Loggie — Capability Brief

Cryptographic Integrity Infrastructure for Defense and Government Systems

Document Classification: UNCLASSIFIED

Brief Version: 2.0.0 (Executive)

Platform Version: 1.4.0

Date: 2026-02-12

Repository Commit: `9304063`

A companion Technical Deep-Dive document is available with full cryptographic primitive tables, source file references, integration footprint, and API surface detail.

Table of Contents

A. Executive Summary

Loggie is a cryptographic integrity and verification infrastructure for document provenance, authenticated messaging, and verifiable identity management. It is deployed as a software layer beneath existing operational systems — it does not replace databases, SIEMs, or records management platforms. It adds an independent, verifiable integrity chain to the records they produce.

Core capability: Loggie enables organizations to create, seal, verify, and audit digital records with cryptographic guarantees of authenticity and integrity — without exposing sensitive content to any external system.

Key properties:

- **No sensitive data egress.** Plaintext content, private keys, and personally identifiable information never leave the customer-controlled environment. Only cryptographic commitments (hashes, signatures, encrypted ciphertext) cross the trust boundary.
- **Post-quantum readiness.** NIST-standardized post-quantum algorithms (ML-KEM-768, ML-DSA-65) alongside classical primitives (X25519, ECDSA), providing hybrid security against both classical and quantum adversaries.
- **Deployment flexibility.** Operates identically across connected, on-premises, and fully air-gapped environments. All deployment modes produce identical integrity guarantees.
- **Verifiable integrity chain.** Documents and messages receive cryptographic seals composed of content hashes, Merkle tree commitments, and digital signatures. These seals support independent, third-party verification without access to the original plaintext.
- **Non-repudiation.** Every sealed record carries a Dilithium (ML-DSA-65) digital signature bound to its creator's cryptographic identity. This provides demonstrable authorship that survives independent audit, litigation, and inspector general review.

Loggie comprises 35+ packages across five architectural layers. It is designed for integration into systems that require strong data provenance guarantees — including CMMC-scoped environments, supply chain verification workflows, and classified enclave operations.

B. System Overview

What Loggie Is

Loggie is a **cryptographic integrity and verification infrastructure** that provides:

1. **Sealed Envelopes** — End-to-end encrypted message containers with post-quantum key encapsulation and authenticated encryption. Each envelope carries a digital signature for sender authentication.
2. **Identity Management** — Self-sovereign identity creation with cryptographic key generation (classical and post-quantum), key rotation, and a keyring architecture that isolates secret material from application code.
3. **Document Integrity** — Content hashing, metadata sanitization, and Merkle tree computation for batched integrity proofs without modification of the original content.
4. **Notarization Anchoring** — Optional submission of Merkle roots to smart contracts for immutable, timestamped proof of existence.

What Loggie Is Not

- Loggie is **not a blockchain**. It uses existing networks for optional notarization.
- Loggie is **not a system of record**. It sits beneath operational systems, adding an independent integrity layer.
- Loggie does **not transmit plaintext**. All egress data is encrypted or is a cryptographic commitment.
- Loggie does **not require network connectivity** for core cryptographic operations.

Defense-Relevant Use Cases

Operational Domain	Capability	Value
CMMC evidence anchoring	Cryptographic seals on policy documents, access logs, incident reports	Tamper-evident audit trail that survives third-party assessment
Supply chain provenance	Hash-based integrity proofs on firmware, BOMs, supplier attestations	Independent verification without exposing source material
Autonomous mission logs	Sealed envelopes on UAS telemetry, AI decision events, flight records	Non-repudiable mission record with post-quantum signatures
Incident reconstruction	Merkle tree commitments over forensic evidence chains	Litigation-defensible, timestamped proof of evidence integrity
Air-gapped integrity	Full cryptographic operations in SCIFs and classified enclaves	No network dependency; sneakernet-exportable commitment hashes

C. Architecture

Loggie is organized into five distinct architectural layers. Each layer has clearly defined responsibilities and trust assumptions.

