
Making yourself audit-ready with Verae DataCubes

Verae

Sep 16, 2026

CONTENTS

How to **prepare an organization** for HIPAA-aligned, SOC 2, or ISO 27001 work using Verae DataCubes on **Peergos** (encrypted cryptree + hashed IPFS) and HPKE on an **untrusted NATS** broker.

This document is not a HIPAA, SOC 2, or ISO certificate. Peergos's public pentests are **not** your Type II or ISO registrar certificate. They are **component security evaluations** you can attach.

PEERGOS VERIFICATION (WHAT WAS ACTUALLY AUDITED)

This chapter records **sourced** facts. It does **not** turn Peergos, Verae, or DataCubes into a HIPAA, SOC 2, or ISO 27001 certificate.

1.1 Claim to check

Data at rest is stored in an encrypted append-only Peergos filesystem; backup/restore is managed via a distributed, hashed, encrypted Peergos IPFS system, designed and evaluated in the EU as an audited secure filesystem.

1.2 Independent security audits (Europe)

2019 — Cure53 (Berlin, Germany). Pentest, source-code audit, **crypto/design review** (May–June 2019). No fundamental architectural or cryptographic problems; issues fixed. Cure53: the platform **passed this evaluation**.

2024 — Radically Open Security B.V. (Amsterdam, Netherlands). Crystal-box pentest + code audit of Peergos web-ui (Sep–Nov 2024). **0** extreme / high / elevated; **2** moderate, **6** low; Peergos states all fixed; no data exposure or integrity compromise (mostly UI crashes).

Both firms are **EU-based**. Both full reports are **public**. That is a **security evaluation of the Peergos protocol/implementation**, not a HIPAA/SOC 2/ISO management-system certificate.

1.3 EU funding / NLnet

- Peergos received funding from the **European Union Horizon 2020** programme under **NGI-POINTER**, grant **871528** (stated on the 2024 audit post).
- The 2024 audit was supported by **NLnet** (Netherlands).

1.4 Hosted instance location

Peergos's hosted privacy notice states peergos.net uses servers in **Germany**. **Self-hosted ns1 / customer cryptree is a different processing location** and must be documented in *your* RoPA / BAA pack.

1.5 Encrypted filesystem + hashed IPFS (design)

- **Crypttree**: tree of symmetric keys; fine-grained read/write caps; sibling names hidden; writes **signed**.
- **Chunks**: up to 5 MiB, padded, independently encrypted (Salsa20-Poly1305); random 256-bit keys (not convergent).
- **IPFS**: content-addressed blocks. Peergos **verifies the hash** of every block read from or written to IPFS. Clients verify hashes and signatures from a Peergos server.
- **CHAMP**: encrypted crypttree nodes under **random 32-byte labels**.
- Sharing: hybrid **X25519 + ML-KEM-1024**.

“Append-only” in Verae DataCubes is the **cube chain JSONL** (application integrity). Peergos uses signed updates and immutable blocks. Do not conflate the two layers in an audit interview.

1.6 Not verified / do not claim

- “Peergos is HIPAA certified” — **false**. HIPAA is not a product certificate.
- “Peergos is SOC 2 / ISO 27001 certified” — **not shown**. Published work is pentest/code audit, not Type II or ISO registrar.
- “EU certified Peergos as a government-standard filesystem” — **overstated**. What exists: EU grant funding, NLNet-funded ROS pentest, Cure53 Berlin design+code audit.
- “Hosts with disk/backup never need a BAA/DPA” — **legal conclusion, not a technical fact**. See *BAAs and DPAs when ciphertext has no host keys*.

ARCHITECTURE (AUDIT INTERVIEW)

```
Endpoint (keys stay here / HSM)
  |
  | HPKE content (NPE suite)
  | routing: dest + subject in the clear
  |
Untrusted NATS (cannot read bodies)
  |
  |
Verae DataCube chain (append-only hashes)
  |
  | written through Peergos client
  |
Peergos crypttree (encrypted names, sizes, graph)
  |
  | chunks → CID / hash
  |
IPFS (distributed, hash-verified ciphertext)
```

- **At rest:** Peergos crypttree + IPFS. Hosts with disk/backup see **opaque hashed ciphertext**, not PHI, if they lack keys.
- **In transit (NATS):** HPKE-Base to directory public keys. Broker is honest-but-curious: destinations yes, bodies no.
- **Integrity:** cube JSONL chain + dual hash + IPFS CID check on restore (re-fetch blocks, re-verify hashes — not a plaintext tape).

NATS remains an untrusted router. Production E2E is NPE/HPKE. Passthrough means destination in the clear; the body is ciphertext.

See companion [peergos-for-compliance.pdf](#) (same folder) for controls, custody, and variables chapters.

[peergos-for-compliance.pdf](#)

BAAS AND DPAS WHEN CIPHERTEXT HAS NO HOST KEYS

This is **guidance for an evidence pack**, not legal advice.

3.1 The operator statement

Data at rest lives in the **encrypted Peergos crypttree**. Restore uses **distributed, hash-verified, encrypted IPFS blocks**. Hosts and backup media see **opaque blobs** (and random CHAMP labels), not filenames, not PHI, not social graph — if keys never leave the client / customer HSM.

That is **true of the Peergos design** (see *Peergos verification (what was actually audited)*). It **changes the BAA/DPA conversation**. It does **not** auto-delete the need for contracts.

3.2 HIPAA (US) — Business Associate

A Business Associate is a person who **creates, receives, maintains, or transmits ePHI** for a covered entity.

