
Peergos for Compliance

Verae

Sep 16, 2026

CONTENTS

1	Remaining items	3
1.1	Done in the lab (ns1)	3
1.2	Still later	3
2	Architecture	5
3	Console, workspace, and catalog	7
3.1	Two identity planes	7
3.2	Workspace	7
3.3	Drag and drop	7
3.4	Hash registry and stamps	7
4	PFC Inspect (Pattern A)	9
4.1	Host path	9
5	NATS addresses	11
5.1	Hub (already on ns1)	11
5.2	PFC leaf clients	11
5.3	Subjects	11
6	Host ns1 70.88.205.138	13
6.1	Existing	13
6.2	Public Caddy maps (Let's Encrypt)	13
7	Go services	15
8	Configuration system	17
9	Operations	19
10	Walkthroughs	21
10.1	Console login (separate from Peergos Drive)	21
10.2	Cubes, catalog, copy+stamp	23
10.3	Users	25
10.4	Inspect (k-of-n)	26
10.5	Audit	27
10.6	Config SPA (unsigned preview only)	28

10.7	Secure message success vs failure	28
11	Technical controls (not a certificate)	29
11.1	Access control	29
11.2	Cryptography	29
11.3	Audit and logging	29
11.4	Key custody	30
11.5	Integrity	30
12	Key custody	31
12.1	Lab (ns1)	31
12.2	Production	31
13	Variables	33
13.1	Signed wrapper payload (PFC_SIGNED_CONFIG)	33
13.2	Process environment	33

Technical control plane for Verae DataCubes on Peergos, NATS (untrusted broker), NPE (HPKE E2E), and ns1 70.88.205.138.

This is **not** a HIPAA/SOC 2/ISO certificate.

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

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

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

REMAINING ITEMS

This is **not** a HIPAA, SOC 2, or ISO certificate.

1.1 Done in the lab (ns1)

- Fail-closed profiles, append-only cube chain, Shamir inspect wrap, log-before-reveal.
- Console at <https://pfc.georgelambert.org>: session cookie, TOTP, optional Google SSO (button hidden until `GOOGLE_CLIENT_ID`).
- Open JSON cube APIs return 401 when `PFC_REQUIRE_AUTH=1`.
- Workspace cube `<user>-ws` and library catalog (Client, Author, Subject, Title).
- Drag-and-drop from the desktop and cube-to-cube; catalog prompt; SHA-256 registry never replaces the first timestamp receipt.
- Periodic leaf stamp sync (default 15s): only unregistered hashes go to `verae.ts.request` (local-dual if the central TS is down).
- Pattern A **PFC Inspect** installed for Drive user `peergos`; OrgTool bridge copies cryptree outbox to `/opt/pfc/bridge`.
- Let's Encrypt for `pfc / config.pfc / docs.pfc.georgelambert.org`.

1.2 Still later

- WebAuthn on the console.
- `PFC_REQUIRE_NPE=1` and the Go NPE sidecar (lab still uses `pfc-lab-xor / plain`).
- Customer HSM (lab uses `lab_hsm_stub`).
- Mode-A exam zip / `verae.cube.export` ISO packager.
- nsc JWT on `nats-a/b/c`; BAAs; pentest; SOC 2 Type I/II; ISO ISMS.

Do not expose NATS 4222 / 7422 or Kubo 5001 on the WAN.

ARCHITECTURE

Layers: DataCube (evidence) → Peergos crypttree (container) → NATS/NPE (messages) → Python/Go leaf services on ns1 → Caddy hostnames.

A **workspace** DataCube is the user's primary container. Other cubes are indexed with library cards. Each object identity is SHA-256 plus the original `created_ts`. Copies preserve that identity; the dest chain records copy.

Peergos HTML5 apps **never** open NATS (Pattern A outbox only).

Go services each run an **in-process NATS server** that **leaf-connects** to `nats-a 10.10.10.21:7422` (cluster verae with `nats-b/c`). The live console is the Python admin on `:18780`.

