

OPAUDIT 002 — The site had picture containers but no post containers

Audit ID	2026-09-07-002
Report date	2026-09-07 (compiled from the incident record)
First observed	Foundational — present from early on; named sharply in the CMS architecture review (August 2026).
Theme	A missing data-model idea, invisible to a code review, that quietly broke ownership, counts, comments and identity.
Surfaces	Publishing; comments/likes; post counts; public profile; the fediverse identity of a post.
State	WATCHED WORKING The going-forward fix is confirmed. One tail of old comments is still being moved across (see §7).
Watched live?	Yes — Sean confirms new posts are born correctly and the counts/profile read right.
Reporter	Sean + Claude; the architectural gap was named in the CMS architecture review.
Related	OPAUDIT 001, 009 (federation identity of a post depends on this).
Publication	Public. A structural/operability lesson, no exploitable detail.

1. SUMMARY

This is the deepest one, and the least visible. For a long time a published photograph was stored as a bare *image*, not as a *post*. That sounds like an internal detail, but with no post to hang things on, a photo's ownership, its date, its comments and likes, the collections it belonged to, and its identity out on the fediverse had no single home. Everything that needs “the post” to point at had nothing to point at.

2. WHAT A HUMAN SAW

Comments imported from Flickr came unstuck from their photos. Post counts collapsed to near-zero on photoblogs. A profile with thousands of live, federating photos reported “no posts yet.” A repair tool kept having to convert loose photos into posts, and sites kept drifting back to the broken state after it was run.

3. ROOT CAUSE

Legacy solo publishing wrote a photograph as a row in the images table with no post record behind it (the post link was empty). So anything that counted or displayed “posts” either missed this content entirely or had nowhere to attach related data. Worse, the Flickr comment importer wrote the *image*'s id into the column meant for the *post*, which detached every imported comment from the thing it was a comment on.

4. EVIDENCE AT DIAGNOSIS

Total-post counts were badly wrong on imported installs — a photoblog with roughly ten thousand imported posts counted as near-zero, because the counter looked only at the posts table while the content lived in the images table with no post wrapper. The public profile had the same blind spot: it listed only grouped posts and never the standalone images where photoblog content actually lives.

5. WHAT WE DID

Two things. Going forward, posting a new photo now creates a real post in the background, built to match exactly what the repair tool and the carousel poster already produced — so nothing else on the site can tell the difference, and web addresses never change (federation stays stable). Behind that, a proper post-model inventory and a transactional conversion, plus a single shared definition of “a post” used by the dashboard, the heartbeat and the fleet rollup so the counts can’t drift apart again. The public profile was made mode-aware so it counts and lists both standalone images and grouped posts.

6. VERIFIED LIVE

New posts are born correctly, counts read right, and profiles show the real content. Confirmed by Sean on live sites.

7. EYEBALL NEXT

A tail of older comments are still being moved across from the photo to the post. They are real people’s words, so that migration is being done carefully rather than rushed. It is the one part of this not yet finished.

8. THE DOCTRINE THIS INCIDENT CREATED

Define what a “post” is once, and make everything use that one definition. A code review reads the code that exists; it cannot see a missing idea. The most expensive bug here wasn’t a wrong line — it was an absent one, and the fix was to give the whole system a single home for the thing it kept assuming was there.