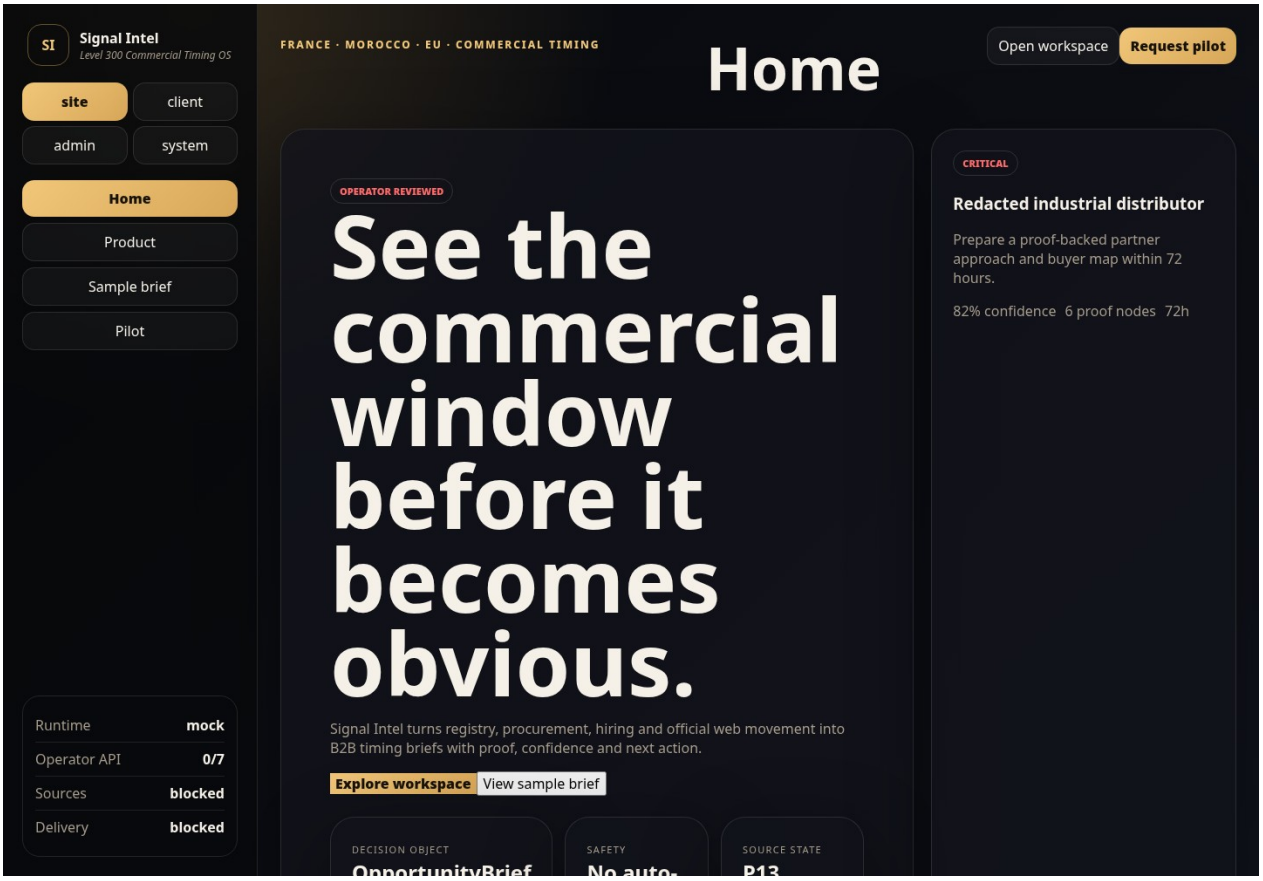
Transforming fragmented market signals into evidence-backed commercial dossiers

A multi-source intelligence architecture connecting acquisition, evidence lineage, qualification, human review gates and client delivery.