CONSOLE, WORKSPACE, AND CATALOG

Public UI: <https://pfc.georgelambert.org> (Caddy to 127.0.0.1:18780).

3.1 Two identity planes

- **Peergos Drive** – password plus TOTP/WebAuthn for cryptree. Google does not unwrap cubes.
- **PFC console** – password plus TOTP (Google OIDC optional). Inspect remains k-of-n officer shares.

3.2 Workspace

Each console user gets a primary DataCube <username>-ws (profile `internal`, kind `workspace`). Other cubes are catalogued inside it like a library.

Creating an **internal** cube requires Client, Author, Subject, Title. Those fields append to:

- `cubes/<id>/catalog/card.json`
- the workspace catalog JSONL
- the **cube** chain and the **workspace** chain

3.3 Drag and drop

- Desktop files onto the drop zone or a cube row: prompt with that cube's catalog defaults. SHA-256 is hashed in the browser first.
- An object chip from another cube: keep this cube's catalog defaults, or edit for this item only. The original SHA-256 and timestamp receipt are never rewritten.

Copy into another cube appends a `copy` row (original `created_ts`, source cube/seq, receipt). Optional file bytes are re-wrapped on the dest cube. HIPAA bytes still need an inspect session.

3.4 Hash registry and stamps

Global registry: `registry/hashes.jsonl`. First-seen `created_ts` and first receipt win.

Every chain append (object, copy, catalog, comment, memo, link) queues unregistered hashes (content SHA-256 and/or envelope SHA-256). A daemon thread (`PFC_STAMP_SYNC`, default 15 seconds) flushes them

through the NATS leaf (`verae.ts.request`). Already registered document hashes are skipped. New comments get a **new** envelope hash and a new timestamp.

Comments, memos, and chain links append on the local cube immediately; the leaf batch only timestamps hashes that have no receipt yet.

PFC INSPECT (PATTERN A)

The Drive app **PFC Inspect** is Pattern A: STORE_APP_DATA only. The iframe writes /outbox/<job>.json. It never opens NATS or HTTPS to the console.

Installed with OrgTool as Drive user peergos:

```
/.apps/pfc-inspect/  
/peergos/inspect/          (source tree)
```

4.1 Host path

1. App PUT data/outbox/*.json (subjects verae.inspect.open|join|access|close).
2. pfc-peergos-bridge (OrgTool bridge, password from @env / PEERGOS_PASSWORD) copies cryptree outbox into /opt/pfc/bridge.
3. pfc-connector publishes NATS. PFC_INTERNAL_TOKEN lets inspect run without a console session cookie. Google still cannot unwrap cubes.
4. Inbox JSON is copied back into cryptree.

Officers on a hipaa cube are share names (for example bob,carol), not the cube author and not Google accounts.

NATS ADDRESSES

5.1 Hub (already on ns1)

- Cluster: nats-a/b/c CTs 511–513, IPs 10.10.10.21–23, port 4222, routes 6222.
- **Leaf listen:** 10.10.10.21:7422 (and b/c).
- Do not bind leaf or client to 0.0.0.0 / vmbr0.

5.2 PFC leaf clients

PFC_LEAF_HUB=nats-leaf://10.10.10.21:7422

Lab Python broker (laptop): nats://127.0.0.1:14222.

5.3 Subjects

Subject catalog: see companion nats-service-endpoints.pdf (same folder as this PDF) or the NATS chapter tables below.

Core verae.ts.request|receipt|verify|batch, verae.storage.pin|hydrate|replicate,
verae.pfc.replica.{a,b}.ingest, verae.cube.append|export, verae.inspect.
open|join|access|close|revoke, verae.pfc.workspace.ensure, verae.pfc.catalog.search,
verae.pfc.cube.copy, verae.pfc.stamp.request, verae.pfc.health|cube.create|object.
put, verae.host.apps.*, verae.gossip.*.

Ledger: ledger.evt.{org}.{session}, ledger.review.*, ledger.hold.*, job.{kind}.{org}.
{id}, svc.ledger.*.

