Data Security Overview

How Gentrail protects customer data end-to-end — across every deployment model, from shared SaaS to fully air-gapped on-premises. The same five-pillar security envelope applies to every mode, scaled in strength but never absent. This briefing covers *where* your data lives, *how* it is protected at each step of its lifecycle, and the specific controls that apply to your chosen deployment.

AUDIENCE CIO · CISO · SECURITY & COMPLIANCE TEAMS **CLASSIFICATION** SHAREABLE

PART I **Deployment** where the data physically lives, and who operates it

1 — PICK ONE

The Four Deployment Models

Every customer maps to exactly one model. The two axes that matter: **where the data physically lives**, and **who operates it**. These two axes also determine every cell in the security control matrix in §6.

Availability. M3 · Bring Your Own Cloud is available today. M1 · Shared SaaS, M2 · Dedicated SaaS, and M4 · On-Premises are on the roadmap (*coming soon*). The same five-pillar security architecture in §4 is designed to apply uniformly across all four models as each becomes generally available. Within this document, individual controls labeled *coming soon* are part of the target architecture and in active development; all other controls are live in the current release.

M1 · SHARED SAAS — COMING SOON

Shared SaaS

DATA LIVES	our cloud
WE OPERATE	everything
ISOLATION	logical (per-org)
NETWORK	full SaaS

M2 · DEDICATED SAAS — COMING SOON

Dedicated SaaS

DATA LIVES	our cloud, your region
WE OPERATE	everything
ISOLATION	physical stack
NETWORK	full SaaS

M3 · BYOC — AVAILABLE NOW

Bring Your Own Cloud

DATA LIVES	your cloud account
WE OPERATE	nothing at runtime
ISOLATION	your account
NETWORK	artifacts pulled at install

M4 · ON-PREM — COMING SOON

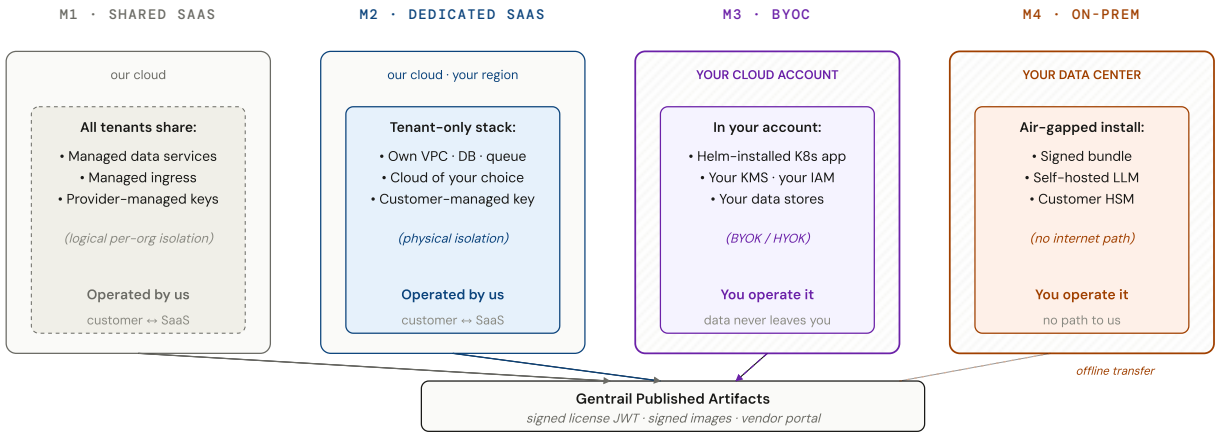
On-Premises

DATA LIVES	your data center
WE OPERATE	nothing
ISOLATION	your hardware
NETWORK	none (air-gapped)

2 — THE VISUAL

Where Your Data Sits in Each Model

The single most important question a regulated enterprise buyer asks is "where does our data actually live, and what touches it?" This diagram answers it for all four models. The shaded boundary is the customer's environment.

