KwalitAI private report

kwalitai

Commit	e391c685963b5fe79acc2594a7fad4be88105c1d
Review date	17 October 2026

© Doolta

CODE QUALITY AUDIT

kwalitai

Commit e391c68 · Reviewed 17 October 2026 · Whole repository · 32 files, 5 units

SCORE

78 /100

0 50 100

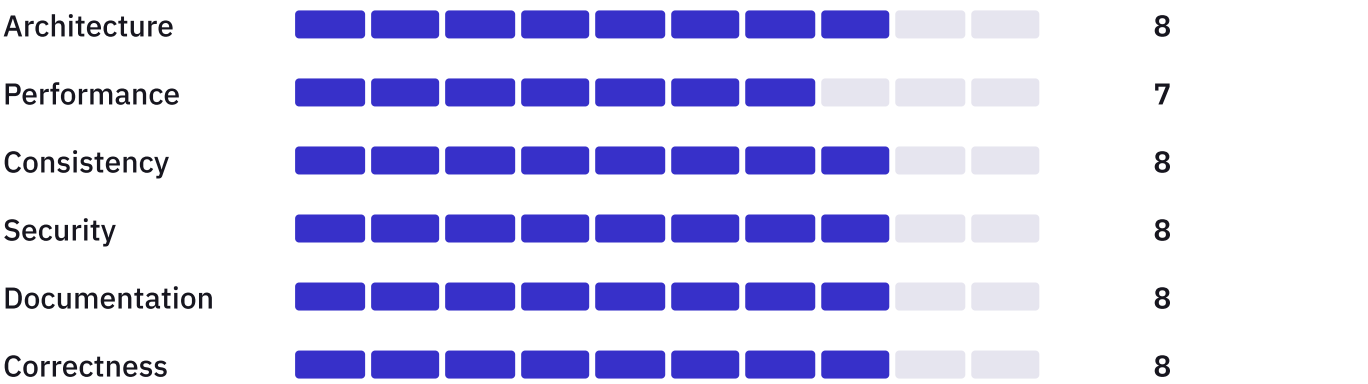
EVOLUTION

Not compared

No earlier review is compared with this one. A later review made with the same method reads "stable" when the score is identical; until the noise floor is measured, any other move is withheld, never shown as a number.

Six axes

Graded 1 to 10



9
Defects to act on, all Medium, plus 7 check gaps

0
Critical or High

88.8%
Test coverage

13
Leads checked and cleared

In short

The codebase is uniformly good, and the one movement since review_kwalitai_20261017_2 is real. Commit 9f47c79 made the complexity gate measure what the thresholds say, and the five cyclomatic-11 functions came down to 10 or below, so Consistency rises from 7 to 8. Performance is the weakest axis at 7.

Limits. Branch protection could not be read (there is no origin remote), so every wired CI job is treated as blocking.

Findings at a glance

By axis and severity

AXIS	GRADE	CRITICAL	HIGH	MEDIUM	LOW	DEFECTS	CHECK GAPS	CLEARED
Architecture	8	0	0	0	0	0	0	0
Performance	7	0	0	2	0	**2**	1	2
Consistency	8	0	0	0	0	0	2	5
Security	8	0	0	5	0	**5**	4	4
Documentation	8	0	0	0	0	0	0	0
Correctness	8	0	0	2	0	**2**	0	2
Total	**78/100**	**0**	**0**	**9**	**0**	**9**	**7**	**13**

Severity columns count defects. Check gaps: quality checks that are missing or too loose, counted per check and never added to the defects. Cleared: leads the review checked and found not to be defects, listed after the findings.

How the findings could have been caught



- **Tool-detectable** a static check would catch it
- **Found by review** a test or a reader finds it
- **Check gap** a quality check is missing

9 defects and 7 check gaps. Shares of defects leave the check gaps out.

Measured on this commit

88.8%
Test coverage

5 of 6
Check families wired in every unit

56%
Defects a configured check should catch

32
Files reviewed

Findings and fixes

In priority order

1 Medium Security Performance Architecture

Wire the detectors behind today's 5 tool-detectable defects. (a) Add `[ -n "$$TRUSTED_PROXIES" ]` to `make preflight` (Makefile:106-113): without it, production can start with every visitor sharing one contact limit. (b) Add the shared semgrep pack, at least `go.sql-select-no-limit` and `go.query-in-loop`, to the repo's semgrep config, and run it from `make audit` (Makefile:53). (c) Pin all 19 `uses:` to commit SHAs and add `zizmor .github/workflows` to the CI lint job (ci.yml:43)

Each of these has been raised since 2026-10-16, apart from (a), which is new.

