

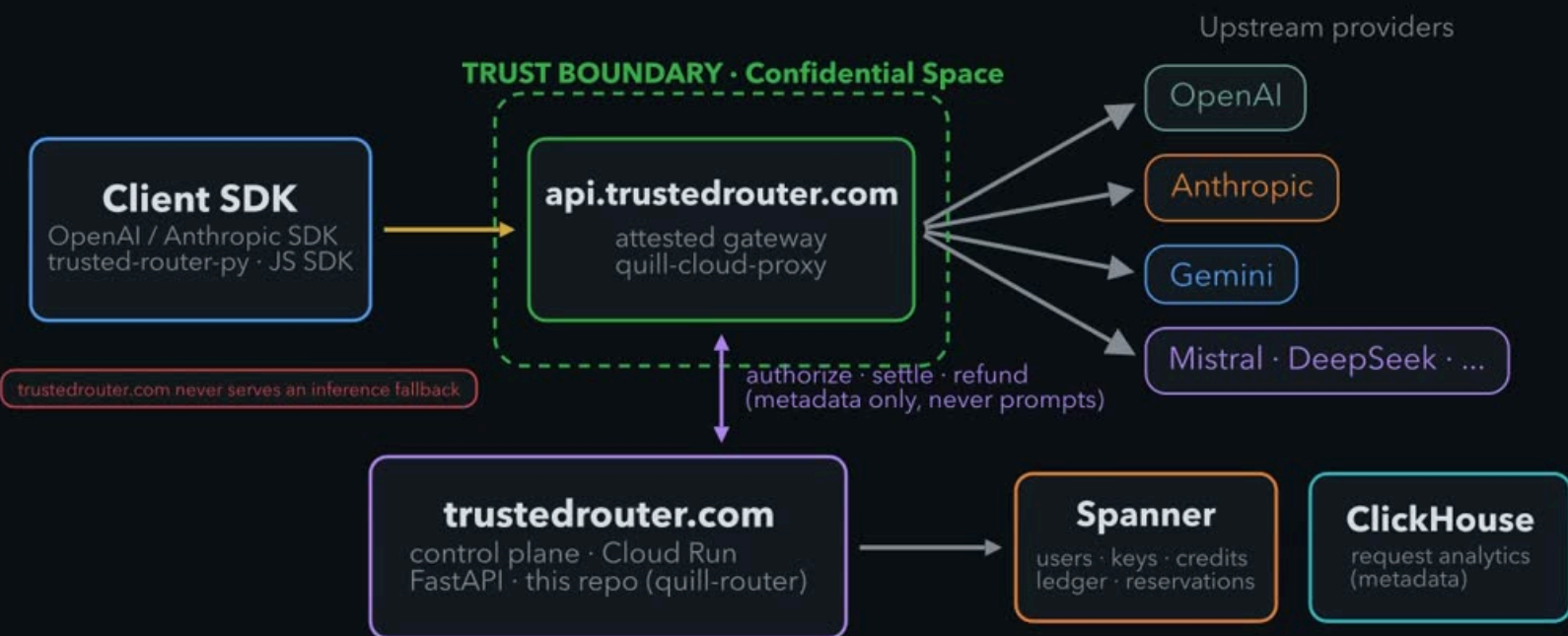
TrustedRouter Field Manual

Setup, architecture, money, data, infrastructure, security, the

Lore-Hex ecosystem and the leadership view, for quill-

01 The Big Picture

router



Spanner holds strongly-consistent money and identity state. ClickHouse holds analytics.

EDITION

2026-10-19
main @ 778120494

SCOPE

quill-router + 46 Lore-Hex
repos
13 chapters · 13 videos

MEASUREMENT

sha256:018bf584cb5506c1bd7f8bc3eec25bb1
432ecda0b4a833e8b4c29b7ce0637d37

About this edition

This handbook was produced by a parallel deep read of the `quill-router` repository and the Lore-Hex GitHub organization on 2026-10-19. Twelve research agents each owned one area, read the code, and cited it as `path:line`. One agent verified the Day-1 setup on a real Mac. Eight more rendered 13 Manim explainer videos, whose key frames illustrate every chapter opener and storyboard strip.

Nothing here touched production. Investigations were read-only: no deploys, no production database queries, no pushes. The one live call was the public attestation endpoint.

Reading order. Parts follow the path from zero to CTO: Orientation (hour one), Day 1 (run it and change it), Week 1 (understand the system), Month 1 (own the risky parts), Ecosystem (know the org), and Leadership (run engineering). Each chapter opens with its key sections and, where a video exists, a storyboard of it.

Trust the code over this book. `main` moves about 30 commits a day. Every claim cites a file so you can check it. Chapter 01 lists the places where the repo's own docs have already drifted from the code.

Companion site. The same content, with search, playable videos and dark mode, is in `trustedrouter-onboarding/site/`. Serve it with `python3 -m http.server 8765 --directory site`.

CONTENTS

Contents

PART I • ORIENTATION

01	Start here	5
	What TrustedRouter is, the three planes, the rules that bite on day one, and where the repo's docs are out of date.	
02	System teardown	9
	The whole system on one page, what each layer is made of, and the codebase in numbers.	

PART II • DAY 1

03	Local setup & configuration	14
	A verified Day-1 setup on macOS, real troubleshooting, and the complete environment variable reference.	
04	Contributing & testing	42
	The three CI gates, the test taxonomy, branch and PR conventions, and a first-PR walkthrough.	

PART III • WEEK 1

05	Architecture & request lifecycle	53
	Subsystem map, the three planes, and a chat completion traced end to end.	
06	Product surface & docs	67
	Every API surface, router alias, provider, script and doc in the repo, indexed.	
07	Security & trust	82
	Attestation, the trust boundary, privacy tiers, key storage, BYOK and the threat model.	

PART IV • MONTH 1

08	Money & billing	96
	Microdollars, the typed ledger, reserve/settle/refund, async settlement and payment rails.	
09	Data & storage	109
	The Store protocol, Spanner, Postgres, ClickHouse and the multi-cloud separation.	
10	Infrastructure & deploys	121
	Topology, CI/CD, rollout.sh and ENV_VARS, Terraform, runbooks and incidents.	

PART V • ECOSYSTEM

11	Platform repositories	138
	quill-cloud-proxy in depth, plus the infrastructure and provider repos in Lore-Hex.	
12	SDKs, apps & benchmarks	149
	Every SDK, integration, app and benchmark repo, with a feature matrix.	

PART VI • LEADERSHIP

13	The CTO view	161
	History, velocity, team, codebase health, risk register and a 30/60/90-day plan.	

APPENDIX

A	Storyboards	172
	Every key frame from the Manim explainer videos, in order.	

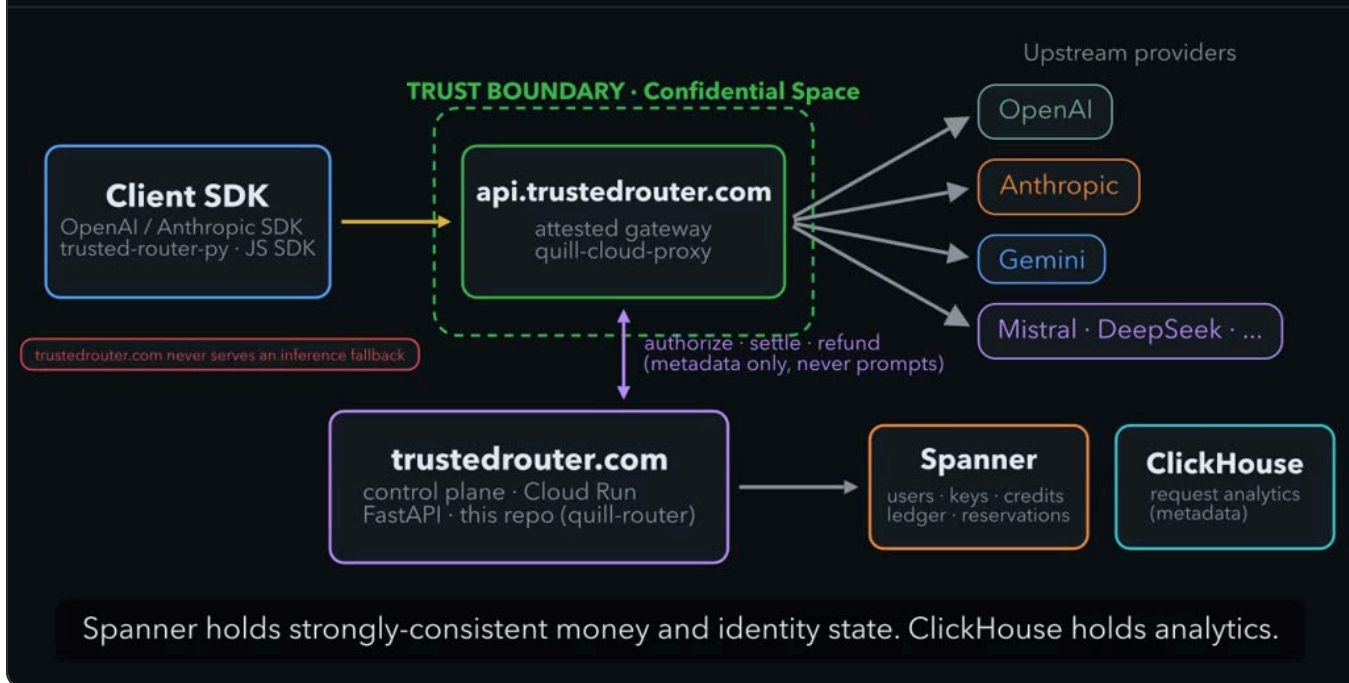
01

PART I

Orientation

1. Start here
2. System teardown

01 The Big Picture



PART I · CHAPTER 01 · HERO STILL: VIDEO 01 AT 0:42

Start here

What TrustedRouter is, the three planes, the rules that bite on day one, and where the repo's docs are out of date.

- TrustedRouter in one paragraph
- The numbers, as of 2026-10-19
- Rules that will bite you on day one
- Rebuilding this manual
- The three planes
- How to use this manual
- Where the repo's own docs are out of date

Storyboard · Video 01: The big picture

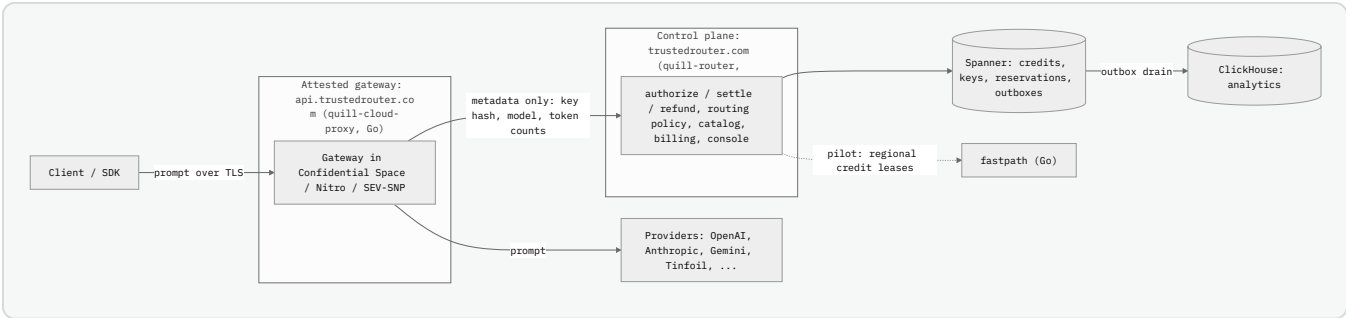


TrustedRouter in one paragraph

TrustedRouter (Lore Hex Corp) sells one OpenAI- and Anthropic-compatible API in front of 100+ model providers. The claim that sets it apart: the code path that touches your prompt runs in hardware-attested confidential VMs, and you can verify with a nonce that the running binary is the published source. Customers point their SDK at `https://api.trustedrouter.com/v1`, pick a privacy floor (`min_privacy: zdr` or `confidential`) and a model, and pay from prepaid credits.

The three planes

Most onboarding confusion comes from not knowing which plane a piece of code lives in.



PLANE	REPO	SEES PROMPTS?	OWNS
Attested gateway	<code>Lore-Hex/quill-cloud-proxy</code>	Yes, and only here	TLS termination in the TEE, provider calls, streaming, fallback, attestation endpoint
Control plane	<code>Lore-Hex/quill-router</code> (this repo)	No in production; inference routes mount only in local/test	Auth, keys, routing policy, catalog and pricing, credit ledger, payments, website, console
Fastpath	<code>quill-router/fastpath/</code>	No	Pilot admission against regional credit leases, running for two workspaces

The numbers, as of 2026-10-19

METRIC	VALUE
First commit	2026-05-03 (production the same day)
Commits / merged PRs	3,213 / about 1,711
Python source / tests	about 194k / 343k lines (1.77 test lines per source line)
Test files / tests	855 files, about 36.7k tests, sharded 12 + 8 + 6×2 in CI
Top-level modules in <code>src/trusted_router</code>	371, of which about 98 are the <code>async-settle</code> shadow program
<code>Store</code> protocol methods	210 (AGENTS.md still says about 97)
Providers in catalog	108
GitHub workflows	28
Incidents written up since Sep 5	25
Human committers	effectively one architect plus one front-end engineer; most code is written by Codex and Claude agents

How to use this manual

Follow the track in the left rail. Each stage assumes the one before it.

1. **Hour 1.** Read this page, then watch the six explainer videos (about 8 minutes in total).
2. **Day 1.** [Local setup & config](#) gets the app running on your Mac with verified commands. [Contributing & tests](#) covers the three CI gates and how a PR reaches production.
3. **Week 1.** [Architecture](#) shows where each module fits. [Product surface](#) maps every endpoint and doc. [Security & trust](#) explains what the attestation claim proves and what it does not.
4. **Month 1.** These are the high-risk areas. [Money & billing](#), [Data & storage](#), [Infra & deploys](#).
5. **Ecosystem.** [Platform repos](#) (start with quill-cloud-proxy) and [SDKs, apps & benchmarks](#).
6. **Leadership.** [The CTO view](#) covers history, velocity, the risk register, and a 30/60/90-day plan.

Rules that will bite you on day one

RULE	WHY
<code>uv run ruff check .</code> && <code>uv run mypy</code> && <code>uv run pytest -q</code> before claiming done	These are CI's three gates. mypy is a real gate.
Never run <code>ruff format .</code>	CI only checks. Formatting the tree produces a huge unrelated diff.
Patch backend classes in tests, never the <code>STORE</code> proxy	<code>monkeypatch.setattr(STORE, ...)</code> once broke 136 later tests (issue #333).
Rebase before trusting a failure	An hourly price bot commits to <code>main</code> .
Merging to <code>main</code> deploys	Path-filtered on <code>src/**</code> , <code>scripts/deploy/**</code> , <code>frontend/src/**</code> , <code>Dockerfile</code> , <code>lockfile</code> .
Config is code	Cloud Run env comes from <code>ENV_VARS</code> in <code>scripts/deploy/rollout.sh</code> . Hand edits are overwritten.
Money changes need differential tests	<code>reserve/settle/refund</code> . Run the <code>proof_oracle</code> marker explicitly.
Read-only is not load-safe	No <code>LIKE</code> /regex/JSON scans on <code>tr_entities.body</code> . Use <code>python -m scripts.inspect_payment_owner</code> .
Merging under <code>infra/</code> applies Terraform	There is no plan gate, and it can revert hand edits.

Where the repo's own docs are out of date

The research agents found these discrepancies. Trust the code over the doc until the doc is fixed.

DOC CLAIM	REALITY	SOURCE CHAPTER
AGENTS.md: <code>Store</code> has about 97 methods	210 in <code>store_protocol.py</code>	Data
README: activity data in Bigtable, store-backed rate limits	Bigtable retired 2026-09-28; Spanner outbox feeds ClickHouse; rate limits are per process	Architecture
README attestation snippet pipes to <code>jq</code> , reads <code>image_digest</code> and <code>pcrs</code>	The endpoint returns a JWT; the digest is at <code>submods.container.image_digest</code> ; GCP has no PCRs	Security
Runbook: deploy locks expire after 90 minutes	<code>cloud_rollout.py</code> never reclaims a lock on age alone	Infra
<code>docs/spanner-reliability.md</code> : fixed processing units	Terraform now uses the autoscaler	Data

DOC CLAIM	REALITY	SOURCE CHAPTER
docs/security.md: trust.trustedrouter.com is the control plane	Signed bundles publish from the gateway repo to GitHub Pages	Security
HANDOFF items 4.2-4.4	Obsolete since the Bigtable retirement	Data

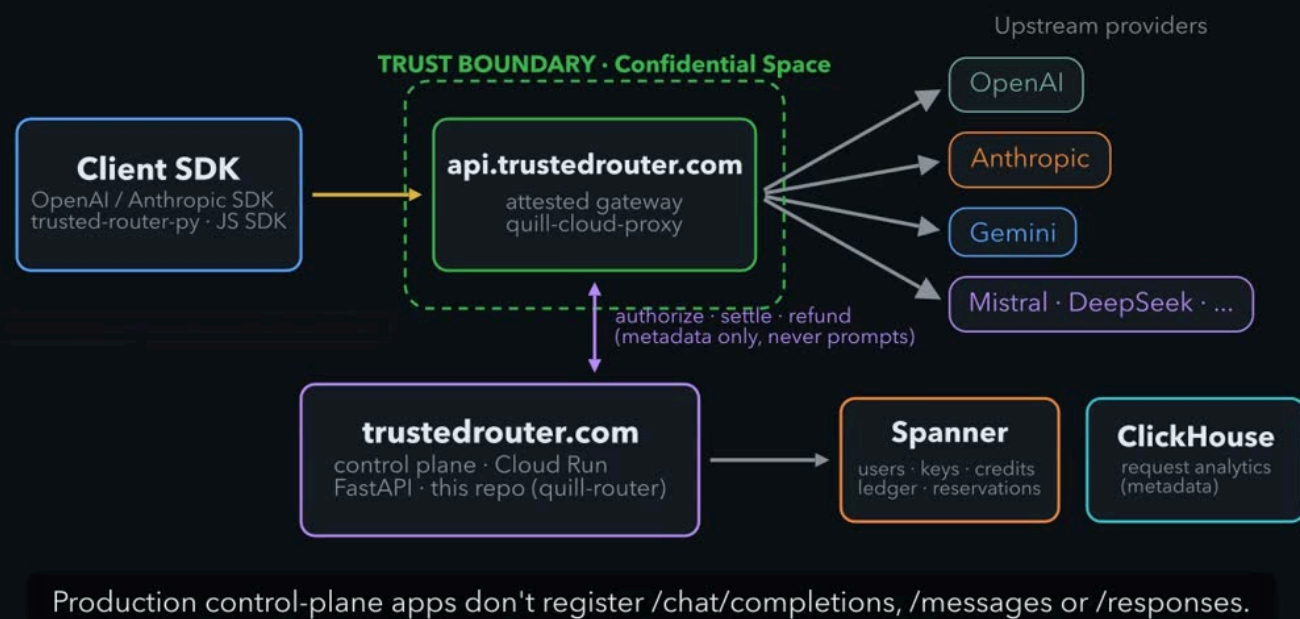
Rebuilding this manual

The source lives in `~/Code/trustedrouter/trustedrouter-onboarding/`:

```
cd ~/Code/trustedrouter/trustedrouter-onboarding
uv run --no-project --with markdown python build.py      # research/*.md -> site/
python3 -m http.server 8765 --directory site              # http://localhost:8765
```

Edit any chapter in `research/`, rebuild, and refresh the page. Manim scene sources are in `manim/`, and its README gives the render commands.

01 The Big Picture



PART I · CHAPTER 02 · HERO STILL: VIDEO 01 AT 0:47

System teardown

The whole system on one page, what each layer is made of, and the codebase in numbers.

- The system on one page
- The codebase in numbers

→ Teardown: what each layer is made of

Storyboard · Video 01: The big picture

0:27

0:33

0:38

0:42

0:47

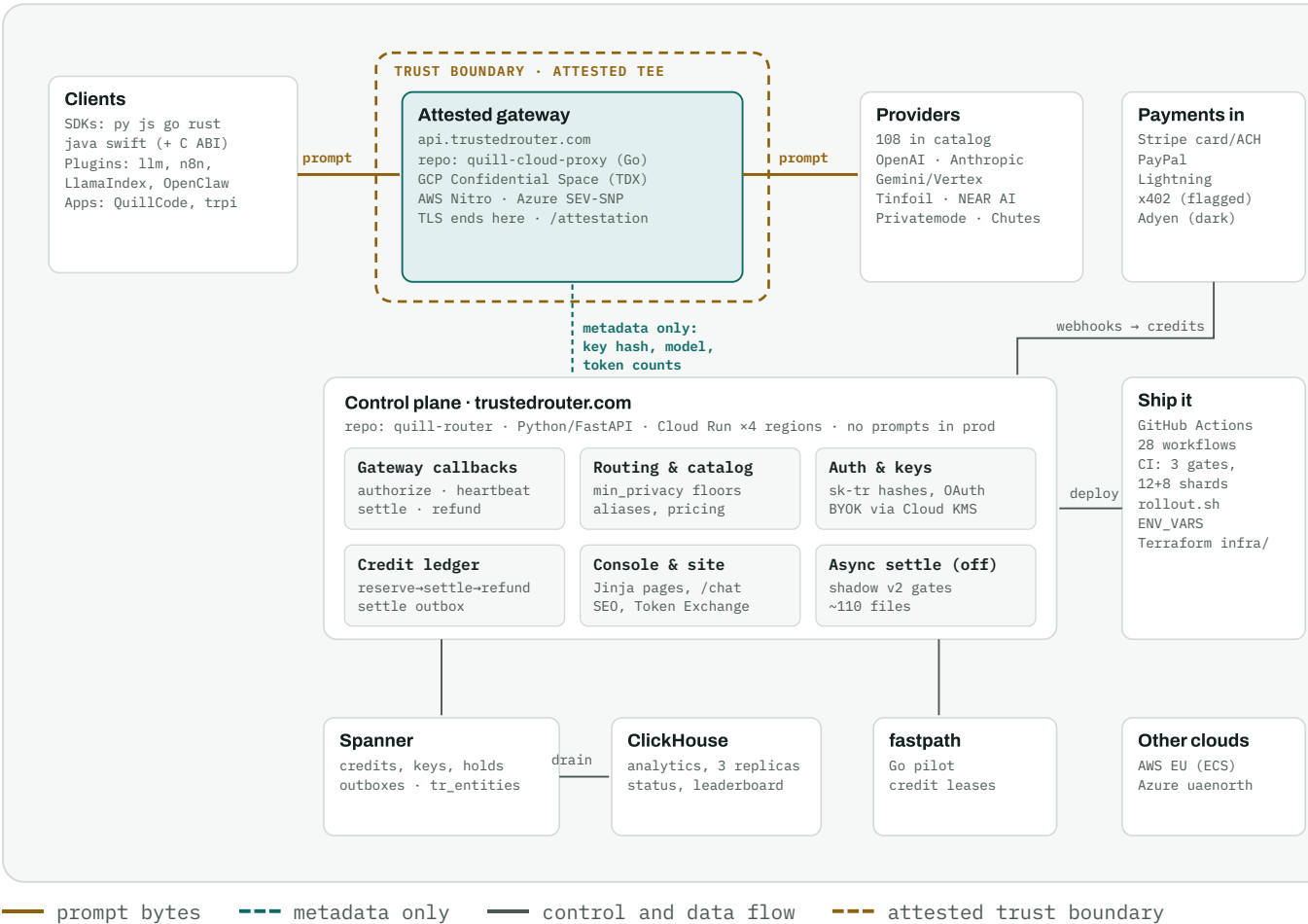
0:55

0:59

1:04

The system on one page

The dashed outline is the hardware trust boundary. Prompts exist only inside it and at the provider; everything below it sees metadata.



Teardown: what each layer is made of

LAYER	BUILT WITH	LIVES IN	OWNED STATE	BLAST RADIUS IF WRONG
Attested gateway	Go, Confidential Space / Nitro / SEV-SNP	quill-cloud-proxy	None persistent; BYOK key cache in memory	Privacy claim. A digest mismatch is caught by trust-drift.
Gateway callbacks	FastAPI, routes/internal/gateway.py (6k lines)	quill-router	Authorizations, reservations	Money: double charge or free usage
Credit ledger	Spanner typed tables, settle outbox, reaper	quill-router	tr_credit_balance, tr_key_limit, tr_reservation	Money; covered by the proof oracle and TLA+
Routing & catalog	Import-time catalog, 92 provider JSON files	quill-router	Prices, privacy flags	Mispricing, or a privacy floor routed to a non-qualifying provider
Storage layer	Store protocol (210 methods), 3 production backends	quill-router	Everything durable	Spanner CPU incidents; cross-cloud drift

LAYER	BUILT WITH	LIVES IN	OWNED STATE	BLAST RADIUS IF WRONG
Analytics	Spanner outbox → ClickHouse (3 replicas)	quill-router/clickhouse	Usage events, rollups	Status page, leaderboard, reports (not money)
Delivery	GitHub Actions, rollout.sh, Terraform	.github/, scripts/deploy/, infra/	Config-as-code ENV_VARS	Every merge to main can ship

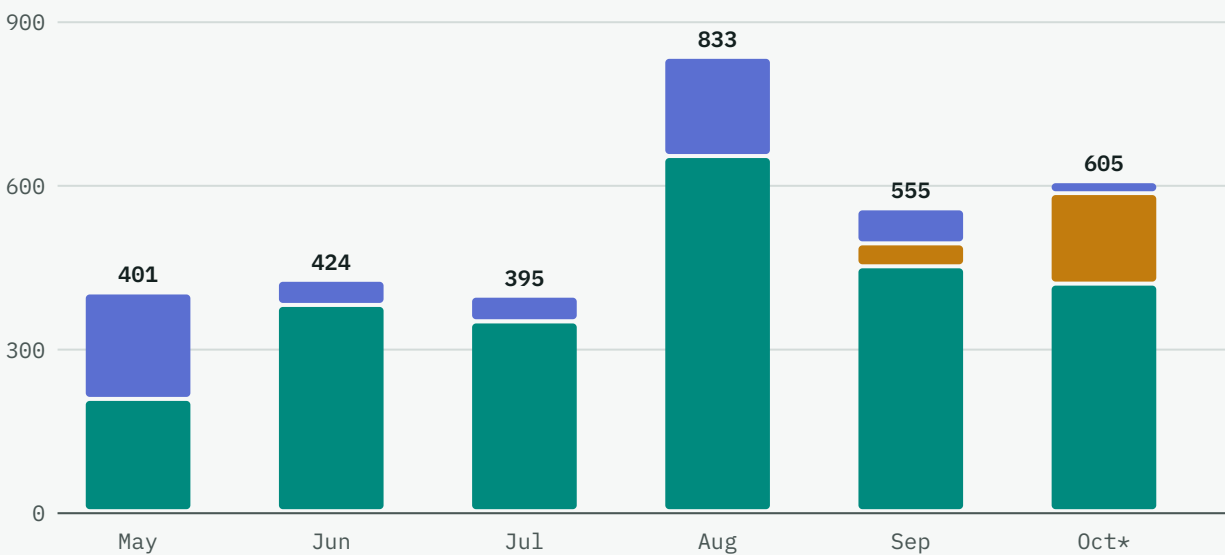
The codebase in numbers

3,213 commits since 2026-05-03	1,711 merged PRs	194k lines of Python source
343k lines of Python tests	28 GitHub workflows	25 incident write-ups

Commits per month, by author

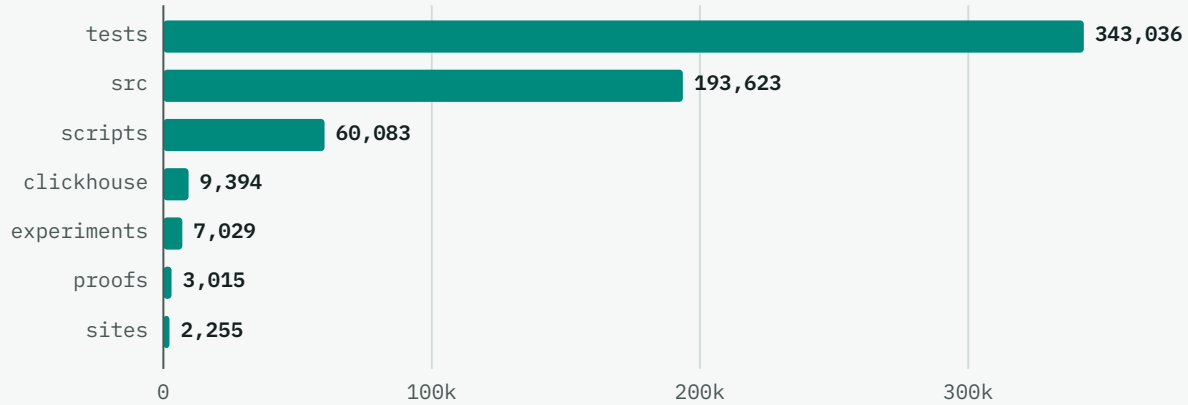
Oct* covers Oct 1–19. Bots are the hourly pricing bot and Dependabot.

■ Joseph Perla ■ Jasmine Lee ■ Bots



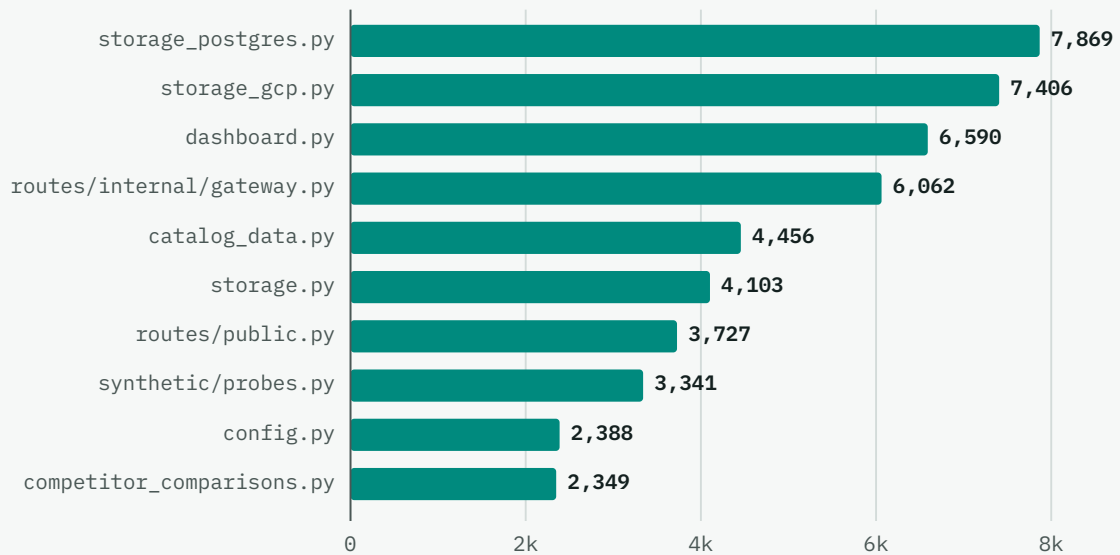
Python lines by top-level directory

Tests outweigh source by 1.77 to 1.



Largest source modules

Lines of code in src/trusted_router. The four largest are also the most-changed files.



Human-edited hotspots, last 90 days

FILE	CHANGES
src/trusted_router/routes/internal/gateway.py	141
src/trusted_router/dashboard.py	141
src/trusted_router/storage_gcp.py	140
src/trusted_router/config.py	134
scripts/deploy/rollout.sh	119
src/trusted_router/static/og/providers/manifest.json	112
src/trusted_router/storage_postgres.py	105
src/trusted_router/routes/public.py	102
src/trusted_router/catalog_data.py	100
src/trusted_router/storage.py	96

02

PART II

Day 1

1. Local setup & configuration
2. Contributing & testing

03 Day 1: Local Setup

```
zsh - ~/code

$ uv run uvicorn trusted_router.main:app --reload
INFO:      Will watch for changes in these directories: ['../quill-router']
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      Application startup complete.
```

Step 4: run the FastAPI app with auto-reload. Defaults are local and offline.

PART II • CHAPTER 03 • HERO STILL: VIDEO 03 AT 0:31

Local setup & configuration

A verified Day-1 setup on macOS, real troubleshooting, and the complete environment variable reference.

-
- | | |
|--|--|
| → 0. TL;DR (copy-paste, ~2 minutes on a warm machine) | → 1. Prerequisites |
| → 2. Python environment | → 3. The three CI gates (from AGENTS.md: "run all three before claiming done") |
| → 4. Run the app locally and exercise it end-to-end | → 5. Frontend / npm |
| → 6. Configuration: how settings are resolved | → 7. Env var reference |
| → 8. Troubleshooting (real issues hit during verification) | → 9. Safety reminders for local work |
| → 10. Verification log (raw timings) | |
-

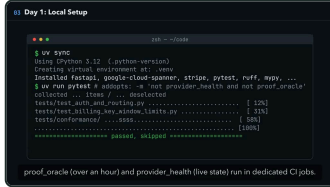
Storyboard · Video 03: Day-1 setup



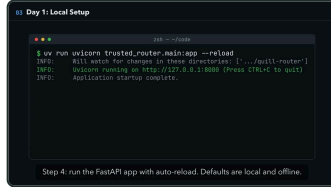
0:14



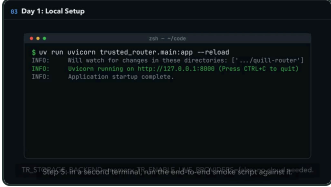
0:19



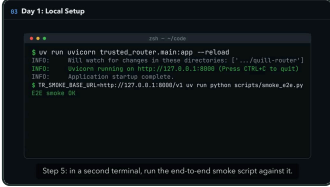
0:25



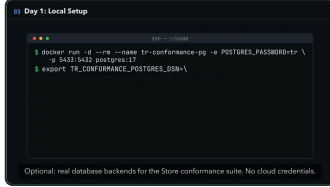
0:31



0:35



0:39

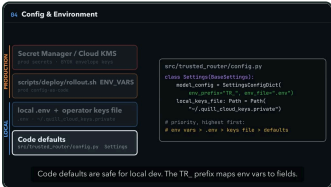


0:48

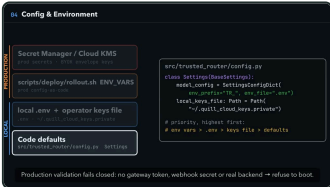


1:08

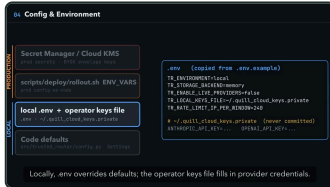
Storyboard · Video 04: Config & environment



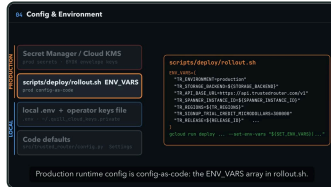
0:16



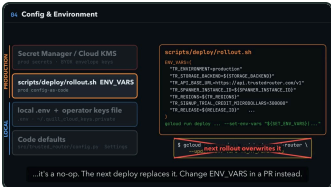
0:19



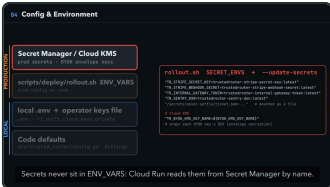
0:25



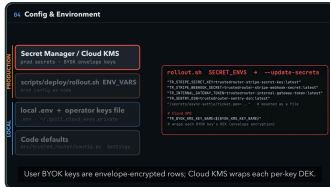
0:33



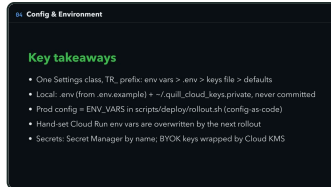
0:45



0:52



0:55



1:05

Every command below was run on a real Mac by the research agent on 2026-10-19 against `main @ 778120494`: macOS 26 (Darwin 25.6.0) on Apple Silicon, uv 0.11.6, Node v24.14.1, Docker CLI 29.6.1. ✔ = ran and passed exactly as written. ⚠ = problem found; details are given next to it. The repo stayed clean (`git status` empty) after every step.

0. TL;DR (copy-paste, ~2 minutes on a warm machine)

```
cd quill-router
uv sync # ✔ ~10 s (238 packages resolved)
uv run ruff check . && uv run mypy # ✔ ~0.5 s + ~8-10 s
uv run pytest -q -x tests/conformance -k memory # ✔ fast memory-backend contract
uv run uvicorn trusted_router.main:app --port 18080 --reload # ✔ starts in about 1 s
```

You do **not** need a `.env`, cloud credentials, Docker, or provider keys to run the app or most of the test suite. The defaults are local-safe: `TR_ENVIRONMENT=local`, `TR_STORAGE_BACKEND=memory` and `TR_ENABLE_LIVE_PROVIDERS=false`, so providers are mocked.

1. Prerequisites

TOOL	VERSION	WHY	VERIFIED HERE
Python	3.12 (<code>.python-version</code> ; <code>requires-python >=3.11</code> ; mypy targets 3.12; Docker image is <code>python3.12-bookworm-slim</code>)	Runtime	✅ uv auto-provisioned CPython 3.12.13
uv	any recent version (CI uses <code>astral-sh/setup-uv@v7</code> with <code>version: latest</code>)	Package/venv manager. Always run things through <code>uv run</code> .	✅ 0.11.6
Node + npm	CI pins Node 20 (<code>actions/setup-node</code> , <code>node-version: "20"</code>)	ESLint, Stylelint, <code>tsc</code> (dashboard TS → committed JS), Playwright	✅ Node 24.14.1 also works
Playwright Chromium	<code>@playwright/test ^1.62.1</code>	Browser smoke tests (<code>tests/browser/</code> , <code>tests-e2e/</code>)	✅ <code>npx playwright install chromium</code> (~95 MiB, about 12 s)
Docker	any	Only needed for the Postgres / pgadapter / Spanner-emulator conformance backends	⚠️ CLI is installed, but the daemon was not running on this Mac (see Troubleshooting T1). Not needed for anything else.
gcloud / AWS / Azure CLIs	—	Not needed for local dev. Deploys go through CI (see 04-infra-deploy).	n/a

Install hints (macOS):

```
curl -LsSf https://astral.sh/uv/install.sh | sh      # uv
brew install node@20    # or nvm install 20
# Docker Desktop / OrbStack / colima: only if you will run postgres/spanner conformance
```

2. Python environment

```
uv sync
```

✅ **Result:** Resolved 238 packages , installed the dev group (the default) into `.venv/`. **Time:** 9.8 s wall-clock with a warm uv cache. A second run takes 19 ms (Checked 214 packages).

Notes * The **dev** dependency group includes `docling` and `sec-parser`, which pull a large ML stack (torch, transformers). A cold-cache first sync therefore downloads a lot more than the production image needs. Production uses `uv sync --frozen --no-dev`. * CI runs `uv sync --frozen`. Use `--frozen` too if you want to be sure you are not silently changing `uv.lock`. * Do not `pip install` into the venv by hand. `uv run` re-syncs to the lockfile.

3. The three CI gates (from AGENTS.md: "run all three before claiming done")

```
uv run ruff check . && uv run mypy && uv run pytest -q
```

GA TE	COMMAND	RESULT ON THIS MAC	TIME
Lin t	<code>uv run ruff check .</code>	✅ All checks passed!	0.55 s
Typ e s	<code>uv run mypy</code>	✅ Success: no issues found in 533 source files	9.6 s (warm) / 8.3 s with a cold <code>--cache-dir .</code> mypy is <i>not</i> slow here, because of <code>follow_imports = "silent"</code> and <code>files = ["src/trusted_router"]</code> .
Test s	<code>uv run pytest - q</code>	Not run in full on purpose (multi-hour on one machine). Collection alone: 38,221 / 40,067 tests collected (1,846 deselected) in 29 s	—

Rules worth tattooing (AGENTS.md): * **Never run** `ruff format .`. CI only runs `ruff check`, and formatting produces a huge unrelated diff. * Coverage must stay $\geq 70\%$. CI shards the suite with `pytest-split` and runs `-n 4 --dist loadgroup`. * The default `addopts` deselects `proof_oracle` (over an hour of differentials) and `provider_health` (live provider state). If you touch `async settle` or `billing`, run the oracle explicitly: `uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py`. * CI's lint job also runs `uv run python scripts/update_model_changes.py --check`. ✅ It exited 0 here.

3a. Fast, targeted pytest runs (what to use during the inner loop)

```
# Storage contract against the in-memory backend (postgres/spanner params auto-skip without DSNs)
uv run pytest -q -p no:cacheprovider -x \
  tests/conformance/test_store_semantics.py tests/conformance/test_key_management.py -k memory
```

✅ 90 passed, 14 skipped, 416 deselected, 6 warnings in 1.07s (10 s wall-clock including uv and imports)

```
uv run pytest -q -p no:cacheprovider tests/test_billing.py tests/test_inference_errors.py
```

✅ 82 passed, 142 warnings in 7.72s

```
uv run pytest --collect-only -q          # count / sanity-check collection
```

✅ 29 s

Tips * `-n 4` (xdist) is available for bigger subsets. * `tests/conftest.py` forces `TR_STORAGE_BACKEND=memory`, and the operator keys file is **ignored under pytest** unless `TR_ALLOW_LOCAL_KEY_FILE_IN_TESTS=1`. Tests are hermetic by default. * **Pollution trap (issue #333)**: never `monkeypatch.setattr(STORE, ...)`. `STORE` is a `__getattr__` proxy. Patch the backend class instead. Run the full suite before you merge. * The warnings you will see are FastAPI `on_event` deprecation notices. They are harmless.

4. Run the app locally and exercise it end-to-end

```
uv run uvicorn trusted_router.main:app --port 18080          # add --reload while editing
```

✅ Log output:

```
INFO:      Started server process [9685]
INFO:      Waiting for application startup.
```

```
INFO: Application startup complete.
INFO: Uvicorn running on http://127.0.0.1:18080 (Press CTRL+C to quit)
```

Ready in under 1 s. (Production instead runs `python -m trusted_router.serve`, which reads `PORT`, `WEB_CONCURRENCY` / `UVICORN_WORKERS`, `UVICORN_PROXY_HEADERS`, `UVICORN_FORWARDED_ALLOW_IPS` and `UVICORN_TIMEOUT_KEEP_ALIVE`, and keeps the listener open briefly after `SIGTERM`.)

4a. Health and catalog

```
curl -s localhost:18080/health          # {"status":"ok"}    200, ~2 ms
curl -s localhost:18080/v1/health      # {"status":"ok"}    200
curl -s localhost:18080/v1/models/count # {"data":{"count":692}}
curl -s localhost:18080/v1/models | jq '.data[5][].id'
# "trustedrouter/auto" "trustedrouter/free" "trustedrouter/cheap" "trustedrouter/fast" "trustedrouter/green"
```

`/`, `/docs`, `/status` and `/openapi.json` all return 200. `/healthz` is **not** a route. It returns a markdown-flavoured 404 that points agents at `llms.txt` and `sitemap.xml`.

4b. Local auth: no OAuth needed

In `local / test` you can act as any user by sending the `x-trustedrouter-user` header (this is what `scripts/smoke_e2e.py` and the tests do). The first call auto-creates a "Personal Workspace".

```
B=http://127.0.0.1:18080/v1
H='x-trustedrouter-user: dev@example.com'
curl -s $B/workspaces -H "$H"
KEY=$(curl -s -X POST $B/keys -H "$H" -H 'content-type: application/json' \
  -d '{"name":"local dev"}' | jq -r .key)      # sk-tr-v1-...
```

4c. Fund the workspace, then chat (the non-obvious bit)

A brand-new local workspace has **\$0**, so the first chat returns `402 {"error":{"code":402,"message":"Insufficient credits","type":"insufficient_credits"}}`. Trial credit is only granted on the OAuth/email signup paths. With no `TR_STRIPE_SECRET_KEY`, `/billing/checkout` returns a `"mode":"mock"` session, and in `local / test` the Stripe webhook accepts **unsigned** JSON. So you can credit yourself like this:

```
WS=$(curl -s $B/workspaces -H "$H" | jq -r '.data[0].id')
curl -s -X POST $B/internal/stripe/webhook -H 'content-type: application/json' -d '{
  "id":"\`evt_local_1\`","\`type\`":"checkout.session.completed",
  "data":{"object":{"amount_total":2500,"payment_intent":"\`pi_local_1\`",
  "payment_status":"paid","\`metadata\`":{"workspace_id":"\`$WS\`"}}}}'
# {"data":{"credited":true,"credited_microdollars":25000000,"event_id":"evt_local_1",...}}
curl -s $B/credits -H "$H"      # available: 25.0
```

(The event is idempotent by `id`. Change `evt_local_N` to top up again.)

```
curl -s $B/chat/completions -H "authorization: Bearer $KEY" -H 'content-type: application/json' \
  -d '{"model":"trustedrouter/auto","messages":[{"role":"user","content":"hello"}]}'
```

Response (trimmed):

```
{"object":"chat.completion","model":"z-ai/glm-5.3-flash",
  "choices":[{"message":{"role":"assistant","content":"TrustedRouter response from z-ai/glm-5.3-
```

```
flash.{"finish_reason":"stop"}],
"usage":{"prompt_tokens":1,"completion_tokens":11,"total_tokens":12},
"trustedrouter":{"content_stored":false,"requested_model":"trustedrouter/auto","selected_model":"z-ai/glm-5.3-flash","rollover_failures":[]}]}
```

Streaming ("stream":true)  emits an event: trustedrouter.route SSE frame and then standard chat.completion.chunk deltas. GET /v1/activity?group_by=none  shows the metadata-only row (cost_microdollars: 3 , no prompt content).

How the fake provider works: services/inference.py builds ProviderClient(LocalKeyFile(settings.local_keys_file), live=settings.enable_live_providers). With live=False (the default, **and also in production**, where real inference happens in the attested gateway, quill-cloud-proxy), providers.py returns mock_text(model.id) → "TrustedRouter response from <model>." No network calls are made. Set TR_ENABLE_LIVE_PROVIDERS=true and put real keys in the keys file (section 6) to call real upstreams from your laptop.

Other observations: * No auth → 401 Missing Authentication header  * cerebras/llama3.1-8b → 400 model_not_supported. That model is no longer chat-capable in the catalog (see T3).

Stop the server with Ctrl-C (or kill -f "uvicorn trusted_router.main:app").

4d. scripts/smoke_e2e.py stale

```
TR_SMOKE_BASE_URL=http://127.0.0.1:18080/v1 uv run python scripts/smoke_e2e.py
```

 E2E smoke failed: expected 54 OpenRouter route classifications, got 95. The script (last touched 2026-05-04) hard-codes 54 coverage rows and uses cerebras/llama3.1-8b. It also assumes new workspaces have credit. Treat it as reference reading, not a working check. Use the curl flow above instead.

5. Frontend / npm

Root package.json (trusted-router-frontend). The dashboard TypeScript in frontend/src/ compiles into committed JS under src/trusted_router/static/.

SCRIPT	COMMAND	WHAT IT DOES	VERIFIED
install	npm ci	213 packages	 0.6 s
lint	npm run lint → eslint frontend/src	TS lint	 clean, 0.8 s
lint:css	npm run lint:css → stylelint src/trusted_router/static/*.css	CSS lint	 clean, 0.6 s
typecheck	npm run typecheck → tsc -noEmit	Type check only	 clean, 0.5 s
build	npm run build → tsc	Writes src/trusted_router/static/*.js. CI runs npx tsc and then git diff --exit-code -- src/trusted_router/static/dashboard.js src/trusted_router/static/token-index.js. If you edit .ts, commit the regenerated .js.	 the output matched the committed JS (no diff)

SCRIPT	COMMAND	WHAT IT DOES	VERIFIED
test:browser	<code>npm run test:browser → playwright test (playwright.config.js, testDir: tests/browser)</code>	Homepage/console/chat smoke. Its <code>webServer</code> starts uvicorn on :18081 with <code>TR_ENVIRONMENT=test</code> <code>TR_STORAGE_BACKEND=memory</code> <code>TR_RATE_LIMIT_ENABLED=false</code> and fake OAuth client IDs	✅ <code>npx playwright test tests/browser/homepage.spec.js --project=chromium → 9 passed (5.4 s)</code>
build-world-map	<code>node scripts/build-world-svg.mjs</code>	Regenerates the world SVG asset	not run (it writes tracked files)

First-time browser install: `npx playwright install chromium` ✅ (CI uses `--with-deps chromium`).

`tests-e2e/` is a *separate* npm project (its own `package.json` and `playwright.config.ts`) for the `/chat` playground, with 19 specs. It mocks `/v1/chat/completions` via `page.route()`:

```
cd tests-e2e && npm install && npm run install:browsers # chromium webkit firefox
npm test -- --project=chromium # also: test:headed, test:debug, test:ui
```

(Not run by this agent. The README says the server auto-starts via `webServer`.)

6. Configuration: how settings are resolved

All runtime config lives in **one** Pydantic Settings class: `src/trusted_router/config.py::Settings` (2,388 lines, **354 fields**). It is built by `get_settings()` (no caching side effects).

- `env_prefix="TR_": field storage_backend ↔ env TR_STORAGE_BACKEND`.
- `env_file=".env"` (relative to the CWD, gitignored). `.env.example` is the template.
- `extra="ignore": unknown TR_* vars are silently ignored, so a typo does nothing.`
- `hide_input_in_errors=True`: validation errors don't echo secret values.
- Validators fail closed. For example, `TR_ENVIRONMENT=production` alone refuses to start: `TR_SERVICE_SURFACE=combined` is restricted to `local/test` unless the temporary `TR_ALLOW_DEPLOYED_COMBINED_SURFACE=true` migration bridge is explicitly enabled ✅ (verified). `TR_REMEDIATOR_MODE` must be exactly `off|observe|act`.

Source priority (highest wins), from `Settings.settings_customise_sources`:

1. init kwargs (tests)
2. process environment (`TR_*`)
3. `.env` in the CWD
4. **operator keys file** `~/.quill_cloud_keys.private` (`_LocalKeyFileSource`). Only for the 53 secret/integration fields listed in `_LOCAL_KEY_FALLBACKS`, looked up by the **un-prefixed upper-case name** (e.g. `STRIPE_SECRET_KEY`, `GOOGLE_CLIENT_ID`, `SENTRY_DSN`). Skipped entirely under pytest.
5. file secrets (pydantic default; unused)
6. code defaults in `config.py`

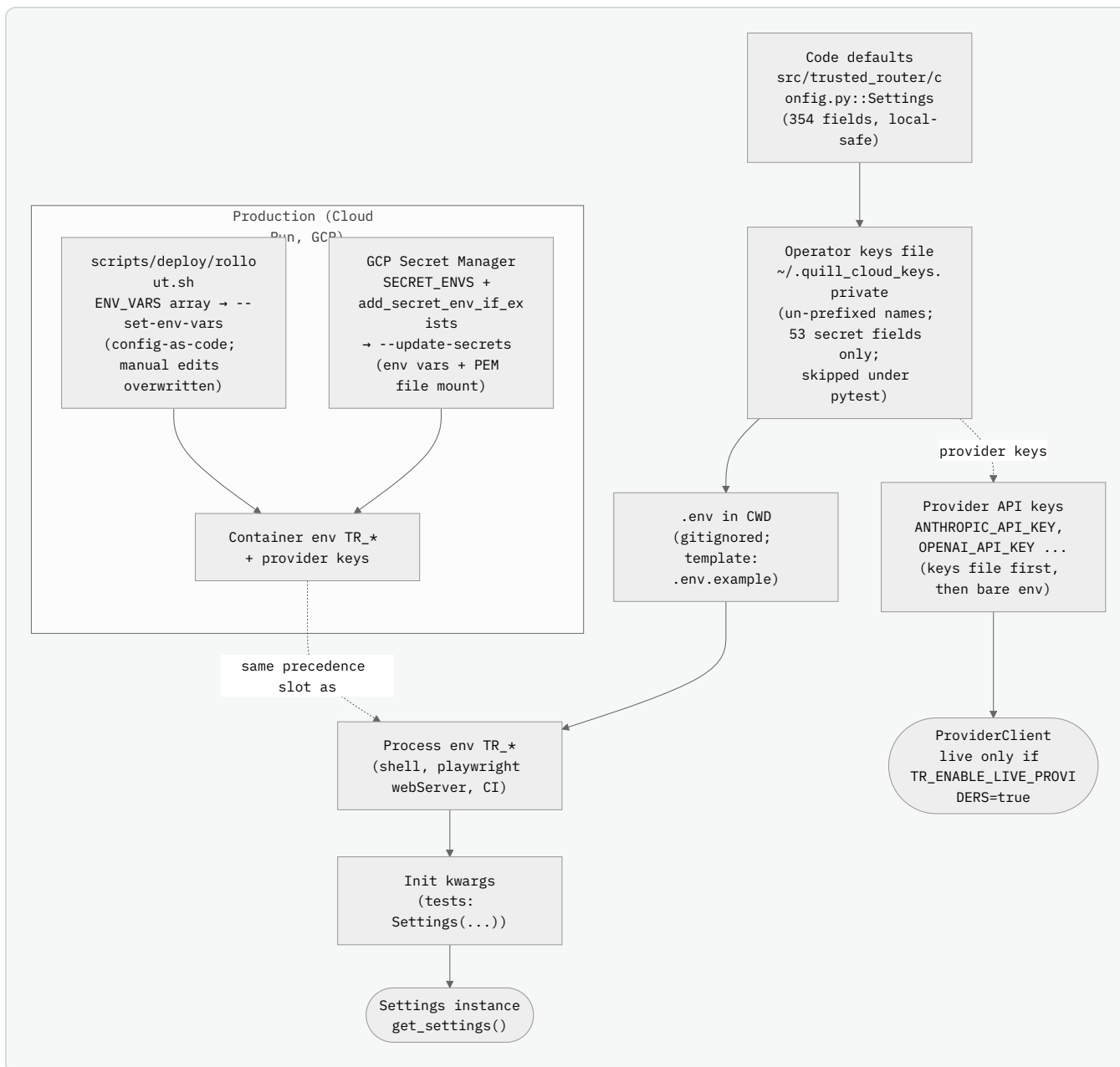
The operator keys file

- Dotenv-style, **never committed**. Parsed by `secrets.LocalKeyFile` (`KEY=value`, `#` comments, quotes stripped). Values are never logged.

- README documents `/Users/jperla/claude/.quill_cloud_keys.private`. That is Joseph's path. For you it is `~/.quill_cloud_keys.private` (the code default). It does not exist on this Mac, and nothing needs it.
- Two consumers: 1. **Settings fallback** (above): integration secrets such as Stripe, OAuth, SES, PayPal, Adyen, Sentry, `INTERNAL_GATEWAY_TOKEN` and `BOOTSTRAP_MANAGEMENT_KEY`. 2. **Provider client** (`providers.py::_secret`): provider keys such as `ANTHROPIC_API_KEY`, `OPENAI_API_KEY`, `GEMINI_API_KEY`, `CEREBRAS_API_KEY`, `DEEPSEEK_API_KEY` and `MISTRAL_API_KEY`. It checks the keys file first and then the bare process env. These are **not** `TR_`-prefixed.
- Aliases (`secrets.KEY_ALIASES`): `CLAUDE_API_KEY` → Anthropic, `CHATGPT_API_KEY` → OpenAI, `STRIPE_KEY` → `STRIPE_SECRET_KEY`, `ZHIPU_API_KEY` / `Z_AI_API_KEY` → `ZAI_API_KEY`, `TOGETHERAI_API_KEY` → Together, `DASHSCOPE_API_KEY` → Alibaba, and Vertex project/region/token aliases.
- Vertex in production uses the service account and ADC, **not** a long-lived key in this file.
- **⚠ Gotcha (code-verified):** `TR_LOCAL_KEYS_FILE` changes the path the **provider client** reads, but `_LocalKeyFileSource` reads `settings_cls.model_fields["local_keys_file"].default`, which is always `~/.quill_cloud_keys.private`. So the Settings fallback for Stripe, OAuth and the rest **ignores** `TR_LOCAL_KEYS_FILE`. `.env.example` sets it to Joseph's path. If you copy `.env.example` verbatim, provider keys are read from a non-existent file, while integration secrets still come from `~/.quill_cloud_keys.private`.

Production: config-as-code

- Runtime env for Cloud Run is the `ENV_VARS` array in `scripts/deploy/rollout.sh` (~120 `TR_*` entries; joined with `^|^` and passed as `--set-env-vars`). **Setting a Cloud Run env var by hand is a no-op.** The next rollout overwrites it. To change prod config, edit `rollout.sh` in a PR.
- Notable prod values: `TR_ENVIRONMENT=production`, `TR_SERVICE_SURFACE=combined` with `TR_ALLOW_DEPLOYED_COMBINED_SURFACE` (temporary bridge until #712), `TR_RATE_LIMIT_ENABLED=false` (no trusted edge client identity yet), `TR_ENABLE_LIVE_PROVIDERS=false`, `TR_REMEDIATOR_IN_PROCESS_ENABLED=false`, `TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS=3000000`, and `TR_STORAGE_BACKEND=${STORAGE_BACKEND}` (`spanner-clickhouse` per README).
- Secrets come from **GCP Secret Manager** via `--update-secrets`:
- Mandatory `SECRET_ENVS`: `TR_SENTRY_DSN`, `TR_STRIPE_SECRET_KEY`, `TR_STRIPE_WEBHOOK_SECRET`, `TR_INTERNAL_GATEWAY_TOKEN`, and the async-settle Ed25519 PEM **mounted as a file** at `/secrets/async-settle/ticket.pem`. A missing one fails the deploy.
- About 67 optional ones, added through `add_secret_env_if_exists` only when the secret exists in the inventory; otherwise they are explicitly `--remove-secrets 'd`. These include provider keys (`ANTHROPIC_API_KEY` ... `ZERO_G_API_KEY`), OAuth client ID/secret, Twilio/Telnyx/Veriff, PayPal/Routable/Adyen, AWS keys, `AXIOM_API_TOKEN` and the federation/operator tokens.
- Non-GCP deployments (AWS/Azure) have their own workflows (`deploy-aws-control-plane.yml`, `deploy-azure-control-plane.yml`) and use Workload Identity Federation (`TR_GOOGLE_EXTERNAL_CREDENTIAL_CONFIG` in `scripts/entrypoint.sh`, which refuses key material). See 04-infra-deploy.



Read the top chain as "lowest priority → highest priority". Each later source overrides the one before it.

7. Env var reference

7a. The ones you actually care about locally

ENV VAR	DEFAULT	PURPOSE	REQUIRED LOCALLY?
TR_ENVIRONMENT	local	local / test unlock dev affordances: the x-trustedrouter-user header, the unsigned Stripe webhook, mock checkout, combined surface. Anything else is treated as deployed and fails closed.	No (default is right)
TR_STORAGE_BACKEND	memory	memory (in-process and lost on restart), postgres, spanner / spanner-clickhouse (prod)	No

ENV VAR	DEFAULT	PURPOSE	REQUIRED LOCALLY?
TR_ENABLE_LIVE_PROVIDERS	false	true makes real upstream calls using the keys file / env	No (true only to hit real models)
TR_LOCAL_KEYS_FILE	~/quill_cloud_keys.private	Path for provider keys (see the gotcha above)	No
TR_SERVICE_SURFACE	combined	Process role: combined\ public\ actions\ control\ internal\ observer	No
TR_RATE_LIMIT_ENABLED	true	Per-IP (240/min), per-key (1200/min) and internal (6000/min) limits. Turn it off for load or browser tests (Playwright does).	No
TR_RATE_LIMIT_IP_PER_WINDOW / _KEY_ / _INTERNAL_ / _WINDOW_SECONDS	240 / 1200 / 6000 / 60	Limits	No
TR_POSTGRES_DSN	""	Postgres-wire system of record (Azure, AWS DSQL, Spanner-PG)	Only with postgres backend
TR_GCP_PROJECT_ID / TR_SPANNER_INSTANCE_ID / TR_SPANNER_DATABASE_ID	quill-cloud-proxy / None / None	Spanner target	Only with spanner backends (never point at prod)
TR_INTERNAL_GATEWAY_TOKEN	None	Protects /v1/internal/gateway/* (authorize/settle/refund)	No. Required outside local.
TR_BOOTSTRAP_MANAGEMENT_KEY	None	Seeds a management API key at startup	No
TR_STRIPE_SECRET_KEY / TR_STRIPE_WEBHOOK_SECRET	None	Real Stripe. Unset means mock checkout and an unsigned local webhook.	No
TR_GOOGLE_CLIENT_ID / _SECRET , TR_GITHUB_CLIENT_ID / _SECRET	None	OAuth sign-in. Each is independently optional.	No (use the header instead)
TR_SENTRY_DSN	None	Error reporting. Disabled in local unless TR_SENTRY_LOCAL_ENABLED.	No
TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS	300000	Starter credit on OAuth/email signup	No
TR_EMAIL_SIGNUP_ENABLED	false	Plain-email POST /v1/signup (credit-farming vector, so off by default)	No
TR_ALLOW_LOCAL_KEY_FILE_IN_TESTS	unset	1 lets pytest read the keys file	No
TR_CONFORMANCE_POSTGRES_DSN	unset	Enables the postgres conformance params	Only for conformance
TR_CONFORMANCE_SPANNER_PG_DSN	unset	Enables the spanner-pg conformance params	Only for conformance
TR_SMOKE_BASE_URL , TR_SMOKE_USER , TR_SMOKE_INTERNAL_TOKEN	http://127.0.0.1:18080/v1 , smoke@example.com , —	scripts/smoke_e2e.py inputs	No
PORT , WEB_CONCURRENCY / UVICORN_WORKERS , UVICORN_*	—	trusted_router.serve (container endpoint)	No (use uvicorn directly)
Provider keys ANTHROPIC_API_KEY , OPENAI_API_KEY , GEMINI_API_KEY , ...	—	Upstream auth (no TR_ prefix)	Only with live providers

`.env.example` (18 lines) contents, annotated: `TR_ENVIRONMENT=local`, `TR_ENABLE_LIVE_PROVIDERS=false`, `TR_LOCAL_KEYS_FILE=/Users/jperla/...` (⚠️ change it or drop it), `TR_STORAGE_BACKEND=memory`, and Spanner/Bigtable IDs (inert with `memory`; `TR_BIGTABLE_*` are **not** Settings fields any more, so `extra="ignore"` drops them). The rest are empty secrets plus the rate-limit numbers. ✅ Loading it verbatim gives `local memory False ... True combined` and `import trusted_router.main` succeeds.

7b. Direct `os.environ` reads outside Settings

Most are in ops scripts, the synthetic monitor and deploy tooling, not in the request path: * **Storage bootstrap scripts:** `TR_STORAGE_BACKEND`, `TR_GCP_PROJECT_ID`, `TR_SPANNER_INSTANCE_ID`, `TR_SPANNER_DATABASE_ID` (20+ reads each in `scripts/`), `TR_POSTGRES_DSN`, `TR_SPANNER_POOL_SIZE`, `TR_SCHEMA_*`. * **Synthetic monitor** (`synthetic/cli.py`, `image_generation.py`, `video_generation.py`): `TR_SYNTHETIC_*`, `TR_MONITOR_API_KEY`, `TR_EMBEDDINGS_*`. * **Deploy mutex/outcome:** `TR_DEPLOY_*`. * **Smoke:** `TR_SMOKE_*`, `TR_API_BASE`, `TR_API_KEY`. * `serve.py`: `PORT`, `WEB_CONCURRENCY`, `UVICORN_*`. `entrypoint.sh`: `TR_GOOGLE_EXTERNAL_CREDENTIAL_CONFIG`. * Provider keys via `providers.py::_secret` (keys file, then env), plus `DATABRICKS_HOST/TOKEN`, `VERTEX_PROJECT_ID`, etc. * `CLOUDSDK_AUTH_CREDENTIAL_FILE_OVERRIDE`, `GITHUB_TOKEN` and `GITHUB_STEP_SUMMARY` in CI/ops scripts.

7c. Full Settings reference (all 354 fields, auto-generated)

Generated by introspecting `Settings.model_fields` and taking the first sentence of the comment directly above each field in `config.py`. Secret-looking fields never show values: they read `None` / `" "` or `"(non-empty code default)"`. "Keys-file fallback = yes" means the field can come from `~/quill_cloud_keys.private` under its un-prefixed upper-case name. Grouping is by name prefix, so it is approximate. **Required for local: none of them.** Every field has a default, and the defaults target `local` + `memory`. Production requirements are enforced by the validators (`production_is_fail_closed`, `operational_analytics_problems`, ...) and supplied by `rollout.sh` + Secret Manager.

Core runtime & surfaces (37)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
<code>TR_ENVIRONMENT</code>	<code>"local"</code>		
<code>TR_RELEASE</code>	<code>"local"</code>		
<code>TR_SOURCE_REVISION</code>	<code>" "</code>		
<code>TR_SERVICE_NAME</code>	<code>"trusted-router"</code>		
<code>TR_SERVICE_SURFACE</code>	<code>"combined"</code>	One image serves several deliberately disjoint process roles.	
<code>TR_ALLOW_DEPLOYED_COMBINED_SURFACE</code>	<code>False</code>	Temporary compatibility bridge for the pre-split GCP service.	
<code>TR_API_BASE_URL</code>	<code>"https://api.trustedrouter.com/v1"</code>		
<code>TR_TRUSTED_DOMAIN</code>	<code>"trustedrouter.com"</code>		
<code>TR_TRUSTED_DOMAIN_ALIASES</code>	<code>"allyrouter.com,uptimerouter.com"</code>	Additional first-party control-plane domains.	
<code>TR_SECURITY_CONTACT_EMAIL</code>	<code>"security@trustedrouter.com"</code>		
<code>TR_SUPPORT_EMAIL</code>	<code>"help@trustedrouter.com"</code>		
<code>TR_ENABLE_LIVE_PROVIDERS</code>	<code>False</code>		
<code>TR_LOCAL_KEYS_FILE</code>	<code>"~/quill_cloud_keys.private"</code>		

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_PUBLIC_CLIENT_OBSERVED_ENABLED	False	Client-observed figures include upstream provider-caused failures and can understate a healthy gateway.	
TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS	300000	Starter credit granted exactly once with a new email/social OAuth account's first workspace.	
TR_SIWE_DOMAIN	None	MetaMask uses public-key crypto (no shared secret).	yes
TR_RETIREMENT_NOTICES_MODE	"off"	Daily route-retirement notices.	
TR_RETIREMENT_NOTICES_PREVIEW_EMAIL	None		
TR_STABLECOIN_CHECKOUT_ENABLED	True		
TR_USER_MODELS_DISPATCH_ENABLED	False	Whether the gateway will AUTHORIZE requests against user-provided models.	
TR_SPECULATIVE_PROVIDER_SHA256_ENABLED	False	PR3 observes only;	
TR_SPEND_LEASE_ACCEPTED_GCP_IMAGE_DIGESTS	" "	Audited break-glass addition to the signed Stage D runtime policy.	
TR_SPEND_LEASE_TRUST_ELIGIBILITY_ENABLED	False	Arms the authorize-time billing-pause gate (trust_eligibility.billing_paused_tx).	
TR_MAX_WORKSPACES_PER_OWNER	25		
TR_OPERATOR_TOKEN	" "		
TR_OPERATOR_IDENTITY	" "		
TR_REAP_SNAPSHOT_BOOKING_ENABLED	False	Decision 70's billing change.	
TR_READ_ONLY	False	Operational read-only flag.	
TR_REGIONS	"us-central1,us-east4,europe-west4,us-west1"	Attested gateway regions.	
TR_PRIMARY_REGION	"us-central1"		
TR_SERVING_REGION	" "	TR_SERVING_REGION identifies this process's deployment, not the primary Spanner/journal region.	
TR_REGIONAL_API_HOSTNAME_TEMPLATE	"api-{region}.quillrouter.com"		
TR_USER_CREDIT_TRANSFER_DAILY_CAP_MICRODOLLARS	100000000	Same-plane user-to-user transfers are final.	
TR_MAX_REQUEST_BODY_BYTES	4194304		
TR_MAX_IN_FLIGHT_REQUEST_BODY_BYTES	67108864		
TR_MAX_CONCURRENT_REQUEST_BODIES	16		
TR_REQUEST_BODY_READ_TIMEOUT_SECONDS	30.0		

Storage backends (Spanner/Postgres/ClickHouse) (25)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_STORAGE_BACKEND	"memory"		
TR_POSTGRES_DSN	" "	Postgres-wire system of record for the non-GCP deployments.	
TR_POSTGRES_IAM_AUTH	" "	Aurora DSQL passwords are short-lived IAM tokens.	
TR_POSTGRES_IAM_REGION	" "	Normally inferred from <id>.dsql.<region>.on.aws.	
TR_GCP_PROJECT_ID	"quill-cloud-proxy"		
TR_SPANNER_INSTANCE_ID	None		
TR_SPANNER_DATABASE_ID	None		
TR_GENERATION_RECORDS_ENABLED	False		
TR_CLICKHOUSE_URL	" "	Legacy in-process ClickHouse mirror.	
TR_CLICKHOUSE_USER	" "		
TR_CLICKHOUSE_PASSWORD	" "		
TR_CLICKHOUSE_BENCHMARK_TABLE	"provider_benchmark_samples"		
TR_PROVIDER_ANALYTICS_CLICKHOUSE_URL	" "	Private, read-only provider portal connection.	
TR_PROVIDER_ANALYTICS_CLICKHOUSE_USER	"tr_provider_read"		
TR_PROVIDER_ANALYTICS_CLICKHOUSE_PASSWORD	" "		
TR_PROVIDER_ANALYTICS_CLICKHOUSE_DATABASE	"tr"		
TR_PROVIDER_ANALYTICS_CLICKHOUSE_TABLE	"provider_benchmark_samples"		
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_URL	" "		
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_USER	"tr_control_read"		
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_PASSWORD	" "		
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_DATABASE	"tr"		
TR_OPERATIONAL_ANALYTICS_OUTBOX_ENABLED	False	Durable tenant-activity and synthetic-status stream.	
TR_OPERATIONAL_ANALYTICS_SINK	"outbox"	HOW operational telemetry leaves the process.	
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_WRITE_USER	"tr"		
TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_WRITE_PASSWORD	" "		

Security, auth & OAuth (49)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SOCIAL_REWARDS_ENABLED	False	Enabled explicitly in the primary deployment configuration.	
TR_SOCIAL_REWARDS_PLANE_ID	" "		
TR_SOCIAL_REWARDS_ISSUER_ID	"gcp-primary"		
TR_SOCIAL_REWARDS_YOUTUBE_URL	" "	YouTube stays hidden and unclaimable until an official channel is configured.	
TR_SOCIAL_REWARDS_CLAIMS_PER_MINUTE	30	Shared account request throttle;	
TR_GOOGLE_DATA_MANAGER_ENABLED	False	Metadata-only Google Ads conversion delivery.	
TR_GOOGLE_DATA_MANAGER_ACCOUNT_ID	None		
TR_GOOGLE_DATA_MANAGER_LOGIN_ACCOUNT_ID	None		
TR_GOOGLE_DATA_MANAGER_SIGNUP_ACTION_ID	None		
TR_GOOGLE_DATA_MANAGER_ACTIVATED_ACTION_ID	None		
TR_GOOGLE_DATA_MANAGER_PURCHASE_ACTION_ID	None		
TR_GOOGLE_DATA_MANAGER_KMS_KEY_NAME	None		
TR_GOOGLE_DATA_MANAGER_BATCH_SIZE	500		
TR_GOOGLE_DATA_MANAGER_LEASE_SECONDS	300		
TR_GOOGLE_DATA_MANAGER_MAX_ATTEMPTS	20		
TR_GOOGLE_DATA_MANAGER_TIMEOUT_SECONDS	20.0		
TR_GOOGLE_DATA_MANAGER_REPAIR_LOOKBACK_DAYS	90		
TR_GOOGLE_DATA_MANAGER_STATUSES_POLL_ATTEMPTS	6		
TR_GOOGLE_DATA_MANAGER_STATUSES_POLL_SECONDS	2.0		
TR_ATTRIBUTION_COOKIE_KEY	None	Split public/control services may hold the base64-encoded, already-derived cookie key without receiving the legacy root that also authori...	yes
TR_ATTRIBUTION_COOKIE_SECRET	None		yes
TR_INTERNAL_GATEWAY_TOKEN	None		yes
TR_BOOTSTRAP_MANAGEMENT_KEY	None		yes
TR_BYOK_KMS_KEY_NAME	None		yes
TR_BYOK_ENVELOPE_KEY_B64	None		yes

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_BYOK_ENVELOPE_KEY_REF	"trustedrouter/byok-envelope-key/v1"		yes
TR_BYOK_REGISTRATION_ENABLED	True	BYOK key REGISTRATION wraps the secret with the byok-envelope KMS key.	
TR_GOOGLE_CLIENT_ID	None	OAuth providers — independently optional.	yes
TR_GOOGLE_CLIENT_SECRET	None		yes
TR_GOOGLE_OAUTH_REDIRECT_URL	None		yes
TR_GOOGLE_OAUTH_LOGIN_AVAILABLE	None	The public service must render login links without receiving the OAuth client secrets owned by the control service.	
TR_GOOGLE_ALIAS_CREDENTIALS_JSON	"{}"	Backup domains use independent provider credentials so login remains available even if the canonical domain or its OAuth app is unavailable.	yes
TR_GITHUB_CLIENT_ID	None		yes
TR_GITHUB_CLIENT_SECRET	None		yes
TR_GITHUB_OAUTH_REDIRECT_URL	None		yes
TR_GITHUB_OAUTH_LOGIN_AVAILABLE	None		
TR_GITHUB_ALIAS_CREDENTIALS_JSON	"{}"	GitHub OAuth Apps permit exactly one callback URL.	yes
TR_FEDERATION_PEER_TOKEN	" "	--- Lazy API-key federation -----HOME side: the token peers present to /internal/federation/resolv...	
TR_FEDERATION_HOME_BASE_URL	" "	PEER side: the home plane this deployment resolves unknown keys from, and the token it presents.	
TR_FEDERATION_HOME_TOKEN	" "		
TR_FEDERATION_CREDIT_INBOUND_TOKEN	" "	INBOUND: what this plane requires to be credited.	
TR_FEDERATION_CREDIT_PEER_BASE_URL	" "	OUTBOUND: the plane this deployment may push credits TO, and the token it presents there.	
TR_FEDERATION_CREDIT_PEER_TOKEN	" "		
TR_FEDERATION_DEFERRED_SETTLEMENT_ENABLED	False	OFF by default.	
TR_FEDERATION_DEFERRED_MAX_OUTSTANDING_MICRODOLLARS	25000000	The bound on how much unsettled debt one workspace may run up on this plane.	
TR_FEDERATION_DEFERRED_AUTHORIZATION_TTL_SECONDS	7200	How long a deferred authorization may sit unsettled before the reaper reclaims its estimate and releases the key-limit escrow.	
TR_FEDERATION_SETTLEMENT_INBOUND_TOKENS	" "	HOME side: which peers may apply usage to this plane's ledger, as "plane=token,plane=token".	
TR_FEDERATION_SETTLEMENT_HOME_TOKEN	" "	PEER side: the token this plane presents to the home plane's apply-usage endpoint.	
TR_FEDERATION_SETTLEMENT_WORKSPACE_DAILY_CAP_MICRODOLLARS	100000000	HOME's aggregate clamp: maximum usage one peer may apply to one workspace per UTC day.	

