

ATX ComplianceAI

Ground Every Answer. Isolate Every Tenant. Prove Every Citation.

A multi-tenant compliance knowledge plane, callable by any AI agent over MCP

1. What Is ATX ComplianceAI?

ATX ComplianceAI is a multi-tenant knowledge-base-as-a-service for regulatory compliance. It ingests the regulatory corpus (statutes, agency rules, standards and guidance) alongside each customer's own policies, contracts and regulator correspondence, normalizes them into a versioned section model with a typed relationship graph, and serves grounded, citation-backed retrieval to any AI agent over MCP. It is a data plane, not a chatbot: it returns verified facts with provenance and never composes a narrative answer.

At a glance:

- **Grounded, never narrative.** Every retrieval returns grounded, partial or no_answer with citations anchored to an immutable section version, effective date and source URL. There is no unattributable prose to audit.
- **Tenant isolation at the data boundary.** A single chokepoint re-checks the org scope of every row, chunk, graph node and audit write it returns, rather than trusting the query filter. A mismatch fails the call and files an override event.
- **Time travel over the corpus.** Bitemporal section versions and a materialized supersession chain: ask what a rule said on any date, and pinned citations never drift when the regulation is amended.
- **Relationship-aware.** A typed graph (cites, implements, impacts, contradicts, amends, supersedes) links a customer's policy to the regulation behind it, and flags the ones that no longer agree.
- **MCP-native.** Fourteen read tools over stdio and Streamable HTTP, plus REST/OpenAPI and a typed Python SDK. One registry generates all three; all three pass the same isolation chokepoint.
- **Audit-ready by default.** An append-only, monthly-partitioned event store across eight namespaces, with a non-disableable seven-year retention floor and an invariant monitor that watches it.
- **Runs where your data lives.** Closed network, in-VPC or fully air-gapped Docker, so sensitive data never leaves your environment.

2. What We Have Today

The KB runs today as a Python service over Postgres, Qdrant, FalkorDB and Redis, with Celery workers, a generated operator CLI and a generated client SDK. What is live:

Capability	What It Does
Versioned Document Spine	Document to Section to SectionVersion to Chunk, bitemporally versioned. A document update aligns against the prior version, cascades the diff and supersedes; it never overwrites.
Doctype-Driven Retrieval	Dense and BM25 candidates, fused and cross-encoder reranked, then either ranked primary-doctype sections or a cross-doctype synthesis that ties a policy to the regulation governing it.

Capability	What It Does
Tristate Grounding	grounded / partial / no_answer, with every claim carrying its section version, chunk, source URL and effective date. The gate is a tool-internal check, not a reasoning step.
Tenant-Isolation Chokepoint	One decorator at every store read, with a universal-corpus sentinel for shared regulation. Mismatch means refusal plus an audited override row; the tool surface has no route around it.
Typed Relationship Graph	Background edge derivation under a per-tenant allow-list, LLM validation, staleness revalidation, and edge invalidation when a document version changes underneath.
Controlled Vocabulary	Curated concept tags scoped per tenant, with deprecate, merge, rename and promote lifecycles and supersession chains, so scope tagging stays consistent as the corpus grows.
Append-Only Audit Fabric	A partitioned Postgres event store across query, ingest, edge, vocabulary, override, marker and feedback namespaces, plus a standalone monitor polling its invariants every ten minutes.
Three Transports, One Registry	A single tool-registration decorator generates the OpenAPI spec, the MCP manifest, the kb-client SDK and the kb-admin CLI. No contract is hand-written, so none can drift.
Operator Surface	Twenty-six admin REST endpoints covering tenant lifecycle, per-tenant config under change-control tiers, allow-list, vocabulary, parsers, documents, edges, audit search and identity.
Pluggable Parsers	Per-doctype parsers register as entry-point plugins, so onboarding a customer's own doctype is a plugin and a config edit rather than a fork of the pipeline.

3. The MCP Surface We Expose

Any MCP client connects over stdio for development or Streamable HTTP in production and gets fourteen read tools. Claimed identity travels with every call, and every tool passes the same isolation chokepoint, so a client reaches exactly what its tenant owns and nothing else. Administrative operations are deliberately never exposed over MCP or the SDK; they are REST-only, behind a separate authorization tier.

MCP Tool	What It Does
search	Doctype-routed retrieval. A primary doctype alone returns ranked sections; add a secondary doctype and it confirms the seeds and synthesizes the cross-doctype relationships, with citations.
get_section	One section version, current or the version effective as of a given date, with its chunks and scope tags.
get_section_versions	A section's full version timeline, oldest first, optionally windowed by effective date.

MCP Tool	What It Does
get_version_chain	Walks the materialized supersession lineage, so a citation can be traced forward across amendments.
get_document	A document version's metadata and scope, with its sections in document order.
get_document_sources	Inbound cites and implements relations: which regulations govern this policy.
get_document_impact	Outbound impacts, contradicts, amends and parallels relations: what this document reaches.
get_neighbors	Depth-1 incoming and outgoing neighbors of a section version, each with its edge-type provenance.
summarize_neighbors	Neighbor counts by edge type and target doctype, so an agent can decide whether a hop is worth taking.
get_path	The fewest-hop path between two nodes, bounded by hop count and edge type.
get_concept	A controlled-vocabulary concept: label, status and supersession chain.
list_concepts	Paginated vocabulary browse, filtered by prefix, org scope and deprecation status.
describe_schema	The node and edge taxonomy plus this tenant's inventory, so a caller can orient itself without hard-coded knowledge.
submit_feedback	Records a verdict on a prior query as an audited feedback event that feeds the revalidation loop.

4. What We're Introducing

Capability	What It Solves
Calibrated Confidence	The tristate verdict gains a calibrated confidence score from a shared grounding engine, with an offline recall and accuracy harness so retrieval quality is measured, not asserted.
Model Freedom	Every LLM, embedding and rerank call moves behind a config-selected router, so a customer can run hosted, self-hosted or their own endpoints without being locked to one vendor's embeddings.
Traced and Costed Calls	Observability rides the router: tokens, cost and latency captured per model call and attributed per tenant, with no per-team instrumentation work.
Profile-Driven Pipelines	Ingest and retrieval become an ordered registry of steps, so a new corpus shape is a profile rather than a fork of the pipeline.
Continuous Regulatory Ingest	Scheduled pull of regulatory source updates with automatic version alignment and cascade, replacing today's operator-driven upload.
Hardened External MCP	The Streamable-HTTP ingress opened to third-party MCP clients: per-client scoping, rate limits and a semver-pinned published contract.
Grounding Gates in CI	Retrieval quality gated in the deployment pipeline against a labelled compliance question set, so a regression blocks the release instead of reaching a customer.

Refusal as a Safety Feature. The KB refuses rather than guesses. When the evidence does not support the question, search returns no_answer instead of a low-confidence paragraph, so the agent in front of it escalates to a human rather than inventing a citation a regulator will later read.

Operator Workflow. kb-admin, a generated CLI, drives tenant onboarding, per-tenant config under change-control tiers that prompt for written justification on protected fields, vocabulary curation and audit search. A Docker sandbox stack runs the whole system end to end for integration tests against the real transports.

ATX Labs · Built for Regulated AI Systems