OMNAFY

HIPAA and the AI governance layer

Omnafy's compliance posture, pre-certification

Audience	Compliance officers and auditors
Version	1.0
Published	August 31, 2026
Online	omnafy.com/whitepapers/hipaa-compliance-posture/

Abstract

Two boundaries on this document, stated before anything else so nothing later can be read more broadly than it should be.

Nothing here is a certification claim. Omnafy holds no SOC 2 report and has engaged no auditor. HIPAA has no certification regime at all, so the posture described in section 3 is a matter of design, contract, and evidence rather than attestation. Section 8 describes SOC 2 readiness, meaning the artifact base a future examination would draw on, and names the gap between documented design and examined control wherever a reader might otherwise infer more.

HIPAA and SOC 2 are the whole scope, deliberately. ISO 27001 and GDPR are out of scope today: the platform has no EU data subjects, no EU deployment, and no customer demand for either framework. Section 10 enumerates the exclusions.

With that said, the substance. Most platforms treat HIPAA and SOC 2 as a controls-documentation exercise layered onto an existing design. Our position is the reverse. The five architectural invariants the platform is built on were chosen so that the compliance story is a description of the architecture rather than a promise laid over it. Protected health information never reaches infrastructure we operate, because every component that sees a payload runs inside the customer's own cloud account. Model inference runs under the customer's own agreement, never through credentials we hold. And the audit record an auditor will query exists as a structural property of the one enforcement path rather than as a logging convention each component has to remember.

This paper maps that design onto each framework's obligations, assigns responsibility per control family, states the data lifecycle including retention and deletion, and inventories the artifacts we hand over when a compliance team asks for evidence.

1. Why the architecture does the work

Four design decisions carry most of the compliance weight. Each is a decision of record elsewhere in our documentation; this paper only maps them.

DESIGN DECISION	WHAT IT BUYS COMPLIANCE
Regulated data stays in the customer's boundary; it never enters our control plane	Our business-associate exposure is confined to redacted metadata, and the surface that processes protected health information stays under agreements the customer already holds
Redaction at the boundary under a deny-by-default field policy	The upward data flow is enumerable and reviewable. "What does Omnafy receive" has a finite, auditable answer

DESIGN DECISION	WHAT IT BUYS COMPLIANCE
In-boundary execution: gateway, runtime, adapters, transcripts, and full-fidelity audit all deploy in the customer's own cloud account	The customer's existing cloud agreement and account controls cover the entire protected-data path, including model inference
Total, attributable audit: every action, every outcome, the full principal chain, append-only	HIPAA audit controls and SOC 2 monitoring criteria are satisfied by one structural property of one enforcement path

The consequence to state once: when an auditor asks to see every access to protected health information by an AI caller, the answer is not a log-aggregation project. It is a query against a single append-only trail that exists because no path bypasses the gateway that writes it.

2. Where the data actually lives

Compliance conversations go wrong when the residency question is answered in generalities, so here it is as a table. Every category of data the platform touches has an assigned place it may live and, the operative property, a place it must never appear.

DATA CLASS	MAY LIVE IN	MUST NEVER APPEAR IN
Protected health information	Customer business systems, adapter traffic, transcripts, full-fidelity audit, the customer's model path	The Omnafy control plane in any form: redacted audit store, metering, telemetry, logs
Tool-call payloads and transcripts	Customer-side stores only	Anything crossing the deployment boundary
Tool results delivered to staff AI clients	The staff member's client and the model path the customer has sanctioned behind it	The Omnafy control plane
Full-fidelity audit events	Customer-side append-only store	The Omnafy SaaS
Redacted audit envelopes	Generated customer-side, stored in the SaaS	Constrained by content rather than location: no payload fields, no protected data, ever
Policy state	Authoritative in the SaaS, replicated to the customer gateway	Contains no regulated data by construction
Definitions	SaaS registry, replicated to the customer runtime	Must never embed regulated data or credentials, validated at draft and again at approval
Secrets and credentials	Customer-side secret store; SaaS-side store for our own service credentials	Definitions, transcripts, audit events, telemetry, source control
Metering and health telemetry	Crosses upward	Any field not affirmatively classified as safe

One row deserves a section rather than a cell, and it is section 5.

