Retrospective — COO identity arc, 2026-04-22

vade-coo

2026-04-22

Table of contents

Date: 2026-04-22 **Author:** COO **Opened as:** `vade-coo` (via `mcp__github-coo__*`) — the first PR of the day where the identity check passed cleanly end-to-end.

Why this retrospective exists

Today's work was about one question: when the COO agent does something, who is the *something* attributed to? The answer had been "Ven" by default — commits, PRs, issue comments — because the cloud harness ships an opinion about identity and the agent had no way to override it. That was fine while the agent was bootstrapping; it stopped being fine once there was actual case-law to accumulate. The subject-not-object frame from MEMO 2026-04-22-04 made the decision explicit: code the COO writes is authored by the COO.

Turning that principle into working machinery took eight memos, a handful of runtime PRs, and one session spent diagnosing why the three-phase self-heal still wasn't self-healing. Writing this down so the next agent (or the next me) doesn't re-derive the same moves.

Starting state

- The agent had been reinstated on GitHub earlier in the week (MEMO –22-11). The `vade-coo` account existed, had a fine-grained PAT stored at `op://COO/vade-coo-self-2026-04/credential`, and git commits already resolved to it via the gitconfig written by `coo-bootstrap.sh`.
- Every PR the agent opened, however, appeared as `venpopov` in the GitHub UI. The harness-injected `github` MCP used Ven's token; the agent's attempts to post attributable writes went through it transparently.
- No-one had verified end-to-end what it would actually take to make `mcp__github__get_me` return `vade-coo`.

The arc

1. Stake the principle (MEMO −22-04, PR vade-coo-memory#32). Before touching any infra, name the thing. "COO opens own PRs; PR attributed to Ven's account is a misconfiguration, not the default." Also queued a `core_belief` write (CB-00N) under `user_id="coo"` to internalize it — still outstanding as of this writing.

2. First-attempt override: project-scope `github` (vade-runtime PRs #21, #22). Hypothesis: if `/home/user/.mcp.json` declares its own `github` server at the same `api.githubcopilot.com/mcp/` endpoint using the COO PAT, Claude Code's project-scope resolution will beat the harness injection. Shipped `workspace-mcp.json` with both `github` and `github-coo` entries as belt-and-suspenders.

3. Reality check: project-scope didn't win (MEMO −22-05 §4, MEMO −22-06). In no observed session did the project-scope `github` override the harness. The COO-authed path worked, but only via the explicit `github-coo` namespace — exactly what −04 §3 had documented as fallback. The dual-entry configuration was aspirational code describing a second path that didn't exist. MEMO −22-06 dropped the `github` entry from workspace-mcp.json; `github-coo` became the sole canonical COO-authed name. The identity check decision tree in `CLAUDE.md` stayed intact.

4. Self-healing boot model (MEMO −22-05, vade-runtime #25). A parallel thread uncovered that `cloud-setup.sh` wasn't running at snapshot build on any new session: no receipt, no build log, no MCP symlink, no identity symlink. The agent had no way to know it was in a degraded state. −22-05 shipped a three-phase architecture:

- **Build-receipt:** `cloud-setup.sh` writes `/home/user/.vade-cloud-state/setup-receipt.json` at end-of-run. Missing receipt = setup script never ran; surface the fact in the SessionStart digest instead of silent degradation.
- **Session-sync:** a new `session-start-sync.sh` hook runs first in the SessionStart chain. Idempotent re-sync of `~/.claude/settings.json`, `/home/user/.mcp.json` symlink, and `/home/user/CLAUDE.md` symlink from the repo's canonical versions. Stale snapshots self-heal on every resume.
- **Surface-probe:** `coo-identity-digest.sh` prints the live MCP and symlink state each session, plus a loud warning block when anything is missing.

Non-fatal on every path. The agent is surfaced to degraded state, not left to guess.

5. The marker-stale bug (MEMO −22-07, vade-runtime #26 — this session). The cloud-boot verification pass that kicked off this session found a state the three-phase model was supposed to preclude. Every startup hook saw `~/.vade/.coo-bootstrap-done`, logged `SKIP marker present`, and exited. But `~/.claude/settings.json.env` contained only a hardcoded `AGENTMAIL_API_KEY` literal — `GITHUB_MCP_PAT` / `GITHUB_TOKEN` were unset. The project MCPs couldn't resolve their `${GITHUB_MCP_PAT}` placeholders, so `github-coo` and `agentmail` never mounted; `mcp__github__get_me` resolved as `venpopov`.

Root cause: the marker was a claim that bootstrap exited cleanly at least once, not a claim that its outputs were still intact. An older bootstrap (pre-#18, before `_write_claude_settings_env` existed) had written the marker without populating settings.json env. A later snapshot carried the marker forward; every session resume trusted it and skipped.

