

zapier.georgelambert.org · Markdown indexes

VERAE TIME x ZAPIER

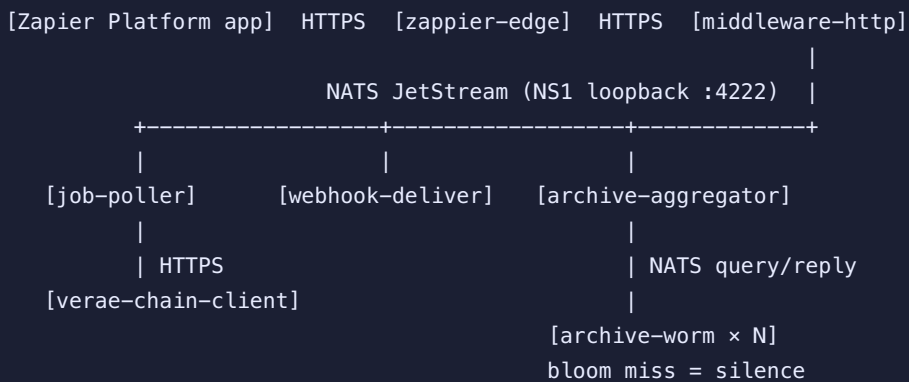
Module catalog and NATS addresses

Modules and NATS message contracts

Master map of every service in the Zapier [?](#) middleware [?](#) chain [?](#) WORM-archive path.

Zapier cloud **never** connects to NATS. Only middleware workers and archives do.

Who talks to whom (HTTPS vs NATS)



Module catalog

Module	Repo (Forgejo)	Runtime	Listens	Sends
zapier-platform-app	<code>verae-zapier-app</code>	Zapier cloud	User Zap steps	HTTPS to zappier-edge
zappier-edge	<code>zappier-edge</code>	Public HTTPS :3000	Zapier, portal, admin	HTTPS to middleware
middleware-http	<code>verae-middleware</code>	Public HTTPS :3100	zappier-edge	NATS jobs.watch; HTTP to chain; wait on jobs.events
job-poller	<code>verae-job-poller</code>	Worker	<code>verae.zapier.jobs.watch</code>	<code>verae.zapier.jobs.events</code> ; HTTP GET chain status
webhook-deliver	<code>verae-webhook-deliver</code>	Worker	<code>verae.zapier.webhooks.deliver</code> + events	HTTPS POST Zapier REST Hook
request-splitter	<code>verae-request-splitter</code>	In middleware	HTTP body / multipart	chain hash; <code>verae.archive.put</code>
archive-aggregator	<code>verae-archive-aggregator</code>	Worker / in wait	<code>includeAttached</code> on wait	<code>verae.archive.query</code> ; reads <code>verae.archive.reply.<id></code>
archive-worm	<code>verae-archive-worm</code>	N copies	<code>verae.archive.query</code> , <code>verae.archive.put</code>	<code>verae.archive.reply.<id></code> if bloom hits
tree-node	<code>verae-tree-node</code>	N copies (WORM role)	<code>verae.archive.query</code> , <code>verae.archive.put</code> kind <code>tree</code>	<code>verae.archive.reply.<id></code> if bloom hits
zapier-simulator	<code>verae-zapier-simulator</code>	Local HTTP :3847	operator browser	in-process replay of all addresses
fleet	<code>verae-fleet</code>		operator	spawns workers; keepFloor on tree-node

Module	Repo (Forgejo)	Runtime	Listens	Sends
		Local HTTP :3850		
overview	overview	Static	—	high-level system map
nats-process	verae-nats-process	Worker template	verae.example.process.in	.out / .reply.*
zapier-user-docs	zapier-user-docs	Static	—	catalog /user-docs/
verae-chain-client	verae-chain-client	Library	—	HTTPS api.veraetime.net or MOCK
docs-master	zapier-docs-master	Static	—	published on zapier.georgelambert.org

Addresses (subjects)

Address	Kind	Payload (required fields)
<code>verae.zapier.jobs.watch</code>	JetStream work queue	<code>tenantId</code> , <code>jobId</code> , <code>tokenRef</code> , <code>enqueuedAt</code> , <code>attempt</code> , <code>maxAttempts</code> , <code>intervalMs</code> , <code>traceId</code>
<code>verae.zapier.jobs.events</code>	JetStream events	<code>event (timestamp.completed\ failed\ timeout)</code> , <code>tenantId</code> , <code>jobId</code> , <code>status</code> , <code>traceId</code> , <code>emittedAt</code>
<code>verae.zapier.webhooks.deliver</code>	JetStream work queue	<code>hookId</code> , <code>tenantId</code> , <code>targetUrl</code> , <code>event</code> , <code>payload</code> , <code>attempt</code> , <code>traceId</code>
<code>verae.zapier.usage</code>	optional	<code>tenantId</code> , <code>action</code> , <code>amount</code> , <code>at</code>
<code>verae.archive.put</code>	JetStream	<code>sha256</code> , <code>tenantId</code> , <code>kind (publicMeta\ privateMeta\ file\ tree)</code> , <code>record</code> , <code>traceId</code>
<code>verae.archive.query</code>	pub to all archives	<code>correlationId</code> , <code>sha256</code> , <code>tenantId</code> , <code>kinds[]</code> , <code>traceId</code>
<code>verae.archive.reply.<correlationId></code>	replies	<code>archiveId</code> , <code>sha256</code> , <code>records[]</code> , <code>traceId</code>

Completed job JSON (back to Zapier)

See `docs/02-architecture/archive-nats.md`. Seal receipts from chain; extra receipts and `files[]` from archive aggregation when `includeAttached` is true.