Security & PHI Posture Summary

A summary of the security controls and deployment posture for healthcare / PHI use, rated by what is **enforced in code**. A full, control-by-control review is available on request.

Overview

MessageFoundry is an open-source (AGPL) HL7 v2 integration engine, written in Python. It runs as a headless engine that owns a message store (SQLite by default, or an enterprise RDBMS) and routes and transforms HL7 between connections; a separate admin console operates it over a localhost API. This summary describes the **Phase 1 deployment posture**: a single host, the API bound to `127.0.0.1`, and authentication enabled. The trust boundary is the localhost API and the host itself.

Built & enforced today

- **Authentication, fail-closed** — local accounts with argon2id password hashing, account lockout, and constant-time login, plus Active Directory over **LDAPS** (certificate verification on by default). The engine refuses to serve unauthenticated requests off loopback and denies access when no auth is configured. *(Kerberos / Windows SSO is experimental; MFA is on the roadmap.)*
- **Authorization (RBAC)** — deny-by-default, fixed built-in roles, enforced on every engine and admin route, with **per-channel data-scope confinement** enforced at the store-query level.
- **Session management** — opaque, high-entropy server-side tokens (stored only as a hash), idle and absolute timeouts enforced per request, and immediate revocation on password change, disable, or role/scope change.
- **Audit & tamper-evidence** — a user-attributed, hash-chained audit log covering authentication, authorization denials, administrative changes, and operator PHI-access / replay actions, with a verification command. *(Tamper-evidence, not prevention — integrity also rests on host/file access controls.)*
- **Encryption at rest** — AES-256-GCM authenticated encryption for message bodies when a store-encryption key is set; pair with full-disk/volume encryption for complete at-rest coverage.
- **Durable reliability** — the message store *is* the queue (transactional inbox/outbox); every received message is recorded with a disposition **before it is acknowledged**, so nothing is silently dropped, with retries, dead-lettering, and replay.
- **PHI-safe AI assistance** — the optional in-editor assistant is code-only by construction (it sends graph names and editor code, never message bodies), under a central, environment-aware policy and a role-based permission.

Partial / on the roadmap — disclosed, not hidden

A few controls a PHI deployment will want are not built yet. We state them plainly; until they land, run the engine on localhost or behind your own TLS-terminating proxy.

- **TLS for the API and MLLP** — not yet built; Phase 1 is a loopback posture. (Active Directory LDAPS and the SQL Server backend already use TLS.)
- **Multi-factor authentication** — roadmap; today, layer MFA via an external AD / SSO front.
- **At-rest coverage** — the body cipher is optional and key-based; full coverage of all stored fields relies on volume encryption.
- **Data lifecycle** — retention / purge enforcement, centralized log redaction, and a de-identification framework are on the roadmap.
- **Outbound destination allow-listing** — roadmap.

Capabilities at a glance

Domain	Maturity	Summary
Authentication	BUILT	argon2id local + LDAPS Active Directory; fail-closed; lockout (local accounts); MFA roadmap; Kerberos experimental.
Authorization & RBAC	BUILT	Deny-by-default, fixed roles, enforced per route; per-channel data-scope confinement.
Session management	BUILT	Opaque server-side tokens, idle + absolute timeouts, immediate revocation.
Audit & tamper-evidence	BUILT	Hash-chained, user-attributed audit log with a verification command (evidence, not prevention).
Encryption at rest	PARTIAL	AES-256-GCM body cipher (optional, key-based); volume encryption for full coverage.
Data in transit & network	PARTIAL	Loopback default + DoS caps + LDAPS / DB-TLS; API and MLLP TLS on the roadmap.
PHI handling, logging & retention	PARTIAL	Strong access audit; centralized log redaction and retention enforcement on the roadmap.
Supply-chain & secure SDLC	PARTIAL	Pinned + hashed lockfile, Dependabot, disclosure policy; deeper SAST / secret-scanning on the roadmap.
AI assistance governance	PARTIAL	Code-only by construction, central clamped policy, RBAC permission; managed BAA provider on the roadmap.