Rate limiting & admission (7)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_RATE_LIMIT_ENABLE	True		
TR_RATE_LIMIT_WINDOW_SECONDS	60		
TR_RATE_LIMIT_IP_PER_WINDOW	240		
TR_RATE_LIMIT_KEY_PER_WINDOW	1200		
TR_RATE_LIMIT_INTERVAL_PER_WINDOW	6000		
TR_SETTLE_PER_KEY_LIGHT_LIMIT	16	Settlement has a wider gate because it closes already-issued holds, but one hot key must not consume every Spanner session in an instance.	
TR_RATE_LIMIT_CLIENT_IP_MODE	"untrusted"	Only front doors that overwrite X-TrustedRouter-Client-IP may opt into edge_header.	

Billing & payments (Stripe/PayPal/Adyen/Routable/x402/Veriff) (49)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_STRIPE_WEBHOOK_SECRET	None		yes
TR_STRIPE_SECRET_KEY	None		yes
TR_STRIPE_CARD_FEE_BASIS_POINTS	290	Standard US Stripe processing schedules.	
TR_STRIPE_CARD_FEE_FIXED_CENTS	30		
TR_STRIPE_STABLECOIN_FEE_BASIS_POINTS	150		
TR_STRIPE_STABLECOIN_FEE_FIXED_CENTS	0		
TR_STRIPE_ACH_FEE_BASIS_POINTS	80		
TR_STRIPE_ACH_FEE_FIXED_CENTS	0		
TR_STRIPE_ACH_FEE_MAX_CENTS	500		
TR_PAYPAL_CLIENT_ID	None		yes
TR_PAYPAL_CLIENT_SECRET	None		yes
TR_PAYPAL_WEBHOOK_ID	None		yes
TR_PAYPAL_API_BASE_URL	"https://api-m.paypal.com"		yes
TR_ROUTABLE_ENABLED	False	Creator USD cash-outs.	yes
TR_ROUTABLE_API_TOKEN	None		yes
TR_ROUTABLE_WEBHOOK_SECRET	None		yes
TR_ROUTABLE_COMPANY_ID	None		yes

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_ROUTABLE_TEAM_MEMBER_ID	None		yes
TR_ROUTABLE_WITHDRAW_FROM_ACCOUNT_ID	None		yes
TR_ROUTABLE_API_BASE_URL	"https://api.routable.com"		yes
TR_ADYEN_ENABLED	False	Adyen is staged dark until the test merchant, HMAC webhook, and end-to-end checkout canary are all green.	
TR_ADYEN_API_KEY	None		yes
TR_ADYEN_CLIENT_KEY	None		yes
TR_ADYEN_HMAC_KEY	None		yes
TR_ADYEN_REFERENCE_KEY	None		yes
TR_ADYEN_MERCHANT_ACCOUNT	None		yes
TR_ADYEN_ENVIRONMENT	"test"		
TR_ADYEN_LIVE_ENDPOINT_PREFIX	None		yes
TR_ADYEN_CHECKOUT_API_VERSION	72		
TR_ADYEN_WEB_VERSION	"6.41.0"		
TR_ADYEN_CARD_FEE_BASIS_POINTS	0	Keep processor pricing explicit.	
TR_ADYEN_CARD_FEE_FIXED_CENTS	0		
TR_X402_ENABLED	False		
TR_X402_ALLOW MOCK_PAYMENTS	False		
TR_X402_NETWORK	"base"		
TR_X402_NETWORK_ID	"eip155:8453"		
TR_X402_STRIPE_API_VERSION	"2026-03-04.preview"		
TR_X402_MAX_FUND_DOLLARS	"500"		
TR_X402_RATE_LIMIT_WINDOW_SECONDS	60		
TR_X402_RATE_LIMIT_KEY_PERIOD_WINDOW	10		
TR_X402_RATE_LIMIT_WORKSPACE_PER_WINDOW	30		
TR_X402_SETTLE_RATE_LIMIT_PER_WINDOW	30		
TR_X402_SETTLE_WORKSPACE_PER_WINDOW	120		
TR_VERIFF_ENABLED	False		yes
TR_VERIFF_API_KEY	None		yes

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_VERIFF_SHARED_SECRET_KEY	None		yes
TR_VERIFF_BASE_URL	"https://stationapi.veriff.com"		yes
TR_TRUST_STRIPE_ACCOUNT_ID	" "	Same account pin used by the deployed backfill and reconciliation jobs.	
TR_LIGHTNING_FUNDING_TOKEN	" "		

Money path: settle/outbox/async settle/fastpath (50)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SPECULATION_SHADOW_WORKSPACES	[]		
TR_SPECULATION_SHADOW_ROUTES	[]		
TR_SPECULATION_SHADOW_IMAGES	[]		
TR_SPECULATION_SHADOW_PRODUCERS	[]		
TR_SPECULATION_SHADOW_PRODUCER	" "		
TR_SPECULATION_SHADOW_SLOTS	{{}}		
TR_SPECULATION_SHADOW_IMAGE_POLICY_VERSION	0		
TR_SPECULATION_SHADOW_POLICY_EXPIRES_AT	0		
TR_SPECULATION_SHADOW_PLANE	"gcp"		
TR_SPECULATION_SHADOW_ISSUER	" "		
TR_SPECULATION_SHADOW_AUDIENCE	" "		
TR_SPECULATION_SHADOW_KEY_ID	" "		
TR_SPECULATION_SHADOW_PRIVATE_KEY_FILE	" "	Independently provisioned mounted file.	
TR_ASYNC_SETTLE_SHADOW_ATTACHMENT_BUCKET	" "	Dormant async v1 authorize metadata;	
TR_ASYNC_SETTLE_SHADOW_POLICY	" "		
TR_ASYNC_SETTLE_SHADOW_HEALTH_REGION	" "		
TR_ASYNC_SETTLE_SHADOW_WORKSPACES	" "		

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_ASYNC_SETTLE_PILOT_WORKSPACES	" "		
TR_ASYNC_SETTLE_ENABLED	False		
TR_ASYNC_SETTLE_PROTECTION	False		
TR_ASYNC_SETTLE_TICKET_KEY_ID	" "		
TR_ASYNC_SETTLE_TICKET_ISSUER	" "		
TR_ASYNC_SETTLE_TICKET_AUDIENCE	" "		
TR_ASYNC_SETTLE_TICKET_PRIVATE_KEY_FILE	" "	Independently mounted purpose key.	
TR_ASYNC_SETTLE_PILOT_CAMP_MICRO	0		
TR_ASYNC_SETTLE_TICKET_TIMEOUT_SECONDS	300		
TR_ASYNC_SETTLE_AUTHORITY_EPOCH	0	Operator-owned local settlement epoch, NOT the billing pause epoch or shadow grant generation.	
TR_STAGE_D_HEARTBEAT_ENABLED	True	Stage D is inert until an attested enclave calls the new endpoint.	
TR_STAGE_D_REQUIRE_HEARTBEAT_DECLARATION	False	Off until every accepted enclave image declares its heartbeats at boot registration.	
TR_STAGE_D_ELIGIBILITY_ENABLED	False	Emergency kill added 2026-09-03: declares no request Stage D eligible.	
TR_STAGE_D_PILOT_WORKSPACE_IDS	"45819281-0ce9-4811-a0cd-c660ab3a116d"		
TR_FASTPATH_PILOT_ENABLED	False	The fast path's pilot (docs/design/fast-admission-pilot.md): an authorize for a workspace in the set, from a control plane in one of the ...	
TR_FASTPATH_PILOT_WORKSPACES	" "		
TR_FASTPATH_PILOT_REGIONS	"us-central1"		
TR_FASTPATH_FRONT_DOOR_TIMEOUT_SECONDS	1.5	The front door answers an admitted authorize in a few milliseconds;	
TR_STAGE_D_POLICY_REFRESH_SECONDS	60		
TR_STAGE_D_POLICY_CERT_IDENTITY	"https://github.com/Lore-Hex/quill-cloud-prox...		
TR_STAGE_D_POLICY_OIDC_ISSUER	"https://token.actions.githubusercontent.com"		
TR_SETTLE_OUTBOX_FAST_DRAIN_ENABLED	False	Durable settle outbox (docs/design/durable-settle-outbox.md).	
TR_SETTLE_OUTBOX_POLL_INTERVAL_SECONDS	300		

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SETTLE_OUTBOX_HEALTH_PUBLISH_INTERVAL_SECONDS	2		
TR_SETTLE_OUTBOX_CLAIM_BATCH	500		
TR_SETTLE_OUTBOX_WORKER_CONCURRENCY	1		
TR_SETTLE_OUTBOX_LEASE_SECONDS	300		
TR_SETTLE_OUTBOX_PASS_BUDGET_SECONDS	240		
TR_SETTLE_OUTBOX_ENABLED	False		
TR_STAGE_D_PROBE_API_KEY	" "	Second internal probe credential, used by the enclave rollout gate for the Stage D cohort.	
TR_ASYNC_SETTLE_SHADOW_PROBE_API_KEY	" "		yes
TR_ASYNC_SETTLE_SHADOW_PROBE_REPETITIONS	1		
TR_ASYNC_SETTLE_SHADOW_PROBE_CADENCE_MINUTES	4		

Trust / attestation (24)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_TRUST_GCP_SOURCE_COMMIT	None		
TR_TRUST_GCP_IMAGE_REFERENCE	None		
TR_TRUST_GCP_IMAGE_DIGEST	None		
TR_TRUST_GCP_RELEASE_URL	" "	Production control-plane trust aliases resolve the current gateway release from this canonical, independently published record.	
TR_TRUST_GCP_RELEASE_FALLBACK_URLS	" "	Independently hosted copies keep the backup-domain trust pages live when the canonical DNS zone is unavailable.	
TR_TRUST_AWS_SOURCE_COMMIT	None	Both are SETS, not scalars.	
TR_TRUST_AWS_IMAGE_REFERENCE	None		
TR_TRUST_AWS_PCR0	None		
TR_TRUST_AWS_ACCEPTED_PCR0S	" "		
TR_TRUST_AZURE_SOURCE_COMMIT	None		
TR_TRUST_AZURE_IMAGE_REFERENCE	None		
TR_TRUST_AZURE_HOSTDATA	None		

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_TRUST_AZURE_ACCEPTED_HOSTDATA	" "		
TR_TRUST_AZURE_ATTESTATION_ISSUERS	" "	Azure serves from more than one region and each region has its own MAA instance, so the issuer a verifier sees depends on which region an...	
TR_TRUST_AWS_RELEASE_URL	" "	Where each plane's AUTHORITATIVE record lives.	
TR_TRUST_AZURE_RELEASE_URL	" "		
TR_TRUST_QUALIFYING_PROVIDERS	"stripe, x402"		
TR_TRUST_TIER3_MIN_DAYS	30		
TR_TRUST_TIER3_MIN_PAID_MICRODOLLARS	50000000		
TR_TRUST_TIER_JOB_CURRENCY	1	Workspaces the trust-tier job recomputes at once.	
TR_TRUST_TIER_SHADOW_ENABLED	True	Shadow the per-workspace pass with one bulk snapshot (trust_tier_bulk.py) and log any workspace it acted on that the bulk selection missed.	
TR_TRUST_RECONCILE_INTERVAL_SECONDS	900		
TR_TRUST_RECONCILE_MAX_AGE_SECONDS	3600		
TR_ATTESTATION_EXPECTED_PCR0	None	Pin the enclave's PCR0 (EIF measurement).	

Observability (Sentry/Axiom/synthetic/remediator) (45)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_CLIENT_EVENTS_ENABLED	False	Client-observed reliability beacons remain off until the ClickHouse node runbook has been completed.	
TR_CLIENT_EVENTS_SUCCESS_SAMPLE_RATE	0.01	Sampling policy returned to SDKs for ordinary successful requests.	
TR_CLIENT_EVENTS_PAUSE_SECONDS	0	A positive value pauses SDK delivery without reading or storing a body.	
TR_CLIENT_EVENTS_SYNTHETIC_WORKSPACE_IDS	[]	Workspaces whose beacons are synthetic regardless of the client bit.	
TR_CLIENT_EVENTS_MAX_BODY_BYTES	65536	Hard request and abuse-control bounds for the fire-and-forget endpoint.	
TR_CLIENT_EVENTS_KEY_PERIOD_MINUTE	60		
TR_CLIENT_EVENTS_WORKSPACE_PER_MINUTE	300		
TR_CLIENT_EVENTS_WRITE_CONCURRENCY	4	Bounds blocking outbox writes so telemetry cannot consume the money path's pool.	
TR_CLIENT_EVENTS_FLUSH_SECONDS	30	Flush cadence returned to SDKs in the accepted policy.	
TR_SENTRY_DSN	None		yes

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SENTRY_TRACES_SAMPLE_RATE	0.05		
TR_SENTRY_LOCAL_ENABLED	False		
TR_AXIOM_DATASET	"trusted-router-logs"	Axiom log shipping.	
TR_AXIOM_URL	"https://api.axiom.co"		
TR_AXIOM_LOG_LEVEL	"INFO"	Levels at and above this go to Axiom.	
TR_SENTRY_FLOODGATE_ENABLED	True		
TR_SENTRY_FLOODGATE_WINDOW_SECONDS	3600		
TR_SENTRY_FLOODGATE_MAX_EVENTS_PER_FINGERPRINT	3		
TR_SENTRY_FLOODGATE_MAX_EVENTS_PER_WINDOW	50		
TR_SENTRY_FLOODGATE_MAX_FINGERPRINTS	2048		
TR_SYNTHETIC_MONITOR_REGION	None		
TR_SYNTHETIC_MONITOR_API_KEY	None		yes
TR_SYNTHETIC_MONITOR_MODEL	"trustedrouter/monitor"		
TR_SYNTHETIC_CONTROL_PLANE_BASE_URL	None	Exact HTTPS control-plane origin synthetic canaries exercise.	
TR_SYNTHETIC_CANONICAL_ATTESTED	False	The canonical target serves a self-signed cert minted inside the TEE (AWS Nitro standalone deployments): probes skip CA verification and ...	
TR_SYNTHETIC_SCHEDULER_INTERVAL_SECONDS	0	0 = disabled (the default, so GCP/AWS keep their existing schedulers and nothing double-runs).	
TR_SYNTHETIC_SCHEDULER_ROTATION_COUNT	8	Completions per pass.	
TR_SYNTHETIC_REGIONAL_PROBES_ENABLED	True	Per-region alias/Cloud-Run probing is a GCP-topology concept: it derives api-{region}.quillrouter.com aliases and *.run.app direct URLs f...	
TR_SYNTHETIC_IMAGE_PROBE_ENABLED	True	Image generation is sampled by a SEPARATE scheduled job (-m trusted_router.synthetic.image_generation), not by the 1-minute synthetic run.	
TR_SYNTHETIC_STATUS_PROBE_TYPES	" "	Public readers describe the monitor plane, not their own credentials.	
TR_SYNTHETIC_CONTROL_PLANE_HEALTH_URL	None	Explicit control-plane /health URL attached to the canonical target.	
TR_SYNTHETIC_GATEWAY_REGION_TARGETS	" "	Empty = today's behaviour, exactly one canonical target.	
TR_SYNTHETIC_MONITOR_TIMEOUT_SECONDS	20.0	Per-probe HTTP timeout for real provider-effective synthetic checks.	
TR_SYNTHETIC_RUN_DEADLINE_SECONDS	240.0	Hard wall-clock ceiling for one admitted synthetic run.	

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SYNTHETIC_REMEDIATOR_DEADLINE_SECONDS	15.0	Remediation is an observe-only monitor read running through an internal HTTP surface during the service split.	
TR_SYNTHETIC_MONITOR_MONTHLY_GRANT_DOLLARS	200.0	Monthly self-funding for the monitor workspace, applied lazily on its own gateway-authorize path (synthetic/funding.py).	
TR_SYNTHETIC_STATUS_SAMPLE_LIMIT	5000		
TR_SYNTHETIC_STATUS_RAW_RETENTION_DAYS	14		
TR_SYNTHETIC_STATUS_ROLUP_RETENTION_MONTHS	24		
TR_SYNTHETIC_STATUS_US_URL	"https://status-us.trustedrouter.com/status.j..."		
TR_SYNTHETIC_STATUS_EU_URL	"https://status-eu.trustedrouter.com/status.j..."		
TR_REMEDIATOR_MODE	"observe"	Fleet peer list ("name=base_url,..."): every deployment watches every deployment's public status page — its own included, which proves th...	
TR_REMEDIATOR_INTERVAL_SECONDS	120		
TR_REMEDIATOR_IN_PROCESSES_ENABLED	True	Cloud Run request-based CPU may pause background coroutines between requests.	
TR_SYNTHETIC_FLEET_PEERS	"gcp=https://trustedrouter.com,aws=https://aw..."		

Notifications & email (SES/Twilio/Telnyx/ops chat) (33)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_OPS_CHAT_WEBHOOK_URLS	" "	Optional three-cloud Matrix support fanout.	
TR_OPS_CHAT_WEBHOOK_SECRET	None		
TR_OPS_CHAT_WEBHOOK_TIMEOUT_SECONDS	3.0		
TR_AWS_ACCESS_KEY_ID	None	Amazon SES — used for optional wallet email attach/verification.	yes
TR_AWS_SECRET_ACCESS_KEY	None		yes
TR_AWS_REGION	"us-east-1"		yes
TR_SES_FROM_EMAIL	None		yes
TR_SES_FROM_NAME	"TrustedRouter"		yes
TR_SES_CONFIGURATION_SET	"trustedrouter-default"	Configuration set used on every SendEmail call so SES emits bounce + complaint events to our SNS topic (subscribed at /internal/ses/notif...	
TR_SES_ALERT_FROM_EMAIL	"alerts@alerts.trustedrouter.com"	Operational alerts use a dedicated authenticated subdomain and configuration set.	yes
TR_SES_ALERT_FROM_NAME	"TrustedRouter Alerts"		yes

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_SES_ALERT_CONFIGURATION_SET	"trustedrouter-alerts"		yes
TR_SES_NOTICES_FROM_EMAIL	"notices@notices.trustedrouter.com"	Customer product notices (route retirements) send from their own authenticated subdomain and configuration set, separate from sign-in mai...	yes
TR_SES_NOTICES_FROM_NAME	"TrustedRouter"		yes
TR_SES_NOTICES_CONFIGURATION_SET	"trustedrouter-notices"		yes
TR_NOTIFY_ENABLED	False	----notifications (email / sms / voice to the account owner) -----Delivery credentials.	
TR_NOTIFY_SMS_AVAILABLE	False	Flip to True once A2P 10DLC is approved on the primary SMS carrier;	
TR_TELNYX_API_KEY	None		
TR_TELNYX_FROM_NUMBER	None		
TR_NOTIFY_VOICE_REPEAT_UNANSWERED	True	The ORGANIZATION id from Telnyx /v2/whoami — not the number's connection id and not the TeXML application id, both of which answer 404 here.	
TR_NOTIFY_SMS_PRIMARY_CARRIER	"twilio"		
TR_NOTIFY_VOICE_PRIMARY_CARRIER	"telnyx"		
TR_TELNYX_TEXML_ACCOUNT_ID	None		
TR_TELNYX_TEXML_APPLICATION_ID	None	The TeXML application, which is where the outbound voice profile hangs.	
TR_TWILIO_ACCOUNT_ID	None		
TR_TWILIO_AUTH_TOKEN	None		
TR_TWILIO_API_KEY_ID	None		
TR_TWILIO_API_KEY_SECRET	None		
TR_TWILIO_FROM_NUMBER	None		
TR_NOTIFY_PRICE_MICRODOLLARS	20000	\$0.02 is positive on every path, thinnest on a Twilio VOICE fallback (~\$0.006 margin on a one-minute minimum).	
TR_NOTIFY_PUSH_PRICE_MICRODOLLARS	0	It is also the only channel with no carrier and no regulator in the path, which makes it the one we can offer on day one without 10DLC or...	
TR_NOTIFY_MAX_PER_HOUR	30	Per-workspace ceilings.	
TR_NOTIFY_MAX_VOICE_PER_HOUR	4		

Legal entity metadata (14)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_LEGAL_ENTITY_NAME	"Lore Hex Corp"		
TR_LEGAL_ENTITY_TYPE	"Delaware C Corporation"		
TR_LEGAL_ENTITY_ADDRESS	"1111 Brickell Ave, Floor 10, Miami, FL 33131"		
TR_LEGAL_ENTITY_STREET	"1111 Brickell Ave, Floor 10"	Structured components of the same address.	
TR_LEGAL_ENTITY_CITY	"Miami"		
TR_LEGAL_ENTITY_REGION	"FL"		
TR_LEGAL_ENTITY_POSTAL_CODE	"33131"		
TR_LEGAL_ENTITY_COUNTRY	"US"		
TR_LEGAL_ENTITY_PHONE	"+1-305-239-7350"		
TR_LEGAL_ENTITY_EIN	"41-5339728"		
TR_LEGAL_ENTITY_DUNS	"144992055"		
TR_LEGAL_ENTITY_DATE_ESTABLISHED	None		
TR_LEGAL_SIGNATORY_NAME	"Joseph Perla"		
TR_LEGAL_SIGNATORY_TITLE	"CEO"		

Other (21)

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_ANALYTICS_OUTBOX_ENABLED	False	Stage 1 live analytics outbox.	
TR_NEW_SIGNUPS_ENABLED	True	Global, creation-only operational brake.	
TR_EMAIL_SIGNUP_ENABLED	False	Plain-email POST /v1/signup (no OAuth) is a credit-farming vector: an open endpoint that mints a management key + trial credit for ANY ...	
TR_ENABLE_SENTRY_TEST_ROUTE	False		
TR_GATEWAY_AUTHORIZE_MAX_IN_FLIGHT_PER_KEY	4	A contended Spanner authorize can occupy a worker for the full RPC budget.	
TR_OBSERVER_INTERNAL_TOKEN	None	Dedicated to the externally reachable observer/status processes.	
TR_CHECKOUT_CARD_FEE_MINIMUM_CENTS	80	Card-style rails have a purchase-fee floor.	
TR_AUTH_SESSION_TTL_SECONDS	2592000		
TR_OAUTH_AUTHORIZATION_CODE_TTL_SECONDS	600		
TR_PARTNER_INQUIRY_EMAIL	None	Destination for TrustedOS partner-inquiry form submissions (/trustedos).	

ENV VAR	DEFAULT	PURPOSE (FROM CODE COMMENT, FIRST SENTENCE)	KEYS-FILE FALLBACK
TR_ACTIVATION_REMINDER_INTERVAL_SECONDS	0	Durable post-signup activation reminders.	
TR_MULTI_REGION_ENABLED	True		
TR_PHONE_VERIFICATION_REQUIRES_FUNDING	None		
TR_CUSTOM_MODELS_REQUIRE_VERIFICATION	None		
TR_IDENTITY_SESSION_STALE_AFTER_DAYS	7		
TR_HEARTBEAT_GRACE_SECONDS	300		
TR_REQUEST_RECORD_WRITE_MODE	"legacy"	Expand/contract switch for per-request Spanner records.	
TR_BROADCAST_INLINE_DRAIN_ENABLED	False	Local/test drains broadcast jobs opportunistically after settlement so tests and demos are deterministic.	
TR_MARKETING_REGIONS	"us-central1,europe-west4,us-east4,us-west1,a..."		
TR_EXTERNAL_LIVE_REGIONS	"aws-eu-west-1,aws-eu-west-3,azure-australiae..."	Non-GCP deployments that have PASSED their own end-to-end smoke (verify_deployment.sh --expect-monitor).	
TR_AUTO_MODEL_ORDER	" "	IDs follow OpenRouter naming exactly to line up with the ingest snapshot — moonshotai/... not kimi/..., mistralai/... not `mistral/...	

8. Troubleshooting (real issues hit during verification)

#	SYMPTOM	CAUSE	FIX
T 1	failed to connect to the docker API at unix:///Users/.../.docker/run/docker.sock ... no such file or directory	Docker CLI installed but the daemon was not running	Start Docker Desktop, OrbStack or colima. You only need it for the postgres/spanner conformance backends (commands in AGENTS.md, ports 5433/5434, TR_CONFORMANCE*_DSN). Not verified on this Mac.
T 2	POST /v1/chat/completions → 402 Insufficient credits on a fresh local workspace	Header-auth users get no trial credit, and the memory store starts at \$0	Credit yourself with the unsigned local Stripe webhook in \$4c
T 3	400 model_not_supported: cerebras/llama3.1-8b	The catalog moved on, and old examples (README smoke script) still use this model	Use trustedrouter/auto or pick an ID from /v1/models
T 4	smoke_e2e.py: expected 54 OpenRouter route classifications, got 95	The script's hard-coded expectations are stale	Use the \$4 curl flow. Fixing the script is a good first PR.
T 5	Provider keys "not found" even though TR_LOCAL_KEYS_FILE points at your file / you copied .env.example	.env.example hard-codes Joseph's path. The Settings fallback always reads the default path anyway (\$6 gotcha).	Put the file at ~/.quill_cloud_keys.private, or export the provider keys in the shell

#	SYMPTOM	CAUSE	FIX
T 6	Changing a <code>TR_*</code> var "does nothing"	Typo (<code>extra="ignore"</code>), or the var is not a Settings field (e.g. <code>TR_BIGTABLE_*</code> in <code>.env.example</code>), or the shell env is overriding <code>.env</code>	<code>uv run python -c "from trusted_router.config import get_settings as g; print(g().<field>)"</code>
T 7	App refuses to start: <code>TR_SERVICE_SURFACE=combined</code> is restricted to <code>local/test</code> ...	You set <code>TR_ENVIRONMENT</code> to something other than <code>local / test</code>	Keep <code>local</code> for dev. Deployed surfaces are configured only in <code>rollout.sh</code> .
T 8	<code>/healthz 404</code>	The route doesn't exist	Use <code>/health</code> or <code>/v1/health</code>
T 9	Local bursts get 429	<code>TR_RATE_LIMIT_ENABLED=true</code> by default (240 req/min per IP)	<code>TR_RATE_LIMIT_ENABLED=false</code> for load or browser testing
T 10	Your keys file is ignored in tests	By design: <code>_running_under_pytest()</code> skips it	<code>TR_ALLOW_LOCAL_KEY_FILE_IN_TESTS=1</code> (rarely what you want)
T 11	CI <code>frontend</code> job fails on <code>git diff --exit-code</code>	You edited <code>frontend/src/*.ts</code> without committing the rebuilt JS	<code>npx tsc</code> , then commit <code>src/trusted_router/static/*.js</code>
T 12	Lots of <code>on_event</code> is deprecated warnings in pytest	FastAPI lifespan deprecation	Harmless
T 13	A test fails on your branch but passes on <code>main</code>	<code>main</code> moves fast (an hourly price-refresh bot commits to it)	Rebase first (AGENTS.md)

9. Safety reminders for local work

- Local dev needs **zero** cloud credentials. If something asks for `gcloud` auth, stop and check `TR_STORAGE_BACKEND`.
- Never point `TR_SPANNER_*` / `TR_POSTGRES_DSN` at production. AGENTS.md: `unindexed tr_entities.body` searches caused the 2026-09-06 CPU incident. Use `python -m scripts.inspect_payment_owner` for Stripe attribution and ClickHouse for analytics.
- Pushing to `main` auto-deploys the control plane when `src/**`, `scripts/deploy/**`, `frontend/src/**`, `Dockerfile`, `pyproject.toml` or `uv.lock` change.
- Money code (`reserve / settle / refund`) changes need a differential or conformance test, never review-by-eye.

10. Verification log (raw timings)

STEP	COMMAND	OUTCOME	WALL TIME
sync	<code>uv sync</code>	✅ 238 resolved	9.8 s (warm cache)
lint	<code>uv run ruff check .</code>	✅	0.55 s
types	<code>uv run mypy</code>	✅ 533 files	9.6 s warm / 8.3 s cold cache
conformance subset	<code>pytest tests/conformance/{store_semantics,key_management} -k memory</code>	✅ 90 passed / 14 skipped	10.2 s
unit subset	<code>pytest tests/test_billing.py tests/test_inference_errors.py</code>	✅ 82 passed	10.4 s
collect	<code>pytest --collect-only -q</code>	✅ 38,221 selected	31.6 s
catalog check	<code>scripts/update_model_changes.py --check</code>	✅ exit 0	—

STEP	COMMAND	OUTCOME	WALL TIME
server	<code>uvicorn ... --port 18080</code> → health, models, keys, webhook credit, chat, stream, activity	✓	about 1 s startup
smoke script	<code>scripts/smoke_e2e.py</code>	⚠ stale assertion (54 vs 95)	—
npm	<code>npm ci, lint, lint:css, typecheck, npx tsc</code> (no diff)	✓	< 1 s each
playwright	<code>npx playwright install chromium + homepage.spec.js</code>	✓ 9 passed	12 s + 5.7 s
docker conformance	postgres / pgadapter	⚠ not run (daemon down)	—

05 The Contribution Loop



Full suite: monkeypatching the STORE proxy once broke 136 later tests. Patch the class.

PART II • CHAPTER 04 • HERO STILL: VIDEO 05 AT 0:43

Contributing & testing

The three CI gates, the test taxonomy, branch and PR conventions, and a first-PR walkthrough.

- 1. The non-negotiables (from AGENTS.md)
- 2. Tool configuration (pyproject.toml)
- 3. CI anatomy (.github/workflows/ci.yml)
- 4. Dev loop
- 5. Test taxonomy
- 6. Branch, commit and PR conventions
- 7. Your first PR: walkthrough
- 8. PR checklist

Storyboard · Video 05: The contribution loop

0:18

0:22

0:43

0:47

0:57

1:00

1:10

1:14

Scope: how code gets from an idea to production in `quill-router`. This covers the three CI gates, the test suite's structure, branch and commit habits, how review works (mostly coding agents), and a first-PR walkthrough with a checklist.

All paths are relative to `quill-router/`. Repo state: `main @ 778120494`, 2026-10-19.

1. The non-negotiables (from AGENTS.md)

`AGENTS.md` is the entry point for every coding agent and human (`CLAUDE.md:3` only points at it). These are its hard rules:

RULE	SOURCE	WHY
CI has three gates . Run all three locally before saying you're done: <code>uv run ruff check . && uv run mypy && uv run pytest -q</code>	<code>AGENTS.md:27-31</code>	CI runs exactly these (<code>.github/workflows/ci.yml:27-30</code> , <code>:259-264</code>)
Never run <code>ruff format .</code>	<code>AGENTS.md:33-34</code>	CI only runs <code>ruff check .</code> Formatting the tree produces a huge unrelated diff
Coverage must stay ≥ 70%	<code>AGENTS.md:35</code>	Enforced once over all combined shards (<code>ci.yml:286-332</code> , <code>--fail-under=70</code> at <code>:332</code>)
<code>mypy</code> is a real gate	<code>AGENTS.md:36</code>	Passing <code>ruff</code> and <code>pytest</code> locally still fails CI if <code>mypy</code> fails
Plain <code>pytest</code> skips <code>proof_oracle</code> and <code>provider_health</code> . If you touch <code>async settle</code> or <code>billing</code> , run the oracle yourself	<code>AGENTS.md:37-40</code> , <code>pyproject.toml:77</code>	<code>uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py</code>
Run the FULL suite , not only your new file	<code>AGENTS.md:42-45</code>	<code>STORE</code> proxy pollution (issue #333)
Rebase before you trust a failure	<code>AGENTS.md:47-49</code>	<code>main</code> moves fast. The hourly price bot commits to it
Money code (<code>reserve</code> / <code>settle</code> / <code>refund</code> / <code>credits</code>) needs a differential or conformance test. Review by eye is not enough	<code>AGENTS.md:81-87</code>	Highest-risk code in the repo
Do not weaken a conformance test to make a backend pass	<code>AGENTS.md:91-93</code>	<code>tests/conformance/</code> is the executable contract for the <code>Store</code> Protocol
Read-only is not load-safe: no <code>LIKE</code> / <code>regex</code> / <code>JSON</code> scans of <code>tr_entities.body</code> in prod	<code>AGENTS.md:70-79</code> , <code>CLAUDE.md:4-6</code>	The 2026-09-06 Spanner CPU incident
Deploy production-bound changes yourself unless told to hold, and verify after	<code>AGENTS.md:53-58</code>	Joseph's default

The STORE pollution trap (issue #333)

`STORE` is a module-level `_StoreProxy` (`src/trusted_router/storage.py:3880`, `:3947`). It forwards calls to a backend that can be swapped. `monkeypatch.setattr(STORE, "x", fake)` creates an instance attribute that shadows `__getattr__`. On teardown, `monkeypatch` "restores" a bound method of the *old* backend, which silently breaks later tests. It once broke 136 unrelated tests (issue #333, "Inject the store instead of importing the `STORE` module global"). **Patch the backend class** (for example `InMemoryStore.create_workspace`), never the proxy. The `conftest` does this itself and says so in a comment (`tests/conftest.py:163-215`, especially `:179`). An autouse `reset_store` fixture resets the in-memory store before every test (`tests/conftest.py:105-109`).

2. Tool configuration (pyproject.toml)

TOOL	CONFIG	LINES
ruff	line-length = 100 , target-version = "py311" . Rules E, F, I, UP, B, S (pycodestyle, pyflakes, isort, pyupgrade, bugbear, bandit). Ignored: S101 (assert in tests), B008 (FastAPI Depends in defaults), E501 (long lines allowed)	pyproject.toml:105-115
mypy	python_version 3.12 . Only checks src/trusted_router (tests are not type-checked). disallow_untyped_defs , check_untyped_defs , warn_unused_ignores , ignore_missing_imports , follow_imports = silent , allow_untyped_globals	pyproject.toml:95-103
pytest	timeout = 600 with timeout_method = "thread" (a wedged shard once cost 2.5 h). Default addopts = -m "not provider_health and not proof_oracle" . asyncio_mode = auto , testpaths = ["tests"] , pythonpath = ["src"]	pyproject.toml:64-80
markers	proof_oracle (expensive frozen-main differential, has its own CI job), provider_health (live provider/catalog state), catalog_as_built (reads the catalog without test vehicles)	pyproject.toml:81-85
coverage	branch = true , source = ["trusted_router"]	pyproject.toml:87-93
dev deps	pytest, pytest-asyncio, pytest-cov, pytest-httpx, pytest-split , pytest-timeout, pytest-xdist , hypothesis , mypy, ruff	pyproject.toml:43-62

xdist and `--dist loadgroup`. CI runs `-n 4 --dist loadgroup`. The `loadgroup` part matters: each server-backed conformance backend (postgres, spanner-pg) is pinned to one `xdist_group`, so its DDL runs serially (`tests/conformance/conftest.py:225-241`, `ci.yml:227-232`).

`.test_durations` **sharding**. `pytest-split` splits the suite using recorded per-test durations (38,116 entries, about 49k test-seconds). Splitting by test count produced 7 m vs 21 m shards (`ci.yml:244-248`). The file **must be recorded on CI runners**. Local runs skip the conformance suites, record them as ~0 s, and unbalance the shards (`.github/workflows/refresh-test-durations.yml:1-12`). To refresh it, dispatch that workflow, or merge shard artifacts with `scripts/ci/merge_shard_durations.py` (`ci.yml:178`). Example: PR #1816 "ci: refresh `.test_durations` from main run 38858291020's shard artifacts".

3. CI anatomy (.github/workflows/ci.yml)

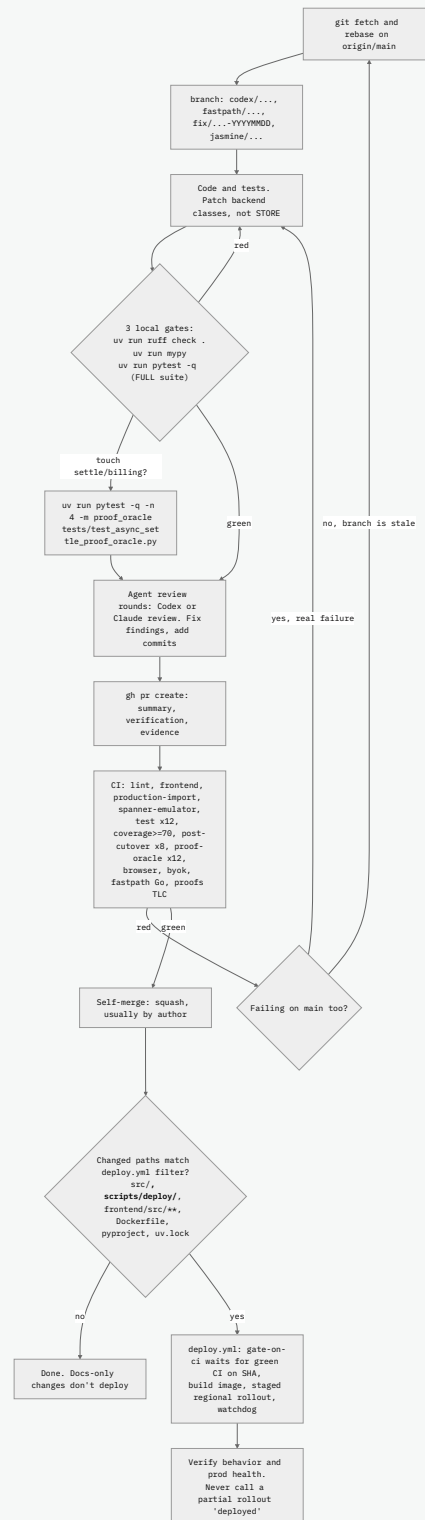
Triggers: push to `main`, PRs to `main`, and `workflow_dispatch`. The price bot uses `dispatch` because `GITHUB_TOKEN` pushes don't trigger workflows (`ci.yml:3-11`). The jobs run in parallel, so wall time is the slowest job (`ci.yml:13-16`).

JOB	WHAT IT RUNS	LINES
lint	<code>ruff check .</code> , <code>mypy</code> , <code>scripts/update_model_changes.py --check</code> (catalog change history is current)	:18-32
spanner-emulator	Native GoogleSQL conformance, prod SQL acceptance and schema drift, package object budgets	:34-100
frontend	<code>tsc</code> (committed <code>.js</code> must match <code>.ts</code>), ESLint on <code>frontend/src</code> , Stylelint on static CSS	:102-118
production-import	<code>uv sync --no-dev</code> , then <code>import trusted_router.main</code> (catches dev-only imports)	:120-130
test	12 shards , each <code>pytest -n 4 --dist loadgroup --splits 12 --group N</code> , with Postgres and PGAdapter service containers. Coverage per shard without a floor	:132-284
coverage	Combines all shard <code>.coverage</code> files and enforces 70% . Fails if any shard failed	:286-346

JOB	WHAT IT RUNS	LINES
test-post-cutover	8 shards, same suite with the lifecycle clock pinned past every scheduled cutover	:348-416
proof-oracle	6 shards × 2 clocks, -m proof_oracle, no coverage	:419-457
browser	node --test tests/js/homepage_*.test.cjs and npm run test:browser (Playwright, tests/browser/)	:459-503
byok-format-ordering	scripts.check_format_ordering (BYOK formats vs. what every enclave accepts)	:505-542
fastpath, fastpath-spanner, fastpath-walks	Go: gofmt, vet, go test -race, TLC comparisons, emulator-backed store/owner/auditor tests	:544-670
proofs	./proofs/check.sh (TLC model-checks every TLA+ spec)	:672-684

Other workflows that matter to contributors: - `deploy.yml` . Push to `main` touching `src/**`, `scripts/deploy/**`, `scripts/entrypoint.sh`, `frontend/src/**`, `Dockerfile`, `.gcloudignore`, `pyproject.toml`, `uv.lock` or `deploy.yml` auto-deploys to Cloud Run (`deploy.yml:28-37`). Its `gate-on-ci` job polls the CI run for the SHA and refuses to deploy unless it is green (:57-80). A `hotfix` input can skip the gate. A supersede check skips stale deploys (:424-445). - `proofs-guards.yml` . Runs only when `proofs/**` changes. 16 shards re-run every guard-table row (`guard_sweep.py --verify`): each guard is removed and must break exactly what its row claims (`proofs-guards.yml:1-55`). - `typed-audit.yml` . Daily at 11:43 UTC. Not a PR gate. Runs `scripts/audit_typed_counters.py` against prod via WIF, plus a Spanner schema-drift audit against the emulator (`typed-audit.yml:1-70`). - `refresh-prices.yml` . Hourly at :37 . Scrapes provider prices. A sandboxed, LLM-rewritable parser tier is AST-whitelisted. The bot commits the snapshot to `main` (`refresh-prices.yml:1-30`). That accounts for the [bot] hourly price refresh: ... commits by `trustedrouter-pricing-bot` (163 commits in 60 days). - `refresh-test-durations.yml` (manual), `dependency-scan.yml` (OSV), and `Dependabot` (`.github/dependabot.yml`).

4. Dev loop



5. Test taxonomy

`tests/` has 922 entries: **855** `test_*.py` files at the top level, ~60 support modules (fakes, mutation tables, fixture generators, oracles), and 6 subdirectories. About 36.7k tests pass locally (PR #1860 body: "36,751 passed").

By area (top-level `test_*.py` , keyword buckets, counts approximate)

AREA	≈ FILES	EXAMPLES	NOTES
Async settle / shadow v2 / proof oracle	11 9	<code>test_async_settle_drain_oracle.py</code> , <code>test_async_settle_proof_oracle.py</code> , <code>test_async_settle_shadow_v2_gate_window.py</code>	108 files start with <code>test_async_*</code> . Backed by <code>tests/async_settle_*_mutations.py</code> and <code>shadow_v2_*</code> generators. 9 files carry <code>proof_oracle</code>
Provider adapters & BYOK	11 9	<code>test_deepseek_*</code> , <code>test_together_*</code> , <code>test_tinfoil_*</code> , <code>test_byok_*</code> , <code>test_provider_*</code> (26)	One file or cluster per upstream
Money: billing, credit, settle, Stripe, Lightning, receipts	10 8	<code>test_billing_*</code> (16), <code>test_credit_*</code> (15), <code>test_settle_*</code> (15), <code>test_lightning_*</code> (10), <code>test_stripe_*</code>	Needs differential or conformance coverage (AGENTS.md:81-87)
Pricing, catalog, models	10 4	<code>test_pricing_*</code> (13), <code>test_catalog_*</code> (15), <code>test_leaderboard_*</code>	Includes dated lifecycle files like <code>test_late_september_2026_retirements.py</code>
Gateway, routing, request path	88	<code>test_gateway_*</code> (24), <code>test_routing_*</code> , <code>test_authorize_*</code> , <code>test_video_*</code> (12), <code>test_fastpath_*</code>	
Deploy, infra, cloud, ops, trust	69	<code>test_deploy_*</code> , <code>test_cloud_*</code> , <code>test_synthetic_*</code> (11), <code>test_trust_*</code> (11), <code>test_ci_workflow.py</code>	Workflow YAML and deploy scripts are tested too (<code>tests/deploy_script_harness.py</code>)
Storage backends & analytics	63	<code>test_spanner_*</code> (8), <code>test_postgres_*</code> (6), <code>test_clickhouse_*</code> (15), <code>test_analytics_outbox.py</code>	
Auth, accounts, keys, console, OAuth	42	<code>test_oauth_*</code> (8), <code>test_console_*</code> (10), <code>test_signup_*</code> , <code>test_company_*</code>	
Public site, docs, blog	23	<code>test_public_*</code> (13), <code>test_blog_featured.py</code> , <code>test_batch_docs.py</code>	Copy and SEO assertions
Other (attribution, axiom, config, clients, lifecycle, ...)	12 0	<code>test_config.py</code> , <code>test_client_*</code> , <code>test_lifecycle_clock.py</code>	

Subdirectories and adjacent suites

LOCATION	WHAT	HOW IT RUNS
<code>tests/conformance/</code>	22 tests plus Spanner DDL/SQL inventory, manifests and allowlists. The Store Protocol contract across <code>memory</code> , <code>postgres</code> and <code>spanner-pg</code> (AGENTS.md:89-105)	Server backends skip locally unless <code>TR_CONFORMANCE_POSTGRES_DSN</code> or <code>TR_CONFORMANCE_SPANNER_PG_DSN</code> is set. Docker commands are in AGENTS.md:95-103 . In CI, service containers in <code>test</code> and the emulator in the <code>spanner-emulator</code> job
<code>tests/fakes/</code>	In-process fakes. <code>tests/fakes/spanner.py</code> is a large fake Spanner (see PR #1860 lock fix)	Imported by tests
<code>tests/fixtures/</code>	Static fixtures	

LOCATION	WHAT	HOW IT RUNS
tests/ browse r/	23 Playwright *.spec.js (homepage, console, chat, OAuth funding, token exchange...)	npm run test:browser. playwright.config.js boots uvicorn on :18081 with TR_STORAGE_BACKEND=memory
tests/ js/	Node node --test homepage regression tests	CI browser job
tests/ shell/	Shell-script tests	
tests- e2e/	Separate TypeScript Playwright project for the /chat playground. 22 specs (streaming, multi-model, picker, persistence, voice...) with mocked SSE	Own package.json and playwright.config.ts. See tests-e2e/README.md
proofs/ /	TLA+ specs (CreditDebt, TerminalOrder, LeaseLifecycle, AuditorCommit, KeyCapFence, AutoRefillHandoff, SurfaceCutover) model-checked by TLC (jar pinned by sha256)	./proofs/check.sh (CI proofs job). check_mutants.py and guard_sweep.py --verify (proofs-guards.yml)
fastpa th/ (Go)	Go tests incl. -race and TLC comparison	CI fastpath* jobs

What "proofs" means here

A spec covers a protocol whose failures are *interleavings*, cases a unit test can't reach. The spec is written **before** the code (proofs/README.md:5-7). Every spec needs: -a .cfg naming each invariant -small bounds, stated -no CONSTRAINT -TypeOK first -liveness where the design promises progress -assumptions listed, each with a widening mutant (proofs/README.md:39-55)

manifest.toml lists **mutants**: one text replacement that must violate a named invariant. If nothing can break a claim, the claim isn't shown to say anything (proofs/README.md:60-77). *.guards.toml tables say which guard protects which claim. proofs-guards.yml proves each table is true. The same idea of mutation tables shows up in the Python tests (tests/async_settle_mutations.py, tests/gateway_edge_mutations.py, ...).

6. Branch, commit and PR conventions

Who commits (last 60 days)

Joseph Perla 1,181 · Jasmine Lee 190 (+18 as jasminelee) · trustedrouter-pricing-bot 163 · dependabot[bot] 13.

Branch names (remote branch prefixes)

codex/ 359 · fastpath/ 66 · fix/ 39 · async-settle/ 36 · fable/ 20 · test/ 15 · infra/ 11 · feat/ 11 · ci/ 11 · dependabot/ 10. Also jasmine/<topic> for Jasmine's work.

- codex/<topic>-YYYYMMDD is agent-driven work, e.g. codex/bedrock-provider-20261018, codex/transcription-20261015.
- fix/<topic>-YYYYMMDD is a dated bug fix, e.g. fix/settlelog-cancel-release-20261018.
- <subsystem>/<topic> covers longer programs, e.g. fastpath/edge-e1-role, async-settle/shadow-v2-opt-in.
- Generated chores: deprecations-202610190004, featherless-stepfun-202610180100, deps/security-bumps.

Commit subjects

Two styles coexist. Most commits (~410 of the last 800 non-merge commits) are **plain imperative sentences**. The rest use a **lowercase subsystem prefix** (fastpath: 59, infra: 25, docs: 19, catalog: 17, test(s): 31, deploy: 12, async settle shadow v2: 12, fix: 12, pricing: 11, ci: 8, proofs: 8). Squash merges add (#NNNN). Design-program commits cite the design section and step: (design 15.8 step 5) points to docs/design/async-settle-shadow-v2-activation.md §15.8. (E1, second increment) refers to staged increments.

Real examples:

```
deploy: shadow v2 opt-in pins for the probe workspace (design 15.8 step 5) (#1853)
tests: fake Spanner reads hold the commit lock (#1860)
service: the edge as a part of the node (E1, second increment) (#1859)
catalog: add frontier, advanced, professional, efficient routing tiers (#1855)
store: refuse a terminal retried after its lease closed (#1843)
infra: pin the fast path release d99cdc049 (a retried terminal after close answers from the disposition) (#1851)
fix(bedrock): require explicit upgraded gateway origin
Schedule provider model retirements from notices received 2026-10-18 (#1852)
Serve the landscape homepage unconditionally and remove the old one
ci: refresh .test_durations from main run 38858291020's shard artifacts (#1816)
[bot] hourly price refresh: 2026-10-18T17:35Z
```

Commit **bodies** are long, plain prose: the mechanism, then the evidence. Agent-authored commits end with Co-Authored-By: Claude Follow-up commits on a PR often record review rounds, for example (from 28e447a0b):

```
Review round one (Codex, 2026-10-19), five findings, all closed: the edge
listens and checks its upstream before admission takes the ring row, ...
```

PR bodies

There's no PR template. The common shape, drawn from #1853, #1858, #1860 and #1857: 1. **What changed and why**. Root cause for fixes ("Fix (the class, not the line)", #1860). 2. **Tests and verification with numbers**. "On the old fake, both tests fail with the exact CI error. With the fix, both pass", "The full suite passes locally: 36,751 passed", "293 focused control-plane tests passed". 3. **Explicit integrity statements**. "No test was skipped, retried or loosened." 4. **Rollout notes and opt-out** for deploy-affecting changes ("Opt-out: empty the four lines", #1853). 5. **Hold markers and companion PRs**, e.g. "DO NOT MERGE UNTIL GATEWAY VERIFIED. Companion gateway: Lore-Hex/quill-cloud-proxy#535" (#1858). The author then posts release-evidence comments with CI run URLs before merging. 6. UI PRs embed QA screenshots committed under docs/homepage/qa/ (#1857). 7. Agent-generated PRs end with 🤖
Generated with [Claude Code](https://claude.com/claude-code).

Review culture: who reviews

- .github/CODEOWNERS assigns * @jperla and says so directly: the company "currently has one person in it... this file becomes load-bearing at first hire".
- No branch protection and no rulesets on main (gh api .../branches/main/protection returns 404, rulesets []). **Every one of the last 40 merged PRs had 0 GitHub reviews and was merged by its author** (jperla or jasminelee).
- The substitutes are **agent review rounds** and **CI gates**. Claude (often "Claude Fable" per the co-author trailers) writes code. **Codex** reviews it in numbered rounds, and each round's findings are closed in follow-up commits ("Review round ten (Codex, 2026-10-19)..."). Some PR bodies log P1/P2 findings per round: "round 1 ... P1 none, one

P2; round 2: APPROVE" (#1853). The controls that don't depend on headcount are gated CI and measured or attested deployments (CODEOWNERS header, SECURITY.md).

- Bots: trustedrouter-pricing-bot (hourly prices; CI is dispatched on its branch before main advances), Dependabot, and generated deprecation PRs (deprecations-YYYYMMDDhhmm).

Agent tooling in the repo

- AGENTS.md is the shared instructions for codex-cli and Claude Code. CLAUDE.md defers to it and repeats the Spanner query safety rule.
- .claude/launch.json holds one launch config: `uv run uvicorn trusted_router.main:app --port 18080`.
- skills/trustedrouter-model-advisor/SKILL.md is a pointer skill. The canonical copy lives in Lore-Hex/LLM-advisor and is also served through the TrustedRouter MCP server's Skills Extension. It's a product skill for end-user agents, not a dev-workflow skill.
- Some deep areas have their own handoff docs that agents must read first: docs/storage-portability/HANDOFF.md (AGENTS.md:10-23) and sites/token-exchange/ADDING-MARKETS.md (AGENTS.md:5-8).

7. Your first PR: walkthrough

Worked example: a small, non-money change in `src/trusted_router`.

1. **Setup.** Install `uv` and Node. Run `uv sync` and `npm ci`. If you'll touch browser tests, also run `npm install playwright chromium`. Start the app with `uv run uvicorn trusted_router.main:app --reload` (README.md:134-139).
2. **Read** AGENTS.md. If the area has a handoff or design doc (docs/design/..., docs/storage-portability/HANDOFF.md), read that too.
3. **Branch from a fresh main.** Run `git fetch origin && git switch -c fix/<topic>-${date +%Y%m%d} origin/main`.
4. **Write the failing test first.** Put it in the matching `tests/test_<area>_*.py` (look at the bucket table above). Use the `client`, `user_headers`, `inference_key` and `inference_headers` fixtures (tests/conftest.py:217-260). If you must fake storage behavior, patch `InMemoryStore.<method>`, **not** `STORE.<method>`. For flaky or concurrency bugs, force the interleaving deterministically instead of relying on timing (the PR #1860 pattern).
5. **Implement.** Fully type every new function: mypy's `disallow_untyped_defs` applies to `src/`. Keep lines reasonable, though E501 is off. Don't reformat files you didn't change.
6. **Run the three gates.** Use `uv run ruff check . && uv run mypy && uv run pytest -q`. Add `-n auto` to speed up the full run. If `ruff check` complains, fix it or run `uv run ruff check --fix <your files>`, **never** `ruff format .`.
7. **Extra gates by area:** -Settle or billing: `uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py`. -Store or backends: start the Postgres/PGAdapter containers and set `TR_CONFORMANCE_*_DSN` (AGENTS.md:95-103). -Frontend TS/CSS: `npm run tsc`, `npm run eslint frontend/src`, `npm run stylelint "src/trusted_router/static/*.css"`, `npm run test:browser`. -Catalog data: `uv run python scripts/update_model_changes.py --check`. -proofs/: `./proofs/check.sh` and `python3 proofs/guard_sweep.py --verify <Spec>`.
8. **Get a review round.** Ask Codex or Claude to review the diff. Address findings in follow-up commits that name the round.
9. **Open the PR** with `gh pr create`. Body: what and why, root cause, tests with real numbers, "nothing skipped or loosened", and rollout or opt-out notes.
10. **CI red?** Rebase onto `origin/main` and re-run before you debug. A failure that also happens on `main` is not yours, but tell someone, because it blocks every deploy.

11. **Merge** (squash). If your paths match the `deploy.yml` filter, the merge **deploys to production**. Watch the deploy run and verify the behavior and health in prod (`AGENTS.md:53-58`). Runtime env vars only change through `scripts/deploy/rollout.sh ENV_VARS` (`AGENTS.md:64-66`).

8. PR checklist

- ☐ Rebased on current `origin/main`
- ☐ `uv run ruff check .` is clean (and I did **not** run `ruff format .`)
- ☐ `uv run mypy` is clean, with all new `src/` functions typed
- ☐ `uv run pytest -q` passes on the **full** suite, not only my file
- ☐ No `monkeypatch.setattr(STORE, ...)`. Backend classes are patched instead (#333)
- ☐ Coverage won't drop below 70%: new code has tests
- ☐ Settle or billing touched: `-m proof_oracle run`, plus a differential or conformance test added
- ☐ Store touched: conformance passes on memory/postgres/spanner-pg, and no conformance test weakened
- ☐ Frontend touched: `.ts` compiled and the `.js` committed, eslint, stylelint and Playwright pass
- ☐ Catalog or model data touched: `scripts/update_model_changes.py --check` passes
- ☐ `proofs/` touched: `check.sh` and the guard sweep pass, with each new claim backed by a mutant
- ☐ No ad hoc production Spanner scans were used in the investigation (`AGENTS.md:68-79`)
- ☐ Runtime config changed through `scripts/deploy/rollout.sh ENV_VARS`, not by hand
- ☐ PR body covers what and why, root cause, verification numbers, "nothing skipped or loosened", and rollout or opt-out notes
- ☐ Companion PRs (e.g. `quill-cloud-proxy`) linked, and hold noted if ordering matters
- ☐ At least one agent review round done, with findings closed
- ☐ I know whether this merge auto-deploys (path filter), and I'll verify prod afterwards

03

PART III

Week 1

1. Architecture & request lifecycle
2. Product surface & docs
3. Security & trust

02 Life of a Chat Completion

1 Authenticate key

2 Route by model + privacy

3 Reserve credits

4 Stream from provider

5 Settle

6 Metadata-only usage

7 Analytics outbox

8 ClickHouse

```
{ "model": "anthropic/claude-sonnet-4.6",  
  "provider": {  
    "min_privacy": "confidential",  
    "allow_fallbacks": true } }
```

zdr · zero-retention floor

confidential (e2e) · confidential compute + E2EE

- ✓ route A satisfies floor
- ✗ route B weaker privacy, filtered out
- ✗ none left? fail closed, never downgrade

Routing honors model, models[] and provider filters (order, only, ignore, sort...).

PART III · CHAPTER 05 · HERO STILL: VIDEO 02 AT 0:26

Architecture & request lifecycle

Subsystem map, the three planes, and a chat completion traced end to end.

- 1. Overview in one page
- 2. System context (C4 level 1)
- 3. One image, six process "surfaces"
- 4. Subsystem map
- 5. Request lifecycle: production /v1/chat/completions
- 6. Control plane vs attested gateway vs fastpath/
- 7. Must-read files (in reading order)
- 8. Glossary
- 9. Gotchas

Storyboard · Video 02: Life of a request

0:17

0:26

0:37

0:46

1:11

1:20

1:29

1:36

Scope: quill-router (this repo) at commit 778120494 (2026-10-19). All paths are relative to the repo root /Users/elevenarms/Code/trustedrouter/quill-router . The attested gateway is in a **separate repo**, quill-cloud-proxy , which is not checked out here. Anything this doc says about the gateway's internals comes from this repo's contract code, docs and comments, not from gateway source.

1. Overview in one page

TrustedRouter is an OpenAI-, Anthropic-and OpenRouter-compatible LLM router. Its main selling point is that the prompt path is verifiable: prompts go only through an attested enclave, and the control plane never sees prompt or output bytes. Three planes share the work:

PLANE	HOSTNAME	CODE	SEES PROMPTS?	OWNS
Attested gateway (API / prompt plane)	api.trustedrouter.com (alias api.quillrouter.com, regional api- <region>.quillrouter.com)	quill-cloud-proxy (separate repo), running in GCP Confidential Space (plus AWS/Azure variants)	Yes , only inside the enclave	TLS termination in the enclave, provider calls, streaming, fallback attempts, token counting, inference receipts
Control plane	trustedrouter.com (aliases allyrouter.com, uptimerouter.com)	this repo, Python/FastAPI, trusted_router.main:app on Cloud Run	No (except the browser /chat-proxy , see Gotchas)	Accounts, workspaces, API keys, BYOK envelopes, catalog and pricing, routing decisions , credit reserve/settle/refund , billing (Stripe/Adyen/PayPal/Lightning/x402), console UI, public site, status, trust page
Fast path (pilot)	internal VPC only	fastpath/ (Go)	No	Lease-based fast admission and batched settlement, so the per-request Spanner reservation leaves the hot path. Pilot on 2 workspaces

The README states the trust boundary directly (README.md:127-131): " api.trustedrouter.com is the attested prompt path and must terminate TLS inside Confidential Space. trustedrouter.com is the control plane and must never serve a production inference fallback." The code enforces it: /chat/completions , /messages , /responses and /embeddings are mounted on the control plane **only when** environment in {local,test} (src/trusted_router/main.py:880-883 , main.py:912-919).

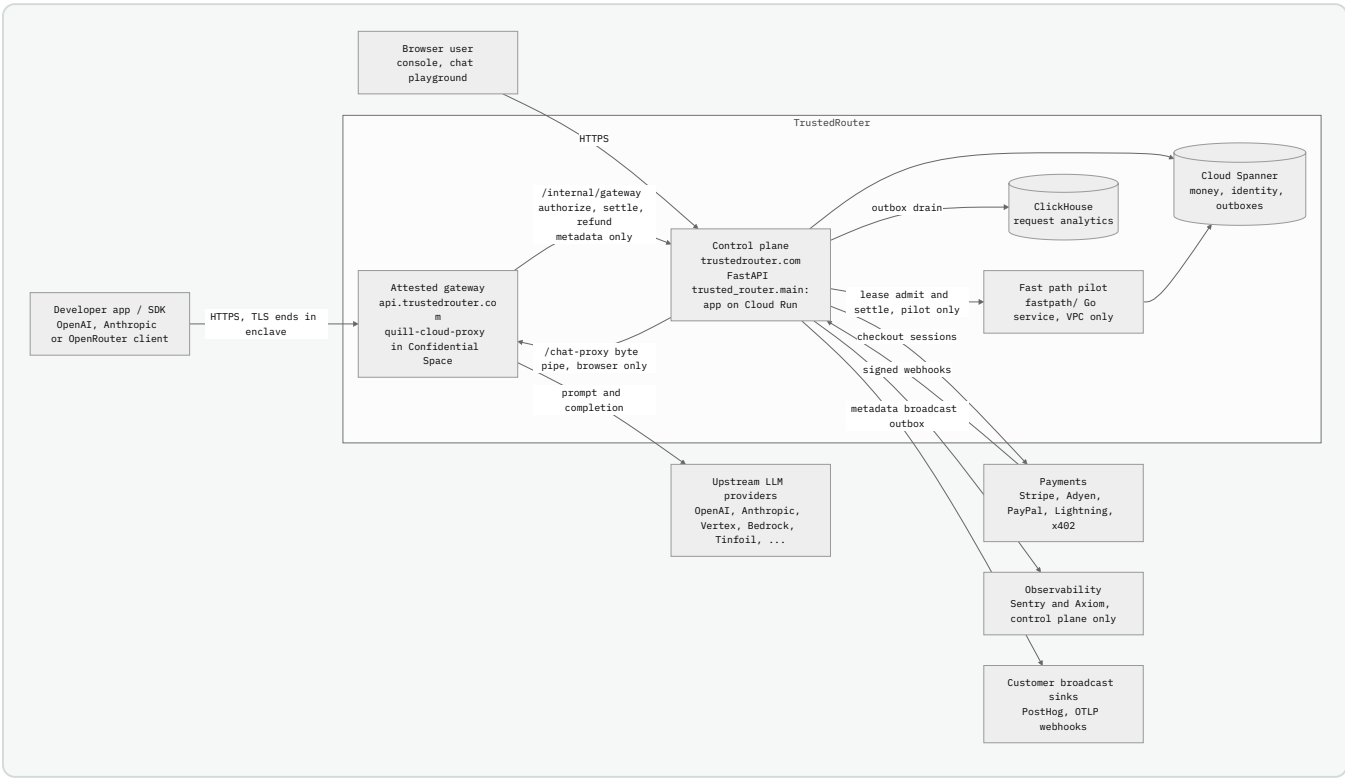
What the control plane does for each production request. The enclave makes two or three server-to-server calls to the control plane's /internal/gateway/* routes (src/trusted_router/routes/internal/gateway.py:2453-2600):

- authorize : validate the key hash, resolve the model and routing candidates (applying privacy filters), estimate cost, and **reserve** credits and key caps atomically in Spanner. It returns authorization_id , route_candidates , any BYOK ciphertext, and the regions.
- Optional heartbeat (Stage D): usage heartbeats during long streams.
- settle (success) or refund (failure or disconnect): book actual tokens to the hold, write the generation record, and enqueue analytics, activity, broadcast and auto-refill work.

Prompt and output bytes never cross this boundary. Only hashes, token counts and metadata do.

Storage in production is TR_STORAGE_BACKEND=spanner-clickhouse (scripts/deploy/rollout.sh:274). **Spanner** is the system of record for money and identity. **ClickHouse** holds request analytics, fed through a durable Spanner outbox. Postgres-wire backends (Azure/AWS standalone clouds) and an in-memory store (tests/local) implement the same Store protocol.

2. System context (C4 level 1)



Multi-cloud: each cloud (GCP, AWS, Azure) is a **standalone TrustedRouter**. Only identity federates; credits, keys and analytics stay in their own cloud (docs/storage-portability/multi-cloud-separation.md). Peer planes resolve keys lazily from the home plane (src/trusted_router/services/federation.py) and forward deferred debt home (src/trusted_router/services/home_settlement.py).

3. One image, six process "surfaces"

One container image serves several **process roles**, chosen by `TR_SERVICE_SURFACE` (src/trusted_router/config.py:438-445). `create_app()` registers only the routes that belong to the chosen surface (main.py:709-909).

SURFACE	PURPOSE	NOTABLE ROUTES	DEPLOY SCRIPT
combined	Local/test default. In prod it is the legacy pre-split service , permitted only with <code>TR_ALLOW_DEPLOYED_COMBINED_SURFACE=true</code> (main.py:235-244)	everything	scripts/deploy/rollout.sh:408-410
public	Anonymous marketing site, catalog, docs, pre-serialized OpenAPI	register_public_routes , catalog , /openapi.json	scripts/deploy/public_surface.sh
actions	Anonymous public actions, memory-backed	register_public_action_routes	—
control	Authenticated account plane: console, keys, billing, OAuth, MCP, chat proxy, webhooks	register_console_routes , register_key_routes , ...	trust jobs use it

SURFACE	PURPOSE	NOTABLE ROUTES	DEPLOY SCRIPT
internal	Machine-to-machine gateway billing authority : /internal/gateway/*, observer internals	register_gateway_internal_routes	scripts/deploy/internal_surface.sh
observer	Regional/standalone-cloud status and observer endpoints (Postgres)	register_observer_internal_routes	azure_control_plane.sh , aws_eu_control_plane.sh

The six-service split is in progress (issue #712 per `config.py:446-449`). Until it completes, the legacy `combined` service still runs in production next to the split `public` and `internal` companions.

