CORTEX ARCHITECTURE PROOF

Cortex architecture

Last updated: May 20, 2026

How Cortex enforces isolation, auditability, and sovereign control

This page describes how Cortex is built: the data flow, the isolation model, the identity and access boundaries, the retention and deletion controls, the audit ledger, the model governance options, and the deployment modes available to buyers reviewing Cortex for legal, regulated SaaS, or enterprise use.

It is intended as a depth reference for security, legal, compliance, and architecture teams during a vendor review. Specific certifications and contractual terms are confirmed per engagement.

Data flow at a glance

Every request that touches a tenant's context follows the same path:

1. The caller (a user, an agent, an internal service, or a customer-facing application) authenticates against the platform's identity boundary.
2. The request is scoped to a single tenant namespace. Cortex refuses requests that cannot be tied to exactly one tenant.
3. Retrieval, prompt assembly, and tool invocation happen inside the tenant's Private Compute Unit. The tenant's memory, embeddings, secrets, and policy live inside that unit and never leave it for the purpose of serving another tenant.
4. Model calls go to the model provider chosen for that tenant or workflow, under the governance policy configured for that namespace.
5. The full call (caller, tenant, retrieved context, prompt, model, response, tools invoked) is recorded in the per-tenant audit ledger before the response is returned.

   The same path applies whether Cortex is consumed by a chat copilot, by a background agent, or by a programmatic API client. There is no fast path that bypasses tenant scoping or audit.

Per-tenant isolation model

Cortex is organized around the Private Compute Unit (PCU): a logical and operational boundary that wraps one tenant's memory, policy, keys, and execution.

- Each tenant has its own memory namespace. Embeddings, documents, prior agent state, and conversation history are partitioned by tenant identifier at the storage layer, not by

application-level filters.

- Each tenant has its own encryption keys. Data at rest is encrypted with keys scoped to that tenant, so a leaked key cannot decrypt another tenant's material.
- Retrieval is constrained at the platform layer. A query issued inside tenant A's PCU cannot reach storage, indices, or caches belonging to tenant B, even if a prompt or tool attempts to ask for it.
- Cross-tenant operations are not a runtime capability. There is no "search across all tenants" surface in the product. Aggregations for operators are computed on metadata only, never on tenant content.

This is the same pattern productized in the legal vertical, where each client or matter is given its own PCU. Cortex generalizes that pattern beyond legal.

Identity and access boundaries

Cortex maps to the identity systems organizations already run, rather than introducing a parallel directory.

- Authentication: SSO via the customer's identity provider (OIDC or SAML), with support for service accounts and machine identities for agents and backend callers.
- Authorization: role and attribute based access, enforced at the PCU boundary. Roles map to existing organizational roles (for example, matter team, client team, internal copilot user, operator, auditor).
- Agent identity: every agent and every tool call carries a verifiable identity. Agent actions are attributable to the agent, the human on whose behalf it acts, and the tenant it is scoped to.
- Operator access: platform operators do not have standing access to tenant content. Break-glass procedures exist for incident response, require explicit approval, and are themselves recorded in the audit ledger.
- Network boundary: PCUs can be exposed only to the customer's network, to a defined set of allowlisted callers, or to the public internet under the customer's policy.

Retention and deletion controls

Retention is a per-tenant policy, not a global default.

- Configurable retention windows: memory, transcripts, retrieved snippets, and tool outputs can each be retained for a different duration, defined per tenant and per workflow.
- Hard deletion: deletion requests purge the underlying records, including embeddings and derived indices. Deletion is verifiable in the audit ledger.
- Export: tenants can export their memory, audit ledger, and configuration in a documented

format, so they are never locked in.

- Legal hold: when a tenant requires it, specific data can be placed under legal hold and exempted from retention based deletion until the hold is released.
- Backups: backups follow the same tenant scoping and the same retention policy as live data, and are deleted on the same schedule.

Audit ledger

The audit ledger is a first class part of the product, not an add on.

- Scope: every prompt, retrieval, agent action, tool invocation, model call, configuration change, and access event is recorded.
- Granularity: each record includes who acted, on behalf of which tenant, against which memory, with which model, with which policy in force at that moment.
- Tamper evidence: records are append only and chained so that after the fact modification is detectable.
- Access: tenants can query their own ledger through the product and via API. The ledger is exportable for offline review by legal, compliance, or internal audit.
- Defensibility: the ledger is designed to answer the questions that come up under regulatory review or in litigation, namely what the agent knew, when it knew it, and what it did with that knowledge.

Model governance options

Cortex does not bind buyers to a single model or a single provider.

- Open weight inside the boundary: run open weight models inside the tenant's PCU, so prompts and context never leave the boundary. Suitable for the most sensitive workloads.
- Frontier providers under enterprise contracts: route to commercial frontier models under contracts that include zero retention or contractually bounded retention, no training on customer data, and regional processing commitments.
- Mixed routing per workflow: assign different models to different workflows or different sensitivity tiers inside the same tenant. Routing policy is part of the tenant configuration and is recorded in the audit ledger.
- Provider change: switching providers does not require re-architecting the tenant. Memory, policy, and audit ledger are independent of the model in use.

Deployment modes

Cortex supports the deployment modes that regulated and sovereign buyers expect.

-

Public deployment: multi tenant Cortex on shared infrastructure, with the per-tenant isolation guarantees described above. Suitable for teams that do not have a hard residency or single tenant requirement.

- Private deployment: single tenant Cortex on dedicated infrastructure managed by GREENPOW. Suitable for organizations that need a dedicated control plane and dedicated data plane.
- Sovereign region: deployment pinned to a specific jurisdiction, with processing, storage, and key management inside that jurisdiction. Suitable for public sector, regulated finance, health, and other markets with hard residency requirements.
- On premises or customer cloud: Cortex deployed inside the customer's own data center or cloud account, with GREENPOW providing the software and operational guidance.

Deployment mode is chosen per tenant. A single customer can run different tenants in different modes if their portfolio requires it.

What this page is, and what it is not

This page describes the architecture Cortex is designed around. It is a starting point for a vendor review, not a substitute for one. Buyers in scope for a Cortex deployment receive:

- A detailed architecture and security review under NDA.
- A walkthrough of the relevant deployment mode and the controls that apply to it.
- Confirmation of the certifications, contractual terms, and operational commitments that apply to the specific engagement.

To start that conversation, see the Cortex product page or contact GREENPOW directly.

Source: greenpow.com/en/products/cortex/architecture

Source: greenpow.com/en/products/cortex/architecture

Source: greenpow.com/en/products/cortex/architecture

Source: greenpow.com/en/products/cortex/architecture

Source: greenpow.com/en/products/cortex/architecture