
NATS Service Endpoints

Verae

Sep 16, 2026

CONTENTS

1	Passthrough routing	3
2	Endpoint records	5
2.1	Secure-messaging subjects	5
2.2	Timeouts and filters	5
3	Errors and Network Error Bundle	7
4	Signed configuration	9
5	Message tracing	11
6	Passthrough	13

Canonical catalog for Verae / PFC / secure-messaging on <https://git.georgelambert.org/marchon/nats-service-endpoints>

The NATS broker is **untrusted**. Production E2E is NPE (HPKE). Lab may use `pfc-lab-xor`. This is **not** a HIPAA/SOC 2/ISO certificate.

Companion PDFs in this directory (relative, not Markdown):

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

PASSTHROUGH ROUTING

```
sender --(ct, to, lookup_id)--> NATS subject
|                               |
|                               v
|                               sm-leaf / NPE
|                               |
+----- lookup_id only ----- v
                               recipient mailbox
```

Header (clear): to, from_lookup_id, alg, error_token. Body: ciphertext. Broker must not log ct.

ENDPOINT RECORDS

Full JSON: `catalog/endpoints.json`. PFC import: `catalog/pfc-derived.json` from <https://git.georgelambert.org/marchon/peergos-for-compliance/src/branch/main/schemas/subjects.json>

2.1 Secure-messaging subjects

- `verae.sm.send` – passthrough deliver; from console/leaf; reject unsigned config and empty ciphertext (unless `plain-lab`).
- `verae.sm.dead` – dead letter; Network Error Bundle in body.
- `verae.sm.error` – emit bundle: `ct_sender` + `ct_system`.
- `verae.sm.log.summary` – central summaries (`codes`, `lookup_id`, `dest_class`).
- `verae.admin.config.sign` – Ed25519 wrap; unsigned files rejected.
- `verae.admin.history.append` – prev, new, unified diff on DataCube chain.

2.2 Timeouts and filters

Each derived row in `catalog/pfc-derived.json` has `timeout_sec` and `filter`. Request-reply defaults to 5s, replica ingest 4s, pin 30s. SM send rejects missing `to`, empty ciphertext (unless `plain-lab`), plaintext body, and summary records that contain `ct`.

ERRORS AND NETWORK ERROR BUNDLE

Never put the intended recipient payload in an error object.

- **ct_sender** – failure status encrypted so **only the sender** can read it.
- **ct_system** – bounce report encrypted to the **system public key** for logging/recovery (lookup_id, error_code, dest_class, respond flag).
- **Dead letter** – `verae.sm.dead` with the same bundle.
- **Central log** – `verae.sm.log.summary`: codes only.

The sender identifies the original message with `lookup_id`, not by decrypting what they sent.

SIGNED CONFIGURATION

A config file is unusable unless it is an Ed25519 wrapper:

```
{ "payload": { ... }, "sig": { "alg": "ed25519", "key_id": "...", "signature":  
  ↪ "hex" } }
```

Select algorithms inside **payload** (the signed object):

- `crypto.mode`: `npe` | `lab-xor` | `plain-lab`
- `routing.mode`: `passthrough`
- `admin.history_cube`: cube id for admin-history
- `crypto.system_key_id`: system public key id
- `logging.mode`: `summary`

After an admin console change: sign again, then append previous file, new file, and unified diff to the admin-history DataCube (`verae.admin.history.append`).

MESSAGE TRACING

Happy path:

1. Load signed config (reject **if** unsigned).
2. seal(body) -> envelope **with** to + ct + lookup_id.
3. Publish verae.sm.send.
4. Leaf acks {accepted:true, lookup_id}.
5. Recipient opens ct **with** NPE (prod) **or** lab-xor (lab).

Failure path:

1. Leaf cannot deliver.
2. Build Network Error Bundle (no mail body).
3. Publish verae.sm.dead **and** verae.sm.error.
4. Sender opens ct_sender; ops opens ct_system.
5. Summary to verae.sm.log.summary.

PASSTHROUGH

Destination mailbox id and the NATS subject may be in the clear so routers can work. The **body** is ciphertext. After send, the sender cannot open that ciphertext; they keep a `lookup_id` only. The same rule applies to the responder.