4. Subsystem map

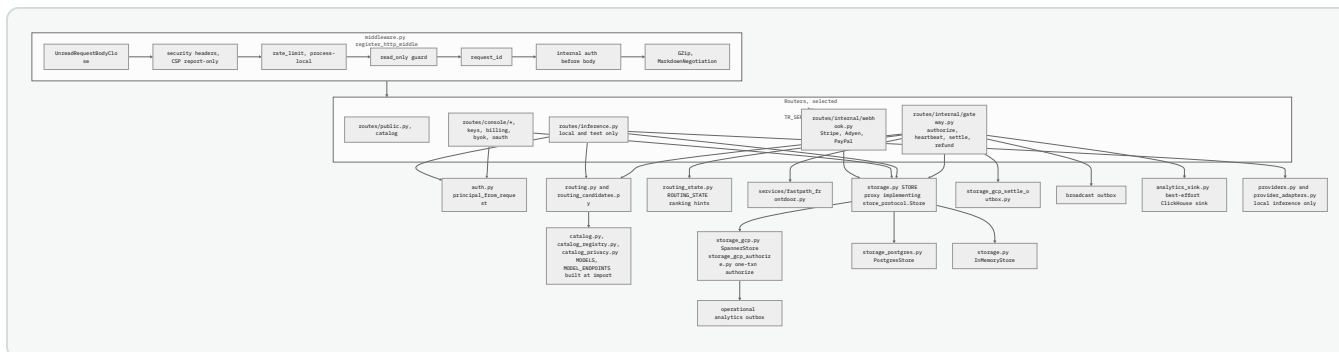
`src/trusted_router/` has **371 top-level modules** and **533 .py files** once subpackages are included (`routes/` 77, `services/` 62, `synthetic/` 22, `content/`, `evals/`). Grouped by prefix and responsibility:

SUBSYSTEM	KEY MODULES	RESPONSIBILITY
App shell	<code>main.py</code> , <code>middleware.py</code> , <code>serve.py</code> , <code>config.py</code> , <code>errors.py</code> , <code>schemas.py</code> , <code>types.py</code>	FastAPI factory, surface-based route registration, middleware chain (request id, rate limit, read-only mode, security headers, gzip, markdown negotiation), uvicorn runner with a SIGTERM drain, the ~2.4k-line <code>Settings</code>
AuthN/AuthZ and keys	<code>auth.py</code> , <code>scopes.py</code> , <code>security.py</code> , <code>storage_keys.py</code> , <code>storage_gcp_keys.py</code> , <code>storage_auth*</code> , <code>oauth_provider.py</code> , <code>wallet_auth.py</code> , <code>routes/keys.py</code> , <code>routes/auth.py</code> , <code>routes/oauth*.py</code>	Principal resolution: dev header, then session cookie, then bearer <code>sk-tr-...</code> or <code>trsess-...</code> . Keys are salted SHA-256 with a high-entropy lookup hash. Management vs inference scopes. OAuth "Sign in with TrustedRouter" and app keys
BYOK	<code>byok_crypto.py</code> , <code>key_management.py</code> , <code>storage_byok.py</code> , <code>storage_gcp_byok.py</code> , <code>byok_aad_backfill.py</code> , <code>byok_v1_attestations.py</code> , <code>routes/byok.py</code>	Envelope-encrypt customer provider keys (per-secret DEK wrapped by KMS). The gateway receives ciphertext plus <code>byok_cache_key</code>
Catalog and pricing	<code>catalog.py</code> , <code>catalog_data.py</code> (4.5k lines), <code>catalog_registry.py</code> , <code>catalog_ingest.py</code> , <code>catalog_privacy.py</code> , <code>catalog_capabilities.py</code> , <code>pricing.py</code> , <code>provider_lifecycle.py</code> , <code>model_aliases.py</code> , <code>data/*.json</code>	Builds <code>MODELS / MODEL_ENDPOINTS</code> at import time from the OpenRouter snapshot and provider manifests. Uniform markup (cost x1.055, \$0.01/M floor). Privacy tiers per endpoint. Retirements and price cutovers
Routing	<code>routing.py</code> , <code>routing_candidates.py</code> , <code>routing_state.py</code> , <code>provider_ranking.py</code> , <code>provider_reliability.py</code> , <code>request_capabilities.py</code> , <code>model_regions.py</code> , <code>regions.py</code> , <code>enclave_regions.py</code>	Parse OpenRouter-style <code>provider{order,only,ignore,allow_fallbacks,min_privacy,zdr,data_collection,sort,max_price,require_parameters}</code> . Expand meta-models (<code>auto</code> , <code>zdr</code> , <code>e2e</code> , <code>eu</code> , <code>cheap</code> , <code>tiers</code>). Filter and sort endpoint candidates. Process-local latency/throughput hints
Providers and adapters (local/test inference only)	<code>providers.py</code> , <code>provider_adapters.py</code> , <code>provider_streaming.py</code> , <code>provider_types.py</code> , <code>adapter.py</code> , <code>provider_payloads.py</code>	<code>ProviderClient</code> for OpenAI-compatible, Anthropic and Gemini. Converts between Messages/Responses and Chat formats. In production the enclave makes the real provider calls
Inference routes (local/test)	<code>routes/inference.py</code> , <code>services/inference.py</code> , <code>services/inference_quota.py</code> , <code>services/inference_errors.py</code>	Local implementation of <code>/chat/completions</code> , <code>/messages</code> , <code>/responses</code> , <code>/embeddings</code> , with the same reserve/settle semantics, so the dev loop needs no enclave
Gateway contract (prod hot path)	<code>routes/internal/gateway.py</code> (6k lines), <code>gateway_boot.py</code> , <code>gateway_timing.py</code> , <code>receipt_keys.py</code> , <code>stage_d.py</code> , <code>stage_d_policy.py</code> , <code>client_context.py</code> , <code>request_tags.py</code>	<code>authorize</code> , <code>heartbeat</code> , <code>settle</code> , <code>refund</code> , <code>disposition</code> , <code>register-boot</code> , <code>validate</code> , <code>key</code> , <code>resolve-custom-model</code> . Boot-signed requests (<code>X-TR-Boot-Auth</code>). Stage D heartbeat cohort

SUBSYSTEM	KEY MODULES	RESPONSIBILITY
Money core	money.py , storage_gcp_authorize.py , storage_gcp_counter_dml.py , storage_gcp_counters.py , storage_gcp_credit_shards.py , credit_debt.py , typed_balance.py , spend_windows.py , strict_budget.py , billing_policy.py	Integer microdollars . One-transaction authorize (credit DML, key-cap DML, tr_reservation INSERT, authorization INSERT). 16 credit shards per workspace. Debt rules. Daily/weekly/monthly key windows
Settle durability	storage_gcp_settle_outbox.py , services/settle_outbox*.py , services/auto_refill*.py , storage_gcp_legacy_reservations.py	Durable settle outbox (enabled in prod, rollout.sh:550) so a settle that fails partway is never lost or reaped for free. Auto-refill handoff to the payment-owning process
Async settle (flag-off, shadow)	async_settle*.py (98 modules, 86 of them async_settle_shadow_v2*), services/async_settle*.py , billing_snapshot.py , async_settle_ticket.py , routes/settlements.py	Future: an enclave settle that returns within 2 s and finalizes durably later. Today TR_ASYNC_SETTLE_ENABLED=false and shadow-v2 evidence collection runs only for the probe workspace (rollout.sh:660-664)
Fast path bridge	services/fastpath_frontdoor.py , services/fastpath_settle.py , fastpath/ (Go)	Pilot: the control plane asks the Go front door to admit under a regional lease instead of a Spanner reservation. off , busy or invalid means take today's path
Speculation (shadow)	speculation_protocol.py , services/speculation_shadow.py , storage_gcp_speculation_shadow.py , routes/internal/speculation.py	Ed25519 JWS wire contract for speculative provider starts. Shadow only (TR_SPECULATIVE_PROVIDER_SHADOW_ENABLED=false)
Meta / "fusion" models	catalog_data.py:2773-2830 (trustedrouter/synth , fusion , fusion-code , selector , mapreduce , polyphemus-1.0) , polyphemus.py , evals/fusion*.py	Router-defined virtual models. The control plane resolves candidates and billing (decide_route_endpoint_candidates). Multi-model orchestration runs in the gateway
Storage abstraction	store_protocol.py (~97 methods), storage.py (InMemoryStore , STORE proxy , create_store), storage_gcp.py (7.4k), storage_postgres.py (7.9k), storage_models.py , storage_errors.py	Store protocol. The tests/conformance/ suite is the executable contract across the memory, postgres and spanner-pg backends
Analytics	analytics_sink.py , storage_gcp_operational_analytics_outbox.py , storage_operational_analytics.py , operational_analytics*.py , provider_analytics.py , clickhouse/ (top-level drain/rollup jobs)	A durable Spanner outbox feeds ClickHouse with one-way surrogate IDs and no content. Best-effort benchmark sink. Rollups and Parquet archive
Broadcast	storage_broadcast.py , services/broadcast*.py , routes/broadcast.py , routes/internal/broadcast_queue.py	Customer-configured PostHog/OTLP export, metadata-only unless include_content is set (and then only the enclave gets the destination)
Billing and payments	services/stripe_billing.py , adyen_billing.py , paypal_billing.py , lightning.py , x402_billing.py , routes/billing.py , routes/internal/webhook.py , credit_transfer.py , partner_billing.py , custom_model_billing.py , app_markup_billing.py	Top-ups and idempotent webhook crediting. App and creator markups, payouts
Trust and fraud	trust_tiers.py , trust*.py , *_trust_history.py , verification*.py , phone_verification.py , veriff_verify.py , signup_gate.py	Account trust tiers from payment, refund and dispute history. Identity verification gates
Custom / user models	custom_model_rules.py , user_model_rules.py , storage_*custom_models.py , storage_*user_models.py , services/user_model*.py	Custom models (hidden prompt plus markup) and user-provided models (BYO endpoint with owner payouts)
Web UI / public site	dashboard.py (6.6k), homepage.py , routes/console/* , routes/public.py , templates/ , static/ , seo*.py , og.py , markdown_negotiation.py , content/ , frontend/	Server-rendered Jinja console and marketing site, docs, llms.txt , markdown for agents
Synthetic monitoring and ops	synthetic/* (probes, status, leaderboard, remediator, drift), routes/internal/synthetic.py , sentry_config.py , axiom_config.py	In-process probe loop, /status , burn-rate SLOs (router_core vs control_plane), remediator
Media	image_generation.py , speech.py , transcription.py , video_billing.py , storage_*video_jobs.py , routes/internal/video_jobs.py	Route types images , audio.speech , audio.transcriptions , videos , with their own reservation rules

SUBSYSTEM	KEY MODULES	RESPONSIBILITY
Growth / marketing	<code>acquisition.py</code> , <code>marketing*.py</code> , <code>google_ads*.py</code> , <code>social_rewards*.py</code> , <code>competitor_comparisons.py</code> , <code>token_exchange.py</code>	Funnel attribution, ads conversions, comparison pages

Component diagram (control plane internals)



5. Request lifecycle: production /v1/chat/completions

5.1 Step by step

- Client → enclave.** The client calls `https://api.trustedrouter.com/v1/chat/completions` with `Authorization: Bearer sk-tr-v1-...`. TLS terminates **inside** the Confidential Space enclave. The enclave parses the body, derives `estimated_input_tokens`, `input_modalities` and `requested_parameters`, and hashes the API key. The raw key and the prompt stay inside the enclave.
- Enclave → control plane** `POST /internal/gateway/authorize` (`routes/internal/gateway.py:2492`). Body is `GatewayAuthorizeRequest` (`src/trusted_router/schemas.py:366-431`): `api_key_lookup_hash`, `model`, `models`, `provider`, `token estimates`, `stream`, `region`, `tags`, `metadata`, `idempotency_key`, `route_type`, and so on.
 - Middleware:** `internal_auth_before_body_middleware` checks the internal token before the body is read (`middleware.py:153-161`, `_internal_auth_before_body` at `middleware.py:473`). Then request id and the process-local rate limit.
 - Per-key admission:** `_AUTHORIZE_ADMISSION.try_acquire(subject)` returns 429 when one key has too many authorizations in flight. The handler then moves to a worker thread (`gateway.py:534-587`). All gateway storage IO is synchronous and runs through `run_in_threadpool`.
 - Internal auth:** `require_internal_gateway` compares the bearer or `x-trustedrouter-internal-token` with `TR_INTERNAL_GATEWAY_TOKEN` in constant time, and is fail-closed outside local/test (`routes/internal/_shared.py:53-65`).
 - Key + workspace snapshot** (`gateway.py:856-940`): one folded read loads the key, workspace, BYOK rows and boot record. It rejects disabled or expired keys (401) and wrong-scope keys, and verifies `X-TR-Boot-Auth` if present (`gateway_boot.py`). The workspace must be billing-active (`assert_workspace_billing_active`). Tags merge from the key and the request.
 - Idempotency replay:** a matching `idempotency_key` plus `request_fingerprint` returns the original authorization, which prevents double reservation (`gateway.py:~1200-1320`).
 - Model resolution and routing:** `normalize_routing_inputs(body_dict, settings, resolved_region)` (`gateway.py:1321`, `routing.py:926`) gives `NormalizedRoutingInputs(model_ids, preferences)`.
 - `provider_route_preferences` (`routing.py:633-746`) validates every `provider.*` field. An unknown field returns 400 with the exact name. `min_privacy` maps through `PRIVACY_TIER_ALIASES` (`catalog_data.py:61-89`): `any / standard =0`, `no_store =1`, `zdr =2`, `confidential / e2e / e2ee =3`. A typo is a 400 so the request can't

silently weaken. `provider.zdr=true` adds the ZDR requirement.

- Meta-model aliases (`trustedrouter/auto`, `zdr`, `e2e`, `eu`, `tiers`, ...) expand to ordered pools (`routing_candidates.py:138-420`). `trustedrouter/zdr` and `trustedrouter/e2e` carry implicit privacy floors (`ROUTING_MODEL_MIN_PRIVACY_TIERS`).
- Dispatch is by route type: `chat` → `chat_route_endpoint_candidates` (`routing.py:253-289`), or embeddings, decide (fusion/selector), image, speech, transcription, video, fetch, user model (`gateway.py:~1410-1470`).
- **Privacy filter:** `_apply_endpoint_provider_filters` keeps only endpoints where `endpoint_meets_privacy_requirement(endpoint, req)` holds for every requirement (`catalog_privacy.py:231-251`). *Confidential* needs verified confidential compute **and** provider E2EE **and** explicit ZDR, and **never** the model's own vendor. Unknown values fail closed. If nothing passes, the request gets a 400 ("No route candidates match the requested provider filters"). It never falls back to a weaker route.
- `data_collection=deny` is a soft filter. It is relaxed when it would empty the set (`_filter_candidates_soft_data_collection`, `routing.py:~1080-1106`). `min_privacy` is never relaxed.
- BYOK eligibility per candidate (`_byok_configs_for_candidates`, `_eligible_gateway_endpoint_candidates`), service-tier filtering, then `ROUTING_STATE.rank(...)` reorders using process-local latency/throughput hints (`routing_state.py:71-179`; hints only, never an authority). `allow_fallbacks=false` trims to one candidate (`gateway.py:~1580-1597`).
- **Estimate:** `_endpoint_cost_microdollars` over the candidates (the max across candidates when auto fallback is possible), plus receipt fee, custom-model markup, app markup and additional hosted-tool cost (`gateway.py:~1595-1690`).
- **Fast path pilot:** for pilot workspaces with credit-only candidates, `_fastpath_admission` calls the Go front door. If admitted, the typed store reserves nothing and the lease holds the money (`gateway.py:~1820-1835`, `5087`).
- **Reserve** (the money step): `_typed_store.authorize_gateway_typed(...)` (`gateway.py:~1866`, `storage_gcp.py:5479`, implemented in `storage_gcp_authorize.py`). One Spanner read-write transaction runs the idempotency read, then conditional credit DML on a credit shard, then pause/strict-budget checks, then conditional key-cap DML, then the `tr_reservation` INSERT, then the `gateway_authorization` INSERT (`storage_gcp_authorize.py:1-14`). Insufficient balance returns 402. If a peer plane has deferred settlement enabled, a local refusal may fall back to the home plane.
- **Response:** `GatewayAuthorizeData` (`schemas.py:325-348`) with `authorization_id`, prospective `generation_id`, `route_candidates` (provider, endpoint, upstream model, inference location, BYOK ciphertext and cache key), `regions`, `broadcast_destinations`, `usage_type`, timing phases (`key_lookup_ms`, `routing_ms`, `store_ms`, `post_commit_ms`).

2. **Enclave → provider.** The enclave tries `route_candidates` in order. A failure before any output token moves on to the next candidate, and it records `route_fallbacks` and `route_failures`. It streams SSE back to the client and counts tokens.

3. **Optional** `POST /internal/gateway/heartbeat` (Stage D, `gateway.py:611-680`). Boot-signed usage heartbeats during long streams. If the final settle is lost, the reaper can book a `reaped_snapshot` charge from the frozen Stage D pricing snapshot (`stage_d.py`) instead of releasing the hold for free. Eligibility depends on a signed accepted-image policy (`stage_d_policy.py`).

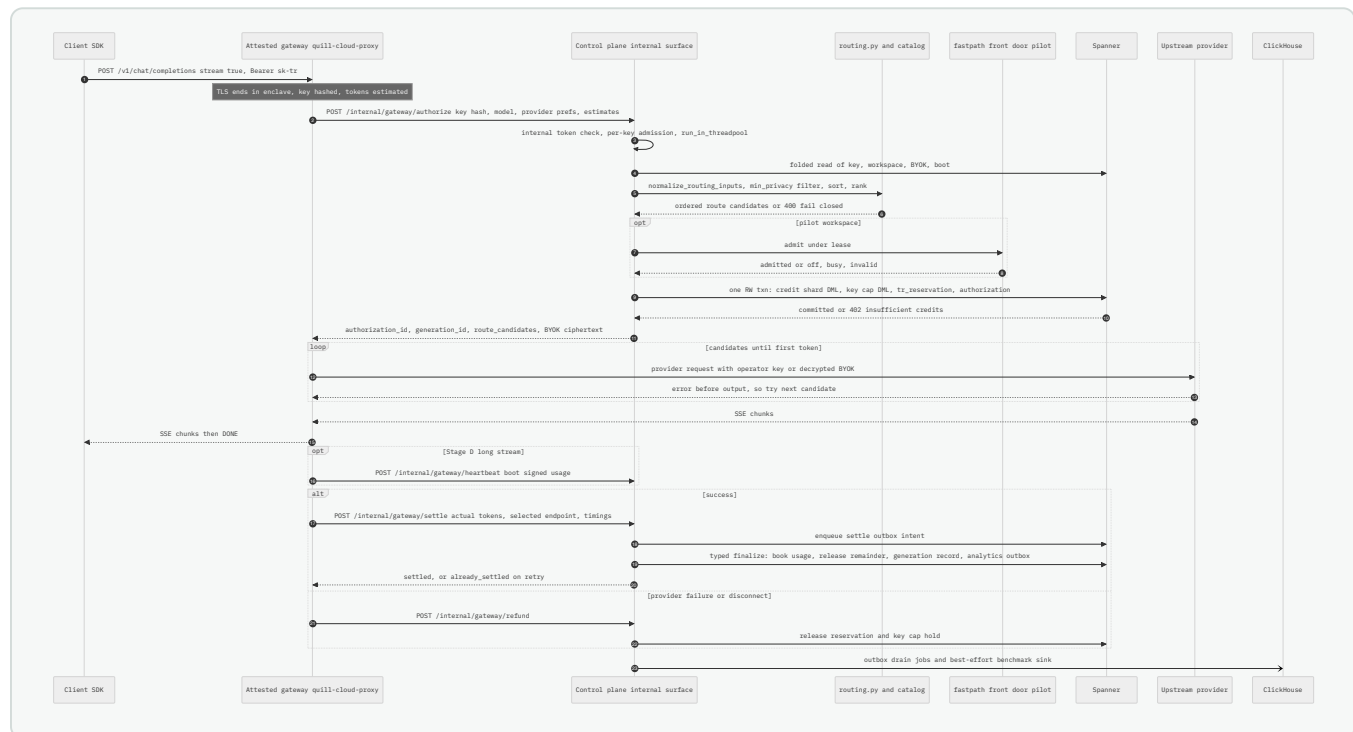
4. **Enclave →** `POST /internal/gateway/settle` (`gateway.py:2552`) with `GatewaySettleRequest` (`schemas.py:566+`): `authorization_id`, actual input/output/cache/reasoning tokens, `selected_model` / `selected_endpoint`, `finish_reason`, `streamed`, `elapsed_seconds`, `first_token_seconds`, `route_fallbacks`, and client context. - `AsyncSettlementRoute` (`routes/settlements.py`) intercepts only when async-settle protection is on **and** the gateway sent `X-TR-Settlement-Mode: async-v1|sync`. Otherwise it is the legacy handler. - `_settle_gateway_with_admission_sync` (`gateway.py:3721`) applies per-key settle admission (503 + Retry-After when overloaded). Then `_settle_gateway_authorization` (`gateway.py:3776`):

- **One-shot:** a settle on an already-settled authorization returns `already_settled` and never charges twice (`_already_settled_gateway_data`, `gateway.py:5271`).
- Prompt-cache normalization (`normalized_prompt_accounting`, `services/settle_outbox_apply.py:112`). Anthropic `input_tokens` **exclude** cached tokens, while OpenAI and Gemini prompt counts **include** them.
- Actual cost in microdollars from the authorized endpoint's frozen terms, then a `Generation.from_settle_body(...)` metadata record. No content.
- **Durable settle outbox** (`TR_SETTLE_OUTBOX_ENABLED=true` in prod): enqueue the settle intent first (`storage_gcp_settle_outbox.py`), then `typed_finalize_gateway_authorization` atomically books usage, releases the unused hold, writes the bounded `tr_generation` record and the activity/analytics outbox rows, and marks the outbox row done. If finalize fails, the outbox drain (`services/settle_outbox_drain.py`) replays it. Deterministic generation IDs make replays idempotent.
- Post-commit side effects: auto-refill scheduling (Stripe, owned by the `control` surface), creator/app/custom-model payouts, metadata broadcast enqueue, and benchmark samples to the best-effort `analytics_sink` (`analytics_sink.py`, a bounded queue that drops rows rather than block).
- Fast path authorizations are settled through `_settle_leased_authorization` (`gateway.py:5181`).

5. **Refund** (`POST /internal/gateway/refund`, `gateway.py:2561`) uses the same `_settle_gateway_authorization(success=False)`. It releases the reservation and key-cap hold after a provider failure or client disconnect before output. Expired holds are reaped (`storage_gcp_authorize.py:1224`), and that is why the settle outbox and Stage D exist.

6. **Analytics.** Spanner outbox rows are drained into ClickHouse by the `clickhouse/ingest_*` jobs, using one-way workspace/key surrogates (`storage_gcp_operational_analytics_outbox.py:1-12`). ClickHouse failures never affect inference or settle (`docs/clickhouse-reliability.md`).

5.2 Sequence diagram (streaming chat with settle)



5.3 The local/test path (same semantics, no enclave)

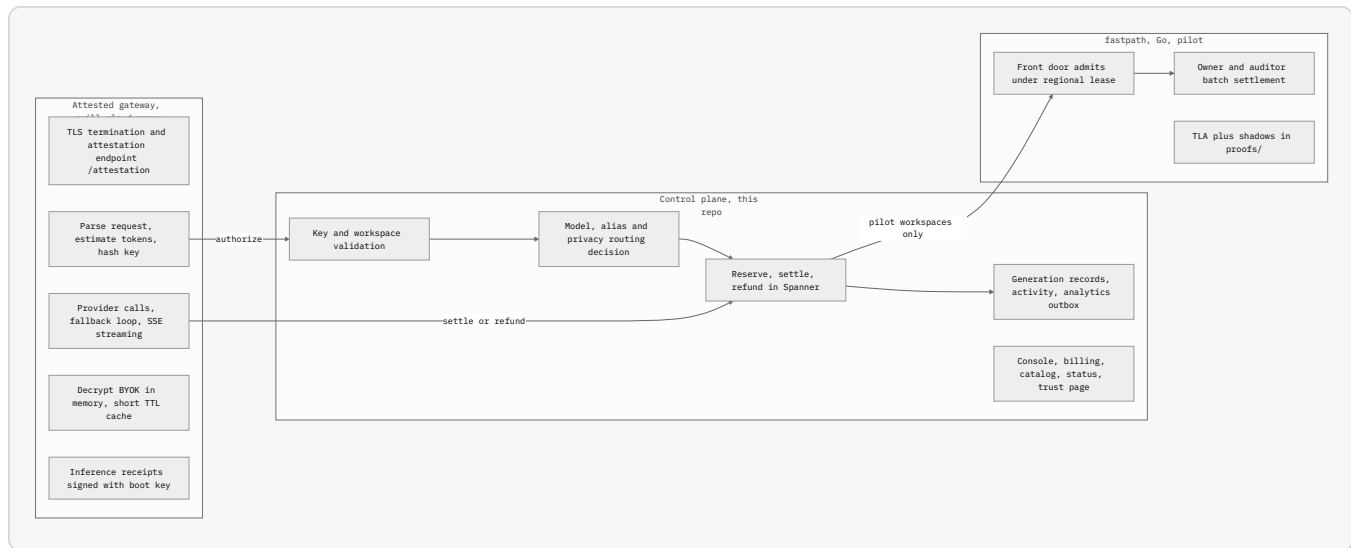
In `local / test`, `routes/inference.py` mounts `/chat/completions` and the other inference routes (`routes/inference.py:150-420`):

- `InferencePrincipal` → `require_inference_key` → `principal_from_request` (`auth.py:90-177`, `auth.py:361`).
- `provider_route_preferences` + `chat_route_candidates` or `chat_route_endpoint_candidates`, then `_local_routes_meeting_privacy_floor` (`routes/inference.py:106`).
- More than one candidate, or `trustedrouter/auto`, means meta-route rollover (`run_chat_candidates` or `_candidate_stream_bytes`). Otherwise a single candidate (`run_chat` or `run_chat_stream`).
- `services/inference.py:_run_stream` (`services/inference.py:162-218`) opens `reserved_quota(...)` (`services/inference_quota.py:84`), which reserves the key limit and then credits. It streams from `ProviderClient.stream_chat`, computes the actual cost, calls `ticket.settle(actual_cost)`, and runs `STORE.add_generation(...)`. An exception refunds (`QuotaTicket`, `inference_quota.py:42-80`).
- Live upstream calls happen only with `TR_ENABLE_LIVE_PROVIDERS=true` (default false, `config.py:486`) and keys from the local keys file.

5.4 Anthropic `/v1/messages`

- **Client config:** `ANTHROPIC_BASE_URL=https://api.trustedrouter.com` (no `/v1`, the SDK adds it), `ANTHROPIC_API_KEY=sk-tr-...` (`README.md:62-64`).
- **Production:** the enclave serves `/v1/messages`. The control-plane contract is the same `authorize / settle / refund`. Model and provider filters go through the same `chat_route_endpoint_candidates` and privacy logic. The settle path normalizes Anthropic cache accounting, where `input_tokens` excludes `cache_read_input_tokens` and `cache_creation_input_tokens` (`schemas.py:574-578`). Coverage registry: `openrouter_coverage.py:99`.
- **Local/test** (`routes/inference.py:307-351`): `_require_messages_model` (`:929`), then the privacy floor, then `messages_to_chat_body` (`adapter.py:83-99`), which turns Anthropic `system` into a leading system chat message and copies allowed params. The request then runs on the same `run_chat / run_messages_stream`, and the result is wrapped by `_anthropic_messages_envelope` (`routes/inference.py:506`). On the outbound side, `chat_to_anthropic` (`adapter.py:102+`) rebuilds Anthropic payloads for the Anthropic provider adapter (`provider_adapters.py:338-465`).
- `/v1/responses` works the same way through `responses_to_chat_body` and `_responses_api_envelope`.

6. Control plane vs attested gateway vs fastpath/



CONCERN	ATTESTED GATEWAY	CONTROL PLANE	FASTPATH
Prompt and output bytes	Handles them, never logs them	Never (prod)	Never
Who chooses the provider	Executes the ordered candidates and decides fallback at runtime	Decides the candidate set and order , applies privacy floors	—
Money	Reports actual usage	Authoritative reserve/settle/refund	Lease admission for pilot workspaces. Python still owns key, model, estimate, authorization row and generation record (<code>services/fastpath_frontdoor.py:1-22</code>)
Trust anchor	Hardware attestation, published image digest	Ordinary Cloud Run, Sentry/Axiom allowed	VPC-internal
Billing authority it trusts	Only <code>https://trustedrouter.com</code> is baked into the measured image (<code>docs/design/fast-admission-pilot.md</code>)	—	Called by the control plane in the pilot, not by the enclave
Status	Live in <code>us-central1</code> , <code>us-east4</code> , <code>eu-west4</code> , <code>us-west1</code> (<code>README.md:436-445</code>)	4 Cloud Run regions behind a global LB	Pilot: <code>TR_FASTPATH_PILOT_ENABLED=true</code> for 2 workspaces (<code>rollout.sh:630-632</code>)

The long-term design (`docs/design/fast-admission-and-batched-settlement.md`) has the enclave call the fast path directly, removing the control plane from request latency. That has not been built.

7. Must-read files (in reading order)

#	PATH:LINE	WHY
1	<code>AGENTS.md:1</code>	Hard rules: the three CI gates, the <code>STORE</code> monkeypatch trap, money-code test rules, Spanner query safety, auto-deploy on push to main
2	<code>README.md:127</code>	Trust boundary, internal gateway contract (<code>:501-515</code>), scale and multi-region shape
3	<code>src/trusted_router/main.py:227</code>	<code>create_app()</code> : surface-gated route registration, background loops, exception→HTTP mapping, inference gate at <code>:912</code>

#	PATH:LINE	WHY
4	src/trusted_router/config.py:438	service_surface and the ~2.4k-line Settings . Every TR_* flag lives here
5	src/trusted_router/middleware.py:113	Middleware order and why (raw ASGI vs BaseHTTPMiddleware, internal auth before body)
6	src/trusted_router/auth.py:90	principal_from_request : credential precedence and sk-tr- / trsess- prefix dispatch
7	src/trusted_router/routes/internal/gateway.py:1	The production hot path. Authorize :534 / :856 , settle :590 / :3776 , refund :2561 , route registration :2453
8	src/trusted_router/schemas.py:366	GatewayAuthorizeRequest and GatewaySettleRequest (:566): the wire contract with quill-cloud-proxy
9	src/trusted_router/routing.py:633	provider_route_preferences : every OpenRouter routing knob, min_privacy parsing
10	src/trusted_router/routing.py:253	chat_route_endpoint_candidates and the filter/sort helpers (:1080-1300)
11	src/trusted_router/catalog_privacy.py:231	endpoint_meets_privacy_requirement : the exact meaning of ZDR and confidential
12	src/trusted_router/catalog_data.py:61	Privacy tier constants and aliases. Meta-model IDs at :2472-2830
13	src/trusted_router/catalog_registry.py:1	Import-time construction of MODELS / MODEL_ENDPOINTS and uniform pricing (money-critical)
14	src/trusted_router/routing_candidates.py:138	Meta-model pools: auto, zdr, e2e, eu, cheap, monitor, tiers
15	src/trusted_router/storage_gcp_authorize.py:1	The one-transaction authorize. Reaper at :1224
16	src/trusted_router/storage.py:3880	_StoreProxy / STORE (:3947) and create_store (:3976). Backend selection
17	src/trusted_router/store_protocol.py:1	The Store contract: reserve :913 , settle :922 , authorize_gateway_typed :1162 , typed_finalize_gateway_authorization :1215
18	src/trusted_router/storage_gcp_settle_outbox.py:1	Durable settle outbox (exactly-once via INSERT-as-claim plus a lease fence)
19	src/trusted_router/routes/inference.py:150	Local inference routes. The easiest way to see the whole lifecycle in one process
20	src/trusted_router/services/inference_quota.py:84	reserved_quota / QuotaTicket : reserve→settle/refund in miniature
21	src/trusted_router/adapters.py:1	Chat ↔ Messages ↔ Responses conversions
22	src/trusted_router/providers.py:142	ProviderClient and its adapters (provider_adapters.py)
23	src/trusted_router/gateway_boot.py:1	Enclave boot registration and per-request X-TR-Boot-Auth signatures
24	src/trusted_router/services/fastpath_frontend.py:1	Fast path pilot bridge and its fallback semantics
25	src/trusted_router/money.py:1	Microdollar units and signup credit. Never floats
26	src/trusted_router/analytics_sink.py:1	Why analytics must never affect a request
27	src/trusted_router/routes/chat_proxy.py:1	The one place prompt bytes cross the control plane (browser playground)
28	docs/storage-portability/multi-cloud-separation.md:1	Identity federates, credits do not

#	PATH:LINE	WHY
2	docs/design/durable-settle-	Why a lost settle used to mean free inference
9	outbox.md:1	

8. Glossary

TERM	MEANING
Workspace	The billing and tenancy unit. Owns credits, API keys, BYOK configs and members. <code>api_key.workspace_id</code> is the trusted attribution, never a header (<code>gateway.py:890</code>)
Management key vs inference key	Key scopes. Signup issues a one-time management key. Inference requires an inference-scoped key (<code>auth.py:312-345</code> , <code>scopes.py</code>)
Lookup hash / key hash	API keys are stored as salted SHA-256. A separate high-entropy lookup hash allows point reads. The enclave sends hashes, not raw keys
Microdollars	Integer money unit, 1 USD = 1,000,000 (<code>money.py:5</code>). All ledger math is integer
Reserve	Pre-authorization hold of the estimated cost on workspace credit (and key cap) before the provider call. In prod it is one Spanner transaction inside <code>authorize</code>
Authorization	The <code>gateway_authorization</code> row created at authorize. Carries frozen terms, candidates and the reservation id. Its terminal dispositions are <code>settled</code> , <code>refunded</code> , <code>reaped_snapshot</code> , and so on
Settle	Book actual usage against the authorization. Releases the unused remainder and records the generation. One-shot and idempotent
Refund	Release a reservation without charging (provider failure, disconnect)
Reaper	Background release of expired holds. Historically it released without charging, which is why the settle outbox and Stage D snapshots exist
Settle outbox	Durable Spanner table recording settle intents so an interrupted settle can be replayed exactly once
Async settle (v1) / shadow-v2	A planned mode where the enclave's settle returns fast and finalization is deferred. It currently only collects shadow evidence (flag off)
Fast path / lease	Go service (<code>fastpath/</code>) that admits requests against a pre-granted regional credit lease and batches settlement. Pilot only
Stage D	Heartbeat cohort. Boot-signed enclaves send usage heartbeats so long streams can be billed from a frozen pricing snapshot if the final settle is lost
Boot / boot key	Per-boot enclave identity registered at <code>/internal/gateway/spend-lease/register-boot</code> . Signs authorize, heartbeat and disposition requests, plus inference receipts
Inference receipt	Client-verifiable signed receipt from the enclave. A receipt fee (basis points) is frozen on the authorization
BYOK	Bring Your Own Key: a customer's provider key, envelope-encrypted (KMS-wrapped DEK). The enclave gets the ciphertext and decrypts it in memory
ZDR	Zero Data Retention, <code>min_privacy=zdr</code> (tier 2): the endpoint has an explicit zero-retention policy
No-store	Tier 1: the provider does not store request or response content
Confidential / E2E / E2EE	Tier 3: verified confidential compute and provider-side E2EE and explicit ZDR, never served by the model's own vendor. Fails closed
min_privacy	Hard request-level privacy floor in <code>provider.min_privacy</code> . Never relaxed
data_collection=deny	Soft retention preference, relaxed if it would leave no candidates
Meta-model	A router-defined virtual model (<code>trustedrouter/auto</code> , <code>zdr</code> , <code>e2e</code> , <code>eu</code> , <code>cheap</code> , <code>free</code> , <code>monitor</code> , <code>frontier</code> , <code>fusion</code> , <code>selector</code> , <code>polyphemus-1.0</code> , ...) that expands to an ordered candidate pool or an orchestration

TERM	MEANING
Candidate / route candidate	One <code>(Model, ModelEndpoint)</code> pair the gateway may try, in order
Endpoint	A specific provider deployment of a model, with price, region, privacy flags and capabilities
Surface	Process role selected by <code>TR_SERVICE_SURFACE</code> (combined, public, actions, control, internal, observer)
Home plane / peer plane	Multi-cloud: keys are issued on the home plane, and peers federate them lazily with zero local balance
Custom model / user model	Custom: a catalog model plus a hidden prompt and markup. User-provided: a BYO endpoint whose owner earns payouts
Broadcast	Customer-configured metadata export (PostHog, OTLP) through a durable outbox
router_core vs control_plane SLO	Status separates prompt-path availability from dashboard/billing availability
Conformance suite	<code>tests/conformance/</code> , the executable <code>Store</code> contract across backends

9. Gotchas

1. **Prod control plane does not serve inference.** `_control_plane_inference_enabled` returns true only for local/test (`main.py:912-919`). If you debug a prod chat bug in `routes/inference.py`, you are reading the wrong code. Prod goes through the enclave → `routes/internal/gateway.py`.
2. `/chat-proxy/v1/chat/completions` **terminates TLS on the control plane.** It forwards bytes unparsed to the attested API for the browser playground only (`routes/chat_proxy.py:1-30`). SDK traffic must never use it.
3. **The STORE proxy.** `monkeypatch.setattr(STORE, ...)` poisons later tests (once 136 failures). Patch the backend class (`AGENTS.md:42-45`). `typed_billing_store()` exists because `isinstance` on the proxy lies (`storage.py:3954-3968`).
4. **The catalog is built at import time.** `MODELS` and `MODEL_ENDPOINTS` come from `data/*.json` when the module loads. An hourly price-refresh bot commits to `main`, so rebase before trusting a failure (`AGENTS.md:47-49`).
5. **Money code needs differential/conformance tests.** Run the proof oracle when touching settle/billing: `uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py` (`AGENTS.md:39-40`, `81-87`).
6. **Async settle is huge and dormant.** 98 modules (`async_settle_shadow_v2_*` alone is 86) are shadow/evidence machinery behind `TR_ASYNC_SETTLE_ENABLED=false`. Do not mistake it for the live settle path, which is `_settle_gateway_authorization`.
7. **Config is code.** Cloud Run env vars come from the `ENV_VARS` arrays in `scripts/deploy/rollout.sh` and the companion surface scripts. Setting them by hand is overwritten on the next rollout. Pushing `src/**` to main **auto-deploys** (`AGENTS.md:60-66`).
8. **Combined surface is fail-closed in deploys.** A missing `TR_SERVICE_SURFACE` defaults to `combined`, and that raises outside local/test unless the migration bridge flag is set (`main.py:235-244`, `rollout.sh:12-15`).
9. **Gateway storage is synchronous.** Every gateway handler runs in `run_in_threadpool` to avoid head-of-line blocking. AnyIO's 40-token pool is shared, which is safe only because Cloud Run concurrency is ~8 (`gateway.py:2454-2467`).
10. **Rate limiting is process-local** by design (`middleware.py:144-146`). Fleet-wide limits belong at the front door. (The README's "rate limits use the configured store" line is out of date.)
11. **min_privacy vs data_collection**. The first is a hard filter and the second is soft. ZDR and confidential compute are separate dimensions: a TEE without a retention promise does not satisfy ZDR (`routing.py:1131-1135`).
12. **Anthropic cache-token semantics differ.** Anthropic's `input_tokens` excludes cached tokens, while OpenAI and Gemini include them. Settle normalizes this (`schemas.py:574-578`, `settle_outbox_apply.py:112`).

- 13. **Never scan** `tr_entities.body` **in prod Spanner**, even read-only with `LIMIT`. Use `scripts.inspect_payment_owner` and ClickHouse (`AGENTS.md:70-79`). A body scan caused the September 6 CPU incident.
- 14. **The README mentions Bigtable** for activity. The current write path is the Spanner `tr_generation` bounded record plus the Spanner analytics outbox → ClickHouse (`storage_gcp_generation_records.py:1-6`, `docs/storage-portability/analytics-ingestion.md`). Treat Bigtable references as historical unless the code says otherwise.
- 15. **Unknown routing fields return 400 and unimplemented OpenRouter fields return 501** `not_supported_in_alpha`. Nothing is silently ignored (`README.md:334-336`).
- 16. **Each cloud is its own TrustedRouter**. An AWS-issued key does not work on GCP, and credits don't move without an explicit transfer (`multi-cloud-separation.md`).
- 17. **Analytics must never affect money**. The benchmark sink drops rows when its queue is full, and the outbox enqueue rides the settle commit only on the one-commit happy path (`AGENTS.md:83-87`, `analytics_sink.py:1-25`).

07 The Product Surface

2 Router aliases

"model": "trustedrouter/..."

trustedrouter/auto

Fallback across providers. NO implicit privacy floor.

trustedrouter/zdr

Hard floor: zero-data-retention routes only.

trustedrouter/e2e = /confidential

Hard floor: confidential compute + E2EE.

trustedrouter/eu

EU-focused pool. Routing policy, not a residency guarantee.

Combo models

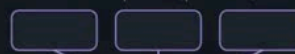
Zeus

Prometheus

Plato

Socrates

model panel (parallel)



judge / synthesizer

versioned multi-model graphs

/eu prefers EU providers. Pair it with the EU gateway endpoint for in-region traffic.

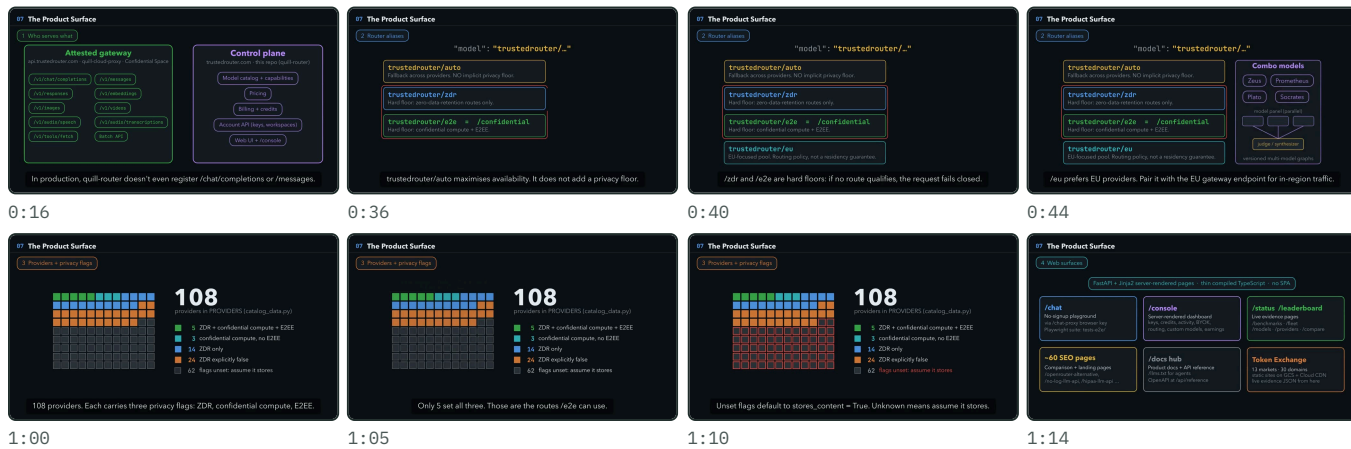
PART III • CHAPTER 06 • HERO STILL: VIDEO 07 AT 0:44

Product surface & docs

Every API surface, router alias, provider, script and doc in the repo, indexed.

- 1. Product mindmap
- 2. Feature table: endpoint → owning module
- 3. Web UI pages (server-rendered Jinja2, served by FastAPI)
- 4. Growth programs
- 5. Supported providers
- 6. frontend/ and front-end toolchain
- 7. sites/ — Token Exchange static sites
- 8. Other non-core directories at a glance
- 9. Repo directory map (top level)
- 10. scripts/ categorized (92 top-level entries)
- 11. docs/ index
- 12. Gotchas for newcomers in this area

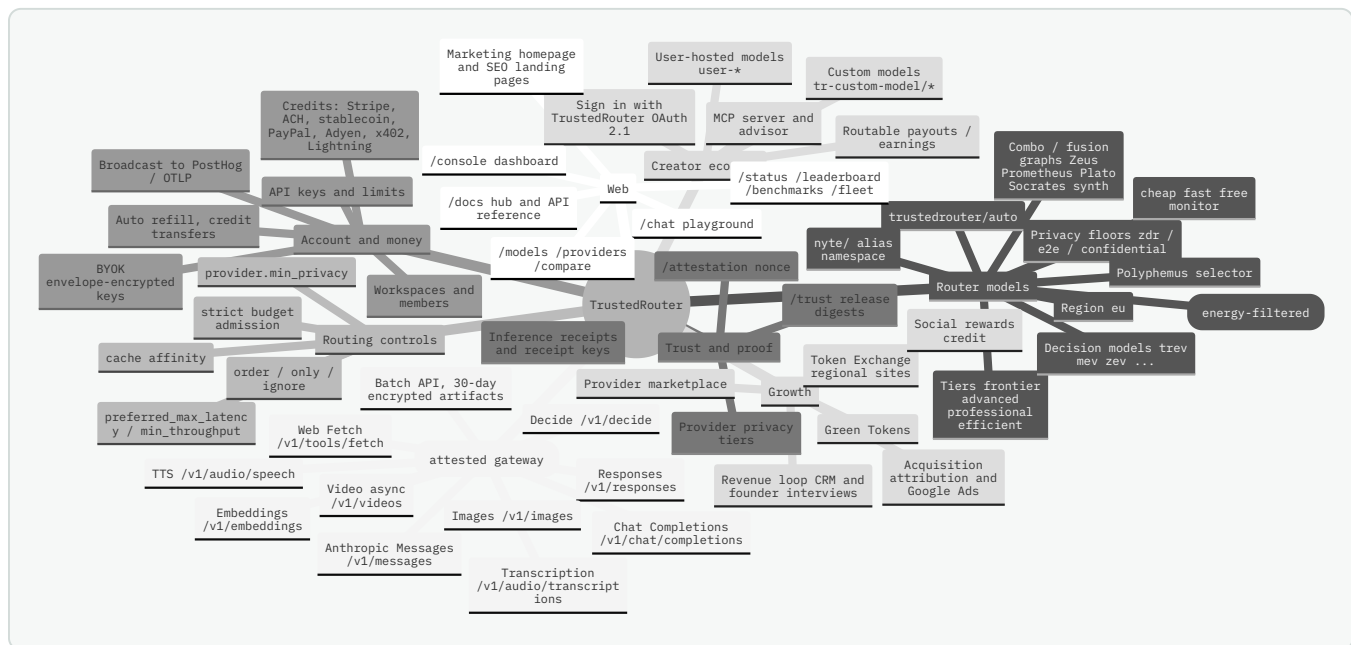
Storyboard · Video 07: The product surface



Scope: what a user sees (APIs, router aliases, web pages, growth programs), plus everything in the repo outside the core Python router: `frontend/`, `sites/`, `clickhouse/`, `experiments/`, `interviews/`, `release/`, `skills/`, `scripts/`, and a navigable index of `docs/`. Repo root: `/Users/elevenarms/Code/trustedrouter/quill-router` (all paths below are relative to it). Snapshot: `main @ 778120494, 2026-10-19`.

The most important architecture fact for this area: the repo is the **control plane** (`trustedrouter.com`). Prompt-bearing endpoints (chat, messages, responses, embeddings, images, video, TTS, transcription, fetch, batch) run in the **attested gateway** (`api.trustedrouter.com`, in the separate `quill-cloud-proxy` repo, running in GCP Confidential Space). In production the control plane does not register `/chat/completions`, `/messages`, `/responses` or `/embeddings` (README.md:211-214). Media endpoints are registered here only as `not_supported()` stubs that exist for the OpenAPI document (`src/trusted_router/routes/compat.py:36-130`). The gateway calls back into this repo via `/internal/gateway/{authorize,settle,refund}` (`src/trusted_router/routes/internal/gateway.py:2492-2561`, README.md:501-515). This repo owns **catalog**, **pricing**, **capability metadata**, **routing candidates**, **billing**, and the **web UI** for every feature.

1. Product mindmap



2. Feature table: endpoint → owning module

"Gateway" = real implementation lives in `quill-cloud-proxy` ; this repo supplies metadata/billing plus a stub. Router mounts every API router at both `/` and `/v1` (`src/trusted_router/main.py:727-728`).

2a. Inference & media (user-facing API on `api.trustedrouter.com/v1`)

FEATURE	ENDPOINT	CONTROL-PLANE MODULE(S)	NOTES / DOC
Chat Completions	POST /v1/chat/completions	routes/inference.py:157, services/inference.py, providers.py, provider_adapters.py	Registered on control plane only when control-plane inference enabled (main.py:880-884); production = gateway
Anthropic Messages	POST /v1/messages	routes/inference.py:307	Claude Code base URL https://api.trustedrouter.com (README.md:62-64)
Responses	POST /v1/responses	routes/inference.py:385	Polyphemus selector uses route type responses.polyphemus.select (polyphemus.py:3-4)
Embeddings	POST /v1/embeddings, GET /v1/embeddings/models	routes/inference.py:353, routes/catalog.py:489	
Image generation	POST /v1/images (stub), GET /v1/images/models, GET /v1/images/models/{author}/{slug}/endpoints	image_generation.py:1-40 (data-only capability list), routes/catalog.py:495,505, routes/compat.py:128	docs/image-generation.md
Video generation (async)	POST /v1/videos, GET /v1/videos/models, GET /v1/videos/{jobId}, .../content	routes/compat.py:110-125 (stubs), video_billing.py, storage_video_jobs.py, routes/internal/video_jobs.py	docs/video-generation.md

FEATURE	ENDPOINT	CONTROL-PLANE MODULE(S)	NOTES / DOC
Text-to-speech	POST /v1/audio/speech	speech.py:1 ("never accepts speech text"), routes/compat.py:61, data/speech_pricing.json	docs/text-to-speech.md
Transcription	POST /v1/audio/transcriptions	transcription.py:1-15, routes/compat.py:10, data/transcription_pricing.json	docs/transcription.md
Web Fetch	POST /v1/tools/fetch (model deepseek/fetch)	web_fetch.py:7-9, routes/compat.py:39-58	Credits only; not ZDR/confidential; docs/web-fetch.md
Batch API	gateway	—	Retains enclave-encrypted artifacts up to 30 days (README.md:194-196); public page /docs/batch (routes/public.py:1330)
Decide	POST /v1/decide	catalog_data.py:2551-2567 (decision model IDs)	Public page /docs/decide (routes/public.py:1334)
Rerank, private models	POST /v1/rerank, GET /v1/private/models/...	routes/compat.py:14-30	Not supported (OpenRouter-compat stubs)
Browser chat proxy	POST /chat-proxy/v1/chat/completions	routes/chat_proxy.py:93, routes/internal/chat_browser_key.py	Powers no-signup /chat

2b. Catalog, routing, trust (public, no key)

FEATURE	ENDPOINT	MODULE
Model catalog	GET /v1/models, /models/count, /models/picker, /models/{author}/{slug}/endpoints	routes/catalog.py:539-633, catalog.py, catalog_data.py, catalog_registry.py
ZDR endpoints list	GET /v1/endpoints/zdr	routes/catalog.py:715
Regions, providers	GET /v1/regions, GET /v1/providers	routes/catalog.py:748,758, regions.py, provider_locations.py
User-provided models	GET /v1/models/user-provided[{id}]	routes/user_models_public.py:15,26
Model change history	GET (see module)	routes/model_changes.py:78,94, model_changes.py, docs/model-changes.md
Model aliases	(resolution)	model_aliases.py:66-81 (nyte/ ↔ trustedrouter/, auto-routing → auto), catalog_aliases.py (reviewed model equivalences), docs/model-aliases.md
Request capabilities	in /v1/models trustedrouter.capabilities	request_capabilities.py, docs/design/model-capabilities.md
Receipts	GET /.well-known/inference-receipt-keys, /trust/receipt-keys.json	routes/public.py:2511-2512, receipt_keys.py, docs/client-receipts.md
Trust/release proof	/trust/control-plane.json, /trust/{gcp,aws,azure}-release.json, /trust/pcr0-aws.txt, /trust/image-digest-gcp.txt	routes/public.py:2482-2715, trust.py
Status / history	GET /status, /status.json, /status/history	routes/public.py:2128,2217,2232, synthetic/status.py
Leaderboard	/leaderboard, /leaderboard/video, /leaderboard/video.json	routes/public.py:2136,2156,2173, frontend/src/leaderboard.ts, docs/status-leaderboard-evidence.md

FEATURE	ENDPOINT	MODULE
Benchmarks	<code>/benchmarks</code> , <code>/benchmarks/reports/{period}.json</code> , <code>/rankings</code>	<code>routes/public.py:1922-1950</code> , <code>benchmark_reports.py</code> , <code>monthly_benchmarks.py</code>
Fleet	<code>/fleet</code> , <code>/fleet.json</code>	<code>routes/public.py:2185,2194</code>
Model picker data	<code>/choose</code> , <code>/choose/catalog.json</code> , <code>/ai-iq/models.json</code>	<code>routes/public.py:1664-1690</code> , <code>choose_catalog.py</code> , <code>ai_iq.py</code>
MCP	<code>POST /mcp</code> , <code>POST /mcp/advisor</code> , <code>/.well-known/mcp.json</code>	<code>routes/mcp.py:74</code> , <code>routes/mcp_advisor.py:350</code> , <code>routes/public.py:2013-2014</code> , <code>mcp_skills.py</code>
LLM-readable docs	<code>/llms.txt</code> , <code>/llms-full.txt</code> , <code>/docs/llms.txt</code>	<code>routes/public.py:2077-2107</code>
OpenAPI	<code>/api/reference</code> , <code>/redoc</code>	<code>routes/public.py:1120-1210</code> , <code>public_openapi.py</code>

2c. Account/control-plane API (authenticated, `trustedrouter.com/v1`)

FEATURE	ENDPOINTS	MODULE
API keys	<code>GET /key</code> , <code>GET/POST /keys</code> , <code>/keys/bulk-delete</code> , <code>GET/PATCH/DELETE /keys/{hash}</code>	<code>routes/keys.py:89-243</code>
Credits & billing	<code>/credits</code> , <code>/credits/summary</code> , <code>/billing/checkout</code> , <code>/billing/paypal/orders/{id}/capture</code> , <code>/billing/x402/{fund,settle}</code> , <code>/billing/portal</code> , <code>/billing/payment-methods/setup</code> , <code>/credits/coinbase</code>	<code>routes/billing.py:54-289</code> , <code>services/stripe_billing.py</code> , <code>services/paypal_billing.py</code> , <code>services/adyen_billing.py</code> , <code>services/x402_billing.py</code>
Credit transfers	<code>POST/GET /credits/transfers</code>	<code>routes/credit_transfers.py:51,68</code>
Activity	<code>/activity</code> , <code>/generation</code> , <code>/generation/content</code>	<code>routes/activity.py:18-95</code>
BYOK	<code>/byok/providers/{provider}</code>	<code>routes/byok.py:28,41</code> , <code>byok_crypto.py</code>
Broadcast observability	<code>/broadcast/destinations[...]</code>	<code>routes/broadcast.py:31-113</code> , <code>README.md:220-232</code>
Workspaces	<code>/workspaces[...]</code> , <code>/organization/members</code>	<code>routes/workspaces.py:21-131</code>
Custom models	<code>GET/POST /custom-models</code> , <code>GET/PATCH/DELETE /custom-models/{id}</code>	<code>routes/custom_models.py:30-141</code> , <code>custom_model_rules.py</code> , <code>docs/custom-models.md</code>
User-hosted models	<code>/user-models[...]</code> incl. <code>clock-in/out</code> , <code>heartbeat</code> , <code>probe</code> , <code>rotate-secrets</code> , <code>earnings/transfer</code>	<code>routes/user_models.py:54-351</code> , <code>docs/design/reverse-harness-contract.md</code>
Sign in with TrustedRouter (OAuth 2.1)	<code>/oauth/authorize</code> , <code>/oauth/token</code> , <code>/oauth/apps</code> , <code>/auth/approve</code> , <code>/auth/fund</code>	<code>routes/oauth_keys.py:70-281</code> , <code>routes/oauth_apps.py:80-116</code> , <code>oauth_provider.py</code> , <code>docs/sign-in-with-trustedrouter.md</code>
Payouts	<code>/payouts[...]</code> , <code>/payouts/onboarding</code>	<code>routes/payouts.py:56-164</code> , <code>docs/routable-payouts.md</code>
Notify (SMS/voice)	<code>/notify</code> , <code>/notify/phone/{start,confirm}</code> , <code>/notify/texml</code>	<code>routes/notify.py:59-213</code>
Bedrock group buy	<code>/v1/bedrock-group-buy[...]</code> , <code>/bedrock-group-buy/pledge</code>	<code>routes/bedrock_group_buy.py:44-222</code>
Acquisition events	<code>POST /analytics/events</code>	<code>routes/acquisition.py:74</code> , <code>acquisition.py:1-4</code>
Lightning feedback	<code>POST /lightning/feedback</code>	<code>routes/lightning_support.py:25</code>

2d. Router model IDs (aliases a user types into model)

MODEL ID	MEANING	SOURCE
trustedrouter/auto (auto-routing , nyte/auto)	Automatic fallback across providers; no implicit privacy floor	catalog_data.py:2472 , model_aliases.py:66-81
trustedrouter/zdr	Hard zero-data-retention floor	catalog_data.py:2519,2529-2532
trustedrouter/e2e / trustedrouter/confidential	Hard confidential-compute + E2EE floor	catalog_data.py:2521-2523,2531
trustedrouter/eu	EU-focused pool; routing policy, not a residency guarantee; pair with api-europe-west4.quillrouter.com	catalog_data.py:2517 , README.md:255-259
trustedrouter/frontier/advanced/professional/efficient	Fixed ordered pools, one model per attempt	catalog_data.py:2480-2515 , README.md:234-244
trustedrouter/cheap , /fast , /free , /monitor	Price/speed pools; monitor = internal uptime pool	catalog_data.py:2474-2478,2534
trustedrouter/green	Energy-eligible directly hosted models, hard provider filter	catalog_energy.py:5 , docs/green-tokens.md
Combo/fusion: zeus-*, prometheus-*, plato-*, socrates-*, synth , fusion , advisor	Versioned multi-model graphs (parallel panel + judge/synthesizer, optional advisor)	docs/combo-models-september-2026.md , evals/fusion_*.py , scripts/fusion_*.eval.py
trustedrouter/polyphemus-1.0	Selector model (picks a model; generation billed normally)	polyphemus.py:1-4 , docs/design/polyphemus-1.0.md
trustedrouter/{trev,mev,zev,lev,gev,dev,oev,gemmev}-1.0	Named decision models for /v1/decide	catalog_data.py:2551-2559
tr-custom-model/{username}-{slug}	Creator custom model (catalog model + hidden prompt)	docs/custom-models.md

Routing controls: provider.min_privacy , explicit order/filters, preferred_max_latency / preferred_min_throughput , cache affinity, strict budget admission — docs/routing-controls-20260927.md , routing.py , routing_candidates.py , strict_budget.py .

3. Web UI pages (server-rendered Jinja2, served by FastAPI)

Stack: Jinja2 templates (src/trusted_router/views.py:1 "Shared Jinja2 environment"), static assets mounted at /static (routes/public.py:995), thin compiled TypeScript. No SPA framework; the console is fully server-rendered (frontend/src/dashboard.ts:1-11).

SURFACE	ROUTES	TEMPLATES / ASSETS
Homepage	/ (routes/public.py:1214), homepage.py	templates/homepage/*.html , content.json
Chat playground (no signup)	/chat (routes/public.py:2368)	static/chat.js , chat_controls.js , chat.css ; docs/chat-playground-controls.md ; Playwright suite tests-e2e/
Console dashboard	/console , /console/{activity,api-keys,authorized-apps,broadcast,byok,credits,credit-transfers,custom-models,earnings,routing,settings,social-rewards,user-models,welcome,account/verification,account/preferences}	routes/console/*.py , templates/console/*.html , static/console.js

SURFACE	ROUTES	TEMPLATES / ASSETS
Auth & OAuth consent	<code>/auth/{slug}/login</code> (Google/GitHub), wallet SIWE	<code>routes/oauth.py:92-100</code> , <code>templates/auth/*</code> , <code>frontend/src/dashboard.ts</code>
Catalog pages	<code>/models</code> , <code>/models/{id}</code> , <code>/providers</code> , <code>/providers/{slug}[/performance]</code> , <code>/compare/models/...</code>	<code>routes/public.py:2275-2402</code>
Docs hub	<code>/docs</code> , <code>/docs/{migrate-from-openrouter,agent-setup,tagging,provider-routing,performance-routing,spend-controls,model-precision,receipts,telemetry,prompt-caching,batch,decide,web-search,video,mcp,notify,evals,provider-conformance,synth,fusion,x402,user-models,custom-models}</code>	<code>routes/public.py:1274-1378,1701</code>
Status/leaderboard/benchmarks	see 2b	<code>frontend/src/leaderboard.ts</code>
SEO/comparison landing pages (~60)	<code>/compare/*</code> , <code>/openrouter-alternative</code> , <code>/hipaa-llm-api</code> , <code>/no-log-llm-api</code> , <code>/deepseek-api-privacy</code> , etc.	<code>routes/public.py:1252-1660</code> , <code>competitor_comparisons.py</code> , <code>seo_catalog.py</code> , <code>seo_meta.py</code>
Growth pages	<code>/token-exchange</code> , <code>/new-york-token-exchange</code> , <code>/token-exchange/{savings,security}</code> , <code>/green-tokens</code> , <code>/index</code> (NYTE token index), <code>/providers/marketplace</code> , <code>/badge</code> , <code>/media-kit</code>	<code>routes/public.py:1717-1795</code> , <code>frontend/src/token-index.ts</code>
Legal/compliance	<code>/legal/{dpa,baa,soc2-readiness,hipaa-readiness,subprocessors}[/terms]</code> , <code>/privacy</code> , <code>/terms</code>	<code>routes/public.py:1834-1910</code>

4. Growth programs

PROGRAM	WHAT IT IS	WHERE
Token Exchange	Static enterprise-acquisition microsites: 13 markets / 30 owned domains (global <code>thetokenexchange.com</code> , NY <code>nytokenexchange.com</code>), served from a public GCS bucket + Cloud CDN behind the existing global LB. Pages poll live price/privacy evidence from <code>GET /token-exchange/evidence/{profile}.json</code> ; the only form posts to the email-gated <code>/token-exchange/brief</code> (buyer deck / security pack). First-party <code>/token-exchange/events</code> collector → Axiom. "Token Exchange" is positioning: buying LLM tokens like a commodity market with verifiable provider evidence.	<code>sites/token-exchange/README.md:7-39</code> , <code>sites/token-exchange/markets.json</code> , <code>ADDING-MARKETS.md</code> (read before adding a market, <code>AGENTS.md:5-8</code>), <code>src/trusted_router/token_exchange.py:1-30</code> (evidence profiles shared-gcp/new-york/europe/dubai), <code>routes/public.py:969-985,2199</code> , <code>docs/marketing/token-exchange.md</code> , <code>exchange_analytics.py</code>
Green Tokens	<code>trustedrouter/green</code> alias: provider-declared energy eligibility, hard filter, fallback stays in pool	<code>docs/green-tokens.md:1-8</code> , <code>catalog_energy.py</code>
Social rewards	Promo credit for new Google/GitHub signups: 150,000 µ\$ at signup + 30,000 µ\$ per social action (GitHub star, X follows, LinkedIn, optional YouTube), max \$0.30	<code>docs/social-rewards.md:1-15</code> , <code>social_rewards.py</code> , <code>routes/console/social_rewards.py:69-104</code>
Revenue loop	Human-approved outreach: founder audio interviews → transcribed quotes → Google Sheet CRM → short quote-based messages. No unapproved posts/sends.	<code>docs/revenue-loop.md:1-20</code> , <code>docs/outreach-*</code> , <code>docs/runs/</code> , <code>interviews/</code> , <code>docs/founder-interview.md</code>

PRO GRA M	WHAT IT IS	WHERE
Acq uisit ion attri buti on	First-party, metadata-only funnel + Google Ads conversions/experiments	<code>acquisition.py</code> , <code>marketing_funnel.py</code> , <code>marketing_experiments.py</code> , <code>google_ads_*.py</code> , <code>scripts/axiom_growth/</code> , <code>docs/marketing/</code>
Prov ider mar ketp lace	Supplier onboarding with catalog JSON schema	<code>routes/public.py:1717-1760</code> , <code>provider_portal.py</code>
Ligh tnin gRo uter	<code>lightningrouter.ai</code> funding site: Bitcoin Lightning → USD credits	<code>experiments/lightning_router/README.md</code> , <code>docs/lightning-*.md</code> , <code>scripts/lightning/</code>
Alia ses	<code>allyrouter.com</code> , <code>uptimerouter.com</code> first-party aliases	<code>docs/operations/*-alias.md</code>

5. Supported providers

Source of truth: `PROVIDERS` dict in `src/trusted_router/catalog_data.py:468` (108 entries; `Provider` dataclass at `catalog_data.py:26-50`, default `stores_content=True` — "assume stored" unless cited). OpenAI-compatible base URLs/env keys: `OPENAI_COMPATIBLE_PROVIDERS` in `src/trusted_router/providers.py:42-138`; native adapters (Anthropic, Gemini, Cohere) in `provider_adapters.py`. Per-provider onboarding notes: `docs/providers/`.

Legend: ZDR / CC (confidential compute) / E2EE = explicit flag in catalog (Y = True, N = False, · = unset/unverified). Prepaid = TR-managed key available (`supports_prepaid`). Line = `catalog_data.py` line.

PROVIDER (SLUG)	LINE	Z D R	C C	E 2 E E	PREP AID
TrustedRouter (<code>trustedrouter</code>)	487	Y	Y	Y	Y
Tinfoil (<code>tinfoil</code>)	823	Y	Y	Y	Y
Privatemode (<code>privatemode</code>)	839	Y	Y	Y	Y
NEAR AI (<code>near-ai</code>)	872	Y	Y	Y	Y
Chutes (<code>chutes</code>)	1090	Y	Y	Y	Y
Phala (<code>phala</code>)	782	Y	Y	N	Y
Confidential AI (<code>confidential-ai</code>)	1365	Y	Y	N	Y
OG Private Computer (<code>zero-g</code>)	1451	·	Y	N	Y
Cerebras, Together, Parasail, Baseten, Telnyx, Inceptiontron, Nebius, Regolo, Zro, Cohere, Voyage	617, 714, 929, 1003, 1021, 1130, 1505, 2042, 1965, 2286, 2309	Y	·	·	Y
Engy, ScaleDown, Greenference	1201, 1386, 1988	Y	N	N	Y

PROVIDER (SLUG)	LINE	Z D R	C C	E 2 E E	PREP AID
Anthropic, OpenAI, Google AI Studio, Google Vertex, DeepSeek, Mistral, DeepInfra	544, 558, 582, 596, 630, 649, 1483	N	·	·	Y
Telluvian, OpenRouter, Meta, Venice, Neurometric, Pearl, StepFun, Relace, Recraft, Black Forest Labs, Decart, NVIDIA NIM, W&B, Nscale, Databricks, Azure AI Foundry, Fortanix	469–2324	N	N	N	Y
Kimi, Z.AI, Fireworks, xAI Grok, Novita, SiliconFlow, Lightning AI, GMI, FriendliAI, Wafer, Crusoe, Makora, DigitalOcean, Cloudflare Workers AI, Morph, Atlas Cloud, StreamLake, MiniMax, Thinking Machines, Xiaomi MiMo, Alibaba, LTX, Runway, Kling, Upstage, Sail Research, Perplexity, Reka, NextBit, AkashML, Abliterate, Mancer, Aion Labs, SambaNova, Arcee, Inception, Sakana, ElevenLabs, Krea, BytePlus, IO Intelligence, Boundless, Amazon Bedrock, Lyceum, Scaleway, Featherless, Jina, TypeSafe, Vercel AI Gateway, fal.ai, Tencent TokenHub	668–2219	·	·	·	Y
Perceptron, Modal, Riverflow, Hyperbolic, OVHcloud, Vultr, Liquid AI, Baidu Qianfan, Darkbloom, Hugging Face, Poolside	1826–2268	·	·	·	N (BYO K/cat alog only)

Notes: Messages-native: Anthropic, TrustedRouter, Azure. Embeddings: OpenAI, Google AI Studio, Together, Tinfoil, Privatemode, Nscale, Lyceum, Jina, Cohere, Voyage. Media-only providers: Recraft, BFL, Decart, Krea, fal (image); LTX, Runway, Kling (video); ElevenLabs (speech). Provider geography is informational only (docs/provider-geography-api.md).

6. frontend/ and front-end toolchain

ITEM	DETAIL
Source	<code>frontend/src/</code> — 4 files, ~919 LOC: <code>dashboard.ts</code> (sign-in modal + MetaMask SIWE against <code>/v1/auth/wallet/*</code> , <code>dashboard.ts:1-11</code>), <code>leaderboard.ts</code> (filter/sort/paginate), <code>token-index.ts</code> (NYTE token-index SVG charts, progressive enhancement), <code>globals.d.ts</code>
Build	<code>npm run build = tsc</code> ; compiles <code>frontend/src/**/*.ts</code> → <code>src/trusted_router/static/</code> (<code>tsconfig.json</code> : <code>rootDir</code> , <code>outDir</code> , <code>module: none</code> , <code>target es2020</code> , <code>strict</code> , <code>noUnused*</code>) — outputs are plain scripts, no bundler
Lint	<code>npm run lint</code> → ESLint flat config <code>eslint.config.mjs</code> (typescript-eslint, no-explicit-any warn, eeqeq, prefer-const; ignores <code>build static/dashboard.js</code>); <code>npm run lint:css</code> → stylelint (<code>.stylelintrc.json</code> , <code>stylelint-config-standard</code>) over <code>src/trusted_router/static/*.css</code>
Tests	<code>npm run test:browser</code> → Playwright (<code>playwright.config.js</code> : boots uvicorn with memory store on :18081, <code>testDir tests/browser</code>); separate <code>/chat</code> suite in <code>tests-e2e/</code> (own <code>package.json</code> , <code>playwright.config.ts</code>)
Other	<code>build:world-map</code> → <code>scripts/build-world-svg.mjs</code> (d3-geo + world-atlas); <code>alpinejs</code> remains a devDependency but dashboard is "No Alpine. Plain DOM."
Most UI JS	Hand-written JS/CSS directly in <code>src/trusted_router/static/</code> (75 entries: <code>chat.js</code> , <code>console.js</code> , <code>custom_models.js</code> , <code>earnings.js</code> , ...) — not compiled from TS
Deploy trigger	<code>frontend/src/**</code> changes auto-deploy the control plane (AGENTS.md:52-55)

7. sites/ — Token Exchange static sites

`sites/token-exchange/`: Python generator (`build.py`, `template.html`, `markets.json`), per-region evidence (`evidence-
{dubai,europe,london,tokyo,shared-gcp}.json`, `refresh_evidence.py`, `stage_evidence.py`), press kit (`press.*`, `press-assets/`), member badges (`badge.*`, `member_badges.py`), social images (`social.py`, `social-images/`), `deploy.py` (GCS + URL-map), Node verifiers (`verify*.cjs`), and pytest files (`test*.py`). See §4.

8. Other non-core directories at a glance

DIR	CONTENTS
<code>clickhouse/</code>	23 numbered SQL migrations (<code>001_provider_benchmark_samples.sql</code> ... <code>023_activity_generation_id_projection_replicated.sql</code> , each with replicated + single-node variants); Python jobs for outbox ingest (<code>ingest_outbox.py</code> , <code>ingest_operational_outbox[_postgres].py</code>), rollups (<code>rollup_analytics.py</code> , <code>rollup_client_events.py</code> , <code>rollup_synthetic.py</code> , <code>rollup_reservation_overruns.py</code>), archive/restore (<code>archive_daily.py</code> , <code>verify_archive*.py</code>), public snapshots (<code>build_public_snapshots.py</code>), freshness checks, workspace directory refresh; and ~25 systemd <code>tr-clickhouse-*.service/.timer</code> units. Runbook: <code>docs/clickhouse-reliability.md</code> .
<code>experiments/lightning_router/</code>	Separate BUSL app for <code>lightningrouter.ai</code> Lightning funding (own pyproject, Dockerfile, cloudbuild, Playwright tests, <code>web/</code>)
<code>interviews/</code>	Founder interview transcripts and cleaned excerpts (2026-05-04: security-trust, positioning, objections, customer-migration); raw audio not committed (<code>interviews/README.md</code>)
<code>release/</code>	<code>first-window-plan.json</code> + schema: P6 async-settle-shadow first-window release pin template (router + enclave SHAs/images, REVIEW placeholders)
<code>skills/trusted_router_model_advisor/</code>	Pointer <code>SKILL.md</code> to canonical <code>Lore-Hex/LLM-advisor</code> skill (model selection advisor)
<code>fastpath/</code>	Go fast-admission service (<code>cmd/{fastpath,fastpathctl,fastpathwatch,loadgen}</code> , <code>internal/</code>) shadowing TLA+ specs
<code>proofs/</code>	TLA+ specs + configs (AuditorCommit, CreditDebt, KeyCapFence, LeaseLifecycle, AutoRefillHandoff, ...) and mutation checker
<code>infra/</code>	Terraform for static plumbing (Cloud Armor, LB certs, Spanner, fastpath, SES, AWS alarms, WIF)
<code>tests/</code> , <code>tests-e2e/</code>	pytest suite (incl. <code>tests/conformance/</code>), and <code>/chat</code> Playwright suite

9. Repo directory map (top level)

PATH	PURPOSE
<code>src/trusted_router/</code>	FastAPI control plane (routes, services, storage backends, catalog, billing, templates, static)
<code>tests/</code>	Python tests; conformance suite is the Store contract
<code>tests-e2e/</code>	Playwright <code>/chat</code> E2E
<code>frontend/</code>	TypeScript sources compiled into <code>src/trusted_router/static/</code>
<code>sites/</code>	Static Token Exchange microsites

PATH	PURPOSE
clickhouse/	Analytics schema, ingest/rollup jobs, systemd units
fastpath/	Go fast-admission service
proofs/	TLA+ formal models
infra/	Terraform
scripts/	Ops, deploy, pricing, smoke, backfill tooling (\$10)
docs/	Design docs, runbooks, incidents, product docs (\$11)
experiments/	LightningRouter funding app
interviews/	Founder interview material for marketing copy
release/	Release-plan pin templates
skills/	Agent skill pointer
settled	0-byte stray file (artifact)
fulltest.out, aws_ses_followup.md	Stray working files at root
AGENTS.md, CLAUDE.md, README.md, SECURITY.md, LICENSE (BUSL-1.1), NOTICE	Agent rules, overview, licensing
pyproject.toml, uv.lock, package.json, tsconfig.json, eslint.config.mjs, .stylelintrc.json, playwright.config.js, Dockerfile	Build/tooling config

10. scripts/ categorized (92 top-level entries)

CATEGORY	ENTRIES
Deploy & release (scripts/d eploy/, 124 files)	rollout.sh (config-as-code ENV_VARS, AGENTS.md:57-59), deploy-gcp.sh, endpoint.sh, publish-trust-gcs.sh, multi-cloud (aws_*, azure_*, cloud_rollout.py, cloud_*_gate.sh), ClickHouse cluster (clickhouse_*), Spanner migrations (migrate_*.sh), surfaces/edge (public_surface*.sh, internal_surface*.sh, gateway_edge*, service_surface_url_map.py), verification (verify_deployment.sh, verify_*_complete.sh, watchdog.py)
Smoke / health probes	smoke_e2e.py, smoke_all_providers.py, smoke_byok_prod.py, smoke_openai_sdk.py, prod_embeddings_probe.py, run_synthetic_monitor.py, provision_synthetic_monitor.py, validate_synthetic_monitor_key.py, verify_gateway_authorization.py, verify_trust_measurements.py
Pricing & catalog	pricing/ (refresh pipeline: providers, parsers, vertex, speech, transcription, video sources), ingest_openrouter_catalog.py, check_openrouter_coverage.py, check_price_coverage.py, check_price_spike.py, detect_provider_drift.py, classify_provider_routes.py, update_provider_throughput_rank.py, model_change_snapshot.py, update_model_changes.py, backfill_model_changes.py, check_request_capabilities.py, export_provider_check_contract.py, generate_public_openapi.py, import_tr_benchmarks.py, providers/sync_azure_foundry.py, reprobe_user_models.py
Billing / money ops (high risk)	grant_credit.py, check_credit_row.py, consolidate_starter_credit.py, expand_billing_shards.py, shard_workspace.py, settle_negative_shards.py, stress_credit_shards.py, cleanup_legacy_credit_json.py, check_no_v1_envelopes.py, audit_typed_counters.py, backfill_lifetime_topups.py, set_lifetime_topup.py, backfill_leased_key_usage.py, backfill_reservation_terminal_at.py, generate_usage_cost_vectors.py, record_usage_baselines.py, reconcile_app_markup_payouts.py, reconcile_custom_model_payouts.py, fastpath/credit_debt_vectors.py
Async settle / shadow	async_settle/ (16: shadow v2 collect/pack/activation/export, drain capacity, pilot enablement), provision_async_settle_shadow_probe_workspace.py, provision_fastpath_p1_workspace.py, provision_stage_d_probe_workspace.py
Trust / identity / payments	inspect_payment_owner.py (the only sanctioned Stripe ownership lookup, AGENTS.md:66-74), reconcile_provider_trust.py, set_trust_override.py, process_trust_demotion_remainders.py, reconcile_veriff_decisions.py, check_adyen_readiness.py, backfill_byok_aad_v2.py, provision_provider_access.py
Data hygiene / Spanner	audit_spanner_schema.py, tombstone_sweep.py, tombstone_sweep_plugin.py, cleanup_smoke_signups.py, check_cloudstaleness.py

CATEGORY	ENTRIES
Marketing / growth / assets	marketing/ (OG renderers, token-exchange browser smokes), axiom_growth/ (Axiom datasets, dashboards, alerts, onboarding validation; own Dockerfile/cloudbuild), marketing_funnel_report.py, build_google_ads_experiment_matrix.py, build_media_kit.py, build_social_rewards_preview.py, generate_blog_og.py, generate_decision_blog_og.py, generate_provider_og.py, generate_region_og.py, generate_company_signin_buttons.py, import_company_affiliations.py, provision_creator_pilot.py, find_gabriella.py
Evals	fusion_full_eval.py, fusion_live_eval.py, fusion_micro_eval.py
Frontend build/visual	build-world-svg.mjs, build_tokens_css.py, visual_review.mjs, check_format_ordering.py
Lightning	lightning/ (19: bitcoin/LND/Lexe/BTCPay node setup, web deploy)
CI / security	ci/merge_shard_durations.py, security/check_repo_security_coverage.py, ops/policies/

11. docs/ index

63 top-level files + 18 subdirectories. Categories: **Product** (user-facing feature docs), **Routing/Catalog, Billing/Payments, Trust/Security, Ops/Runbook, Analytics/Data, Design** (architecture proposals), **Growth/Marketing, Provider, Evidence** (dated snapshots).