STATUS	TEAM	DEMO	SOURCE
Advanced platform · Controlled validation	**Founder-led platform build**	**Protected validation environment planned**	**Private repository**



Primary visual evidence

ANASS BENAMEUR · FULL-STACK PRODUCT ENGINEER

The product, the problem and the proof

A commercial opportunity is almost never one clean data point — it's a registry filing, a hiring pattern, a procurement notice and a timing signal that only mean something together. Elevya Signal Intelligence is the platform I designed to collect those fragments across source families, keep the evidence lineage of where each piece came from, run them through pattern detection and qualification, and stop before a human reviewer decides whether the resulting brief is good enough to send.

18	37	40+	Human gate
COMPONENTS IN ARCHITECTURE AUDIT	OFFLINE TESTS PASSED IN SMOKE REPORT	ORDERED MIGRATIONS	REVIEW BEFORE DELIVERY

Context

Multiple source families — registries, job boards, procurement portals, company sites — a canonical data layer, pattern detection, watchlists, briefing generation, production-readiness tooling, and a client-facing workspace. The captured product interface runs in explicit mock mode; live sources and delivery are currently blocked by design until the readiness gates pass. Its acquisition layer for job-market signals reuses the same adapter architecture as the standalone ATS Intelligence Engine in this portfolio, extended here with registry, procurement and hiring-velocity sources feeding one canonical model.

My role and ownership

- Platform and source-ecosystem architecture
- Canonical schemas and migration ordering
- Quality gates and production-readiness harnesses
- Evidence, pattern and briefing workflow design
- Client workspace and opportunity-brief interface

Personal contribution

Architecture, data model, evidence workflow, gates, readiness tooling and product interface.

Why the existing workflow needed a system

Without lineage, an automated alert is just noise with a timestamp — there is no way to tell whether it's corroborated by a second source, whether the underlying data is stale, or whether it's safe to put in front of a client. Independent source scripts with inconsistent state, signals with no shared identity, and no clear separation between observation, interpretation and delivery meant client output depended on manual assembly and production readiness was nearly impossible to verify honestly.

Before and after

BEFORE

- Independent source scripts with inconsistent state
- Signals without shared identity or evidence lineage
- No clear separation between observation, interpretation and delivery
- Client output dependent on manual assembly
- Production readiness difficult to verify

AFTER

- Source manifest and ordered migrations
- Canonical entities, events and lineage records
- Pattern, validation, watchlist and briefing layers
- Operator review checklist before delivery
- Preflight, smoke, audit and soak commands for readiness evidence

Meaningful result

A traceable path from multi-source evidence to a human-reviewed opportunity brief.

How the system works

Raw signals arrive from source adapters into a canonical schema, each one carrying its evidence trail forward rather than collapsing into an opaque number. Pattern detection looks for corroboration across sources — a hiring spike plus a registry change plus a procurement notice is a categorically different signal than any one of those alone, and the system is built to treat it that way.

Operational workflow

01	02	03	04	05
Sources	**Evidence**	**Qualification**	**Dossier**	**Gate**
Registry, hiring, procurement	Canonical events and lineage	Patterns and confidence	Opportunity brief	Human review and delivery

SYSTEM ARCHITECTURE

Elevya Signal Intelligence Platform

Evidence-led architecture diagram · generated from the verified project workflow

Anything that clears qualification lands in a client workspace as a brief with confidence, supporting evidence and a range, sitting in an explicit review state until an operator signs off — the system is built to refuse to auto-send, on purpose. A control CLI, ordered PostgreSQL migrations, scheduler support and a readiness-audit harness sit underneath: one recorded run passed the quality gate, 37 offline tests and a migration dry-run, then correctly failed at the live-connectivity stage because DATABASE_URL and live source credentials weren't present — the harness refusing to mark an incomplete environment production-ready, exactly as it should.

Architecture and decisions

The production bundle includes a control CLI, ordered PostgreSQL migrations, scheduler support, architecture audit tooling and a readiness-audit harness. The job-market acquisition layer inside this platform shares its adapter architecture directly with the standalone ATS Intelligence Engine case study in this portfolio — built once, proven independently, and extended here with the registry, procurement and hiring-velocity layers that turn a normalized job feed into a corroborated commercial signal. That relationship is stated here on purpose.

