

OPAUDIT 012 — Adding a blog to the fleet never actually made it follow the others

Audit ID	2026-09-07-012
Report date	2026-09-07
First observed	2026-09-07, surfaced while repairing the cron outage (OPAUDIT 011). Fleet on 0.7.656D.
Theme	Discovery is not connection — an invariant that was never implemented.
Surfaces	The 25-site hub/spoke all-to-all peer-follow mesh; onboarding a new blog to the fleet.
State	FIXED — CONFIRMING The mesh is fully repaired and watched working; the permanent auto-reconciler shipped as v0.7.657D but is not yet deployed.
Watched live?	Yes, for the repair — all 600 peer relationships now exist, first full audit found zero missing, and backfill was proven end to end (see §6). The permanent reconciler awaits deployment.
Reporter	Sean + Codex (established the exact missing-edge pattern, ran the live recovery, shipped the reconciler).
Related	OPAUDIT 011 (the cron outage that exposed this); OPAUDIT 006 (the intentionally-stale external relay row, not a fleet peer).
Publication	Public. No exploitable detail.

1. SUMMARY

The fleet is meant to be an all-to-all network: every blog follows every other, so a post on one reaches all the rest. It wasn't. Blogs had been added to the fleet's *roster* — the list of who exists — but adding them never actually established the follow relationships. Being on the list is not the same as being connected, and nothing ever did the connecting.

2. WHAT A HUMAN SAW

The fleet was no longer all-to-all. One blog's profile showed a single accepted outgoing follow where it should have had two dozen. Fleet Stats reported 449 outgoing follows across 25 sites — but a true 25-node all-to-all mesh needs 600 directed relationships before counting any external follows, so the network was materially incomplete.

3. ROOT CAUSE

Connecting or adding a node updates the multisite *discovery roster*, but it does not reconcile the actual following ledger. The multisite action labelled `RUN JOBS — ALL SPOKES` refreshes roster data only — it never

establishes peer follows. This was an *unimplemented invariant*, not a mysterious deletion: the system assumed “on the roster” meant “following,” and nothing ever made that true.

4. EVIDENCE AT DIAGNOSIS

The missing-edge pattern was exact, not random. Twenty-one established spokes each lacked the same four actors (the hub plus three of the newest spokes); those four newer nodes each had only one outbound relationship and lacked all 24 peers. That accounts for exactly 180 missing directed edges, and $600 - 180 = 420$ peer edges plus unrelated external follows is consistent with the observed 449. A precise boundary like that points at an operation that was never run for those nodes, not at random loss.

5. WHAT WE DID

Repair: live recovery sent only the *missing* follows, through each site’s own authenticated CMS — adding directed relationships only, never deleting an external follow, skipping self, validating each actor, and advancing gradually to avoid a burst of Accepts and catalogue backfills. **Permanent fix (0.7.657D):** the cron now adds one missing fleet peer per tick from the hub’s canonical roster; a Follow sends its exact queue row; and handshake responses (Accepts) are prioritised ahead of large content backfills so a new relationship completes promptly instead of waiting behind a catalogue.

6. VERIFIED LIVE — AND WHAT REMAINS

The first repaired handshakes were accepted within seconds. A controlled backfill test using two peers (one and two posts) confirmed both accounts and their historical posts appeared in the hub’s home feed — proving Follow, Accept, backfill-job creation, paced delivery, inbound ingest, and reader display all work end to end once the follow exists. All 600 directed peer rows now exist, and the first full audit found zero missing on every site. What remains: the permanent auto-reconciler is shipped as `v0.7.657D` but not yet deployed, and some Accepts are still pending on 0.7.656D (inline-Accept fallbacks queued behind large backfills — the exact thing 657D reprioritises). So this stays **FIXED — CONFIRMING** until 657D is live and every site reports zero missing and zero pending fleet peers.

7. EYEBALL NEXT

After 657D deploys, verify all 25 sites report zero missing and zero pending fleet peers — and do not mistake the intentionally-stale external relay row (OPAUDIT 006) for a fleet peer. Separately, and kept separate on purpose: prove *why* some pre-existing edges were absent beyond the unimplemented-reconciler explanation. An earlier PhotoFriday leave-cleanup could remove individual ambiguous relationships before 657D, but that does not explain the exact four-node boundary; preserve the evidence rather than guessing.

8. THE DOCTRINE THIS INCIDENT CREATED

Discovery is not connection. Updating a roster of who exists is not the same as establishing the relationships between them — and an invariant you assume but never implement fails silently and

looks exactly like data loss. If a network is supposed to be all-to-all, something has to actively reconcile it to that shape, and prove it did.

SnapSmack — DING DONG BELL operability ledger — snapsmack.ca/ding-dong-bell.php · Operability audits record whether a feature actually performs its job, proven not claimed. Security audits live separately at snapsmack.ca/buzzers.php.