11a. Top-level docs

DOC	CATEGORY	SUMMARY
image-generation.md	Product	Normalized /v1/images API; discover per-endpoint capabilities first
video-generation.md	Product	Async /v1/videos ; Seedance/LTX/Gemini Omni launch models; microdollar reserve/settle
text-to-speech.md	Product	/v1/audio/speech , OpenRouter-compatible, raw audio
transcription.md	Product	/v1/audio/transcriptions in attested gateway; audio parsed in memory only
web-fetch.md	Product	/v1/tools/fetch page → Markdown (DeepSeek Fetch)
chat-playground-controls.md	Product	/chat model settings (reasoning on/off/default)
custom-models.md	Product	tr-custom-model/{user}-{slug} : catalog model + hidden prompt
model-aliases.md	Product	nyte/ = trustedrouter/ ; auto-routing spellings
green-tokens.md	Product/Growth	trustedrouter/green energy-filtered pool
combo-models-september-2026.md	Product	Zeus/Prometheus/Plato/Socrates combo graph composition
scaledown.md	Product	ScaleDown task models (four specialized text tasks)
sign-in-with-trustedrouter.md	Product	OAuth 2.1 authorization server for apps
client-receipts.md	Trust	Enclave receipt signing-key discovery endpoints
routing-controls-20260927.md	Routing	Latency/throughput prefs, cache affinity, strict budget admission
provider-routing-evidence.md	Routing/Evidence	2026-10-09 seven-day provider prior snapshot
provider-geography-api.md	Routing	Geography metadata is informational only
request-contract-diagnostics.md	Routing	Chat/Responses option validation before billing auth
gateway-contract-warnings.md	Routing	Adapter rejection (400/422) warnings

DOC	CATEGORY	SUMMARY
openrouter-coverage.md	Routing	ROUTE_COVERAGE as OpenRouter API parity source
model-changes.md	Catalog	Public model change history workflow
cache-read-pricing.md	Billing	Cache-read pricing flow + per-provider audit (76 sources)
async-settlement-billing-v1.md	Billing	Frozen dormant billing v1 contract
async-settle-frozen-main-inventory.md	Billing	Frozen-main execution inventory (330 modules)
authorize-workspace-read-proof.md	Billing	Retired mutation proof for authorize read collapse
adyen-rollout.md	Payments	Adyen as dark prepaid funding rail
paypal-inbox-reconciliation.md	Payments	Refunded-uncredited PayPal captures
routable-payouts.md	Payments	Creator payouts via Routable
trust-provider-slice1b.md	Payments/Trust	PayPal/Adyen adverse trust facts
lightning-btcpay.md	Payments	BTCPay dashboard on tr-bitcoin-1
lightning-funding-production.md	Payments	LightningRouter funds USD credits; no prompts
lightning-lexe.md	Payments	Lexe backend migration prep
lightning-liquidity.md	Payments	Receiving liquidity (Lexe now)
lightning-receiving-node.md	Payments	Merchant receiving node, not routing
bitcoin-node.md	Payments	Bitcoin backend for LightningRouter
security.md	Security	OAuth app redirect/security notes
client-telemetry.md	Analytics	Client-observed reliability telemetry contract v1
clickhouse-reliability.md	Analytics/Ops	Canonical ClickHouse analytics runbook
clickhouse-sharding.md	Analytics	Sharding gate (1 shard × 3 replicas today)
clickhouse-synthetic-backfill.md	Analytics	Executed backfill of synthetic flag
status-leaderboard-evidence.md	Analytics	How status/leaderboard monthly views are computed
spanner-reliability.md	Ops	Spanner nam6 Enterprise Plus posture
runbook.md	Ops	Operator runbook keyed by symptom
runbook-github-outage.md	Ops	GitHub down vs repo gone
provider-independent-deploys.md	Ops	Provider health separate from release gates
speculation-protocol-v1.md	Design	Speculation wire contract (pure module)
speculation-protocol-v1-mutations.md	Design/Evidence	Mutation evidence for speculation protocol
speculation-shadow-runbook.md	Ops	Speculative invocation router shadow (flag off)
speculation-shadow-validation.md	Evidence	PR3 round 6 validation
speculation-shadow-mutations.json	Evidence	Mutation data
confidential-model-coverage-2026-10-15.md	Evidence	Confidential coverage (Qwen 3.8 27B listing)
provider-inference-locations-2026-09-26.md	Evidence	Physical execution-location evidence
venice-deepseek-september-2026.md	Provider	Venice DeepSeek Flash retirement migration
founder-interview.md	Growth	Founder interview workflow for copy
revenue-loop.md	Growth	Human-approved outreach loop, Google Sheet CRM

DOC	CATEGORY	SUMMARY
revenue-loop-google-sheet.csv	Growth	CRM column template
outreach-approval-packet.md	Growth	Per-lead approval packet
outreach-quote-bank.csv	Growth	Founder quote bank
social-approval-queue.csv	Growth	Social post approval queue
social-rewards.md	Growth	Signup + social action promo credit
openrouter-seo-audit.md	Growth	2026-06-02 SEO audit vs OpenRouter

11b. Subdirectories

DIR	CATEGORY	CONTENTS
storage-portability/ (9)	Architecture — read first (AGENTS.md:10-23)	HANDOFF.md, multi-cloud separation, analytics ingestion, AWS-EU/Azure canary, coverage map, Spanner emulator conformance & SQL inventory
design/ (46)	Design	See 11c
providers/ (16)	Provider	See 11d
incidents/ (25)	Ops	Postmortems 2026-09-05 ... 2026-10-18 (Spanner CPU from operator search, credit rebalance convoy, grant storm, fastpath lease locks, ...)
runbooks/ (12)	Ops	Fastpath P0/P1/alerts/deploy/key-rotation/turn-off/watch, DDoS edge, gateway billing edge, identity-verification support, SES notices/reputation
operations/ (6)	Ops	Service-surface bulkheads, public/internal surface splits, AllyRouter/UptimeRouter aliases, Fireworks retirements
marketing/ (17)	Growth	Ahrefs/Google Ads experiments, attribution, Axiom growth, OG spec, launch assets, token-exchange, creator-program
homepage/ (5)	Growth	Homepage mockup audit, QA, evidence, social card
blog-drafts/ (8), blog-evidence/ (1)	Growth	Blog drafts ("attestation is all you need", etc.) + evidence JSON
brand/ (3)	Growth	Brand tokens JSON, content-layout, screenshots
runs/ (30)	Growth	Revenue-loop approval packet runs
compliance/ (2)	Trust	hipaa/, soc2/ readiness
security/ (1)	Trust	edge-surfaces.json
evals/ (2)	Evidence	Fusion DRACO eval notes + live results
analytics/ (1)	Analytics	Onboarding attempts
latency/ (1)	Design	RPC pause fold C3
pr-previews/ (2)	Growth	Docs phone layout, homepage v2 previews
token-exchange/ (1)	Growth	hero.jpg asset

11c. docs/design/ (46)

THEME	DOCS
Async settle & shadow (14)	async-settle-outbox-v1, async-settle-shadow-v1, async-settle-shadow-v2 (+ -activation, -bundle, -bundle-round{2,3,6,7}, -live-export, -p5-inputs, -process-inventory, -runtime-harness-interface, -startup-harness-router) — dormant/offline evidence pipeline to certify async settle
Fast admission (8)	fast-admission-and-batched-settlement (approved v25), -enclave-direct (take Python off the hot path), -spike, -tracecheck, -pilot, -p1-results, -production-rollout

THEME	DOCS
Billing ledger (9)	billing-typed-counters, billing-dual-system-deletion-handoff, retire-json-credit-ledger, credit-row-sharding-handoff, key-usage-row-sharding, durable-settle-outbox, settle-one-read, rpc-diet-c1, trust-tier-incremental
Payments/id entity (4)	x402-stablecoin-funding, oauth-scopes-and-app-economy, company-affiliations, byok-aad-v2-migration (+ byok-aad-v2-attestations.json)
Catalog/routing (7)	model-capabilities, request-tagging, polyphemus-1.0, provider-latency-probe, catalog-refresh-2026-09-14, catalog-refresh-2026-09-22, together-september-2026-retirements
Infra (3)	clickhouse-high-availability, two-cloud-rollouts, reverse-harness-contract
Asset	shared-header-widths-2026-10-08.png

11d. docs/providers/ (16)

abliterate.md, boundless-zro-hyperbolic.md, byteplus.md, confidential-ai.md, confidential-ai-e2e-audit-2026-09-08.md (partial; do not enable E2E filter), greenference.md, meta.md, microsoft-decision-1.md, telnyx-native-catalog.md, tencent-tokenhub.md, deepseek-v41-september-2026.md, near-ai-2026-09-11-recovery.md, near-ai-deepseek-september-2026.md, 2026-09-20-alert-repairs.md, 2026-09-20-gemini-vertex-discovery.md, 2026-09-20-phala-native-discovery.md.

12. Gotchas for newcomers in this area

1. Media/chat handlers in this repo are stubs or metadata; behavior lives in `quill-cloud-proxy`. Change capability lists here (`image_generation.py`, `speech.py`, `transcription.py`, `web_fetch.py`) and the gateway must agree; drift is called out as a security bug (`image_generation.py:3-6`).
2. `trustedrouter/auto` has **no** privacy floor; only `zdr / e2e / confidential` (or `provider.min_privacy`) enforce one (`catalog_data.py:2525-2532`).
3. Provider privacy flags default to "assume stored" (`catalog_data.py:35-41`). Never set ZDR/CC/E2EE without a citation.
4. Most static JS is hand-written in `src/trusted_router/static/`; only 3 files compile from `frontend/src`. Do not edit `static/dashboard.js` (build output, ESLint-ignored).
5. Token Exchange market additions must follow `sites/token-exchange/ADDING-MARKETS.md` (AGENTS.md:5-8).
6. Root contains stray artifacts (`settled 0 bytes`, `fulltest.out`, `aws_ses_followup.md`).

08 Security & Trust

3 · Privacy tiers

stronger ↑

confidential TEE + E2EE + ZDR (Tinfoil, Phala)

zdr contractual zero data retention

no_store provider does not store content

standard default · assume stored

```
"provider": {"min_privacy": "zdr"}
```

- ✓ route A flagged zdr
- ✓ route B confidential
- ✗ route C no flag → assume stored

no route qualifies? fail closed before dispatch

min_privacy sets a hard floor: no_store, zdr, or confidential.

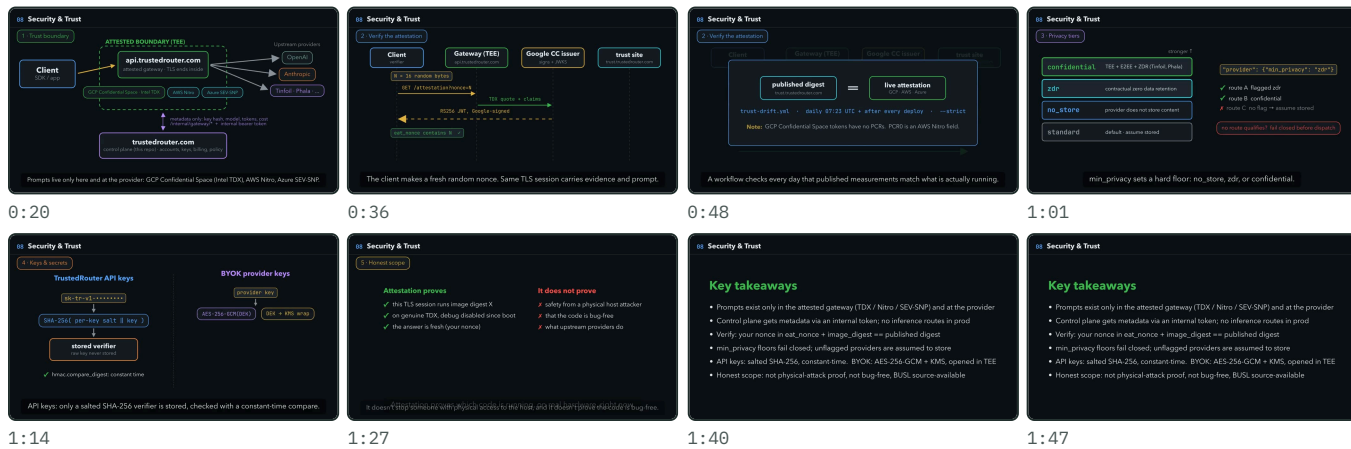
PART III · CHAPTER 07 · HERO STILL: VIDEO 08 AT 1:01

Security & trust

Attestation, the trust boundary, privacy tiers, key storage, BYOK and the threat model.

- 1. The one-paragraph model
- 2. Honest scope (read this before you say anything to a customer)
- 3. Trust-boundary diagram
- 4. Attestation: what a real response looks like
- 5. Client receipts (signed per-request evidence)
- 6. Privacy tiers & fail-closed routing
- 7. "Never logged": how prompt/output privacy is enforced in this repo
- 8. Credentials & secrets
- 9. Edge, WAF, rate limits, body limits
- 10. Supply chain & security workflows
- 11. License (BUSL-1.1) implications
- 12. Security controls → implementation → enforcing test

Storyboard · Video 08: Security & trust



Research pack for the "0-to-CTO" onboarding. Scope: how TrustedRouter's privacy claim is built, what it does and does not prove, where every control lives in `quill-router`, and which test keeps it honest. Researched against `quill-router@778120494` (2026-10-19) plus read-only `gh api` reads of `Lore-Hex/quill-cloud-proxy`. Paths are relative to `quill-router/` unless prefixed `quill-cloud-proxy:`.

1. The one-paragraph model

TrustedRouter is split into two planes, and the split is the security architecture:

- **Attested gateway / prompt path** (`api.trustedrouter.com`, alias `api.quillrouter.com`). A Go binary (`quill-cloud-proxy:enclave-go/`) runs inside GCP Confidential Space on Intel TDX. AWS Nitro Enclaves and Azure SEV-SNP Confidential Containers serve as additional planes. TLS terminates **inside** the workload. Prompts and outputs exist only here and at the upstream provider.
- **Control plane** (`trustedrouter.com`, this repo, FastAPI on Cloud Run). It handles accounts, keys, billing, routing policy, catalog, the trust page and metadata. In production it **never sees prompt or output content**. It does not even register inference routes (`src/trusted_router/main.py:880`, `:912`).

The gateway calls the control plane's `/internal/gateway/*` routes (`authorize`, `settle`, `refund`) with an internal bearer token. Those calls carry only metadata: key hash, model, token counts and cost. Customers get evidence by fetching a Google-signed attestation JWT bound to their nonce **and** to the TLS session. They compare the container `image_digest` against the published release on `trust.trustedrouter.com`.

2. Honest scope (read this before you say anything to a customer)

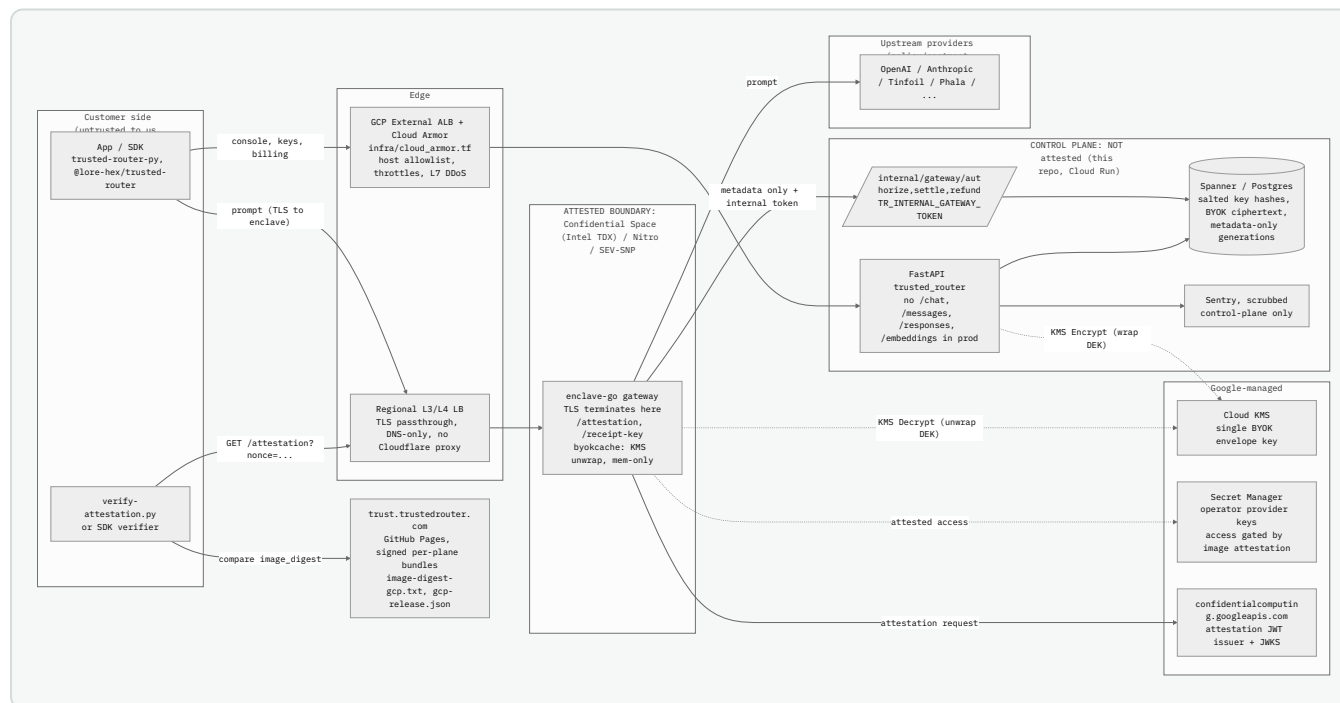
From `README.md:109-114`, `docs/security.md`, `quill-cloud-proxy:tools/verify-attestation.py`, and `docs/compliance/soc2/README.md`:

WHAT ATTESTATION DOES PROVE	WHAT IT DOES NOT PROVE
The container serving <i>this</i> TLS session has image digest X, running on genuine TDX hardware with secure boot and debug disabled since boot (<code>dbgstat: disabled-since-boot</code>)	That the open-source code is bug-free or free of a logging bug
The response is fresh, because your random nonce is in <code>eat_nonce</code>	Anything about the upstream provider (OpenAI, Anthropic, ...). Provider handling is policy or contract, cited per provider in the catalog. Only the <code>confidential</code> tier adds provider-side TEE + E2EE
The cert you are talking to is bound into the token: workload TLS cert + RFC 9266 tls-exporter hashes in <code>eat_nonce</code>	Resistance to a nation-state with physical host access, or a break in Google, Intel or AMD roots of trust
Secret Manager / KMS access is gated on the image (change one line, get a new digest, and secret access fails)	The control plane . It is not attested. It is trusted for billing, metadata and BYOK ciphertext storage only
	On Azure, MAA re-attests the SNP report, so Microsoft is a trusted third party. Receipt keys from AWS and Azure are published <code>verified=false</code> (<code>docs/client-receipts.md</code>)

Compliance status (`docs/compliance/soc2/README.md`): SOC 2 Type I and II are *not obtained*. HIPAA, ISO 27001 and privacy certifications are *not obtained*. The DPA and BAA are drafts. The repo contains a readiness binder (`docs/compliance/soc2/` , `docs/compliance/hipaa/`), not a report. Lore Hex Corp has one person. `.github/CODEOWNERS` says so explicitly and leans on attested measurements as the control that doesn't depend on headcount.

Batch API is the documented exception to "no retention". It stores per-artifact AES-256-GCM ciphertext with Cloud KMS-wrapped DEKs and deletes after 30 days (`docs/security.md` "Prompt And Output Data"; `quill-cloud-proxy/docs/design/batch-api.md`). This guarantee is weaker than zero-retention: it depends on deployed KMS/IAM policy.

3. Trust-boundary diagram



The dashed/colored line that matters is **TEE ↔ CP**. Content crosses into the TEE and to providers, never into CP. CP crosses into the TEE only with ciphertext envelopes and policy.

4. Attestation: what a real response looks like

I fetched `GET https://api.trustedrouter.com/attestation?nonce=<16 random bytes hex>` on 2026-10-19.

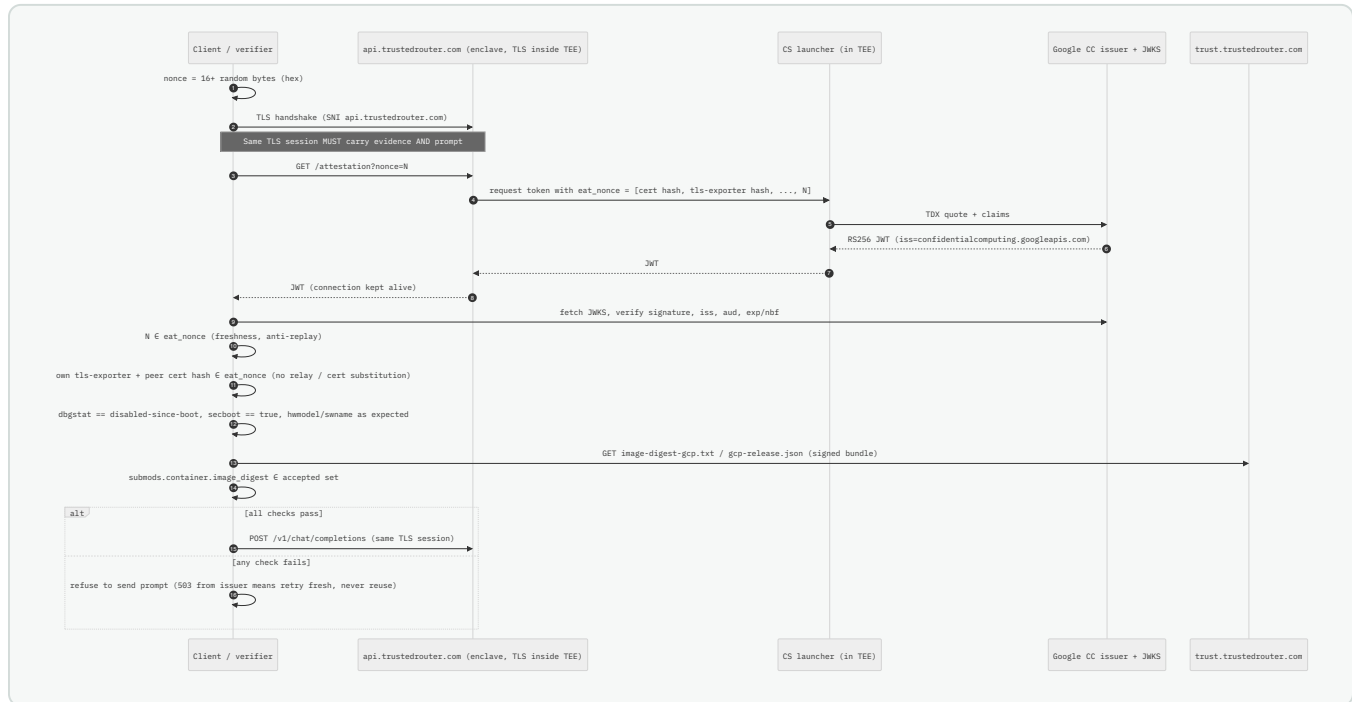
(Disclosure: called twice. The first output was truncated, so I re-fetched to save the token.) The body is a **raw RS256 JWT, not JSON**. Decoded claims, trimmed:

```
// header
{"alg": "RS256", "kid": "1be6ff7287d640316ddc929e1a8b21fbca7e54e7", "typ": "JWT"}
// payload
{
  "aud": "quill-cloud",
  "iss": "https://confidentialcomputing.googleapis.com",
  "sub": "https://www.googleapis.com/compute/v1/projects/quill-cloud-proxy/zones/us-central1-b/instances/quill-enclave-mig-us-j71s",
  "iat": 1792432251, "exp": 1792435851,
  "eat_nonce": [
    "680d068a...e63f", // binding hash (workload TLS cert / tls-exporter, per gcp-attestation-issuance.md)
    "0a8633b2...ff00", // binding hash
    "223d263e...5096", // binding hash (differs per TLS session)
    "6794d51294742e4b2d85989c38d20dd6" // <- MY nonce
  ],
  "hwmodel": "GCP_INTEL_TDX", "swname": "CONFIDENTIAL_SPACE", "swversion": ["260900"],
  "secboot": true, "dbgstat": "disabled-since-boot",
  "submods": {
    "confidential_space": {"support_attributes": ["LATEST", "STABLE", "USABLE"], "monitoring_enabled": {"memory": false}},
    "container": {
      "image_reference": "us-central1-docker.pkg.dev/quill-cloud-proxy/quill/enclave-multi:gcp-release-8cad0a1e",
      "image_digest": "sha256:28ada804dd8befa51e0bc98732e4fd6f211b175ef15bfe08b74afe34ebbf6e9f",
      "image_id": "sha256:9f076a33...9939",
      "restart_policy": "Always",
      "env": { "QUILL_ANTHROPIC_SECRET": "trustedrouter-anthropic-api-key", "...": "131 entries; secret *names*, never values" }
    }
  }
}
```

Doc bug worth fixing: the README's "For the nerds" snippet (`README.md:93-99`) is wrong in three places. 1. It pipes to `jq .`, but the body is a JWT, so `jq` fails. 2. It refers to a top-level `image_digest`, but the claim lives at `submods.container.image_digest`. 3. It lists `pcrs`, but GCP Confidential Space tokens have no PCRs. PCR0 exists on the AWS Nitro COSE document.

The env block also tells you that the operator provider keys are **Secret Manager references** that the attested image resolves at boot.

4.1 Client verification sequence



Key subtleties, from `quill-cloud-proxy:tools/verify-attestation.py` and `docs/gcp-attestation-issuance.md`:- **Exporter binding alone is insufficient**. A relay could launder the client's exporter through the nonce channel. The enclave accepts only *one* caller nonce, so you must send your own random nonce *and* check the exporter. - A new TLS connection needs fresh evidence. No nonce-bound evidence is cached server-side. Issuance is bounded to 1 active mint, 4 waiters, a 5 s queue and a 30 s issuer timeout. Temporary failures return 503 and must not be bypassed.

4.2 Trust drift workflow

- `scripts/verify_trust_measurements.py` compares every **published** measurement on the control plane's trust records against a **live** attestation from GCP `api_base_url`, AWS, and every Azure `regions[].attestation_url`. It exits non-zero on mismatch or on an un-contacted endpoint when run with `--strict`.
- `.github/workflows/trust-drift.yml` runs it daily at 07:23 UTC and after every successful "Deploy TR control plane". It uses `--strict` and never gates PRs on prod state. Offline proofs live in `tests/test_trust_measurement_drift.py` (e.g. `test_gcp_digest_drift_fails_and_prints_both_values`, `test_gcp_matches_the_accepted_set_during_a_roll`).
- Why it exists: the Quill trust bucket once published a PCR0 that matched no running enclave. A stale measurement teaches verifiers to ignore mismatches.
- Publishing happens in `quill-cloud-proxy`. `make gcp-release` writes `trust-page/image-digest-gcp.txt` and `gcp-release.json`. Per-plane `publish-trust-{gcp,aws,azure}.yml` sign bundles. `publish-trust-page.yml` deploys GitHub Pages at `trust.trustedrouter.com`, with an S3 mirror at `trust.quill.lorehex.co`. Publishers refuse unsigned or mismatched copies (`tools/check-trust-signatures.py`, `tools/check-trust-copies.py`).
- The control-plane side of the trust records is `src/trusted_router/trust.py` (`gcp_release :69`, `aws_release :149`, `azure_release :268`, `trust_html :338`).
- *Inconsistency*: `docs/security.md` says `trust.trustedrouter.com` "should point at the control-plane/trust hosting", but the gateway README says it is GitHub Pages. Confirm before you document it externally.

5. Client receipts (signed per-request evidence)

`docs/client-receipts.md`, `src/trusted_router/receipt_keys.py`, `src/trusted_router/routes/public.py:2511`.

- Each enclave generates an **Ed25519 receipt signing key at boot**. Receipts (compact JWS) go **only to the requesting client and are never stored**.
- The enclave publishes the public key at `/receipt-key`, committed into its attestation:
- `commitment = sha256(domain || 0x00 || pubkey)` (`receipt_keys.py:99`)
- A scheduled collector (`POST /v1/internal/gateway/receipt-keys/collect`, internal token) connects to every gateway A-record by IP with correct SNI, then stores `(kid, att_sha256)` rows. Rows are append-only: a kid collision never overwrites key material. This was fixed after the 2026-09-10 version-retention defect.
- Public discovery is at `GET /.well-known/inference-receipt-keys` or `/trust/receipt-keys.json`, paginated (≤ 250 versions, ≤ 1 MiB).
- `verified=true` currently means the GCP CS JWT signature, issuer, validity, audience and non-debug state were checked (`verify_gcp_attestation_chain`, `receipt_keys.py:437`, `dbgstat check` at :497). AWS and Azure rows are `verified=false` until in-package chain verifiers exist. **Clients must not treat `verified=false` as hardware-anchored**.
- Receipt JWS verification for clients is implemented in `quill-cloud-proxy:tools/verify-attestation.py` (`verify_receipt_*`, `verify_receipt_stream`).

6. Privacy tiers & fail-closed routing

Vocabulary (`src/trusted_router/catalog_data.py:61-67`):

TIER	IN T	MEANING
<code>standard / any</code>	0	No tracked posture. Assume stored . This is the default for every provider until a cited flag says otherwise
<code>no_store</code>	1	Provider does not store request/response content
<code>zdr</code>	2	Contractual/policy zero data retention
<code>confidential</code> (aliases <code>e2e</code> , <code>e2ee</code>)	3	Verified confidential compute and provider E2EE and explicit ZDR and host is not the model's own vendor (currently Tinfoil, Phala)

- Request surface: `provider.min_privacy` is parsed at `src/trusted_router/routing.py:694-708`. Unknown values return 400. Meta-models `trustedrouter/zdr`, `trustedrouter/e2e` and `trustedrouter/eu` set floors.
- Enforcement is `endpoint_meets_privacy_requirement` (`src/trusted_router/catalog_privacy.py:231`). It is **not** a numeric comparison: retention and confidential compute are orthogonal, and unknown returns `False` (fail closed). Applied in `_apply_provider_filters` (`routing.py:1121`), which also notes "A TEE route with no tracked retention promise must not satisfy deny/ZDR".
- Floors compose: the resolved floor dominates everything requested, regardless of order. If no endpoint qualifies, the request **fails before dispatch** rather than degrading.
- Gateway replay of stored routes re-checks the floor (`tests/test_gateway_replay_privacy.py`).
- The TR gateway hop itself is always attested. These tiers describe only the **upstream** provider (`catalog_privacy.py` module docstring).
- Broadcast observability is metadata-only by default. Content export requires explicit `include_content`, and content-enabled encrypted destinations are returned only to the attested gateway (`README.md` "Broadcast Observability").

7. "Never logged": how prompt/output privacy is enforced in this repo

Defense in depth, from structural to cosmetic:

1. **Structural: the control plane cannot receive prompts in prod.** `register_inference_routes` runs only when `_control_plane_inference_enabled` (`main.py:880`, `:912-919`), which returns true for `environment` in `{local, test}` only. An outage cannot silently fall back to a non-attested handler.
2. **Contract: gateway↔control-plane calls carry metadata only.** Authorize, settle, refund, activity and generation metadata carry no content (`docs/security.md` "BYOK Secrets"). Generation rows are limited to id, workspace, key hash, model, provider, tokens, cost, usage type, speed, finish reason and status.
3. **No content-storage switch.** `content_storage_enabled: true` is rejected with `content_storage_disabled`. `GET /generation/content` always returns `content_not_stored` (`src/trusted_router/types.py:118`).
4. **Sentry is control-plane-only and scrubbed** (`src/trusted_router/sentry_config.py`). - `send_default_pii=False`, `max_request_body_size="never"` (`:240-241`). - `before_send`, `before_send_log` and `before_breadcrumb` (`:301, :315, :326`) recursively `_scrub` (`:563`). - `SENSITIVE_KEYS` (`:21`) covers authorization, cookie, api_key, byok, secret, token, password, messages, prompt, completion, content, output, input, body, and more. - Also scrubbed: string fragments/prefixes such as `sk-tr-v1-` (`:47, :83`). - A floodgate caps events at 3 per fingerprint and 50 per hour per process. - **No Sentry in the enclave image.**
5. **Logging hygiene tests** (`tests/test_application_logging.py`, `tests/test_scrub_totality_property.py`). Hypothesis property tests check that no declared secret survives any container shape, scrubbing is idempotent, cycles terminate and the Axiom filter ships no canary.
6. **The gateway retention table** (`quill-cloud-proxy` README "What gets retained"): no prompt, completion, bearer tokens or key hashes. Only per-device daily aggregate counts.

8. Credentials & secrets

8.1 API keys

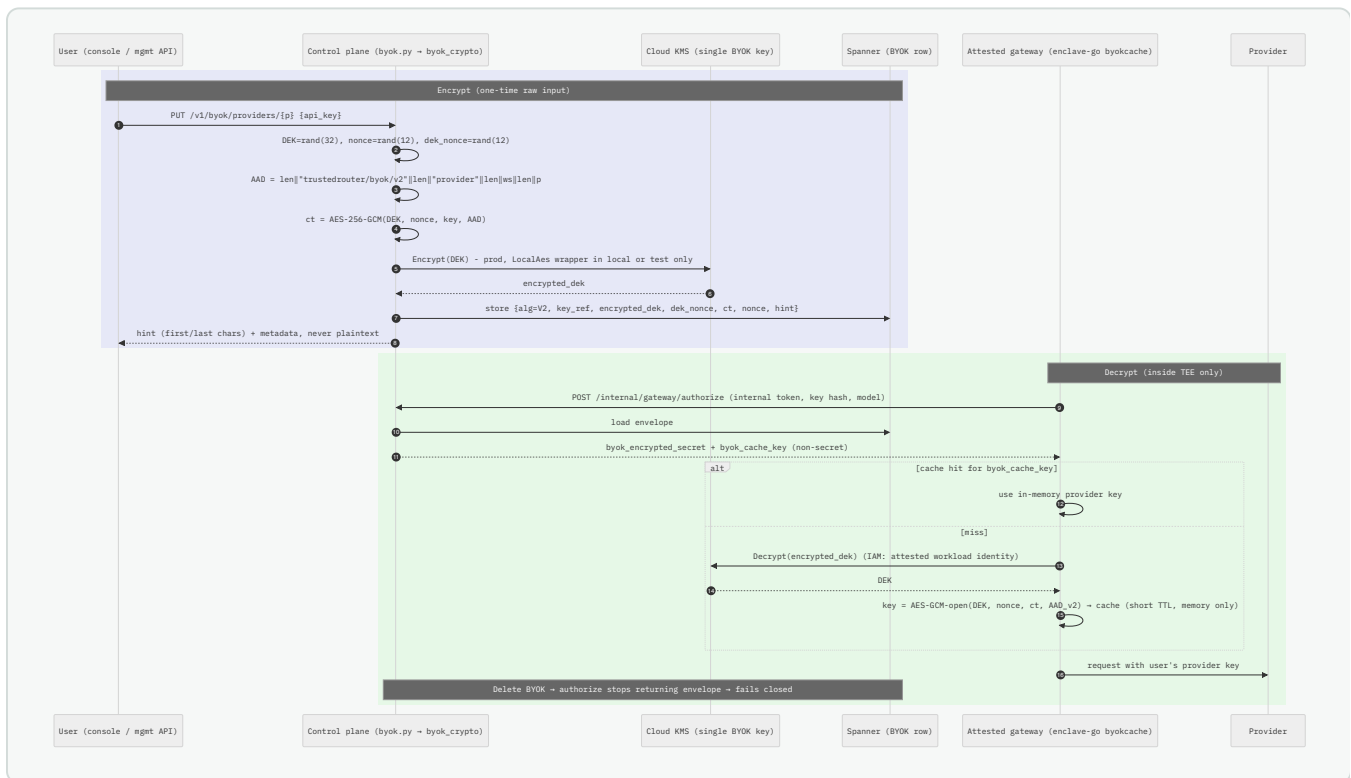
`src/trusted_router/security.py`:-Key format: `sk-tr-v1-<token_urlsafe(32)>` (`:8`). Key ID: `key_<token_urlsafe(18)>` (`:12`).-Verifier: per-key random 32-byte salt, digest `SHA-256(salt || key)` (`:16-21`), compared constant-time (`hmac.compare_digest`, `:28, :38`).-Lookup index: `lookup_hash_api_key = SHA-256(key) unsalted` (`:24`). This is acceptable because keys carry ~256 bits of entropy (no dictionary attack). It gives O(1) lookup, and the salted digest remains the verifier. Sessions use the same lookup-hash scheme (`storage_auth_sessions.py:48`).-The public `hash` field is an opaque key ID, *not* the verifier (`docs/security.md` "API Keys"). Tested by `tests/test_stubs_and_security.py:298 test_api_key_secrets_are_salted`.

8.2 BYOK envelope encryption

`src/trusted_router/byok_crypto.py`, `src/trusted_router/key_management.py`,
`src/trusted_router/routes/byok.py`.

- `encrypt_byok_secret` (`:52`):
- Generates a random 32-byte DEK and a 12-byte nonce.
- Encrypts the provider key with `AES-256-GCM(DEK)`.
- Wraps the DEK through `KeyWrapper` (Cloud KMS Encrypt in prod) with a separate 12-byte nonce.
- Stores only `{algorithm, key_ref, encrypted_dek, dek_nonce, ciphertext, nonce}` plus a first/last display hint.

- **AAD v2** (:285) is length-prefixed and injective: `"trustedrouter/byok/v2" | namespace | workspace_id | context`. It binds ciphertext to the workspace and provider. Three namespaces keep secret families from opening as one another:
- `provider`: BYOK, opened only by the enclave
- `control`: broadcast keys/headers, never shipped to the enclave
- `user_model`: opened on both sides, deliberately
- It must stay byte-identical to `quill-cloud-proxy:enclave-go/internal/byokcache/cache.go aadV2`. Both repos pin a hex vector (`tests/test_byok_aad_namespace_property.py::test_the_v2_vector_matches_the_enclave`).
- V1 envelopes are rejected before any unwrap (:306-312).
- Wrapper selection (`key_management.py:180`): an explicit local key, then `TR_BYOK_KMS_KEY_NAME` (`GcpKmsKeyWrapper`, :113), then an HKDF-derived dev key **only in local/test**, else raise. There is no silent dev-key fallback in prod.
- One KMS key wraps every BYOK DEK. BYOK keys are rows, not one Secret Manager secret each.
- `byok_cache_key` (:215) = `"byokcache:v1:" + sha256(workspace, provider, all envelope fields)`. The gateway uses it for short-TTL, memory-only plaintext caching. Rotation produces a new key. Delete means authorize stops returning the envelope, and the cache entry expires by TTL.
- Returned to the gateway in `/internal/gateway/authorize` (`routes/internal/gateway.py:5528-5529`).



Operators can prove the full contract with `TR_BYOK_PROD_SMOKE=1 uv run python scripts/smoke_byok_prod.py`. It is **destructive and manual, for security releases only**. It covers an isolated workspace, rotation, settlement, delete, and fail-closed afterwards.

8.3 Sessions, OAuth, Sign in with TrustedRouter

- **Principal resolution** (`src/trusted_router/auth.py:90`) tries three sources in order: 1. `x-trustedrouter-user` dev header, rejected outside local/test (:185) 2. `tr_session` cookie 3. Bearer, dispatched strictly by prefix: `sk-tr-` means API key only, `trsess-` means session only, anything else returns 401. There is no cross-shape fallback.

- **Session cookie** (`auth.py:378`): `HttpOnly`, `SameSite=Lax`, `Secure` in every deployed env. The companion `tr_signed_in=1` hint cookie is deliberately non-`HttpOnly` and carries no auth power.
- **Login OAuth** (Google/GitHub) is in `src/trusted_router/oauth_provider.py`. GitHub picks the primary *verified* email (:181). The OAuth state cookie is `HttpOnly/Secure/Lax` (`tests/test_session_cookie_attributes.py:91`).
- **TrustedRouter as OAuth 2.1 AS** (`docs/sign-in-with-trustedrouter.md`, `docs/security.md`):
 - Redirect URLs are exact-match.
 - PKCE S256 is mandatory on `/v1/oauth/token`. The verifier is 43-128 unreserved chars.
 - Codes and consent are short-lived and single-use.
 - Consent is server-bound: approval posts only an opaque id plus a constant-time-compared CSRF token.
 - Scopes: `inference profile balance:read activity:read`.
 - The default app budget is \$20/mo. A suggested budget is never auto-applied above \$20.
 - The token endpoint has a 30/min per-IP bucket.
- **Nuance**: the deprecated legacy `/v1/auth/keys` path still accepts `plain` PKCE (`tests/test_oauth_key_delegation.py:189`, `tests/test_oauth_increment_d.py:196`).
- **Internal tokens** (`src/trusted_router/routes/internal/_shared.py:30-67`): path-scoped credential classes `gateway`, `observer`, `operator` and `lightning`, each with its own secret, compared in constant time.
- Prod returns 403 if the token is unset.
- Config refuses to start if a surface holds a secret it doesn't own (`SERVICE_SURFACE_SECRET_OWNERS`, `config.py:247-257`, check at :1981).
- Federation directory, settlement and credit secrets are valid only on their exact routes.

9. Edge, WAF, rate limits, body limits

- **Cloud Armor** (`infra/cloud_armor.tf`, owner-applied Terraform, sole writer). Rules:
 - 900: deny non-canonical Host → 403
 - 1000: `/chat-proxy/` throttle 120/60 s
 - 1100: state-changing request throttle 300/60 s
 - 1150: throttle 12000/60 s
 - 1200: throttle 2400/60 s
 - L7 DDoS defense is enabled
- `scripts/deploy/_edge_security.sh` reconciles non-Terraform policies. Inventory lives in `docs/security/edge-surfaces.json`. It has 7 surfaces across GCP, AWS and Azure, and `live_state_verified: false`. Several are `p0_external` or `p0_integration`. Treat the edge rollout as **in progress**.
- **AWS WAF / App Runner** helpers are `scripts/deploy/_aws_waf.sh` and `_aws_app_runner_security.sh` (`tests/test_ddos_edge_deploy.py`).
- `api.trustedrouter.com` **must be DNS-only** (no Cloudflare proxy) so TLS reaches the TEE.
- **App limiter** (`src/trusted_router/request_limits.py`):
 - Process-local, never storage-backed.
 - Defaults to `TR_RATE_LIMIT_CLIENT_IP_MODE=untrusted`: one shared `untrusted_lb` bucket.
 - `X-Forwarded-For`, `CF-Connecting-IP` and `X-TrustedRouter-User` are ignored. `edge_header` mode is allowed only behind the GCP ALB, which overwrites `X-TrustedRouter-Client-IP`.
- **Body limits** (`src/trusted_router/request_body_limit.py:79`): 4 MiB per request, 64 MiB in flight, 16 concurrent bodies, 30 s read deadline. CL+TE smuggling framing is rejected.
- **Headers** (`src/trusted_router/middleware.py:345`): HSTS, `X-Frame-Options: SAMEORIGIN`, and a nonce-based CSP that is still **report-only** (`tests/test_security_response_headers.py:61`).

10. Supply chain & security workflows

WORKFLOW	WHAT
<code>.github/workflows/dependency-scan.yml</code>	OSV-Scanner. Full scan with SARIF to code-scanning on push/daily. PR mode diffs base vs head. Actions are pinned by SHA
<code>.github/dependabot.yml</code>	Weekly updates for uv, npm, docker and github-actions
<code>.github/workflows/trust-drift.yml</code>	Published measurement vs live attestation (\$4.2)
<code>.github/workflows/cloud-security-baseline.yml</code>	Daily read-only GCP CIS-style drift check (<code>scripts/deploy/gcp_security_baseline_check.sh</code>). Opens an issue on drift. The checkers never mutate (<code>tests/test_cloud_security_baseline.py</code>)
<code>.github/workflows/check-repo-security-coverage.yml</code>	Weekly check of branch protection and security settings across public Lore-Hex repos (<code>scripts/security/check_repo_security_coverage.py</code>)
<code>.github/workflows/typed-audit.yml</code> , <code>ci.yml</code>	ruff + mypy + pytest (coverage ≥70%) gate every change
<code>SECURITY.md</code>	GitHub private vulnerability reporting. Ack within 72 h, assessment within 5 business days. No bounty. Safe harbour. Customer-content exposure is top severity and notification does not wait for RCA

11. License (BUSL-1.1) implications

LICENSE : Business Source License 1.1. Licensor is Lore Hex Corp. **Additional Use Grant: None**. Change License is Apache-2.0, four years after each version's publication. - Anyone may read, copy, modify, and make **non-production** use. Running it in production (e.g. a competing hosted router) requires a commercial license until the change date. - The security relevance is that "source-available, not OSI open source" is the honest phrasing. It still gives verifiers what they need: the ability to read and rebuild the attested gateway (`quill-cloud-proxy` is also BUSL-1.1) and compare digests. SDKs are Apache-2.0 / MIT. Marketing copy in `competitor_comparisons.py` already says this plainly. Keep it that way.

12. Security controls → implementation → enforcing test

#	CONTROL	IMPLEMENTED AT	TEST THAT ENFORCES IT
1	No inference routes on prod control plane	<code>src/trusted_router/main.py:880</code> , <code>:912</code>	<code>tests/test_stubs_and_security.py:1101</code> <code>test_production_control_plane_does_not_register_inference_routes</code> ; <code>tests/test_contract_regressions.py:195</code>
2	Prompt/output never in metadata store	<code>routes/internal/gateway.py</code> (metadata-only contract)	<code>tests/test_stubs_and_security.py:1134</code> <code>test_prompt_output_never_enter_metadata_store</code>
3	Content storage cannot be enabled	workspace PATCH handler, <code>types.py:118</code>	<code>tests/test_stubs_and_security.py:55</code> , <code>:69</code>
4	Sentry scrubbing (keys, fragments, prefixes, bodies)	<code>src/trusted_router/sentry_config.py:21,47,83,240-241,301-332,563</code>	<code>tests/test_security_contracts.py:255</code> , <code>:281</code> ; <code>tests/test_stubs_and_security.py:1150</code> ; <code>tests/test_scrub_totality_property.py</code>
5	No Sentry in enclave	policy (<code>docs/security.md</code>)	<code>tests/test_stubs_and_security.py:1181</code> (△ silently passes if sibling <code>quill-cloud-proxy</code> isn't checked out, likely a no-op in CI)
6	Prod config fails closed (internal token, Stripe secrets, Sentry DSN, storage)	<code>src/trusted_router/config.py:1443</code> , <code>:1950-1990</code>	<code>tests/test_stubs_and_security.py:1018</code> , <code>:1049</code> , <code>:1076</code> ; <code>tests/test_security_contracts.py:50</code> , <code>:238</code>

#	CONTROL	IMPLEMENTED AT	TEST THAT ENFORCES IT
7	Surface-scoped secret ownership	config.py:247-257 , :1981	config surface tests in tests/test_security_contracts.py / tests/test_internal_surface_deploy.py
8	Salted SHA-256 API keys, constant-time verify	src/trusted_router/security.py:16-38	tests/test_stubs_and_security.py:298 test_api_key_secrets_are_salted
9	Dev user header disabled in prod	src/trusted_router/auth.py:185	tests/test_security_contracts.py:142 ; tests/test_stubs_and_security.py:539
10	Strict bearer-prefix dispatch	auth.py:90-177	tests/test_auth_store_collapse.py (trsess-/sk-tr-paths)
11	Session cookie HttpOnly/Secure/Lax	auth.py:378-410	tests/test_session_cookie_attributes.py:112 , :177
12	Internal token constant-time, required in prod	routes/internal/_shared.py:53-67	tests/test_security_contracts.py:50 , :81 ; tests/test_contract_regressions.py:195
13	OAuth PKCE S256, exact redirect, single-use server-bound consent	routes/oauth.py , oauth_app_policy.py	tests/test_oauth_increment_d.py:48 , :122 ; tests/test_oauth_key_delegation.py:176
14	BYOK envelope: KMS-wrapped DEK, no plaintext	src/trusted_router/byok_crypto.py:52-97 , key_management.py:113,180	tests/test_byok_crypto.py:39 ; tests/test_storage_gcp.py:221 ; tests/test_oauth_and_console.py:1410 ; tests/test_billing.py:855
15	BYOK AAD v2 injective + namespace separation + enclave parity	byok_crypto.py:285-312	tests/test_byok_aad_namespace_property.py:198,219,271,289,348
16	No dev wrapping key outside local/test	key_management.py:193-199	tests/test_byok_crypto.py ; tests/test_byok_v1_removal_gate.py
17	Privacy floor fail-closed, order-independent	catalog_privacy.py:231 , routing.py:694-708,1108-1135	tests/test_routing_privacy_floor_property.py ; tests/test_auto_route_privacy_floor.py:191,207,222 ; tests/test_gateway_replay_privacy.py ; tests/test_catalog_privacy_coherence_property.py
18	Receipt-key log append-only, GCP chain verified	receipt_keys.py:99,437,497	tests/test_receipt_key_log.py:65,322 ; tests/test_receipt_key_versions_migration.py
19	Published measurement == running enclave	scripts/verify_trust_measurements.py , .github/workflows/trust-drift.yml	tests/test_trust_measurement_drift.py:345,382,395,407
20	Synthetic-probe attestation evidence checks (nonce binding, COSE/JWT, cert binding, PCR0 pin)	src/trusted_router/synthetic/probes.py (_attestation_evidence)	tests/test_attestation_evidence.py ; tests/test_attested_transport.py:84,126,135
21	Rate limiter ignores spoofable headers; local-only	src/trusted_router/request_limits.py:31,89	tests/test_stubs_and_security.py:876-1018 , :1193 , :1253
22	Request body limits / smuggling	src/trusted_router/request_body_limit.py:79	tests/test_api_edge_contracts.py , request-body tests
23	Security headers / CSP nonce	src/trusted_router/middleware.py:345	tests/test_security_response_headers.py
24	Stripe webhook signature required	Stripe webhook route	tests/test_security_contracts.py:171,215,238
25	Cloud Armor / WAF configuration	infra/cloud_armor.tf , scripts/deploy/_edge_security.sh , _aws_waf.sh	tests/test_ddos_edge_deploy.py ; tests/test_public_surface_edge.py ; tests/test_internal_surface_edge.py
26	Cloud baseline checkers read-only	scripts/deploy/*_security_baseline*.sh	tests/test_cloud_security_baseline.py:50,90,96

13. Threat model

#	THREAT / ACTOR	ASSET	MITIGATION	RESIDUAL RISK
T 1	TR operator or insider wants to read prompts	Prompt/output	Prompts only inside TEE. Control plane has no inference routes. Image change means a new digest, which drift check and customer verifiers detect. Secrets gated on image	Operator ships a malicious build <i>and</i> publishes its digest. Detection then relies on someone reading the source diff. Single-person org
T 2	Cloud host admin / hypervisor	Prompt in memory	Intel TDX memory encryption, <code>dbgstat disabled-since-boot</code> , <code>monitoring_enabled.memory=false</code>	Physical/side-channel attacks. Trust in Google, Intel and AMD roots
T 3	Network MITM / malicious relay	Prompt in transit	TLS terminates in TEE. Cert hash + <code>tls-exporter</code> + caller nonce bound in <code>eat_nonce</code> . Same-session rule	Clients that skip verification get only ordinary TLS
T 4	Replay of old attestation	Verifier trust	Caller nonce, short <code>exp</code> , no evidence caching	None significant if verifier checks nonce
T 5	Stale or drifted published measurement	Verifier trust	<code>trust-drift.yml</code> daily + post-deploy, <code>--strict</code> . Signed per-plane bundles	Up to ~24 h detection for enclave-side drift
T 6	Control plane compromise	Keys, BYOK, billing	Salted key digests. BYOK ciphertext only (KMS unwrap needs attested identity). Secrets split per surface	Attacker can mint keys, alter routing policy (e.g. privacy flags), or read metadata. Control plane is not attested
T 7	Leaked BYOK DB rows	User provider keys	AES-256-GCM + KMS-wrapped DEK + AAD bound to workspace/provider/namespace	KMS IAM misconfiguration. Single KMS key for all users
T 8	Secrets leaking into logs/Sentry	Keys, prompts	Scrubbers, <code>max_request_body_size=never</code> , property tests, no Sentry in enclave	New field names not in <code>SENSITIVE_KEYS</code> . Third-party library logging
T 9	Upstream provider retains or trains on data	Prompt	Privacy tiers. Default "assume stored". <code>zdr / confidential</code> floors fail closed	ZDR is contract/policy, unverifiable by TR. Catalog flags can be wrong
T 10	Spoofed client IP / auth headers	Rate limits, identity	Untrusted mode, ignored XFF/CF headers, dev header off in prod	Per-instance limits only. Fleet protection relies on Cloud Armor
T 11	DoS / volumetric	Availability	Cloud Armor throttles + L7 DDoS, body limits, Cloud Run max caps	Edge rollout incomplete on AWS/Azure (<code>edge-surfaces.json</code>)
T 12	OAuth app abuse / code interception	User credits	PKCE S256, exact redirect, server-bound consent, \$20 default budget, verified owner	Legacy <code>/v1/auth/keys</code> still allows plain PKCE
T 13	Gateway → control plane impersonation	Billing ledger	Internal token (constant-time), surface ownership, dedicated federation secrets	Shared bearer secret (not mTLS/attestation-bound)
T 14	Dependency / supply-chain compromise	Everything	OSV-Scanner, Dependabot, SHA-pinned actions, attested image	A malicious dependency inside the attested image <i>would</i> be attested. Attestation proves identity, not benign behavior
T 15	Batch API data at rest	Batch prompts/results	Per-artifact AES-GCM, KMS-wrapped, 30-day deletion	Weaker than zero retention. Depends on KMS/IAM policy
T 16	Azure MAA / non-GCP planes	Verifier trust	Pinned MAA issuer + <code>HOST_DATA</code> , receipts <code>verified=false</code>	Microsoft is a trusted third party

14. "Never do this" list for contributors

1. **Never register inference routes on the production control plane**, or add any non-attested fallback for `/chat/completions`, `/messages`, `/responses` or `/embeddings`. Don't touch `_control_plane_inference_enabled`.
2. **Never log, store, or send to Sentry/Axiom/analytics any prompt, message, completion, or output content**. That includes debug logs, exception messages that embed request bodies, and "temporary" diagnostics. When you add a new sensitive field name, add it to `SENSITIVE_KEYS` / `SENSITIVE_STRING_FRAGMENTS`, not to a second list.
3. **Never add Sentry (or any third-party telemetry SDK) to quill-cloud-proxy/enclave-go or any attested image**.
4. **Never add content fields to gateway↔control-plane contracts** (authorize, settle, refund, activity, generation).
5. **Never store a raw BYOK/provider key**, return it in an API response, or log it. Use `encrypt_byok_secret`. Never decrypt `provider`-namespace envelopes in the control plane.
6. **Never change** `_aad_v2` without changing `enclave-go/internal/byokcache/cache.go` in lockstep and updating both pinned vectors. A mismatch breaks every BYOK key silently.
7. **Never introduce a silent dev-key / memory-backend / missing-secret fallback** for deployed environments. Production must fail closed at startup.
8. **Never compare secrets with** `==`. Use `constant_time_equal` / `hmac.compare_digest`.
9. **Never trust** `X-Forwarded-For`, `CF-Connecting-IP`, `X-TrustedRouter-User`, **or Host** for identity or rate-limit keys. Never enable `edge_header` mode behind a front door that doesn't overwrite the header.
10. **Never compare privacy tiers numerically** for enforcement. Use `endpoint_meets_privacy_requirement`. Never flip a provider's ZDR/E2EE/confidential flag without a cited source. Unknown means "assume stored".
11. **Never put** `api.trustedrouter.com` **behind Cloudflare proxy** or any TLS-terminating front door.
12. **Never publish a measurement without the matching signed bundle**, and never weaken `trust-drift.yml --strict`, issuer queue limits or verifier checks to make a red run green.
13. **Never reuse one internal secret across surfaces** (gateway, observer, operator, federation, lightning), and never treat federation secrets as generic internal credentials.
14. **Never claim SOC 2, HIPAA or ISO certification**, "open source" (it's BUSL-1.1 source-available), or that attestation covers upstream providers.
15. **Never run** `scripts/smoke_byok_prod.py` **casually**. It is destructive and meant for security releases only. Never run unbounded "read-only" Spanner scans (see `AGENTS.md`).
16. **Never weaken a security test or conformance test** to make a change pass.

15. Open issues / follow-ups spotted during research

- `README.md:93-99` attestation snippet is inaccurate: raw JWT, not JSON; `image_digest` is nested; GCP has no `pcrs`.
- `tests/test_stubs_and_security.py:1181` (`test_no_sentry_in_enclave_code`) returns early when the sibling repo is absent, so it probably enforces nothing in CI.
- `docs/security.md` vs the `quill-cloud-proxy` README disagree on where `trust.trustedrouter.com` is hosted.
- CSP is still report-only.
- The edge-surface inventory is `live_state_verified: false`, with several PO items pending.
- The legacy `/v1/auth/keys` path still accepts `plain` PKCE.
- A single KMS key wraps all BYOK DEKs, and the internal gateway token is a shared bearer secret rather than attestation-bound. Both are reasonable today and worth revisiting at scale.

04

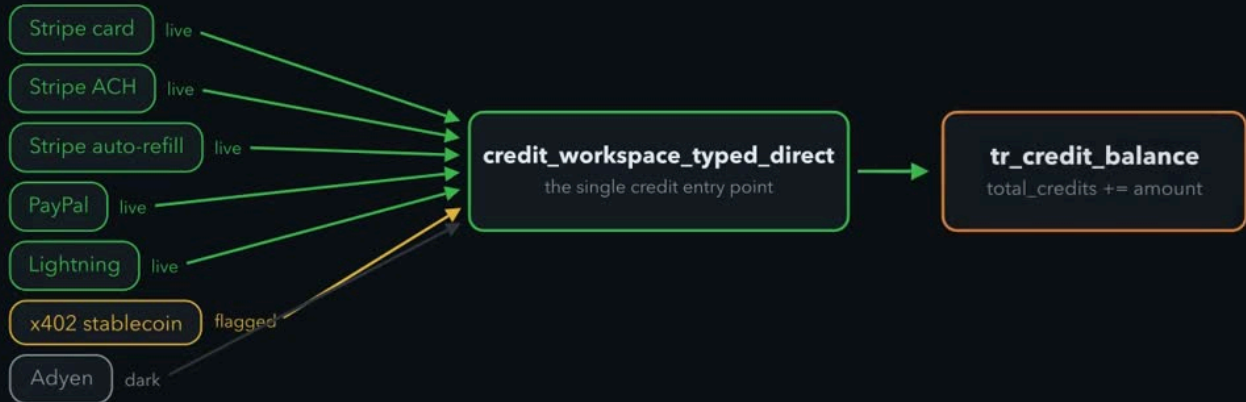
PART IV

Month 1

1. Money & billing
2. Data & storage
3. Infrastructure & deploys

09 Money & Billing

4. Payments in



Every money transaction locks the credit row before the key row, so there is no lock cycle.

PART IV • CHAPTER 08 • HERO STILL: VIDEO 09 AT 1:06

Money & billing

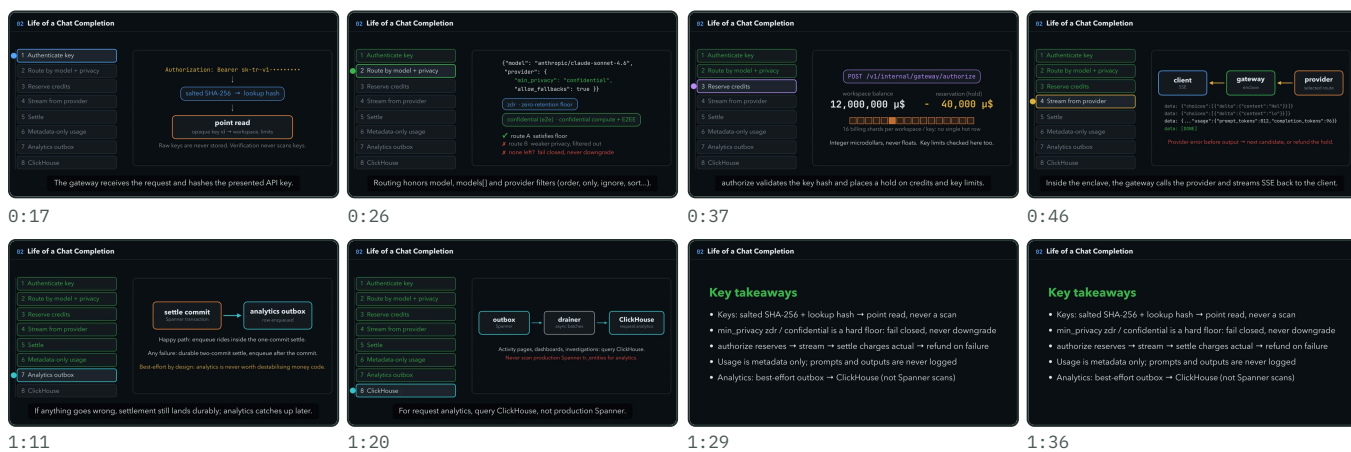
Microdollars, the typed ledger, reserve/settle/refund, async settlement and payment rails.

- 1. Overview (the 5-minute version)
- 2. Invariants (memorize these)
- 3. Reservation state machine
- 4. Settle sequence: one-commit happy path vs two-commit fallback
- 5. Payment ingestion flow (Stripe webhook → credits)
- 6. Key modules
- 7. What the async-settle vocabulary means
- 8. Payment rails
- 9. How to safely change money code (checklist)
- 10. Open risks and sharp edges

Storyboard · Video 09: Money & billing



Storyboard · Video 02: Life of a request



Research snapshot of `quill-router` at `main` @ `778120494` (2026-10-19). Read-only investigation; no production systems touched. Source paths are relative to the repo root; `src/trusted_router/` is abbreviated `tr/` in tables.

AGENTS.md: "reserve / settle / refund and anything touching credits are the highest-risk code in the repo. Changes there need a differential or conformance test, never review-by-eye."

1. Overview (the 5-minute version)

Unit of account. All money is integer **microdollars** (1 USD = 1,000,000 μ \$). See `tr/money.py`. Floats exist only for response compatibility (`microdollars_to_float`). Token cost is $(tokens * rate_micro_per_million + 500_000) // 1_000_000$ (round half-up), `tr/money.py:57`.

Customer price. Upstream provider cost $\times 1.055$ (`STANDARD_PRICE_MARKUP_BASIS_POINTS = 550`, `tr/pricing.py:126`), ceiling-rounded, with a floor of \$0.01 per million tokens. Prices come from a committed snapshot (`tr/data/openrouter_snapshot.json`) that an **hourly price-refresh bot** (`.github/workflows/refresh-prices.yml`, `scripts/pricing/refresh.py`) regenerates and auto-commits to `main`, which auto-deploys. Big price jumps are held unless they appear in `tr/pricing_approvals.py`.

The ledger ("typed book"). Since 2026-06-26 the only authoritative money book is three native Spanner tables. The JSON-blob book in `tr_entities` was retired (`docs/design/retire-json-credit-ledger.md` , `billing-dual-system-deletion-handoff.md`):

TABLE	KEY	MONEY COLUMNS
<code>tr_credit_balance</code>	<code>(workspace_id, shard)</code>	<code>total_credits</code> , <code>total_usage</code> , <code>reserved</code> , <code>in_debt</code> , <code>billing_pause_causes</code>
<code>tr_key_limit</code>	<code>(key_hash, shard)</code>	<code>limit_micro</code> (NULL = uncapped), <code>usage</code> , <code>byok_usage</code> , <code>reserved</code> , <code>include_byok</code>
<code>tr_reservation</code>	<code>reservation_id</code>	<code>credit_reserved_micro</code> , <code>key_reserved_micro</code> , <code>actual_micro</code> , <code>settled</code> , <code>expires_at</code> , <code>terminal_at</code> , <code>idempotency_scope</code>

Available balance on a shard = `total_credits - total_usage - reserved` . A hot workspace can be split across N shards. Each shard is an **independent sub-budget**, so the global cap holds by construction (`docs/design/credit-row-sharding-handoff.md`). The JSON `credit` entity is now **metadata only**: auto-refill config and Stripe customer/payment-method ids.

Per-request lifecycle. The attested enclave (quill-cloud-proxy) calls the control plane's internal routes: 1. `POST /internal/gateway/authorize` → `authorize_atomic` **reserves** an estimate with one conditional DML per row: `UPDATE ... SET reserved = reserved + @est WHERE headroom >= @est AND NOT in_debt` . Row-count 1 means accepted, 0 means 402. The reservation row is inserted with `expires_at` . 2. Optional `POST /internal/gateway/heartbeat` records `started_at` plus the usage delivered so far. The reaper can use this later. 3. `POST /internal/gateway/settle` (success) or `/refund` (failure) → `_settle_gateway_authorization` prices the actual usage, freezes the amount into a `SettleOutboxRow` , then finalizes: -**claim** the reservation (`UPDATE tr_reservation SET settled=true ... WHERE settled=false` ; first writer wins), -mark the authorization settled, write generation/activity, -**release** the exact holds and book `actual` (credit row first, then key row). 4. **Reaper** (`reap_expired_reservations_result`) reclaims holds that were never settled. It releases them free, or with Decision 70 snapshot booking it books the last heartbeat usage snapshot. It **never** frees a hold that a `pending / dead settle-outbox` row still intends to charge. 5. **Settle-outbox drain** (`/internal/gateway/settle-outbox/drain` , every 5 min) replays any intent whose inline finalize didn't land. It applies the **frozen** amount through a narrow primitive.

Why it's this complicated. Two failure directions: -**Double charge**. Prevented by the first-writer-wins reservation claim. N replays book at most once. -**Lost charge**. The request was served but the settle never landed, and the reaper released it free. Prevented by the **durable settle outbox plus the reaper guard**. Per the design, the reaper guard is the *sole* correctness mechanism for this direction. Drain timing only buys latency.

Async settlement (in flight, dormant). This is a multi-month program to move finalize off the response path. The enclave gets a 202 after one durable intent commit, and the drain books the charge. It is gated behind an enormous evidence apparatus: "shadow v2", "gates", "packs", "certificates", and a "frozen-main differential" `proof_oracle` . These prove that the change alters **no money behavior** before anyone flips it on. Production today: `TR_ASYNC_SETTLE_ENABLED=false` , `TR_ASYNC_SETTLE_PROTECTION=false` , `TR_ASYNC_SETTLE_SHADOW_POLICY=shadow-v2` for one dedicated probe workspace (`scripts/deploy/rollout.sh:659-664`).

Money in. Payment rails (Stripe card/ACH/auto-refill, x402 stablecoin via Stripe, PayPal, Adyen (dark), Lightning (separate LightningRouter workers)) all funnel into **one** function: `credit_workspace_typed_direct` (`tr/storage_gcp.py:3183`). It is idempotent on a `stripe_event` marker entity keyed by a rail-specific event id, and it writes a trust "payment fact" in the same transaction.

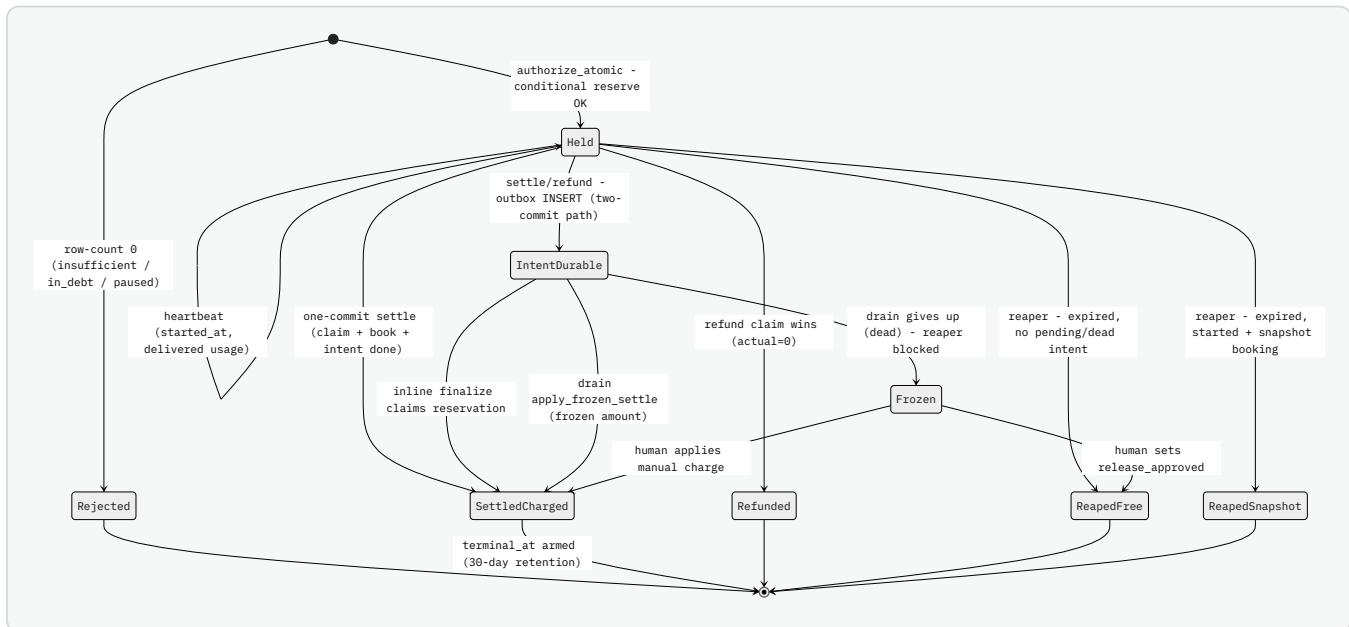
Money out. Creator earnings from app markup and custom-model markup go to `tr_earnings_balance` and are paid via **Routeable** ACH (dark, `TR_ROUTEABLE_ENABLED=false`).