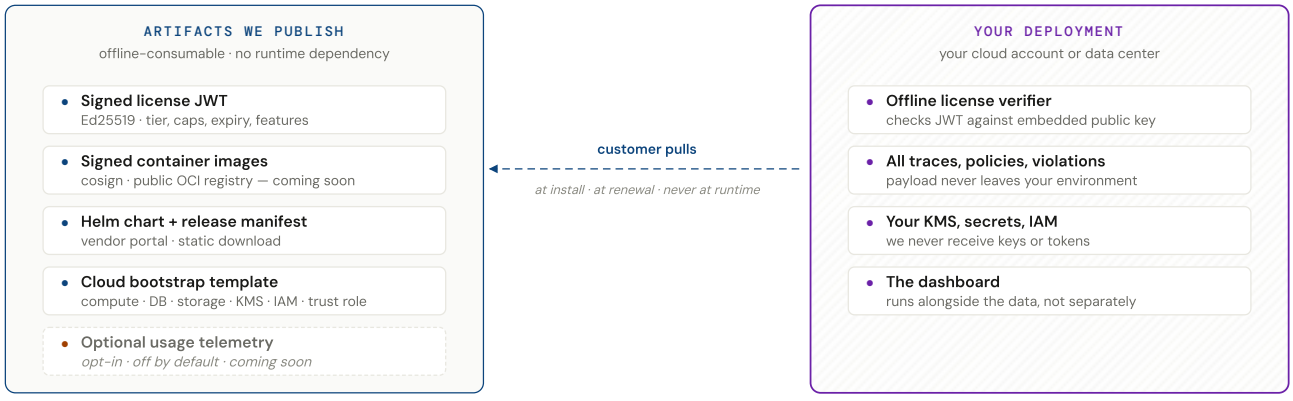


M1 / M2 we operate directly. M3 pulls signed artifacts from us at install and renewal — never at runtime. M4 receives the same artifacts via offline channel. Hatched fill = customer's own environment. ***M3 is available today; M1, M2 and M4 are coming soon.***

3 — WHAT WE SHIP VS WHAT YOU RUN

No Phone-Home — Offline License, Pulled Artifacts

For BYOC (M3) and On-Prem (M4), Gentrail follows the established self-hosted enterprise pattern (HashiCorp Vault, Elastic, Confluent, Splunk, Cribl): we **publish signed artifacts**, your deployment **verifies them offline**. There is no runtime callback to our cloud — license validity does not depend on us being reachable. The only optional connection is opt-in usage telemetry, off by default and never containing customer data.



Our cloud is on the install / upgrade path, not the runtime path. If our cloud is offline, your deployment keeps running — license JWT is verified locally against an embedded public key.

Enforcement curve. Soft enforcement: warnings at 30 / 14 / 7 / 1 days pre-expiry. Post-expiry we refuse *new agent registrations* but **never disable the data plane** for a paying customer who is past-due. A governance product that silently stops auditing during an incident is unacceptable. Hard stop is reserved for cryptographic invalidity (tampered license, bad signature), not expiry.

PART II **Data Security** — how customer data is protected inside every model

4 — PILLARS

Five Layers of Defense, Every Model

Every deployment model implements the same five pillars. The *strength* of each control varies by model (shared key vs HYOK; logical vs physical isolation — see §6) but the *presence* of every pillar is non-negotiable. If a pillar is missing for a model, the model is not shippable.

P1 · MINIMIZE — COMING SOON

Data Minimization

SDK-side redaction. Field-level allow-list. Tokenization of PAN / SSN / email. We never persist what we don't need.

P2 · ISOLATE

Tenant Isolation

Logical (M1) → physical stack (M2) → customer-cloud (M3) → customer data center (M4). Every query carries an org boundary enforced in code *and* in the storage schema (org-partitioned keys).

P3 · ENCRYPT

Encryption & Keys

TLS 1.3 in transit. AES-256 at rest (storage-level). Per-tenant envelope encryption — *coming soon*. Key custody follows the deployment: M1 / M2 — provider-managed KMS in our cloud. M3 — customer's native KMS, customer holds the root. M4 — customer HSM or on-prem KMS, air-gapped from us.

P4 · RESTRICT

Identity, Authorization & Access

AuthN: password + session today; SSO (OIDC / SAML), SCIM provisioning and mandatory MFA — *coming soon*. **AuthZ:** role-based access with org-scoped grants enforced on API calls today; standardized role set, custom roles and step-up auth on sensitive ops — *coming soon*. **Infra:** least-privilege cloud IAM (workload-identity roles); mTLS service-to-service — *coming soon*.