3. The agreement chain

Four business associate agreements cover the platform's topology, three held by the customer and one held by us with our own infrastructure provider, plus one relationship the customer holds that no agreement of ours can substitute for. The customer in this chain is a covered entity, or for future customers an upstream business associate; the chain composes identically in either case.

Customer to cloud provider. The customer's own cloud agreement covers everything inside their account: the gateway, the runtime, adapters, the transcript and full-fidelity audit stores, and, when they choose the in-account model service, inference itself. We deliver versioned artifacts into this account. We are not a party to this agreement and do not need to be, because we do not operate that infrastructure.

Customer to model provider, for the automation path. When the customer calls a model API directly rather than the in-account service, their own agreement with that provider covers the model path. The credentials are theirs. We never proxy regulated data through model credentials we hold. Only one of the two automation model-path agreements is required, depending on the chosen path, and both may coexist.

Customer to Omnafy. We sign a business associate agreement covering the control-plane metadata stream: redacted audit envelopes, metering counters, health telemetry, approval-queue projections, policy state, and definitions. By design none of this contains protected health information, and we sign anyway, for the reasons in section 4.

Omnafy to our cloud provider. Section 4's argument obligates us consistently downstream. If the control-plane metadata is worth an agreement from us because its identifiability is not trivially arguable away, then the provider hosting that metadata in our account is our subcontractor handling it, and HIPAA requires a business associate to hold agreements with such subcontractors. We therefore maintain our own agreement covering the SaaS account. The position cannot be half-held: we do not simultaneously argue that the metadata is debatable enough that we sign and then stay silent about the party hosting it.

Customer to the model providers behind sanctioned staff clients. Under governed workspace access, tool results delivered to a staff client, including protected data a scoped principal is entitled to read, travel to that client and its model backend. That path sits outside every agreement of ours, by design. The customer sanctions clients whose model path meets its own compliance bar and holds the agreement with the provider behind each. Section 5 develops this.

The chain has a property that matters to a covered entity's risk assessment: **on every protected-data path, the operative agreements are contracts the customer holds directly with providers who sign them routinely.** We do not insert ourselves as a subcontractor business associate on any of them. So the customer's risk assessment of Omnafy reduces to assessing a metadata stream, which is a far smaller question than assessing a vendor that processes protected health information.

4. What Omnafy is, and is not

Our position is that we **are** a business associate of the customer, scoped to the control-plane metadata we receive, and we sign on that basis.

That deserves explanation, because the architecture is engineered so that no protected health information reaches us, and one could argue no agreement is needed at all. We reject that argument deliberately, for three reasons.

Defense in depth over definitional comfort. The redaction pipeline is deny-by-default and reviewed, but a redaction defect is a conceivable failure and is a tracked threat in our threat model. If a field carrying protected data ever crossed the boundary, the difference between "business associate with a signed agreement and breach-notification obligations" and "vendor with no HIPAA relationship arguing about conduit status" is the difference between an incident with a defined procedure and a legal crisis. The agreement is pre-positioned handling for the failure the architecture works to prevent.

The metadata is not trivially non-PHI. Redacted envelopes carry principal chains, tool names, timestamps, and payload hashes. Our design position is that this envelope contains no protected health information, but identifiability arguments about metadata are exactly the kind of question a covered entity's counsel should not have to win. The agreement makes the question moot.

The buyer expects it. A security or compliance leader evaluating us should not need a novel legal theory to approve the vendor. "They sign, and here is the enumerated list of what they receive" is an approvable posture. "They say they don't need to" is not.

What we are **not**, and what the agreement therefore does not cover:

Not a covered entity. We provide services to covered entities and business associates and have no treatment, payment, or health-care-operations role of our own.