2. Invariants (memorize these)

#	INVARIANT	WHERE ENFORCED
11	All money is <code>int</code> microdollars. No floats in ledger math. Rounding is half-up per component (token cost) or ceiling (markup).	<code>tr/money.py</code> , <code>tr/pricing.py:134-170</code> , <code>tr/billing_snapshot.py</code> (int64 overflow is an explicit error)
12	Per shard: <code>total_usage + reserved <= total_credits</code> at reserve time. The reserve is a single conditional <code>UPDATE</code> that takes the write lock directly, with no read→upgrade. This is the fix for the wound-wait deadlock.	<code>reserve_credit</code> <code>tr/storage_gcp_counter_dml.py:68</code> ; design docs/design/billing-typed-counters.md §2
13	Shards partition the workspace budget, so global overspend at admission is impossible. Sum over shards of credits/usage/reserved gives the workspace totals.	<code>docs/design/credit-row-sharding-handoff.md</code> ; <code>tr/storage_gcp_credit_shards.py</code>
14	First writer wins on <code>tr_reservation.settled false→true</code> . Only the claim winner touches counters. Replays are 0-row no-ops.	<code>claim_reservation</code> <code>tr/storage_gcp_counter_dml.py:781</code>
15	Release subtracts the exact recorded hold (<code>reserved >= @hold guard</code>). Row-count must be 1. A 0-row release must never be silently accepted, because it strands the hold and loses the charge.	<code>release_credit</code> <code>tr/storage_gcp_counter_dml.py:299</code>
16	Lock order: credit row before key row, everywhere (authorize, finalize, reaper, top-up). No cycle.	<code>settle_atomic</code> docstring <code>tr/storage_gcp_authorize.py:958</code> ; <code>tests/test_settle_lock_order.py</code>
17	A settle can exceed its hold (actual > estimate). The overrun is booked anyway (<code>test_settle_books_usage_always.py</code>). If a shard goes negative it is covered from other shards, or every shard is marked <code>in_debt</code> while the signed sum is negative. Marked rows refuse reservations. Inflows repay negative rows first, lowest shard first.	<code>tr/credit_debt.py</code> , <code>tr/storage_gcp_credit_debt.py:177</code> ; TLA+ proofs/CreditDebt.tla
18	Reaper guard. The reaper never free-releases a reservation whose authorization has a <code>tr_settle_outbox</code> row in <code>pending</code> or <code>dead</code> . This is re-checked inside the read-write claim transaction. A scan-time filter alone is insufficient (MF2). Only a human-set <code>release_approved</code> re-permits a free release.	<code>tr/storage_gcp_authorize.py:1397-1440</code> , <code>_CLAIM_RESERVATION_GUARDED_SQL</code> <code>tr/storage_gcp_counter_dml.py:41</code>
19	Frozen amount wins (MF5). The drain books <code>actual_cost_micro</code> exactly as stored. It never reprices, even across deploys, catalog changes, or endpoint retirement. It also routes on the stored <code>settle_origin</code> (MF4).	<code>apply_frozen_settle</code> <code>tr/services/settle_outbox_apply.py:136</code>
11 0	Outbox PK is (<code>authorization_id, intent_kind ∈ {settle, refund}</code>). Enqueue is INSERT-as-claim (<code>ALREADY_EXISTS</code>), not an upsert.	<code>SpannerSettleOutbox.enqueue</code> <code>tr/storage_gcp_settle_outbox.py:500</code>
11 1	<code>finalized=False</code> is ambiguous. <code>already_released_free</code> on a row that intended a charge is an invariant violation → dead + alert , never "recovered".	<code>ApplyOutcome</code> <code>tr/services/settle_outbox_apply.py:78</code> ; design §3
11 2	Analytics never destabilizes money. The benchmark analytics-outbox INSERT rides in the money commit only on the one-commit happy path. Every failure falls back to the two-commit settle, with the enqueue after commit and best-effort.	<code>tr/storage_gcp_analytics_outbox.py</code> docstring; AGENTS.md "Money code"
11 3	Payment credit is exactly-once per external payment. The <code>stripe_event</code> marker plus the trust payment fact (keyed by provider + <code>PaymentIntent/capture id</code>) are written in the same transaction as the <code>total_credits</code> bump.	<code>credit_workspace_typed_direct</code> <code>tr/storage_gcp.py:3183</code>
11 4	Refunds and chargebacks never silently debit customer credits. They become adverse trust events with recovery debt ("unrecovered principal"), absorbed from future inflows/releases, or an inbox entry for review.	<code>record_adverse_trust_event</code> <code>tr/storage_gcp.py:3324</code> ; <code>absorb_unrecovered_recovery_tx</code> <code>tr/storage_gcp_trust.py:282</code>
11 5	Flag-off byte identity. Every new money mechanism ships dormant. With the flag off, SQL, parameters, and RPC sequence must be byte-identical to before. This is proven by the frozen-main differential.	<code>claim_reservation_statement</code> docstring; <code>tests/test_async_settle_proof_oracle.py</code>

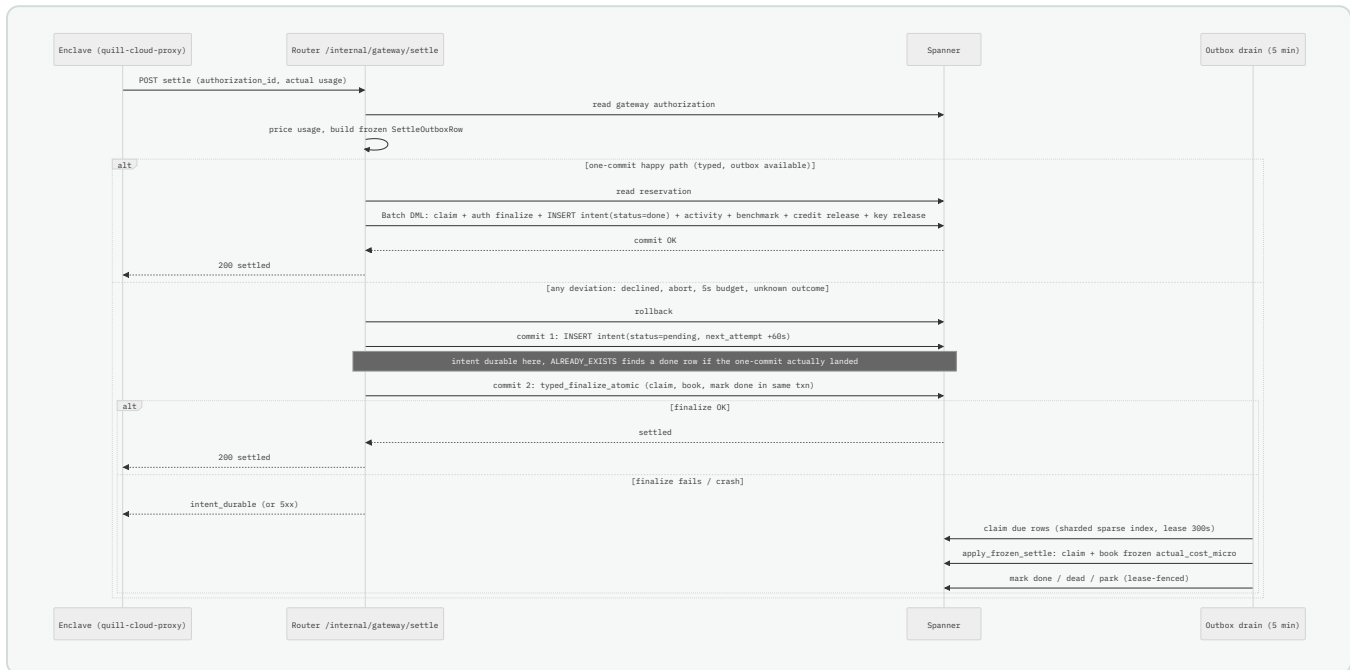
#	INVARIANT	WHERE ENFORCED
11 6	Async acceptance commits responsibility, never money . The async enqueue transaction must not claim, book, release, or touch credit/key rows. Its reservation predicate (<code>settled=false</code> , row-count exactly 1) prevents acknowledging an unbookable charge.	<code>tr/storage_gcp_async_settle.py:28</code> ; <code>docs/design/async-settle-outbox-v1.md §4</code>
11 7	Standing auditor tripwire: <code>typed.reserved == Σ open typed holds</code> (both directions, shard-aware). It runs daily in CI.	<code>audit_typed_invariants</code> <code>tr/storage_gcp_counter_reconcile.py:242</code> ; <code>.github/workflows/typed-audit.yml</code>

3. Reservation state machine



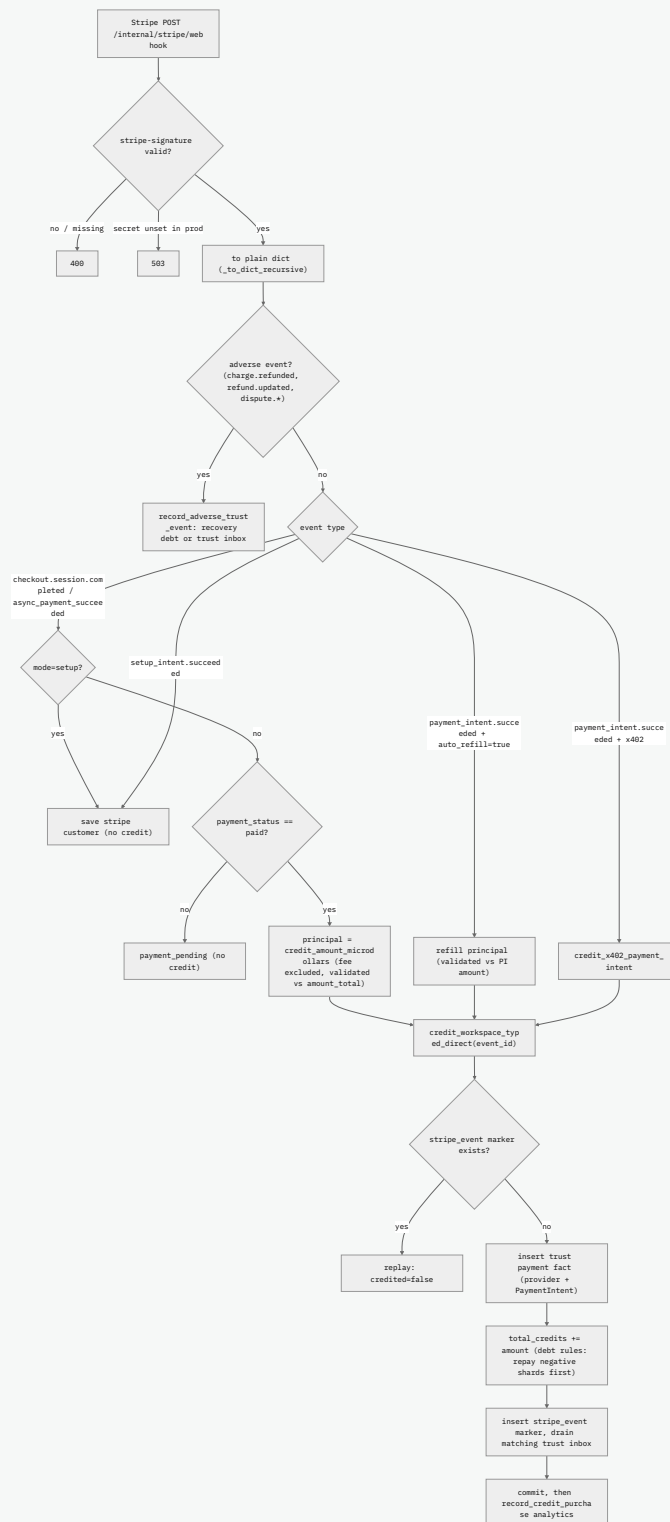
Notes: - "Claim" is the `settled false→true` transition. Every arrow into a terminal state goes through `claim_reservation` , so concurrent inline, drain, and reaper attempts resolve to exactly one winner. - A settle intent and a refund intent can coexist (different PKs). The first reservation claimant wins. The loser drains to `already_settled_with_charge` , plus a low-priority review flag if a refund lost to a charge. - Fast-admission pilot (`TR_FASTPATH_PILOT_ENABLED=true` for two workspaces): the hold lives in a **lease** on the Go "front door" service, not in `tr_reservation` . The reservation row "holds nothing". Settlement goes through `_settle_leased_authorization` / `tr/services/fastpath_settle.py` . TLA+ specs `Leaselifecycle` , `TerminalOrder` , `AuditorCommit` , `KeyCapFence` are **planned**. `CreditDebt` and `AutoRefillHandoff` are **implemented**.

4. Settle sequence: one-commit happy path vs two-commit fallback



Key facts (docs/design/durable-settle-outbox.md §7, tr/routes/internal/gateway.py:4413-4600):- Per authorize+settle round trip, one-commit takes read-write commits from **4** → **2**. - If the one-commit transaction actually landed but reported failure, the fallback is still safe. The enqueue hits `ALREADY_EXISTS` on a `done` row, finalize claims 0 rows, and nothing is charged twice. - With `settle_outbox_enabled=True` (production) and typed request records, a failed enqueue returns **503** ("Settlement durability is temporarily unavailable"). The router does not proceed without durability. - Honest scope: durability starts at the intent commit. A crash before that still depends on enclave redelivery, which is a bounded **in-memory** retry channel (1,024 jobs, 6 attempts) and does not survive enclave loss.

5. Payment ingestion flow (Stripe webhook → credits)



Card events key the idempotency marker on the Stripe **event id**. ACH keys it on `stripe_checkout:{payment_intent}` so that `completed` and `async_payment_succeeded` dedupe (`_checkout_credit_event_id`, `tr/routes/internal/webhook.py:545`). One 2026-09-04 fix matters here: if a backfill scan wrote the trust fact before

the webhook arrived, the webhook still credits exactly once. Before the fix it answered `credited:false`, Stripe stopped retrying, and the payment was zeroed (`tr/storage_gcp.py:3221-3250`).

6. Key modules

6.1 Ledger, reserve, settle, refund, reaper

MODULE	PATH:LINE	ROLE
Money primitives	<code>tr/money.py:57</code> (<code>token_cost_microdollars</code>), <code>:69</code> (<code>dollars_to_microdollars</code>)	Integer μ \$ math, display formatting, checkout limits (\$10k Stripe max, \$10 PayPal min, \$0.30 signup credit)
Custom pricing	<code>tr/pricing.py:126</code>	5.5% markup, floor, tier selection (<code>select_price_tier</code> :74, last tier is the fallback), cache pricing
Counter DML	<code>tr/storage_gcp_counter_dml.py:68</code> <code>reserve_credit</code> · <code>:299</code> <code>release_credit</code> · <code>:472</code> <code>reserve_key</code> · <code>:622</code> <code>release_key</code> · <code>:781</code> <code>claim_reservation</code> · <code>:810</code> <code>claim_reservation_statement</code>	The single-statement conditional DML primitives
Authorize	<code>tr/storage_gcp_authorize.py:468</code> <code>authorize_atomic</code>	One DML-only txn: credit hold (shard candidates), key hold, reservation + authorization insert, idempotency replay
Typed finalize	<code>tr/storage_gcp_authorize.py:1693</code> <code>typed_finalize_atomic</code>	Claim → auth finalize → generation/activity → credit release → key release. Optional one-commit <code>settle_outbox_intent</code> , payouts
Claim-gated settle	<code>tr/storage_gcp_authorize.py:958</code> <code>settle_atomic</code>	Narrow settle/refund used by the drain and reaper, with in-txn outbox guard
Reaper	<code>tr/storage_gcp_authorize.py:1224</code> <code>reap_expired_reservations_result</code> , <code>:1397</code> <code>_finalize_reaped_reservation_atomic</code>	Guarded reclaim. Decision 70 snapshot booking (<code>TR_REAP_SNAPSHOT_BOOKING_ENABLED=true</code> in prod)
Payout txs	<code>tr/storage_gcp_authorize.py:2162/2233/2290</code>	User-model, app-markup, and custom-model-markup payouts into <code>tr_credit_movement</code> + <code>tr_earnings_balance</code> inside the settle txn (<code>INSERT OR IGNORE</code> idempotent)
Debt rules	<code>tr/credit_debt.py:57-125</code> , <code>tr/storage_gcp_credit_debt.py:177</code>	Cover / mark / repay (pure math + Spanner application)
Shards	<code>tr/storage_gcp_credit_shards.py</code> , <code>tr/storage_gcp_credit_rebalance.py</code> , <code>scripts/expand_billing_shards.py</code> , <code>.github/workflows/reshard-billing-workspace.yml</code>	Shard choice, lazy rebalancing of fragmented sub-budgets
Live balance	<code>tr/typed_balance.py:28</code> <code>live_credit_summary</code>	Read path for display/API (typed snapshot)
Auditor / repair	<code>tr/storage_gcp_counter_reconcile.py:242</code> <code>audit_typed_invariants</code> , <code>:548</code> <code>repair_typed_reserved</code> ; <code>scripts/audit_typed_counters.py</code>	Standing money tripwire
Gateway routes	<code>tr/routes/internal/gateway.py:2492</code> <code>authorize</code> · <code>:2500</code> <code>heartbeat</code> · <code>:2552</code> <code>settle</code> · <code>:2561</code> <code>refund</code> · <code>:2584</code> <code>settle-outbox/drain</code> · <code>:2628</code> <code>home-settlement/drain</code> · <code>:2648</code> <code>deferred/reap</code>	Enclave-facing internal API
Settle handler	<code>tr/routes/internal/gateway.py:3776</code> <code>_settle_gateway_authorization</code> ; one-commit attempt <code>:4437</code> ; two-commit enqueue <code>:4484</code>	Pricing, frozen intent, commit-path choice
Config flags	<code>tr/config.py:1042</code> <code>reap snapshot</code> · <code>:1059-1066</code> <code>settle outbox</code> · <code>:967-979</code> <code>async settle</code> · <code>:1023</code> <code>fastpath</code>	Defaults off. Production values live in <code>scripts/deploy/rollout.sh</code> <code>ENV_VARS</code>

6.2 Durable settle outbox

MODULE	PATH:LINE	ROLE
Native outbox store	tr/storage_gcp_settle_outbox.py:491 SpannerSettleOutbox (enqueue :500, claim :627, mark :681, park :811, has_intent :1104, purge_done :1131)	INSERT-as-claim, lease-fenced state machine, 16 generated queue_shard s, sparse tr_settle_outbox_due_v2 index
Statement builders	tr/storage_gcp_settle_outbox.py:178 intent_insert_statements , :289 resolved_intent_statements , :362 mark_done_unleased_tx	Shared by one-commit, two-commit, and async enqueue
Frozen apply	tr/services/settle_outbox_apply.py:78 ApplyOutcome , :136 apply_frozen_settle	8+ outcome contract (settled_now, already_settled_with_charge, already_released_free, reservation_missing, park_typed_unavailable, invalid_row, reaped_snapshot, ...)
Drain	tr/services/settle_outbox_drain.py:72 drain_settle_outbox , :195 _resolve_row	Budget 240 s, lease 300 s, limit ≤500
Fast worker (dormant)	tr/services/settle_outbox_worker.py	Warm worker for async; settle_outbox_fast_drain_enabled=false
Auto-refill handoff	tr/services/auto_refill.py:58 , tr/services/auto_refill_outbox_drain.py	Refill attached to the settle-outbox row; TLA+ AutoRefillHandoff
Analytics outbox (best effort)	tr/storage_gcp_analytics_outbox.py , tr/storage_gcp_operational_analytics_outbox.py	Benchmark/activity delivery toward ClickHouse; 16 hash shards to avoid a monotonic-key hotspot

6.3 Async settlement and shadow (dormant)

MODULE	ROLE
tr/billing_snapshot.py:436 build_snapshot , :477 evaluate , :520 validate_envelope , :369 canonical_hash	Frozen billing v1 contract. Pure, strict-JSON, canonical-hash pricing evaluator. A Go twin lives in quill-cloud-proxy. Shared 759-case fixture tests/fixtures/async_settlement/billing_v1.json . Doc: docs/async-settlement-billing-v1.md
tr/async_settle_ticket.py	Detached signed settlement tickets (separate-purpose JWS)
tr/services/async_settle.py	Authorize-time projection + fail-closed bounded admission cache
tr/services/async_settle_handler.py	X-TR-Settlement-Mode: async-v1 boundary: parse → verify ticket → price → intent → 202, or sync_required
tr/storage_gcp_async_settle.py:28,37	Async enqueue: intent INSERT + retention clears + async_reservation_admission_statement (reservation must be open; row-count exactly 1)
tr/async_settle_fence.py	Capability so only the frozen apply primitive can book an async-tagged intent. Legacy claims get a NOT EXISTS fence (claim_reservation_statement)
tr/async_settle_shadow_*.py (v1, ~12 modules)	Shadow observation lifecycle: cohort, compare, evidence, journal, health, projection, wire
tr/async_settle_shadow_v2_*.py (~95 modules: gate_*, pack_*, collect_*, a1_*, a2_*, certificate, export*, native*, activation_*)	Shadow v2 offline evidence/gate machinery (see §7)
tr/storage_gcp_async_settle_shadow.py	Shadow-only complete-key transactions; no money writes

6.4 Payments and payouts

MODULE	PATH:LINE
Stripe webhook	tr/routes/internal/webhook.py:44

MODULE	PATH:LINE
Single credit entry point	<code>tr/storage_gcp.py:3183</code> <code>credit_workspace_typed_direct</code> ; <code>memory twin tr/storage.py:2129</code> ; <code>protocol tr/store_protocol.py:762</code>
Adverse events (refund/dispute)	<code>tr/storage_gcp.py:3324</code> ; <code>tr/storage_gcp_trust.py</code>
Stripe checkout / fees / Radar	<code>tr/services/stripe_billing.py</code> , <code>tr/services/stripe_fees.py</code> (card 2.9%+30¢, ACH 0.8% capped at \$5, stablecoin 1.5%: <code>tr/config.py:722-731</code>), <code>tr/services/stripe_radar.py</code>
Billing routes	<code>tr/routes/billing.py:80</code> <code>checkout</code> · <code>:143</code> <code>PayPal capture</code> · <code>:168/211</code> <code>x402 fund/settle</code> · <code>:248</code> <code>portal</code> · <code>:289</code> <code>coinbase</code> (deprecated)
Operator grants	<code>scripts/grant_credit.py</code> ; <code>payment owner lookup scripts/inspect_payment_owner.py</code> (the only sanctioned way to attribute a Stripe id; see AGENTS.md)

7. What the async-settle vocabulary means

TERM	MEANING	WHY IT EXISTS
Async settle (v1)	The enclave opts in (<code>X-TR-Settlement-Mode: async-v1</code>). The router verifies a signed ticket, prices with frozen billing v1, commits one transaction containing the pending outbox intent, and returns 202 with <code>settlement_status=pending</code> . The existing drain books it later.	Settle p50 was 103–703 ms depending on region (europe-west4 is worst). Async takes waiting for finalize off the response path. It reuses the outbox, frozen amount, claim gate, lease fence, and reaper guard: "no second settlement queue".
Admission	Whether a request may use async: eligible cohort (typed, local, ordinary Credits; OpenAI/Anthropic; chat/responses; exact usage), paid tier 2/3, workspace $\text{pending-sum} \leq \text{cap}$ (\$5 pilot, \$25/\$100), fleet drain $\text{p95} \leq 5$ s.	Exposure control. The doc states plainly that this is a lag guard, not a linearizable cap .
Protection	<code>async_settle_protection</code> . Fences legacy and sync writers so they cannot rewrite or race an async-tagged intent. Must be on before admission and turned off last.	Rolling-deploy safety (<code>tr/config.py</code> Settings docstring gives the disable order).
Shadow	Run the async pricing/normalization path (Python + Go) on real opted-in traffic without it booking anything. Sync books; shadow compares.	A seven-day clean window of zero unexplained disagreements is an exit criterion before admission.
Shadow v2	Amended evidence policy (2026-10-11, A1/A2/A3 amendments). It requires a closed partition $T = A + B + C$ per terminal attempt bucket, where A = compared eligible, B = authenticated exclusion, C = unresolved/lost, and C must be 0 .	v1's gate could not certify a real fleet: missing health publisher, idle boots, recovered checkpoints counted as loss, historical boots summed as if concurrent.
Gates (G1–G11)	Executable PASS predicates, each with named gap literals (58 in total): policy/schema, deployment roster, authorize obligations, terminal partition, eligible comparisons, timer/health, fleet load (CPU < 45%), durability, resets, a 604,800 s (7-day) clock, external proofs. Implemented in <code>async_settle_shadow_v2_gate_*.py</code> / <code>gates.py</code> . Pilot tools: <code>fleet-budgets-pre</code> → <code>fleet-budgets</code> → <code>shadow-serving</code> → <code>pre-flip</code> → <code>post-flip</code> .	One shared predicate implementation, so the report and the pilot can't disagree. Unevaluable means failure, never a skip.
Packs	<code>async_settle_shadow_v2_pack_*</code> : offline packaging of each day's original evidence (Monitoring pages, CI archives, trial records, budgets) into a bounded, hash-chained "daypack" with predecessors, so a gate can be evaluated over a reviewed, immutable input graph.	Evidence must be the original bytes, finite and reviewable, not a re-derived summary.
Collect / export / native	Dormant read-only collectors (<code>collect_*</code>), offline original-byte export (<code>export*</code>), and native readers that do only finite PK-range reads with no retries.	Gathering evidence must not create production load (cf. the September 6 CPU incident).
Certificate	<code>async_settle_shadow_v2_certificate.py</code> : a separate-purpose exclusion signature . It proves that a terminal attempt was authentically out of cohort (bucket B) without a price comparison.	Lets the partition close without counting excluded traffic as loss, and without an unsigned "trust me" flag.
A1 / A2 / A3	Design amendment rounds of shadow v2 (<code>a1_*</code> , <code>a2_*</code> modules). A1: health/obligations/close-result. A2: report/failure contract and digests. A3: live export and activation inputs.	Each round answered adversarial review findings.

TERM	MEANING	WHY IT EXISTS
Frozen-main differential	tests/fakes/frozen_main/package.tar.gz is a byte-pinned tarball of the router source at a BASE commit, loaded under a separate import alias (frozen_main). Tests run the same authorize→settle→drain scenario through frozen code and live code against the fake Spanner, then assert identical responses, DB state, drain outcomes, SQL trace , and RPC counts. An execution guard rejects any accidental call into live code.	Mechanically proves "flag-off = byte-identical money behavior" instead of relying on review. Inventory: 330 modules / 1,665 execution entries (docs/async-settle-frozen-main-inventory.md). Re-pin with scripts/async_settle/freeze_reference.py .
proof_oracle	pytest marker for those differentials (over an hour; sharded in CI). Skipped by default (pyproject.toml:77).	Run explicitly when touching billing: uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py
Mutations (tests/*_mutations.py)	Hand-written mutants of the money code that the test suite must kill.	Shows a test actually guards the property, not just that it passes.
proofs/	TLA+ specs model-checked by TLC in CI (./proofs/check.sh , proofs-guards.yml). Each claim must be broken by a listed mutant. Each guard has a swept table.	Covers interleaving bugs that tests can't schedule (debt across shards, auto-refill handoff, lease lifecycle).

8. Payment rails

RAIL	DIRECTION	STATUS	ENTRY POINT	IDEMPOTENCY KEY	NOTES
Stripe Checkout (card)	In	Live	POST /billing/checkout → webhook checkout.session.completed	Stripe event id	Fee passed through separately; principal is validated against amount_total
Stripe ACH	In	Live	Same, payment_method=ach	stripe_checkout: {payment_intent}	Async success/failure events; no auto-refill PM capture
Stripe auto-refill	In	Live	maybe_charge_after_settle off-session PI → payment_intent.succeeded (auto_refill=true)	event id; refill attached to outbox row	Triggered after settle when balance < threshold
x402 stablecoin (USDC via Stripe)	In	Flagged (x402_enabled)	/billing/x402/fund , /settle; webhook PI	PaymentIntent	Refunds go to manual review. Design: docs/design/x402-stablecoin-funding.md
PayPal	In	Live	/billing/paypal/orders/{id}/capture , /internal/paypal/webhook	paypal_capture: {capture_id}	PENDING captures are never credited. Refunded-before-credit reconciliation: docs/paypal-inbox-reconciliation.md
Adyen	In	Dark (TR_ADYEN_ENABLED=false)	/internal/adyen/webhook (HMAC) → credit_adyen_notification	checkout reference (HMAC-bound)	Webhook verification is independent of the enable flag, so late payments still credit. Refunds/chargebacks go to manual review. docs/adyen-rollout.md
Lightning (LightningRouter)	In	Live, separate workers	/internal/lightning/credit → LightningCredits.credit	lightning: {payment_hash} + immutable binding	No BTC balance stored. LND / Lexe (SGX) backends. BTCPay is a dashboard only. docs/lightning-*.md
Coinbase	In	Deprecated	/credits/coinbase → deprecated()	n/a	

RAIL	DIR ECT ION	STATUS	ENTRY POINT	IDEMPOTENCY KEY	NOTES
Signup / operator grants	In	Live	<code>credit_workspace_once , scripts/grant_credit.py</code>	grant event id	\$0.30 starter credit, granted atomically at account creation
Cross-plane credit transfer	Mov e	Live	<code>tr/credit_transfer.py</code> state machine + <code>storage_gcp_credit_transfer.py</code>	transfer id	Multi-cloud: "identity federates; credits do not". Transfers are explicit and conserve value
Federated deferred settlement	Mov e	Flagged	<code>/internal/gateway/home-settlement/drain , deferred/reap</code>	outbox row	A peer plane records spend while home is unreachable and forwards it later
App / custom-model markup	Inter nal	Live	payout txs in <code>typed_finalize_atomic</code>	<code>app_markup_pay out: {authorization_ id}</code>	Owner gets 70% of markup (<code>CUSTOM_MODEL_OWNER_SHARE_BASIS_POINTS=7000</code>). Reconcile with <code>scripts/reconcile_app_markup_payouts.py</code>
Routable	Out	Dark (<code>TR_ROUTABLE_ENABLED=f alse</code>)	<code>tr/routes/payouts.py , /internal/routable/webhook</code>	payable external id	US ACH to identity-verified creators, debiting <code>tr_earnings_balance . docs/routable-payouts.md</code>

9. How to safely change money code (checklist)

- Read first:** `docs/design/durable-settle-outbox.md` §3-§7, `docs/design/billing-typed-counters.md`, `AGENTS.md` "Money code", and the `async-settle` design docs if you touch `settle` or `outbox`.
- Ship dormant.** Put new behavior behind a default-off flag. Flag-off SQL, params, and RPC sequence must stay byte-identical. Production flags live only in `scripts/deploy/rollout.sh` `ENV_VARS`. Hand-set Cloud Run env vars are overwritten.
- Prove it, don't eyeball it.** Add a differential, conformance, or mutation test: - `tests/conformance/` for Store-protocol behavior (memory, postgres, spanner-pg). Never weaken one. - Frozen-main oracle when touching `settle`, `outbox`, `authorize`, or `billing`: `uv run pytest -q -n 4 -m proof_oracle tests/test_async_settle_proof_oracle.py`. If your change is intended, re-pin via `scripts/async_settle/freeze_reference.py` and regenerate the inventory. - Add a mutant (`tests/*_mutations.py`) for any new guard. Confirm that removing the guard fails a test.
- Fake fidelity (MF6).** `tests/fakes/spanner.py` is a substring SQL interpreter. Any predicate you add to an already-matched SQL string needs a matching fake change plus `_require_pred`, or the fake silently ignores it and your guard tests are theater.
- Patch the class, not STORE.** `monkeypatch.setattr(STORE, ...)` poisons later tests (issue #333). Run the **full** suite: `uv run ruff check . && uv run mypy && uv run pytest -q`. Never run `ruff format .`.
- Preserve the spine.** Claim-first-writer-wins, exact-hold release with row-count == 1, credit-then-key lock order, the in-txn reaper guard, frozen amount and origin, INSERT-as-claim, and no new read→write upgrade on hot counter rows.
- Never touch analytics inside the money txn** except the documented one-commit ride-along with full fallback.
- Interleavings?** If the change is a protocol (leases, handoffs, multi-shard), extend or add a TLA+ spec in `proofs/` with mutants and a guard table. CI runs `./proofs/check.sh`.
- Rebase before trusting a failure.** The hourly price bot moves `main`.
- Production safety.** Never run raw or ad-hoc DML on billing tables. Use reviewed CLIs (`scripts/grant_credit.py`, `audit_typed_counters.py`, `ramp_typed_billing.py`). Never full-scan `tr_entities.body`. Use `scripts/inspect_payment_owner.py` for Stripe attribution. Verify with the typed audit after any mutation.

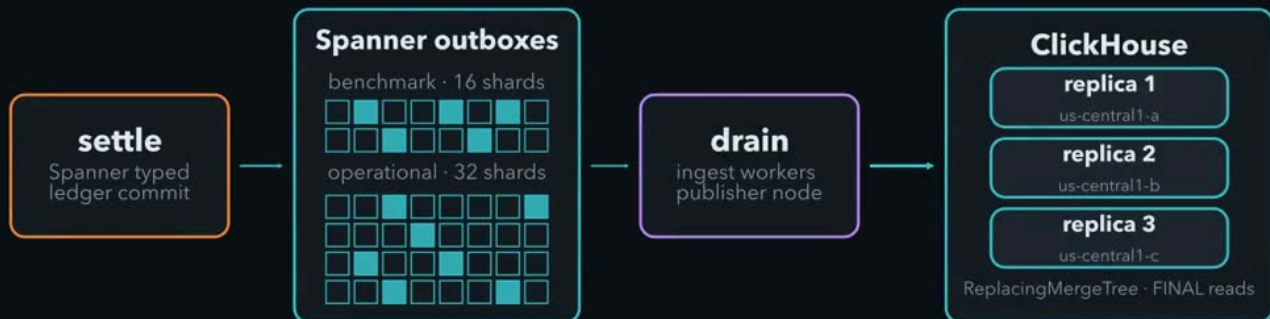
11. **Review gate.** Historical practice for billing PRs: an adversarial review (codex) **before** merge, no auto-merge armed in parallel, and owner (Joseph) sign-off for any flag flip that changes live enforcement.
12. **Hotspots.** Any new Spanner index or key on a monotonic timestamp must lead with a hash shard (cf. the `tr_settle_outbox_due_v2` incident, `durable-settle-outbox.md` §10.1). Don't "fix" latency alerts by loosening thresholds.

10. Open risks and sharp edges

1. **Pre-intent loss window.** A crash between receiving settle and the intent commit relies on enclave redelivery. That is an in-memory channel and does not survive enclave loss. Enclave-side durability is a cross-repo change with no owner go yet (`durable-settle-outbox.md` §9).
2. **Overrun exposure.** A settle above its hold books anyway and can push shards into debt (`in_debt` marks). Async adds pending exposure that the design admits has **no C-only numerical worst-case bound**. The admission predicate is a cached lag guard, not a hard cap (`async-settle-outbox-v1.md` §6). This is unresolved policy before any pilot.
3. **Complexity of the shadow apparatus.** About 95 `shadow_v2` modules plus about 130 `async-settle` test files, for a feature that is still off. It is a large maintenance surface, and very few people can reason about it. Its own docs call the production observations "gatekeeper-reported". Risk: evidence tooling drifts from the live code, or blocks unrelated changes through frozen pins and inventories.
4. **Frozen-main re-pin churn.** Any intended change to pinned modules (gateway, authorize, settle outbox, config, partner_billing, ...) needs a re-freeze plus inventory regeneration. It is easy to forget, and the oracle is skipped by default locally.
5. **Dead outbox rows freeze holds indefinitely.** They need a human runbook (manual charge, then `release_approved`). Confirm alerting on `dead` count and age exists and is watched.
6. **Snapshot booking is live** (`TR_REAP_SNAPSHOT_BOOKING_ENABLED=true`). The reaper now books money based on heartbeat-delivered usage, which is a newer billing behavior (Decision 70). Interactions with outbox intents and partial streams deserve scrutiny.
7. **Fast-admission pilot** (Go front door, leases) runs for real workspaces, yet 4 of its 6 TLA+ specs are still "planned". The hold lives outside Spanner on that path.
8. **Webhook handler shape.** `tr/routes/internal/webhook.py` is a long if-chain over dicts. The 2026-05-23 `StripeObject.get` regression shows SDK upgrades can silently break crediting. Synthetic chain tests are the safety net.
9. **Refund/dispute handling is mostly manual** for Adyen, x402, and PayPal edge cases. Recovery debt can stay unrecovered (alerts via `alert_unrecovered_principal`).
10. **The price bot auto-commits to main hourly and auto-deploys.** A parser self-heal driven by an LLM is sandboxed and spike-gated, but a bad price ships straight into the frozen per-request cost. The settle outbox will faithfully book it (MF5).
11. **Multiple fake backends** (memory `storage.py`, postgres, spanner-pg, fake Spanner). Behavior parity relies on the conformance suite plus `_require_pred` discipline.

10 Data & Storage

4 Analytics pipeline



A drain daemon on the ClickHouse publisher node reads each shard and inserts.

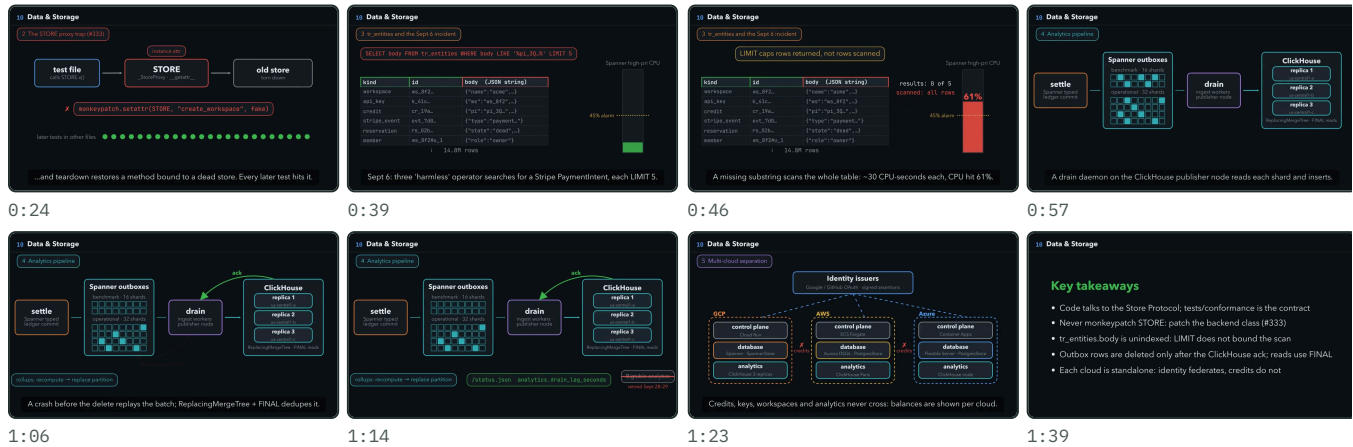
PART IV · CHAPTER 09 · HERO STILL: VIDEO 10 AT 0:57

Data & storage

The Store protocol, Spanner, Postgres, ClickHouse and the multi-cloud separation.

- 1. Overview: the data layer in brief
- 2. Data stores and what lives where
- 3. tr_entities and why JSON-body scans are dangerous
- 4. Store backends
- 5. The STORE proxy trap (issue #333)
- 6. Running conformance locally
- 7. Migrations and schema
- 8. Analytics pipeline: request → outbox → ClickHouse
- 9. Multi-cloud separation
- 10. Query-safety rules for operators
- 11. HANDOFF status summary (docs/storage-portability/HANDOFF.md)

Storyboard · Video 10: Data & storage

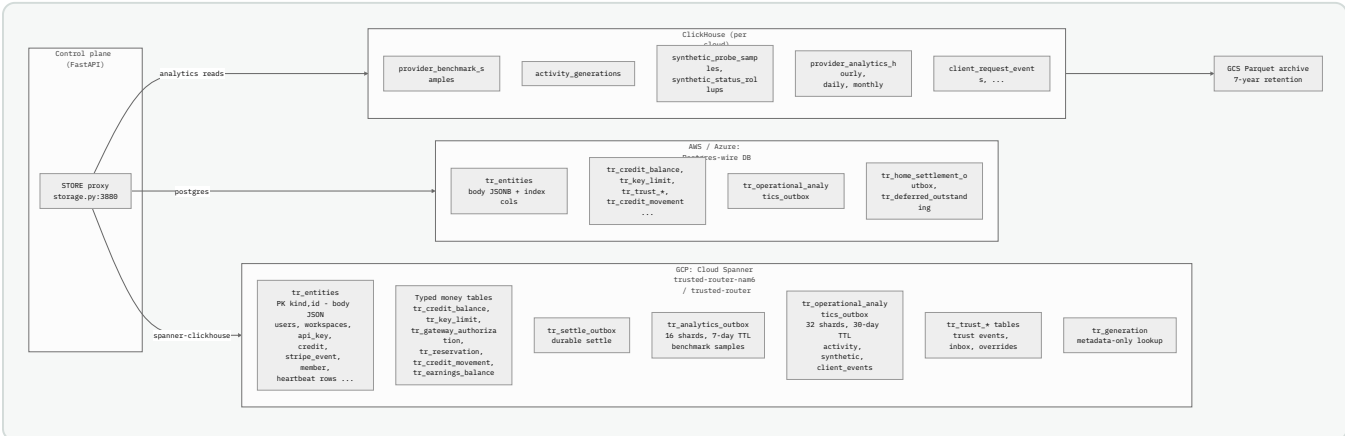


Research note for the 0-to-CTO onboarding pack. Paths are relative to `quill-router/` and cited as `path:line`.
Snapshot taken 2026-10-19 at `main` 778120494. Several source docs are older than the code, and the places where they disagree are flagged [STALE].

1. Overview: the data layer in brief

- **One abstraction, Store.** `Store` is a `runtime_checkable Protocol` in `src/trusted_router/store_protocol.py:76`. `AGENTS.md` says “~97 methods” (`AGENTS.md:91`). [STALE] The Protocol body (lines 76-1146) now declares **210 methods**. There are also four narrower capability Protocols: `TypedBillingStore` (`store_protocol.py:1147`), `SnapshotReaperStore` (`:1241`), `ShadowTransaction` (`:1254`) and `ShadowStore` (`:1259`).
- **Three production backend values** are accepted by `create_store` (`src/trusted_router/storage.py:3974-4085`) and by config validation (`src/trusted_router/config.py:1576`):
 - `memory`: `InMemoryStore` (`storage.py:150`), used by tests and local dev.
 - `postgres`: `PostgresStore` (`src/trusted_router/storage_postgres.py:542`), used by AWS (Aurora DSQL) and Azure (Flexible Server).
 - `spanner-clickhouse`: `SpannerStore` (`src/trusted_router/storage_gcp.py:389`). This is GCP production. Spanner is the system of record and ClickHouse serves analytics reads.
- **A module-global proxy.** `STORE` is a `_StoreProxy` (`storage.py:3880`) that forwards calls through `__getattr__` (`storage.py:3936`). The app calls `configure_store()` to swap its target (`storage.py:3950`). It is convenient and it is a trap for tests (see §5).
- **The executable contract is** `tests/conformance/`. Every test in it talks only to the `Store` Protocol. A backend is conformant when it passes the suite unchanged (`tests/conformance/conftest.py:1-17`).
- **Spanner holds a generic table and typed tables.** The legacy generic table is `tr_entities(kind, id, body JSON-string, updated_at)` with PK (`kind, id`) (`scripts/deploy/infra.sh:33`). Money and request state has moved into typed tables such as `tr_credit_balance`, `tr_gateway_authorization`, `tr_reservation` and `tr_settle_outbox`. Their full DDL is generated in `tests/conformance/spanner_ddl.py`.
- **Analytics is ClickHouse only.** Bigtable analytics were retired on 2026-09-28 and the code was deleted on 2026-09-29 (`docs/clickhouse-reliability.md:133-142`; `docs/storage-portability/README.md:6-11`). Rows reach ClickHouse through durable outboxes in the cloud's OLTP database. Each outbox has a drain daemon on the ClickHouse publisher node.
- **Each cloud is a separate TrustedRouter.** Identity federates across clouds. Credits, keys, workspaces and analytics stay in their own cloud (`docs/storage-portability/multi-cloud-separation.md:9-28`).

2. Data stores and what lives where



STORE	HOLDS	AUTHORITY	RETENTION
Spanner tr_entities	Users, workspaces, members, API keys and lookups, legacy credit entities, stripe_event / webhook_event dedup, consent requests, wallet users, the operational_outbox_heartbeat row, and more. Kinds in use: storage_gcp*.py _read/_write_entity calls.	System of record	No general TTL. rate_limit rows only get a generated-column TTL (scripts/deploy/migrate_entity_ttl.sh :2-7 ; tests/conformance/spanner_ddl.py:316)
Spanner typed tables	Billing shards, reservations, authorizations, credit movements, trust ledger	System of record, and the money path	terminal_at plus 30-day ROW DELETION POLICY on authorization, reservation and settle outbox (docs/spanner-reliability.md:318-333)
Spanner outboxes	tr_analytics_outbox (benchmark), tr_operational_analytics_outbox (activity/synthetic/client)	Durable queue. Rows are deleted only after ClickHouse acks them	7 d and 30 d TTL (spanner_ddl.py:30 , :119)
Postgres (DSQL / Flexible Server)	The same logical model. tr_entities.body is JSONB with projected index columns (src/trusted_router/storage_postgres_schema.sql:1-65)	System of record for that cloud	Per-cloud
ClickHouse	Content-free request metadata, provider benchmarks, synthetic probes, rollups. No prompts, outputs, keys or raw workspace IDs (docs/clickhouse-reliability.md:9-13)	Derived. Never on the inference or settle path	Raw 400 d, synthetic raw 14 d, hourly 3 y, daily/monthly no TTL (docs/clickhouse-reliability.md:15-25)
GCS Parquet	Daily immutable, fingerprint-verified exports	Cross-cloud and cross-version recovery source	7 y (docs/clickhouse-reliability.md:68-92)

Spanner instance facts:

- Config nam6 , Enterprise Plus, managed autoscaler from 1,000 to 10,000 PU with a 45% high-priority CPU target (infra/spanner_trusted_router.tf:10-39).
- prevent_destroy is set (infra/spanner_trusted_router.tf:36-38).
- PITR is 7 days, with daily full backups plus incrementals every 4 hours, and a 30-day copy in the trustedrouter-dr project (docs/spanner-reliability.md:10-23).
- **[STALE]** That runbook line still says "400 until 2026-09-25 and 600 after". The Terraform has since moved to the autoscaler.

- The Terraform deliberately leaves databases, schema and backups to the deploy scripts (`infra/spanner_trusted_router.tf:1-4`).

3. `tr_entities` and why JSON-body scans are dangerous

Model. Each row is `kind STRING(64)`, `id STRING(512)`, `body STRING(MAX)` (JSON) and `updated_at` (commit timestamp), with `PRIMARY KEY (kind, id)` (`scripts/deploy/infra.sh:33`). Reads that touch the generic table:

- **Point reads:** `SELECT body FROM tr_entities WHERE kind=@kind AND id=@id` (`src/trusted_router/storage_gcp.py:7316-7342`).
- **Range reads:** a list uses `kind=@kind AND STARTS_WITH(id, @prefix)` (`storage_gcp.py:7344-7375`). This bounds the scan to one kind's key range.
- **Secondary indexes** are emulated as extra rows whose id is `<owner>#<rest>`.
- On Postgres, `_` in owner ids is a LIKE wildcard. A prefix must therefore be escaped with `_like_prefix` (`storage_postgres.py:1107`), or one tenant's scan can return another tenant's rows (`docs/storage-portability/HANDOFF.md:626-628`).
- **Body:** nothing in `body` is indexed on Spanner.

Bloat. In July, issue #334 counted these numbers (`HANDOFF.md:232-246`):

- 14.8M rows backing 185 workspaces and 251 keys.
- About 5.45M dead `reservation`, `reservation_idemp` and `gateway_authorization_idempotency` rows.
- About 9.4M rows still growing with no TTL.

Status of the remediation:

- New terminal generations are no longer written to `tr_entities` (`docs/spanner-reliability.md:331-333`).
- `rate_limit` rows now have a TTL.
- The historic rows remain until a separate, reviewed cleanup.

The September 6 CPU incident (`docs/incidents/2026-09-06-spanner-operator-search-cpu.md`):

- Three "harmless" operator searches for a Stripe PaymentIntent ran against `tr_entities`. Each used a leading-wildcard `body LIKE` or `id LIKE` with no PK predicate, and each had `LIMIT 5`.
- Each query consumed about 30 CPU-seconds and ran for about 35 s (lines 7-12). That pushed high-priority CPU to 61% against the 45% alarm (line 16).
- **Root cause:** `LIMIT` caps the rows returned, not the rows scanned. Searching for an absent substring scans the whole table (lines 29-32).
- **Remediation:** `scripts/inspect_payment_owner.py` does one exact Stripe GET, then at most two complete-PK Spanner reads. They run at LOW priority with the `tr_ops_payment_owner` tag, 5 s deadlines and no retries (lines 45-56).

A related incident on 2026-09-24 had the same lesson: a periodic job read `tr_reservation` without a covering index (`docs/incidents/2026-09-24-spanner-overrun-rollup-scan.md`; `docs/spanner-reliability.md:177-181`).

4. Store backends

BACK END ID	CLASS	WHERE IT RUNS	MONEY PATH	CONFORMANCE
memory	InMemoryStore storage.py:150	Tests, local dev; default storage_backend (config.py:488)	Implements the JSON reserve/settle path	Always runs; cannot be skipped
postgres	PostgresStore storage_postgres.py:542 (7.9k lines). Schema is in storage_postgres_schema.sql and applied by apply_schema()	AWS Aurora DSQL (scripts/deploy/aws_eu_control_plane.sh:2), Azure Flexible Server (Citrus was rejected for now: scripts/deploy/azure_control_plane.sh:33-39)	One SQL implementation; exactly-once credit relies on INSERT ... ON CONFLICT DO NOTHING rowcount (HANDOFF.md:48-62)	postgres backend; skips unless TR_CONFORMANCE_POSTGRES_DSN is set
spanner-pg (test only)	PostgresStore over PGAdapter on the Spanner PG dialect	CI only. Proves Spanner's PG subset can run the same code (tests/conformance/conftest.py:18 2-213)	Same as postgres	Skips unless TR_CONFORMANCE_SPANNER_PG_DSN is set
spanner-clickhouse	SpannerStore storage_gcp.py:389 (7.4k lines + ~60 storage_gcp_*.py modules)	GCP production. rollout.sh refuses any other value (scripts/deploy/rollout.sh:274- 281)	Typed authorize/settle (storage_gcp_authorize.py , storage_gcp_counter_dml.py). The legacy JSON reserve path was removed (billing phase C1)	Covered by spanner-fake (in-process, tests/fakes/spanner.py) and spanner-emulator (native GoogleSQL emulator, opt-in)

Notes:

- TypedBillingStore **capability**. Always check it with typed_billing_store() (storage.py:3954-3968), not isinstance(STORE, ...). isinstance does not see methods that are only reachable through the proxy's __getattr__. That mistake silently routed typed billing down the legacy path once (codex #97).
- **Known divergences are registered as strict xfail**. They are not skips, so they stay visible (tests/conformance/conftest.py:236-313). Examples:
 - _C1_LEGACY_MONEY: native Spanner does not route the Protocol's JSON reserve methods.
 - _NATIVE_ROLLUPS_FROM_CLICKHOUSE_WORKER.
- **A method that exists is not a method that works**. PostgresStore._not_implemented() satisfies mypy and structural checks, then raises at runtime. tests/test_store_protocol_conformance.py:222-280 statically fails if an enclave-facing route calls one of those methods. This rule was added after 11 gateway methods raised on Postgres (router#452, HANDOFF.md:381-389).
- **Analytics sink abstraction**. _StoreProxy.record_provider_benchmark writes to the store first, then calls _ANALYTICS_SINK (storage.py:3912-3934). The in-process ClickHouse sink from PR #291 (src/trusted_router/analytics_sink.py) is **disabled by design**: Cloud Run gives background threads no CPU between requests, and wait_for_async_insert=0 loses rows (docs/storage-portability/analytics-ingestion.md:72-114). The default is NullAnalyticsSink (storage.py:3877).

5. The STORE proxy trap (issue #333)

- **Mechanism**. STORE forwards through __getattr__. Calling monkeypatch.setattr(STORE, "x", fake) installs an instance attribute on the proxy. On teardown, monkeypatch "restores" it as a method bound to a store that no longer exists, and every later test hits the dead store. This once broke 136 tests (AGENTS.md:42-45 ; HANDOFF.md:733-735).

- **Second failure mode.** A proxy-level patch only intercepts *external* callers. A self-call inside the store (for example `ensure_user` calling `self.create_workspace`) binds `self` to the real backend and bypasses the patch (`tests/conftest.py:179-185`).
- **Rule: patch the backend class**, as in `monkeypatch.setattr(InMemoryStore, "create_workspace", ...)`. Two places show the pattern:
- the autouse fixture `tests/conftest.py:168-195`
- the comment at `tests/test_settle_outbox_drain.py:102`
- **Reset between tests.** The autouse `reset_store` fixture puts back a fresh `InMemoryStore` if a test swapped the target, then calls `reset()` (`tests/conftest.py:105-109`).
- **Run the full suite.** Pollution only shows up across files.

6. Running conformance locally

The memory backend alone needs nothing:

```
uv run pytest tests/conformance -q -k "backend=memory"
```

Postgres and Spanner-PG both run against containers and need **no cloud credentials** (`AGENTS.md:95-105`; `tests/conformance/conftest.py:196-200`). The commands below have four steps:

1. Start the Postgres 17 container.
2. Start the Spanner PG dialect container (PGAdapter plus the emulator).
3. Run the suite. `--dist loadgroup` is required with `-n`: the Spanner PG emulator rejects concurrent DDL, and plain `-n 4` produced 14 setup errors (`conftest.py:225-242`; `.github/workflows/ci.yml:207-213`).
4. Stop both containers.

```
# 1. Postgres 17
docker run -d --rm --name tr-conformance-pg -e POSTGRES_PASSWORD=tr -p 5433:5432 postgres:17
export TR_CONFORMANCE_POSTGRES_DSN=postgresql://postgres:tr@localhost:5433/postgres

# 2. Spanner PG dialect (PGAdapter + emulator)
docker run -d --rm --name tr-spanner-pg -p 5434:5432 \
  gcr.io/cloud-spanner-pg-adapter/pgadapter-emulator:latest
export TR_CONFORMANCE_SPANNER_PG_DSN=postgresql://localhost:5434/tr-conformance

# 3. Run the suite
uv run pytest tests/conformance -q -n 4 --dist loadgroup

# 4. Clean up
docker stop tr-conformance-pg tr-spanner-pg
```

The native GoogleSQL emulator backend (`spanner-emulator`) mirrors the CI job at `.github/workflows/ci.yml:34-46`. It is opt-in and requires a loopback host (`tests/conformance/spanner_emulator.py:158-172`). The run below exports the two variables CI sets: `SPANNER_EMULATOR_HOST` and `TR_CONFORMANCE_EMULATOR_SCHEMA=1`.

```
docker run -d --rm --name tr-spanner-emu -p 127.0.0.1:9010:9010 -p 127.0.0.1:9020:9020 \
  gcr.io/cloud-spanner-emulator/emulator@sha256:c6f3402f2599684f295a0fdefb6fbbbf18a0e43e309ff5456ccb452a4570a79
export SPANNER_EMULATOR_HOST=127.0.0.1:9010 TR_CONFORMANCE_EMULATOR_SCHEMA=1
uv run pytest tests/conformance -q -k "backend=spanner-emulator"
```

Traps:

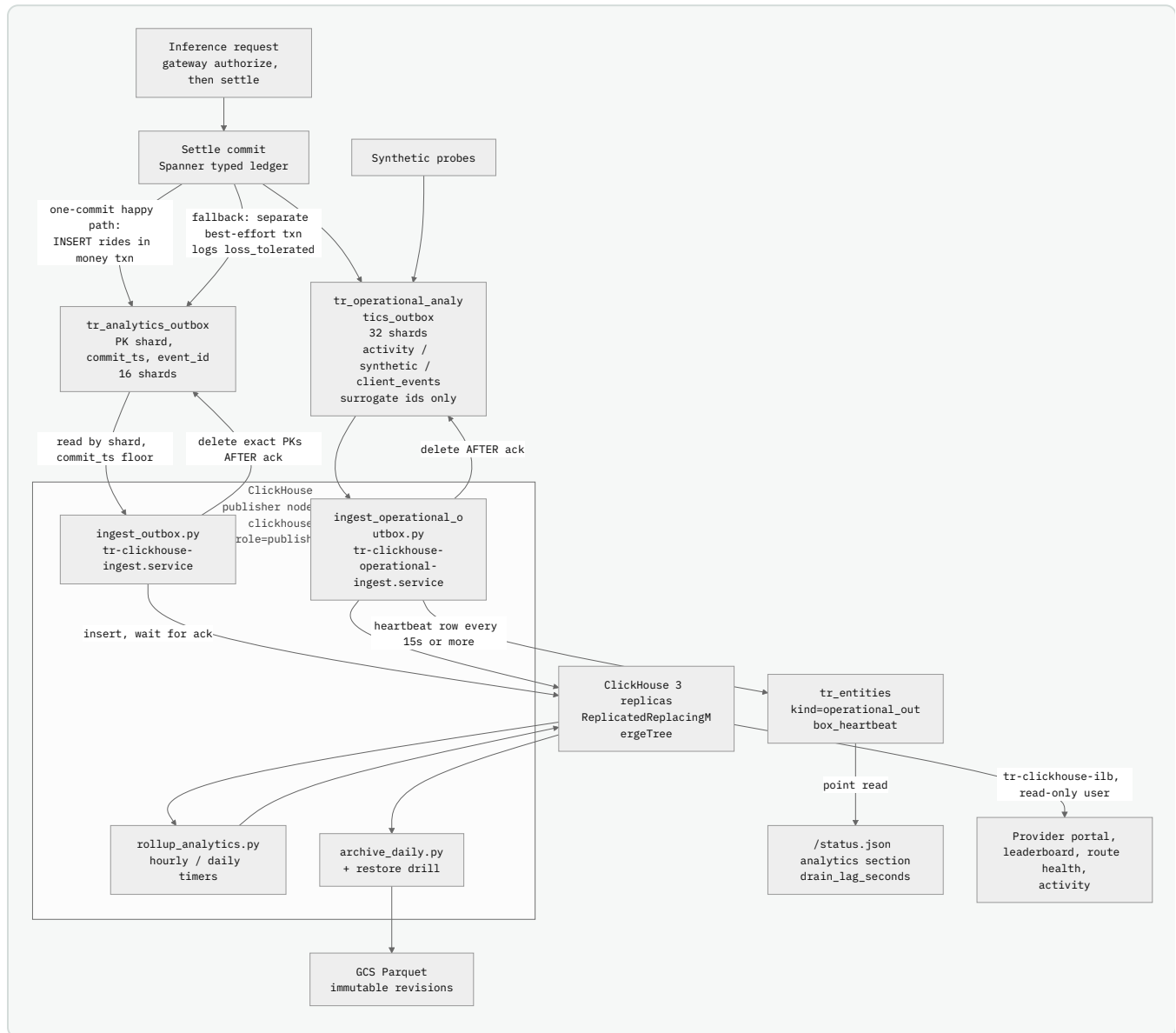
- **A skipped backend proves nothing.** Without the DSN env vars, the `postgres` and `spanner-pg` backends `pytest.skip()` and the run still looks green (`HANDOFF.md:612-619`).
- **Do not weaken a conformance test to make a backend pass** (`AGENTS.md:93`).
- **CI gates.** CI runs all three gates: `uv run ruff check . && uv run mypy && uv run pytest -q` (`AGENTS.md:29-31`). Never run `ruff format .`
- **Not a local-only command.** `scripts.audit_spanner_schema` (`docs/storage-portability/spanner-emulator-conformance.md:130-141`) reads **production** schema metadata. It is an operator tool, not a local test.

7. Migrations and schema

- **Location.** Spanner DDL lives in idempotent shell scripts under `scripts/deploy/`, not in Terraform:
- `infra.sh` for `tr_entities`
- `migrate_typed_counters.sh`
- `migrate_money_primitives.sh`
- `migrate_request_retention.sh`
- `migrate_spend_lease.sh`
- `migrate_receipt_key_versions.sh`
- `migrate_generation_records.sh`
- `migrate_analytics_outbox.sh`
- `migrate_operational_analytics_outbox.sh`
- `migrate_async_settle_*.sh`
- `migrate_fastpath.sh`
- `migrate_entity_ttl.sh`
- `migrate_gateway_request_index.sh`
- `migrate_speculation_shadow.sh`
- `migrate_trust_reconciliation.sh`
- **Pattern.** Each script checks `INFORMATION_SCHEMA` and only adds what is missing. Most are additive. Several are dry-run unless you pass `--apply` (for example `migrate_request_retention.sh:2-3`). The script header states any prerequisite ordering.
- **When they run.** The `migrate-schema` job in `.github/workflows/deploy.yml:242-335` runs them *before* `deploy` (`deploy.yml:415, :451`). A merge can therefore migrate the schema even if the deploy is later skipped (`multi-cloud-separation.md §7`, around line 410).
- **Schema drift guard.** `tests/conformance/spanner_ddl.py` is *generated* from those scripts (`python -m tests.conformance.spanner_schema_source`). It pins a SHA-256 per source script (`spanner_ddl.py:7-24`). CI provisions this DDL on the emulator. Allowlists live in `spanner_schema_drift_allowlist.json` and `spanner_ddl_exemptions.json`.
- **Postgres** applies `storage_postgres_schema.sql` from `PostgresStore.apply_schema()` (`storage.py:4029-4030`). Aurora DSQL has two restrictions:
- It rejects `DESC` index keys, so all indexes are ascending (`storage_postgres_schema.sql:25-35`).
- It needs `CREATE INDEX ASYNC`.
- **Rules:**
- Never run large unbatched DML against production Spanner during a rolling deploy (`AGENTS.md:68`).
- Roll out typed readers before writers. Do not roll back past a revision that can read typed tables once typed rows exist (`docs/spanner-reliability.md:350-353`).
- Repeat cloud-local migrations in every cloud. A GCP success says nothing about AWS or Azure (`multi-cloud-separation.md:30-46`).

- **ClickHouse DDL** is numbered `clickhouse/001_...sql` through `023_...sql`. It comes in replicated (GCP cluster) and `*_single_node.sql` (AWS/Azure) variants. Apply scripts:
- `scripts/deploy/clickhouse_replicated_migrate.sh`
- `_clickhouse_single_node_schema.sh`
- `aws_eu_clickhouse_schema_apply.sh`

8. Analytics pipeline: request → outbox → ClickHouse

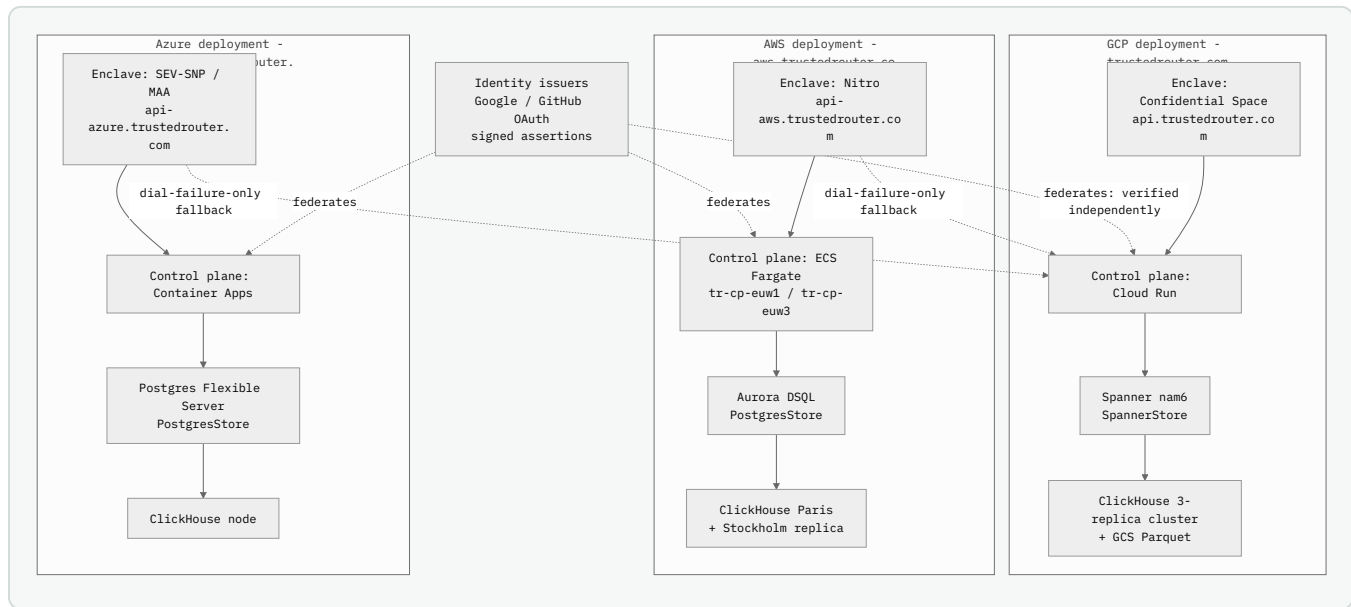


Key properties:

- **At-least-once delivery, made safe.**
- The ClickHouse ack happens *before* the exact Spanner PK delete, and the delete is the cursor (`clickhouse/ingest_outbox.py:1-21`).
- A crash replays the batch. Exact readers use `FINAL` over `ReplacingMergeTree` (`docs/clickhouse-reliability.md:43-45`).

- **The cursor is `commit_ts`, never `created_at`.** An event-time cursor permanently strands late commits (`HANDOFF.md:701-703`; `analytics-ingestion.md:196-213`).
- **Shard by hash.** A blake2b hash of `event_id` picks the shard, so the monotonic `commit_ts` does not hot-spot one Spanner split (`src/trusted_router/storage_gcp_analytics_outbox.py:12-36`). There are 16 benchmark shards and 32 operational shards (`src/trusted_router/storage_operational_analytics.py:31`).
- **Analytics never destabilizes money.**
- The benchmark enqueue is best-effort.
- It rides inside the settle commit only on the one-commit path, which has a full two-commit fallback (`storage_gcp_analytics_outbox.py:1-10`; `AGENTS.md:83-87`).
- Failures log `loss_tolerated` / `repairable_via` (`src/trusted_router/storage_gcp_generations.py:152-230`).
- **Privacy split.** Operational rows carry one-way surrogates (`analytics_surrogate`, `storage_operational_analytics.py:67`). Raw workspace and key ids never leave Spanner (`storage_gcp_operational_analytics_outbox.py:1-12`).
- **No additive materialized views.** MVs run per insert block, so replays inflate them: one measurement saw 3 loads of 30,832 rows produce 92,500 aggregate samples (`HANDOFF.md:713-715`). Rollups instead recompute into a staging table, check `sum(attempts)`, then replace the partition atomically (`docs/clickhouse-reliability.md:94-109`).
- **Drain freshness is published out of band.**
- Every control plane publishes `analytics.{available, backend, drain_lag_seconds, outbox_depth, generated_at}` in `/status.json` (`docs/clickhouse-reliability.md:279-311`).
- On Spanner the plane reads the poller's heartbeat row instead of the outbox. Reading the outbox head walks deleted versions at about 0.9 s CPU per call.
- A missing or stale heartbeat (>180 s) publishes `poller_stale` (`clickhouse-reliability.md:313-326`).
- **Deploy order:** install the VM poller first, then the control plane (`clickhouse-reliability.md:123-131`).
- **Postgres drain (AWS/Azure).** `clickhouse/ingest_operational_outbox_postgres.py` has **no timestamp cursor**, because `now()` is transaction start time. Instead it loops `SELECT` → `insert` → `DELETE`. It retries `SQLSTATE 40001`, which DSQL raises for routine optimistic-concurrency aborts (lines 1-30).
- **Direct sink (optional).** `operational_analytics_direct.py` writes operational telemetry straight to ClickHouse. The polling drain once cost about 25% of Spanner CPU and helped collapse authorize/settle on launch day (2026-08-25) (lines 1-27). GCP still pins `TR_OPERATIONAL_ANALYTICS_SINK=outbox` (`scripts/deploy/rollout.sh:602`). The direct sink is available but **not selected** on GCP.
- **Topology.**
- Three nodes: `tr-clickhouse-1/2/3` in `us-central1-a/b/c`, `e2-standard-4`, 500 GB SSD each, no external IP, a three-voter Keeper, one logical shard (`docs/clickhouse-reliability.md:31-62`).
- Workers run only on the node whose metadata says `publisher`. Takeover is fenced (`clickhouse-reliability.md:233-257`).
- **Never restart all three nodes together** (`clickhouse-reliability.md:153-154`).
- **Sharding gate.**
- Add a shard when peak rows/s exceeds 25% of measured replicated ingest, when p95 query latency exceeds 250 ms, or when replication lag exceeds 30 s (`docs/clickhouse-sharding.md:15-25`).
- Distribute tenant activity by the opaque tenant hash and provider data by event id.
- Never shard on a raw workspace id.

9. Multi-cloud separation



Credits, API keys, workspaces, usage and analytics do not cross the dotted lines. Only identity federates.

- **Why identity federates and money does not.**
- An identity is a signed assertion. Every cloud can verify it on its own, even when partitioned from the others.
- A balance has a conservation law. Sharing it would need a single authority (the cross-cloud hop lands on the hot path) or 2PC (availability becomes the AND of both clouds). Partitioning the money is the only option that keeps the clouds independent (multi-cloud-separation.md:50-81).
- Moving credit between clouds would be a new feature: an explicit, asynchronous, idempotent transfer shaped like the settle outbox (multi-cloud-separation.md:83-92).
- **Product consequence.** The UI must show balances per cloud. A 402 must name the cloud it checked. Keys must be visibly scoped to one cloud (multi-cloud-separation.md:96-107).
- **Dial-failure-only fallback.** The enclaves try their own plane first and fall back to trustedrouter.com only on a dial failure. Each plane has its own database and the idempotency key does not travel, so retrying a request that reached a server could escrow or bill twice (HANDOFF.md:266-287; enclave-go/internal/trustedrouter/endpoints.go).
- **Keyless WIF.** The GCP pool multicloud with provider aws-workloads exists for operations only. Do not build data flow on it (HANDOFF.md:209-218).
- **Definition of done for a cloud.** A cloud is done when scripts/deploy/verify_cloud_complete.sh <cloud> exits 0.
- It checks five stages (registry, analytics section present, available, lag bound, outbox enabled), then requires two in-cloud count() readings ten minutes apart that show growth (multi-cloud-separation.md:204-273).
- Exit codes: 0 = VERIFIED, 5 = NOT YET OBSERVABLE, 1 = NOT VERIFIED (multi-cloud-separation.md §"Exit codes").
- The rule comes from AWS-EU. It ran for 15 days with no drain installed and accumulated about 470k undelivered outbox rows with no alarm (multi-cloud-separation.md:210-227).

10. Query-safety rules for operators

1. **Read-only is not load-safe.** -Never search production `tr_entities.body` with `LIKE`, regex or JSON predicates unless you have a complete indexed key. - `LIMIT` caps the rows returned, not the rows scanned (`AGENTS.md:70-79`).
2. **Never discover entity kinds** with wildcard `COUNT(*)` queries. Use code (the `_read/_write_entity` call sites) or bounded `SPANNER_SYS` table-size stats (`docs/spanner-reliability.md:303-316`).
3. **Payment attribution.** Use `uv run python -m scripts.inspect_payment_owner ...` (`docs/spanner-reliability.md:188-208`). Missing metadata is a **stop condition**, not permission to full-scan.
4. **Request, token, provider and latency analytics come from ClickHouse.** Never aggregate raw Spanner generation, entity or outbox rows (`docs/spanner-reliability.md:183-186`).
5. **CPU investigations** use bounded `SPANNER_SYS.TXN_STATS_TOP_MINUTE` and `LOCK_STATS_TOP_MINUTE` (`docs/spanner-reliability.md:221-251`).
6. **Never loosen the 45% CPU alarm** to make an operator scan pass (`AGENTS.md:79`).
7. **Periodic jobs must read through a covering index** (`docs/spanner-reliability.md:177-181`).
8. **No large unbatched DML during a rolling deploy.** Use batched or partitioned DML only after confirming there are no readers (`AGENTS.md:68` ; `HANDOFF.md:244-246`).
9. **ClickHouse access.** -Connect through an SSH local forward over IAP. A plain IAP TCP tunnel is correctly refused by the `10.0.0.0/8` user ACL (`HANDOFF.md:724-731`). -When comparing replicas, use `FINAL` and a fixed cutoff at least 5 minutes in the past (`docs/clickhouse-reliability.md:197-208`).
10. **SQL NULL is three-valued.** Use `ifNull()` when porting Python predicates (`HANDOFF.md:708-711`).
11. **Never write** `id LIKE 'prefix%'` **with unescaped** `_`. Use `_like_prefix` (`HANDOFF.md:626-628`).

11. HANDOFF status summary (`docs/storage-portability/HANDOFF.md`)

The HANDOFF was last updated 2026-08-05, with later inline notes up to 2026-09-14. Where newer docs and code supersede it, this summary is reconciled against them.

Done

ITEM	EVIDENCE
Separation decision	#307 (<code>HANDOFF.md:30</code>)
<code>PostgresStore</code> merged and passing conformance, incl. Spanner PG dialect	#310, #322 (<code>HANDOFF.md:31-32</code>)
ClickHouse 3-zone cluster, durable outbox ingestion, Parquet archive, rollups, private reads	<code>HANDOFF.md:33-39</code> ; <code>docs/clickhouse-reliability.md</code>
AWS and Azure serving and attesting; AWS and Azure (uaenorth) own control planes	<code>HANDOFF.md:40-43</code>
Postgres can serve <code>authorize</code> (11 <code>NotImplementedError</code> methods fixed)	router#452 (<code>HANDOFF.md:381-389</code>)
TLS-resumption attestation mis-bind fixed	qcp#108 (<code>HANDOFF.md:390-397</code>)
Bigtable analytics retired (2026-09-28/29)	<code>docs/clickhouse-reliability.md:133-142</code>
Request-state TTLs; <code>rate_limit</code> entity TTL	<code>docs/spanner-reliability.md:318-348</code> ; <code>migrate_entity_ttl.sh</code>
Azure operational-analytics outbox and drain live	<code>multi-cloud-separation.md</code> §"What is missing right now"

[STALE] in HANDOFF. Items §4.2 (route-health cutover), §4.3 (leaderboard proof) and §4.4 (reconciler repair) describe Bigtable-era work. The `prove_*.py` and reconcile scripts were deleted once ClickHouse became the only store (`docs/storage-portability/README.md:6-11`). Treat those items as closed or moot.

Next / open

- **Issue #334: `tr_entities` cleanup.** Dead and duplicate historic rows still need a reviewed, batched removal (`HANDOFF.md:230-246`; `docs/spanner-reliability.md:331-333`).
- **ACME has no fallback (#56).** A Let's Encrypt outage takes down TLS for GCP and Azure (`HANDOFF.md:398-405`).
- **Azure hostdata is published nowhere.** Third parties cannot verify Azure attestation (`HANDOFF.md:406-412`).
- **Azure region two (australiaeast).** Blocked on four human-applied IAM grants. There is no cross-region failover yet (`HANDOFF.md:448-538`).
- **Azure shared ACME cache.** Seal `tr-cross-cloud-sa-key` into the bundle and set `QUILL_ACME_CACHE_GCS_BUCKET` (`HANDOFF.md:540-599`).
- **The AWS control plane's catalog lags GCP (448 vs 451 models).** Reconcile it (`HANDOFF.md:91-92`).
- **Nines cannot yet be demonstrated.** Sample counts are too low (`HANDOFF.md:418-431`).
- **Fleet freshness workflow is `workflow_dispatch` only.** Enable the cron once every cloud publishes analytics (`docs/clickhouse-reliability.md:348-360`).
- **Direct operational sink.** Decide whether to cut GCP over from the outbox to cut Spanner CPU (`operational_analytics_direct.py:1-27`; `rollout.sh:602`).
- **Citus re-validation.** Required before Azure scales beyond Flexible Server (`azure_control_plane.sh:33-39`).