2 Medium Correctness

Reject or warn when an identical report is re-ingested under another repository

In `existingEvaluation` (internal/ingest/ingest.go:181), compare the found row's `repo_id` with the requested repository. Today the operator is told "already ingested", and the report stays attached to the wrong repository. Seam: `TestIngestSameReportOtherRepo`.

3 Medium Correctness

Cover the production ingest wiring: `TestServeIngestEndToEnd` (POST /ingest through `poolIngester`, serve.go:101, now 0%) and `TestRunIngest` (main.go:261, now 33.3%)

A mistake between `serve` and `ingest.Run` passes `make ci` today.

4 Medium Security

Give the contact limiter a global ceiling, and key IPv6 clients on their /64 (internal/web/contact.go:167)

Seams: `TestContactLimiterIPv6Prefix` and `TestContactLimiterGlobalCeiling`.

5 Medium Security

Log the `repo` value only after the token check, or truncate it to 200 bytes (internal/web/ingest.go:46, publish.go:35)

6 Medium Consistency Check gap

Make formatting and error checking failing gates: `formatters: [gofmt]`, `errcheck` and `unused` in `.golangci.yml:13`, and drop `-U1000` at :43

Expected new findings: 0.

7 Medium Security

Add `gitleaks detect --redact` to `make audit` (Makefile:53)

Nothing stops a committed credential today.

8 Medium Correctness

In `storeReport` (internal/ingest/report.go:83), fsync before the rename, and verify an existing copy against `report_sha256` instead of trusting `os.Stat`

Seam: `TestStoreReportReplacesCorruptCopy`.

9 Medium Performance

Add a page size to `PGStore.Directory` (internal/web/store.go:94) and a request logger in `newRouter` (cmd/kwalitai/serve.go:33)

10 Medium Documentation

Replace the template text in SECURITY.md:5-6 and CONTRIBUTING.md:3 with the project's actual policy

11 Low Performance Security

Add FuzzUnpack seeded from TestUnpackRejects (internal/ingest/archive_test.go)

Other findings

20 outside the priority list

Medium Consistency Check gap

`Makefile:125`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: no format gate. gofmt runs only inside make tidy, which rewrites files; .golangci.yml has no formatters section and CI never checks formatting. Tree is gofmt-clean at HEAD (checks/format_tidy.txt), so the gate is absent, not red

Medium Performance Check gap

`Makefile:53`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: shared semgrep pack absent from make audit and CI, semgrep not installed. RC11, RC13, RC15 to RC19 and RC24 rules never run; 2 live instances found by reading (ingest.go:207, store.go:94)

Medium Security Check gap

`cmd/kwalitai/serve.go:68`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: hardening presence probes absent (spproof installed, no .spproof.yml). Server timeouts, SetTrustedProxies and the SMTP deadline are in place by hand; no check fails if one is removed

Medium Security Check gap

`.github/workflows/ci.yml:30`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: no workflow security linter (zizmor or pinact) in make audit or CI; 19 of 19 action references across both workflows are pinned by mutable tag

Medium Security Check gap

`Location withheld`

Withheld from this sample: a security finding still open at the follow-up review. A Security finding is shown only once fixed.

Medium Security Check gap

`.github/workflows/ci.yml:43`

recurrence since review_kwalitai_20261017 and review_kwalitai_20261017_2: no secret scanner (gitleaks or trufflehog) in make audit or CI; 0 hits at HEAD per a local trufflehog filesystem run (checks/trufflehog.txt)

Medium Performance Tool-detectable

`internal/ingest/ingest.go:207`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: one INSERT per finding_counts key inside a loop in the ingest transaction (bounded by axis x severity x class, at most 7x4x6 = 168 rows)

Medium Performance Tool-detectable

`internal/web/store.go:94`

recurrence since review_kwalitai_20261017 and review_kwalitai_20261017_2: Directory selects every published repository with no LIMIT and GET /r renders all rows, so page size and query time grow with the catalogue

Medium Security Tool-detectable

`.github/workflows/release.yml:27`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: goreleaser-action, checkout and setup-go pinned by mutable tag in the contents: write release job that receives GITHUB_TOKEN

Medium Security Tool-detectable

`.github/workflows/ci.yml:136`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: 16 action references pinned by mutable tag, including third-party trivy-action in the security-events: write scan job

Medium Security Tool-detectable

`Makefile:106`

make preflight gates make up on POSTGRES_PASSWORD and APEX_DOMAIN only; an empty TRUSTED_PROXIES boots prod behind Traefik with every visitor sharing one contact bucket (3 messages, then 1 per 10 minutes), so one sender can keep POST /contact at 429 for everyone. README and env.sample document the risk; nothing checks it