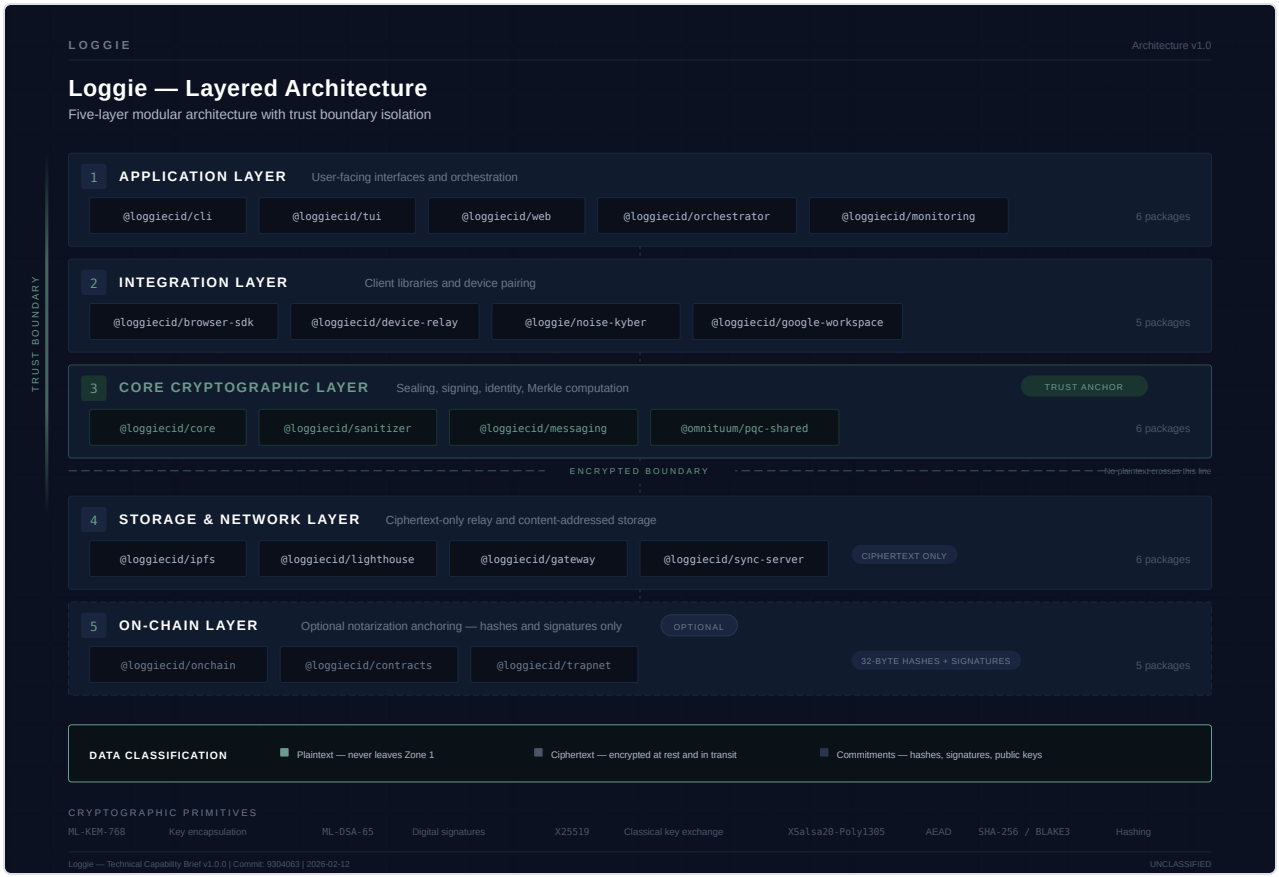

Figure 1: Loggie layered architecture with trust boundary indication.

Layer 1 — Application. Operational interfaces: CLI, terminal UI, web interface, orchestrator for automated workflows.