Forbidden: verae.llm.turn.> bodies, browser nkeys.

NPE: npe.inbox.<id_fp>, HPKE ChaCha, max 900 KiB payload.

HOST NS1 70.88.205.138

Hostname: NS1.GEORGELAMBERT.ORG.

6.1 Existing

- Caddy /etc/caddy/sites/*
- Peergos: CT 502 peergos-a 10.10.10.12:8000 → peergos.georgelambert.org
- Links: CT 503 peergos-b 10.10.10.13:8000 → links.peergos.georgelambert.org
- NATS cluster CTs 511–513
- Forgejo git.georgelambert.org

6.2 Public Caddy maps (Let's Encrypt)

- pfc.georgelambert.org → 127.0.0.1:18780 (Python pfc-py-admin, PFC_REQUIRE_AUTH=1)
- config.pfc.georgelambert.org → 127.0.0.1:18781 (pfc-configd)
- docs.pfc.georgelambert.org → /opt/pfc/docs/html
- sync.pfc.georgelambert.org → /opt/pfc/docs/html/sync
- georgelambert.org/pfc/ → site tree deep-link page (does not rewrite other websites)
- peergos.georgelambert.org → CT 502 10.10.10.12:8000

Python IPFS pin API is 127.0.0.1:18782 (Kubo 127.0.0.1:15001). OrgTool bridge: pfc-peergos-bridge.service; connector: pfc-connector.service. Session secret and Google client live in /opt/pfc/etc/admin.env (mode 0600, not git).

Do not put S3 keys or PEERGOS_PASSWORD in this document.

GO SERVICES

Repo: `peergos-compliance-go`.

Each of `pfc-admin`, `pfc-pin`, `pfc-repl` starts `internal/leaf`: a local NATS (127.0.0.1 random port) with `LeafNode`. Remotes to the hub.

On `ns1` the **live console HTTP** is Python `pfc-py-admin` (conflicts with Go `pfc-admin.service`). Go `pin` is disabled so Python IPFS owns the `pin` queue. `pfc-configd` still serves the configuration SPA.

Build:

```
cd peergos-compliance-go
make linux    # GOOS=linux GOARCH=amd64
```

Units: `/etc/systemd/system/pfc-*.service` on `ns1`, User `root` or `pfc`, bind HTTP to 127.0.0.1 only.

CONFIGURATION SYSTEM

Repo: `peergos-compliance-config`.

- Browser wizard: `web/index.html`
- Bash: `./pfc-config` uses `dialog(1)` (Debian installer) or Python `curses` (kernel `menuconfig`: arrows, Enter, cycle profiles).

Schema: `schema.json`. Output: `~/pfc.json / $PFC_JSON`.

OPERATIONS

Laptop tests:

```
cd ~/research/peergos-for-compliance
make test
tail -f PROGRESS.log
```

Console and docs:

```
https://pfc.georgelambert.org
https://docs.pfc.georgelambert.org
```

Python admin on ns1:

```
rsync lib/pfc/ and pfc_admin/ to /opt/pfc/python/
sudo systemctl restart pfc-py-admin pfc-connector pfc-peergos-bridge
```

Stamp interval: PFC_STAMP_SYNC (seconds, default 15).

Do not open 4222/5001/7422 on the public IP.

WALKTHROUGHS

Operator click-paths for the PFC console and config SPA. Screenshots are captured by `scripts/capture_ui.py` (Playwright) against a local admin with `PFC_REQUIRE_AUTH=1`.

This is **not** a HIPAA/SOC 2/ISO certificate.

Vocabulary: **passthrough** means destination in the clear, body ciphertext; **lookup_id** is the sender handle; **NATS** is the untrusted broker; inspect officers are **share names**, not Google accounts.

10.1 Console login (separate from Peergos Drive)

Capability. Create a console session. Drive login cannot unwrap cubes.

Steps.

1. Open <https://pfc.georgelambert.org> — username, password, TOTP.
2. First account becomes **admin**. Sign up, then enroll TOTP (authenticator URL).
3. Confirm a 6-digit code. Session cookie `pfc_sess`.