Critical engineering decisions

DECISION 01

Preserve evidence lineage

Signals are commercially credible only when the source, timing and transformation path remain inspectable.

DECISION 02

Separate detection from delivery

Pattern and qualification layers can produce a candidate brief, but external delivery stays review-gated.

DECISION 03

Test production readiness explicitly

Preflight, smoke, migration, benchmark and soak commands create evidence instead of relying on a deployment label.

DECISION 04

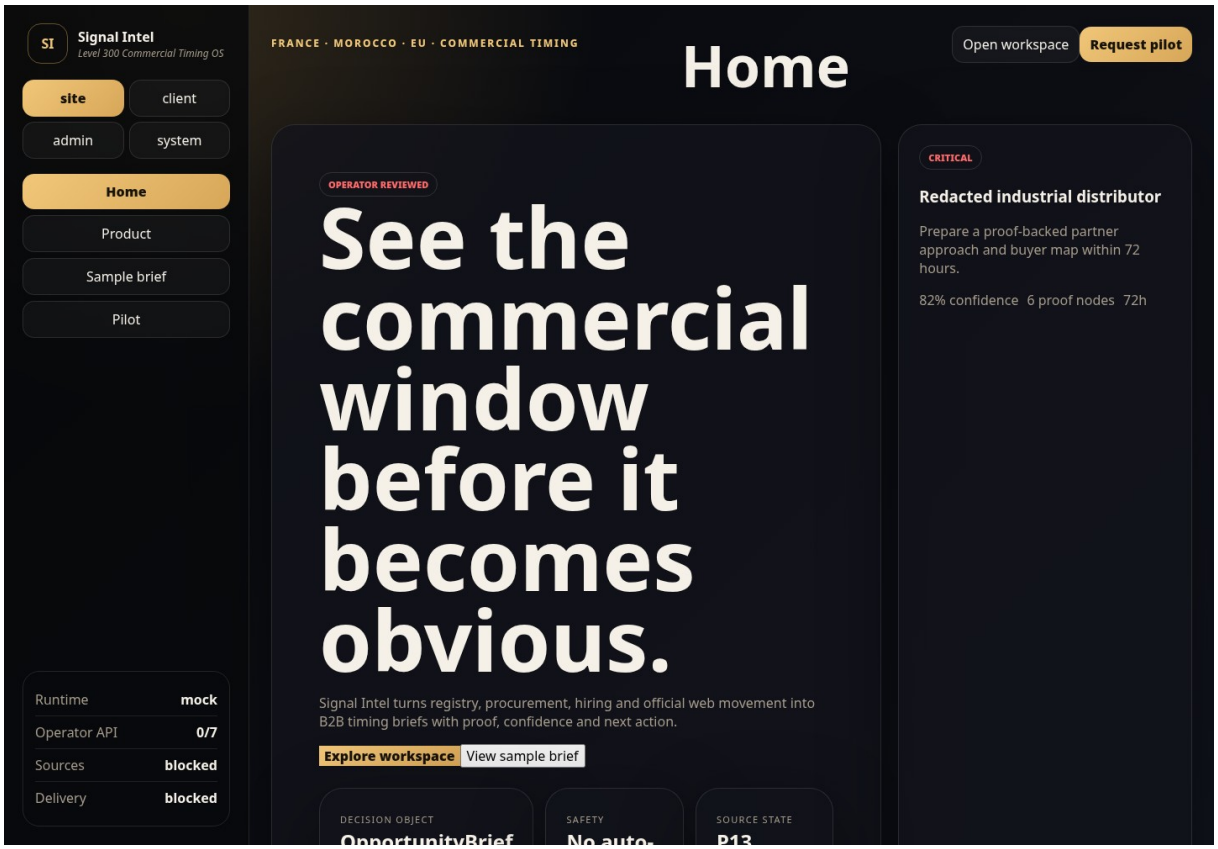
Expose limitations in the UI

The captured build visibly reports mock runtime, blocked sources and blocked delivery rather than pretending to be live.