Layer 2 — Integration. Libraries for embedding Loggie capabilities into external applications. A membrane pattern isolates secret keys behind a keyring interface. Post-quantum secure device pairing via Noise-Kyber.

Layer 3 — Core Cryptographic Engine. Sealing/unsealing, key generation, digital signatures, identity hashing, envelope construction, metadata stripping, and Merkle tree computation. Backed by the Omnituum PQC shared library.

Layer 4 — Storage and Network. Integration with IPFS, Filecoin, Lighthouse, self-hosted gateways, and sync servers. All data in this layer is encrypted — storage nodes handle ciphertext only.

Layer 5 — On-Chain (Optional). Notarization anchoring, identity registration, inbox management. Only 32-byte hashes and signatures go on-chain. Entirely optional; disabled in air-gapped deployments.

D. Deployment Models

Core cryptographic operations — sealing, signing, hashing, and verification — function identically across all three models with no feature degradation.

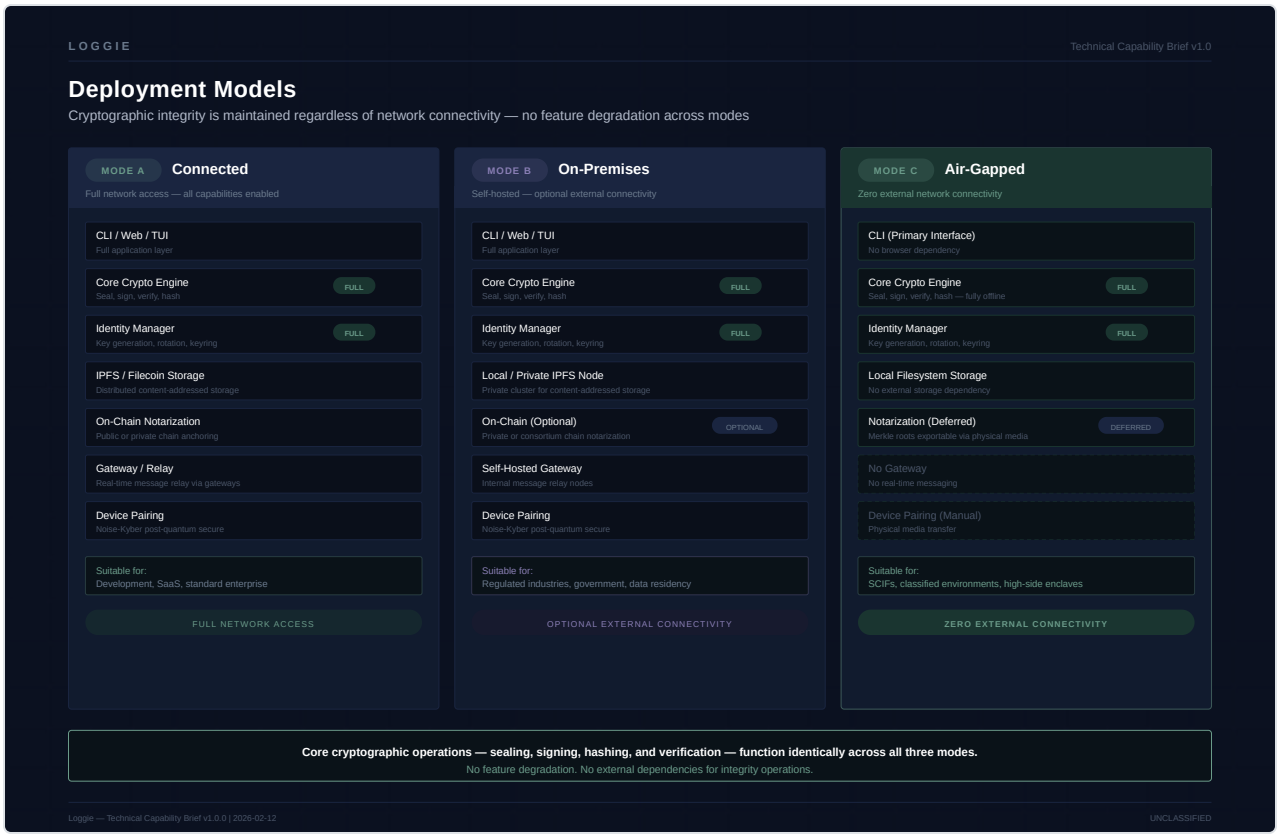

Figure 2: Deployment models. Cryptographic integrity is maintained regardless of network connectivity.

Mode A: Connected

Full network access. Suitable for development, SaaS, and standard enterprise use.

Mode B: On-Premises

Self-hosted infrastructure with optional external connectivity. Suitable for regulated industries, government systems, and environments with data residency requirements.

Mode C: Air-Gapped

Zero external network connectivity. Suitable for SCIFs, classified environments, and high-side enclaves. CLI-primary interface, local filesystem storage, all cryptographic operations fully offline. Merkle roots exportable via physical media for external verification.

Deployment Decision Matrix

Requirement	Connected	On-Premises	Air-Gapped
Cryptographic sealing	Full	Full	Full
Digital signatures	Full	Full	Full
Integrity verification	Full	Full	Full
Real-time messaging	Yes	Self-hosted	No
On-chain notarization	Public chain	Private chain	Deferred
Content-addressed storage	IPFS/Filecoin	Private IPFS	Local FS
Device pairing	Noise-Kyber	Noise-Kyber	Manual