Why. `PFC_REQUIRE_AUTH=1`. HIPAA APIs require `totp_ok`.

Outcome. Nav: Cubes, Catalog, Users, Inspect, Audit.

Storage. `PFC_DATA` UserStore JSON. Not Peergos cryptree.

PFC Console

Sign in

Console login is separate from Peergos Drive. Google is optional SSO. TOTP is required for hipaa cubes.

Username	Create account
	username
Password	password
	email
TOTP (if enrolled)	Sign up
000000	
Log in	

PFC Console
Cubes
Catalog
Users
Inspect
Audit
alice - enroll 2FA
Log out

Enroll TOTP

Scan with an authenticator, then confirm a code.

otpauth://totp/PFC:alice?secret=FJ0I6GBLCY4ZKR36WPUC6YQTQ4ATVYQ2&issuer=PFC&digits=6&period=30

Code

Confirm 2FA

DataCubes

Cube id
demo-cube

Profile
hipaa (lab stub)

Author id (must not be an inspect officer)
org-author

Inspect officers (usernames, comma — k-of-n, not the author)
bob,carol

Client
client / matter owner

Author
document author

Subject
subject

Title
title

Append text
demo record

Or file
Choose File No file chosen

Append

Copy object sha256 into dest cube
source cube
dest cube
object sha256
stamp: hash + original timestamp

☐ also copy file bytes into dest
Copy + stamp workspace

Create cube

Cubes — drop local files onto a row, or drag an object chip onto another cube. Known SHA-256 reuses the original timestamp receipt.

id	profile	title
Drop files from the desktop onto this cube (demo-cube), or onto a cube row.		

Objects in current cube (drag onto another cube row)

Comment / memo / link
comment

text or memo

target cube (for link)

target sha256 (optional)

Append to local chain

Flush stamp batch to leaf

New chain envelopes and unregistered SHA-256s sync to the NATS leaf about every 15s. Already-registered hashes are not restamped.

10.2 Cubes, catalog, copy+stamp

Capability. Workspace cube <user>-ws holds library cards (Client, Author, Subject, Title). Identity-preserving copy keeps original timestamp and SHA-256.

Steps.

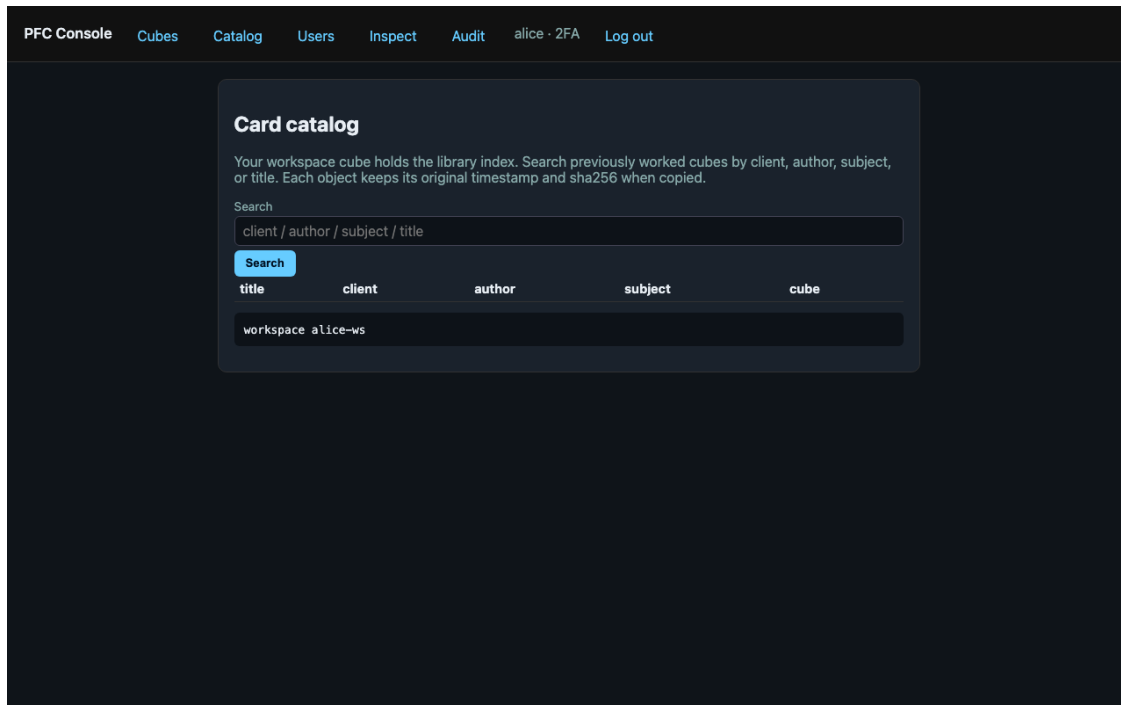
1. Create cube (hipaa lab stub or internal). Author must not be an inspect officer.

