SECURITY & COMPLIANCE

Security & Compliance Overview

How the Tactis platform protects Protected Health Information for skilled-nursing facilities — architecture, controls, and their mapping to the HIPAA Security Rule and the SOC 2 Trust Services Criteria.

DOCUMENT	Security & Compliance Overview
AS OF	26 July 2026
CLASSIFICATION	Public — may be shared freely
AUDIENCE	Security & compliance reviewers

Program status — as of 26 July 2026

The platform's security architecture is implemented and described in the present tense throughout this document. The items below are program and contractual milestones whose status is stated explicitly so that no reader has to infer it. This block is the authoritative status of record for the dated items and is updated on each issue of this document.

ITEM	STATUS AS OF 26 JULY 2026
Platform architecture & technical controls (this document)	`Implemented` Built and in code; see each section and Appendix A/B.
Subprocessor BAAs — AWS, Anthropic, OpenAI	`Being executed at go-live` The platform does not process customer PHI until these are in force.
OpenAI zero-data-retention (ZDR)	`Enabling at go-live` Applied to OpenAI API usage before any customer PHI is processed.
Customer (facility) BAA	`Offered at contracting` Tactis enters a BAA with each facility as part of onboarding.
Legal entity formation	`In progress` Completing formation; BAAs are executed by the formed entity.
SOC 2 examination (Type I / II)	`Available on request` Controls are designed to the 2017 TSC; no report has been performed to date (see §3).
Independent HIPAA risk analysis & written policies	`Go-live program` Formal risk analysis and policy set are produced as part of go-live (see §3).

Where a claim in this document depends on one of the milestones above, that dependency is stated at the point of the claim.

01 EXECUTIVE SUMMARY

Security built into the architecture, not bolted onto it

Tactis is a multi-tenant SaaS platform that helps skilled-nursing facilities manage survey deficiencies, plans of correction, QAPI, and related documentation — work that necessarily involves Protected Health Information (PHI). Two architectural decisions define our security posture, and both are implemented today:

01

Per-facility cryptographic isolation, with crypto-shred offboarding

Every facility's PHI is encrypted at rest under a data-encryption key unique to that facility, wrapped by a customer-managed key in AWS KMS. One facility's key can never read another's data — the facility identifier is bound into the encryption itself. When a facility offboards, destroying its key renders its PHI permanently unreadable everywhere it exists, including inside retained backup snapshots. Deletion becomes a mathematical guarantee rather than a best-effort sweep.

02

A tamper-evident, hash-chained provenance ledger

Every access to PHI, and every disclosure of PHI to a subprocessor, is recorded in an append-only, cryptographically hash-chained ledger. The chain makes tampering evident: altering or deleting any historical entry breaks the chain and is detected by verification. The database itself refuses updates, deletes, and truncation of the ledger.

Around these two pillars sit the controls a security reviewer expects: mandatory two-factor authentication, per-request tenant scoping enforced at the data layer, least-privilege cloud IAM, encrypted transport and storage, secrets held outside the application, immutable and scanned container images, and scheduled encrypted backups with a documented recovery plan. The remainder of this document describes each of these precisely and maps them to the HIPAA Security Rule (Appendix A) and the SOC 2 Trust Services Criteria (Appendix B).

A note on how to read this document: security reviewers are right to distrust adjectives. Wherever this document makes a claim, it names — in monospace — the component or control that substantiates it, and it distinguishes what is implemented from what is a documented target. Deeper materials, including source-level walkthroughs, are available under NDA.

02 COMPLIANCE POSTURE

HIPAA Security Rule and SOC 2 — stated precisely

HIPAA Security Rule

There is no such thing as a “HIPAA-certified” product. Tactis is **designed to satisfy the HIPAA Security Rule** — its administrative, physical, and technical safeguards — and Appendix A provides a control-by-control mapping to §164.308, §164.310, §164.312, and §164.316. As a cloud service, the physical safeguards for the underlying data centers are inherited from AWS under the shared responsibility model and the AWS Business Associate Addendum. The formal, written HIPAA risk analysis and the full policy-and-procedure set are produced as part of go-live; the technical safeguards they govern are already implemented and are the subject of this document.

SOC 2

SOC 2 is not a certification a company grants itself — it is an independent auditor's report against the AICPA Trust Services Criteria. To describe our status honestly:

- Tactis's controls are **designed and mapped to the SOC 2 2017 Trust Services Criteria (with the 2022 revised points of focus)** — the Common Criteria (CC-series) and the Availability criteria. Appendix B provides that mapping.
- **No SOC 2 examination (Type I or Type II) has been performed to date.** We describe this as SOC 2 readiness, never as “SOC 2 certified” or “SOC 2 compliant.”
- An independent examination can be undertaken at a customer's request.