Not a business associate on the protected-data path. Referral documents, patient identifiers, clinical content, tool-call payloads, transcripts, and full-fidelity audit events never transit infrastructure we operate. The customer's agreements with their cloud and model providers cover that path, and our

agreement explicitly does not extend to data classes that, per section 2, must never appear in our control plane.

Not a model provider or model intermediary. Inference happens on the customer's path with the customer's credentials. We hold no model credentials on any regulated path, so there is no model-path subcontractor relationship to paper over. Our one subcontractor relationship is our cloud provider hosting the SaaS metadata, covered by the agreement in section 3.

Not a decision-maker on clinical questions. Tier T3 exists precisely so that professional judgment stays structurally human. No component of ours, and no agent governed by one, decides a T3 question.

5. The flow our architecture cannot constrain

The largest egress channel for protected health information in the entry product is the governed one. We state it rather than eliding it, because a compliance reviewer will find it and should find it here first.

When a scoped staff principal reads records through a desktop AI client, the result travels to that client and whatever model backend sits behind it. It leaves the customer's cloud account lawfully, by design, because delivering the result is the point of the product. The residency invariant constrains our infrastructure and the automation model path. It does not, and cannot, constrain what happens inside a client the customer has sanctioned to receive results.

The platform bounds this flow with two mechanisms. **Scope**, meaning minimum-necessary grants, decides what a principal can obtain at all. **Audit**, meaning every delivery is on the record, makes what was delivered to whom queryable.

The customer bounds it with a third: sanctioning only clients whose model path meets their compliance bar, and holding the agreement with the provider behind each. That is a named customer responsibility in the matrix below rather than fine print, and our threat model records it as an explicit carve-out from scope, exactly as a compromised identity provider is the customer's identity control.

6. Mapping to the Security Rule

HIPAA Security Rule obligations fall on the covered entity and on each business associate for the systems it operates. This table maps the rule's safeguard areas to the platform mechanisms that address them in the customer's deployment: the design substrate the customer's own compliance program builds on. It is a design mapping rather than an attestation, and section 7 states who owns which half of each row.

SECURITY RULE AREA	PLATFORM MECHANISM
Access control, unique user identification	Every caller is a principal, one identity model for humans and agents and services, no shared automation accounts because agent principals are minted per run
Access control, authorization and least privilege	Scopes with tier ceilings and tenant slices, two enforcement points, principal-scoped downstream credentials, fail closed everywhere
Access control, automatic logoff analogue	Short-lived tokens for every principal class, no long-lived static credentials
Audit controls	Append-only audit events for every action and every outcome including denials, with the full principal chain, write-path separation, and integrity checking
Integrity	Append-only stores, manifest hash pinning at registration, versioned and diffable definitions in the registry
Person or entity authentication	Human principals resolve through the customer's own identity provider over OAuth 2.1 and OIDC; the gateway refuses any session it cannot resolve
Transmission security	TLS on every channel; boundary redaction under a deny-by-default field policy on the one flow that leaves the customer boundary
Risk analysis	Our threat model: enumerated vectors, mitigations, and explicitly accepted residual risks, maintained as a standing risk analysis for the AI-access surface
Workforce security	Deprovisioning inherited from the customer identity provider, policy-plane disable effective on the next call, approval authority bound to named human principals

Minimum-necessary discipline, a Privacy Rule concept, has a structural analogue: visibility filtering means an AI caller never even sees tools outside its scope, and scope grants are the customer's instrument for confining each principal, human or agent, to the smallest tool surface its role requires.

7. Shared responsibility

Responsibility for each control family splits three ways: Omnafy for the platform and the SaaS control plane, the customer for their deployment and their people and their policy decisions, and the cloud and model providers under the customer's direct agreements. This matrix is the skeleton for both the HIPAA conversation and a future SOC 2 system description; the SOC 2 notion of complementary user-entity controls corresponds to the customer column.