Medium Correctness Found by review

`internal/ingest/ingest.go:181`

re-ingesting an identical report directory under another --repo (schema 1, or schema 2 with a null repo; CLI and POST /ingest ?repo=) conflicts on report_sha256, and existingEvaluation returns the first repo's evaluation with no warning because the hashes match; the operator sees 'already ingested' and a later publish of the intended repo fails with 404. Confirmed by reading ingest.go:154 and 181-193, not executed

Medium Correctness Found by review

`internal/ingest/report.go:83`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: report copy is not fsynced before the rename and an existing destination is trusted on os.Stat alone, although report_sha256 is stored and could verify it; documented in the doc comment, not fixed, not tested

Medium Security Found by review

`internal/web/contact.go:167`

recurrence since review_kwalitai_20261017_2: contact limiter keys each bucket on the full client address with no global ceiling, so the number of emails sent is bounded only by how many addresses a sender controls (an IPv6 /64 holds 2^64); whether the edge accepts IPv6 was not verified

Medium Security Found by review

`internal/web/ingest.go:46`

recurrence since review_kwalitai_20261017_2: unauthenticated POST /ingest and POST /publish (publish.go:35) are logged at Warn with the caller-chosen repo query value before the token check, up to the 1 MB default header limit, so an anonymous client controls log volume within the Traefik limit of 10 per minute per IP

Not scored Correctness

`cmd/kwalitai/main.go:261`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: ingestDir (CLI ingest success path) is 33.3% covered and poolIngester.Ingest (cmd/kwalitai/serve.go:101, the production adapter behind POST /ingest) is 0%

Not scored Performance

`cmd/kwalitai/serve.go:29`

recurrence since review_kwalitai_20261016, review_kwalitai_20261017 and review_kwalitai_20261017_2: no access log and no metrics surface: gin.New without a request logger; slog records handler errors and ingest, publish and contact outcomes only

Not scored Documentation

`SECURITY.md:5`

recurrence since review_kwalitai_20261017 and review_kwalitai_20261017_2: template instruction text shipped in the public security policy (the paragraph telling the author to adjust the rows); CONTRIBUTING.md:3 carries the same kind of template line

Not scored Performance

`internal/ingest/archive.go:83`

no benchmark and no fuzz target in the repo (0 Benchmark, 0 Fuzz functions); Unpack and countFindings parse untrusted or operator input and are natural fuzz targets

Not scored Security

review host lacks semgrep, gitleaks, zizmor and conform; tree-sitter complexity and dependency analysis failed; their coverage is listed under Checks not run

Structural opportunities

Beyond single findings

A · Architecture, Correctness

Lens 1 (make illegal states unrepresentable)

Give the six axes one named type owned by `internal/score`. The order is restated by `ingest.axisNames` (review.go:27), by `web.AxisNames` (page.go:12), by the positional INSERT at ingest.go:149-158 and by the positional `Scan` at store.go:140-141. A swap in any one of them compiles, and it would silently store or show one axis's score under another's name. Justification: a test asserting that the two Go lists are equal is the smaller alternative. It is rejected because the two SQL sites bind by position and that test cannot see them.

Cost: a spike over about 6 files, with no behaviour change.

B · Security, Architecture, Documentation

Lens 3 (explicit versus implicit contracts and trust)

Write down who guarantees what between Traefik and the app. The client IP, the security headers, the write rate limit and TLS all depend on edge configuration. The rules are spread over README, env.sample and compose comments, and only two settings are checked at deploy. An ADR "edge versus application responsibilities", with one check per property, turns today's TRUSTED_PROXIES gap into a checklist item. Justification: the single preflight line in recommendation 1 is the smaller alternative. It fixes this instance, but the next edge-owned property would again live only in labels.

C · Correctness, Security

Lens 3 and lens 5 (root cause)

`report_sha256` is stored as the integrity value of the report copy, but no code ever compares a copy with it. The fsync issue and the trust-on-stat issue are two symptoms of that one missing check, raised in four reviews. A `VerifyReport` used by `storeReport` closes both. Justification: fsync alone is rejected because it prevents new corruption but never detects existing corruption.

Checked and cleared

13 leads · not counted

Performance · 2

- resolved lead: io.LimitReader on the server response has no overflow check, but a truncated JSON body fails to decode, so truncation cannot produce wrong data
- resolved lead: regexp.MustCompile inside lightToken runs only at package init (8 calls), never per request

Consistency · 5

- resolved since review_kwalitai_20261016: gocyclo min-complexity 10 and gocognit 15 now report exactly above the thresholds (commit 9f47c79); the probe at 7 shows max cyclomatic 10 over 129 functions
- resolved since review_kwalitai_20261017_2: the five functions at cyclomatic 11 (runServe, runSend, Read, countFindings, resolve) are now 10 or below
- resolved lead: 31 errcheck check-blank hits from the review config are deliberate discards (fmt to stderr, deferred Close and Rollback, best-effort RemoveAll, AfterFunc SetDeadline, fs.Sub on an embedded path, json.Unmarshal of a schema-checked metrics block) or test code
- resolved lead: 14 govet shadow hits on err are scoped re-declarations with no lost error
- resolved lead: shellcheck SC2209 on bump=patch is a deliberate string assignment