BUSINESS ASSOCIATE AGREEMENTS

Tactis operates as a Business Associate to its customers and enters a BAA with each facility. In turn, PHI is disclosed only to subprocessors that are themselves covered by a BAA (see §8). The execution status of these agreements is stated in the Program status block at the front of this document; the platform does not process customer PHI until the subprocessor BAAs are in force.

03 DATA PROTECTION & ENCRYPTION

Field-level encryption with per-facility keys

PHI is not protected only by disk encryption underneath the database. The specific columns that carry PHI — free-text clinical findings, plan-of-correction narratives, resident labels, uploaded document text and filenames, and dozens more — are individually encrypted at the application layer before they are written, under a key that belongs to a single facility.

The encryption envelope

Each protected value is stored as a self-describing envelope using **AES-256-GCM** authenticated encryption. The facility identifier is bound into the encryption as additional authenticated data, so a ciphertext copied into another facility's row fails authentication rather than silently decrypting — cross-tenant data confusion is caught by the cryptography itself, not merely by application logic. A distinctive prefix on every envelope makes the system fail loud: any plaintext that ever reached an encrypted column is trivially findable, and any value that reached a screen without being decrypted appears as visible ciphertext rather than as subtly wrong data.

Per-facility keys, envelope encryption, and key management

A single customer-managed key (CMK) in AWS KMS is the root of the scheme. For each facility, the application asks KMS to generate a data-encryption key (DEK); that DEK is used locally to encrypt the facility's columns, and only the KMS-wrapped form of the DEK is ever stored. The wrapped DEK is held in AWS SSM Parameter Store — deliberately **off the database volume** — so it does not ride along inside database backup snapshots. Plaintext DEKs exist only transiently in application memory. The CMK has AWS-managed annual key rotation enabled and a 30-day deletion window.

Search over encrypted data (blind index)

Randomized encryption defeats ordinary search. To preserve search without decrypting on the database, Tactis uses a **keyed blind index**: each searchable word is stored as an HMAC-SHA-256 of the token under a per-facility index key that is independent of the data key. A query is tokenized and HMAC'd the same way, narrowing to candidate records that are then decrypted and ranked in application memory. The index key is destroyed together with the data key on offboarding, so search dies with the data.

Encryption in transit

All external traffic is served over TLS 1.2/1.3 with HTTP Strict Transport Security. The uploads bucket rejects any non-TLS request through an `aws:SecureTransport` bucket policy, and object storage is server-side encrypted.

SCOPE, STATED PLAINLY

Field-level encryption protects PHI at rest. Like any system that must read, search, and reason over data, Tactis decrypts values into memory to process a request; the platform does not claim end-to-end or homomorphic encryption. Uploaded file bytes in object storage are protected by server-side encryption and are removed during offboarding (see §11) rather than by field-level encryption.

04 MULTI-TENANT ISOLATION

One facility cannot reach another's data

Isolation is enforced at the data-access layer, not left to individual queries to remember. Every database model operation is automatically scoped to the caller's facility, and the same seam performs the encryption and decryption described in §3.

Request-scoped tenant context

The facility a request acts within is derived on the server from the authenticated session — never read from a request body or a client-supplied parameter — and carried through the request in a context that the data layer reads. On organization switch, the active facility is re-derived from an authoritative membership record per request rather than trusted from the session.

Automatic query scoping

A database extension intercepts every model read and write and injects the facility predicate: reads are filtered to the facility, writes are stamped with it, and updates and deletes first verify the target row belongs to the caller's facility. Because per-facility encryption rides the same seam, a value written for one facility cannot be decrypted for another even if a query were somehow mis-scoped — the facility-bound authentication tag would fail.

Isolation that fails safe on drift

A structural check runs at start-up: every data model that carries a facility identifier must be explicitly declared as either tenant-scoped or deliberately unscoped, or the application refuses to start. Adding a new PHI-bearing table without a scoping decision is a deploy-time failure, not a latent isolation gap discovered later in production.

05 AUDIT & ACCOUNTABILITY

A tamper-evident record of every PHI access

Tactis records provenance for PHI through **Noctra**, an append-only, hash-chained ledger. Every PHI access, mutation, disclosure, and lifecycle action is sealed into a per-facility chain in which each entry cryptographically commits to the one before it.

