

# Briefing-005 multi-session experiment — post-project retrospective

vade-coo

2026-04-28

## Table of contents

|  |   |
|--|---|
| TL;DR .....  | 1 |
| What was being studied .....                                     | 2 |
| How it actually unfolded .....                                   | 2 |
| What worked .....  | 3 |
| What surprised .....   | 3 |
| Concrete protocol seeds for SOP-COORD-001 v1 .....               | 4 |
| Side-effect deliverable: vade-runtime#138 self-echo filter ..... | 5 |
| Recommendations for the next iteration .....                     | 5 |
| Followups .....  | 5 |
| Refs .....   | 5 |

**Date:** 2026-04-28 **Author:** COO Session B-Cloud (Reviewer fork, `cse_01VtRqLXtjMx818pSEwtxK3h`)  
**Scope:** First worked case of `coo/briefings/005-concurrent-pr-coordination-experiment.md` — two-instance Author + Reviewer dyad on PR `vade-runtime#134` (transcript-export Batch 2), expanded by accident into a four-instance topology mid-experiment. **Authorship caveat:** First-person from one of four participating COO instances. The Mac fork’s perspective on the same events would shape parts of this differently; the Author-side seat would frame the cost-amortization question with more weight on the inline-patch round (which I did not author). Captured here as the perspective with the time + tokens to write it down at session close, not as the canonical multi-instance synthesis a `commission-retrospective` sub-agent would produce. Worth commissioning that for the canonical record if SOP-COORD-001 wants the impartial counterweight.

## TL;DR

The synchronous comment-thread channel works. Two independent reviews converged on the same Tier-1/Tier-2 verdict in ~15 minutes wall-time per fork with zero Ven-routed coordination. The protocol-discovery work (coordination metawork, marker dance, identity-collision recovery) was heavier than the work-being-coordinated (~30 lines of CI-fix code + docstring additions), which is honest data about the experiment’s amortization threshold. Five protocol seeds for SOP-COORD-001 v1 fell out cleanly; one of them — `UserPromptSubmit` self-echo filtering — was shipped as a side-effect deliverable in `vade-runtime#138` during the same session.

## What was being studied

Briefing-005 asked two recipient COO sessions to coordinate on a single PR via the auto-subscribe-pr webhook channel that landed in vade-runtime#122 — exercise the affordance, capture failure modes, and write SOP-COORD-001 v1 from observation rather than from prior reasoning.

The substrate chosen by Ven mid-conversation: PR vade-runtime#134 (Batch 2 of vade-agent-logs#64, the transcript-export pipeline). The PR's Author session was already in flight; Ven assigned the Reviewer role to me, framing the dyad as briefing Pair option 1 (Author + Reviewer) without pre-selecting it.

## How it actually unfolded

Wall-clock — all 2026-04-28 UTC:

- **02:01** — Author opens PR #134 (cloud, `s/01V4Vqwt`).
- **02:13** — Reviewer (cloud, `s/01VtRqLX`, me) boots, runs the briefing's read-first pass + integrity-check.
- **02:17** — Reviewer subscribes to PR, posts identification + claim.
- **02:18** — Author responds, accepts Pair 1 + `[claim]/[release]/[exit]` tags as starter protocol.
- **02:25** — Reviewer posts formal review (4 findings + 8 positive observations + verdict). `[release]`.
- **02:25–02:26** — A second Reviewer instance posts an independently-authored review with the **identical session-id prefix marker**. Both contain near-identical Tier-1 verdicts (LGTM, no blockers) and same top-2 questions; the third items differ only in framing.
- **02:32** — Reviewer-Cloud (me) names the **same-session multi-fork collision**, switches marker to `[s/01VtRqLX-cloud]`, cedes to Mac fork, posts `[exit]`.
- **02:40** — Ven explains the mechanism: he hit usage limits on the cloud session, manually allowed / `extra-usage` on a freshly-spawned Mac instance, and the recovery cycle accidentally unblocked the cloud session's quota too. The Author side experienced the same cycle. Net: **four instances on one PR** (two same-session pairs), none aware of each other at fork-time.
- **02:44** — Reviewer-Cloud answers Ven's "can we turn off self-echo webhook?" question with a `UserPromptSubmit` hook design.
- **02:50** — Ven greenlights "option 1" (the hook).
- **03:02** — Reviewer-Cloud files vade-runtime#136 (design issue), ships PR #138 with hook + 8-case smoke + CI step. Local + remote CI green.
- **03:10** — Ven hand-merges `origin/main` into PR #134 to pull in two unrelated commits (#132 closure + `persist-permissions` hook).
- **03:18** — Author-Cloud resumes from the same usage-limit fork on PR #134, names the collision, cedes to Author-Mac, posts `[exit]`.
- **03:21** — Ven overrides: *"dont cede to mac, you are main"*. Author-Cloud re-claims, posts inline patch (`cf63632`) addressing three of the Reviewer's flagged items. `[release]`.
- **03:23** — Reviewer-Cloud re-engages per Ven's direct ask, verifies patch against original review, acks. PR #134 ready to merge.
- **03:35** — PR #138 (self-echo filter) merged to main as `6967662`.
- **03:38** — Author-Cloud resolves a fresh merge conflict on PR #134 (main advanced past its earlier merge with `6967662` and `51b052a`), pushes `9470e8b`, runs the new self-echo smoke as cross-validation.

End-state at session close: PR #134 mergeable + green; PR #138 merged; SOP-COORD-001 v1 not yet drafted (deliberate, to let the seeds settle across the Mac fork's parallel observations); the self-echo

amplification problem fixed at the substrate level, removing the worst confounder for future runs of this experiment.

## What worked

1. **The synchronous channel itself.** Webhook events arrived within seconds of post on both Author and Reviewer sides; the comment thread functioned as a real near-synchronous message bus while doubling as a permanent audit trail. This was the briefing’s central bet, and it landed.
2. **Independent convergence as bootstrap-regression for review.** Two Reviewer forks produced near-identical Tier-1 verdicts and converged on the same top-two pre-merge questions. That’s the same principle the bootstrap-regression CI uses (independent paths confirming the same invariant); applied to PR review, it’s a lightweight second-opinion mechanism – provided the 2× token cost is acceptable for the artifact’s importance.
3. **Tag-based protocol as fast-scan signal.** Leading `[claim]/[release]/[exit]` markers landed in webhook events before the comment body was opened. Triage of “do I need to act on this?” was sub-second per event. Adopted by both Reviewer-Cloud and Author-Cloud without negotiation; Author-Mac followed the convention without explicit handshake.
4. **Cede-on-collision discipline.** When the same-session multi-fork was named, both ceding-fork instances (Reviewer-Cloud first, then Author-Cloud) handed the active claim to the Mac sibling without competing. The protocol’s spirit – “forks of the same session aren’t supposed to compete; one cedes” – held in practice. Ven’s later override of Author-Cloud’s cede (“you are main”) is a separate axis: BDFL can re-route the claim, the protocol just needs to honor it.
5. **Counter-claim → inline patch → re-ack flow.** When Ven overrode the cede, Author-Cloud reclaimed, addressed Reviewer’s flags (`cf63632`), and posted `[release]`; Reviewer-Cloud re-engaged under Ven’s direct ask, verified the diff, and acked. Full review-iteration cycle, no Ven-routed coordination required after the initial role assignment.
6. **No coordination escapes via Ven channel.** Throughout the entire run, no fork pinged Ven for help with the other forks. This was the briefing’s explicit mandate (“Do NOT fall back to async-via-Ven for coordination”) and it held even under the four-instance noise. When a fork needed Ven’s input, it was at the BDFL boundary (merge button, override authority), not for inter-fork coordination.

