zapier.georgelambert.org · Markdown indexes
VERAE TIME × ZAPIER
01 system
site/overview/01-system.md

1. System

Verae Time proves a SHA-256 existed at a given time. Zapier lets customers register and look up hashes from the tools they already use. Billing and API keys live on **zappier-edge**. Timestamping, wait, webhooks, and archive fan-out live behind **verae-middleware**. Durable messaging is the **central Verae NATS.IO 3-server JetStream cluster**.

Hard rule

Zapier never connects to NATS, tree nodes, WORM archives, or `api.veraetime.net`. Those hops are middleware and workers only. If a trace ever shows a Zapier hop on a `verae.*` subject, do not push the app.

Planes

Plane	What	Network
Zapier cloud	`verae-zapier-app` via **access-zapier**	HTTPS to `:3024` only (not NATS)
Direct web	`verae-access-web` customer browser	HTTPS `:3021`
Direct customer API	`verae-access-api` `x-api-key`	HTTPS `:3022`
S2S leaf	`verae-access-leaf`	NATS `verae.access.leaf.in` / HTTPS `:3023`
Authz	`verae-access-authz`	`verae.access.authz.check` before any internal subject
Commercial edge	`zappier-edge` portal, admin, Stripe	Loopback `:3000`; public doors are access-*
Account balance	`zappier-account-balance` prepaid SoT	NATS `verae.billing.*` + HTTP `:3010`
CS / sales / accounting	`zappier-customer-service`, `zappier-sales-pricing`, `zappier-accounting-export`	Private HTTP; NATS statement/adjust
Middleware HTTP	`/zapier/v1/*` job id + wait	HTTPS from access-zapier / loopback edge
NATS cluster	JetStream subjects under `verae.*`	Private; loopback or SSH tunnel
Workers	poller, webhook-deliver, aggregator	NATS + HTTPS to chain or Zapier hooks
Archives	WORM + tree nodes	NATS broadcast query; bloom miss = silence
Chain	Verae timestamping	HTTPS `api.veraetime.net` or MOCK
Control	`verae-fleet`	Operator HTTP `:3850` on `0.0.0.0`; SSH to extra machines

Request in one sentence

A Zap step POSTs to zappier-edge (`x-api-key`); edge meters the call and, when `ZAPPIER_UPSTREAM` is set, forwards timestamp/receipt/hash to middleware `/zapier/v1`. Middleware splits hash vs files, writes the hash (or Merkle root) to the chain, publishes `verae.zapier.jobs.watch`, and either returns `jobId` or waits on `verae.zapier.jobs.events`. Meter events, CS credits, and portal reloads go over NATS `verae.billing.*` to **account-balance** (source of truth). Customers review the same statement in the portal; CS (`:3011`) and sales (`:3012`) review it on their department UIs. Attached metadata and bulk-summary **leaves** are found later by broadcasting `verae.archive.query` to every archive/tree node.

See diagrams.