Traps (most costly first)

- **Green status pages can lie.** Peer planes answered 200 while their `authorize` path was broken, because the probes went through enclaves that dialled GCP. Always ask which backend actually served the request (`HANDOFF.md:605-610`).
- **A conformance backend that skips proves nothing.** Stand up the containers (`HANDOFF.md:612-619`).
- **STORE is monkeypatch-hostile.** Patch the class (#333).
- **`created_at` is not commit order.** Never build a cursor on it.
- **A ClickHouse MV is not covered by `ReplacingMergeTree`.**
- **A deploy that pins a value is not done until the code reading it is deployed.** A component with no code or DNS reference can still be load-bearing: `tr-eu` ran the probes through EventBridge (`HANDOFF.md:666-680`).
- **The full test suite dirties tracked OG assets.** It rewrites `static/og/providers/*`. Do not `git commit -a` after running it (`HANDOFF.md:691-699`).
- **Credentials need two env vars.** You need both `GOOGLE_APPLICATION_CREDENTIALS` and `CLOUDSDK_AUTH_CREDENTIAL_FILE_OVERRIDE`. Never export them in a shell profile (`HANDOFF.md:746-756`).

06 The Deploy Pipeline

Inside the deploy job



Then a 3-minute canary on router_core, plus a billing-path gate.
Secondaries ramp serially, each with the same watchdog gates.

PART IV • CHAPTER 10 • HERO STILL: VIDEO 06 AT 1:03

Infrastructure & deploys

Topology, CI/CD, rollout.sh and ENV_VARS, Terraform, runbooks and incidents.

- 1. Deployment topology
- 2. Push-to-main to production
- 3. GitHub Actions workflows
- 4. ENV_VARS inventory (config-as-code)
- 5. Rollout control-plane mechanics
- 6. Terraform (infra/)
- 7. fastpath/ and release/
- 8. Observability
- 9. Runbook and incident index
- 10. Deploy safety rules

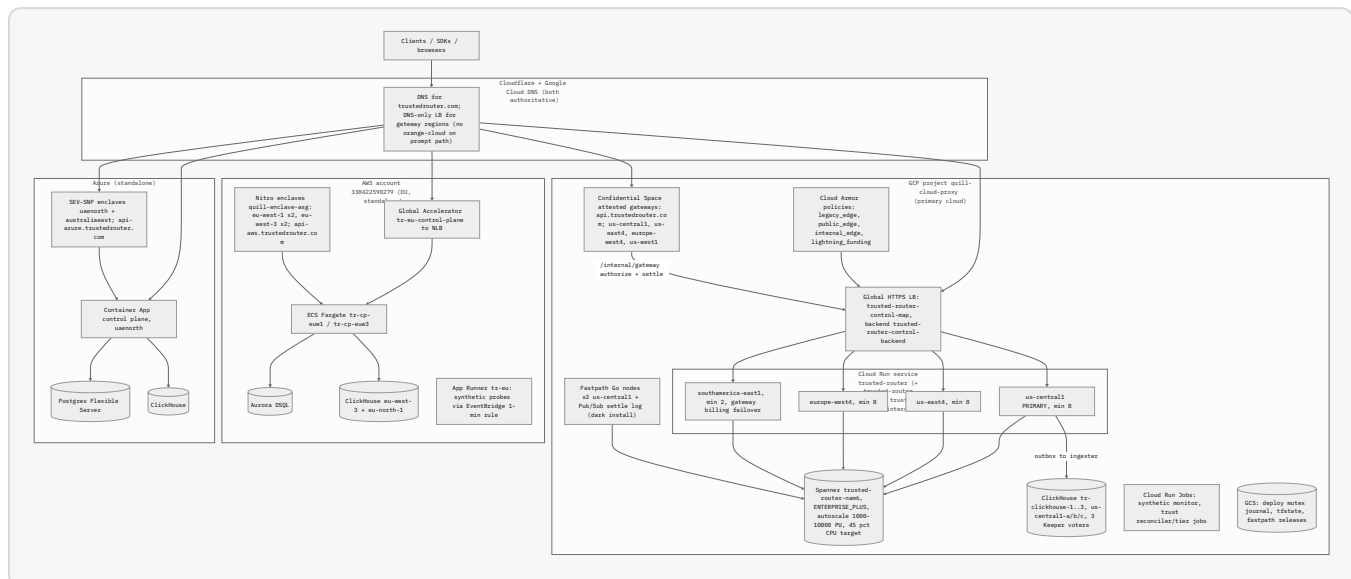
Storyboard · Video 06: The deploy pipeline



Source repo: `quill-router` (TrustedRouter control plane) at `main` @ 778120494, read on 2026-10-19. Every claim cites `path:line` relative to the repo root. Nothing here was checked against live production. Where docs and code disagree, both are noted.

Trust boundary in one sentence. This repo deploys the **control plane** (billing, keys, routing metadata, website/console). The **attested prompt path** (Confidential Space / Nitro / SEV-SNP enclaves) is deployed from the separate `quill-cloud-proxy` repo (`.github/workflows/deploy.yml:1-7`). If a control-plane deploy goes wrong, the site goes down or keys stop working. Prompt confidentiality is not affected.

1. Deployment topology

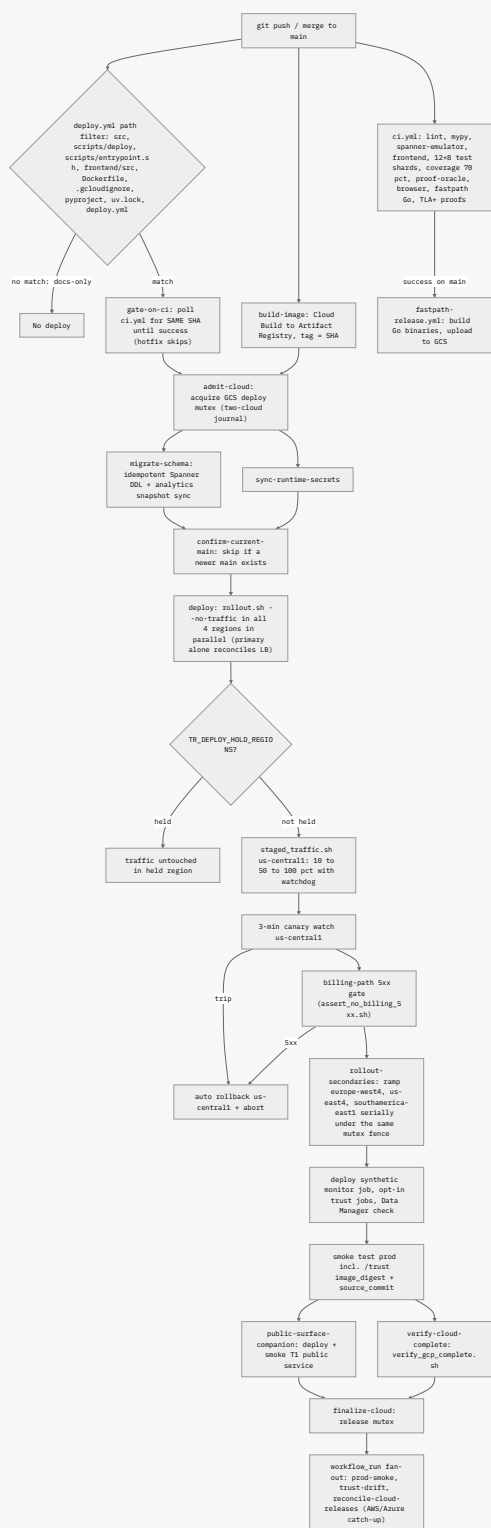


Facts behind the diagram:

FACT	SOURCE
GCP project <code>quill-cloud-proxy</code> , primary region <code>us-central1</code> , service <code>trusted-router</code>	<code>scripts/deploy/_lib.sh:8, :22, :102</code>

FACT	SOURCE
Control-plane regions <code>us-central1,us-east4,europe-west4,southamerica-east1</code> ; warm set excludes São Paulo	<code>scripts/deploy/_lib.sh:30, :37</code>
Min instances 8/8/8/2; concurrency 8; max 12 per revision (sized against the 45% Spanner CPU alarm); memory 2Gi	<code>scripts/deploy/_lib.sh:46-63, :88</code>
Ingress <code>internal-and-cloud-load-balancing</code> , so <code>run.app</code> cannot bypass Cloud Armor	<code>scripts/deploy/_lib.sh:93</code>
Enclave regions <code>us-central1, us-east4, europe-west4, us-west1</code> ; <code>us-west1</code> is gateway-only with no Cloud Run	<code>src/trusted_router/enclave_regions.py:7, :19;</code> <code>scripts/deploy/_lib.sh:10-21</code>
LB backend <code>trusted-router-control-backend</code> , Serverless NEG per region attached by <code>rollout.sh</code>	<code>scripts/deploy/rollout.sh:1-5, :1036-1110</code>
Spanner <code>trusted-router-nam6</code> , <code>nam6</code> , ENTERPRISE_PLUS, autoscaler 1000-10000 PU, 45% HP CPU target, <code>prevent_destroy</code>	<code>infra/spanner_trusted_router.tf:10-38</code>
Cloud Armor policies, which <code>tr-deploy</code> can attach but cannot edit	<code>infra/cloud_armor.tf:1-25, :30, :161, :229, :285</code>
Each cloud is a standalone deployment. Identity federates. Credits, keys and analytics stay in their own cloud.	<code>docs/storage-portability/multi-cloud-separation.md §1</code>
GCP/AWS/Azure gateways, TEEs and control planes; AWS Fargate behind Global Accelerator	<code>docs/storage-portability/HANDOFF.md:66-77</code>
AWS "what runs where" (enclave ASG, ECS, App Runner <code>tr-eu</code> runs synthetics)	<code>docs/storage-portability/HANDOFF.md:103-123</code>
ClickHouse GCP cluster: 3 nodes, <code>us-central1-a/b/c</code>	<code>docs/storage-portability/HANDOFF.md:33, :175-189</code>
AWS ClickHouse <code>eu-west-3</code> (Paris) + <code>eu-north-1</code> (Stockholm)	<code>scripts/deploy/aws_eu_clickhouse.sh:28,</code> <code>scripts/deploy/aws_eu_north_clickhouse.sh:53</code>
Azure: Container Apps in <code>uaenorth</code> on Postgres Flexible Server; enclaves <code>uaenorth</code> + <code>australiaeast</code>	<code>scripts/deploy/azure_control_plane.sh:2, :66,</code> <code>:111</code>
Control-plane URLs per cloud: <code>trustedrouter.com</code> / <code>aws.</code> / <code>azure.</code>	<code>scripts/deploy/cloud_rollout.py:36-40</code>
Cloudflare and Google Cloud DNS are both authoritative; region drain goes through Cloudflare DNS-only LB with no proxying on the prompt path	<code>docs/runbook.md:112-125, :1324-1370</code>
Fastpath: 2 nodes in <code>us-central1</code> , no external IP, empty <code>workspace allow-list</code>	<code>infra/fastpath.tf:1-24, :231</code>

2. Push-to-main to production



Step-by-step citations (all `.github/workflows/deploy.yml` unless noted):

1. **Trigger and path filter:** :26-37 . Manual `workflow_dispatch` with `hotfix: true` skips the CI gate and the watchdog baseline (:38-43 , :73 , :516-524).
2. **Concurrency:** group `deploy-trusted-router` , `cancel-in-progress: false` . One deploy runs at a time, with one pending slot that a newer push can replace. The GCS mutex is what actually serializes traffic changes (:45-49).
3. **gate-on-ci** polls ci.yml's conclusion for the same SHA (:57-107). ci.yml and deploy.yml start in parallel on push, so without this gate a red commit could roll out everywhere (:17-21).
4. **build-image:** Cloud Build pushes `${REGION}-docker.pkg.dev/${PROJECT_ID}/${REPO}/${SERVICE}:${SHA}` and uses a registry `buildcache` tag for layer caching (:109-208). Auth uses WIF to `tr-deploy@quill-cloud-proxy` , with no JSON keys (:9-12 , :133).
5. **admit-cloud** takes the production mutex with `deploy_mutex.sh acquire` (:209-240). The wait is long because a quill-cloud-proxy gateway rollout can hold the lease for about 2h20m (:213-216).
6. **migrate-schema** runs idempotent `migrate_*.sh` steps (money primitives, retention, receipts, outboxes, async-settle admission, fastpath tables) and syncs analytics snapshots (:242-368).
7. **sync-runtime-secrets** validates SES credentials and pushes the ClickHouse reader credential (:370-407).
8. **confirm-current-main** stops a stale push from deploying over a newer main (:409-448).
9. **deploy:** - Re-asserts the mutex (:500-501) and captures pre-deploy revisions for rollback (:526). - Warms all 4 regions with no traffic in parallel (:544-600). Only the primary process may mutate the global LB (`TR_DEPLOY_RECONCILE_LB` , :594-597). - Resolves the primary baseline and honours holds (:657-712). - Staged 10/50/100 ramp in us-central1 (:713-723), then a 3-min canary (:734-742) with automatic rollback (:744-764). - Billing-path 5xx gate (:766-788). - Never-ramped regions have their staged-probe tags swept on failure, because a tagged revision keeps its minimum instances, which costs money (:882-901).
10. **rollout-secondaries:**
 - Imports the same mutex fence (:918-937) and ramps the secondaries serially (:958-970). The order is europe-west4, us-east4, southamerica-east1 (`scripts/deploy/ramp_secondaries.sh:237`).
 - Then deploys the synthetic monitor Cloud Run Job (:972), the opt-in trust jobs (:994) and the Google Data Manager guard (:1005).
 - Smoke-tests prod (:1026-1183), including asserting that `source_commit` , `image_reference` and `image_digest` appear in the trust records (:1160-1170).
11. **public-surface-companion** deploys the isolated T1 public service and runs outside the mutex (:1262-1338).
12. **verify-cloud-complete** runs `bash scripts/deploy/verify_gcp_complete.sh` with `if: always()` (:1340-1399). The long comment at :1340-1380 records why an echo-only stub is unacceptable.
13. **finalize-cloud** releases the mutex and records success or failure (:1401-1428).

The runbook's definition of done: "**deployed** means the primary is live" (all four regions warm, plus us-central1 ramp and canary, about 9 min). Full convergence is the later `rollout-secondaries` + `verify-cloud-complete` (`docs/runbook.md:395-410`). `AGENTS.md:56-58` adds that a queued or partially completed rollout must never be called deployed.

3. GitHub Actions workflows

WORKFLOW	TRIGGER	WHAT IT DOES
<code>ci.yml</code>	push/PR to main, dispatch (:3-11)	The release gate. Jobs: <code>lint</code> (ruff check, mypy, model-changes check, :18-32); <code>spanner-emulator</code> (GoogleSQL conformance, prod SQL acceptance, schema drift, :34-100); <code>frontend</code> (TS build matches committed JS, ESLint, Stylelint, :102-118); <code>production-import</code> (--no-dev import of <code>trusted_router.main</code> , :120-130); <code>test</code> , 12 shards via <code>pytest-split</code> with <code>-n 4</code> and <code>--dist loadgroup</code> , 25-min cap (:132-284, shards :180); <code>coverage</code> , which combines shards and enforces ≥70% once (:286-332); <code>test-post-cutover</code> , 8 shards with the lifecycle clock pinned past cutovers (:348-415); <code>proof-oracle</code> , the frozen-main differential, 6 shards × 2 clocks, no coverage (:419-457); <code>browser</code> (Playwright smoke, :459-503); <code>byok-format-ordering</code> (:505-542); <code>fastpath</code> , <code>fastpath-spanner</code> , <code>fastpath-walks</code> (Go 1.24 module and TLC, :544-670); <code>proofs</code> (./proofs/check.sh, :672-684). Split rationale: wall time becomes the slowest job, and deploy gates on the whole run (:13-16).
<code>deploy.yml</code>	push to main with path filter; dispatch hotfix (:26-43)	GCP Cloud Run control-plane deploy (section 2).
<code>prod-smoke.yml</code>	workflow_run after deploy, cron */30, dispatch (:1-14)	pytest tests/test_smoke_prod.py against trustedrouter.com. Never fails PRs. Uses <code>TR_STRIPE_WEBHOOK_SECRET</code> for the webhook drift guard (:17-45).
<code>trust-drift.yml</code>	daily 07:23 cron, workflow_run after deploy, dispatch (:3-38)	<code>scripts/verify_trust_measurements.py --strict</code> compares published enclave measurements with live attestations (:108-112). Was hourly and changed to daily, because drift only comes from deploys or enclave rebuilds (:4-22).
<code>refresh-price.yml</code>	hourly at :37, dispatch (:20-27)	The price bot. Pulls provider keys from Secret Manager, refreshes prices with LLM "self-heal" for broken parsers, sanity-checks price spikes, coverage audit, routing evidence (:74-332). Pushes a temp branch, dispatches <code>ci.yml</code> on it (GITHUB_TOKEN pushes don't trigger workflows), then advances protected main with the checked SHA (:346-420). Runs one at a time (about 30-45 min). This is why <code>main</code> moves hourly (AGENTS.md:47-49).
<code>fastpath-release.yml</code>	workflow_run of CI success on main (:7-11)	Builds <code>fastpath</code> , <code>fastpathctl</code> , <code>loadgen</code> , <code>fastpathwatch</code> statically, writes SHA256SUMS, and uploads to <code>gs://tr-fastpath-releases-quill-cloud-proxy/fastpath/<sha>/</code> . Nodes pick up the release pinned in <code>infra/fastpath.tf:19</code> .
<code>deploy-aws-control-plane.yml</code>	dispatch only; inputs release_sha, mode (deploy/drain-install), promotion_only, bake_override_reason (:1-25)	Rolls ECS one region at a time via <code>aws_ecs_control_plane.sh</code> (:108-114). That script requires CI green for the exact SHA, the right AWS account and a clean checkout (<code>scripts/deploy/aws_ecs_control_plane.sh:44-61</code>), and rolls a failed region back to its previous task definition (:75-81).
<code>deploy-azure-control-plane.yml</code>	dispatch only; inputs release_sha, bake_override_reason, promotion_only (:1-16)	Requires green CI, then Azure OIDC + GCP WIF, then <code>azure_control_plane.sh</code> (:50-138).

WORKFLOW	TRIGGER	WHAT IT DOES
reconcile-cloud-releases.yml	after deploy, cron :07:22/:37:52, dispatch	scripts.deploy.reconcile_cloud_releases dispatches stale AWS/Azure planes to the last fully verified GCP release (scripts/deploy/reconcile_cloud_releases.py:2).
cloud-staleness.yml	daily 06:17, dispatch (max_age_hours=72)	Flags any plane whose serving release is stale.
infra-apply.yml	PR/push touching infra/**, dispatch (terraform/status/configure-retention/recover x gcp/aws/azure)	terraform fmt/validate on PR. On merge it plans and applies the whole root with no plan gate (infra/README.md:38-46). It also has a mode that inspects or reconciles the deploy journal (:111).
gateway-reliability.yml	dispatch with APPLY	gateway_reliability.sh --apply reconciles alert policies.
typed-audit.yml	daily 11:43	Typed billing invariant audit + production schema audit against the emulator.
provider-catalog-health.yml	hourly :17	pytest -m provider_health (live provider state, kept out of CI).
check-analytics-freshness.yml	every 6h, push on own paths, dispatch	Out-of-band backstop for per-cloud outbox-to-ClickHouse drains. Exists because AWS-EU went 15 days and 470k rows without a drain (:4-15).
check-archive-freshness.yml	daily 07:30	ClickHouse daily archive presence.

WORKFLOW	TRIGGER	WHAT IT DOES
check-client-telemetry-freshness.yml	06:45/18:45	Client telemetry freshness via status.json.
cloud-security-baseline.yml	daily 08:15	GCP/Azure security-baseline drift scripts.
check-repo-security-coverage.yml	weekly Mon 07:40	Repo security settings coverage.
dependency-scan.yml	PR/push/daily 09:23	Vulnerability scan with a new-vs-existing diff, uploaded to code scanning.
daily-embeddings-probe.yml	daily 13:17	Live embeddings probe (Cohere optional).
mirror-repo.yml	daily 05:41	Mirrors the repo to GCS (GitHub-outage insurance).
proofs-guardians.yml	PR/push on proofs/**	TLA+ guard-table sweep, 16 shards.
reshard-billing-workspace.yml	dispatch (status/prepare/finish/online-split, APPLY)	Billing workspace shard changes.
deploy-growth-sync.yml	dispatch with APPLY	Marketing analytics worker.

WORKFLOW	TRIGGER	WHAT IT DOES
token-exchange-sites.yml	PR/push on token-exchange paths	Token Exchange static sites and evidence tests.
lightning-experiment.yml	PR/push on lightning paths	LightningRouter experiment checks.
refresh-test-durations.yml	dispatch	Refreshes <code>.test_durations</code> used for shard balancing.

4. ENV_VARS inventory (config-as-code)

ENV_VARS is defined at `scripts/deploy/rollout.sh:401-676`. It is joined with `|` (:677) and applied with `--set-env-vars` plus `TR_SERVING_REGION` per region (:851). **Setting a Cloud Run env var by hand does nothing, because the next rollout overwrites it** (AGENTS.md:64-66). Secrets are separate: they are listed in `SECRET_ENVS` and pulled from Secret Manager (:101-112). Names only below. No values are listed.

GROUP	ENV VAR NAMES (ROLLOUT.SH LINE)
Identity / surface / release	TR_ENVIRONMENT 407, TR_SERVICE_SURFACE 408 (=combined, legacy bridge), TR_OPERATOR_IDENTITY 409, TR_ALLOW_DEPLOYED_COMBINED_SURFACE 410, TR_RELEASE 417, TR_SOURCE_REVISION 418, TR_GCP_PROJECT_ID 434, TR_REGIONS 477, TR_PRIMARY_REGION 478, (+ TR_SERVING_REGION injected 851)
Routing / domains / providers	TR_API_BASE_URL 423, TR_TRUSTED_DOMAIN 424, TR_TRUSTED_DOMAIN_ALIASES 425, TR_ENABLE_LIVE_PROVIDERS 422, TR_REMEDIATOR_IN_PROCESS_ENABLED 421, VERTEX_PROJECT_ID 480, VERTEX_LOCATION 481, TR_USER_MODELS_DISPATCH_ENABLED 464
Speculation shadow (off)	TR_SPECULATIVE_PROVIDER_SHADOW_ENABLED 403, TR_SPECULATION_SHADOW_WORKSPACES 404, TR_SPECULATION_SHADOW_ROUTES 405, TR_SPECULATION_SHADOW_IMAGES 406
Rate limiting / admission	TR_RATE_LIMIT_ENABLED 415, TR_SETTLE_PER_KEY_INFLIGHT_LIMIT 416
Storage (Spanner)	TR_STORAGE_BACKEND 426, TR_SPANNER_INSTANCE_ID 472, TR_SPANNER_DATABASE_ID 473, TR_SPANNER_POOL_SIZE 479, TR_GENERATION_RECORDS_ENABLED 474, TR_REQUEST_RECORD_WRITE_MODE 608
KMS / encryption	TR_BYOK_KMS_KEY_NAME 475, TR_GOOGLE_DATA_MANAGER_KMS_KEY_NAME 476
Trust / attestation pins	TR_TRUST_GCP_SOURCE_COMMIT 482, TR_TRUST_GCP_IMAGE_REFERENCE 483, TR_TRUST_GCP_IMAGE_DIGEST 484, TR_TRUST_GCP_RELEASE_URL 485, TR_TRUST_GCP_RELEASE_FALLBACK_URLS 486, TR_TRUST_AWS_RELEASE_URL 494, TR_TRUST_AZURE_RELEASE_URL 495, TR_SPEND_LEASE_ACCEPTED_GCP_IMAGE_DIGESTS 643
Trust reconciler / spend lease / heartbeat	TR_SPEND_LEASE_TRUST_ELIGIBILITY_ENABLED 637, TR_TRUST_STRIPE_ACCOUNT_ID 638, TR_TRUST_QUALIFYING_PROVIDERS 639, TR_TRUST_RECONCILE_INTERVAL_SECONDS 640, TR_TRUST_RECONCILE_MAX_AGE_SECONDS 641, TR_HEARTBEAT_GRACE_SECONDS 642

GROUP	ENV VAR NAMES (ROLLOUT.SH LINE)
Auth / OAuth / SIWE	TR_GOOGLE_OAUTH_REDIRECT_URL 496, TR_GITHUB_OAUTH_REDIRECT_URL 497, TR_SIWE_DOMAIN 498
Signup / growth	TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS 429, TR_SOCIAL_REWARDS_ENABLED 431, TR_SOCIAL_REWARDS_PLANE_ID 432, TR_SOCIAL_REWARDS_ISSUER_ID 433, TR_ACTIVATION_REMINDER_INTERVAL_SECONDS 511
Identity verification / SMS	TR_NOTIFY_ENABLED 443, TR_NOTIFY_SMS_AVAILABLE 444, TR_VERIFF_ENABLED 455, TR_VERIFF_BASE_URL 465, TR_CUSTOM_MODELS_REQUIRE_VERIFICATION 456, TR_TELNYX_FROM_NUMBER 466, TR_TELNYX_TEXML_ACCOUNT_ID 467, TR_TELNYX_TEXML_APPLICATION_ID 468, TR_TWILIO_FROM_NUMBER 471
Email (SES) / notices / support	TR_AWS_REGION 499, TR_SES_FROM_EMAIL 500, TR_SES_FROM_NAME 501, TR_SES_ALERT_FROM_EMAIL 502, TR_SES_ALERT_FROM_NAME 503, TR_SES_ALERT_CONFIGURATION_SET 504, TR_SES_NOTICES_FROM_EMAIL 505, TR_SES_NOTICES_FROM_NAME 506, TR_SES_NOTICES_CONFIGURATION_SET 507, TR_RETIREMENT_NOTICES_MODE 514, TR_RETIREMENT_NOTICES_PREVIEW_EMAIL 515, TR_SUPPORT_EMAIL 516, TR_PARTNER_INQUIRY_EMAIL 537
Payments (Adyen / Routable / checkout)	TR_ADYEN_ENABLED 519, TR_ADYEN_ENVIRONMENT 520, TR_ADYEN_MERCHANT_ACCOUNT 521, TR_ADYEN_CHECKOUT_API_VERSION 522, TR_ADYEN_WEB_VERSION 523, TR_ADYEN_CARD_FEE_BASIS_POINTS 525, TR_ADYEN_CARD_FEE_FIXED_CENTS 526, TR_ROUTABLE_ENABLED 530, TR_ROUTABLE_API_BASE_URL 531, TR_CHECKOUT_CARD_FEE_MINIMUM_CENTS 532
Ops alerting / logs	TR_OPS_CHAT_WEBHOOK_URLS 533, TR_AXIOM_DATASET 541, TR_AXIOM_URL 542
Settle outbox (money path)	TR_SETTLE_OUTBOX_ENABLED 550, TR_SETTLE_OUTBOX_FAST_DRAIN_ENABLED 551, TR_SETTLE_OUTBOX_POLL_INTERVAL_SECONDS 552, TR_SETTLE_OUTBOX_HEALTH_PUBLISH_INTERVAL_SECONDS 553, TR_SETTLE_OUTBOX_CLAIM_BATCH 554, TR_SETTLE_OUTBOX_WORKER_CONCURRENCY 555, TR_SETTLE_OUTBOX_LEASE_SECONDS 556, TR_SETTLE_OUTBOX_PASS_BUDGET_SECONDS 557, TR_REAP_SNAPSHOT_BOOKING_ENABLED 562
Analytics / ClickHouse	TR_ANALYTICS_OUTBOX_ENABLED 567, TR_OPERATIONAL_ANALYTICS_OUTBOX_ENABLED 568, TR_CLIENT_EVENTS_ENABLED 578, TR_PROVIDER_ANALYTICS_CLICKHOUSE_URL 582, TR_PROVIDER_ANALYTICS_CLICKHOUSE_USER 583, TR_PROVIDER_ANALYTICS_CLICKHOUSE_DATABASE 584, TR_PROVIDER_ANALYTICS_CLICKHOUSE_TABLE 585, TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_URL 586, TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_USER 587, TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_DATABASE 588, TR_OPERATIONAL_ANALYTICS_SINK 602, TR_OPERATIONAL_ANALYTICS_CLICKHOUSE_WRITE_USER 603
Stage D / fastpath pilots	TR_STAGE_D_HEARTBEAT_ENABLED 609, TR_STAGE_D_ELIGIBILITY_ENABLED 619, TR_STAGE_D_PILOT_WORKSPACE_IDS 626, TR_FASTPATH_PILOT_WORKSPACES 631, TR_FASTPATH_PILOT_ENABLED 632
Async settlement (shadow v2; enable=false)	TR_ASYNC_SETTLE_SHADOW_ATTACHMENT_BUCKET 659, TR_ASYNC_SETTLE_SHADOW_POLICY 660, TR_ASYNC_SETTLE_SHADOW_HEALTH_REGION 661, TR_ASYNC_SETTLE_SHADOW_WORKSPACES 662, TR_ASYNC_SETTLE_ENABLED 663, TR_ASYNC_SETTLE_PROTECTION 664, TR_ASYNC_SETTLE_TICKET_PRIVATE_KEY_FILE 671, TR_ASYNC_SETTLE_TICKET_KID 672, TR_ASYNC_SETTLE_TICKET_ISSUER 673, TR_ASYNC_SETTLE_TICKET_AUDIENCE 674, TR_ASYNC_SETTLE_AUTHORITY_EPOCH 675

Secrets (SECRET_ENVS , names only) at scripts/deploy/rollout.sh:101-112: TR_SENTRY_DSN , TR_STRIPE_SECRET_KEY , TR_STRIPE_WEBHOOK_SECRET , TR_INTERNAL_GATEWAY_TOKEN , plus the async-settle Ed25519 key mounted as a **file** at /secrets/async-settle/ticket.pem . The deploy fails if it is missing. Optional secrets are appended conditionally (:113-135).

Sizing knobs come from _lib.sh rather than ENV_VARS: TR_CONTROL_PLANE_REGIONS , TR_WARM_REGIONS , TR_CLOUD_RUN_MIN_INSTANCES_BY_REGION , TR_CLOUD_RUN_CONCURRENCY , TR_CLOUD_RUN_MAX_INSTANCES , TR_CLOUD_RUN_MEMORY , TR_CLOUD_RUN_INGRESS (scripts/deploy/_lib.sh:30-93).

Deploy-control vars (read by scripts, not by the app): TR_DEPLOY_TARGET_REGIONS , TR_DEPLOY_NO_TRAFFIC , TR_DEPLOY_RECONCILE_LB (deploy.yml:555-561); TR_DEPLOY_HOLD_REGIONS (repo variable, deploy.yml:482 , :916); TR_DEPLOY_MUTEX_* (scripts/deploy/deploy_mutex.sh); TR_CLOUD_BAKE_HOURS , TR_CLOUD_DEPLOY_MODE , TR_CLOUD_BAKE_OVERRIDE (docs/runbook.md:425-470); TR_STAGED_WATCHDOG_SKIP_BASELINE , TR_BILLING_5XX_GATE_SKIP (hotfix only, deploy.yml:722 , :770).

5. Rollout control-plane mechanics

- **Fail-closed entry.** `rollout.sh` refuses to run unless `TR_ALLOW_DEPLOYED_COMBINED_SURFACE=true` is set, which is a temporary bridge until the six-service cutover (#712) (`scripts/deploy/rollout.sh:7-16`). It also refuses unless the exact 40-char source revision matches the checkout and the release (`:21-31`).
- **Staged traffic.** `staged_traffic.sh` ramps 10, 50 and 100% with a synthetic watch between steps. If a watch trips, traffic goes back to 100% on the old revision (`scripts/deploy/staged_traffic.sh:2-12`). The watchdog is `scripts/deploy/watchdog.py` (per-region post-deploy). The default SLO class is `router_core` (`staged_traffic.sh:18`).
- **Billing gate.** `assert_no_billing_5xx.sh` fails the rollout if the candidate revision logged any billing-path 5xx, because Cloud Run readiness cannot see exhausted retries (`scripts/deploy/assert_no_billing_5xx.sh:2-3`).
- **Deploy mutex / two-cloud coordinator.**
- `deploy_mutex.sh` wraps `cloud_rollout.py`, a generation-CAS journal at `gs://tr-deploy-mutex-quill-cloud-proxy/locks/trusted-router-production.json` covering gcp/aws/azure (`scripts/deploy/cloud_rollout.py:1-41`).
- It is fleet-wide and lets one cloud roll out at a time. The enclave gateway rollouts in quill-cloud-proxy share it (`docs/runbook.md:349-357`).
- Inspect it with `deploy_mutex.sh status`. Manual removal is break-glass only, with `--if-generation-match` (`docs/runbook.md:368-385`).
- **Conflict to resolve:** the runbook says "locks expire after 90 minutes" (`docs/runbook.md:386-387`), but `cloud_rollout.py:5` says "Expired/failed reservations are NEVER reclaimed just because time elapsed."
- **Regional hold.** `TR_DEPLOY_HOLD_REGIONS` (comma list or `all`) is a GitHub repo variable. Held regions still get a no-traffic warm, but their traffic is not touched. It is read at job start, so changing it does not affect a running job (`scripts/deploy/_deploy_hold.sh`; `docs/runbook.md:398-421`).
- **Bake ladder / lifeboat** (applies to AWS and Azure):
- When one cloud gets fresh code, another must keep serving a commit at least 24h old, the "lifeboat" (`docs/runbook.md:425-447`).
- The candidate may reach a second cloud only after 24h on first-parent main. Both paths require `status.json overall_status=up`.
- GCP auto-deploys and can never be the lifeboat.
- Canary mode: `TR_CLOUD_DEPLOY_MODE=canary`. The header of `scripts/deploy/cloud_bake_gate.sh` notes that the default rolling mode now uses the atomic two-cloud guard, and the ladder applies in promote/canary mode.
- Override with a recorded reason using the `bake_override_reason` input (`deploy-aws-control-plane.yml:20-24`).
- **Completeness gate.** "Deployed" and "complete" are different states. `verify_cloud_complete.sh {aws|azure|gcp}` is the executable definition of done (`scripts/deploy/verify_cloud_complete.sh:1-30`). It exists because AWS-EU served for 15 days with no analytics drain.
- **Break-glass.** `scripts/deploy/break_glass_deploy.sh [--apply]` ships from a laptop when GitHub Actions is down. It runs `rollout.sh` and a smoke check. It has **no CI gate, no migrations, no staged canary or auto-rollback, and no secret sync or jobs**, all listed in its header (`:1-37`, `:111-136`). See `docs/runbook-github-outage.md`.
- **Image and attestation digest.**
- The control-plane image is tagged with the commit SHA (`deploy.yml:123-127`).
- `rollout.sh` reads the **enclave's** `gcp-release.json` (local `TRUST_FILE` or URL), extracts `source_commit`, `image_reference` and `image_digest` (`scripts/deploy/rollout.sh:85-98`), and injects them as `TR_TRUST_GCP_*` (`:482-486`).
- AWS and Azure measurements are no longer pinned here. Each plane publishes its own record, and the control plane mirrors it by URL (`:487-495`).
- Note: the `TRUST_FILE` default is an operator's laptop path (`scripts/deploy/_lib.sh:126`). In CI the URL path is used.

- `trust-drift.yml` checks published values against live attestations daily and after each deploy.
- **Dockerfile.** `uv:python3.12-bookworm-slim`. Dependency layers sit before the source copy so the build cache works. Bytecode is precompiled because cold starts measured p50 34.6s / p95 217s without it (`Dockerfile:19-29`). Entrypoint `scripts/entrypoint.sh` runs `python -m trusted_router.serve`, whose listener stays open briefly after SIGTERM to avoid zero-latency 500s (`Dockerfile:31-40`).

6. Terraform (`infra/`)

There is one root, with state in `gs://tr-infra-tfstate-quill-cloud-proxy/infra` (`infra/backend.tf`). It owns only static plumbing. Release procedures stay in `scripts/deploy/` (`infra/README.md:1-36`).

FILE	MANAGES
<code>cloud_armor.tf</code>	4 Cloud Armor policies, imported to mirror live state. tr-deploy cannot edit rules, so rule changes need an owner apply before merge (:1-25).
<code>control_lb_certificates.tf</code> , <code>control_lb_certificate_map.tf</code> , <code>control_lb_proxy.tf</code>	Certificate Manager certs, DNS authorizations (Cloud DNS + Route53 <code>_acme-challenge</code>), cert map, HTTPS target proxy.
<code>spanner_trusted_router.tf</code>	Spanner instance and autoscaler, plus a narrow custom role for capacity changes.
<code>gcp_wif.tf</code>	GitHub WIF provider allowlist. Its "adoption safety check" says the first plan must show no changes.
<code>fastpath.tf</code> , <code>fastpath_alerts.tf</code> , <code>fastpath_deploy_roles.tf</code> , <code>fastpath_node_startup.sh.tftpl</code>	Pub/Sub settle-log/records topics, auditor/stager/archive subscriptions, archive and release buckets, node SA, envelope-key secret, Spanner DB IAM, 2 GCE nodes, alert policies.
<code>logging_sampling.tf</code>	Log exclusions that sample successful Cloud Run requests and enclave lifecycle logs.
<code>aws_deploy_role.tf</code>	GitHub OIDC deploy role for AWS.
<code>aws_synthetic_monitoring.tf</code>	EventBridge <code>tr-eu-synthetic-1min</code> , DLQ, alarms.
<code>aws_security_alarms.tf</code> , <code>aws_config.tf</code>	CIS root-usage and unauthorized-API alarms, AWS Config recorder.
<code>ses_notices.tf</code>	SESV2 notices identity, config set, DKIM/SPF/DMARC records.

The key hazard: **every infra apply re-asserts the whole root**. On 2026-09-21 a one-line WIF change also reverted a hand edit and re-enabled a DSQL read path costing about \$500/day for 27 minutes (`infra/README.md:38-63`).

7. `fastpath/` and `release/`

- `fastpath/` is the Go service for the fast-admission / batched-settlement design. Each `internal/*` package is a "shadow" of a TLA+ spec in `proofs/`, and tests check it against the TLC state graph. It runs over Spanner and Pub/Sub. Its schema is applied by `scripts/deploy/migrate_fastpath.sh` (`fastpath/README.md:1-50`).
- Production state is "P0 dark install": 2 nodes, empty allow-list, nothing admits until a workspace is switched on (`infra/fastpath.tf:1-12`).
- Release pipeline: CI green, then `fastpath-release.yml`, then GCS. Terraform pins a commit (`infra/fastpath.tf:19`), and a node picks it up when reset or replaced.
- Runbooks: `docs/runbooks/fastpath-*.md`.
- `release/` holds `first-window-plan.json` and its schema. It is a reviewed plan for the async-settle shadow-v2 "first window": router and enclave SHAs/image digests, workspace, attachment bucket, per-region caps, surge and publishers. The checked-in copy is a template with `REVIEW` placeholders and zero SHAs. Its consumers are

docs/design/async-settle-shadow-v2-activation.md:44, :99 and docs/design/async-settle-shadow-v2.md:3697.

8. Observability

- **Sentry:** `src/trusted_router/sentry_config.py`.
- `send_default_pii=False` (:240). `before_send`, `before_send_log` and `before_breadcrumb` all run `_scrub` (:301-331).
- Key-name denylist `SENSITIVE_KEYS` (:21), substring and prefix scrubbing (:43-95), and bytes redacted wholesale (:586).
- A floodgate rate-limits event floods (:453-500). A noise filter drops untrusted 405s (:334-370).
- The DSN is a secret (`TR_SENTRY_DSN`).
- **GCP alert policies (YAML, applied by scripts):**
- `scripts/deploy/spanner-alerts/`: `api-failures`, `backup-workflow`, `capacity`, `contention`, `high-priority-cpu` (45%, `high-priority-cpu.yaml:11-19`), `latency`.
- `scripts/deploy/gateway-alerts/`: `billing-5xx` ("any internal gateway billing 5xx"), `billing-pressure`, `client-invisible-outage`, `client-telemetry-stale`, `control-plane-memory`.
- `scripts/deploy/clickhouse-alerts/`: `disk-capacity`, `node-availability`.
- Fastpath alerts are in Terraform (`infra/fastpath_alerts.tf`).
- **Paging policy:** router-core 5m/1h burn rate $\geq 14.4\times$ pages immediately; 6h $\geq 6\times$ pages in waking hours; 24h $\geq 3\times$ creates a review item (`docs/runbook.md:105-110`).
- **Synthetic monitoring:**
- GCP: Cloud Run Jobs deployed by `scripts/deploy/synthetic.sh` (uptime probes kept separate from the throughput benchmark, :1-5), redeployed every rollout (`deploy.yml:972`).
- AWS: EventBridge 1-min rule to App Runner `tr-eu /internal/synthetic/run` (`HANDOFF.md:110-117`; `infra/aws_synthetic_monitoring.tf`).
- **Status:** `trustedrouter.com/status.json` (per-cloud status, used by bake gates and freshness checks). Public page `status.trustedrouter.com` is linked from `src/trusted_router/dashboard.py:4620` and allowed in CSP at `middleware.py:458`. The repo does not say who hosts the status page. Trust records are published at `trust.trustedrouter.com`.
- **Logs:** Axiom (`TR_AXIOM_*`), plus Cloud Logging with sampled exclusions (`infra/logging_sampling.tf`).
- **Scheduled health checks:** prod-smoke every 30 min, trust-drift, analytics/archive/telemetry freshness, cloud-staleness, typed-audit, provider-catalog-health (section 3).

9. Runbook and incident index

`docs/runbook.md` (symptom → action; 1638 lines)

ANCHOR (LINE)	SUMMARY
router-core-page (83)	Router-core four-nines burn-rate page: triage and thresholds
region-drain (112)	Drain one gateway region via Cloudflare DNS-only LB, then verify SDK failover
storage-degraded (135)	Spanner or ClickHouse degraded
provider-502 /-429 /-401 (182/213/232)	Upstream provider error classes
gateway-auth-400 (268)	"gateway authorization failed" on a provider smoke

ANCHOR (LINE)	SUMMARY
eu-rollback, min-ready (300/317)	GCP enclave deploy rollback loops / gcloud flag issue
bot-doesnt-deploy (328)	Price bot commits but the catalog stays stale
deployment-mutex (349)	Fleet-wide GCS mutex, status, break-glass delete, regional hold
cloud-bake-ladder (425)	24h lifeboat rule, canary mode
stale-status (500)	Status page shows a region down when it is healthy
missing-model / new-provider / new-model (551/573/606)	Catalog operations
rotate-key, phala-revive (620/632)	Provider key rotation
azure-deploy (681)	Deploying the Azure control plane
stop-remediator (723)	<code>TR_REMEDIATOR_MODE=off</code> instead of editing Terraform-owned targets
settle-outbox (808)	Flip, verify, monitor, roll back the settle outbox
credit-ledger (1051)	Typed credit-ledger operations
authorize-deadlock-burst (1141)	Sentry deadlock/wounded bursts on authorize
reshard-interrupted (1210)	Workspace 503 "billing is paused" after an interrupted reshard
dns-vendor-split (1324)	Cloudflare vs Cloud DNS dual-authoritative NS symptoms
adding-a-cloud (1441)	When a new cloud may be called done
useful-one-liners (1549), billing-pause-gate (1602), route-health-postgres (1615)	Misc

Other runbooks

FILE	SUMMARY
docs/runbook-github-outage.md	GitHub down vs repo gone: different answers. Points to the mirror and break-glass.
docs/runbooks/ddos-edge-hardening.md	Deployment contract for public edge protection and emergency controls
docs/runbooks/gateway-billing-edge.md	<code>/internal/gateway*</code> routing through Iowa and São Paulo
docs/runbooks/fastpath-p0.md / -p1.md	Fast path dark install; synthetic load on one internal workspace
docs/runbooks/fastpath-deploy.md	Deploying fastpath nodes without losing held leases
docs/runbooks/fastpath-watch.md / -alerts.md	Stage watch with baseline; journal-log alerts
docs/runbooks/fastpath-turn-off.md	Disable for a workspace, or stop the fleet
docs/runbooks/fastpath-key-rotation.md	Rotate the envelope sealing key
docs/runbooks/ses-notices.md / ses-reputation.md	Notices sender authorization; SES reputation response (transactional only)
docs/runbooks/identity-verification-support.md	Failed Veriff checks (\$5 per attempt)
docs/speculation-shadow-runbook.md	Speculative-invocation shadow (ships disabled)
docs/spanner-reliability.md, docs/clickhouse-reliability.md	Canonical storage reliability runbooks
docs/provider-independent-deploys.md	Provider health is separate from release gating
docs/operations/service-surfaces.md, public-surface-split.md, internal-surface-split.md	Service-surface bulkheads (combined, public, internal)
docs/operations/allyrouter-alias.md, uptimerouter-alias.md	Independent first-party domain aliases for resilience

FILE	ONE-LINE SUMMARY
2026-09-05-shared-trace-settlement	Stage D made <code>gateway_request_id</code> unique; multi-call requests sharing a trace blocked settlement
2026-09-06-spanner-operator-search-cpu	Three ad hoc wildcard Stripe-ID searches of <code>tr_entities</code> spiked Spanner CPU (origin of the AGENTS.md query rules)
2026-09-08-legacy-billing-shard-contention	A workspace resharded from 1 to 16 hit legacy billing shard lock contention
2026-09-24-spanner-overflow-rollup-scan	Hourly reservation-overflow rollup scan tripped the HP CPU alert
2026-09-24-us-east4-cloud-run-start-stall	us-east4 instances failed to start for 16 min with no deploy in progress
2026-09-25-credit-rebalance-convoy	Starter-credit rebalance convoy caused billing-path 5xx alerts
2026-09-26-regional-ledger-grant-storm	Regional ledger slowdown amplified into a Spanner grant storm
2026-09-30-provider-service-tier-settlement	Settle 500s: OpenAI service tier on a non-OpenAI endpoint
2026-10-03-provider-configuration	New config alerts exposed failing routes that had been excluded from uptime
2026-10-04-gateway-authorization	EU authorize 503s from <code>authorize_strict_budget_busy</code>
2026-10-04-starter-credit-fragmentation	New workspace authorize 503s from fragmented starter credit
2026-10-05-provider-recovery	Adapter parsed only 3 models; retirements blocked discovery of replacements
2026-10-10-shadow-* (4 files) + spanner-shadow-budget	Async-settle shadow: regional read deadlines, SDK retry integration, shutdown-write failure, short release label rejected, txn budget alerts
2026-10-11-oauth-invalid-code	Invalid Google OAuth codes returned 500 instead of 4xx
2026-10-11-shadow-budget-followup	Shadow audit after the regional release reached 100%
2026-10-11-spanner-commit-latency-minute	One-minute 7x commit latency on nam6 paged API failures
2026-10-13-shadow-missing-health-alerts	Unqualified shadow observer caused Spanner API-failure alerts
2026-10-15-gcp-billing-alerts	GCP billing / CIS route-change alerts investigation
2026-10-18-cloud-alerts	Commit-p99 >1s alerts, recovered events vs verified repairs
2026-10-18-fastpath-lease-locks	Fast-path renewal vs auditor commit contention drove commit p99
2026-10-18-settlelog-shutdown	Fastpath settle-log subscription shutdown hang timed out the Go CI gate

10. Deploy safety rules

- Run all three CI gates locally:** `uv run ruff check . && uv run mypy && uv run pytest -q`. Never run `ruff format .`. Coverage must stay $\geq 70\%$ (AGENTS.md:27-40). Run the full suite, because tests pollute each other through the `STORE proxy` (:42-45).
- Rebase before trusting a failure.** The price bot advances main hourly (AGENTS.md:47-49).
- Pushing to main deploys** if a path in the filter changed (AGENTS.md:60-62 ; `deploy.yml`:26-37). Docs-only changes do not deploy.
- Change config only in `ENV_VARS`.** Hand edits to Cloud Run are overwritten (AGENTS.md:64-66).

5. **Never call a queued or partial rollout "deployed"**. Verify behaviour and prod health after deploy, and report the exact blocker if a gate stops it (`AGENTS.md:51-58`).
6. **Keep the gates intact**: CI-green gate, mutex, staged ramp, canary, billing 5xx gate, completeness check. `hotfix=true` is for when prod is already sick (`deploy.yml:38-43`).
7. **One cloud at a time**. Respect the GCS mutex. Delete it only after `status` shows a dead owner, and use `--if-generation-match` (`docs/runbook.md:349-387`).
8. **Use** `TR_DEPLOY_HOLD_REGIONS` to freeze a region during an incident. It is read at job start (`docs/runbook.md:398-421`).
9. **Bake ladder**: fresh code must not serve on every cloud at once. Keep a ≥ 24 h lifeboat, and record any override reason (`docs/runbook.md:425-447`).
10. **Never run large unbatched DML on prod Spanner during a rolling deploy** (`AGENTS.md:68`).
11. **Read-only is not load-safe**. No `LIKE`, regex or JSON scans of `tr_entities.body`. Use `python -m scripts.inspect_payment_owner`, `SPANNER_SYS` stats and ClickHouse. Never loosen the 45% CPU alarm (`AGENTS.md:70-79`).
12. **Money code** (reserve/settle/refund) needs differential or conformance tests. Run `proof_oracle` when touching it (`AGENTS.md:38-40`, `:81-87`).
13. **Terraform has a single owner**. An infra merge re-applies the whole root. Mirror any emergency hand edit in a PR within the hour, and check the apply log for unexpected resources (`infra/README.md:38-63`).
14. **Cloud Armor rule changes** need an owner's credentials before merge, because tr-deploy cannot weaken the edge (`infra/cloud_armor.tf:9-14`).
15. **Do not enable Cloudflare proxying on the prompt path**. It voids attestation (`docs/runbook.md:118-119`; `HANDOFF.md:100`).
16. **Break-glass skips everything except rollout and smoke**. Write down the prior revision and run migrations by hand (`scripts/deploy/break_glass_deploy.sh:13-30`).
17. **A PCR0 re-pin on AWS touches three surfaces**: both Fargate task definitions and App Runner `tr-eu` (`HANDOFF.md:119-123`).
18. **"Finished" is not "complete"**. Bring-up scripts must end in `verify_cloud_complete.sh` (`scripts/deploy/verify_cloud_complete.sh:8-25`).

05

PART V

Ecosystem

1. Platform repositories
2. SDKs, apps & benchmarks

11 The Platform Repos

3 Reproducible builds + attestation



Each release builds a measured image whose hash identifies exactly what will run.

PART V · CHAPTER 11 · HERO STILL: VIDEO 11 AT 0:53

Platform repositories

quill-cloud-proxy in depth, plus the infrastructure and provider repos in Lore-Hex.

- 1. Summary table
- 2. Deep dive: quill-cloud-proxy
- 3. Per-repo sections
- 4. Ecosystem diagram
- 5. Onboarding notes / risks

Storyboard · Video 11: The platform repos

0:23

0:28

0:34

0:53

1:09

1:32

1:40

1:47

Research snapshot: 2026-10-19. Sources: `gh api repos/Lore-Hex/<repo>` (metadata, commits, contributors), shallow clones (`git clone --depth 1`) read without building or running anything, and cross-references in `quill-router` (`src/trusted_router/routes/internal/gateway.py`, `README.md`, `tests/test_provider_check_contract_parity.py`).

"Size" is the GitHub repo size (KB, includes history/assets). LOC counts are raw line counts per language in the shallow clone, including tests and excluding `third_party/` and `node_modules/`.

1. Summary table

REPO	PURPOSE (ONE LINE)	STACK	SIZE / LOC	LAST COMMIT ON MAIN	LICENSE	ACTIVITY / MATURITY	RELATION TO QUILL-ROUTER
quill-cloud-proxy	Attested data-plane gateway (<code>api.trustedrouter.com</code>) on GCP Confidential Space, AWS Nitro Enclaves and Azure confidential ACI	Go 1.26 (enclave), Python (AWS parent), Bash/Python tools	58.5 MB / Go 189k (87k non-test, 452 test files), Py 32k	2026-10-19 (daily)	BUSL-1.1 (Apache-2.0 for code before 2026-07-03; each version becomes Apache-2.0 after 4 yrs)	Very high. jperla 446 commits, github-actions bot 1108 (trust-artifact signing)	Hard runtime dependency. Calls control plane <code>/v1/internal/gateway/{validate,authorize,settle,refund,...}</code> with <code>x-trustedrouter-internal-token</code> . Shares billing contract (frozen billing v1 fixtures)
quill-cloud-infra	Terraform for the proxy's AWS, GCP, Azure infrastructure	HCL	122 KB / 4k TF	2026-08-25 (branches pushed 2026-10-15)	BUSL-1.1	Low cadence, single author (jperla, 4 contributions as counted by GitHub)	Provisions gateway infra. Spanner instance adoption (shared with control plane) lives in <code>envs/gcp-prod</code>
truste-drouter-confidential-stack	Supply-side kit: Helm/Kustomize plus <code>trcs</code> attested TLS shim so providers can serve OpenAI-compatible endpoints from their own GPUs with hardware evidence	Go (stdlib only), Helm, Kustomize	189 KB / Go 4.8k	2026-10-19	Apache-2.0	New (created 2026-09-20), alpha, active PRs	Defines <code>tr-attestation/v0</code> . The relying-party verifier is expected in the TR gateway. Providers then apply on the marketplace
token-exchange-neocloud-in-a-box	Reference "confidential GPU cloud": TDX + NVIDIA CC 8xB200 nodes (DeepSeek-V4.1-Flash, GLM-5.3), first operator Hixtube	Go 1.1k, Python 4.9k	700 KB	2026-10-19	Apache-2.0	Pre-alpha, created 2026-10-19, 19 commits on day one	Builds on confidential-stack. Must pass provider-check. Nodes get listed on the TR marketplace / NY Token Exchange
truste-drouter-onpremise	Self-hosted model gateway (fork of Experiential) with TR as its only first-party hosted backend	Python (huge, 293k LOC), uv, <code>justfile</code>	6.1 MB	2026-10-03	Apache-2.0	Moderate. Mostly upstream sync (SilenNaihin 128, devin bot 20, jperla 11)	Client of public <code>https://api.trustedrouter.com/v1</code> using <code>sk-tr-v1-</code> keys
truste-drouter-provider-check	CLI that checks an OpenAI-compatible provider endpoint against TR's gateway contract (Tiers 1-6)	Python, uv	313 KB / 11.8k	2026-10-11	Apache-2.0	Steady sync PRs (jperla 13)	Vendors contract symbols byte-for-byte from quill-router. <code>quill-router</code> <code>tests/test_provider_check_contract_parity.py</code> + <code>scripts/export_provider_check_contract.py</code> enforce parity

REPO	PURPOSE (ONE LINE)	STACK	SIZE / LOC	LAST COMMIT ON MAIN	LICENSE	ACTIVITY / MATURITY	RELATION TO QUILL-ROUTER
lightning-router	Mirror of LightningRouter (BTC Lightning to TR USD credits) app, frontend, LND tooling	Python 9.7k	1.1 MB	2026-10-14 (bot mirror)	BUSL-1.1	Read-only mirror. PRs disabled. Published by hourly bot from quill-router CI	Source of truth is inside quill-router (<code>routes/internal/lightning.py</code> , <code>services/lightning.py</code> , <code>storage_lightning.py</code>). <code>SOURCE.json</code> pins the upstream commit
reverse-harnesses	<code>npx reverse-harness:</code> expose a local model, agent, or human as a paid TR "user-provided model"	TypeScript/Node	91 KB / 4k	2026-08-17 (pushed 2026-10-19)	Apache-2.0	Small, v0.2.0	Implements <code>CONTRACT.md</code> : owner API <code>POST /v1/user-models</code> , <code>clock-in/heartbeat/clock-out</code> . Attested gateway pushes signed requests to it. Owner keeps 70% in credits
TrustedRouter-Fusion-Draco	DRACO deep-research benchmark reproduction showing TR fusion (panel + MiniMax-M3 fuser = 71.6)	Python 10k	35 MB (data)	2026-08-16	Apache-2.0	Dormant research artifact	Client of TR API via <code>trusted-router-py</code> SDK and <code>trustedrouter/fusion</code>
SREChat	"RoachChat": multi-master, partition-tolerant BEAM chat (one master per cloud), with SRE agents on each cloud	Elixir 31k, Python 8k (agent), TS	4.7 MB	2026-09-20	AGPL-3.0	Moderate (jperla 27)	Ops tool. Its SRE agents call <code>api.trustedrouter.com/v1</code> (<code>trustedrouter/auto</code> fallback) and watch the 3-cloud deployment
OpenChat	AGPL CometChat-compatible drop-in chat server	Elixir 26k, TS	872 KB	2026-08-16	AGPL-3.0	Stable/dormant	No direct TR coupling. Ancestor of SREChat
AllRouter	Local + cloud LLM router (OpenAI + Anthropic APIs), "BackupRouter" preset for Claude Code / Codex via TR	Go 17k	290 KB	2026-08-16	Elastic-2.0	Dormant since Aug (jperla 9). Homebrew tap releases	Client of TR public API (<code>TRUSTEDROUTER_API_KEY</code>). Bursts and fallbacks go through TR
Bursty Router	Local-first proxy that bursts overflow to TR	Go 15k	275 KB	2026-08-16	Elastic-2.0	Dormant (jperla 8)	Client of TR public API. Sibling/predecessor of AllRouter
trustedrouter-eu	GitHub Pages redirect <code>eu.trustedrouter.com</code> to <code>trustedrouter.com/eu</code>	HTML	5 KB	2026-06-11	none	Done	Marketing redirect only

Main takeaways:

- quill-cloud-proxy is the only repo here on the production request path besides quill-router itself. The other repos are SDK-level clients (onprem, AllRouter, BurstyRouter, Fusion-Draco, SREChat agents), supply-side tooling (confidential-stack, neocloud, provider-check, reverse-harness), mirrors (lightning-router), or unrelated/dormant (OpenChat, trustedrouter-eu).
- Almost everything is written by one engineer, Joseph Perla (`jperla`). Bus factor is effectively 1 across the whole platform.
- License mix: BUSL-1.1 for the moat (proxy, infra, lightning), Elastic-2.0 for local routers, Apache-2.0 for ecosystem/supply-side, AGPL for chat.

2. Deep dive: quill-cloud-proxy

2.1 What it is

The prompt-handling data plane of TrustedRouter. It is the code behind `api.trustedrouter.com` (alias `api.quillrouter.com`). quill-router is the **control plane** (keys, billing, catalog, routing decisions, Spanner). It never sees prompts. quill-cloud-proxy is the **attested gateway**. Its stated rule: "The signed workload image is the boundary." The control-plane client package doc says it "sends API-key lookup hashes, model/routing preferences, and token counts; it never sends prompt or completion text."

Created 2026-04-28. Pushed several times per day. CI badges: `ci.yml`, `deploy-enclave-gcp.yml`, `publish-trust-page.yml`.

2.2 Layout

PATH	CONTENTS
<code>enclave-go/cmd/enclave/</code>	The workload binary (~218 files). HTTP server, public routes, per-cloud listeners (<code>listener_{aws,gcp,azure}.go</code>), billing backoff, fusion, batch, images, embeddings, speech, video, Anthropic <code>/v1/messages</code> , Responses API
<code>enclave-go/internal/trustedrouter/</code>	Control-plane client: authorize/settle/refund, retries, async settlement (shadow v2), spend lease, Stage C/D, endpoint failover rules
<code>enclave-go/internal/llm/</code>	Provider adapters: Anthropic, Vertex/Gemini, AI Studio, Bedrock, Azure, Cohere, Databricks, Kimi, MiniMax, Chutes E2EE, Fortanix, Cloudflare gateway, multi-dispatcher (<code>multi.go</code>), egress guard
<code>enclave-go/internal/attestation/</code>	<code>attestation_{aws,gcp,azure}.go</code> : Nitro NSM docs, Confidential Space JWT, Azure MAA
<code>enclave-go/internal/enclavetls/</code>	In-enclave TLS termination (ACME certs; the key never leaves the workload)
<code>enclave-go/internal/bootstrap/</code>	Per-cloud secret bootstrap: GCP Secret Manager gated by attestation, AWS via parent vsock + KMS PCRO, Azure SKR
<code>enclave-go/internal/byokcache/</code>	BYOK envelope decrypt (Cloud KMS-unwrapped DEK + AES-256-GCM), memory-only TTL cache
<code>enclave-go/internal/billingv1/</code>	Frozen billing snapshot v1 contract, mirroring the Python router. 759 shared literal fixtures, SHA-pinned
<code>enclave-go/internal/{spendlease,authcache,receipt,websearch,webfetch,privateMode,decide,speculation,shadow*}</code>	Spend-lease JWS admission, auth caching, signed inference receipts, Exa web search inside the enclave, PrivateMode, <code>/v1/decide</code> routing, speculative/shadow experiments
<code>enclave-go/sidecar/</code>	Separate Go module: <code>attest-sidecar</code> verifies third-party confidential providers (Tinfoil SEV-SNP, NEAR AI / Intel TDX, Chutes) over a Unix socket. It is a separate module because sigstore/go-tuf dependencies broke the main binary's vsock/TLS loop when linked in
<code>enclave-go/Dockerfile.enclave*</code>	Measured images: AWS EIF, <code>gcp.multi</code> (production GCP), <code>gcp.anthropic</code> , <code>azure.multi</code> . Builder images are pinned by digest
<code>parent/</code> (Python, <code>quill_parent</code>)	AWS-only EC2 host process: admin endpoints, raw TCP pump for enclave-terminated TLS, heartbeat, DynamoDB usage aggregates, bootstrap RPC over vsock
<code>tools/</code>	~100 operator scripts: <code>deploy-gcp-mig.sh</code> , <code>deploy-aws-nitro.sh</code> , <code>deploy-azure-aci.sh</code> , <code>release-gcp.sh</code> , <code>verify-pcr0.sh</code> , <code>capture-plane-measurements.py</code> , <code>cloud-rollout.py</code> , DNS reconcilers, secret seal/revoke
<code>trust-page/</code>	Static site published to <code>trust.trustedrouter.com</code> (GitHub Pages) and mirrored to <code>trust.quill.lorehex.co</code> (S3/CloudFront). Holds <code>pcr0*.txt</code> , <code>image-digest-gcp.txt</code> , <code>hostdata-azure.txt</code> , <code>*-release.json</code> , accepted sets, and a <code>.bundle</code> signature per record
<code>docs/</code>	<code>design/</code> , <code>runbooks/</code> (incident response, enclave deploy, AWS EU attestation, Azure enclave, provider onboarding), <code>tls-inside-enclave.md</code> , <code>gcp-attestation-issuance.md</code> , <code>browser-api.md</code>

Go build tags choose the cloud and provider set at compile time: `cloud_aws,llm_bedrock`, `cloud_aws,llm_multi`, `cloud_gcp,llm_vertex`, `cloud_gcp,llm_multi`, `cloud_azure,llm_multi`. Direct dependencies are kept small: AWS SDK (Bedrock), `hf/nsm`, `mdlayher/vsock`, `x/crypto`, `x/net`.

2.3 Public surface (served inside the enclave)

`/v1/chat/completions`, `/v1/responses` (+ `/input_tokens`, `/compact`), `/v1/messages` (Anthropic), `/v1/embeddings`, `/v1/images/*`, `/v1/audio/{speech,transcriptions}`, `/v1/videos`, `/v1/batches`, `/v1/files`, `/v1/models`, `/v1/key`, `/v1/decide`, `/v1/evaluate`, `/v1/tools/fetch`, and `/attestation?nonce=` (nonce-bound evidence that also binds the workload TLS cert). The request boundary is an allowlist: unknown fields return 400 with `error.param`, and known-but-unsupported OpenRouter controls return 501 `not_supported_in_alpha`. `provider.data_collection` is forced to deny.

2.4 Control-plane contract (enclave to quill-router)

- Base URL: `TR_CONTROL_PLANE_BASE_URL`, or the bootstrap `TrustedRouterBaseURL`. Routes are mounted under `/v1` in quill-router (`routes/internal/gateway.py`).
- Auth: shared secret header `x-trustedrouter-internal-token` (`TR_INTERNAL_GATEWAY_TOKEN`). On GCP it is loaded from Secret Manager as `QUILL_TRUSTEDROUTER_INTERNAL_SECRET`, behind the attestation gate.
- Endpoints the enclave calls, ranked by how often the client code references them:

ENDPOINT	PURPOSE
POST <code>/internal/gateway/authorize</code>	Sends <code>api_key_lookup_hash</code> (never the raw key), <code>model</code> / <code>models</code> / <code>provider</code> prefs, estimated input tokens, max output tokens, region, route type, idempotency key, stream flag, attribution (app, referer, tags, user, session). Returns <code>Authorization: authorization_id</code> , <code>credit_reservation_id</code> , <code>workspace_id</code> , chosen <code>provider</code> / <code>upstream_model</code> / <code>endpoint_id</code> , <code>route_candidates</code> , BYOK envelope + cache key, <code>estimated_cost_microdollars</code> , <code>cap_micro</code> , billing snapshot / settlement ticket, optional <code>spend_lease</code> , broadcast destinations
POST <code>/internal/gateway/settle</code>	Sends actual token usage and cost for the <code>authorization_id</code> . Returns the settle result. Retrieved through a settlement retry queue
POST <code>/internal/gateway/refund</code>	Releases the hold on upstream failure (status, error type, elapsed)
POST <code>/internal/gateway/validate</code>	No-hold key validation (models list, metadata routes, error attribution)
POST <code>/internal/gateway/resolve-custom-model</code> , <code>/key</code> , <code>/heartbeat</code> , <code>/usage-heartbeat</code> , <code>/spend-lease/register-boot</code> , <code>/video/jobs/*</code> , <code>/fetch-image</code> , GET <code>/authorizations/{id}/disposition</code>	Custom/user-provided models, key info, liveness, spend-lease boot registration, async video jobs, settlement disposition

- Money-safety rule (`internal/trustedrouter/endpoints.go`): endpoints are an ordered list, but production configures **one** billing authority. Failover to a later endpoint is allowed only on a tagged **dial** failure, because then no request byte could have reached a server. Any later failure is ambiguous, and re-sending could escrow or bill twice. Settle and refund are pinned to the authority that created the hold.
- 402 `insufficient_credits` backoff (`QUILL_BILLING_402_BACKOFF_MS`, default 5 s): identical denied requests are answered locally for one window. The cache keys on digests only and holds at most 4096 entries.
- Spend lease (`internal/spendlease`): a JWS-signed local-admission budget (TTL $\leq$ 60 s) so the enclave can eventually admit without a synchronous authorize round-trip. Stage A is shadow-only. Stage C is gated behind a separately measured flag.
- Shared contracts with quill-router: `billingv1` consumes the same `testdata/async_settlement/billing_v1.json` fixtures, SHA-pinned, with Python-compatible canonical JSON hashing. Signed receipts follow `docs/design/signed-receipts-wire-format.md`. Production rollouts use a coordinator loaded from a pinned quill-

router commit and SHA (`tools/cloud-rollout-source.json`), under a cross-repo "two-cloud limit": at most two clouds change at once while a third stays healthy.

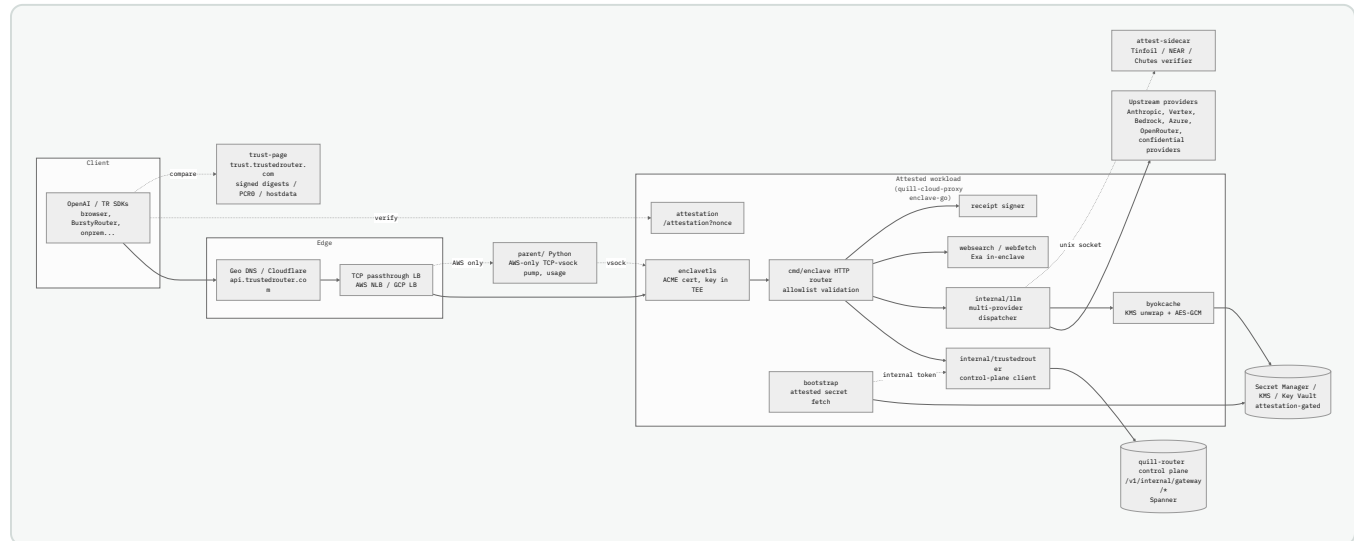
2.5 Per-cloud trust posture

CLOUD	TEE	WHO TERMINATES TLS	HOW SECRETS ARE RELEASED	PUBLISHED MEASUREMENT
GCP (production primary, 4 regions, MIGs)	Confidential Space (single container, no parent)	Enclave (ACME cert, <code>QUILL_ENCLAVE_TLS=true</code> baked into image)	Secret Manager / KMS IAM condition on attested image digest	<code>image-digest-gcp.txt</code> , <code>gcp-release.json</code> , <code>stage-d-accepted.json</code>
AWS (us-east-1, eu-west-3)	Nitro Enclave + EC2 parent	Enclave. Parent is a raw TCP-to-vsock pump (NLB passthrough). Older V1 used ALB termination	KMS key policy on PCR0. Parent fetches the sealed blob and relays it over vsock	<code>pcr0-aws.txt</code> , <code>accepted-pcr0s-aws.txt</code> , <code>aws-release.json</code>
Azure (uaenorth, australiaeast)	Confidential ACI (SEV-SNP)	Enclave	Key Vault Secure Key Release on <code>hostdata</code> via MAA	<code>hostdata-azure.txt</code> , <code>azure-release.json</code>

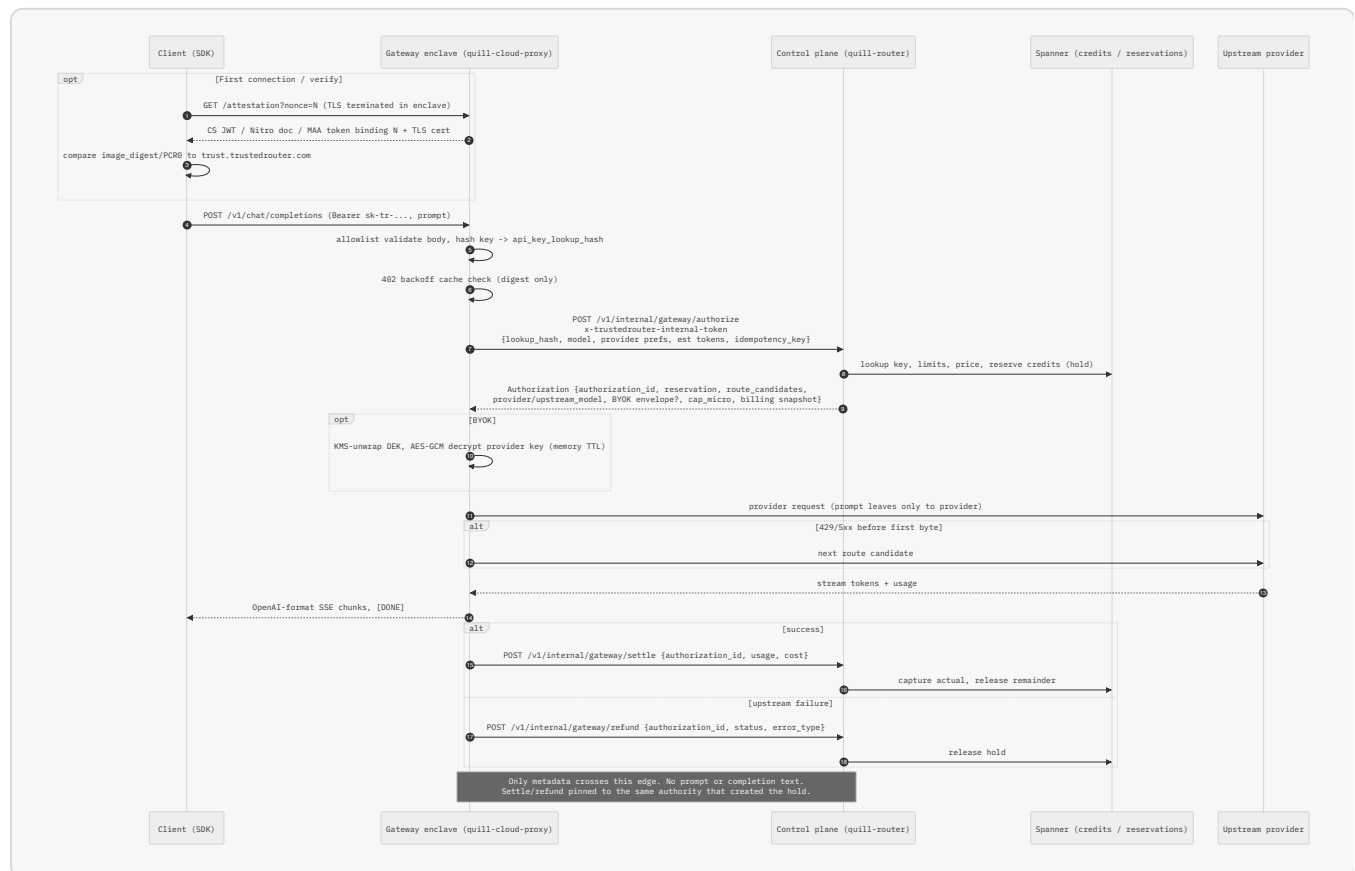
2.6 Build, release, attestation reproducibility

- Local dev: `make sync` (uv), `make check` (ruff, mypy --strict, pytest for the parent; go build/vet/lint/test across all five tag sets; deploy-script guards), `make run-mock` (parent + mock enclave on a Unix socket, `localhost:8443`).
- AWS reproducibility: `tools/verify-pcr0.sh` runs `docker buildx --no-cache --platform linux/amd64` with the exact release build args (`BUILD_TAGS=cloud_aws,llm_multi` , `QUILL_TLS_MODE=acme` , `QUILL_API_HOST=api-aws.trustedrouter.com`) and then `nitro-cli describe-eif` , and compares the result to `trust-page/pcr0.txt` . A test (`tools/test_verify_pcr0_build_args.py`) fails if verify and release args drift.
- GCP release: `make gcp-release` (`tools/release-gcp.sh`) builds `Dockerfile.enclave.gcp.multi` , pushes to `us-central1-docker.pkg.dev/quill-cloud-proxy/quill/enclave-multi:gcp-release-<sha>` , resolves the OCI digest, and writes the trust files. `deploy-enclave-gcp.yml` waits for CI green on the same commit, runs offline Stage D gates, builds in Cloud Build, writes and commits trust artifacts, waits for the signed public trust set, reserves GCP across both repos, then rolls each region in turn: drain DNS, roll MIG, attest healthy, per-instance streaming gate, synthetic canary.
- Signing: each plane's records are signed by `publish-trust-{gcp,aws,azure}.yaml` under that plane's identity, which produces `.bundle` files and explains the ~1100 bot commits. Publishers refuse unsigned or mismatched copies (`check-trust-signatures.py` , `check-trust-copies.py`). AWS and Azure records are captured from live attestations with `capture-plane-measurements.py` .
- Client verification: compare the live Confidential Space JWT `submods.container.image_digest` claim to `image-digest-gcp.txt` . The SDKs, and the "device service" mentioned in the README, check that the cert in the TLS session is the one bound in the nonce'd attestation before sending a prompt. The flow is PCR0/digest → attestation doc → cert fingerprint → TLS session.