CONTROL FAMILY	OMNAFY	CUSTOMER	CLOUD AND MODEL PROVIDERS
Identity and authentication	Principal model, session-to-principal resolution, token-lifecycle rules, agent-principal minting	Operate the identity provider, workforce identity lifecycle, deprovision promptly	Identity primitives

CONTROL FAMILY	OMNAFY	CUSTOMER	CLOUD AND MODEL PROVIDERS
Authorization	Scope grammar and subset enforcement, tier enforcement at both points, fail-closed posture	Assign scopes and ceilings, ratify tiers at registration, configure approval routing, decide graduations	Policy primitives underlying least-privilege credentials
Audit and monitoring	Emit and store audit events, integrity checking, gap alarms, queue-health metrics in the console	Retain and review the customer-side full-fidelity trail, review queue health, act on anomalies	Infrastructure logs in the customer account
Data residency and confidentiality	Redaction pipeline and deny-by-default field policy, tenant partitioning and per-tenant encryption contexts	Operate the account holding all regulated data, own the encryption keys, classify data entering adapters	Encryption at rest and in transit, physical security
Transmission security	TLS on gateway and boundary channels, the enumerated crossing table	Network controls inside their account, so tool servers are reachable only through the gateway	TLS termination and network fabric
Model-path safeguards	Structural exclusion: no Omnafy-held model credentials on the regulated path	Hold and rotate model credentials, sign the provider agreements, choose the model path, sanction the staff AI clients admitted and the model path behind each	Handle inference data per the customer's agreement
Change management	Decision records plus review as the merge gate, versioned artifact delivery, manifest hash pinning with forced re-approval, the definition lifecycle	Review and approve definitions, which is human-only, and manage proprietary adapter changes in their own repositories	Provider-side change management per their attestations
Vulnerability management	Dependency and image scanning in the release pipeline, CVE monitoring for every delivered image, security fixes published as ordinary artifact versions with an advisory to customer contacts	Deploy patched releases within the advisory's stated window, since the customer's pipeline holds the deploy keys, and patch their own proprietary adapters	Patching of the managed services beneath both stacks per provider attestations
Supervision of automation	Approval queues with evidence, graduation thresholds, demotion triggers, re-certification, the kill switch	Staff the queues, make release and reject decisions, sign off promotions, exercise the kill switch when warranted	Not applicable
Incident response and breach notification	Detect and notify for control-plane incidents per our agreement, with demotion triggers and gap alarms as detection surface	Covered-entity breach-notification obligations, incident response inside their account, notification rights under their provider agreements	Provider notification duties under the customer's direct agreements

CONTROL FAMILY	OMNAFY	CUSTOMER	CLOUD AND MODEL PROVIDERS
Physical and environmental	Not applicable, no Omnafy-operated hardware	Not applicable	Entirely the providers' responsibility, inherited through their attestations
Availability and continuity	SaaS operations, per-run budgets and timeouts. Availability as a formally claimed criterion is not yet in scope, per section 8	Continuity for their own account and business systems	Regional infrastructure availability
Workforce and organizational controls	Our staff access controls, currently organizational on a small team, pending formal controls	Training and sanction policies for their workforce, rules for AI-client use	Provider workforce controls per their attestations

One asymmetry in the matrix is deliberate: **we own mechanisms and the customer owns decisions.** The platform enforces that a tier is checked, an approval is recorded, and a graduation is evidence-backed. Whether a given tool is T1 or T2, who staffs a queue, and which pairs graduate are customer judgments the platform records but never makes.

There is a structural backstop on the vulnerability-management row that is unusual and useful. Because upgrades run in the customer's pipeline, we cannot force a patch. But a customer stack that is never upgraded eventually falls outside the boundary protocol's supported version range, at which point it stops replicating policy state, hits its staleness bound, and fails closed. An unmaintained stack does not run stale software ungoverned indefinitely. It stops serving.

8. Data lifecycle

8.1 The redaction pipeline

One component in the customer stack holds credentials to publish to our ingestion endpoint, and every upward byte passes through it. Three rules govern what crosses.