Security · 4

- resolved lead: govulncheck reports 1 advisory in a required module that the code does not call
- resolved lead: Unpack rejects links, devices, absolute and non-local paths, duplicates and file/dir clashes, and caps 64 files and 5 MiB written with overflow detection
- resolved lead: origin() trusts the Host and X-Forwarded-Proto headers, but Traefik routes only the two Host rules and html/template escapes the result; it changes only the requester's own page
- resolved lead: STARTTLS is opportunistic, but net/smtp PlainAuth refuses to send credentials over an unencrypted connection to a non-localhost relay, so a downgrade fails the delivery instead of leaking the password

Correctness · 2

- resolved since review_kwalitai_20261017_2: the app container now receives SMTP_*, CONTACT_*, INGEST_TOKEN_SHA256 and TRUSTED_PROXIES and keeps report copies on the kwalitai-data volume (commit 7cab7e2)
- resolved lead: HEAD /healthz answers 200 for the wget --spider healthcheck and is pinned by a test (main_test.go:25)

CONFIDENCE LIMITS

Branch protection could not be read (there is no origin remote), so every wired CI job is treated as blocking. The production `.env` was not inspected. The new ingest finding is confirmed by reading the code, not by execution. semgrep and tree-sitter did not run, so the resource-safety and structural checks rest on reading the source.

ANNEX A

How this report is made

The score out of 100

An automated review grades six axes from 1 to 10, from the code and the output of its checks. The score comes from those grades by a published rule, formula k1. All axes weigh the same; an axis marked N/A is left out.

```
score = min( round(10 × mean of rated axes), 10 × (lowest axis + 2) )
```

The weakest axis caps the mean. One axis moving one step moves the score by about 1.7 points, so small moves must be read against the noise floor.

THIS REPORT	VALUE
10 × mean of 8, 7, 8, 8, 8, 8 (7.83)	78.3
Cap from the lowest axis: 10 × (7 + 2)	90
Score: the smaller of the two, rounded half down	**78**

Comparing two reports

SHOWN AS	WHEN
Signed change	Same repository, same method, identified by its fingerprint.
Method approximated	Same method label, fingerprint unknown, as for historical reports.
Method changed	Any other case. No number is shown: it would compare unlike things.

Noise floor

Two runs on the same commit may not give the same grades. Until a repeatability study measures that spread, only an identical score reads "stable"; any other move reads "change withheld", never a signed number.

Follow-up verification

One verification, at a later commit you choose within 90 days, re-runs the same review. Tool-detectable findings are marked fixed or still present; the others still present or not found again. New findings are listed. It reports what it finds and certifies nothing.

ANNEX B

Definitions

The six axes

Architecture	How the code is divided into modules and layers, and whether dependencies point in a direction that keeps change local.
Performance	Whether the code avoids needless work, allocation and waiting on the paths that run most often.
Consistency	Whether the same problem is solved the same way: naming, error handling, structure and conventions.
Security	How the code treats untrusted input, secrets, access control and its dependencies.
Documentation	Whether a newcomer can understand, build and use the project from its documentation and comments.
Correctness	Whether the code does what it claims, edge cases and error paths included, and whether tests back that claim.

Severity

Critical	Exploitable or data-losing on a reachable path. Fix before the next release.
High	Likely to cause an incident or a wrong result in normal use. Fix this cycle.
Medium	A real defect with limited reach, or one that makes the next defect likely. Plan it.
Low	Hygiene. Fix when you next touch the code.

How a finding could have been caught

Tool-detectable	A static tool catches it, or a rule the review names would. It does not mean the project runs that tool.
Found by review	A logic defect no static rule catches; tests or a careful reader find it.
Check gap	A quality check that is missing, not run, failing or too loose. Counted per check, never added to defects.
Note	Not a defect: process notes, coverage gaps, resolved leads. Listed as "not scored", with no severity, never counted.

Review record

Repository	`kwalitai`
Commit	`e391c685963b5fe79acc2594a7fad4be88105c1d`, clean tree
Run	17 October 2026, 07:19 to 07:33 UTC · Whole repository, 32 files, 5 units
Engine	10x-review 1.1.0, report schema 2
Method fingerprint	`sha256:380f85a3c9aebf5eee717fd531e6917f2d17eb30cb626828e2b7de2d9399131f`
Review models	claude-opus-5-5 for every axis