P5 · OBSERVE

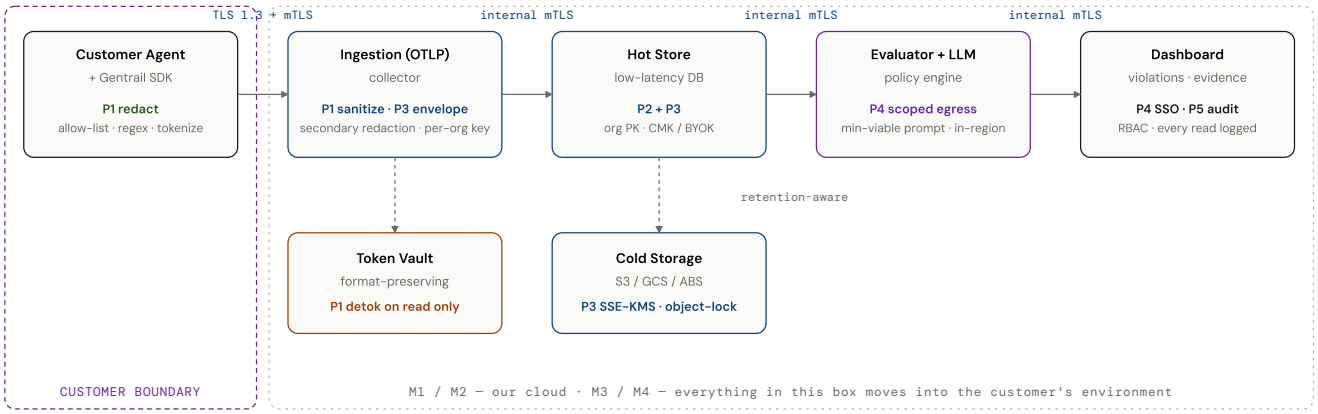
Audit & Response

Structured admin-action audit log, customer-readable in the dashboard today. Tamper-evident hash-chaining, SIEM forwarding (webhook / Kafka / S3), and signed-bundle provenance for the supply chain — *coming soon*. Documented breach-response runbook.

5 — THE VISUAL

Customer Data Lifecycle

Where customer data lives at every step from the agent through to the dashboard, and the protection applied at each hop. Every box is a place where data can leak; every label between boxes is the control that prevents it. The lifecycle is identical in all four models — what changes is the perimeter that wraps it, as shown in §2.



Same lifecycle in all four models. The perimeter changes per §2, the controls per §6.

Reading the labels. Each box names the system and the pillars enforced inside it; each arrow names the protection on the wire. The shorthand is intentionally compressed — the controls behind each label are the same five pillars from §4, written here as they appear in practice.

> **Control roadmap.** The lifecycle above shows the full target architecture. Controls labeled *coming soon* below are in active development; the remaining controls are live today.

OTLP. OpenTelemetry Protocol — the CNCF-standard wire format for traces, metrics, and logs. Gentrail accepts spans only via OTLP over HTTP (gRPC — *coming soon*); customer agents emit it natively or via an OTel Collector that translates from their existing instrumentation. Same protocol in all four models — only the collector's location changes.

P1 redact (at the SDK). `COMING SOON` The earliest possible point — *before* data leaves the customer's process. Three techniques combine: an *allow-list* of permitted span attributes (everything else is dropped), *regex / Presidio detectors* for things that look like emails / SSNs / PANs / JWTs / private keys (matches replaced with redaction markers), and *tokenization* for values that need to remain referenceable but not exposed in plaintext (replaced with a format-preserving fake; the real↔token mapping lives in the customer-side token vault).

P1 sanitize (at the collector). `COMING SOON` Secondary redaction at ingest. Treats every incoming span as untrusted regardless of what the SDK did — re-enforces the allow-list, re-runs the platform's current detector set, drops oversized or malformed fields, and emits a `sanitized` marker as auditable evidence. Defense in depth: if the SDK is misconfigured or out of date, the collector catches it.

P3 envelope (at the collector). COMING SOON Envelope encryption. Each span is encrypted with a fresh per-object *DEK* (Data Encryption Key) using AES-256-GCM; the DEK is then wrapped by a long-lived per-tenant *KEK* (Key Encryption Key) that lives in the KMS / HSM and never leaves it. Three properties this gives us: (1) high throughput — KMS is called once per batch, not once per span; (2) per-tenant cryptographic isolation — same bucket, separate keys; (3) crypto-shredding — destroying the KEK makes every span under it permanently unreadable, the basis of tenant-offboarding and GDPR erasure at scale. **Selective, not whole-record:** only sensitive body fields (`prompt` , `completion` , tool I/O, evidence) are ciphertext. Routing and index fields (`org_id` , `trace_id` , `policy_id` , `agent_id` , `severity` , timestamps) stay plaintext under the tenant boundary so the database can serve dashboard, monitoring, and policy-evaluation queries without ever touching ciphertext. Decryption is a low-volume, audited, on-demand operation triggered only when a permitted role views a specific span or exports evidence.

P1 detok on read only (at the token vault). COMING SOON The mapping from token back to real value is held in a vault separate from the data path. The platform *never* resolves tokens automatically — only an explicit, audited read by a permitted role (with step-up auth, per P4) can request the underlying value. Most reads — dashboards, analytics, evaluator inputs — work on tokens alone and stay out-of-scope for the underlying data class (PCI / PHI).

