

Sami Habbal

Charlotte, NC · samihabbal5@icloud.com · gg5533.github.io · github.com/GG5533

I work on the failures that only appear under real traffic: retries that cause double effects, provider calls with no deadline, agent loops with no ceiling, and state that disagrees with what the payment provider thinks happened.

OPEN-SOURCE RELIABILITY WORK

changedetection.io — LLM client retry semantics

33k★ · Sept 2026

- A fix of mine was merged into master by the project owner ([PR #4390](#)). Their history path-containment guard had no test — disabling it left every existing test passing, because `os.path.basename()` neutralised both traversal fixtures before they reached it. The new test is the only one that fails when the guard is removed.
- Reviewed the retry handling in the LLM client; a contributor implemented the findings in [PR #4384](#) (open, under the owner's review). My regression tests for the retry-budget behaviour were **merged into that PR's branch** as commit `1f0ed3f`. Verified by mutation: deleting `attempt -= 1` fails that test and no other in the file.
- Filed [#4385](#): the AI intent filter fails *closed* on a malformed model response — suppressing the change and caching that verdict — while every other failure path in the same function fails open.

Circle · arc-node — SIGTERM shutdown race

Rust · Sept 2026

- Reviewed a fix for a shutdown race in their consensus node. The fix guards the path where the app task resolves normally, but the `?` on the line above the new check discharges a `JoinError` first — so a panicking task still drops the runtime and kills cleanup mid-flight.
- Independently confirmed by another contributor against a standalone tokio model: "*@GG5533's ? finding is real.*"
- Corrected my own scope publicly when a reviewer showed the durability consequence was narrower than I first claimed. [circlefin/arc-node#361](#)

SELECTED PROJECTS

stripe-fulfilment-kit — fenced leases for order fulfilment

TypeScript · [repo](#)

- A lease implementation that passed 78 tests and was still wrong: a provider call outliving its lease could write failure over a completed order, hiding a paid-for product and reopening it for a duplicate charge. Every test settled an order from its *current* owner; none covered a write from a superseded one.
- Fixed with fencing tokens enforced atomically across processes via `BEGIN IMMEDIATE`, plus regression tests for the case the original suite could not see. [Full write-up](#).

ai-review-gate — adversarial review before output ships

[repo](#)

- Two models review each substantive answer in parallel; either can block the turn and force a revision. It fires after the draft has streamed, and the revision itself is not re-reviewed, so it reduces wrong answers rather than preventing them — stated in the README rather than omitted.
- It has caught claims that outran their evidence, including a durability argument of mine that rested on call ordering I had not checked, and a case where I confused mutation testing (which shows a guard has coverage) with a bug reproducer (which must fail on unmodified code).

TECHNICAL

TypeScript, Node, Python, Rust (reading/review), Next.js, Stripe, SQLite/Postgres, LLM APIs (OpenAI/Anthropic), agent orchestration, distributed-systems debugging, formal reasoning about concurrency (leases, fencing, idempotency).

Every claim above links to a public artifact. Merge status is stated where a contribution is not yet merged.