Technology stack

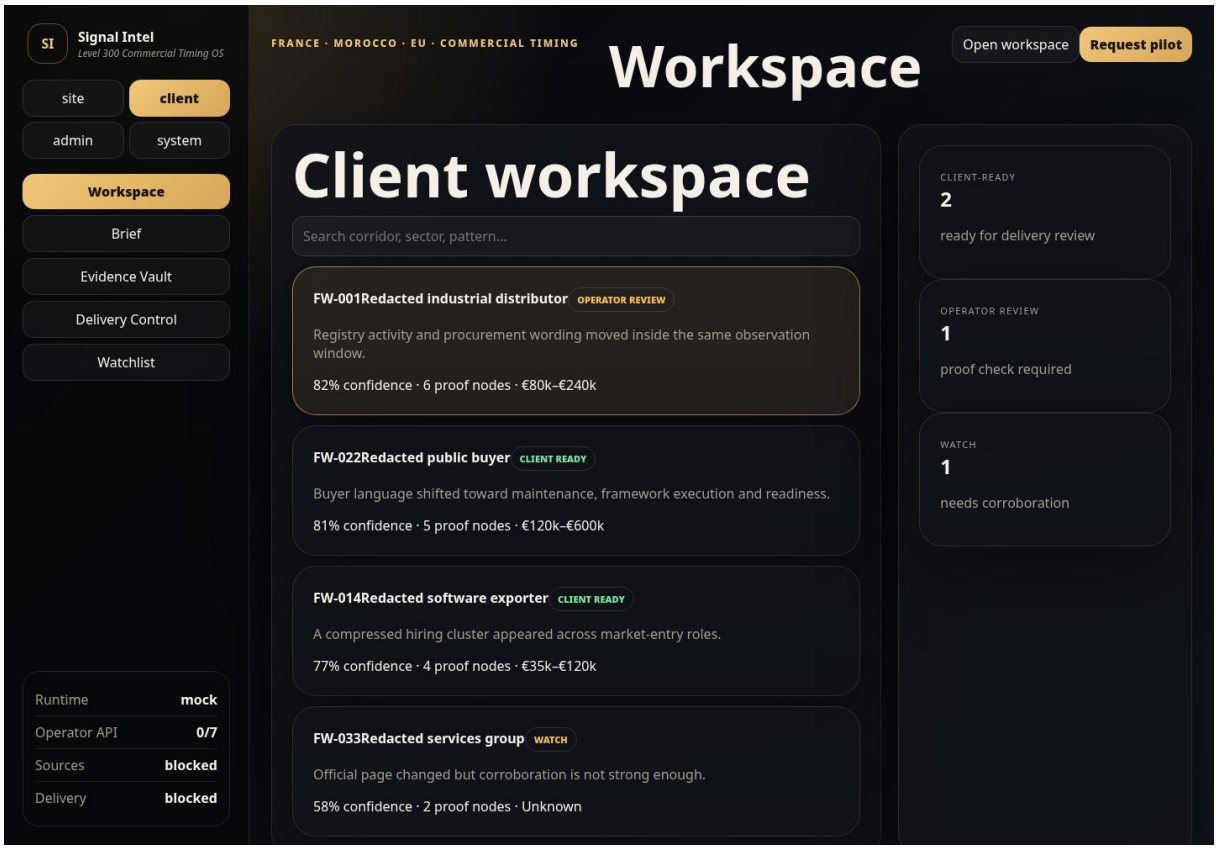
Python	PostgreSQL	Source adapters	Pattern engine	Evidence lineage	React product UI