Classification is affirmative and versioned. A field crosses only if it appears in the field classification table with a safe verdict. The table is a reviewed, versioned artifact shipped with the customer stack. A field absent from the table is dropped, and the drop is counted in health telemetry, so a new unclassified field surfaces as a signal rather than as a silent leak or a silent loss.

Free text never crosses. Error messages are mapped to enumerated error classes before transmission, because a raw vendor error string can embed regulated data. Identifiers cross; prose does not.

Payload hashes are keyed. Where an envelope carries a payload hash for correlation and integrity, the hash is an HMAC under a customer-held key that never leaves the customer account. A plain digest of a low-entropy field such as a date of birth or a phone number is invertible by dictionary. A keyed hash is not, and it also cannot be correlated across tenants.

The failure mode this inverts is the usual one. Under deny-by-default, an unclassified field breaks telemetry rather than confidentiality.

The keying has a second use a compliance team should know about. A customer holding the key and their own full-fidelity record can recompute the hashes and confirm that the envelope we hold corresponds exactly to the event they hold, without any payload crossing the boundary. The two trails are verifiably the same trail.

8.2 Retention

Two regimes, split exactly at the deployment boundary. **Customer-side retention is the customer's decision**, because the stores live in their account under their keys; we ship defaults and expose them as deployment parameters, and we can neither read these stores nor alter their retention. **SaaS-side retention is our commitment**, uniform across tenants.

Append-only stores on both sides use compliance-mode object lock, which makes each retention period a minimum as well as a maximum: a record cannot be deleted before its window expires, by anyone, including us. That is the tamper-resistance property behind the audit-suppression threat, and section 8.3 covers the consequence it has for deletion requests.

The six-year figure below is anchored to HIPAA's documentation-retention requirement at 45 CFR 164.316(b)(2). Customers under different regimes adjust the customer-side windows at deployment.

Customer-side defaults:

DATA CLASS	DEFAULT RETENTION	RATIONALE
Full-fidelity audit events	6 years	Audit-grade record, matching the envelope window so the two trails cover the same period
Transcripts	12 months	Debugging and evaluation value decays; customers treating transcripts as records of care extend this
Evidence bundles	24 months	Must outlive the longest graduation trailing window and re-certification cadence, since deleting a bundle invalidates the evaluation cases built on it
Adapter downstream credentials	Until rotation	Rotated on schedule, never retained past replacement

SaaS-side commitments:

DATA CLASS	RETENTION	RATIONALE
Redacted audit envelopes	6 years from event time	The customer-queryable trail, aligned with the customer-side window
Approval queue projections	90 days after decision or expiry	Working state; the decision itself is an audit event kept 6 years
Policy state, current	Life of the tenant	Every mutation is an audit event kept 6 years
Definitions, all versions including retired	Life of the tenant	The change-management record for agent behavior
Metering detail	13 months	Covers a full billing-dispute year
Metering monthly aggregates	7 years	Financial records, containing counts and dimensions and amounts only
Health telemetry, raw	90 days	Operational value only
Health telemetry, aggregated	13 months	Year-over-year capacity and reliability comparison

The customer's business systems, meaning the systems of record the adapters wrap, are governed by the customer's own retention program. The tables above cover only the copies the platform itself makes.

8.3 Deletion

Inside the customer boundary, deletion is structural. For every class that never crosses, there is no Omnafy-held copy to delete. We have no operator access, no credentials, and no network path into the customer account. The customer deletes transcripts, full-fidelity audit events, and evidence bundles by deleting objects, keys, or the stack itself, on their own authority and their own schedule, subject only to the object-lock windows they configured. A customer who decommissions the stack has removed every copy of their regulated data the platform ever made, because every copy the platform ever made was inside their account.

On our side, deletion is enumerated. Everything we hold for a tenant is listed in section 2, and deletion works per store type. Mutable stores, meaning the registry, policy state, queue projections, metering detail, and telemetry, are hard-deleted by tenant partition. Object-lock-bound stores, meaning the redacted audit envelopes, cannot be deleted early by anyone, deliberately, so deletion is achieved by key destruction: every tenant's SaaS-side data is encrypted under tenant-scoped key material, and destroying that material renders the ciphertext unreadable everywhere at once, including backups and replicas, which inherit the same key scoping. The locked objects then age out physically at their retention expiry.