Compliance alignment

Met = enforced in code for the Phase-1 posture · **Partial** = built but incomplete or contingent · **Gap** = not built.

HIPAA Security Rule — 45 CFR §164.312

Safeguard	Status	Notes
Access control / least privilege	**Met**	Deny-by-default RBAC + per-channel scope; per-request identity. (Coarse-grained; no field-level controls.)
Unique user identification	**Met**	Distinct local / AD accounts; a non-null actor on every audit row.
Automatic logoff (addressable)	**Met**	Idle (30 min) + absolute (12 h) timeouts, fail-closed on clock step-back.
Audit controls	**Met** (roadmap items)	Hash-chained, user-attributed audit of auth / authorization / admin / PHI-access.
Integrity	**Partial**	GCM authentication tag on encrypted bodies + audit hash chain = tamper-evidence. WORM / external anchoring is roadmap.
Person/entity authentication	**Met** (single-factor)	argon2id local + LDAPS AD. MFA on the roadmap.
Transmission security	**Gap** (Phase 1)	No TLS on the API / MLLP yet; mitigated by the loopback-default posture and network isolation.

Selected NIST SP 800-53 controls

Control	Status	Notes
AC-2/3/6 Account mgmt, access enforcement, least privilege	**Met** / **Partial**	Audited lifecycle, store-level scope; fixed roles, no field-level controls.
AC-7 Unsuccessful logon attempts	**Met** (local)	Per-account lockout + rate limiting; AD lockout via the directory's policy.
AC-12 Session termination	**Met**	Idle / absolute timeouts + immediate revocation.
AU-3/9/12 Audit content, protection, generation	**Met** / **Partial**	Rich content, centralized generation; off-host / WORM anchoring is roadmap.
AU-11 Audit retention	**Gap**	Retention enforcement not built.
SC-5 DoS protection	**Met**	MLLP / File / HTTP / WS caps enforced.
SC-7 Boundary protection	**Partial**	Loopback default + auth-off refusal; outbound allow-listing is roadmap.
SC-8/13 Transmission & crypto	**Gap** / **Partial**	No TLS on MLLP / API; approved primitives (AES-256-GCM, argon2id) where applied.

Control	Status	Notes
SC-28 Protection at rest	**Partial**	App-layer AEAD for bodies (optional); full coverage via volume encryption.

> **Regulatory direction (2025 HIPAA Security Rule NPRM).** The proposed rule moves several long-standing *addressable* specifications toward **required** — notably encryption of ePHI at rest and in transit, and multi-factor authentication. Treat TLS, MFA, and full at-rest encryption as near-term requirements for a production PHI deployment, not optional hardening.

Deployment requirements (operator preconditions)

These are the controls the software **cannot enforce** on its own, on which the compliance posture depends. A deployment that skips any of them is materially less secure.

- Keep the API and every MLLP listener on `127.0.0.1` — never bind a network interface without an external TLS-terminating proxy in front.
- Enable full-disk / volume encryption (BitLocker / LUKS) — the at-rest protection for everything outside the body cipher.
- Set the store-encryption key as an environment variable, and **back up / escrow it** (it is non-rotatable; a lost key strands encrypted bodies). Verify the "encrypted N rows" startup log line.
- Rotate the bootstrap admin password on first login and remove the bootstrap secret file.
- Run the engine under a least-privilege service account; lock down the store, config, and file-connector directories with OS ACLs.
- Treat all backups, exports, and the database file as PHI.
- Do not raise the service to DEBUG in production (no centralized log redaction yet).
- For Active Directory: keep certificate verification on, rely on the directory's lockout policy, and revoke sessions explicitly on offboarding.
- Schedule the audit-chain verification out-of-band and alert on failure; store the result off-host.