E. Data Flow and Trust Boundaries

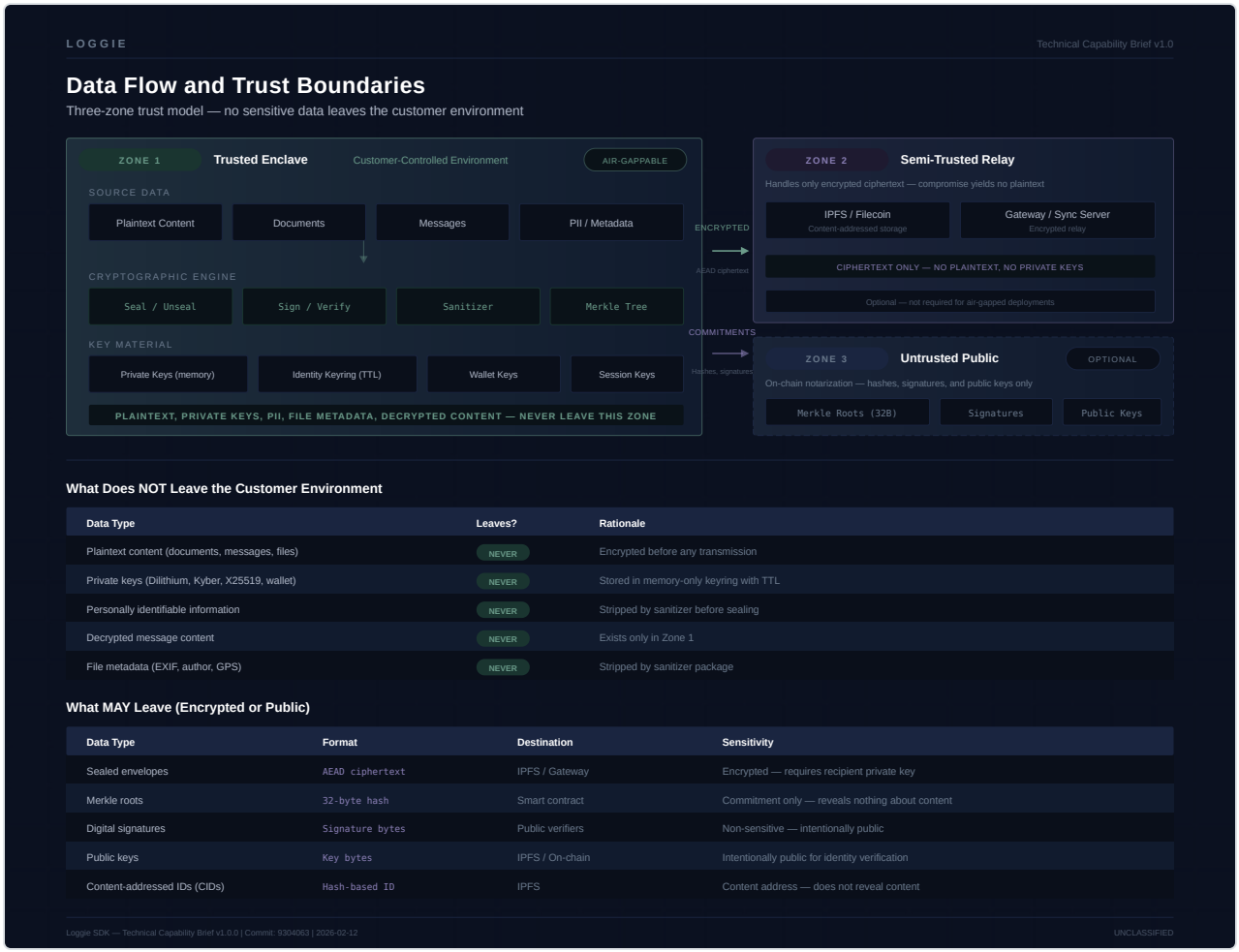

Figure 3: Data flow classification. No sensitive data leaves the customer environment.