Evidence 1 of 4



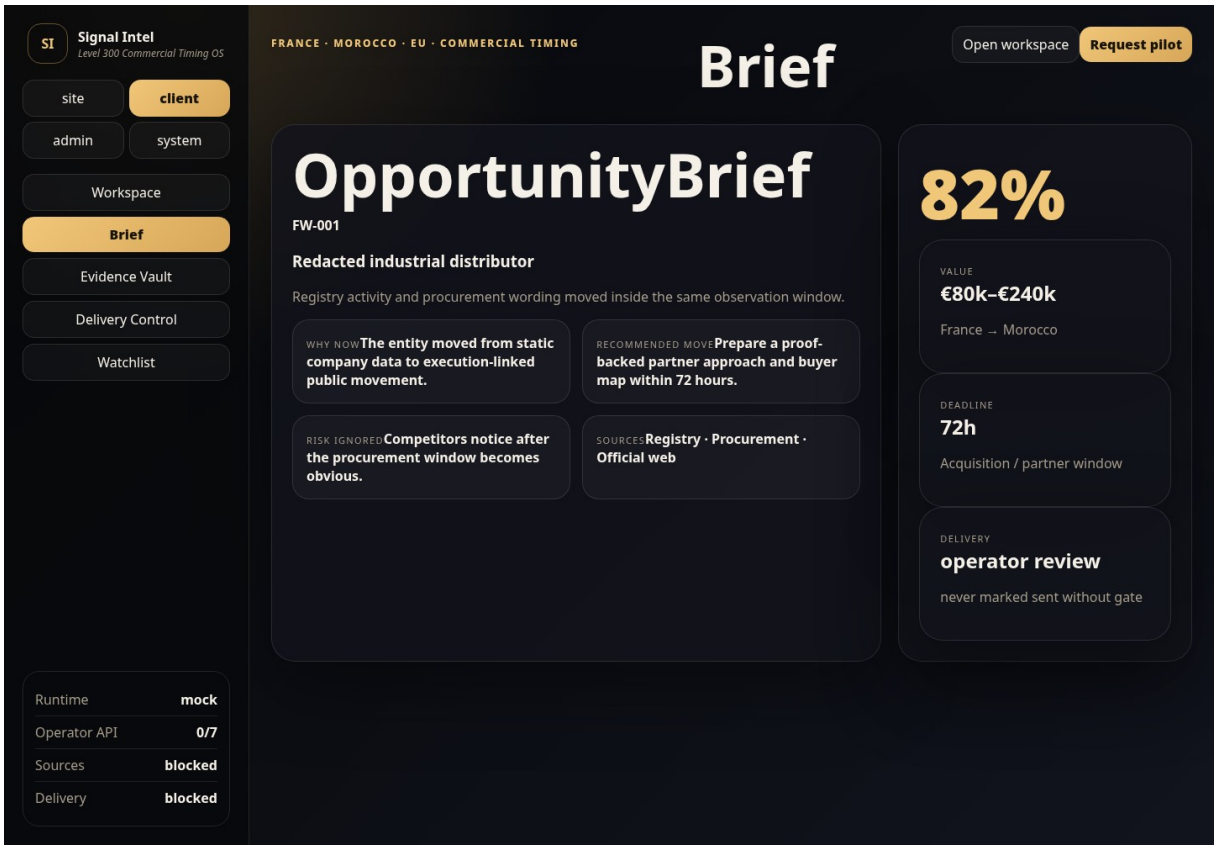
Product shell positioning the platform around commercial timing.

