

zapier.georgelambert.org · Markdown indexes

VERAE TIME x ZAPIER

04 gaps and finish

<site/packages/dev-overview/04-gaps-and-finish.md>

Gaps and finish list

Modules and addresses that are designed but **not fully connected, not running as fleet workers, or not on the live Verae/Zapier path.**

Blocked on the operator (not a NATS bug)

Item	State	To finish
Zapier Platform <code>push</code>	no login	<code>zapier login</code> → register → <code>MIDDLEWARE_BASE_URL=https://zap.zapier.georgelambert.org</code> → push private app
Live Verae	<code>MOCK_VERAE=true</code>	credentials; <code>MOCK_VERAE=false</code> ; prove one seal
nkeys / mTLS	cluster <code>\$G</code> only	<code>verae-nats-accounts</code> sketch; flip only after every <code>NATS_URL</code> has creds
Hash lookup on live OpenAPI	501	Verae API gap or our store

Running on HTTP/NATS but incomplete routing

Module	Running?	NATS	Gap
middleware HTTP + in-process job-poller + webhook-deliver	yes	watch/events/deliver	live Verae + Zapier push
account-balance	yes	billing RR + <code>usage.recorded</code> sub	prove 402 from books on a Zap when prepaid is 0
zapper-edge	yes	<code>usage.recorded</code> / <code>payment.recorded</code> / <code>credit.applied</code> / <code>customer.put</code> if <code>NATS_URL</code>	confirm live process env; Stripe live
access-* + authz	yes	authz + ingress	—
identity	yes (fleet)	bind/lookup	confirm portal always binds <code>veraeUserId</code>
jobs-events mailbox :3030	yes (fleet)	events counter	exclusive consumer off unless <code>JOBS_EVENTS_EXCLUSIVE=1</code>
CS / sales / accounting	yes	statement/adjust	staff UX
archive-worm + tree-node	yes on LXC 510 (keep)	<code>archive.put/query/reply</code>	only 1+1 copies; fleet overlay <code>enabled: false</code> for extra replicas
archive-aggregator	no as fleet worker	query/reply	enable overlay or finish in-process hook (<code>includeAttached</code> TBD in middleware)
request-splitter	library in middleware	<code>archive.put</code>	<code>verae.splitter.in</code> not used
<code>ZAPIER_USAGE</code> / <code>USAGE_WRITER</code>	stream yes	<code>verae.zapier.usage</code>	no consumer; middleware usage is local files
fleet job-poller / webhook-deliver extra copies	overlay false	same subjects	only enable if you want SSH-spread replicas

Module	Running?	NATS	Gap
			instead of in-process (same queue names)

Designed, not a live worker

Module	Address	What to do
verae-nats-process	verae.example.process.*	template only; clone to add a function
verae-activate	none (HTTPS Zapier)	optional tiny app; needs Zapier login
verae-zapier (full app)	HTTPS	push after login
simulator	in-process fake NATS	local :3847 ; not production
verae-nats-accounts	nkeys policy	not flipped

Aux (no product subjects, or HTTP only)

Module	NATS	Notes
verae-keep	none	HTTP health; restarts units
verae-fleet	none	catalog + keepFloor
verae-uptime	none	off-box HTTPS GET of public doors
verae-backup	none	disk / JetStream snapshot; no live event subject
verae-bootstrap / verae-deploy / verae-proxmox / verae-ops	none	install/link
staff-iam / staff-session / staff-ui	none	HTTP cookies
public NATS edge	leaf, token	do not use from Zapier; rotate token; later nkeys LEAF vs INTERNAL

Billing triggers and event logging

Signal	Designed	Running	Finish
Timestamp/verify meter (edge)	<code>verae.billing.usage.recorded</code>	code path exists (<code>zappier/src/app.ts</code> <code>onUsage</code>)	Confirm <code>NATS_URL</code> on live edge; 402 from account-balance when prepaid empty
Middleware quota	<code>local files recordUsage</code>	yes	Either publish <code>usage.recorded</code> too or drop duplicate quota
Optional JS meter	<code>verae.zapier.usage</code> + <code>USAGE_WRITER</code>	stream only	Wire or delete
Portal reload	<code>payment.recorded</code> + <code>balance.adjust</code>	coded	confirm both fire on Stripe/test reload
CS credit	<code>balance.adjust</code> + <code>credit.applied</code>	coded	confirm applied pub on live
Customer upsert	<code>customer.put</code>	coded	—
Authz deny audit	<code>verae.access.authz.deny</code>	running on deny	no SIEM sink
Job lifecycle log	<code>jobs.events</code>	running	mailbox counts only — add structured log/export if needed
Archive put/query	<code>VERAE_ARCHIVE</code>	510 worm/tree	aggregator not in wait path
Public edge	leaf	running	rotate token; do not expose <code>:6222</code>

Ordered finish for message routing

1. **Do not double-start** job-poller: either in-process **or** fleet copies, same queue `job-poller`.

2. Confirm edge `NATS_URL` so `usage.recorded` actually lands; then 402 from books. Unify or drop middleware local `recordUsage`.
3. Enable **archive-aggregator** (or in-process equivalent) so `includeAttached` and tree proofs work on wait/status.
4. Raise worm/tree **replica floors** on 510/more CTs if archive HA is required (fleet overlay min 3).
5. Operator: Zapier `push` + `MOCK_VERAE=false`.
6. Prove: Zap create → `jobs.watch` → live `GET /api/status` → `jobs.events` → wait body + REST Hook.
7. nkeys/mTLS after every client has creds; then lock down the public edge to accounts (LEAF vs INTERNAL).
8. Optional: exclusive `jobs-events` mailbox; SIEM on `authz.deny` and `jobs.events`;
`verae.splitter.in` if splitter is split out; `USAGE_WRITER` or delete `ZAPIER_USAGE`.

What is already enough for a mock Zap

HTTPS zap/mw → mock create → `jobs.watch` → in-process poller → mock status completed → `jobs.events` → wait returns mock certificate. Archive extras and live chain are the missing middle.