use what
Gates	each mission has an autonomy ceiling — propose only, notify only, or write; anything past it waits for a person
The ledger	every action an agent takes is simulated first, then recorded with its cost, its outcome and an undo where the tool allows one
Audit and grants	an append-only audit log; data grants reviewed and revoked explicitly; emergency access on its own audited path

Getting started

Everything in this brochure runs on the same platform. The practical next step is a conversation about which data you want answerable first.

01
Connect what you have

A database, a warehouse, a historian, or just files. Point a connector at it, or drop a spreadsheet on Smart Upload. Sentinel reads in place and builds the element tree with you. **Your data is answerable in plain language the same day.**

02
Sign in with your company account

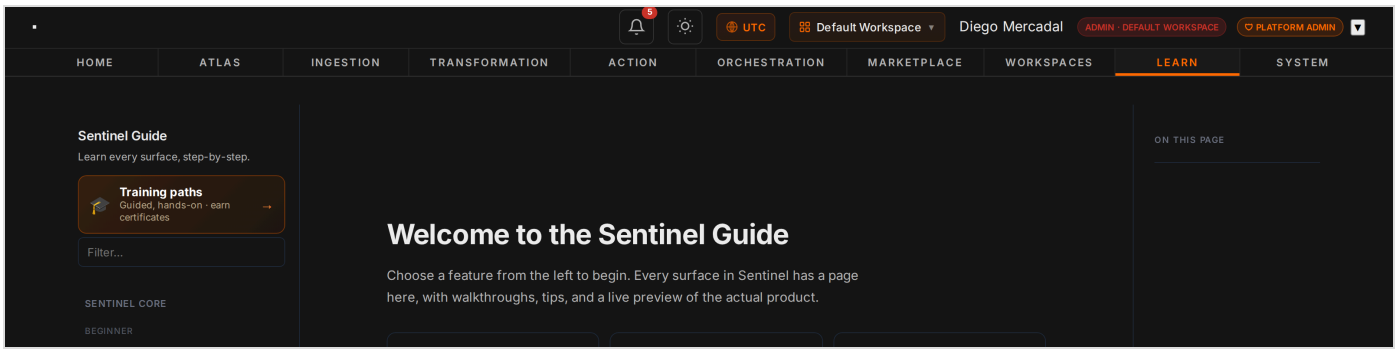
Your people already have Microsoft or Google identities; Sentinel signs them in with those. Roles are set per workspace. **No new passwords, no training course, real permissions from minute one.**

03
Ask, then let it watch

Type the question your team has meant to ask for a year. Then ask Mentor to turn it into a weekly report, and describe a mission for the thing you worry about. **The queue is gone. The watching runs itself.**

04
Give a job its own screen

When one job earns a cockpit, build a module — in code, or with Mentor — and install it on the App's rail. **Business people deliver value with no data team in the loop.**



Learn. A step-by-step guide to every surface, with training paths and a live preview of the product beside each lesson.

Quick reference

What it is	the layer where your data is connected, understood, watched and acted on
The App	one screen for everyone: ask, read, decide
The Workbench	where the data owners build: Atlas, Ingestion, Transformation, Action, Orchestration
Modules	purpose-built cockpits for one job, on the same platform
Mentor	runs the query, shows the work
Missions	crews that meet, argue, and speak only on consensus
Threads	people and agents in one conversation, with the record kept
Email	forward a question; the analysis comes back to your inbox
Build	talk to Mentor, code in Studio, or use your own IDE
Your agent	Claude, Copilot or GPT, connected as you, with your permissions
Deploy	on-premises or cloud; the data stays where it must
Govern	accounts, workspaces, roles, gates, a ledger and an audit log

> Every number the platform shows is computed from your own data, traceable to the query that produced it. Where the data is incomplete, it says so.