2. Append text or drop a file. Catalog prompt: keep cube defaults or edit this item.
3. Copy object into dest cube; stamp scope hash/public/sealed/files.
4. Comments/memos/links append the **cube** chain; unregistered hashes flush to `verae.ts.batch` about every 15s.

Why. First registered SHA-256 wins. Known hashes are not restamped.

Outcome. Object chips; workspace catalog search.

Storage. `cubes/<id>/chain.JSONL`, `catalog/card.json`, hash registry. Pin via `verae.storage.pin`; replicate via `verae.storage.replicate`.



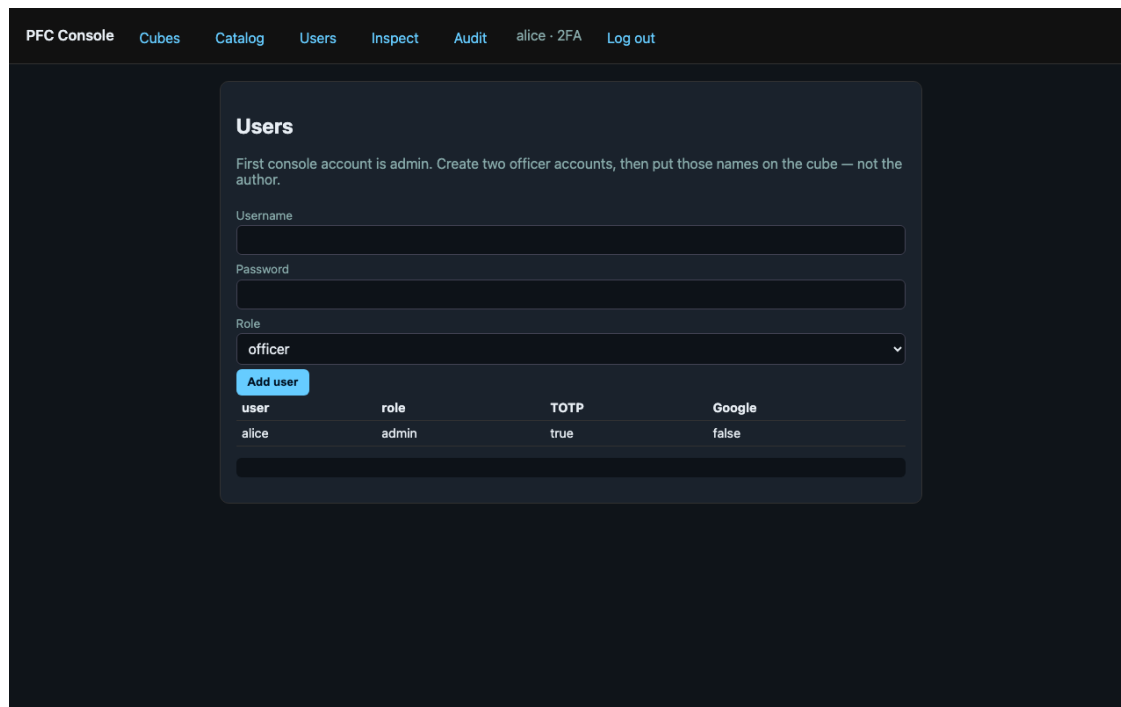
10.3 Users

Capability. Admin creates officer accounts for k-of-n inspect.