Evidence 2 of 4



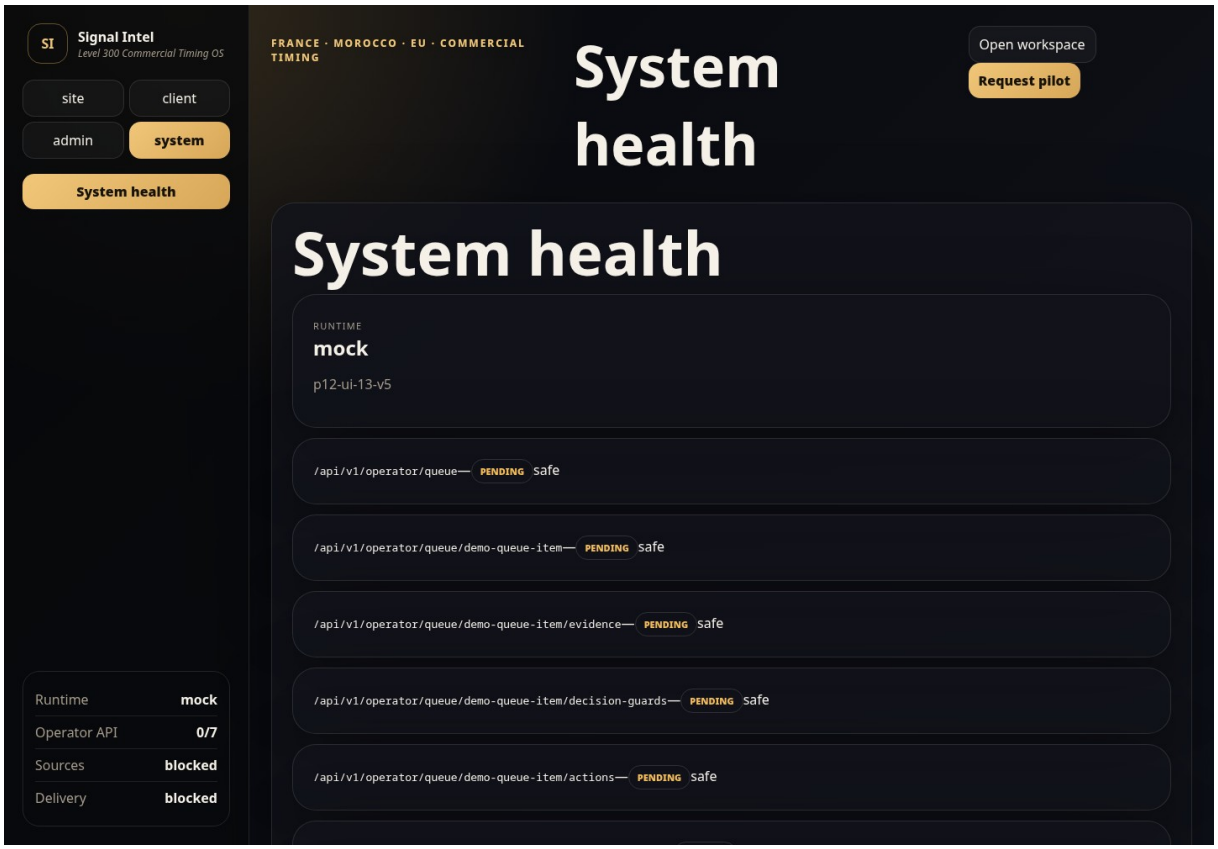
Client workspace listing opportunities and review states.

Evidence 3 of 4



Synthetic opportunity brief with confidence, range and operator-review state.

Evidence 4 of 4



System-health view explicitly showing mock mode and blocked dependencies.

Value, proof and honest limitations

Operational impact

The platform establishes a coherent path from fragmented evidence to a structured opportunity brief. This is the clearest demonstration in the portfolio of designing for trust under uncertainty: a system that would rather show "blocked, missing credentials" than fabricate a green status.

Verified evidence

- Architecture audit covers 18 system components
- Quality gate passed compile, AST, manifest and contract checks
- Offline smoke step passed 37 tests
- Ordered migration dry-run completed successfully
- UI captures show workspace, brief, system health and responsive shell

Current limitations

- The captured UI is a product shell running in mock mode; sources and delivery were blocked.
- A recorded preflight failed because database connectivity and live source credentials were absent.
- Architecture audit results show uneven maturity across modules.
- Several components need consistent caching, raw-artifact archives, metrics or local replay tests.
- Do not describe the platform as stable production until live smoke, benchmark and soak reports pass.

Current status

STATUS	TEAM	DEMO	SOURCE
Advanced platform · Controlled validation	**Founder-led platform build**	**Protected validation environment planned**	**Private repository**

WHY THIS PROJECT MATTERS

A traceable path from multi-source evidence to a human-reviewed opportunity brief.

This case study is designed to be inspectable: strong claims are connected to screenshots, source evidence or explicitly stated limitations.