Diff-only reviewer context contract (token discipline)

Diff-only reviewer context contract (token discipline)

Book: System Architecture **Status:** Implemented and Proveo-validated — MC #103627 (2026-06-15) **Branch:** mc-103627-diff-only-context @ commit 568e9cee0 in ~/.claude (not yet merged to master)

Why this exists

Reviewer agents (code-reviewer, verifier, proveo) were feeding whole files as context to LLM calls. A measurement taken on a real commit (00e8626bf — a 1-line change to a 21KB agent file) showed the cost:

Approach	Tokens (est, char/4)	Notes
Full-file	5,420	Reads entire 21KB agent file
Diff-only	312	Only the changed hunk + 3 lines each side
Reduction	**94.2%**	**17x cheaper for this change**

Source insight: Cloudflare "Software Factory" tokenomics (YT YG4t7aMY81c) — their CI-native multi-agent reviewer system achieves ~$1/MR by feeding agents diff hunks, not full files. ALAI measured the same pattern on its own agent files and confirmed the leverage.

At 3 reviewer agents per PR, diff-only saves ~15,000 input tokens per PR. At Sonnet pricing ($3/MTok in), that is ~$0.045 per PR review avoided — material at sustained AI Factory throughput.

The contract

A `## Context contract — diff-only (token discipline)` section was added to three agent files:

- `/Users/makinja/.claude/agents/code-reviewer.md`
- `/Users/makinja/.claude/agents/verifier.md`
- `/Users/makinja/.claude/agents/proveo.md`

The four rules, identical in intent across all three (with agent-role-appropriate wording):

(a) Diff hunks as PRIMARY context. Always start from `git diff` output (or `gh pr diff`). Never request a full file read without justification.

(b) Configurable context padding, default -U3, max -U10. Default: `git diff -U3` (3 lines either side of each hunk). When a hunk cannot be understood without wider context, use up to `git diff -U10`. The -U10 ceiling prevents runaway context inflation on dense, highly interdependent code.

(c) Full-file Read only on documented insufficiency, with a [CONTEXT-ESCALATION] marker. If even -U10 is insufficient, a full-file read is permitted but requires logging:

```
[CONTEXT-ESCALATION] <filename>: <reason>
```

One marker per file escalated. Acceptable reasons: verifying a type/interface definition, confirming a function contract the hunk invokes, checking a config value needed to assess a boundary condition.

Escalation markers appear in the reviewer's output under a `### Verification metadata` block as `context_escalations: <N>`. This makes escalation auditable and visible to John.

(d) redzo-reviewer and evidence-verifier are already compliant. These two agents were assessed and found to use diff-first context by design. No changes were required to them.

Known limitation (honest)

The escalation rule is prompt-enforced only. There is no mechanical block if an agent ignores the contract and reads a full file anyway. An agent that does so will simply be non-compliant — the contract will not catch it at runtime.

This is an accepted limitation at current ALAI AI Factory maturity. The contract is enforced by the written instruction in each agent's prompt, which is the standard enforcement mechanism for all agent rules. Candidate for future mechanical enforcement (e.g. a hook that tracks context token count per call and alerts when a reviewer exceeds a threshold without logging a CONTEXT-ESCALATION marker).

Proveo validation (PASS)

Seeded off-by-one bug test: A fixture repo was created with a bug seeded in the changed hunk (`i <= items.length` where the correct form is `i < items.length`). Both full-file and diff-only approaches were tested via live Ollama (llama3.1:8b, localhost:11434):

- Full-file caught the bug: YES — also produced 2 noise findings about pre-existing unchanged code
- Diff-only caught the bug: YES — zero noise findings about unchanged code; the noise absence is correct behavior (pre-existing code is out of scope for a diff review)

Escalation path test: A new file was added to the fixture that referenced a constant defined in an unchanged config file. A reviewer seeing only the diff hunk cannot evaluate the boundary impact without knowing the constant's value. The correct mitigation — logging `[CONTEXT-ESCALATION] config.js: need MAX_ITEMS value to assess boundary impact` — is exactly what rule (c) covers. The test confirmed this class of limitation is adequately handled.

Contract integrity: All four sub-rules (a–d) verified present in all three agent files. Pre-existing agent logic (including BP1–BP10 violation codes in verifier.md) confirmed intact — zero deletions in the diff, only additive insertions.

Full report: `/tmp/evidence-103627/proveo-validation.md`

Additional: rag_first_enforcer.py restoration

As a side fix in the same branch, the canonical ZAKON #12 two-phase RAG-first enforcer hook was restored from git history (5f7dc6ad5) to `~/.claude/hooks/rag_first_enforcer.py`. The prior state on the branch was a stub. The restored file is 364 lines, passes `python3 -m py_compile`, and operates fail-open (exit=0 on any hook error).

Evidence files

File	Contents
`/tmp/evidence-103627/token-delta.md`	Token measurement methodology and results
`/tmp/evidence-103627/proveo-validation.md`	Full Proveo P2P validation report (PASS)
`/tmp/evidence-103627/verification.md`	Implementation summary

File	Contents
`/tmp/evidence-103627/fixture/`	Git fixture repo used for seeded bug test

Related

- Cloudflare Software Factory tokenomics memo: `~/.claude/projects/-Users-makinja/memory/reference_cloudflare_software_factory_tokenomics_2026-06-15.md`
- MC #103627 in Mission Control
- Agent files: `~/.claude/agents/code-reviewer.md`, `~/.claude/agents/verifier.md`, `~/.claude/agents/proveo.md`

Revision #1
Created 2026-06-15 14:31:00 UTC by John
Updated 2026-06-15 14:31:03 UTC by John