architecture

Architecture

Flow

1. An MCP client calls `stamp`, `verify`, or `hash_report`.
2. `RoutingLedger` sends `scope=global` to `GlobalLedger` and every other scope to `FileLedger`.
3. `FileLedger` keeps an idempotent JSON list. The same digest, org, and scope returns the existing receipt.
4. `GlobalClient` posts the digest to VeraeTime and, when the body has no certificate yet, polls `/api/status/{jobId}`.
5. `verify` with `want_scope=global` posts `{ "certificate": "<id>" }` to `/api/verify`. Otherwise it checks the local file first, then VeraeTime.

Modules

Module	Role
`server.py`	MCP tools and process entry
`ledger.py`	Receipt model, memory ledger, file ledger, router
`global_client.py`	VeraeTime HTTPS client; blocking I/O runs in a worker thread
`nats_ledger.py`	Optional leaf `locald` client

Idempotency

Local stamps are idempotent per digest + org + scope. Global stamps send `restamp: false` and `wait: true`. VeraeTime may return the same certificate for the same digest. This server does not cache global receipts in the local file.

What this process is not

It does not replace `verae-timestamp` / `locald`. It does not write JetStream. It does not store document bytes. Memory ledger exists for tests and for `mcp dev` when a caller injects it. `main()` always builds `RoutingLedger`.