ATX KomplianceAI

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

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

ATX KomplianceAI is a multi-tenant knowledge-base-as-a-service for regulatory compliance. It ingests the regulatory corpus alongside each customer's own policies, contracts and regulator correspondence, versions everything bitemporally with a typed relationship graph, and serves grounded, citation-backed retrieval to any AI agent over MCP — a data plane, not a chatbot, that 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, cited to an immutable section, version and date.

– **Tenant isolation at the boundary —** One chokepoint re-checks org scope on every read; a mismatch fails closed and logs an override.

– **Time travel over the corpus —** Bitemporal versions and supersession chains mean pinned citations never drift.

– **Relationship-aware —** A typed graph links a customer's policy to the regulation behind it and flags disagreements.

– **MCP-native —** 14 read tools over stdio/HTTP, plus REST/OpenAPI and a typed SDK — one registry, one isolation chokepoint.

– **Audit-ready by default —** Append-only event store, eight namespaces, a seven-year retention floor, an invariant monitor.

– **Runs where your data lives —** Closed network, in-VPC or air-gapped Docker — sensitive data never leaves your environment.

What We Have Today

– **Versioned Document Spine —** Document → Section → Version → Chunk, bitemporal; never overwrites.

– **Doctype-Driven Retrieval —** Dense+BM25 fused, cross-encoder reranked, cross-doctype synthesis.

– **Tristate Grounding —** grounded / partial / no_answer — every claim carries its source.

– **Tenant-Isolation Chokepoint —** One decorator at every store read; no route around it.

– **Typed Relationship Graph —** Background edge derivation, LLM-validated, auto-invalidated on change.

– **Controlled Vocabulary —** Curated per-tenant tags with full lifecycle and supersession.

– **Append-Only Audit Fabric —** Partitioned event store, seven namespaces, invariant monitor.

– **Three Transports, One Registry —** One decorator generates OpenAPI, MCP manifest, SDK and CLI.

– **Pluggable Parsers —** New doctypes onboard as plugins, not pipeline forks.

What We're Introducing

– **Calibrated Confidence —** A confidence score from a shared grounding engine, measured by an offline harness.

– **Model Freedom —** Every LLM/embedding/rerank call sits behind a router — no vendor lock-in.

– **Traced & Costed Calls —** Tokens, cost and latency captured per call and per tenant, free.

– **Profile-Driven Pipelines —** A new corpus shape is a profile, not a pipeline fork.

– **Continuous Regulatory Ingest —** Scheduled source pulls with automatic version alignment and cascade.

– **Hardened External MCP —** Third-party MCP access with per-client scoping and rate limits.

– **Grounding Gates in CI —** Retrieval quality gated pre-release against a labelled question set.

MCP Surface

14 read tools — search, get_section(_versions), get_version_chain, get_document(_sources/_impact), get_neighbors, summarize_neighbors, get_path, get_concept, list_concepts, describe_schema, submit_feedback — over stdio and Streamable HTTP, plus REST/OpenAPI and a typed Python SDK, one registry, one isolation chokepoint. Admin operations stay REST-only, behind a separate authorization tier.

Trust & Operations

Refusal as a safety feature — the KB returns no_answer rather than a low-confidence guess, so the agent in front of it escalates instead of inventing a citation.

Operator workflow — kb-admin drives tenant onboarding, change-controlled config, vocabulary curation and audit search; a Docker sandbox runs the whole system end to end for integration tests.