2.7 Component diagram



2.8 Request sequence (chat completion, synchronous settlement)



3. Per-repo sections

3.1 quill-cloud-infra

- **Purpose:** Terraform for quill-cloud-proxy. AWS `us-east-1` (`envs/prod`), AWS EU `eu-west-3` (`envs/aws-eu-prod`), Azure `uaenorth` `analytics` (`envs/azure-prod`), Azure TEE scaffolding (`envs/azure-tee`), GCP (`envs/gcp-prod`), four enclave regions, analytics in `us-central1`.
- **Key dirs:** `modules/{network,compute,alb,nlb,kms,ecr,dynamodb,iam,s3,github-oidc,cloudtrail,aws,azure,gcp}`, `envs/*`, `gcp/bringup.sh`.
- **Design rules worth knowing:** each cloud keeps its Terraform state in that cloud, so an AWS outage cannot block Azure fixes. Pre-existing resources are *imported, never created* (each env has `import.sh`). Terraform owns the **static** half of the GCP fleet (MIG shells, service account, firewall) and ignores template versions. The **measured** half (instance templates carrying image digests) and Azure enclave container groups stay in `quill-cloud-proxy/tools` because widening and narrowing attestation pins cannot be expressed in plan/apply. It recently adopted the Spanner instance and owns its autoscaling policy.
- **Run:** a one-time bootstrap creates the state bucket and lock table, then `terraform apply -target=module.github_oidc`, and later applies run from GitHub Actions OIDC.
- **License:** BUSL-1.1. **Activity:** last main commit 2026-08-25.

3.2 trustedrouter-confidential-stack

- **Purpose:** lets GPU owners run a TR-compatible endpoint with TLS terminated inside the protected workload and CPU/GPU evidence bound to the TLS key. Tiers: `standard`, `attested`, `attested+custody`.
- **Key dirs:** `charts/confidential-inference` (vLLM/SQLang + shim, Helm is the source of truth), `kustomize/{base,coco-snp,coco-tdx}` (generated, drift-checked), `cmd/trcs` (`shim`, `preflight`, `check-endpoint`, `lint-manifests`), `internal/{shim,attest,preflight,checkendpoint,manifestlint}`, `deploy/dstack`, `docs/{threat-model,attestation-protocol,architecture,hardware,kubernetes-coco}.md`.
- **Run:** `make all`, `make e2e-kind` (kind cluster with a stub model), `helm install inference charts/confidential-inference -f examples/values-dev-gpu.yaml`.
- **Maturity:** alpha. The README says outright that nothing has run on real CC hardware yet. Relying-party verification is explicitly out of scope; it belongs in the TR gateway (quill-cloud-proxy).

3.3 token-exchange-neocloud-in-a-box

- **Purpose:** an 8-GPU TDX + NVIDIA CC "neocloud" whose nodes are independently verifiable and listed on TR and the NY Token Exchange. The first operator is Hixtube. A local pilot runs Qwen3-0.6B.
- **Key dirs:** `cmd/{nib,nib-catalog,nib-pilot-api,nib-domain-check}`, `internal/{verify,evidence,protocol,catalog,modelpack,pilotapi}`, `guest/`, `host/`, `image/`, `models/{glm-5.3,...}`, `evidence/p0..p3` (committed run evidence), `PLAN.md`, `AGENTS.md`.
- **Run:** `make check && make build && ./build/nib --help`.
- **Relation:** consumes `tr-attestation/v0` from confidential-stack. Pilot passes provider-check (18 checks, Catalog v2). The "TrustedRouter generic verifier" is phase P4, not started. Pre-alpha.

3.4 trustedrouter-onprem

- **Purpose:** self-hosted gateway (OpenAI-compatible local API, per-key budgets, aliases, routing policy, trace ingestion/evals) with TR as its only hosted lane. It never silently falls back to another public router.
- **Stack:** Python package `exp` (kept for upstream sync), CLI `tr-onprem`. Upstream is Experiential (commit `121d8888`).

- **Run:** `pip install git+https://github.com/Lore-Hex/trustedrouter-onprem.git, TRUSTEDROUTER_API_KEY=sk-tr-v1-... tr-onprem login && tr-onprem.Dev: uv sync --extra dev; uv run ruff check .; uv run ty check; uv run pytest -q.`
- **Relation:** plain API client of `https://api.trustedrouter.com/v1`. Most commits are upstream syncs.

3.5 trustedrouter-provider-check

- **Purpose:** prospective providers self-test against the exact contract the gateway enforces. Tiers 1-4 gate (catalog, billed sweep, chat semantics, strict SSE / usage / `[DONE]` / first-delta deadline / signed receipts). Tiers 5-6 are advisory (tools, structured output, perf).
- **Key dirs:** `src/tr_provider_check/{contract.py,snapshot.py,checks/,data/contract_snapshot.json}`.
- **Run:** `uv run trustedrouter-provider-check --base-url https://host/v1 --model X [--tier 6 --json out.json]`.
- **Relation (important):** `contract.py` is copied byte-for-byte from quill-router symbols and identified by a content-hash `contract_version`. On the quill-router side, `tests/test_provider_check_contract_parity.py` fails if production symbols change. The fix is to re-run `scripts/export_provider_check_contract.py` and open a sync PR here.

3.6 lightning-router

- **Purpose:** public, BUSL source mirror of LightningRouter (lightningrouter.ai): pay BTC over Lightning, get TR USD credits and an API key without email.
- **Relation:** not a separate service or ledger. The backend is in quill-router (`routes/internal/lightning.py`, `services/lightning.py`, `storage_lightning.py`, `tests/conformance/test_lightning.py`). This repo is auto-published by `github-actions[bot]` after green quill-router CI, and `SOURCE.json` holds the upstream commit plus per-file SHA-256. **Do not open PRs here;** change quill-router instead.
- **Run:** see `experiments/lightning_router/README.md`.

3.7 reverse-harness

- **Purpose:** "be a model on TrustedRouter". Modes: `proxy` (front Ollama/vLLM/llama.cpp), `exec` (spawn an agent per request, JSONL contract), `human` (browser dashboard where typing pauses become chunks). Uses a Cloudflare quick tunnel.
- **Contract (`CONTRACT.md v1`):** register via `POST /v1/user-models` (management key) and receive a `signing_secret` once. `clock-in` triggers a signed canary probe, then `heartbeat` / `clock-out`. Three consecutive owner faults clock you out. Kinds `machine` / `agent` / `human` have different price bounds and budgets. Not attested, not ZDR. Payout is 70% in credits.
- **Run:** `npm run reverse-harness --mode proxy --upstream http://localhost:11434/v1` with `TR_API_KEY`, `--model trustedrouter/user-<slug>`, `TR_SIGNING_SECRET`.
- **Relation:** the attested gateway calls out to the harness (see `docs/design/user-provided-model-egress.md` and `resolve-custom-model` in the proxy). Owner APIs live in quill-router.

3.8 TrustedRouter-Fusion-Draco

- **Purpose:** benchmark and marketing artifact. An agentic DRACO harness (`web_search` + `web_fetch` + `bash`), Gemini-3.1-Pro judge, and native `trustedrouter/fusion` generator. Headline: frontier + open-weights panel with a MiniMax-M3 fuser = 71.6, versus OpenRouter's best published 69.0.
- **Run:** `uv sync` (pulls `trusted-router-py`), then `uv run python scripts/draco_agentic_solo.py ...`, `draco_rejudge.py`, `draco_report.py`, `draco_client_fusion.py`.
- **Relation:** API client. Informs the fusion feature (`fusion.go` in the proxy). Dormant since June/Aug.

3.9 SREChat

- **Purpose:** multi-master BEAM chat that survives losing a cloud. One master per cloud (GCP / AWS / Azure). Coordination-free 53-bit IDs, a Redis-Stream oplog per region, convergent merges, and a chaos test in CI. Exposes the CometChat wire protocol.
- **Agent:** `agent/sre_agent.py` (stdlib Python) runs one agent per cloud with different default models (Kimi K3, GLM 5.2-Fast, DeepSeek). Each falls back to `trustedrouter/auto`. Read-only by default. Handles incident escalation, email, and APNs.
- **Run:** `mix deps.get && mix test && PORT=4000 ... mix run --no-halt`, or `docker compose up --build`.
- **Relation:** internal ops/coordination tool for TR's 3-cloud ops. It consumes the TR API and does not share a contract.

3.10 OpenChat

- AGPL CometChat-compatible chat server (Elixir, Redis), the ancestor of SREChat. No TR integration. Stable since Aug 2026.

3.11 AllRouter

- **Purpose:** one local endpoint (`:8383`) for local and cloud models over the OpenAI and Anthropic APIs, with concurrency, spend, privacy and fallback policy. The **BackupRouter** preset lets Claude Code Desktop/CLI and Codex use TR with Kimi K3 / GLM 5.2 recovery routes.
- **Run:** `brew tap Lore-Hex/homebrew-tap && brew install allrouter`, or `go install github.com/Lore-Hex/AllRouter/cmd/allrouter@latest`.
- **License:** Elastic-2.0. Dormant since 2026-08-16.

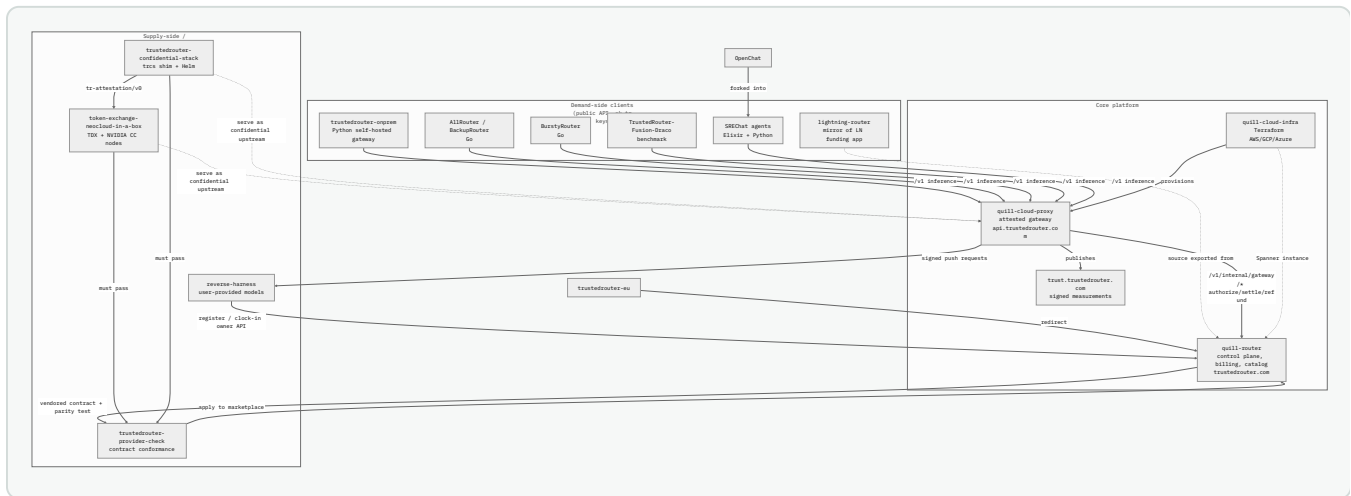
3.12 BurstyRouter

- **Purpose:** local-first proxy that sends to the local rig first and bursts to TR when local is full, failing, or missing the model. Routing directives include `local/<name>`, `provider.only:["local"]`, and aliases. Its marketing describes the `trustedrouter/zdr` and `trustedrouter/e2e` privacy routes.
- **Run:** `brew install burstyrouter`, then `burstyrouter`. **License:** Elastic-2.0. Dormant. Functionally superseded by AllRouter.

3.13 trustedrouter-eu

- **Purpose:** a single `index.html` + `CNAME` on GitHub Pages that redirects `eu.trustedrouter.com` to `trustedrouter.com/eu`. No license file. Nothing to maintain.

4. Ecosystem diagram



5. Onboarding notes / risks

1. **Read quill-cloud-proxy alongside quill-router.** Billing correctness is split across the Python `authorize/settle` handlers and the Go client (`internal/trustedrouter` , `billingv1`). The repos share fixtures, so a change in one usually needs a fixture or parity update in the other.
2. **Deploys are cross-repo coordinated** (two-cloud limit, shared journal, coordinator pinned by SHA). Never delete or expire the journal to unblock a rollout.
3. **Measurement changes break secret access by design.** Any workload code or build-arg change produces a new digest/PCRO/hostdata, and the old one must stay in the accepted set until the new one is attested and published.
4. **Bus factor ~1:** virtually all human commits across these 14 repos are by `jperla` .
5. **Confidential supply-side is pre-hardware:** confidential-stack and neocloud have not run on real CC hardware. Do not market routes from them as "confidential" yet.
6. **Mirrors:** do not edit lightning-router directly; it is generated from quill-router.

12 SDKs, Apps & Benchmarks

trusted-router-sdk-conformance



25 scenarios

retry 503/500/429 · x-should-retry
truncated SSE · redirects · replay
timeouts · cancel · OAuth · typed models

TLS fault server

loopback · throwaway CA
pretends to be api.trustedrouter.com

py driver

js driver

go driver

rust driver

java driver

swift driver

One black-box suite: 25 fault-injection scenarios, run against all six SDKs' real public APIs.

PART V · CHAPTER 12 · HERO STILL: VIDEO 12 AT 0:58

SDKs, apps & benchmarks

Every SDK, integration, app and benchmark repo, with a feature matrix.

- 1. Master table
- 2. SDK feature matrix
- 3. SDKs: testing, release, and conformance
- 4. Integrations and plugins
- 5. Apps
- 6. Benchmarks
- 7. Ecosystem layers
- 8. Which repo do I touch for X?
- 9. Open issues worth a CTO's attention

Storyboard · Video 12: SDKs, apps & benchmarks

0:37

0:42

0:46

0:58

1:02

1:11

1:24

1:41

Snapshot date: 2026-10-19. Sources: `gh repo list Lore-Hex` (47 repos), `gh api` (fork/parent, commits, releases), shallow clones of each repo (READMEs, manifests, workflows, PARITY docs), and public registry lookups (PyPI, npm, crates.io, proxy.golang.org, Maven Central).

Scope: the 32 repos below. The other 15 (`quill-router`, `quill-cloud-proxy`, `quill-cloud-infra`, `trustedrouter-confidential-stack`, `token-exchange-neocloud-in-a-box`, `trustedrouter-onprem`, `trustedrouter-provider-check`, `lightning-router`, `reverse-harness`, `TrustedRouter-Fusion-Draco`, `SREChat`, `OpenChat`, `AllRouter`, `BurstyRouter`, `trustedrouter-eu`) are in `09-repos-platform.md`. $32 + 15 = 47$, so every repo in the org is covered by one of the two docs. All repos are public. None are archived.

"30d commits" counts commits on the default branch since 2026-09-19. Many repos show a last commit of 2026-08-16. On that date the same CODEOWNERS and SECURITY.md commits landed across the org, so 2026-08-16 usually means "no real work since then."

1. Master table

REPO	CATEGORY	LANG	REGISTRY: NAME @ PUBLISHED VERSION	LAST COMMIT (DEFAULT BRANCH)	30D COM MIT S	FORK?	ONE-LINE PURPOSE
trusted-router-py	SDK	Python	PyPI <code>trusted-router-py</code> @ 0.9.0 (<code>import trustedrouter</code> , <code>CLI trustedrouter</code>)	2026-10-15	4	original	Reference SDK: sync and async clients, attestation, receipts, OAuth, orchestration
trusted-router-js	SDK + CLI	TS/JS	npm <code>@lore-hex/trusted-router</code> @ 0.10.1 (bin <code>trustedrouter</code>)	2026-10-15	21	original	JS/TS SDK and the official <code>trustedrouter</code> CLI. Runs on Node 20+, Deno, Bun, and browsers
trusted-router-go	SDK + CLI	Go	Go module <code>github.com/Lore-Hex/trusted-router-go</code> @ v0.5.0	2026-09-20	3	original	Zero-dependency Go SDK. Its <code>PARITY.md</code> is the most detailed spec of the two-plane and retry rules
trusted-router-rust	SDK + C ABI	Rust	crates.io <code>trusted-router</code> @ 0.5.1 . <code>trusted-router-ffi</code> is not on crates.io; it is built as release artifacts	2026-10-16	5	original	Async and blocking Rust, plus a stable C ABI
trusted-router-java	SDK	Java/Kotlin/Android	README says Maven <code>com.trustedrouter:trusted-router:0.4.0</code> . Not found on Maven Central (search and <code>repo1</code> both return nothing)	2026-09-26	3	original	JVM and Android SDK on OkHttp, with a <code>CompletableFuture</code> facade
trusted-router-swift	SDK	Swift	SwiftPM via git tag 0.9.0 (no <code>v</code> prefix). No registry	2026-09-20	3	original	Zero-dependency Swift (URLSession and CryptoKit). Supports macOS 13, iOS 16, tvOS, watchOS, and Linux
trusted-router-sdk-conformance	SDK test harness	Python	not published (<code>tr-conformance</code> CLI)	2026-10-03	3	original	Black-box TLS fault-injection suite. Runs 25 scenarios against all 6 SDKs
llm-trustedrouter	Integration	Python	PyPI <code>llm-trustedrouter</code> @ 0.1.0	2026-10-10	1	original	Plugin for Simon Willison's <code>llm</code> CLI
openclaw-trustedrouter	Integration	TS	npm <code>@lore-hex/openclaw-trustedrouter</code> @ 0.1.0 (also on ClawHub)	2026-10-10	1	original	Provider plugin for OpenClaw

REPO	CATEGORY	LANG	REGISTRY: NAME @ PUBLISHED VERSION	LAST COMMIT (DEFAULT BRANCH)	30D COMMIT S	FORK?	ONE-LINE PURPOSE
hermes-trustedrouter	Integration	Python	not published. Users install it by copying or symlinking into <code>\$HERMES_HOME</code>	2026-09-11	0	original	Model-provider plugin for NousResearch Hermes Agent
agent-zero-trustedrouter	Integration	YAML /config	not published (Agent Zero plugin index or git clone)	2026-08-16	0	original	Chat and embeddings provider for Agent Zero
n8n-nodes-trustedrouter	Integration	TS	npm <code>n8n-nodes-trustedrouter</code> @ 0.1.3	2026-08-16 (pushed 10-02 on another branch)	0	original	n8n community node: Chat Completions, Responses, routing, tools
llama-index-llms-trustedrouter	Integration	Python	PyPI <code>llama-index-llms-trustedrouter</code> @ 0.1.0	2026-08-16	0	original	Thin LlamaIndex <code>OpenAIIlike</code> subclass
trustedrouter-ai-sdk-provider	Integration	TS	intended npm <code>@trustedrouter/ai-sdk-provider</code> 0.1.0. Not on npm	2026-08-16	0	original	Vercel AI SDK provider built on <code>@ai-sdk/openai-compatible</code>
aisdk	Upstream fork	R	upstream CRAN-style <code>aisdk</code> 1.5.0	2026-08-16	0	fork of YuLab-SMU/aisdk	Staging fork. Branch <code>feat/trustedrouter-provider-discovery</code> . Upstream PR #36 is open
aisdk.providers	Upstream fork	R	upstream <code>aisdk.providers</code> 0.1.0	2026-08-16	0	fork of YuLab-SMU/aisdk.providers	Branch <code>feat/trustedrouter-provider</code> . Upstream PR #1 is open
LLM-advisor	Agent skill / MCP	Python + Markdown	ChatGPT directory submission (see <code>submission/</code>)	2026-10-01	2	original	"Model Advisor" skill and public read-only MCP for choosing models by cost, speed, and privacy
QuillCode	App (macOS)	Swift	DMG through GitHub Releases (<code>tester-latest</code>)	2026-09-15 (pushed 10-18)	0	original	"Quill Cowork": a native Swift coding agent. Also ships the locked Confidential Cowork by TrustedRouter build
confidential-cowork-desktop	App (macOS)	TS (Bun)	signed macOS packages linked from <code>trustedrouter.com/confidential-cowork</code>	2026-10-18	7	original (snapshot port of anomalyco/opencode 2.0.32)	"Trusted Cowork" desktop agent. Every request uses <code>trustedrouter/confidential</code> and there is no fallback
trpi	App (CLI)	TS	GitHub Releases binaries (<code>tr-cowork</code>)	2026-10-03	1	original (distribution of earendil-works/pi)	"TR Confidential Cowork" terminal coding harness. Built on Pi, with TrustedRouter as the default provider
FastWorker	Upstream fork / app	Python	none	2026-08-16	0	fork of andrewyng/openworker	Branch <code>add-trustedrouter-provider</code> . Upstream PR <code>andrewyng/openworker#25</code> is open
openclaw	Upstream fork	TS	none	2026-08-16	0	fork of openclaw/openclaw	Branch <code>codex/add-trustedrouter-provider</code> . Upstream PRs #83586, #92544, and #97136 are all closed . The supported path is now the plugin repo

REPO	CATEGORY	LANG	REGISTRY: NAME @ PUBLISHED VERSION	LAST COMMIT (DEFAULT BRANCH)	30D COMMIT S	FORK?	ONE-LINE PURPOSE
hermes-agent	Upstream fork	Python	none	2026-08-16	0	fork of NousResearch/hermes-agent	Branch <code>codex/add-trustedrouter-provider</code> . Upstream PRs #27974 and #53364 are closed . Hermes policy requires a standalone plugin, which is why hermes-trustedrouter exists
QuillUI	App framework	Swift	none	2026-08-15	0	original	Lets Apple SwiftUI/AppKit source build on Linux (Enchanted/Quill Chat port). Not TR-specific
OllamaKit	Library fork (detached)	Swift	none	2026-08-16	0	not a GitHub fork. A detached copy of kevinhermawan/OllamaKit	Adds Alamofire <code>Session</code> injection so Quill Chat can pin certificates. Not TR-specific
QuiLLM	Model research	Python	model files through releases and docs	2026-10-16	5	original	Qwen3 political-censorship reduction experiment for QuillOS. Needs no TR account
PrometheusBench	Benchmark	Python	none. Depends on <code>trusted-router-py>=0.3.0</code>	2026-09-02	0	original	Short-prompt refusal and permissiveness benchmark (bio, LLM research, ...). Has a Fusion run guide
prometheus-biomysterybench	Benchmark	Python	none (<code>huggingface_hub</code> , raw TR HTTP)	2026-08-16	0	original	Local Dockerized harness for Anthropic's BioMysteryBench public preview
Trusted Router-ExploitBench	Benchmark (notes)	YAML /Markdown	none	2026-08-16	0	original (references upstream exploitbench)	Runbook and configs for running ExploitBench V8 through TR, including Fusion as a model
Prometheus-ExploitBench	Benchmark (research)	Python	none	2026-08-16	0	original	First-party ExploitBench reproduction. Documents Opus "thinking-only" degeneration
FreedomBench	Benchmark	Python	none. Depends on <code>trusted-router-py>=0.3.0</code>	2026-08-31	0	original	60-question multiple-choice benchmark on PRC-censored facts. No judge model
trustedrouter-benchmarks	Benchmark	Python	none. Depends on <code>trusted-router-py>=0.3.0</code>	2026-08-16	0	original	IFEval, MMLU-Pro, SciCode, and others on Chinese open-weight models through TR

Activity summary. Repos with active work in the last 30 days: `trusted-router-js` (21), `confidential-cowork-desktop` (7), `trusted-router-rust` (5), `QuiLLM` (5), and `py/go/java/swift/conformance` (3–4 each). Repos dormant since mid-August: most third-party integrations, all R/agent forks, all benchmarks except `FreedomBench` and `PrometheusBench` (both early September).

Version drift and gotchas. -GitHub Releases pages lag the registries. `py` shows release `v0.5.0` while PyPI has 0.9.0. `js` shows `v0.7.0` while npm has 0.10.1. Java shows `v0.2.1` while the latest tag is `v0.4.0`. Treat **tags and registries as the source of truth**, not the Releases page. -The Java SDK's README advertises Maven Central 0.4.0, but the artifact cannot be found on Central. Check this before telling customers to use it. -`trustedrouter-ai-sdk-provider` is not on npm, although the README says `npm install @trustedrouter/ai-sdk-provider`. Its publish workflow only runs on a GitHub Release, and none has been cut. -Swift tags have no `v` prefix (`0.9.0`). Every other SDK uses `vx.y.z`. -Apps

pin old SDKs. confidential-cowork-desktop pins `@lore-hex/trusted-router` **0.4.0**, while npm has 0.10.1. QuillCode requires `trusted-router-swift` from: 0.6.1. -The confidential-cowork-desktop repo description says "based on Pi Desktop", but its README and UPSTREAM.md say it is a port of **OpenCode** 2.0.32. The Pi-based product is `trpi`.

2. SDK feature matrix

Built from each SDK's README and PARITY.md, plus source greps. "Y" means a dedicated public API. "via builder/alias" means it is reachable through the orchestration tool builders or the `trustedrouter/fusion` model alias, without a dedicated method.

FEATURE	PY	JS	GO	RUST	JAVA	SWIFT
Chat Completions (typed)	Y	Y	Y	Y	Y	Y
SSE streaming (chunks/text/raw, fail-closed on missing <code>[DONE]</code>)	Y	Y	Y	Y	Y	Y
Responses API and Responses SSE events	Y	Y	Y	Y	Y	Y
Responses input-token count	Y	Y	Y	Y	Y	Y
Anthropic Messages	Y	Y	Y	Y	Y	Y
Embeddings	Y	Y	Y	Y	Y	Y
Fusion (<code>fusion()</code>)	Y	Y	Y	via <code>synth_tool</code> / <code>FUSION_MODEL</code>	Y (<code>synth</code> , <code>fusion</code>)	via <code>fusionModel</code> alias and builders
Orchestration builders (Synth, Advisor, Selector, MapReduce, Subagent)	Y	Y	Y	Y	Y	Y
Routing and privacy aliases (auto, fast, zdr, e2e, confidential, eu, us, ...)	Y	Y	Y	Y	Y	Y
Gateway attestation verification (GCP Confidential Space JWT, nonce, image, TLS leaf)	Y (<code>[attestation]</code> extra)	Y (WebCrypto)	Y (strongest: TLS-exporter session binding)	Y (exporter passed explicitly)	Y (cert-bound; exporter if caller supplies it)	Y
Attested session helper	Y (<code>session.py</code>)	Y (<code>session.ts</code> , <code>attest --session</code>)	Y	—	—	Y
Inference receipt verification (Ed25519 JWS)	Y (<code>receipts.py</code>)	Y (<code>receipts.ts</code>)	Y (<code>receipts.go</code>)	Y (<code>receipts.rs</code>)	Y (<code>ReceiptVerifier</code>)	Y
OAuth credit delegation (PKCE, loopback)	Y	Y	Y	Y	Y (+ Android deep links)	Y
Control plane: models, providers, regions, credits, activity	Y	Y	Y	Y	Y	Y
Broadcast destinations CRUD	Y	Y	Y	Y	Y	Y
Billing checkout / stablecoin checkout	Y / Y	Y / Y	Y / Y	Y / Y	Y / Y	partial / —

FEATURE	PY	JS	GO	RUST	JAVA	SWIFT
Two-plane split (inference <code>api.trustedrouter.com/v1</code> , control <code>trustedrouter.com/v1</code>)	Y	Y	Y	Y	Y	Y
Retry, Retry-After, <code>x-should-retry</code> , generated idempotency keys	Y	Y	Y	Y	Y	Y
Client telemetry: <code>x-tr-client</code> header	Y	Y	Y	Y	Y	Y
Client telemetry: <code>/v1/client-events</code> beacon	Y	Y	Y	deferred (documented)	Y	Y
Bundled CLI	Y (trusted router)	Y (trustedrouter: chat, models, attest, trust, ...)	Y (cmd/trustedrouter)	—	—	—
Extra runtime targets	async	Node, Deno, Bun, browser	—	blocking facade, C ABI (trusted-router-ffi)	Kotlin, Android, CompletableFuture	iOS, tvOS, watchOS, Linux
Runtime deps	httpx-based	none (no native deps)	none	request/rustls	OkHttp	none
Conformance-suite driver	Y	Y	Y	Y	Y	Y

The Go PARITY.md is the reference for the "two-plane routing" and "stream-open retry" divergences from the older py/js behavior. Rust and Java PARITY.md say they follow the Go contract.

3. SDKs: testing, release, and conformance

SD K	CI (.GITHUB/WORKFLOWS)	RELEASE TRIGGER	PUBLISH MECHANISM	EXTRA GATES
py	ci.yml, conformance.yml, release.yml	push of tag <code>vX.Y.Z</code>	PyPI Trusted Publishing (uv)	ruff, boundary_check.py, mypy, pytest, mutation_check.py
js	ci.yml, conformance.yml, release.yml	push of tag <code>vX.Y.Z</code>	npm granular token	build, lint, type-tests, attw/publint, test
go	ci.yml, conformance.yml	git tag (Go proxy picks it up)	none needed	parity_test.go compile-time symbol gate, deep conformance tests
rust	ci.yml, conformance.yml, live-smoke.yml, release.yml	tag <code>v*</code> (workflow_dispatch refuses non-tag refs)	cargo publish -p trusted-router --locked, plus FFI release build	clippy, rustfmt, live smoke
java	ci.yml, conformance.yml, release.yml	tag <code>v*</code>	vanniktech maven-publish to Maven Central (artifact not currently visible on Central)	errorprone, javadoc, mutation_check.py, gradle wrapper validation
swift	ci.yml, conformance.yml	git tag (no <code>v</code>)	SwiftPM git; scripts/build-artifact.py for a clean archive	—

trusted-router-sdk-conformance

- **What it is:** one black-box suite run against all six SDKs. A thin per-language driver (`drivers/{python,javascript,go,rust,java,swift}`) calls the SDK's real public API. The driver points `https://api.trustedrouter.com` at an ephemeral TLS loopback fault server with a throwaway CA. A shared oracle then checks both the normalized result and the wire transcript.

- **Scenarios (25, `scenarios/*.json`):** success; retry on 503, 500, and 429; `x-should-retry` true/false verdicts; retry exhaustion; transport disconnect; telemetry across retries; reserved-header opt-out; terminal 401; truncated and malformed SSE; cross-origin redirect; generated idempotency keys (with and without a disconnect); unsafe POST replay; body timeout; caller cancellation; credential-free OAuth plus OAuth redirect; chat and Responses typed-model preservation; retry on a truncated or stalled error body.
- **Coverage (docs/coverage.md):** all 6 SDKs pass all 7 capability groups (core retry and telemetry, SSE integrity, redirect isolation, safe replay, body timeout and cancellation, credential-free OAuth, typed model preservation). The v0.2 gate allows **no skips**.
- **Not covered by the shared matrix:** regional affinity/failover, first-frame streaming deadlines, SSE heartbeat idle timing, and **cryptographic attestation validation**. Each SDK covers these only in its own native tests.
- **Wiring:** each SDK's `conformance.yml` calls the reusable workflow `Lore-Hex/trusted-router-sdk-conformance/.github/workflows/sdk-pr.yml@main` on PRs. The harness's own `conformance.yml` runs on PR and push, nightly (cron `17 4 * * *`), and on `repository_dispatch` (`sdk-conformance`). Its manual dispatch takes a ref input for each SDK.
- **Not covered at all:** the integrations, apps, and benchmarks. Integrations do not use the SDKs (see §4).

4. Integrations and plugins

Every first-party integration is a **thin OpenAI-compatible wrapper**. Each one points at `https://api.trustedrouter.com/v1` and reads the model catalog from `GET /v1/models`. **None of them depends on an official SDK**, so none of them gets attestation, receipts, or SDK retry semantics, and the conformance suite does not cover them.

REPO	HOST FRAMEWORK	BUILT ON	FEATURES	TESTS / RELEASE
llm-trustedrouter	llm CLI	llm>=0.27, httpx	chat, streaming, catalog cached 1h, llm <code>trustedrouter models</code>	<code>test.yml</code> (pytest-httpx). Published to PyPI 0.1.0
openclaw-trustedrouter	OpenClaw >=2026.9.9 (peer dep)	native plugin (<code>openclaw.plugin.js</code> on), <code>openai-completions</code> API	chat, streaming, default <code>trustedrouter/auto</code> , its own catalog fetch (catalog is larger than OpenClaw's 4 MiB discovery cap). README links to the attestation docs	vitest <code>test.yml</code> . npm and ClawHub 0.1.0
hermes-trustedrouter	Hermes Agent user plugin	Python provider profile	chat, model list	<code>test.yml</code> . Not published (git copy or symlink)
agent-zero-trustedrouter	Agent Zero	<code>plugin.yaml</code> + <code>conf/</code>	chat and embeddings providers, <code>trustedrouter/*</code> routing aliases	no CI
n8n-nodes-trustedrouter	n8n	n8n-workflow node + credential	Chat Completions, Responses, auto routing, fallbacks, tags, tools	<code>ci.yml</code> . <code>publish.yml</code> runs on a bare semver tag. npm 0.1.3. Waiting on n8n verification
llama-index-llms-trustedrouter	LlamaIndex	<code>llama-index-llms-openai-like</code>	complete, chat, stream	<code>test.yml</code> . <code>publish.yml</code> runs on a GitHub Release. PyPI 0.1.0
trustedrouter-ai-sdk-provider	Vercel AI SDK	<code>@ai-sdk/openai-compatible v1</code>	<code>generateText/streamText</code> through <code>trustedrouter('<model>')</code>	<code>test.yml</code> . <code>publish.yml</code> runs on a GitHub Release. Not on npm yet

REPO	HOST FRAMEWORK	BUILT ON	FEATURES	TESTS / RELEASE
LLM-advisor	Codex, Claude Code, Hermes, ChatGPT	SKILL.md , agents/ , plugins/ , public read-only MCP	Model selection by cost, speed, privacy, and route. Cost estimates. Three-prompt eval planning. No key needed	plugin-validation.yml . ChatGPT directory submission in submission/chatgpt-2026-09.md

Upstream-contribution forks (staging branches only; main is just the CODEOWNERS/SECURITY commits)

FORK	UPSTREAM	TR BRANCH	UPSTREAM PR STATUS	TAKEAWAY
aisdk	YuLab-SMU/aisdk	feat/trustedrouter-provider-discovery	#36 open (2026-07-25)	Waiting on the R maintainers
aisdk.providers	YuLab-SMU/aisdk.providers	feat/trustedrouter-provider	#1 open (2026-07-25)	Waiting on the R maintainers
FastWorker	andrewyng/openworker	add-trustedrouter-provider	#25 open (2026-07-23)	Also branded "FastWorker: open-source desktop AI worker powered by TR"
openclaw	openclaw/openclaw	codex/add-trustedrouter-provider	#83586, #92544, #97136 closed	Abandoned in favor of openclaw-trustedrouter
hermes-agent	NousResearch/hermes-agent	codex/add-trustedrouter-provider	#27974, #53364 closed	Abandoned in favor of hermes-trustedrouter

The forks are thousands of commits behind upstream (openclaw: 63,960; hermes: 43,577). Don't build on their `main`. Rebase the TR branch onto upstream when refreshing a PR.

5. Apps

REPO	WHAT	TR COUPLING	BUILD / TEST / RELEASE
QuillCode ("Quill Cowork" and "Confidential Cowork by TrustedRouter")	Native Swift macOS coding agent: chat, local tools, Git worktrees, Computer Use, automations, terminal, verified updater with rollback	Uses trusted-router-swift (from: 0.6.1). Has <code>TrustedRouterIntegration.swift</code> and <code>TrustedRouterPromptCaching.swift</code> . The Confidential build is locked to <code>trustedrouter/confidential</code> and the official endpoint, applies US/EU policy, and fails closed	10 workflows: <code>ci</code> , <code>live-trustedrouter-smoke</code> (<code>scripts/live-tr-smoke.sh</code>), <code>real-world-smoke</code> , <code>merge-train</code> , signing for three app identities, downloads, website. Ships a DMG through the <code>tester-latest</code> release
confidential-cowork-desktop ("Trusted Cowork")	Bun/TS desktop agent. Snapshot port of OpenCode 2.0.32 onto the V2 core	<code>packages/desktop</code> depends on <code>@lore-hex/trusted-router 0.4.0</code> . PKCE sign-in, model allowlist, <code>trustedrouter/confidential</code> default, no fallback backend, telemetry off. README says it does not verify attestation on the client socket	<code>trusted-cowork-ci.yml</code> . Signed macOS packages linked from the TR site, manual updates. See <code>TRUSTED-COWORK.md</code> and <code>UPSTREAM.md</code>
trpi ("TR Confidential Cowork" CLI, <code>tr-cowork</code>)	Terminal coding harness. TR-first distribution of the Pi Agent Harness	TR catalog is the default, with a live <code>/model</code> selector. State lives in <code>~/tr-confidential-cowork/agent</code>	<code>ci.yml</code> , <code>build-binaries.yml</code> , <code>npm-audit.yml</code> . Binaries on GitHub Releases with SHA256SUMS
FastWorker	Fork of andrewyng/openworker with a TR provider	provider branch only	see §4
QuillUI	Lets Apple Swift app source build on Linux (Enchanted/Quill Chat, Signal port, SceneKit lane)	none directly. QuillOS ecosystem	linux/macOS CI, enchanted-parity, solderscope

8. Which repo do I touch for X?

I WANT TO...	TOUCH	ALSO UPDATE
Add a new API endpoint or field to the SDKs	All six <code>trusted-router-*</code> SDKs. Start with py (reference) or go (PARITY spec)	Each SDK's <code>PARITY.md</code> and <code>CHANGELOG</code> . Add a scenario in <code>trusted-router-sdk-conformance</code> if retry or wire behavior changes
Change retry, idempotency, redirect, or SSE semantics	The SDK transport layers (py <code>_transport.py</code> / <code>_retry.py</code> , js <code>src/internal/transport.ts</code> , go <code>transport.go</code> , rust <code>transport/engine.rs</code> , java <code>internal/Transport.java</code> , swift Core)	trusted-router-sdk-conformance <code>scenarios/</code> + <code>docs/coverage.md</code>
Change attestation or receipt verification	<code>attestation.*</code> and <code>receipts.*</code> in every SDK. Go has the strongest session binding	Not in the conformance matrix, so add native tests in each SDK
Add a model alias (e.g. a new <code>trustedrouter/*</code> route)	Constants in all 6 SDKs (<code>_constants.py</code> , <code>constants.go</code> , <code>constants.rs</code> , <code>Constants.swift</code> , ...)	Integration READMEs (agent-zero lists aliases), LLM-advisor references
Release an SDK	Push tag <code>vX.Y.Z</code> (Swift: <code>X.Y.Z</code>). py, js, rust, and java publish from CI. Go and Swift publish through the tag alone	Bump the version constant (<code>version.go</code> , <code>pyproject.toml</code> , <code>package.json</code> , <code>Cargo.toml</code> workspace, <code>VERSION_NAME</code>). The GitHub Releases page is not kept up to date
Ship or fix the CLI	<code>trusted-router-js</code> (<code>src/cli.ts</code>) is the official npm CLI. py <code>__main__.py</code> and go <code>cmd/trustedrouter</code> also exist	—
Add a C/C++ binding	<code>trusted-router-rust/crates/trusted-router-ffi</code>	Rust <code>release.yml</code> FFI build
Fix the llm / OpenClaw / Hermes / Agent Zero / n8n / LlamaIndex / Vercel AI SDK integration	The matching <code>*-trustedrouter</code> repo (or <code>trustedrouter-ai-sdk-provider</code>)	Publish: PyPI for llm and llama-index, npm for openclaw, n8n, and ai-sdk
Get TR into an upstream agent framework	Upstream PR from the matching fork's TR branch (aisdk, aisdk.providers, FastWorker). For OpenClaw and Hermes, use the plugin repos : upstream rejected in-tree providers	—
Help agents pick models	<code>LLM-advisor</code>	—
Change the native Mac coding app or the Confidential Cowork lock-down	<code>QuillCode</code>	<code>trusted-router-swift</code> if SDK changes are needed
Change the Trusted Cowork (Electron-style, OpenCode-based) desktop	<code>confidential-cowork-desktop</code>	Bump its pinned <code>@lore-hex/trusted-router</code> (still 0.4.0)
Change the terminal coding harness <code>tr-cowork</code>	<code>trpi</code>	—
Run or publish refusal or capability benchmarks	<code>PrometheusBench</code> , <code>FreedomBench</code> , <code>trustedrouter-benchmarks</code> , <code>prometheus-biomysterybench</code> , <code>*-ExploitBench</code>	Use disposable keys (ban-risk warnings)
Linux SwiftUI / QuillOS work	<code>QuillUI</code> , <code>OllamaKit</code> , <code>QuillLLM</code>	Not TrustedRouter product code
Router, gateway, billing, infra	Not here. See <code>09-repos-platform.md</code> (quill-router, quill-cloud-proxy, quill-cloud-infra, ...)	—

9. Open issues worth a CTO's attention

1. **Java is not on Maven Central**, although the README and release workflow say it is. Check the Sonatype credentials and namespace for `com.trustedrouter`.
2. **The Vercel AI SDK provider was never published**. It needs a GitHub Release, and the npm scope `@trustedrouter` has to exist.
3. **Integrations skip the SDKs**. They lose attestation, receipts, and SDK retry semantics, and the conformance suite cannot cover them. Decide whether that is acceptable or whether the Python and TS integrations should wrap `trusted-router-py` / `@lore-hex/trusted-router`.
4. **confidential-cowork-desktop pins SDK 0.4.0**, six minors behind npm 0.10.1. This is a confidential-tier product.
5. **Attestation validation and regional failover are not in the shared conformance matrix**. `docs/coverage.md` says so explicitly.
6. **The Releases pages are stale** for py, js, and java. Customers and agents who read them will see wrong versions.

06

PART VI

Leadership

1. The CTO view

13 The CTO View

25 incident write-ups, Sep 5 to Oct 18



Sep 26 grant storm

94-96% transaction aborts
authorize p50 0.1 s → 3.7 s
fleet-wide

Roughly one incident every two days. Most trace to Spanner or to the shadow system itself.

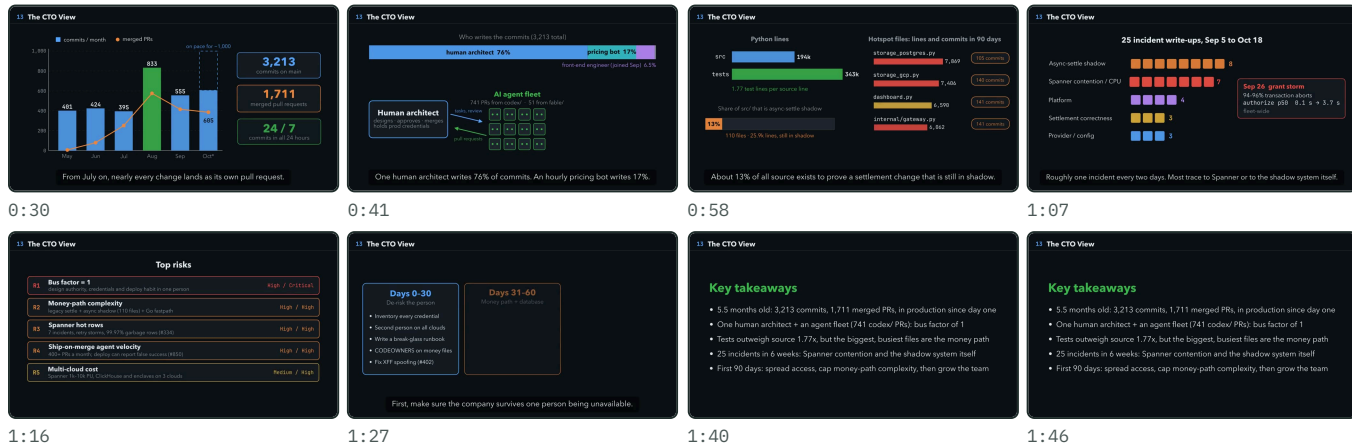
PART VI • CHAPTER 13 • HERO STILL: VIDEO 13 AT 1:07

The CTO view

History, velocity, team, codebase health, risk register and a 30/60/90-day plan.

- TL;DR
- 2. Velocity
- 4. Codebase size and health
- 6. Incidents (25 write-ups, 2026-09-05 → 2026-10-18)
- 8. Synthesis
- 1. Timeline
- 3. Team and contributors
- 5. Open work
- 7. Design docs in flight (docs/design/, 46 files)
- 9. 30/60/90-day plan for a new technical leader

Storyboard · Video 13: The CTO view



Snapshot taken 2026-10-19 at `main @ 778120494`. Everything here comes from read-only `git / gh` queries and repository docs. No production system was touched. Raw numbers: [11-stats.json](#).

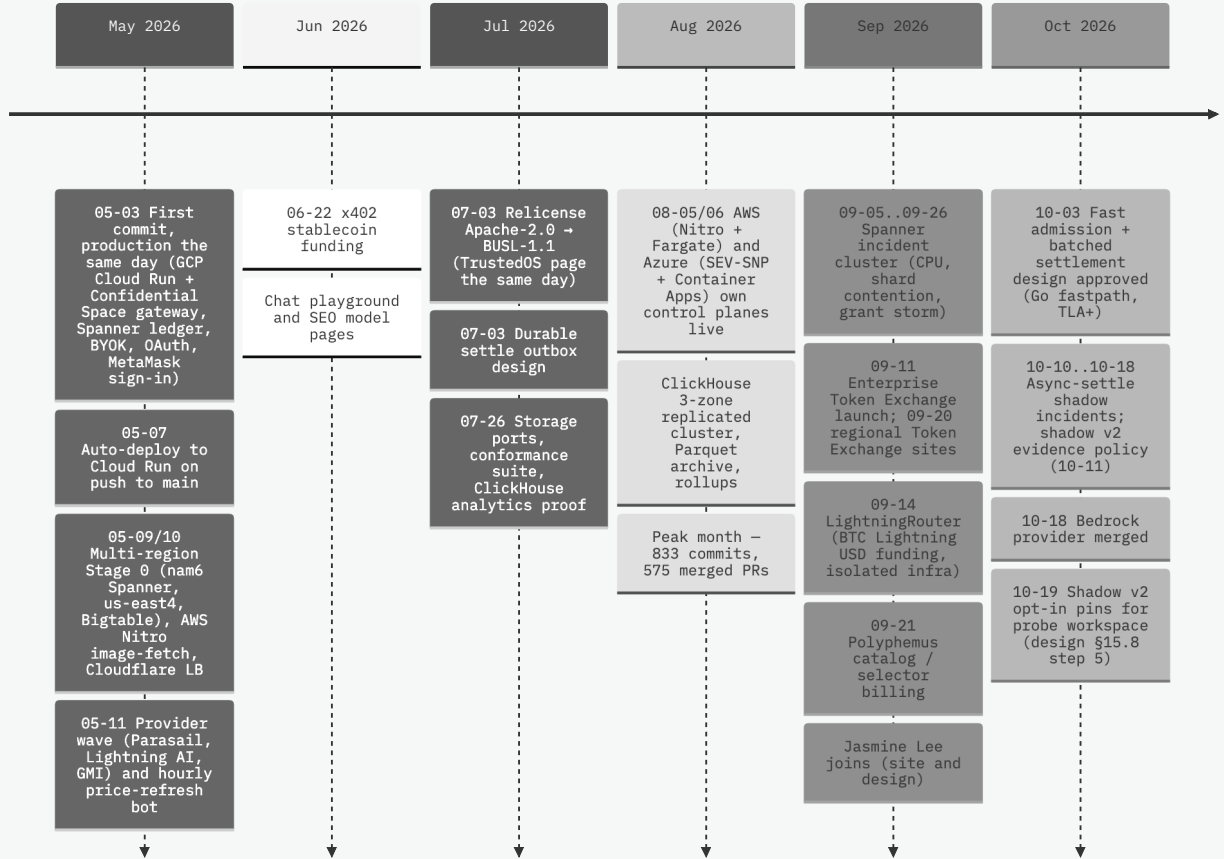
TL;DR

- **Age:** 5.5 months. The first commit, "TrustedRouter alpha control plane", landed 2026-05-03. The project dropped its alpha label the same day.
- **Scale:** 3,213 commits, 1,711 merged PRs, ~194k LOC of Python in `src/`, ~343k LOC of Python tests, a ~20k LOC Go fast-path service and a set of TLA+ proofs. It runs live on **three clouds** (GCP primary, plus AWS and Azure as standalone deployments).
- **Team:** effectively **one founder-engineer (Joseph Perla, 76% of commits)**. He drives a fleet of AI coding agents: 741 merged PRs come from `codex/*` branches and 51 from `fable/*`. A second human (Jasmine Lee, front-end/site work) joined in September 2026. A pricing bot makes 17% of all commits.
- **Velocity:** about 600 commits and about 400 merged PRs a month, with commits spread across all 24 hours. That is agent-paced, not human-paced.
- **Biggest risks:** bus factor of 1. Money-path complexity (async settle "shadow v2", 110 source files and ~26k LOC of evidence machinery). Spanner contention, the source of most incidents. A large multi-cloud cost footprint for a single operator.
- **Moat:** hardware attestation (Confidential Space / Nitro / SEV-SNP) plus public BUSL source. Anyone can verify that the running binary matches the repo.

1. Timeline

Dates are inferred from commit messages and the history of key files.

TrustedRouter (quill-router) – 2026

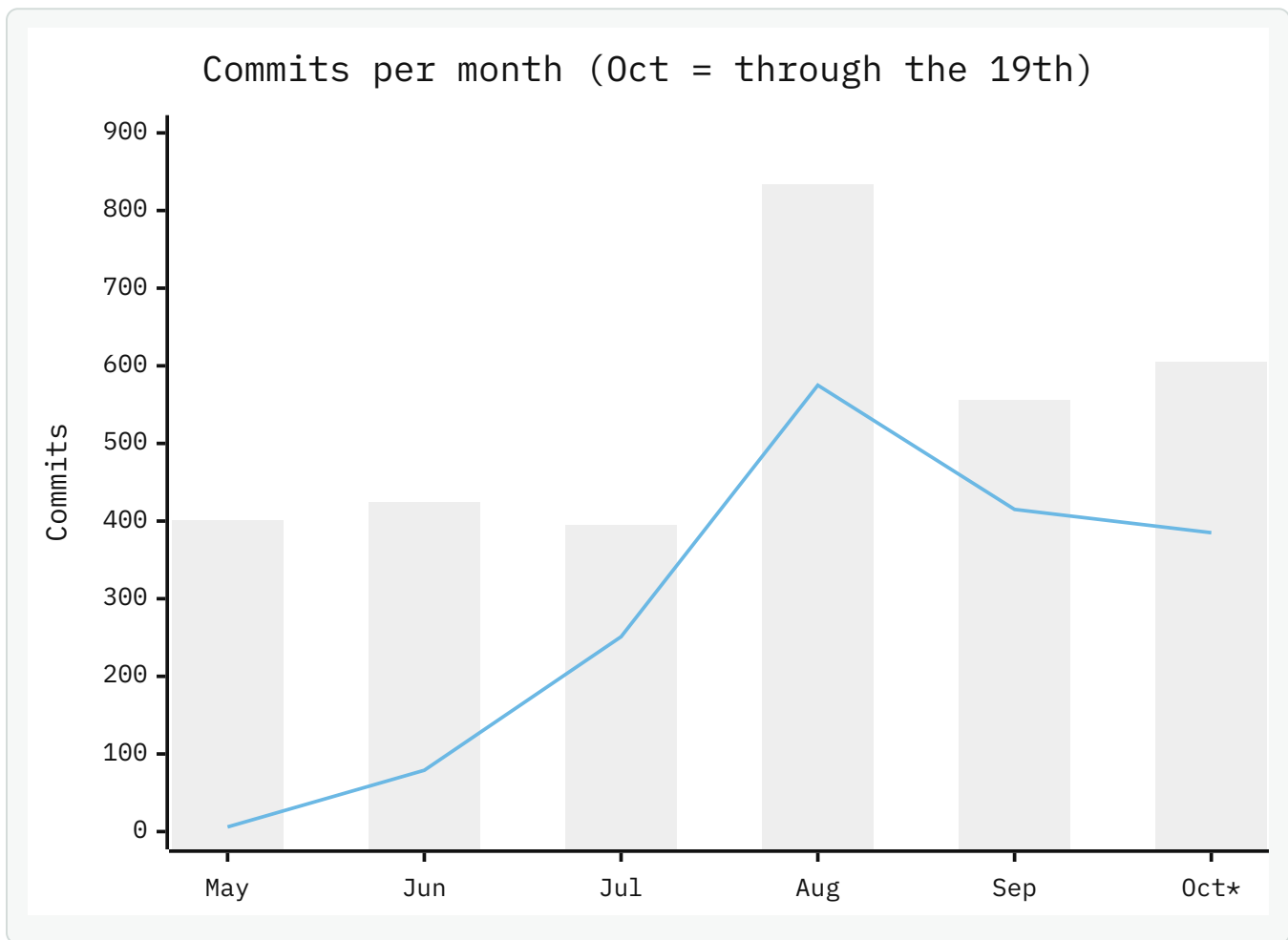


Milestone table

DATE	MILESTONE	EVIDENCE
2026-05-03	First commit; "Drop alpha framing — TrustedRouter is production now"	a0e42330c
2026-05-07	GitHub Actions auto-deploys the control plane to Cloud Run	commit log
2026-05-09	Multi-region Stage 0: nam6 Spanner, Bigtable, us-east4	commit log
2026-06-22	x402 stablecoin funding	commit log
2026-07-03	Relicense to BUSL-1.1. Earlier code stays Apache-2.0; each version converts to Apache-2.0 after 4 years	8155ac666 , README
2026-07-26	Storage behind ports, conformance suite, ClickHouse proof (#288/#289)	commit log
2026-08-06	AWS and Azure control-plane independence live	docs/storage-portability/HANDOFF.md
2026-09-11	Enterprise Token Exchange (#1163)	commit log

DATE	MILESTONE	EVIDENCE
2026-09-14	LightningRouter (#1175)	commit log
2026-10-03	Fast admission / batched settlement design approved (v25), with TLA+ specs	docs/design/fast-admission-and-batched-settlement.md
2026-10-11	Async settle shadow evidence policy v2	docs/design/async-settle-shadow-v2.md
2026-10-19	Shadow v2 §15.8 step 5 deployed (#1853)	HEAD

2. Velocity



Bars show commits. The line shows merged PRs.

MONTH	COMMITTS	MERGED PRS	JOSEPH	PRICING BOT	JASMINE	DEPENDABOT
2026-05	401	6	206	195	-	-
2026-06	424	79	378	46	-	-
2026-07	395	251	348	47	-	-
2026-08	833	575	651	171	-	11

MONTH	COMMITTS	MERGED PRS	JOSEPH	PRICING BOT	JASMINE	DEPENDABOT
2026-09	555	415	449	62	42	2
2026-10*	605	385	417	22	166	-

Observations: - The workflow moved from direct pushes (May) to a **PR-per-change** workflow (July onward). Only 191 merge commits exist; most PRs are squash-merged. - **October is on pace for about 1,000 commits**, the highest month yet: 207 commits in the last 7 days. - Commit hours: every hour from 00:00 to 23:00 local time has 47 to 190 commits. Agents work around the clock and the founder reviews and merges. - Active days: 166 of 170 calendar days.

3. Team and contributors

CONTRIBUTOR	COMMITTS	SHARE	ROLE
Joseph Perla	2,449	76.2 %	Founder/CEO-CTO. Approves every design ("Joseph selected...", "approved by Joseph"). Default is to deploy what merges
trustedrouter-pricing-bot	543	16.9 %	Hourly price/catalog refresh. Commits straight to <code>main</code>
Jasmine Lee	208	6.5%	Front-end, design, Token Exchange sites, homepage. Two git identities (190 + 18)
dependabot[bot]	13	0.4%	Dependency bumps. 10 dependabot PRs sit open, the oldest from 2026-08-31

AI agents as a workforce. Merged-PR branch prefixes: `codex/` 741, `fix/` 114, `fastpath/` 80, `fable/` 51, `feat/` 39, `async-settle/` 36. Design docs record adversarial review rounds by "Codex and Fable". `AGENTS.md` is written for agents first. In practice the engineering org is **one human architect plus several agent reviewers and implementers**.

Bus factor: 1. Joseph is the sole approver of money-path design, the holder of production credentials (README points at his local key file), and the author of 76% of the code. Jasmine's work stays mostly within `sites/`, the homepage and templates.

4. Codebase size and health

LOC by language (tracked files, `wc -l`)

LANGUAGE	LOC	NOTES
Python	618,435	<code>tests</code> 343k · <code>src</code> 194k · <code>scripts</code> 60k · <code>clickhouse</code> 9k · <code>experiments</code> 7k · <code>proofs</code> 3k
JSON	270,072	Mostly provider catalogs (92 provider files) plus OpenRouter snapshot
Markdown	57,641	46 design docs, 25 incident reports, runbooks
Go	54,518	<code>fastpath/</code> (fast admission). 19.7k <code>src</code> / 34.8k <code>tests</code>
HTML (Jinja)	25,356	Server-rendered console and marketing
Shell	23,685	Deploy/rollout (<code>scripts/deploy/rollout.sh</code> is config-as-code)
JavaScript / TS / CSS	17.6k / 4.3k / 15.2k	Frontend, e2e (Playwright)
Terraform / SQL	1.9k / 1.9k	<code>infra/</code> (Spanner, Cloud Armor, LB)

Tracked files: 3,814. Workflows: 28. TLA+ models in `proofs/` (AuditorCommit, AutoRefillHandoff, and others).

Test-to-code ratio

- **Python: 1.77 lines of test per line of source** (886 test files vs 533 source files).
- **Go: 1.77 to 1** as well.
- CI gates: `ruff check`, `mypy` (strict gate), `pytest` with **at least 70% coverage**, plus conformance suites against memory, Postgres and Spanner-PG, and a sharded `proof_oracle` differential suite (more than 1 hour).

Largest source modules

FILE	LOC	SMELL
<code>storage_postgres.py</code>	7,869	One backend implements a ~97-method <code>Store</code> Protocol
<code>storage_gcp.py</code>	7,406	Spanner backend. 3rd most churned file
<code>dashboard.py</code>	6,590	UI god-module. Churned in 141 commits in 90 days
<code>routes/internal/gateway.py</code>	6,062	The hot path (authorize/settle). Churned in 141 commits in 90 days
<code>catalog_data.py</code>	4,456	Catalog logic and data
<code>storage.py</code>	4,103	<code>STORE</code> module-global proxy (issue #333)
<code>routes/public.py</code>	3,727	
<code>synthetic/probes.py</code>	3,341	

Async-settle footprint: 110 files and ~25.9k LOC in `src/` matching `async_settle*`, most of it `async_settle_shadow_v2_*` evidence, gate and certificate machinery. That is about 13% of all source code, spent proving a settlement change that is still in shadow.

Hot files (last 90 days)

All authors: the top 30 are all catalog JSON (`openrouter_snapshot.json` 303 commits, each `provider_models/*.json` about 250 to 300), driven by the hourly pricing bot.

Humans and agents, non-test:

FILE	COMMITTS (90D)
<code>routes/internal/gateway.py</code>	141
<code>dashboard.py</code>	141
<code>storage_gcp.py</code>	140
<code>config.py</code>	134
<code>scripts/deploy/rollout.sh</code>	119
<code>static/og/providers/manifest.json</code>	112
<code>storage_postgres.py</code>	105
<code>routes/public.py</code>	102
<code>catalog_data.py</code>	100
<code>storage.py</code>	96
<code>store_protocol.py</code>	77
<code>.github/workflows/deploy.yml</code>	68

Reading: churn concentrates in the money/hot path (gateway, storage) and in config/deploy. Those are the highest-risk files and also the most edited.

5. Open work

38 open PRs (12 drafts). Themes: - **New providers / catalog** (about 12): Hixtube, Regolo, CoralBricks, NextBit EU, Verda, Redpill/Meta/General Compute/Infomaniak, Poolside, Fortanix-on-Azure. - **Routing correctness:** context-window-aware routing (#1800); tool/response_format capability filtering (#1798). - **Money path / async settlement:** regional Bigtable settlement journal (#1403, dormant), stage C/D admission (#1296, #1055), idempotent-replay key-hold leak (#1239), fastpath renewal-lock release (#1844). - **Infra:** control LB URL map into Terraform (#1862). - **Dependabot backlog:** 10 PRs, the oldest from 2026-08-31 (actions/checkout 4→7, setup-gcloud 2→3, psycopg, google-cloud-storage 3.x).

30 open issues (only 15 were ever closed through Issues; most work is tracked in PRs and design docs). Notable: - #402 `_client_ip` trusts the first `X-Forwarded-For`. **Spoofable rate-limit identity** (open since 2026-08-02). - #334 Spanner is **99.97% garbage**: 5.4M dead rows plus 9.4M live rows with no TTL. - #333 Inject the store instead of using the `STORE` module global (test pollution; once broke 136 tests). - #128 Hot-tenant shard-0 counter-row lock contention (root cause of settle latency). - #850 Deploy pipeline starves under merge pressure (each run defers to a newer main and still reports success). - #1178 Repository security coverage gap. #1720 CI Postgres image rate-limited by ECR. - Freshness alarms left open: #817 ClickHouse archive, #638 AWS-EU analytics drain, #630 client telemetry. - Multi-cloud lag: #1272 retirements not yet promoted to AWS/Azure; #1825 Azure peer-status stall.

6. Incidents (25 write-ups, 2026-09-05 → 2026-10-18)

CLUSTER	CO UN T	EXAMPLES	PATTERN
Spanner contention / CPU	7	09-06 operator <code>LIKE</code> searches on <code>tr_entities.body</code> . 09-08 legacy billing shard contention. 09-24 overrun rollout scan. 09-25 credit rebalance convoy. 09-26 regional ledger grant storm (94–96% aborts, authorize p50 0.1 s → 3.7 s fleet-wide). 10-11 commit-latency minute	Hot rows, unbounded scans, retry amplification
Async-settle shadow	8	10-10 regional read deadline, SDK reset, shutdown, source revision, Spanner budget. 10-11 budget follow-up. 10-13 missing health alerts. 10-18 settlelog shutdown	The evidence/shadow system itself produces incidents
Settlement correctness	3	09-05 shared trace ID blocked multi-call settle. 09-30 provider service-tier settlement. 10-04 starter-credit fragmentation	Edge cases in money accounting
Provider / config	3	10-03 provider configuration. 10-05 provider recovery. 10-04 gateway authorization	Upstream variance
Platform	4	09-24 us-east4 Cloud Run start stall (33/42 instances failed probes). 10-11 OAuth invalid code → 500. 10-15 one container's Spanner <code>DeadlineExceeded</code> (unresolved). 10-18 fastpath lease lock contention	

Write-ups are thorough. They separate confirmed causes from open questions, keep tenant data out, and refuse to overclaim fixes. That is a strong engineering culture. The incident rate, about 1 every 2 days in the last 6 weeks, is the headline concern.

7. Design docs in flight (docs/design/ , 46 files)

PROGRAM	STATUS	DOCS
Async settle shadow v2	Amended 2026-10-11 → rounds 2-7. \$15.8 activation steps executing (step 5 deployed 10-19)	async-settle-shadow-v2*.md (14 files), async-settle-outbox-v1, durable-settle-outbox
Fast admission + batched settlement ("100T tokens/month")	Approved v25 → v42+. Go fastpath/ spike, TLA+ model-checked. Pilot and production-rollout docs exist	fast-admission-*.md (8 files)
Billing simplification	Retire JSON credit ledger, typed counters, dual-system deletion, credit and key-usage row sharding	5 docs
Data/analytics	ClickHouse HA, request tagging, provider latency probe	
Trust / security	BYOK AAD v2 migration (with attestations JSON), trust-tier incremental	
Product	OAuth scopes and app economy, x402 stablecoin funding, Polyphemus 1.0, model capabilities, company affiliations	
Ops	Two-cloud rollouts, catalog refreshes, provider retirements	

8. Synthesis

Architecture strengths

- Verifiable trust boundary.** The prompt path (`api.trustedrouter.com`) terminates TLS inside attested enclaves (GCP Confidential Space, AWS Nitro, Azure SEV-SNP/MAA). The control plane never sees prompts, and the public repo matches the measured image digest. That is a real, demonstrable differentiator.
- Storage behind a Protocol, with an executable contract.** `tests/conformance/` runs memory, Postgres and Spanner-PG. One `PostgresStore` covers Aurora DSQL, Cosmos/Citus and Spanner-PG.
- Multi-cloud as independent deployments.** Identity federates but credits do not, which avoids a cross-cloud ledger.
- Formal rigor on money.** TLA+ specs, differential `proof_oracle` tests and conformance tests are required for any reserve/settle/refund change.
- Ops as code.** Env vars live in `rollout.sh`, Terraform covers Spanner and Cloud Armor, prod smoke runs on every push, and a trust-drift workflow checks the published image.
- Analytics separated from OLTP.** ClickHouse (3-zone, Keeper, Parquet archive with 7-year retention) offloads Spanner.

Tech debt and risk register

#	RISK	LIKE LIHO OD	IMP ACT	NOTES / MITIGATION
R 1	Bus factor = 1 (founder holds design authority, credentials, deploy habit)	High	Critical	Document credential custody, add a second on-call and approver for money code, keep a break-glass runbook
R 2	Money-path complexity: two settle systems (legacy + async shadow), 110 shadow files, fastpath in Go	High	High	Freeze new shadow surface area. Set a dated exit criterion to cut over or delete. Cap evidence code
R 3	Spanner hot-row contention / retry storms (7 incidents, #128, #334 garbage)	High	High	Land row sharding and the fast-admission leases. Add TTL/GC for dead rows. Enforce bounded-query linting

#	RISK	LIKE LIHO OD	IMP ACT	NOTES / MITIGATION
R 4	Ship-on-merge with 400+ PRs/month, mostly agent-authored	High	High	#850 shows the deploy pipeline can report success it did not achieve. Add canary/bake gates and protect money files with CODEOWNERS
R 5	Spoofable client IP for rate limits (#402)	Medium	Medium	Trust only the LB-appended XFF hop
R 6	Multi-cloud drift (AWS/Azure lag on retirements; Azure peer stall; AWS-EU analytics drain)	Medium	Medium	Single release train across clouds. Per-cloud freshness SLOs
R 7	Cost sprawl: Spanner Enterprise Plus nam6 (1k–10k PU autoscale, 5 replica locations), 3-zone ClickHouse xclouds, Bigtable, enclaves on 3 clouds, Cloudflare	Medium	High	FinOps review. The nam6→nam7 option is documented. Consider whether Azure/AWS traffic justifies always-on footprints
R 8	God modules (<code>gateway.py</code> 6k, <code>dashboard.py</code> 6.6k, <code>storage_*</code> 7–8k) with highest churn	High	Medium	Split along the Store Protocol and route boundaries. Fix the <code>STORE</code> global (#333)
R 9	Catalog/pricing correctness at scale (92 providers, hourly bot to main, held-price issues #1383, Perplexity fixed-fee #1700)	Medium	Medium	Billing mispricing is direct money loss. Keep the hold-for-review gate
R 10	Dependency rot (10 dependabot PRs open since August)	Medium	Low / Medium	Batch-merge weekly
R 11	Unresolved prod anomalies (10-15 container DeadlineExceeded; EU ClickHouse disconnects #1654)	Medium	Medium	Track to closure
R 12	License/legal (BUSL may deter OSS contribution; the claim boundary must stay honest)	Low	Medium	Keep the "never claim" list from founder interview workflow

Cost drivers (qualitative; no billing data was queried)

- **Spanner:** Enterprise Plus, nam6 multi-region, 5 replica locations, autoscaled 1,000–10,000 PU (floor raised from 400→600→1,000 after the September CPU incidents). Likely the largest fixed line. Issue #334 says 99.97% of rows are garbage, so storage cost is inflated.
- **ClickHouse:** three-zone replicated cluster with 500 GB SSD per replica on GCP, plus per-cloud instances and a GCS Parquet archive.
- **Compute:** Cloud Run in several regions with warm minimum instances. Confidential Space VMs. AWS Nitro + Fargate. Azure SEV-SNP + Container Apps.
- **Bigtable** (regional quota, analytics legacy) still runs alongside.
- **Upstream inference:** pass-through COGS, offset by routing markup.

Strategic bets

1. **Attestation as the moat.** The "OpenRouter-compatible, but verifiable" wedge, open BUSL code, nonce-challenged attestation, and the "Attestation is all you need" thesis. Defensible because competitors (OpenRouter, Portkey, Cloudflare AI GW, LiteLLM) cannot easily retrofit an enclave-terminated prompt path.
2. **Confidential/E2EE pools** (`min_privacy: confidential` , fail-closed) and ZDR pools. Privacy is the product, not a feature flag.
3. **Token Exchange** (enterprise and regional landing sites). Moves up-market toward a compute marketplace.
4. **Alternative payment rails:** x402 stablecoins, Lightning (LightningRouter), Adyen/PayPal/Stripe. These reach agents and crypto-native buyers.
5. **SDK and app ecosystem:** JS (`@lore-hex/trusted-router`) and Python (`trusted-router-py`) SDKs, "Sign in with TrustedRouter", OAuth scopes and app economy, a one-prompt migration for coding agents.

6. **Scale bet:** fast admission and batched settlement designed for **100T tokens/month**. This builds for a scale far beyond current load and must be paced against R2.
7. **Breadth:** 92 provider catalogs, image/video/TTS/transcription, task models (ScaleDown), Polyphemus.

