

A job outside the gate's closure: contract-tests

SAMPLE · example-org/example-service · synthetic, describes no real repository

Finding

The job all-green, shown as “all checks passed”, runs under always() and puts toJSON(needs) in its environment, where a jq predicate exits non-zero on any result other than success or skipped. It therefore rejects a failed dependency, and sees exactly the three jobs in its needs. contract-tests is not among them, so when it runs and fails the gate is not obliged to notice.

Classification

Gate covers	changes, unit-tests, lint
Outside the gate	contract-tests, triggered by needs.changes.outputs.api == 'true'
Severity	Latent. The job is path-gated, so on many pull requests it is skipped, which the gate correctly allows. The gap is that when it does run and fail, the gate is not obliged to see it.
Classification	proven_structural_gap , not proven_enforced_merge_gap. The exclusion is proven from the file. Whether this gate is an enforced required check was NOT established here, because no required-check list was supplied.

What is proven, and what is not

Proven	contract-tests is not in the gate's needs, is not reachable to the gate through the needs graph, and has no dependent that the gate inspects. Its failure reaches the gate neither directly nor indirectly.
Also proven	The gate's failure step genuinely rejects a bad result. Run with --exercise, the verifier executes that step against constructed inputs and observes exit 0 when all dependencies pass and exit 1 when one fails.
Not established	Enforcement. No required-check list was supplied, so this record does not claim the gate is required, and a gate-like name is never treated as evidence.
Not claimed	That contract-tests ought to block a merge. That is repository policy, not a fact about the file, and the certificate records it as unresolved.

Fix

Add contract-tests to all-green.needs so it enters toJSON(needs). The existing predicate already rejects any result other than success or skipped, so nothing else changes and skipped runs continue to pass.

Verification

```
python3 verify.py ci.yml certificate.json --exercise
python3 verify_report.py certificate.json --commitments commitments.json
python3 verify_commitments.py commitments.json --issuer-keys keys.json
```

The first reproduces the finding from the workflow. The second checks that the analyser behind it was committed to the public log before the record was issued. The third checks the log itself is unaltered. No Arcifact service and no proprietary analyser is required: the verifiers are supplied as inspectable source and recompute everything locally.

Provenance

Synthetic sample. The certificate binds the SHA-256 of the ci.yml beside it and names the analyser digests that produced it. On a real review this section carries the repository, the commit, the git blob and the digest of the exact bytes analysed.