The tension in that arrangement is real and we state it plainly. An append-only, tamper-proof store and a per-record deletion capability are mutually exclusive, and we chose tamper-proofing. If a redaction defect ever placed a regulated field into a locked envelope, per-record excision is impossible by design. The handling is layered instead: the deny-by-default policy makes the event structural and rare rather than probabilistic; the query index, which is mutable, is immediately masked for the affected envelopes, closing every read path the platform offers; the incident proceeds under our signed agreement with its notification obligations; and the locked object, unreadable through any interface and encrypted at rest, ages out at its retention expiry.

That is the actual shape of the guarantee, and it is why we sign the agreement even though the architecture aims to make it unnecessary.

8.4 Offboarding

WHEN	ACTION
Day 0	Tenant credentials for the replication and ingestion endpoints are revoked, and the artifact-pull grant is revoked. A still-running customer stack stops replicating, hits its staleness bound, and fails closed, so automation stops rather than running ungoverned
Days 0 to 30	Export window. The customer exports their redacted audit trail, all definition versions, and a final metering statement. The customer-side stores need no export, because the customer already holds them
Day 30	Tenant partitions are hard-deleted from all mutable stores, and tenant key material is scheduled for destruction. After the key-deletion window elapses, the tenant's envelopes and any backup containing them are unrecoverable
Retention expiry	Locked envelope objects age out of the redacted audit store per their object-lock windows
7 years	Monthly billing aggregates, the only tenant-derived data that outlives offboarding, reach the end of the financial-records period and are deleted

The resulting guarantees, in the form a customer's counsel should test them:

No protected health information is returned or destroyed at offboarding, because none was ever held. The answer to "what do you do with our data when we leave" is that we never had it, verified against the inventory in section 2 rather than asserted.

Everything we do hold is enumerable, and offboarding addresses every row. There is no miscellaneous class with an unspecified fate.

Unreadability arrives within 30 days plus the key-deletion window. After key destruction, nothing tenant-attributable is readable in any store or backup of ours except the billing aggregates above.

The customer's own record survives intact. Offboarding removes our copies and our access. The transcripts and full-fidelity trail in the customer's account remain theirs, on their retention terms.

The trail cannot be selectively edited on the way out. Deletion is total per tenant via key destruction, never per record, so a departing tenant's audit history is destroyed whole or preserved whole, and until destruction it remains append-only and integrity-checked.

9. SOC 2 readiness, and the gap

We are **not SOC 2 certified** and have not engaged an auditor. What exists today is the artifact base a future Type II examination would draw on: controls that are designed, documented, and in several cases structurally enforced, with no attested operating history.

9.1 Criteria in scope

Security, confidentiality, processing integrity, and privacy are the criteria our design speaks to today. **Availability is deliberately deferred.** The SaaS has no formal service-level objectives, redundancy attestations, or continuity testing to point at yet, and claiming the criterion before those exist would be exactly the over-claim this paper avoids.

9.2 Artifacts against criteria

CRITERION	THE CONTROL STORY
Security, logical access	One identity model for every caller, scopes with a computable subset comparison, two redundant enforcement points, fail closed, no long-lived credentials
Security, system operations and monitoring	Total attributable audit as a structural property, demotion triggers and gap alarms as automated monitoring, the threat model as the standing risk assessment
Security, change management	Every platform decision a reviewed and dated record, every behavior change to a governed tool or definition reopening human approval, agent behavior itself under version control
Confidentiality	Every data class carrying an assigned residency and a "must never appear in" constraint, so the confidentiality commitment is enumerable and testable
Processing integrity	Unauthorized calls cannot execute, approvals are evidence-based, autonomy is measured and reversible, cost accounting is per run
Privacy	Personal data governed by the same boundary discipline as protected health information; collection by the SaaS minimized structurally rather than procedurally
Availability	Deferred. Per-run budgets and timeouts exist; nothing else is claimed

9.3 What a Type II would still require

Naming the gap keeps this section honest. Beyond the artifact base above, a Type II examination requires a formal system description, stated control objectives with named owners, an operating-effectiveness evidence period of typically six to twelve months, and organizational controls at Omnafy itself, meaning access reviews and background checks and security training and vendor management, which currently exist informally on a small team.