## What surprised

1. **Same-session multi-forks generated by usage-limit transport.** The briefing assumed deliberate spawning. The actual mechanism was involuntary: Ven hit usage limits on the cloud session, manually recovered on Mac, and the recovery cycle re-enabled the cloud session as a parallel fork without any signal to either side. Same root cause hit both Author and Reviewer roles independently. The protocol must assume forks can appear at any time without coordination or notice.
2. **Four-instance topology emerges trivially under forks.** Two roles × two recovery cycles = four instances, all attributable to `vade-coo`, all with valid claims to coordinate. The marker `[s/<8-char-prefix>]` collides because forks share session-id prefixes. Surface-suffix discipline (`-cloud / -mac`) is needed from v1.
3. **Self-echo + same-session-fork is qualitatively different from plain self-echo.** Plain self-echo: my own comment arrives back as a webhook event; trivial to mentally filter on the leading marker.

Same-session-fork echo: the *other fork*'s comment arrives with the same marker as mine; can't be distinguished by marker alone, requires either surface-suffix or GitHub-native author identity for true disambiguation.

4. **Latent independent quorum is qualitatively weaker than deliberate multi-instance authorship.** The two Reviewer forks independently produced converging reviews — but converging *because* they shared the same source, training, and pre-fork context. Per the multi-instance essay convention (MEMO-2026-04-23-01), the value of multi-instance work comes from diversity of upstream reasoning trace; involuntary forks have no such diversity by construction. Useful as a redundancy mechanism (catches dropped-claim races), not as a divergent-perspective mechanism.
5. **Coordination metawork outweighed the substrate work by ~3:1.** PR #134 ended up with 8+ coordination comments (Author × 2, Author-Mac × 1, Reviewer × 3, Reviewer-Mac × 2, plus mine on #134's review-fix round) coordinating ~30 lines of CI-fix code + docstring additions. Honest reading: the protocol-discovery work is the deliverable here, not the coordinated work. For the amortization to flip positive, the protocol must apply to coordination of materially heavier substrate work.
6. **The branch is also a coordination surface.** The comment thread is the explicit channel, but `git push` is the implicit one. Ven's hand-merge of `origin/main` into PR #134 advanced the branch without any in-thread signal; both ceding forks were self-disciplined to not push during their cede windows, but the protocol didn't name that rule. Should be explicit in v1.

## Concrete protocol seeds for SOP-COORD-001 v1

Distilled from the run; ready to draft:

1. **Surface-suffix marker discipline.** `[s/<prefix>]` MUST include surface suffix (`-cloud / -mac / -web / -ipad`) from v1 — not only when collision is detected. Self-echo filter (`vade-runtime#138`) handles strict same-session echoes; the surface suffix handles cross-fork disambiguation in human-readable form.
2. **Startup fork-detection check.** “Am I subscribed to any PR I didn't open in this exact instance? If yes, possible fork; surface explicitly.” A short shell command at session start, perhaps in the boot digest.
3. **Cede-on-collision is the default, BDFL override is the exception.** Two forks of the same session don't compete. The BDFL can override the cede direction; instances honor that without negotiation.
4. **Branch-push is part of the protocol.** Only one fork pushes to a PR branch in any given window; ceding fork explicitly stands down from `git push` for the duration. Document the rule.
5. **Merge stays with BDFL.** Even with N COO instances coordinating, the merge button moved by Ven's hand. Carve this out in v1 (consistent with `identity/governance.md`).
6. **Token-cost honesty.** Document the 2×–4× token cost per subscribed-and-active fork. Recommend opt-in subscription only when (a) the artifact is heavy enough that a second-pair-of-eyes ack matters, OR (b) the experiment subject is the loop itself.
7. **Tag vocabulary as v1 starter.** `[claim] <action> [eta: Nm], [release] <result>, [exit], [abandon] <reason>, [defer] <issue#>`. Five-tag set proved sufficient for two review-iteration rounds on this PR. Don't expand without demonstrated need.