Hash chaining makes tampering evident

Each ledger entry stores the hash of the previous entry and its own hash over a canonical serialization of its contents. Altering, reordering, or removing any historical entry breaks the chain from that point forward, and a verification pass detects exactly where. The chain is maintained per facility, and appends are serialized with a database advisory lock so concurrent writes cannot fork it.

The database enforces append-only

Tamper-evidence is backed by tamper-resistance at the database: triggers reject any update or delete of a ledger row, and a separate statement-level trigger rejects truncation — so the ledger cannot be wiped in a single statement. Retention pruning after the multi-year floor is a deliberate, privileged operation, not an ordinary capability.

Fail-closed for the events that matter

For the highest-integrity categories — PHI mutation, PHI disclosure, subprocessor export/extract, and facility lifecycle — a record that cannot be sealed causes the underlying operation to fail. An unrecorded disclosure is the one outcome the evidence chain cannot tolerate, so the system refuses to proceed rather than proceed unrecorded. Provenance spans roughly 90 PHI touchpoints across the platform.

06 IDENTITY, AUTHENTICATION & ACCESS

Strong authentication and least-privilege access

Authentication

- **Mandatory two-factor authentication.** TOTP-based second factor with single-use backup codes, enforced at the API boundary as a global guard: a session whose user has not enrolled can do nothing except enroll. The check is server-side, so it cannot be bypassed by talking to the API directly.
- **Invitation-only provisioning.** Account creation requires a valid invitation; open self-service sign-up is disabled.
- **Session hygiene.** Sliding-window sessions with an idle timeout on the order of 30 minutes; secure, HTTP-only, same-site cookies scoped to the application subdomain.
- **Credential-compromise response.** A password reset or change revokes the user's other active sessions server-side, so a stolen session does not survive recovery. Minimum password length of 12; per-endpoint rate limiting on sign-in, two-factor verification, and reset.

Authorization

Access within the application is governed by a capability matrix and permission guards, with privileged actions — such as writing to the shared regulatory knowledge base — gated behind explicit permissions rather than available to any authenticated user. Every authenticated action already runs inside the facility scope described in §4.

Cloud access

The application runs under a cloud role granted only the permissions it needs — pull access scoped to the project's own container repositories, narrowly scoped key, object-storage, OCR, and email grants, and no static credentials on disk. Administrative SSH access to the host is restricted to a single operator IP address; only the public web ports are open to the internet.

07 SUBPROCESSORS & DATA SHARING

PHI leaves the boundary only to BAA-covered destinations

Tactis is an AI-assisted product, so some processing involves third-party model and infrastructure providers. Which providers may receive PHI is governed by policy and enforced in code, and every crossing is recorded.

A policy-enforced egress boundary

Destinations are partitioned into trust zones. Providers covered by a Business Associate Agreement form the PHI zone; providers without one are restricted to a non-PHI zone that may receive only the public regulatory corpus (CMS survey guidance and the like) — never facility or resident data. The boundary is **fail-closed**: a destination with no zone assignment, or content whose classification is not permitted for a zone, is blocked. Facility-specific vector search was moved onto Tactis's own database precisely so that facility content never needs to reach a non-PHI provider.

Every disclosure is receipted

Each PHI disclosure to a subprocessor writes a **content-free** receipt into the provenance ledger — actor, facility, purpose, data classification, policy version, and a hash of the payload, but never the payload itself. The return leg (an embedding, OCR text, or a model completion) is receipted too and linked to the dispatch that caused it. If a subprocessor ever reported an incident, the scope of what was disclosed is answerable from the ledger rather than estimated.

Minimum necessary

Disclosures to model subprocessors follow a documented, routine-and-recurring protocol under 45 CFR §164.514(d)(3). Where a whole document (for example, a CMS Form 2567) must be processed to extract every deficiency, it is sent whole; CMS pseudonymizes residents and staff in that document at the source. Tactis deliberately does not claim automated PHI redaction to subprocessors: a partial redactor would create a promise it could not keep on every miss. The honest, defensible position is to disclose only to BAA-covered processors, minimize by protocol, and record every crossing.

Subprocessor register

SUBPROCESSOR	PURPOSE	RECEIVES PHI?
Amazon Web Services	Cloud infrastructure & hosting, KMS key management, object storage, Textract OCR, SES transactional email, Backup	Yes (under BAA)
Anthropic	Large-language-model generation for regulatory assistance	Yes (under BAA)
OpenAI	Text embeddings and model generation (zero-data-retention)	Yes (under BAA + ZDR)
Pinecone	Vector search over the public regulatory corpus only	No — non-PHI zone
Vercel	Front-end application hosting	No — no PHI by construction

BAA execution and OpenAI ZDR status are stated in the Program status block at the front of this document. PHI is not processed until the subprocessor BAAs are in force.