Three-Zone Trust Model

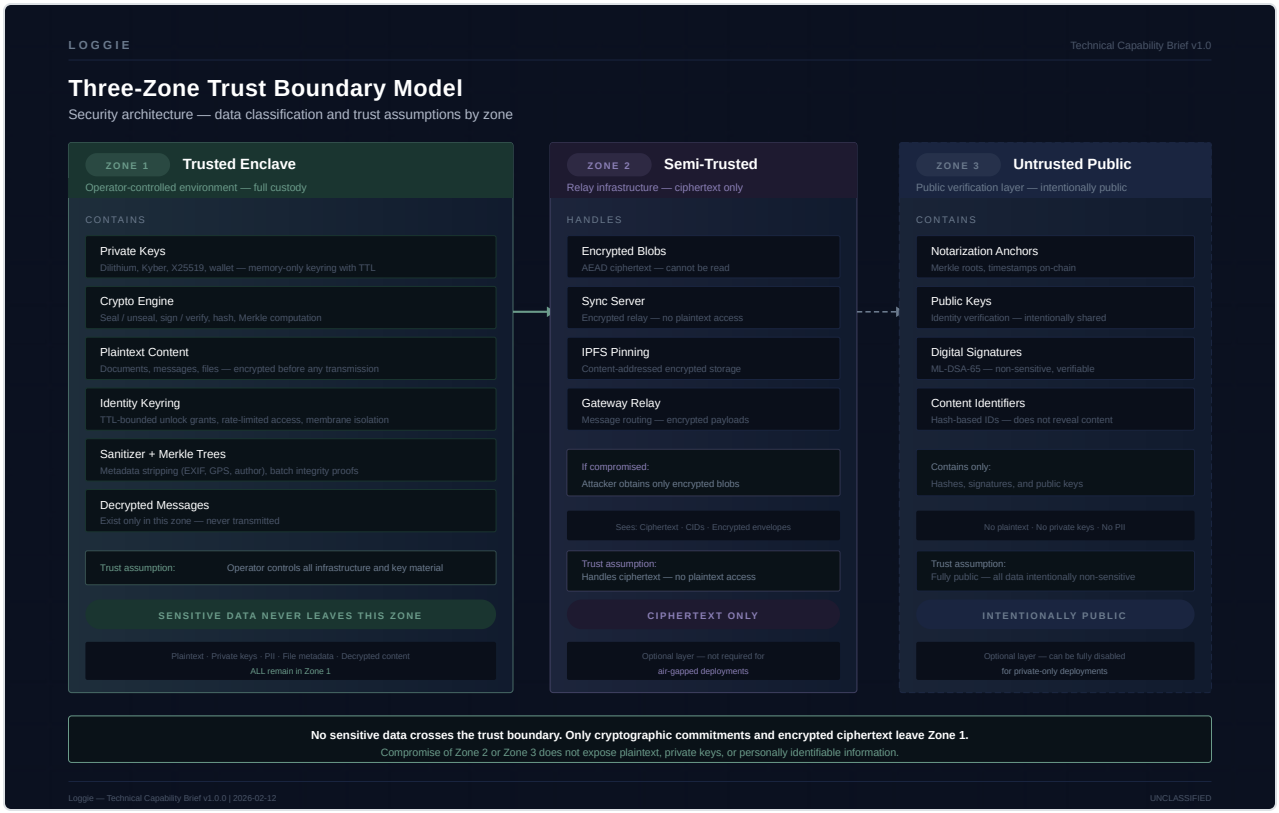

Figure 4: Three-zone trust architecture.

Zone 1 — Trusted Enclave (Customer-Controlled). Contains all sensitive material: private keys, plaintext content, decrypted messages, and the cryptographic engine. This zone never transmits sensitive data.

Zone 2 — Semi-Trusted Relay. Optional relay infrastructure that handles only encrypted ciphertext. Compromise of Zone 2 does not expose plaintext or private keys.

Zone 3 — Untrusted Public. Public verification layer containing only hashes, signatures, and public keys. All data in Zone 3 is intentionally public and non-sensitive.

What Does NOT Leave the Customer Environment

Data Type	Leaves?	Rationale
Plaintext content (documents, messages, files)	**NEVER**	Encrypted before any transmission
Private keys (Dilithium, Kyber, X25519, wallet)	**NEVER**	Stored in memory-only keyring with TTL
Personally identifiable information	**NEVER**	Stripped by sanitizer before sealing
Decrypted message content	**NEVER**	Exists only in Zone 1
File metadata (EXIF, author, GPS, etc.)	**NEVER**	Stripped by sanitizer

What MAY Leave (Encrypted or Public)

Data Type	Format	Destination	Sensitivity
Sealed envelopes	AEAD ciphertext	IPFS / Gateway	Encrypted; cannot be read without recipient's private key
Merkle roots	32-byte hash	Smart contract	Commitment only; reveals nothing about content
Digital signatures	Signature bytes	Public verifiers	Non-sensitive; intentionally public
Public keys	Key bytes	IPFS / On-chain	Intentionally public for identity verification
Content-addressed IDs (CIDs)	Hash-based ID	IPFS	Content address; does not reveal content

F. Security Posture Summary

Threat Model Considerations

- **Harvest-now-decrypt-later:** Mitigated by hybrid post-quantum encryption (ML-KEM-768 + X25519). Resists decryption by future quantum computers.
- **Compromised relay/storage:** Gateway and IPFS nodes see only encrypted blobs. Compromise yields no plaintext.
- **Key exfiltration:** Private keys exist only in memory-only keyring with automatic TTL-based cleanup (5-minute default). Never persisted to disk.
- **Identity impersonation:** Identity documents are cryptographically signed with ML-DSA-65 (post-quantum) signatures that can be independently validated.
- **Metadata leakage:** The sanitizer strips EXIF, author, GPS, and other metadata fields before content leaves the environment.
- **Man-in-the-middle:** Sealed envelopes include authenticated encryption (XSalsa20-Poly1305 AEAD). Tampering breaks the authentication tag.