9. 30/60/90-day plan for a new technical leader

Days 0–30: Learn, stabilize, de-risk the person

- Read `AGENTS.md`, `docs/storage-portability/HANDOFF.md`, `docs/spanner-reliability.md`, the 25 incident docs, and both in-flight design programs.
- Run the three CI gates and the conformance suites locally. Shadow one full deploy and one incident.
- **Bus-factor actions:** inventory every credential and its owner. Get second-person access to GCP, AWS, Azure, Cloudflare, Stripe and the domain registrars. Write a break-glass runbook.
- Put money-path files under CODEOWNERS requiring a human approver (gateway, storage, settle, billing).
- Close the quick security debt: #402 XFF spoofing, #1178. Merge the dependabot backlog.
- Baseline metrics: incident rate, authorize p50/p99, settle error rate, Spanner CPU and aborts, cloud spend by line item.

Days 31–60: Reduce risk in the money path and the database

- With Joseph, set a **written exit criterion and date for async-settle shadow v2**: cut over or delete. Freeze new evidence-machinery surface area.
- Land row sharding (#128, credit/key-usage sharding designs) and Spanner garbage cleanup with TTL (#334). Target zero Spanner-contention incidents for 30 days.
- Fix deploy-pipeline honesty (#850) and add a bake/canary stage for gateway changes.
- Start decomposing `gateway.py`, `dashboard.py` and the `STORE` global (#333) behind the existing Protocol.
- FinOps pass: Spanner PU floor vs. actual load, ClickHouse sizing per cloud, warm-instance counts, the value of Bigtable after the ClickHouse cutover.

Days 61–90: Scale the team and the bets

- Hire or contract one senior backend/SRE engineer for the money path and on-call. Set up a two-person on-call rotation.
- Define the agent-workforce policy: which paths agents may merge unreviewed, and which need human differential review.
- Decide fast-admission rollout pacing against real traffic growth, not the 100T design ceiling.
- Publish SLOs per cloud (availability, attestation freshness, analytics freshness) and align AWS/Azure on the same release train.
- Strategy review with the founder: rank Token Exchange, payment rails, the SDK ecosystem and provider breadth by revenue evidence from the CRM loop, then cut or defer the weakest.

Method notes

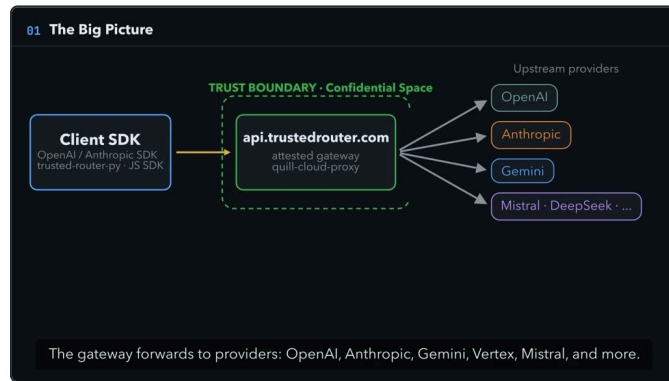
- Commit counts come from `git log` on `main` (all history). Merged PR counts come from `gh pr list --state merged`. Emails are redacted and only names are shown.
- LOC comes from `wc -l` on tracked files (no cloc/tokei available), so it includes blank and comment lines.
- Churn is the number of commits touching a file since 2026-07-21. The "human" view excludes the pricing bot and `tests/`.

- Cost statements are qualitative, from repo docs and Terraform. No billing or production data was queried.

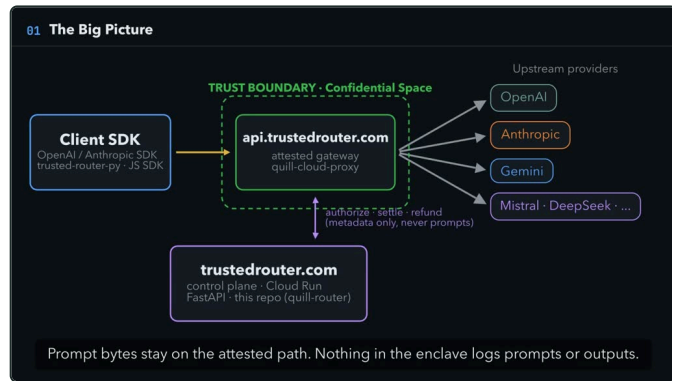
Storyboards

The fully composed frames from each explainer video, taken at the end of every animation step. The MP4s are in `site/videos/` and the Manim sources in `manim/`.

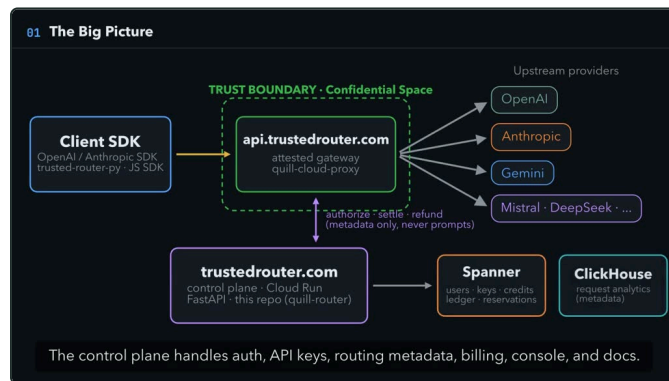
Video 01 · The big picture



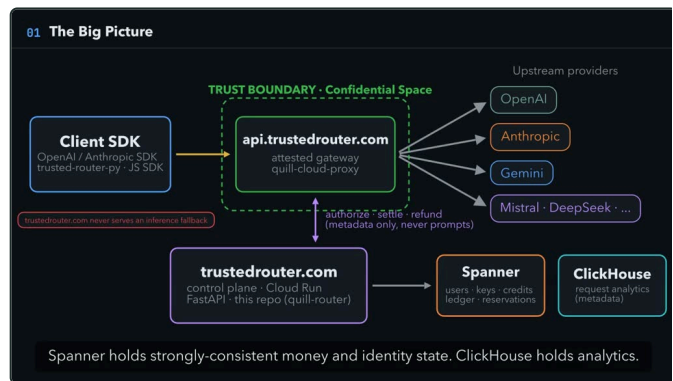
The big picture · 0:27 · held 6.3s



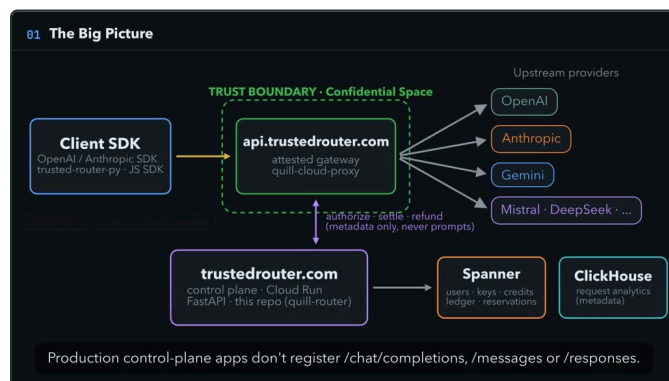
The big picture · 0:33 · held 5.0s



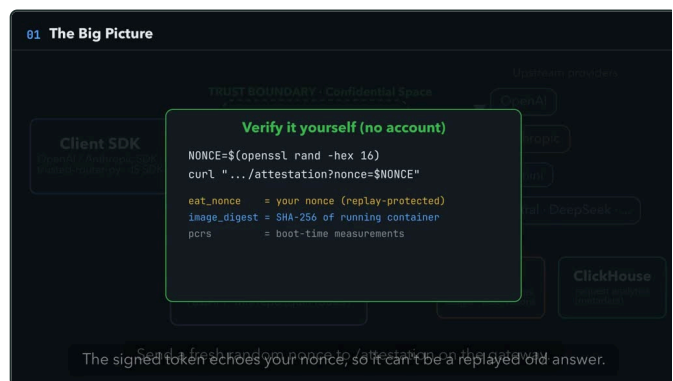
The big picture · 0:38 · held 4.3s



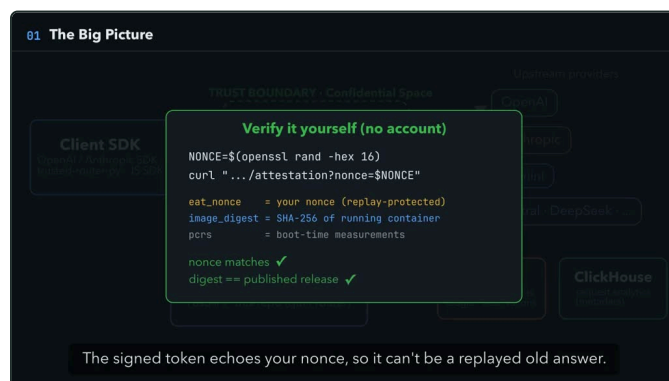
The big picture · 0:42 · held 4.0s



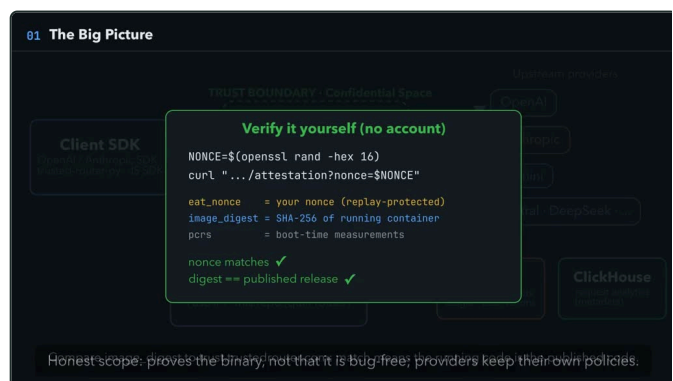
The big picture · 0:47 · held 4.0s



The big picture · 0:55 · held 8.1s



The big picture · 0:59 · held 4.0s



The big picture · 1:04 · held 4.0s

Video 02 · Life of a request

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

Authorization: Bearer sk-tr-v1-.....

salted SHA-256 → lookup hash

point read
opaque key id → workspace, limits

Raw keys are never stored. Verification never scans keys.

The gateway receives the request and hashes the presented API key.

Life of a request · 0:17 · held 2.6s

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

```
{
  "model": "anthropic/claude-sonnet-4.6",
  "provider": {
    "min_privacy": "confidential",
    "allow_fallbacks": true
  }
}
```

zdr · zero-retention floor

confidential (e2e) · confidential compute + E2EE

✓ route A satisfies floor
✗ route B weaker privacy, filtered out
✗ none left? fail closed, never downgrade

Routing honors model, models[] and provider filters (order, only, ignore, sort...).

Life of a request · 0:26 · held 2.6s

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

POST /v1/internal/gateway/authorize

workspace balance
12,000,000 µ\$ - reservation (hold)
40,000 µ\$

16 billing shards per workspace / key: no single hot row

Integer microdollars, never floats. Key limits checked here too.

authorize validates the key hash and places a hold on credits and key limits.

Life of a request · 0:37 · held 2.6s

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

client ← gateway ← provider

SSE enclave selected route

```
data: {"choices":[{"delta":{"content":"lol"}}]}
data: {"choices":[{"delta":{"content":"lo"}}]}
data: {...,"usage":{"prompt_tokens":812,"completion_tokens":96}}
data: [DONE]
```

Provider error before output → next candidate, or refund the hold.

Inside the enclave, the gateway calls the provider and streams SSE back to the client.

Life of a request · 0:46 · held 2.6s

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

settle commit → analytics outbox

Spanner transaction row enqueued

Happy path: enqueue rides inside the one-commit settle.

Any failure: durable two-commit settle, enqueue after the commit.

Best-effort by design: analytics is never worth destabilising money code.

If anything goes wrong, settlement still lands durably; analytics catches up later.

Life of a request · 1:11 · held 2.6s

02 Life of a Chat Completion

- 1 Authenticate key
- 2 Route by model + privacy
- 3 Reserve credits
- 4 Stream from provider
- 5 Settle
- 6 Metadata-only usage
- 7 Analytics outbox
- 8 ClickHouse

outbox → drainer → ClickHouse

Spanner async batches request analytics

Activity pages, dashboards, investigations: query ClickHouse.

Never scan production Spanner tr_entities for analytics.

For request analytics, query ClickHouse, not production Spanner.

Life of a request · 1:20 · held 2.6s

02 Life of a Chat Completion

Key takeaways

- Keys: salted SHA-256 + lookup hash → point read, never a scan
- min_privacy zdr / confidential is a hard floor; fail closed, never downgrade
- authorize reserves → stream → settle charges actual → refund on failure
- Usage is metadata only; prompts and outputs are never logged
- Analytics: best-effort outbox → ClickHouse (not Spanner scans)

Life of a request · 1:29 · held 5.5s

02 Life of a Chat Completion

Key takeaways

- Keys: salted SHA-256 + lookup hash → point read, never a scan
- min_privacy zdr / confidential is a hard floor; fail closed, never downgrade
- authorize reserves → stream → settle charges actual → refund on failure
- Usage is metadata only; prompts and outputs are never logged
- Analytics: best-effort outbox → ClickHouse (not Spanner scans)

Life of a request · 1:36 · held 6.5s

Video 03 · Day-1 setup

03 Day 1: Local Setup

```
zsh - ~/code

$ curl -LsSf https://astral.sh/uv/install.sh | sh # or: brew install uv
installing to ~/.local/bin ... everything's installed!
$ git clone git@github.com:Lone-Hex/quill-router.git && cd quill-router
Cloning into 'quill-router'...
```

Step 1: install uv. It manages Python (3.12 pinned) and all dependencies.

Day-1 setup · 0:14 · held 7.93s

03 Day 1: Local Setup

```
zsh - ~/code

$ uv sync
Using CPython 3.12 (.python-version)
Creating virtual environment at: .venv
Installed fastapi, google-cloud-spanner, stripe, pytest, ruff, mypy, ...
$ uv run pytest
```

Step 3: run the test suite. The default selection matches CI's release jobs.

Day-1 setup · 0:19 · held 5.27s

03 Day 1: Local Setup

```
zsh - ~/code

$ uv sync
Using CPython 3.12 (.python-version)
Creating virtual environment at: .venv
Installed fastapi, google-cloud-spanner, stripe, pytest, ruff, mypy, ...
$ uv run pytest # addopts: -m 'not provider_health and not proof_oracle'
collected ... items / ... deselected
tests/test_auth_and_routing.py ..... [ 12%]
tests/test_billing_key_window_limits.py ..... [ 31%]
tests/conformance/ ...ssss..... [ 58%]
..... [100%]
===== passed, skipped =====
```

proof_oracle (over an hour) and provider_health (live state) run in dedicated CI jobs.

Day-1 setup · 0:25 · held 3.0s

03 Day 1: Local Setup

```
zsh - ~/code

$ uv run uvicorn trusted_router.main:app --reload
INFO: Will watch for changes in these directories: ['.../quill-router']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Application startup complete.
```

Step 4: run the FastAPI app with auto-reload. Defaults are local and offline.

Day-1 setup · 0:31 · held 3.6s

03 Day 1: Local Setup

```
zsh - ~/code

$ uv run uvicorn trusted_router.main:app --reload
INFO: Will watch for changes in these directories: ['.../quill-router']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Application startup complete.
```

Step 5: in a second terminal, run the end-to-end smoke script against it.

Day-1 setup · 0:35 · held 3.5s

03 Day 1: Local Setup

```
zsh - ~/code

$ uv run uvicorn trusted_router.main:app --reload
INFO: Will watch for changes in these directories: ['.../quill-router']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Application startup complete.
$ TR_SMOKE_BASE_URL=http://127.0.0.1:8000/v1 uv run python scripts/smoke_e2e.py
E2E smoke OK
```

Step 5: in a second terminal, run the end-to-end smoke script against it.

Day-1 setup · 0:39 · held 3.5s

03 Day 1: Local Setup

```
zsh - ~/code

$ docker run -d --rm --name tr-conformance-pg -e POSTGRES_PASSWORD=tr \
-p 5433:5432 postgres:17
$ export TR_CONFORMANCE_POSTGRES_DSN=\
postgres://tr:tr@localhost:5433/tr-conformance-pg
```

Optional: real database backends for the Store conformance suite. No cloud credentials.

Day-1 setup · 0:48 · held 5.2s

03 Day 1: Local Setup

```
.env
repo root · copy of .env.example · local only

TR_ENVIRONMENT=local
TR_STORAGE_BACKEND=memory
TR_ENABLE_LIVE_PROVIDERS=false
TR_LOCAL_KEYS_FILE=~/.quill_cloud_keys.private
TR_RATE_LIMIT_ENABLED=true
TR_STRIPE_SECRET_KEY= # empty → mock
TR_INTERNAL_GATEWAY_TOKEN=

~/.quill_cloud_keys.private
operator keys file · dotenv style · NEVER committed

ANTHROPIC_API_KEY=...
OPENAI_API_KEY=...
GEMINI_API_KEY=...
DEEPSEEK_API_KEY=...
STRIPE_SECRET_KEY=...
STRIPE_WEBHOOK_SECRET=...
SENTRY_DSN=...
```

Settings use the TR_ prefix and read .env. Copy .env.example to start.

Day-1 setup · 1:08 · held 8.9s

Video 04 · Config & environment

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
src/trusted_router/config.py
class Settings(BaseSettings):
    model_config = SettingsConfigDict(
        env_prefix="TR_", env_file=".env"
    )
    local_keys_file: Path = Path(
        "~/quill_cloud_keys.private"
    )

    # priority, highest first:
    # env vars > .env > keys file > defaults

Code defaults are safe for local dev. The TR_ prefix maps env vars to fields.
```

Config & environment · 0:16 · held 3.0s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
src/trusted_router/config.py
class Settings(BaseSettings):
    model_config = SettingsConfigDict(
        env_prefix="TR_", env_file=".env"
    )
    local_keys_file: Path = Path(
        "~/quill_cloud_keys.private"
    )

    # priority, highest first:
    # env vars > .env > keys file > defaults

Production validation fails closed: no gateway token, webhook secret or real backend → refuse to boot.
```

Config & environment · 0:19 · held 3.2s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
.env (copied from .env.example)
TR_ENVIRONMENT=local
TR_STORAGE_BACKEND=memory
TR_ENABLE_LIVE_PROVIDERS=false
TR_LOCAL_KEYS_FILE=~/.quill_cloud_keys.private
TR_RATE_LIMIT_IP_PER_WINDOW=240

# ~/.quill_cloud_keys.private (never committed)
ANTHROPIC_API_KEY=... OPENAI_API_KEY=...
```

Locally, .env overrides defaults; the operator keys file fills in provider credentials.

Config & environment · 0:25 · held 3.0s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
ENV_VARS=(
    "TR_ENVIRONMENT=production"
    "TR_STORAGE_BACKEND=${STORAGE_BACKEND}"
    "TR_API_BASE_URL=https://api.trustedrouter.com/v1"
    "TR_SPANNER_INSTANCE_ID=${SPANNER_INSTANCE_ID}"
    "TR_REGIONS=${TR_REGIONS}"
    "TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS=300000"
    "TR_RELEASE=${RELEASE_ID}" ...
)
gcloud run deploy ... --set-env-vars "${SET_ENV_VARS}..."

Production runtime config is config-as-code: the ENV_VARS array in rollout.sh.
```

Config & environment · 0:33 · held 3.0s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
scripts/deploy/rollout.sh
ENV_VARS=(
    "TR_ENVIRONMENT=production"
    "TR_STORAGE_BACKEND=${STORAGE_BACKEND}"
    "TR_API_BASE_URL=https://api.trustedrouter.com/v1"
    "TR_SPANNER_INSTANCE_ID=${SPANNER_INSTANCE_ID}"
    "TR_REGIONS=${TR_REGIONS}"
    "TR_SIGNUP_TRIAL_CREDIT_MICRODOLLARS=300000"
    "TR_RELEASE=${RELEASE_ID}" ...
)
gcloud run deploy ... --set-env-vars "${SET_ENV_VARS}..."

$ gcloud --up next rollout overwrites it. -router \

...it's a no-op. The next deploy replaces it. Change ENV_VARS in a PR instead.
```

Config & environment · 0:45 · held 3.0s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
rollout.sh SECRET_ENVS → --update-secrets
"TR_STRIPE_SECRET_KEY=trustedrouter-stripe-secret-key:latest"
"TR_STRIPE_WEBHOOK_SECRET=trustedrouter-stripe-webhook-secret:latest"
"TR_INTERNAL_GATEWAY_TOKEN=trustedrouter-internal-gateway-token:latest"
"TR_SENTRY_DSN=trustedrouter-sentry-dsn:latest"
"/secrets/async-settle/ticket.pem..." # mounted as a file

# Cloud KMS
"TR_BYOK_KMS_KEY_NAME=${BYOK_KMS_KEY_NAME}"
# wraps each BYOK key's DEK (envelope encryption)
```

Secrets never sit in ENV_VARS: Cloud Run reads them from Secret Manager by name.

Config & environment · 0:52 · held 3.0s

04 Config & Environment

PRODUCTION

- Secret Manager / Cloud KMS
prod secrets · BYOK envelope keys
- scripts/deploy/rollout.sh ENV_VARS
prod config-as-code
- local .env + operator keys file
.env · ~/.quill_cloud_keys.private
- Code defaults
src/trusted_router/config.py Settings

LOCAL

```
rollout.sh SECRET_ENVS → --update-secrets
"TR_STRIPE_SECRET_KEY=trustedrouter-stripe-secret-key:latest"
"TR_STRIPE_WEBHOOK_SECRET=trustedrouter-stripe-webhook-secret:latest"
"TR_INTERNAL_GATEWAY_TOKEN=trustedrouter-internal-gateway-token:latest"
"TR_SENTRY_DSN=trustedrouter-sentry-dsn:latest"
"/secrets/async-settle/ticket.pem..." # mounted as a file

# Cloud KMS
"TR_BYOK_KMS_KEY_NAME=${BYOK_KMS_KEY_NAME}"
# wraps each BYOK key's DEK (envelope encryption)

User BYOK keys are envelope-encrypted rows; Cloud KMS wraps each per-key DEK.
```

Config & environment · 0:55 · held 3.0s

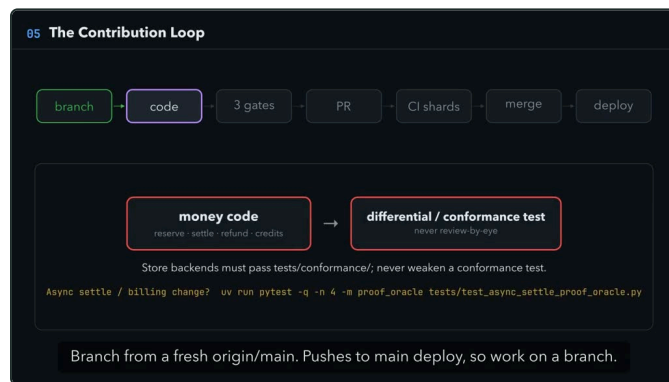
04 Config & Environment

Key takeaways

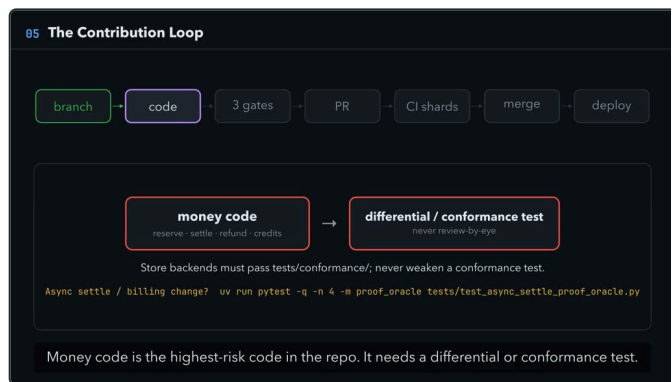
- One Settings class, TR_ prefix: env vars > .env > keys file > defaults
- Local: .env (from .env.example) + ~/.quill_cloud_keys.private, never committed
- Prod config = ENV_VARS in scripts/deploy/rollout.sh (config-as-code)
- Hand-set Cloud Run env vars are overwritten by the next rollout
- Secrets: Secret Manager by name; BYOK keys wrapped by Cloud KMS

Config & environment · 1:05 · held 5.03s

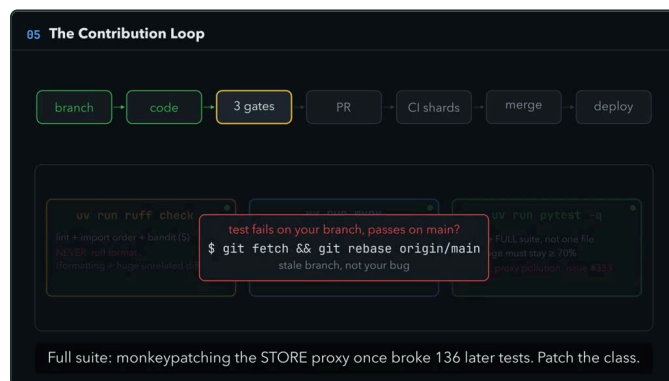
Video 05 · The contribution loop



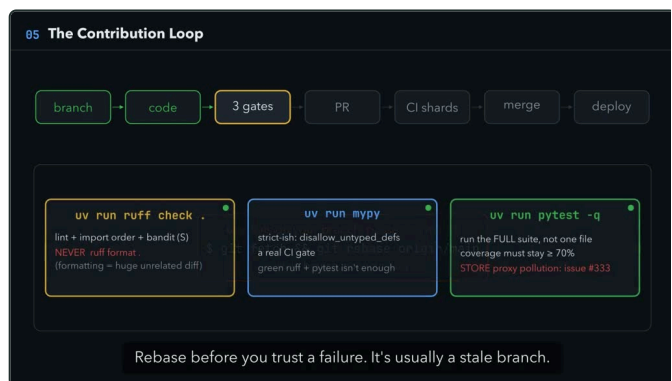
The contribution loop · 0:18 · held 7.53s



The contribution loop · 0:22 · held 3.2s



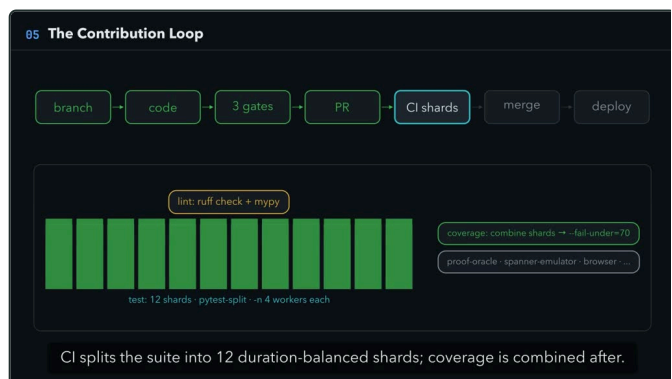
The contribution loop · 0:43 · held 3.8s



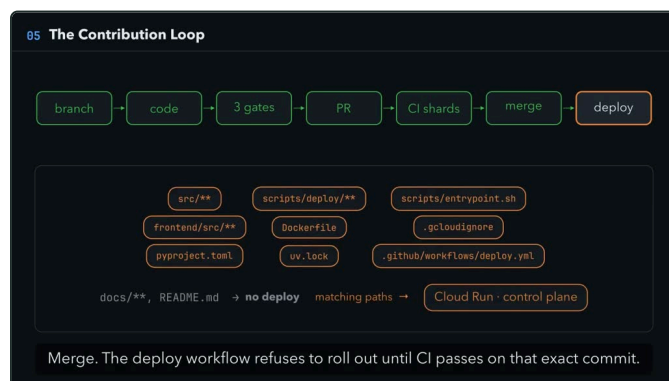
The contribution loop · 0:47 · held 3.8s



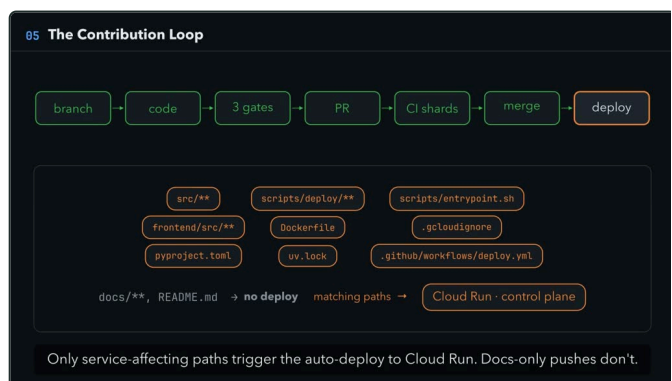
The contribution loop · 0:57 · held 9.6s



The contribution loop · 1:00 · held 3.0s

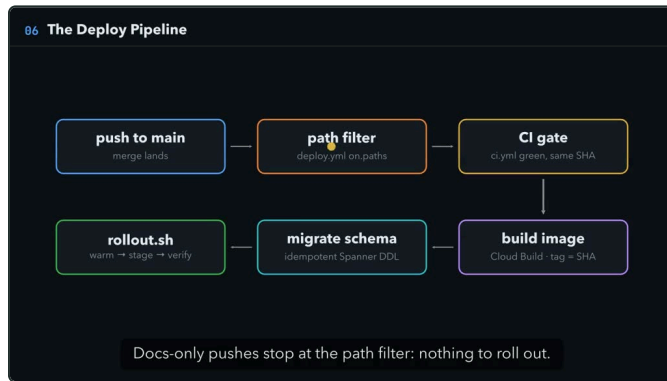


The contribution loop · 1:10 · held 4.8s

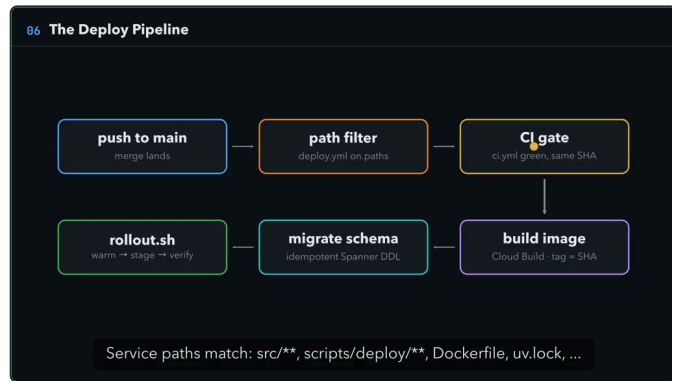


The contribution loop · 1:14 · held 3.2s

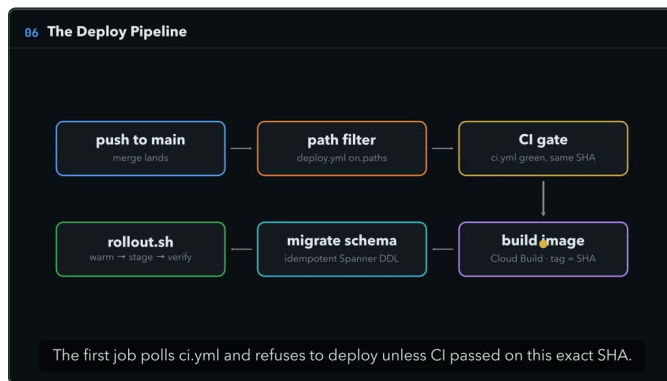
Video 06 · The deploy pipeline



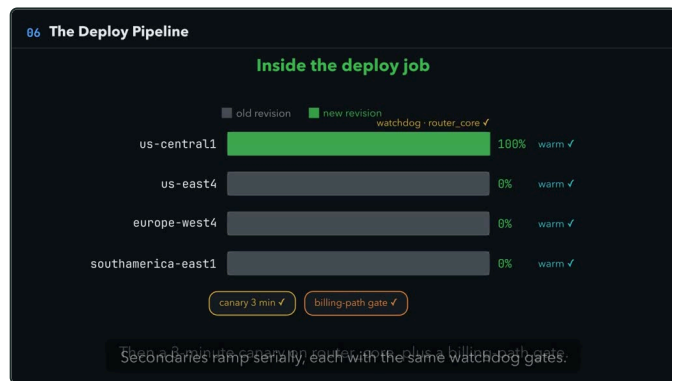
The deploy pipeline · 0:21 · held 5.9s



The deploy pipeline · 0:25 · held 2.8s



The deploy pipeline · 0:29 · held 2.8s



The deploy pipeline · 1:03 · held 26.3s

06 The Deploy Pipeline

Two rules that protect production Spanner

No big unbatched DML during rollout

Old and new revisions serve at the same time for the whole staged ramp.
A large unbatched UPDATE/DELETE contends with live billing traffic in every region.

✓ batch it, and run it outside a rolling deploy

Never full-scan tr_entities

No LIKE / regex / JSON predicates on body without a complete indexed key.
LIMIT 5 limits results, not rows scanned.
Three ad hoc Stripe-ID searches caused the September 6 CPU incident.

✓ inspect_payment_owner · ClickHouse · SPANNER_SYS

During a rolling deploy, never run large unbatched DML against production Spanner.

The deploy pipeline · 1:29 · held 25.4s

06 The Deploy Pipeline

Two rules that protect production Spanner

No big unbatched DML during rollout

Old and new revisions serve at the same time for the whole staged ramp.
A large unbatched UPDATE/DELETE contends with live billing traffic in every region.

✓ batch it, and run it outside a rolling deploy

Never full-scan tr_entities

No LIKE / regex / JSON predicates on body without a complete indexed key.
LIMIT 5 limits results, not rows scanned.
Three ad hoc Stripe-ID searches caused the September 6 CPU incident.

✓ inspect_payment_owner · ClickHouse · SPANNER_SYS

Read-only is not load-safe. Use python -m scripts.inspect_payment_owner for payments.

The deploy pipeline · 1:32 · held 3.0s

06 The Deploy Pipeline

Two rules that protect production Spanner

No big unbatched DML during rollout

Old and new revisions serve at the same time for the whole staged ramp.
A large unbatched UPDATE/DELETE contends with live billing traffic in every region.

✓ batch it, and run it outside a rolling deploy

Never full-scan tr_entities

No LIKE / regex / JSON predicates on body without a complete indexed key.
LIMIT 5 limits results, not rows scanned.
Three ad hoc Stripe-ID searches caused the September 6 CPU incident.

✓ inspect_payment_owner · ClickHouse · SPANNER_SYS

Use ClickHouse for request analytics. Never loosen the 45% CPU alarm to make a scan pass.

The deploy pipeline · 1:36 · held 3.0s

06 The Deploy Pipeline

Key takeaways

- Push to main + service paths → deploy.yml (docs-only never deploys)
- CI must be green on the same SHA; image tagged by commit; schema first
- rollout.sh: warm 4 regions at 0% → us-central1 10/50/100 → canary → secondaries
- Verify: prod smoke, prod-smoke.yml, trust-drift.yml before saying "deployed"
- No unbatched Spanner DML mid-rollout; never full-scan tr_entities

The deploy pipeline · 1:42 · held 5.17s

Video 07 · The product surface

07 The Product Surface

1 Who serves what

Attested gateway

api.trustedrouter.com · quill-cloud-proxy · Confidential Space

/v1/chat/completions

/v1/messages

/v1/responses

/v1/embeddings

/v1/images

/v1/videos

/v1/audio/speech

/v1/audio/transcriptions

/v1/tools/fetch

Batch API

Control plane

trustedrouter.com · this repo (quill-router)

Model catalog + capabilities

Pricing

Billing + credits

Account API (keys, workspaces)

Web UI + /console

In production, quill-router doesn't even register /chat/completions or /messages.

The product surface · 0:16 · held 4.2s

07 The Product Surface

2 Router aliases

"model": "trustedrouter/..."

trustedrouter/auto

Fallback across providers. NO implicit privacy floor.

trustedrouter/zdr

Hard floor: zero-data-retention routes only.

trustedrouter/e2e = /confidential

Hard floor: confidential compute + E2EE.

trustedrouter/auto maximises availability. It does not add a privacy floor.

The product surface · 0:36 · held 10.6s

07 The Product Surface

2 Router aliases

"model": "trustedrouter/..."

trustedrouter/auto

Fallback across providers. NO implicit privacy floor.

trustedrouter/zdr

Hard floor: zero-data-retention routes only.

trustedrouter/e2e = /confidential

Hard floor: confidential compute + E2EE.

trustedrouter/eu

EU-focused pool. Routing policy, not a residency guarantee.

/zdr and /e2e are hard floors: if no route qualifies, the request fails closed.

The product surface · 0:40 · held 3.6s

07 The Product Surface

2 Router aliases

"model": "trustedrouter/..."

trustedrouter/auto

Fallback across providers. NO implicit privacy floor.

trustedrouter/zdr

Hard floor: zero-data-retention routes only.

trustedrouter/e2e = /confidential

Hard floor: confidential compute + E2EE.

trustedrouter/eu

EU-focused pool. Routing policy, not a residency guarantee.

Combo models

Zeus

Prometheus

Plato

Socrates

model panel (parallel)

judge / synthesizer

versioned multi-model graphs

/eu prefers EU providers. Pair it with the EU gateway endpoint for in-region traffic.

The product surface · 0:44 · held 3.4s

07 The Product Surface

3 Providers + privacy flags

108

providers in PROVIDERS (catalog_data.py)

5 ZDR + confidential compute + E2EE

3 confidential compute, no E2EE

14 ZDR only

24 ZDR explicitly false

62 flags unset: assume it stores

108 providers. Each carries three privacy flags: ZDR, confidential compute, E2EE.

The product surface · 1:00 · held 8.57s

07 The Product Surface

3 Providers + privacy flags

108

providers in PROVIDERS (catalog_data.py)

5 ZDR + confidential compute + E2EE

3 confidential compute, no E2EE

14 ZDR only

24 ZDR explicitly false

62 flags unset: assume it stores

Only 5 set all three. Those are the routes /e2e can use.

The product surface · 1:05 · held 4.1s

07 The Product Surface

3 Providers + privacy flags

108

providers in PROVIDERS (catalog_data.py)

5 ZDR + confidential compute + E2EE

3 confidential compute, no E2EE

14 ZDR only

24 ZDR explicitly false

62 flags unset: assume it stores

Unset flags default to stores_content = True. Unknown means assume it stores.

The product surface · 1:10 · held 3.0s

07 The Product Surface

4 Web surfaces

FastAPI + Jinja2 server-rendered pages · thin compiled TypeScript · no SPA

/chat

No-signup playground

via /chat-proxy browser key

Playwright suite: tests-e2e/

/console

Server-rendered dashboard

keys, credits, activity, BYOK,

routing, custom models, earnings

/status /leaderboard

Live evidence pages

/benchmarks /fleet

/models /providers /compare

~60 SEO pages

Comparison + landing pages

/openrouter-alternative,

/no-log-llm-api, /hipaa-llm-api ...

/docs hub

Product docs + API reference

/llm.txt for agents

OpenAPI at /api/reference

Token Exchange

13 markets · 30 domains

static sites on GCS + Cloud CDN

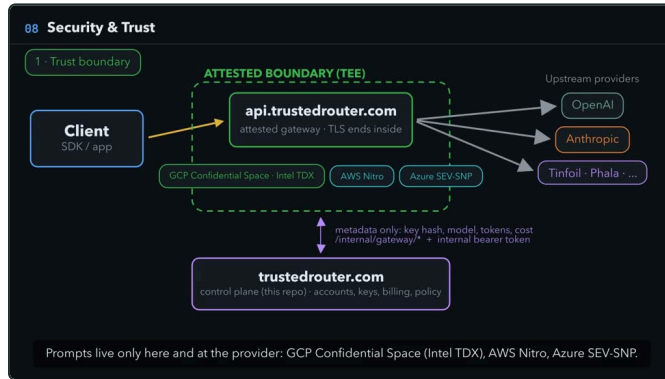
live evidence JSON from here

The product surface · 1:14 · held 3.6s

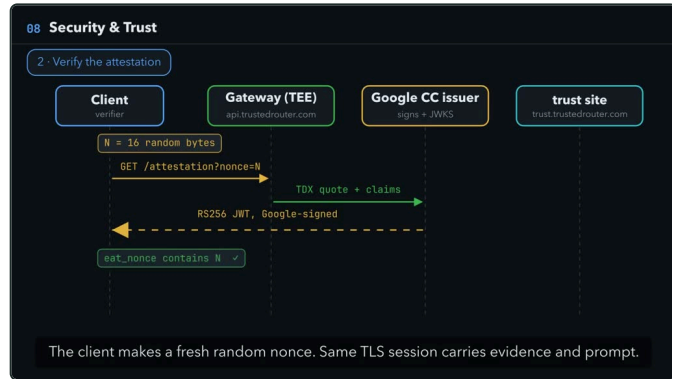
TrustedRouter Field Manual

179

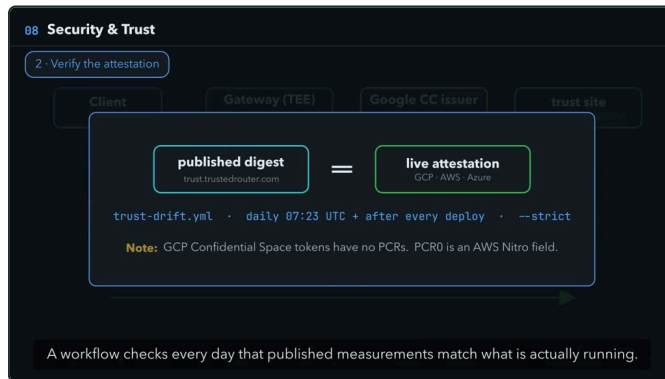
Video 08 · Security & trust



Security & trust · 0:20 · held 7.47s



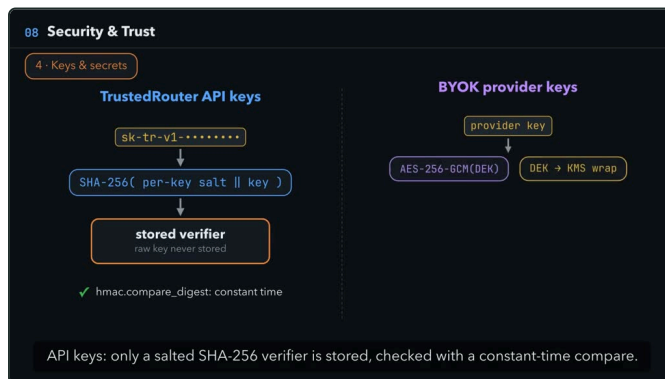
Security & trust · 0:36 · held 5.63s



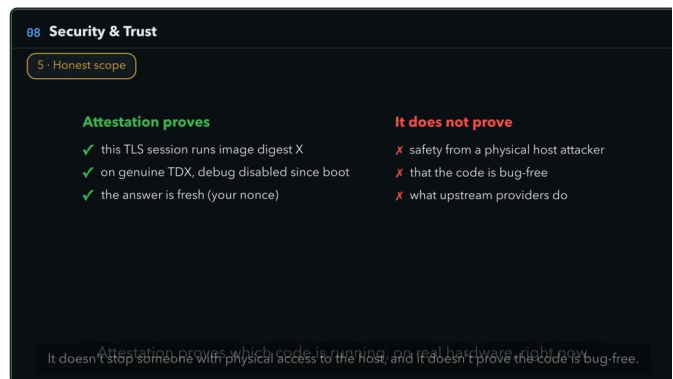
Security & trust · 0:48 · held 6.67s



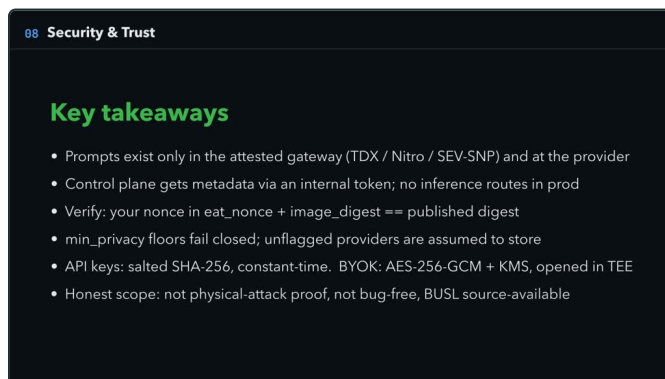
Security & trust · 1:01 · held 4.8s



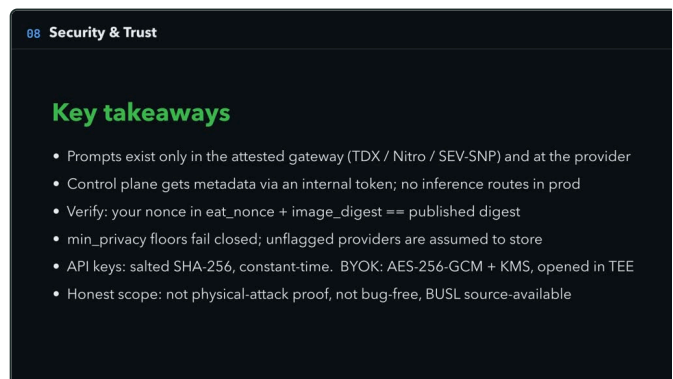
Security & trust · 1:14 · held 9.4s



Security & trust · 1:27 · held 6.9s

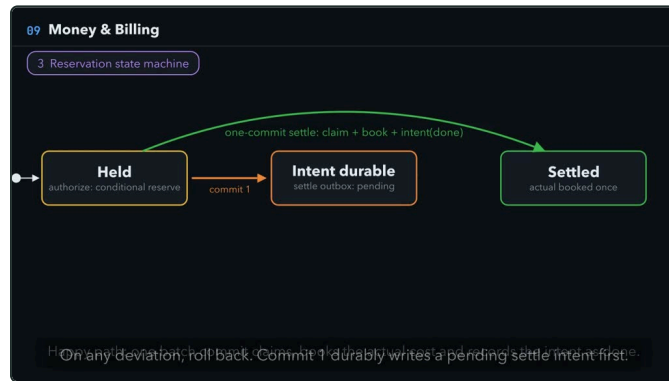


Security & trust · 1:40 · held 6.5s

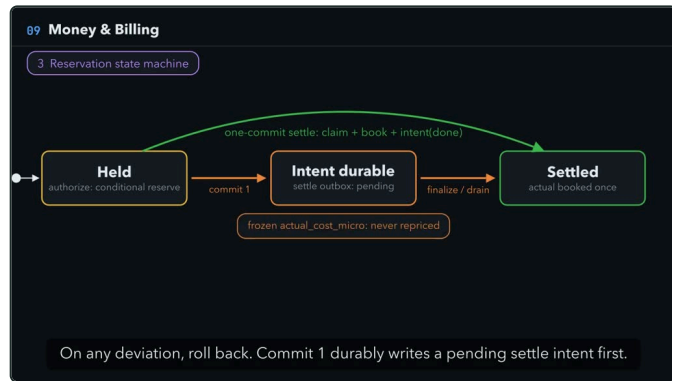


Security & trust · 1:47 · held 7.3s

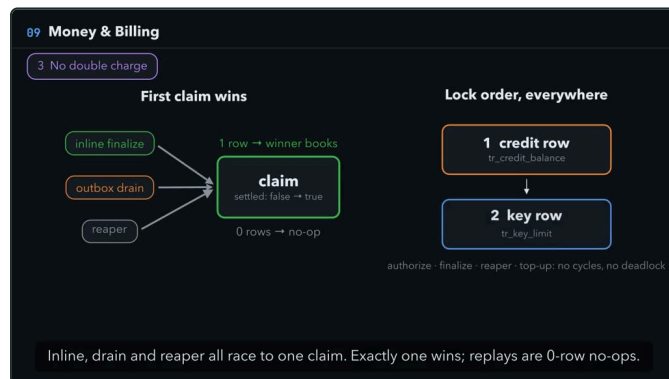
Video 09 · Money & billing



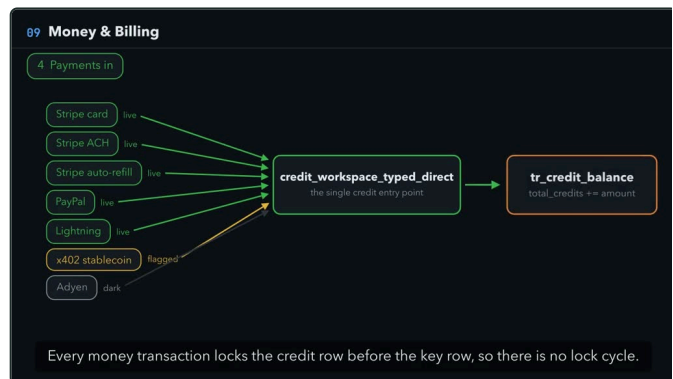
Money & billing · 0:37 · held 10.97s



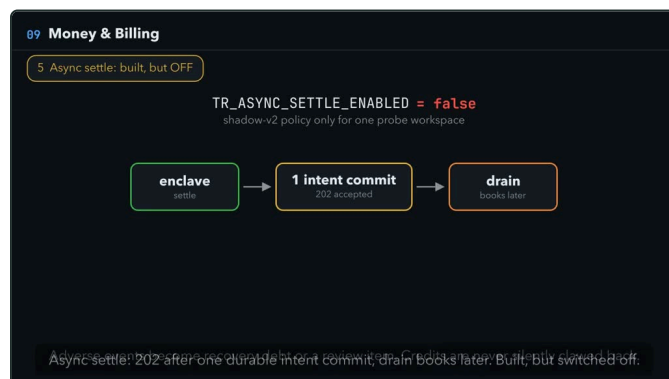
Money & billing · 0:42 · held 4.7s



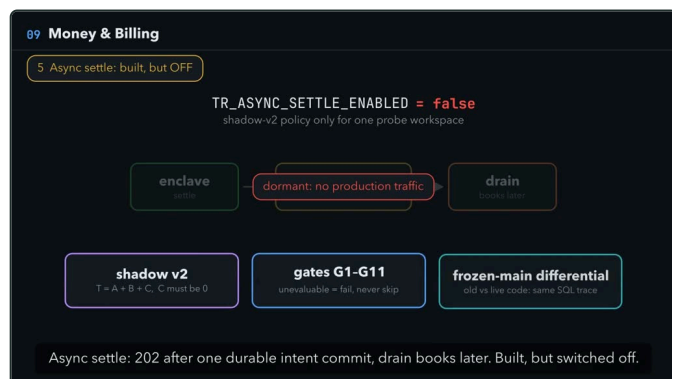
Money & billing · 0:59 · held 10.5s



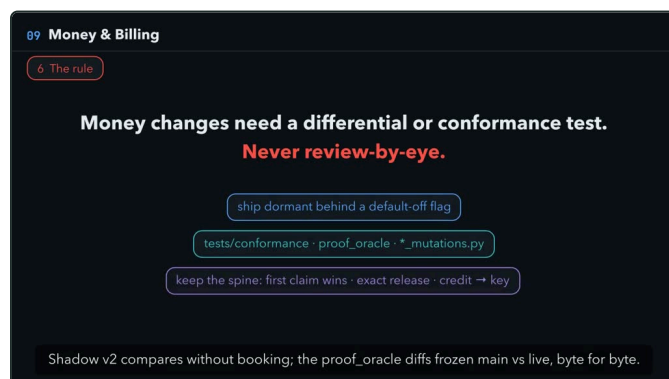
Money & billing · 1:06 · held 5.2s



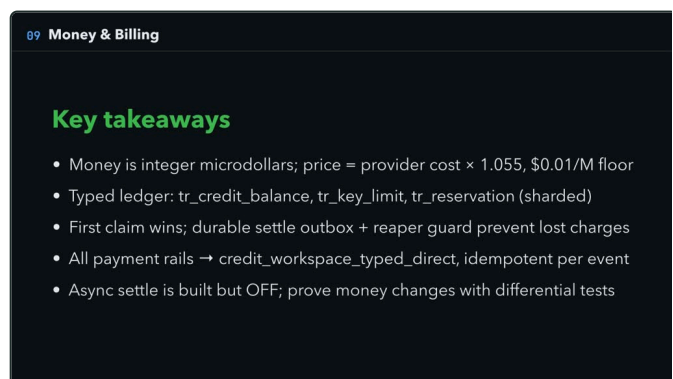
Money & billing · 1:22 · held 4.2s



Money & billing · 1:26 · held 4.1s



Money & billing · 1:32 · held 4.0s



Money & billing · 1:41 · held 5.9s

Video 10 · Data & storage

10 Data & Storage

2 The STORE proxy trap (#333)

instance attr

test file calls STORE.x()

STORE StoreProxy · __getattr__

old store torn down

X monkeypatch.setattr(STORE, "create_workspace", fake)

later tests in other files

...and teardown restores a method bound to a dead store. Every later test hits it.

Data & storage · 0:24 · held 6.8s

10 Data & Storage

3 tr_entities and the Sept 6 incident

SELECT body FROM tr_entities WHERE body LIKE '%pi_3Q-%' LIMIT 5

kind	id	body (JSON string)
workspace	ws_8f2...	{"name": "acme", ...}
api_key	k_41c...	{"ws": "ws_8f2", ...}
credit	cr_19a...	{"pi": "pi_3Q", ...}
stripe_event	evt_7d0...	{"type": "payment", ...}
reservation	rs_02b...	{"state": "dead", ...}
member	ws_8f2#u_1	{"role": "owner", ...}

14.8M rows

Spanner high-pri CPU

45% alarm

Sept 6: three 'harmless' operator searches for a Stripe PaymentIntent, each LIMIT 5.

Data & storage · 0:39 · held 14.63s

10 Data & Storage

3 tr_entities and the Sept 6 incident

LIMIT caps rows returned, not rows scanned

kind	id	body (JSON string)
workspace	ws_8f2...	{"name": "acme", ...}
api_key	k_41c...	{"ws": "ws_8f2", ...}
credit	cr_19a...	{"pi": "pi_3Q", ...}
stripe_event	evt_7d0...	{"type": "payment", ...}
reservation	rs_02b...	{"state": "dead", ...}
member	ws_8f2#u_1	{"role": "owner", ...}

14.8M rows

results: 0 of 5 scanned: all rows

61%

45% alarm

A missing substring scans the whole table: ~30 CPU-seconds each, CPU hit 61%.

Data & storage · 0:46 · held 4.83s

10 Data & Storage

4 Analytics pipeline

settle Spanner typed ledger commit

Spanner outboxes benchmark: 16 shards operational: 32 shards

drain ingest workers publisher node

ClickHouse replica 1 us-central1-a replica 2 us-central1-b replica 3 us-central1-c ReplacingMergeTree - FINAL reads

A drain daemon on the ClickHouse publisher node reads each shard and inserts.

Data & storage · 0:57 · held 8.13s

10 Data & Storage

4 Analytics pipeline

settle Spanner typed ledger commit

Spanner outboxes benchmark: 16 shards operational: 32 shards

drain ingest workers publisher node

ClickHouse replica 1 us-central1-a replica 2 us-central1-b replica 3 us-central1-c ReplacingMergeTree - FINAL reads

rollups: recompute → replace partition

A crash before the delete replays the batch; ReplacingMergeTree + FINAL dedupes it.

Data & storage · 1:06 · held 8.33s

10 Data & Storage

4 Analytics pipeline

settle Spanner typed ledger commit

Spanner outboxes benchmark: 16 shards operational: 32 shards

drain ingest workers publisher node

ClickHouse replica 1 us-central1-a replica 2 us-central1-b replica 3 us-central1-c ReplacingMergeTree - FINAL reads

rollups: recompute → replace partition

/status.json analytics.drain_lag_seconds

~~Single-tenant analytics~~ retired Sept 28-29

Data & storage · 1:14 · held 7.77s

10 Data & Storage

5 Multi-cloud separation

Identity issuers Google / GitHub OAuth · signed assertions

GCP control plane Cloud Run database Spanner · SpannerStore analytics ClickHouse 3 replicas

AWS control plane ECS Fargate database Aurora MySQL · PostgresStore analytics ClickHouse Para

Azure control plane Container Apps database Flexible Server · PostgresStore analytics ClickHouse node

Credits, keys, workspaces and analytics never cross: balances are shown per cloud.

Data & storage · 1:23 · held 6.9s

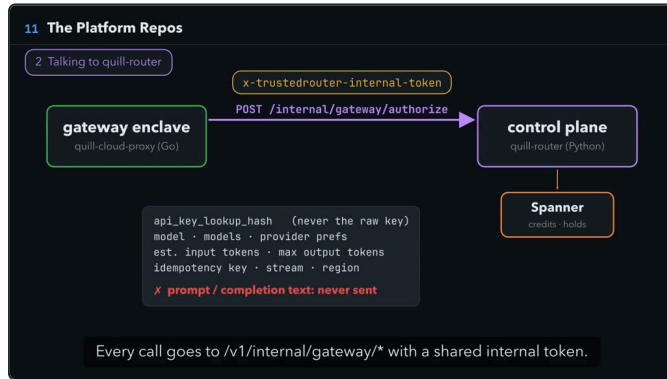
10 Data & Storage

Key takeaways

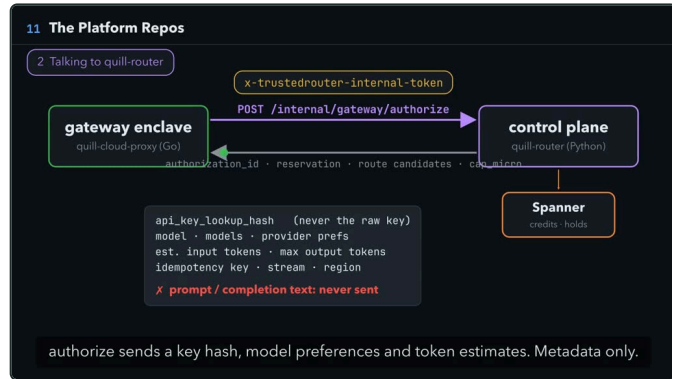
- Code talks to the Store Protocol; tests/conformance is the contract
- Never monkeypatch STORE: patch the backend class (#333)
- tr_entities.body is unindexed: LIMIT does not bound the scan
- Outbox rows are deleted only after the ClickHouse ack; reads use FINAL
- Each cloud is standalone: identity federates, credits do not

Data & storage · 1:39 · held 14.63s

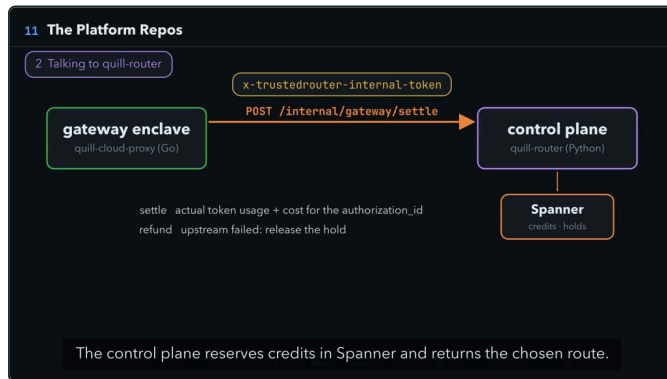
Video 11 · The platform repos



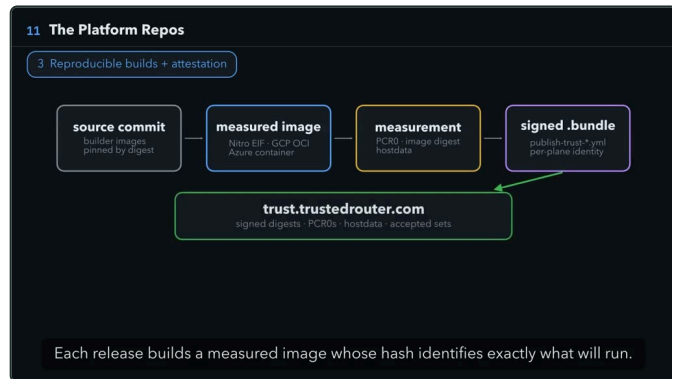
The platform repos · 0:23 · held 7.5s



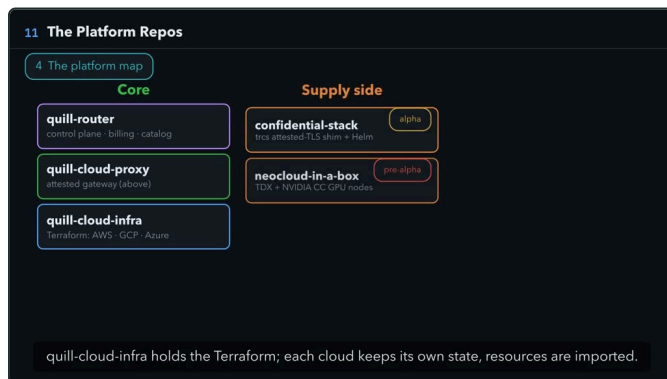
The platform repos · 0:28 · held 4.77s



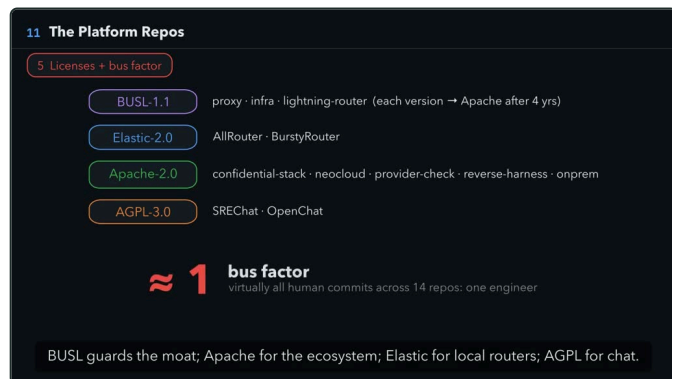
The platform repos · 0:34 · held 5.53s



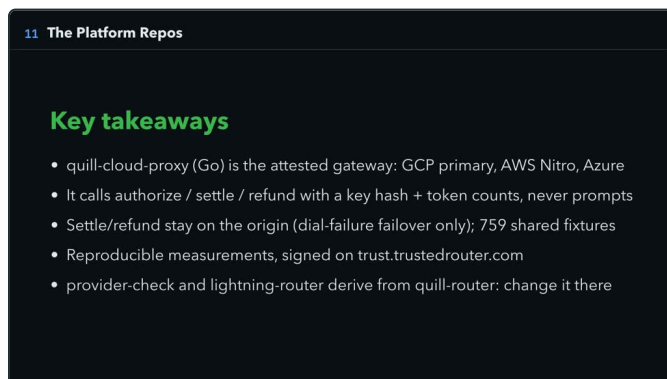
The platform repos · 0:53 · held 12.2s



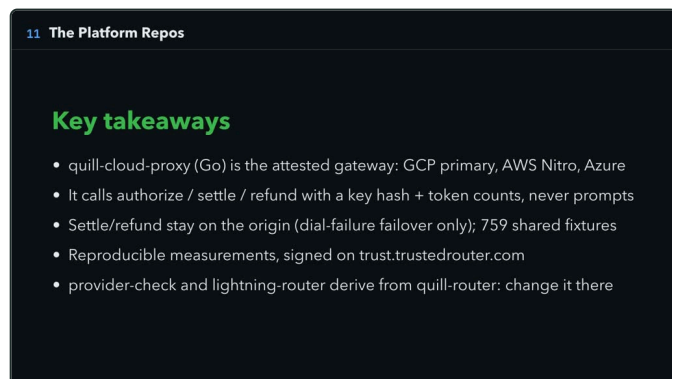
The platform repos · 1:09 · held 6.57s



The platform repos · 1:32 · held 6.33s



The platform repos · 1:40 · held 5.9s



The platform repos · 1:47 · held 6.5s

Video 12 · SDKs, apps & benchmarks

12 SDKs, Apps & Benchmarks

	py	js	go	rust	java	swift
Chat Completions	✓	✓	✓	✓	✓	✓
SSE streaming	✓	✓	✓	✓	✓	✓
Responses API	✓	✓	✓	✓	✓	✓
Anthropic Messages	✓	✓	✓	✓	✓	✓
Embeddings	✓	✓	✓	✓	✓	✓
Attestation verify	✓	✓	✓	✓	✓	✓
Receipts (Ed25519)	✓	✓	✓	✓	✓	✓
OAuth delegation (PKCE)	✓	✓	✓	✓	✓	✓
Bundled CLI	✓	✓	✓	—	—	—

Feature parity is the rule: all six SDKs ship the same core API surface.

SDKs, apps & benchmarks · 0:37 · held 3.8s

12 SDKs, Apps & Benchmarks

	py	js	go	rust	java	swift
Chat Completions	✓	✓	✓	✓	✓	✓
SSE streaming	✓	✓	✓	✓	✓	✓
Responses API	✓	✓	✓	✓	✓	✓
Anthropic Messages	✓	✓	✓	✓	✓	✓
Embeddings	✓	✓	✓	✓	✓	✓
Attestation verify	✓	✓	✓	✓	✓	✓
Receipts (Ed25519)	✓	✓	✓	✓	✓	✓
OAuth delegation (PKCE)	✓	✓	✓	✓	✓	✓
Bundled CLI	✓	✓	✓	—	—	—

Chat, streaming (fail-closed on a missing [DONE]), Responses, Messages, embeddings.

SDKs, apps & benchmarks · 0:42 · held 4.3s

12 SDKs, Apps & Benchmarks

	py	js	go	rust	java	swift
Chat Completions	✓	✓	✓	✓	✓	✓
SSE streaming	✓	✓	✓	✓	✓	✓
Responses API	✓	✓	✓	✓	✓	✓
Anthropic Messages	✓	✓	✓	✓	✓	✓
Embeddings	✓	✓	✓	✓	✓	✓
Attestation verify	✓	✓	✓	✓	✓	✓
Receipts (Ed25519)	✓	✓	✓	✓	✓	✓
OAuth delegation (PKCE)	✓	✓	✓	✓	✓	✓
Bundled CLI	✓	✓	✓	—	—	—

The trust features too: gateway attestation, signed inference receipts, OAuth credit delegation.

SDKs, apps & benchmarks · 0:46 · held 3.7s

12 SDKs, Apps & Benchmarks

trusted-router-sdk-conformance

25 scenarios

retry 503/500/429 · x-should-retry truncated SSE · redirects · replay timeouts · cancel · OAuth · typed models

TLS fault server

loopback · throwaway CA pretends to be api.trustedrouter.com

py driver

js driver

go driver

rust driver

java driver

swift driver

One black-box suite: 25 fault-injection scenarios, run against all six SDKs' real public APIs.

SDKs, apps & benchmarks · 0:58 · held 8.5s

12 SDKs, Apps & Benchmarks

trusted-router-sdk-conformance

25 scenarios

retry 503/500/429 · x-should-retry truncated SSE · redirects · replay timeouts · cancel · OAuth · typed models

TLS fault server

loopback · throwaway CA pretends to be api.trustedrouter.com

py driver

js driver

go driver

rust driver

java driver

swift driver

every SDK PR

nightly 04:17 UTC

repository_dispatch

A shared oracle checks both the result and the wire transcript. The v0.2 gate allows no skips.

SDKs, apps & benchmarks · 1:02 · held 3.7s

12 SDKs, Apps & Benchmarks

trusted-router-sdk-conformance

Not in the shared matrix

✗ cryptographic attestation validation

✗ regional affinity / failover

✗ first-frame deadlines · SSE heartbeat timing

✗ integrations, apps, benchmarks (not covered at all)

integration

lim · nfn · opendaw ...

official SDK

macos · windows · linux

/v1 API

api.trustedrouter.com

Attestation validation and regional failover only get each SDK's own native tests.

SDKs, apps & benchmarks · 1:11 · held 5.8s

12 SDKs, Apps & Benchmarks

Forks: upstream-PR staging

aisdk · aisdk.providers

Yulab-SMU (R)

PR open

FastWorker

andrewyng/openworker

PR open

openclaw

→ use openclaw-trustedrouter

PR closed

hermes-agent

→ use hermes-trustedrouter

PR closed

main is just CODEOWNERS/SECURITY; rebase the TR branch onto upstream

Benchmarks: research artifacts

FreedomBench

50 MiCQ, PnC-censored facts

no CI

last Aug 31

PrometheusBench

rebase / permissions

no CI

last Sep 2

trustedrouter-benchmarks

ifEval · MMLU-Pro · sciCode

no CI

last Aug 16

ExploitBench (TR + Prometheus)

Fusion as a model · Opus study

no CI

last Aug 16

py-SDK based (v=0.3.0); run by hand with disposable keys

Forks only stage upstream PRs. OpenClaw and Hermes said no, so use the plugin repos.

SDKs, apps & benchmarks · 1:24 · held 8.9s

12 SDKs, Apps & Benchmarks

Key takeaways

Four layers: API → 6 official SDKs → integrations → apps

Registries and tags are the source of truth, not GitHub Releases

Conformance: 25 fault scenarios × 6 SDKs, every SDK PR + nightly

Integrations bypass the SDKs: no attestation, receipts or SDK retry

Forks = upstream-PR staging; benchmarks have no CI and are mostly dormant

SDKs, apps & benchmarks · 1:41 · held 5.9s

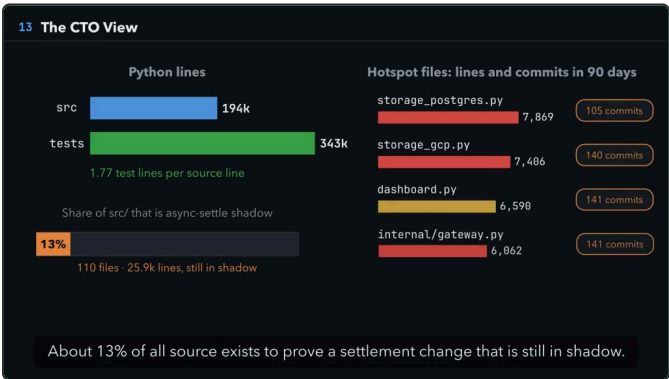
Video 13 · The CTO view



The CTO view · 0:30 · held 11.6s



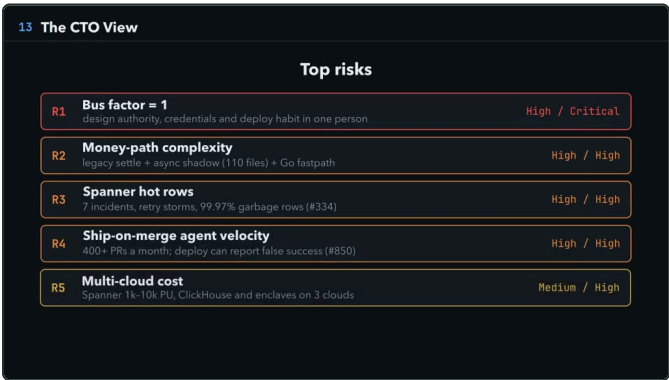
The CTO view · 0:41 · held 7.3s



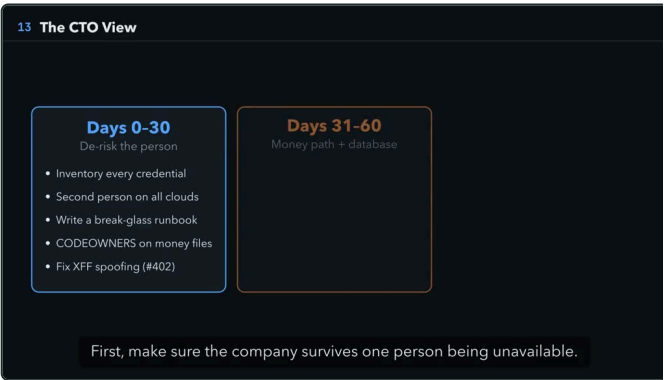
The CTO view · 0:58 · held 10.1s



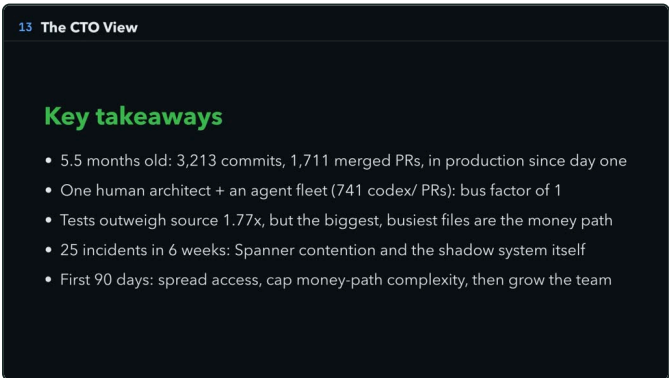
The CTO view · 1:07 · held 6.6s



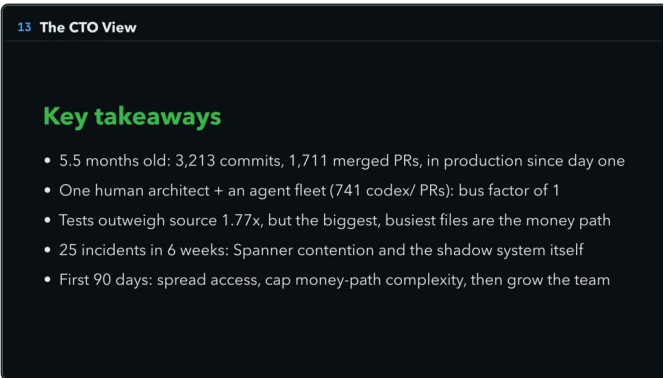
The CTO view · 1:16 · held 5.7s



The CTO view · 1:27 · held 4.8s



The CTO view · 1:40 · held 5.1s



The CTO view · 1:46 · held 6.5s