Technical argument you can make (and should document):

1. ePHI is encrypted **before** it leaves the endpoint (Peergos client / DataCube writer).
2. The storage node, IPFS, disk, and backup hold **ciphertext + hashes**.
3. Private keys **do not** reside on the storage host (customer HSM / client). Hash verification detects tampering.
4. Therefore the host **cannot maintain ePHI in intelligible form**.

HHS distinguishes **conduits** and parties that never have access to ePHI. Many programs still **sign a BAA** when a vendor touches infrastructure, because misconfiguration, support exports, or logs could later expose plaintext. OCR looks at **your** risk analysis.

Practical pack: risk analysis memo citing crypttree + IPFS hash checks + key custody; BAA **or** written determination that the vendor is **not** a BA because they cannot decrypt; list every party with disk/backup (ns1, IPFS cluster, offsite replica).

3.3 GDPR (EU/UK) — processor vs technical measure

Encrypted data can still be **personal data** if it is reasonably attributable (usernames, IPs, invoice identity). GDPR **Art. 32** lists encryption as a security measure.

Peergos is designed so the **server is an adversary** for content and most metadata. EU-based **independent pentests** (Cure53 DE, ROS NL) evaluated that design. Processing location for **self-host** is *your* ns1 / customer region, not peergos.net's Germany hosting, unless you use peergos.net.

You may still need a **DPA (Art. 28)** with anyone who **processes** personal data (usernames, logs, billing). Ciphertext-only storage **narrows** the DPA's technical annex.

3.4 NATS operators

NATS is an **untrusted broker**. Content is HPKE; destinations and subjects are in the clear. A NATS operator **cannot** read bodies without endpoint private keys; **can** see routing metadata; **can** drop/delay copy **ciphertext**. Treat like a transit provider: DPA/BAA depends on whether routing metadata is personal data in your jurisdiction.

3.5 What “making yourself audit-ready” means

You are **not** buying Peergos's Cure53/ROS reports as *your* SOC 2. You are **reusing** those public evaluations as **vendor/component assurance**, then adding your scope, RoPA, key-custody, inspect k-of-n, admin-history, instance evidence, and contracts that match who can actually see plaintext.

AUDIT-READY CHECKLIST

Use this as a working list. Check items only when **evidence exists** (screenshot, log export, signed policy, ticket). This list is not a certificate.

4.1 A. Scope and data map

- Named legal entity and systems in scope (console, Drive, NATS, IPFS, ns1)
- Data-flow diagram: endpoint → HPKE/NATS → crypttree/IPFS
- What is PHI / personal data vs ciphertext vs routing metadata
- Peergos verification chapter attached (*Peergos verification (what was actually audited)*)

4.2 B. Keys and encryption

- Customer holds Peergos/Drive keys; not on storage host
- NPE/HPKE endpoint keys in directory; private 0600; no xor content
- Signed Ed25519 config; unsigned rejected; admin-history prev+new+diff
- HSM or documented lab-to-HSM path

4.3 C. Access

- Console PFC_REQUIRE_AUTH=1, TOTP; Drive login separate
- Inspect k-of-n; author is not an officer; log-before-reveal
- Joiner / mover / leaver for console users
- Google SSO does **not** unwrap cubes

4.4 D. Integrity and backup

- Cube chain append-only JSONL; dual hash
- Hash registry: first SHA-256 and receipt win
- Peergos/IPFS: hash-verified encrypted blocks (restore = re-fetch CIDs)

- Restore test: recover a cube from content-addressed ciphertext without plaintext backup tapes

4.5 E. Logging

- Inspect audit events
- SM summaries: codes + lookup_id only (no bodies)
- Admin-history cube for config

4.6 F. Contracts (counsel)

- Written BA / not-a-BA determination for disk, VM, backup, IPFS
- DPA Art. 28 where usernames/logs/IPs are processed
- NATS operator: metadata vs content in the annex
- Peergos component assurance: attach Cure53 2019 + ROS 2024 PDFs (public; not your org's ISO/SOC report)

4.7 G. Independent audit of *you*

- SOC 2 Type I/II engagement **or** ISO 27001 registrar **or** HIPAA risk analysis + policies (pick the program you actually need)
- Evidence window (Type II / surveillance) if applicable

HOW TO USE THIS PACK

1. Read *Peergos verification (what was actually audited)* so you do not over-claim Peergos audits.
2. Fill *Audit-ready checklist* with **your** instance evidence (ns1, keys, users).
3. Give *BAAs and DPAs when ciphertext has no host keys* to counsel with the data-flow from *Architecture (audit interview)*.
4. Point auditors at live technical surfaces (do not give them private keys).
5. Attach the two **public** Peergos pentest PDFs from the Peergos audits/ tree as **vendor security evaluations**, labeled “not our SOC 2 / ISO certificate”.

Companion system PDF (same folder):

[peergos-for-compliance.pdf](#)

WHAT WE VERIFIED ABOUT PEERGOS (EU)

See the full sourced table in *Peergos verification (what was actually audited)*.

- Encrypted client-side filesystem (cryptree); keys not on the storage server — **yes** (Peergos book + Cure53 design review).
- IPFS blocks content-addressed; Peergos verifies hashes — **yes**.
- Independent **EU** security audits, reports published — **yes, two**: Cure53 Berlin (2019); Radically Open Security Amsterdam (2024).
- “Peergos is HIPAA/SOC 2/ISO certified” — **no**. Those audits are pentest/code/design reviews, not management-system certificates.

Peergos **was designed as a trust-minimized encrypted filesystem, evaluated in Europe** by two specialist firms, with **public reports**. That supports the **at-rest / backup** story. It does **not** finish *your* audit.

Companion PDFs in the **same directory** as this file (relative PDF links, not Markdown):

- [peergos-for-compliance.pdf](#)
- [nats-service-endpoints.pdf](#)
- [secure-messaging.pdf](#)