08 INFRASTRUCTURE & NETWORK SECURITY

A small, hardened cloud footprint

Tactis runs on AWS in `us-east-1`, deployed and managed entirely as code (Terraform), with remote encrypted state and native locking.

- **Encryption at rest.** EBS volumes are encrypted; object storage uses server-side encryption and enforces TLS-only access.
- **Secrets management.** Application secrets and wrapped per-facility keys live in AWS SSM Parameter Store as the source of truth; the on-host environment file is rendered per deploy and holds no unique secret material. A documented rotation procedure exists and has been exercised.
- **Network exposure.** Only ports 80 and 443 are open to the internet; administrative SSH is restricted to a single operator IP. Security headers (HSTS, `X-Content-Type-Options`, `X-Frame-Options`, `Referrer-Policy`) are set at the edge.
- **Container supply chain.** Images use immutable per-commit tags with registry-enforced immutability and scan-on-push; a mutable “latest” tag is not used.
- **Monitoring.** Container logs are centralized to a retained CloudWatch log group, with metric filters and alarms (application 5xx, ledger-write failure, instance health, disk, memory) notifying an SNS topic.

09 BACKUP, DISASTER RECOVERY & CONTINUITY

Backups today, with an honest current-vs-target picture

Tactis has scheduled, encrypted backups and a written disaster-recovery plan and restore-drill procedure. We state current capability and production targets separately, because a reviewer deserves to know which is which.

Backup inventory (implemented)

Backups are managed by AWS Backup — no manual snapshots — into a KMS-encrypted vault, under two plans:

PLAN	SCOPE	SCHEDULE	RETENTION	PURPOSE
Operational	Root & data volumes	Daily	14 days	Fast recovery from a broken host
Audit	Data volume only (by explicit ARN)	Weekly	~6 years	HIPAA §164.316(b)(2) documentation retention — the provenance ledger

Current recovery capability

The live environment is presently single-host in a single availability zone. Stated honestly, the current numbers are a recovery-point objective of roughly 24 hours (daily snapshot; no continuous point-in-time recovery) and a recovery-time objective of a few hours by an operator following the runbook. Backups are currently single-region.

Documented production targets (planned)

The disaster-recovery plan sets production targets of a 1-hour RPO and a 4-hour RTO, reached through a defined production-cutover program: a managed multi-AZ database with point-in-time recovery and automatic failover, cross-region backup copies, and key-durability hardening. These are **targets and planned hardening**, not current capabilities, and are presented as such.

Restore drills

A restore-drill checklist is documented and scheduled to run at least quarterly and after major changes; each run verifies that backups restore, the ledger chain still verifies, and encrypted PHI still decrypts. The first drill run is part of go-live.

10 SECURE DATA LIFECYCLE & OFFBOARDING

Provisioning, and a real purge on the way out

Provisioning

Onboarding a facility mints its dedicated data-encryption key and blind-index key, idempotently. No facility exists in the system without its own key material, and the data key is never regenerated for a facility that already has ciphertext under it.

Crypto-shred offboarding

Offboarding is a genuine purge rather than a best-effort delete. It runs in a fixed, fail-safe order: it first seals a lifecycle record of the purge into the provenance ledger (and aborts if that record cannot be written), then deletes the facility's uploaded file bytes from object storage, then **destroys the facility's key material**. Once the key is gone, every field-encrypted PHI value for that facility is permanently unreadable — in the live database and in every retained backup snapshot, which ordinary row deletion can never reach. The action is irreversible by design and is protected by a slug guard against purging the wrong facility.

11 SECURE DEVELOPMENT & VULNERABILITY MANAGEMENT

Keeping the supply chain and secrets clean

- **Secret hygiene.** A git-history secret scan was performed across the repositories; the few historical credentials it surfaced were rotated, and repositories are private. Secret scanning is run pre-commit going forward.
- **Immutable, scanned images.** Container images are built with immutable per-commit tags and scanned on push; deploys keep a rollback history.
- **Patch management.** Host patching is automated on a weekly maintenance window with reboot-if-needed, via AWS Systems Manager Patch Manager.
- **Dependency integrity.** The Python retrieval service installs from a hashed, fully-pinned lockfile with hash verification, so a tampered or unexpected package fails the install.
- **Remediation discipline.** This document follows a structured security remediation program (54 findings closed at the time of writing) with a status of record maintained per finding.

PROMPT-INJECTION RESISTANCE

Because the product retrieves documents and feeds them to language models, retrieved and external content is structurally delimited as untrusted data using a per-request random-nonce envelope, so an injected instruction inside a document cannot break out of the data channel into the instruction channel.