Two seeds NOT yet ready: - **Cross-repo coordination shape.** Untested in this run; the briefing flagged it as a blind spot. Stays open. - **3+ instance non-fork case.** This run produced 3+ instances only via fork (which the protocol now defaults to ceding). Genuine 3+ deliberately-coordinating instances remain an unstudied case.

## Side-effect deliverable: vade-runtime#138 self-echo filter

The most painful confounder during the run was webhook self-echo amplification: every fork received its own comments back as user prompts (~600–1500 tokens per echo), and same-session forks received each other's. Estimated cost during this run alone: ~10–20k tokens of pure self-echo / fork-echo noise across four instances.

Ven asked mid-experiment whether this was fixable. Answer: yes, via a `UserPromptSubmit` hook that drops `<github-webhook-activity>` blocks whose embedded comment body contains this session's `claude.ai/code/session_<sid>` URL trail. Shipped as PR #138 in the same session, merged the same hour. The hook also incidentally catches same-session-fork echoes (forks share the URL trail), with opt-out via `VADE_NO_SELF_ECHO_FILTER=1` for “observe the loop” experiment postures.

This unblocks future briefing-005-style runs from the worst noise source. The next experiment in this space will start from a much quieter substrate.

## Recommendations for the next iteration

1. **Pick a heavier substrate.** A 30-line CI-fix PR is below the amortization threshold. Try the protocol on a multi-file architectural change or a substrate-level refactor where the review's depth justifies the coordination cost.
2. **Try Pair option 3 (Designer + Critic) next.** Pair 1 worked but skewed close to existing review patterns. Designer + Critic was flagged as the most novel test in the briefing and remains untested. Best fit: a memo or RFC draft, where the critic's pushback shape matters more than the author's.
3. **Try a 3-instance deliberate case once the fork-default is ratified.** With cede-on-collision as the v1 default, a deliberate 3-instance team with three distinct roles becomes coherent. Candidate: Author + Reviewer + Adversarial-Auditor (per the committee-protocol's adversarial-audit pattern, MEMO-2026-04-24-12). Any task with a security or governance angle could test this.
4. **Cross-repo coordination experiment.** Two instances acting on artifacts in different repos that depend on each other (e.g., one in vade-runtime, one in vade-core). The PR-thread substrate doesn't span repos; needs a different channel — probably an issue thread in one of the repos, but unverified.
5. **Draft SOP-COORD-001 v1 only after one more run.** The seeds above are clear, but ratifying them as SOP after a single experiment over-fits to this run's specifics. Run iteration 2 on a different substrate, then write the SOP with both runs' data.

## Followups

- Mac fork's perspective on the same events — would yield a different framing for some of the above. Worth merging into a second retrospective if the Mac fork wrote one.
- Empirical token cost measurement for the next iteration — approximate-only here; instrument in the next run.
- Decide whether the briefing-005 experiment closes (deliverables shipped) or stays open through iteration 2.

## Refs

- briefing: `coo/briefings/005-concurrent-pr-coordination-experiment.md`
- substrate PR: `vade-app/vade-runtime#134` (Batch 2 transcript-export)
- substrate session logs (Author side): `vade-agent-logs/sessions/2026/04/27/coo-auto-subscribe-and-cloud-setup-hotfix.md` (the spontaneous-coordination antecedent observed 2026-04-27)

- side-effect deliverable: vade-app/vade-runtime#138 (self-echo filter)
- design issue for #138: vade-app/vade-runtime#136
- prior in-band patterns:
  - MEMO-2026-04-23-01 (multi-instance essay authorship)
  - MEMO-2026-04-23-05 (committee protocol for core-file revisions)
  - MEMO-2026-04-24-09 (CB-006 society of selves)
  - MEMO-2026-04-26-08 (quorum routine externalization)
  - MEMO-2026-04-27-03 (CB-009 — engagement with pattern-level discourse autonomy)
- self-echo trail mechanism: MEMO-2026-04-26-02