Two further items belong here because auditors and security leaders ask for them by name.

Independent penetration testing of both stacks, covering the gateway ingress, the authentication broker handshake, the SaaS administrative surface, and the tenant partition boundary, is planned and has not yet occurred. No assessment report exists to hand over.

A formalized vulnerability-management program is the operating practice the matrix row in section 7 commits to. A Type II period is what turns that practice into attested evidence.

The insider-risk item in our threat model's accepted-risk list is this same gap seen from the security side. It closes when the organizational controls are formalized, and both documents will be updated together when it does.

10. Evidence inventory

When a compliance team, a prospective auditor, or a covered entity's counsel asks for evidence, these are the artifacts and what each demonstrates. Everything listed is either documentation we hand over or a record the running platform generates.

EVIDENCE ARTIFACT	WHAT IT DEMONSTRATES
Architecture overview and deployment topology	The trust boundaries, the enumerated boundary-crossing table, and the data-residency design: the system description in embryo
Security model	Identity, authorization, invariant enforcement, and the data classification table: the normative control design
Threat model	The standing risk analysis: vectors, mitigations, owners, and explicitly accepted residual risks
Data handling document	The data lifecycle: per-plane inventory, the redaction field policy, retention windows on both sides, and the deletion and offboarding guarantees
The glossary	Fixed control vocabulary, so an auditor's terms map to exactly one definition each
The decision log	Change management for architecture: every decision dated, statused, and reviewable, with the review trail in version history

EVIDENCE ARTIFACT	WHAT IT DEMONSTRATES
The published contracts	The enforced interfaces: what a manifest must declare, what an audit envelope must and must not contain, how scopes compare
The customer's audit trail	Operating evidence: who did what, under whose authority, with what decision, queryable by the customer for their own tenant
Approval-decision and graduation-change event streams	Supervision operating as designed: human decisions with evidence, autonomy earned and revoked on the record
Registration records and pinned manifest hashes	Tool-surface change control: what was ratified, when, by whom, and that nothing changed without re-approval
Metering reports	Per-run cost accounting supporting billing accuracy
Signed agreements	The complete contract chain of section 3

Two practices keep this inventory audit-ready rather than merely audit-adjacent. Every document carries a status line with owner and last-updated date on its first line, so staleness is visible at a glance. And link integrity between evidence artifacts is checked automatically, so the inventory cannot silently fragment.

11. Explicitly out of scope

Restating the introduction's boundaries as commitments, for the avoidance of doubt.

ISO 27001 is not pursued. The SOC 2 readiness work of section 9 is the closest analogue, and an information-security-management-system mapping would be new work undertaken only against a concrete customer requirement.

GDPR is not addressed. No EU data subjects, processing activities, or deployments exist. The boundary and redaction discipline would be the starting point for a data-processing analysis if that changes. No such analysis has been performed.

HITRUST, state privacy statutes, and PCI DSS are unexamined. None currently applies to the platform's data flows.

Any certification claim. None exists. This paper describes design posture for HIPAA and readiness seeds for SOC 2, and should be read that narrowly.

Further reading

Governing AI access to systems of record is the security companion: the five invariants, identity and authorization mechanisms, the threat model summary, and the published accepted-risk list. *The Omnafy platform architecture* covers the planes, the gateway, and the deployment split in engineering detail. *Earned autonomy* covers risk tiers, approval queues, and graduation.

Customers, prospective customers, and their auditors can request the full internal documentation set, including the complete threat model, the data handling document, the audit events contract with its schemas, and the decision log.