A APPENDIX A

HIPAA Security Rule control mapping

A summary mapping of the platform's technical and administrative safeguards to the HIPAA Security Rule. Physical safeguards for the underlying infrastructure are inherited from AWS.

SECURITY RULE CITATION	HOW TACTIS ADDRESSES IT
§164.308(a)(1) — Security management / information system activity review	Provenance ledger records PHI access & disclosure; centralized logs, metric filters & alarms
§164.308(a)(3)–(4) — Workforce & access authorization	Invitation-only provisioning; capability matrix + permission guards; per-request facility scoping
§164.308(a)(4)(ii) — Access controls between tenants	Automatic facility scoping; facility bound as AAD in encryption; boot-time scope assertion
§164.308(a)(5)(ii)(D) — Password management	Minimum length 12; per-endpoint rate limiting; session revocation on reset/change
§164.308(a)(7) — Contingency plan	AWS Backup (operational + audit plans); documented DR plan & restore-drill procedure
§164.308(b)(1) — Business Associate contracts	Egress restricted to BAA-covered subprocessors; per-disclosure receipts (BAA execution: see status block)
§164.310 — Physical safeguards	Inherited from AWS data centers under the shared-responsibility model & AWS BAA
§164.312(a)(1) — Access control (unique ID, encryption)	Per-user authentication + mandatory 2FA; field-level AES-256-GCM with per-facility keys
§164.312(a)(2)(iv) — Encryption & decryption	AES-256-GCM envelope; CMK-wrapped per-facility DEKs; blind-index search key
§164.312(b) — Audit controls	Append-only, hash-chained provenance ledger across ~90 PHI touchpoints
§164.312(c) — Integrity	Hash chaining detects alteration; DB triggers reject update/delete/truncate of the ledger
§164.312(d) — Person/entity authentication	better-auth sessions; mandatory TOTP 2FA + backup codes; server-side session revocation
§164.312(e) — Transmission security	TLS 1.2/1.3 + HSTS; TLS-only object storage; shared-secret auth on internal service calls
§164.316(b)(2) — Documentation retention (6 years)	Weekly audit-retention backup plan (~6 years) of the ledger-bearing data volume

B APPENDIX B

SOC 2 Trust Services Criteria mapping

A summary mapping to the 2017 Trust Services Criteria (with 2022 revised points of focus). This reflects control design; no independent examination has been performed to date.

CRITERIA	HOW TACTIS ADDRESSES IT
CC1 — Control environment	Structured security remediation program with a maintained status of record; least-privilege by default
CC2 / CC3 — Communication & risk assessment	Threat-model & audit roadmap; formal HIPAA risk analysis produced at go-live (see status block)
CC5 / CC6.1 — Logical access, encryption	Field-level AES-256-GCM, per-facility keys, KMS CMK; TLS in transit
CC6.1 / CC6.3 — Access provisioning & authorization	Invitation-only accounts; capability matrix + permission guards; per-request facility scoping
CC6.6 — Boundary protection	Fail-closed egress policy; PHI only to BAA-covered zones; SSH limited to a /32
CC6.7 — Transmission & movement of data	TLS-only; internal service calls authenticated by shared secret; content-free egress receipts
CC6.8 — Malicious/unauthorized software	Immutable, scanned container images; hashed, pinned dependency lockfile
CC7.1 / CC7.2 — Monitoring & anomaly detection	Centralized logs, metric filters, and alarms (5xx, ledger-failure, host, disk, memory) to SNS
CC7.3 / CC7.4 — Incident response	Ledger makes disclosure scope answerable; DR runbook covers declaration & recovery
CC8.1 — Change management	Infrastructure as code with remote encrypted state; immutable image tags + rollback history
A1.2 / A1.3 — Availability: backup & recovery testing	AWS Backup operational + audit plans; documented restore-drill run at least quarterly

— REQUESTING MORE

More detail, under NDA

This overview is intentionally a summary. Source-level walkthroughs, the full findings register, the disaster-recovery plan and restore-drill records, the HIPAA control matrix, and the subprocessor register with executed agreements are available to prospective customers and their security reviewers under NDA. To request them, or to arrange a technical security review, contact your Tactis point of contact.

This is a public overview of the Tactis platform's security posture. This document describes the Tactis platform as of 26 July 2026. Statements about program milestones (BAA execution, legal-entity formation, SOC 2 examination, and zero-data-retention enablement) are current as of that date and are stated in the Program status block; they are updated on each issue of this document.