Steps. Add username/password/role. Officers enroll their own TOTP.

Why. Two-officer presence. Author is not an officer.

Storage. UserStore under PFC_DATA.



10.4 Inspect (k-of-n)

Capability. Viewer-only reveal after k officers present. Log-before-reveal.

Steps.

1. Officer A **Start inspect** → join code (no session yet).
2. Officer B **Join** with the code → `session_id`.
3. **View** object by sha256. **Close** ends the session.

Why. Shamir shares; Google SSO does not decrypt cubes.

Outcome. Audit events. HIPAA file bytes stay gated without a session.

Storage. Pending inspect JSON + audit log on the cube.

The screenshot shows the 'Inspect (k-of-n)' interface in the PFC Console. The top navigation bar includes 'PFC Console', 'Cubes', 'Catalog', 'Users', 'Inspect', 'Audit', 'alice - 2FA', and 'Log out'. The main content area has a dark background with a light gray panel containing the following elements:

- Inspect (k-of-n)** title.
- Instructional text: 'Officer A starts a session (join code). Officer B opens this page in another browser, signs in, and joins. Bytes are shown only after k officers are present.'
- Cube** section with a text input field containing 'demo-cube' and a blue 'Start inspect (this user)' button.
- Join code** section with a text input field.
- Buttons: 'Join as second officer' and 'Refresh presence'.
- Object sha256** section with a text input field.
- Buttons: 'List objects', 'View', and 'Close'.
- A JSON object preview:

```
{  "cube_id": "demo-cube",  "objects": [],  "items": []}
```

10.5 Audit

Capability. List access events for a cube (no payload bodies).

Steps. Enter cube id, **Load**.

Storage. Audit JSONL on the cube.

The screenshot shows the 'Audit log' interface in the PFC Console. The top navigation bar is identical to the previous screenshot. The main content area has a dark background with a light gray panel containing the following elements:

- Audit log** title.
- Cube** section with a text input field containing 'demo-cube' and a blue 'Load' button.

10.6 Config SPA (unsigned preview only)

Capability. Build a `pfc.json` preview. **Load requires an Ed25519 signed wrapper.** Admin POST `/v1/admin/config` signs and appends admin-history (prev + new + diff).

Steps. Fill host / leaf hub / NATS URL / profile / min_ok. Preview JSON. Download. Sign via console or `secure_messaging.signed_config.sign`.

Why. Unsigned files are rejected.

Peergos-for-compliance configuration

Same schema as the bash `pfc-config` menuconfig TUI. Saves JSON in the browser (download) for `/opt/pfc/pfc.json`.

Host (ns1)

Leaf hub

Direct NATS URL (optional)

Lab PSK (PFC_NATS_KEY)

Admin HTTP

Config HTTP

Profile

min_ok

```
{
  "host": "70.88.205.138",
  "leaf_hub": "nats-leaf://10.10.10.21:7422",
  "nats_url": "",
  "nats_psk": "",
  "admin_http": "127.0.0.1:18780",
  "config_http": "127.0.0.1:18781",
  "profile": "internal",
  "min_ok": 2,
  "caddy_hosts": {
    "pfc.georgelambert.org": "127.0.0.1:18780",
    "config.pfc.georgelambert.org": "127.0.0.1:18781",
    "docs.pfc.georgelambert.org": "static:/opt/pfc/docs/html"
  },
  "replicas": {
    "verae.pfc.replica.a.ingest",
    "verae.pfc.replica.b.ingest"
  }
}
```

10.7 Secure message success vs failure

Success. `verae.sm.send` with `to` + ciphertext → `ack accepted=true` + `lookup_id`. Sender cannot reopen `ct`.