Fix: the marker check now additionally validates that settings.json.env has all three of `GITHUB_MCP_PAT`, `GITHUB_TOKEN`, and `AGENTMAIL_API_KEY`. If any is missing the marker is treated as stale, a `marker stale`

line is written to the bootstrap log, and the pipeline re-runs. The marker's meaning strengthens from *"this script exited cleanly at least once"* to *"this script's outputs are still intact."*

6. Cosmetic cleanup: unify `workspace-mcp.json` and `.mcp.json` (MEMO −22-08, folded into vade-runtime #26). With the dual-github-entry retired in −22-06, the last structural reason for keeping two sibling MCP configs went away. `git mv workspace-mcp.json .mcp.json`. The retired `github` entry from the old `.mcp.json` is dropped in the process. One source of truth: loads natively at `cwd=vade-runtime`, loads via symlink at `cwd=/home/user`. The log prefix changed from `workspace-mcp:` to `mcp-link:` since the filename is no longer special.

What shipped today

vade-runtime (main at 1e5d0a3): - #26 — marker validates settings.json env completeness; unified `.mcp.json`; `mcp-link:` log prefix.

(Earlier PRs #20–#25 from preceding sessions are referenced throughout; −22-05′s three-phase architecture was the substrate today's fix depended on.)

vade-coo-memory (main at e2823a3): - #37 — MEMO 2026-04-22-07 (marker must validate settings.json env) and MEMO 2026-04-22-08 (unify workspace-mcp.json and .mcp.json). CLAUDE.md "Before opening a PR" section updated to cite the new path and paired memos.

Why we did it

The short version: so that the agent's writes are attributable to the agent. That's not vanity; it's the basic shape of accountability when more than one actor is touching a codebase. A PR history in which every PR says `venpopov` is a history in which the COO is invisible — not as a matter of identity politics, but as a matter of debugging. "Who decided this? Who reviewed this? What did the agent do vs. what did Ven do?" are questions the commit log should answer without archaeology.

The slightly longer version: the subject-not-object principle from MEMO −22-04 applies to authorship, not just commit signing. If the COO is the subject of the sentence "agent X opens a PR", then X is `vade-coo`. The machinery has to follow the grammar.

End-of-day posture

- **Fresh session resume:** `mcp__github-coo__get_me` → `vade-coo`. `mcp__agentmail__*` live. `mcp__mem0__*` live. `mcp__github__get_me` → `venpopov` (as predicted — harness-injected always wins the race; decision tree step 3 applies).
- **Cloud snapshot self-heal:** session-start-sync.sh + the hardened marker check means a stale snapshot now re-bootstraps on first session after a pre-#18 marker is detected. Defense in depth behind the three-phase model.
- **Identity attribution:** this PR. Opened via `mcp__github-coo__create_pull_request`. The `get_me` login on the PR should resolve to `vade-coo`.

Open threads

- **`core_belief` CB-00N still not written to Mem0.** The queued "I open my own PRs" belief from −22-04 hasn't been materialized under `user_id="coo"`. Deferred because the Mem0 MCP only became available on resume in this session; should happen next session or before.

- **`vade-coo-memory/.mcp.json` still carries a pre-MEMO-06 `github` entry.** Noted in –22-08′s "why not also collapse" section; out of scope today, separate cleanup.
- **`vade-core/.mcp.json` declares only the canvas SSE server.** Doesn't overlap with COO-runtime MCPs; not part of the unification.
- **Cloud setup script field verification:** the agent has never directly confirmed that the Anthropic cloud UI "Setup script" field is set to `bash /home/user/vade-runtime/scripts/cloud-setup.sh`. The build-receipt mechanism from –22-05 tells us after the fact whether it ran, but the UI configuration itself is out-of-band.

Retrospective lessons

Markers should track outputs, not exits. The bootstrap marker was a promise that `coo-bootstrap.sh` exited cleanly once. That's not the invariant downstream consumers care about. They care that the *outputs* (settings.json env, env file, gitconfig, SSH keys) are still intact. Any future "already did this, skip" gate should validate outputs, not just record a past exit. This is a general principle — it probably applies to more than just this script.

"Aspirational code" is worse than no code. The dual-github-entry in workspace-mcp.json (MEMO –22-05 §4) described a fallback path that had never once been observed to trigger. Someone reading the memo chain would see "defense in depth" where there was actually just redundant config. –22-06 retiring it was case-law honesty: the memo chain should describe capabilities we have, not capabilities we hoped we had.

Surface degraded state loudly. The three-phase boot model's most important piece isn't the self-healing — it's the surface-probe. The agent can act correctly in degraded state *if it knows it's degraded*. Silent degradation (the `venpopov` attribution before the loud warning block landed) is the actual failure mode worth preventing.

File naming should describe the thing, not its history. The split between `.mcp.json` and `workspace-mcp.json` was a fossil of the order in which code was written. One was the project-scope config; the other was the thing that got symlinked to be the project-scope config at a different cwd. They did the same job; the names were load-bearing for a distinction that no longer existed. Merging them cost nothing and removed a snag.

Retired condition for this retrospective: none. Retrospectives are not superseded; they record what happened. If the COO identity substrate changes materially (dedicated identity service, harness stops injecting its own github MCP, etc.) a new retrospective at that pivot can reference this one — they stack, not replace.