Key Custody Model

Key Type	Algorithm	Storage Location	Persistence	Access Pattern
Wallet key	ECDSA (secp256k1)	Hardware wallet or encrypted store	Persistent	Transaction signing only
Dilithium signing key	ML-DSA-65	Encrypted local store	Persistent (encrypted)	Message signing
Kyber decryption key	ML-KEM-768	Encrypted local store	Persistent (encrypted)	Message decryption
X25519 key	Curve25519	Encrypted local store	Persistent (encrypted)	Classical key agreement
Session keys	XSalsa20-Poly1305	Memory only	Ephemeral	Per-message encryption

Cryptographic Standards Summary

Function	Algorithm	Standard
Key encapsulation	ML-KEM-768 (Kyber)	NIST FIPS 203
Digital signatures	ML-DSA-65 (Dilithium)	NIST FIPS 204
Key agreement	X25519	RFC 7748
Authenticated encryption	XSalsa20-Poly1305	NaCl secretbox
Backup encryption	AES-256-GCM	NIST SP 800-38D
Content hashing	SHA-256, BLAKE3	FIPS 180-4, BLAKE3 spec
Key derivation	HKDF-SHA-256	RFC 5869
Device pairing	Noise XX + ML-KEM	Noise Framework

Full cryptographic primitive tables with implementation file paths are available in the Technical Deep-Dive companion document.

G. Compliance Mapping

> **Disclaimer:** Loggie does not claim compliance certification. The following mapping identifies capabilities that align with specific control families. Actual compliance requires organizational policies, procedures, and independent assessment beyond the scope of this software.

NIST SP 800-53 Rev. 5 — Control Family Mapping

Control	Description	Loggie Capability	Status
SC-8	Transmission Confidentiality	Sealed envelopes with hybrid PQ encryption	Implemented
SC-12	Cryptographic Key Establishment	ML-KEM-768 + X25519 hybrid key exchange	Implemented
SC-13	Cryptographic Protection	NIST-approved algorithms (FIPS 203, 204)	Implemented
SC-28	Protection of Information at Rest	Encrypted local storage, memory-only keyring	Implemented
SI-7	Software, Firmware, and Information Integrity	Merkle tree proofs, content hashing, sanitization	Implemented
IA-5	Authenticator Management	Key generation, rotation, TTL-based expiry	Implemented
AU-10	Non-Repudiation	Dilithium digital signatures on all sealed envelopes	Implemented

CMMC Level 2 Alignment

CMMC Practice	Loggie Relevance
AC.L2-3.1.1 Authorized access	Keyring membrane enforces access control to cryptographic material
SC.L2-3.13.1 Boundary protection	Three-zone trust model; no sensitive data crosses boundary
SC.L2-3.13.8 CUI encryption	Hybrid PQ encryption for all data in transit
SC.L2-3.13.11 CUI confidentiality	XSalsa20-Poly1305 AEAD; content encrypted at rest
SI.L2-3.14.1 Flaw remediation	Anti-regression CI gates; golden test vectors

Areas Requiring Organizational Completion

The following control areas are not addressable by software alone and require organizational policies and procedures:

- **TODO:** Physical security controls (PE family) — Require facility-level implementation
- **TODO:** Personnel security (PS family) — Require HR policies
- **TODO:** Incident response (IR family) — Require organizational IR plans; Loggie provides audit logging hooks
- **TODO:** Risk assessment (RA family) — Require organizational risk framework
- **TODO:** Security assessment (CA family) — Require third-party assessment of deployed system

- **TODO:** Configuration management (CM family) — Loggie provides deterministic builds; organizational CM policy required

---

Companion Documents

Document	Purpose
Technical Capability Brief (16-page)	Full cryptographic primitives, source file references, integration footprint, API surface
Defense Relevance Brief	Operational use case articulation across four domains
Security FAQ	Conversational security assurance addressing contractor concerns
CMMC Pilot SOW	90-day scoped proof-of-concept deployment
Architecture Diagram	Two-zone deployment architecture (enclave + optional notarization)

End of Executive Capability Brief