Failure. Missing `to` or empty `ct` → `SM-MISSING-TO` / `SM-EMPTY-CT`. Publishes `verae.sm.log.summary` (codes only), `verae.sm.error` (Network Error Bundle), `verae.sm.dead`. `ct_sender` is sender-only; `ct_system` is the ops bounce report.

TECHNICAL CONTROLS (NOT A CERTIFICATE)

This document maps **implemented** technical controls to common HIPAA / SOC 2 / ISO 27001 *themes*. It is **not** a HIPAA, SOC 2, or ISO certificate and must not be presented as one.

NATS is the **untrusted broker**. Production end-to-end is **NPE / HPKE**. **Passthrough** means destination (and subject) may be in the clear so routers work; the **body is ciphertext**. Encrypted processing is used on every NATS hop we control so a compromised broker cannot read PHI.

11.1 Access control

- PFC_REQUIRE_AUTH=1 on the console. TOTP for hipaa cubes.
- Console login is a separate identity plane from Peergos Drive.
- Inspect is k-of-n officer **share names** (not Google accounts).
- Author cannot be an inspect officer.

11.2 Cryptography

- Production: NPE HPKE (ChaCha20-Poly1305) via `npe send`; fail-closed (never falls back to lab-xor when `crypto.mode=npe` or `PFC_REQUIRE_NPE=1`).
- Lab bus: `pfc-lab-xor` with host PSK (`PFC_NATS_KEY`). `PFC_FORBID_PLAIN=1` refuses `alg=plain`.
- Signed Ed25519 config wrapper; unsigned files rejected.
- Dual hash (SHA-256 + BLAKE2b/BLAKE3) on cube objects.

11.3 Audit and logging

- Log-before-reveal on inspect.
- Admin-history cube: previous file, new file, unified diff, actor, hashes.
- `verae.sm.log.summary`: codes + `lookup_id` + dest class — never ciphertext or recipient payload.
- Network Error Bundle: `ct_sender` (sender-only) + `ct_system` (ops).

11.4 Key custody

See the Custody chapter in this PDF. Lab PEMs are 0600 on the host. Production is customer HSM / PKCS#11.

11.5 Integrity

- Hash registry: first SHA-256 and first receipt win; copies keep original timestamp.
- Chain JSONL per cube; stamp unregistered hashes to the timestamp leaf.

KEY CUSTODY

12.1 Lab (ns1)

- Config Ed25519: `/opt/pfc/etc/sm-keys/config.ed25519.pem` (0600). Rotate with `rotate_config_key.py` (re-sign + admin-history).
- NPE: `npe keygen --out /opt/pfc/etc/npe/lab` then `npe keygen --rotate --id ....`. In-box id stays stable.
- Bus PSK: `/opt/pfc/etc/nats.env` (PFC_NATS_KEY). Not git.

12.2 Production

Customer HSM / PKCS#11 for Ed25519 signatures and NPE seed. Private material never in git or world-readable files.

This is **not** a HIPAA/SOC 2/ISO certificate.

VARIABLES

13.1 Signed wrapper payload (PFC_SIGNED_CONFIG)

- `crypto.mode`: `npe` | `lab-xor` | `plain-lab`
- `routing.mode`: `passthrough` (dest in the clear; body ciphertext)
- `logging.mode`: `summary`
- `admin.history_cube`: admin-history cube id
- `crypto.system_key_id`: Network Error Bundle system key

13.2 Process environment

- `PFC_REQUIRE_AUTH`, `PFC_REQUIRE_NPE` (unset on live ns1)
- `PFC_NATS_KEY`, `PFC_FORBID_PLAIN`
- `NATS_URL`, `SM_LEAF_HUB`, `SM_HTTP`
- `PFC_SIGNED_CONFIG`, `PFC_CONFIG_KEY_PEM`, `NPE_BIN`
- `PFC_IPFS_URL`, `PFC_REPL_URL`

Timeouts: NATS request-reply 5s, replica ingest 4s, pin 30s, NPE send 15s.