P4 scoped egress (at the evaluator). The only path that leaves the platform to a third party (the LLM). What goes out is a minimum-viable extraction prompt — policy document text, not span bodies — enforced in the engine today. Zero-Data-Retention provider configuration, in-region endpoints, and customer-chosen LLM endpoints with an egress allow-list — *coming soon*.

TLS 1.3 / mTLS (on every arrow). TLS 1.3 for the external hop (agent → ingest) — modern ciphers, mandatory forward secrecy — enforced at the ingress today. mTLS for every internal hop (workload SVIDs inside the cluster) — *coming soon*; internal traffic is currently segmented with network policies.

6 — THE MATRIX

Controls per Deployment Model

Read across a row to see how a control scales with deployment model. Read down a column to see the full control envelope for a single model — this is the reference for your security team during procurement, alongside §2 and §5. Cells labeled *coming soon* describe the target control for that model and are in active development.

CONTROL DOMAIN	M1 SHARED SAAS COMING SOON	M2 DEDICATED SAAS COMING SOON	M3 BYOC AVAILABLE NOW	M4 ON-PREM COMING SOON
Data residency	Single region	Customer-chosen region	Customer's cloud account	Customer's data center
Tenant isolation	Logical (per-org PK) + IAM	Physical: own VPC, DB, queue, KMS	Customer account = perimeter	Customer hardware = perimeter
Encryption at rest	AES-256, provider-managed KMS	AES-256, CMK	AES-256, BYOK (customer KMS / Vault)	AES-256, HYOK (customer HSM)

CONTROL DOMAIN	M1 SHARED SAAS COMING SOON	M2 DEDICATED SAAS COMING SOON	M3 BYOC AVAILABLE NOW	M4 ON-PREM COMING SOON
Encryption in transit	TLS 1.3 ext; mTLS int	TLS 1.3 + mTLS; customer cert option	TLS 1.3; service-mesh mTLS *coming soon*	TLS 1.3 + customer PKI; no ext egress
Key control	We hold	Customer CMK; we use via grant	Customer holds; we never see key	Customer HSM; key never exits customer
PII redaction	SDK + ingest (default rules)	SDK + ingest + custom policy	SDK + ingest; customer policy *coming soon*	SDK + ingest; customer policy
Tokenization	Shared vault infrastructure, per-tenant KEK; mappings envelope-encrypted at rest	Per-tenant vault namespace + per-tenant KEK	In-cluster vault; never leaves customer *coming soon*	In-cluster vault; never leaves customer
LLM data path	Managed API · ZDR	Bedrock / Vertex / Azure OpenAI in-region	Customer-chosen provider; egress allow-list *coming soon*	Self-hosted only (vLLM / TGI); no egress
Identity / SSO	Password + session; OIDC opt-in	OIDC + SAML + SCIM	Password + session today; OIDC + SAML + SCIM + workload identity *coming soon*	SAML + customer IdP; air-gap auth
Authorization (RBAC)	4 built-in roles; step-up auth on sensitive ops	4 built-in + custom roles; SCIM group → role mapping	Org-scoped role grants today; custom roles + resource-scoped permissions *coming soon*	Custom roles + customer-defined separation of duties
API token scopes	Role-bound; tenant-scoped	Role-bound + IP allow-list; rotation API	Org-scoped tokens today; role / resource scoping + short-lived issuance *coming soon*	Role + resource-scoped; customer-issued only

CONTROL DOMAIN	M1 SHARED SAAS COMING SOON	M2 DEDICATED SAAS COMING SOON	M3 BYOC AVAILABLE NOW	M4 ON-PREM COMING SOON
Staff access to data	Break-glass, audit-logged, time-bound	Break-glass, customer-notified	None at runtime; customer-initiated session	None ever; remote support is screen-share
Audit log	Append-only; 12-mo retention	Append-only + customer SIEM forwarding	Customer owns log store	Customer owns log store
Backup & DR	Cross-AZ; 35-day PITR	Cross-AZ + cross-region opt-in	Customer policy; we ship playbook	Customer policy; offline restore drill
Breach notice SLA	72h (GDPR)	24h contractual	Customer is operator; we notify on artifact CVE	Customer is operator; we notify on artifact CVE

M1 tokenization isolation — design note (coming soon, with M1). M1 keeps a single vault cluster for cost and operational efficiency (the entry tiers serve many self-serve tenants), but each tenant's token mappings are envelope-encrypted at rest with a *per-tenant KEK* in KMS before being written to the vault — the same pattern as P3 envelope on span bodies, applied to the vault entries. Blast radius: an attacker must compromise *both* the vault process *and* a tenant's KMS key to recover that tenant's mappings; backup leaks yield ciphertext only; crypto-shred on tenant offboarding by destroying the tenant KEK. Per-tenant vault *clusters* are provided from M2 up.