

OPAUDIT 003 — Background jobs ran inside web requests and took whole sites down

Audit ID	2026-09-07-003
Report date	2026-09-07 (compiled from the incident record)
First observed	~0.7.358 (paced delivery); the visitor-triggered variant removed at 0.7.639D (September 2026).
Theme	Slow work running where a person is waiting.
Surfaces	Whole-site availability; photoblog landing pages; fediverse delivery.
State	WATCHED WORKING The offending feature was removed, and the outages stopped.
Watched live?	Yes — Sean; sites stopped timing out once the feature was pulled.
Reporter	Sean + Claude.
Related	OPAUDIT 001, 004 (delivery correctness).
Publication	Public. Availability post-mortem, no exploitable detail.

1. SUMMARY

To be kind to other servers, fediverse deliveries were paced — the code paused between sends. The mistake was *where* it paused: inside a live web request. Slow work belongs nowhere near a page a person is waiting for, and this proved it by taking entire sites offline.

2. WHAT A HUMAN SAW

Whole sites timed out with a Cloudflare 524 error, even on a plain page load with nothing to do with federation. A related version leaned on ordinary visitors to trigger background work, which made photoblog landing pages hang behind network maintenance and 524 as well.

3. ROOT CAUSE

Each pause inside a web request held a web-server worker hostage for the duration. Inbound federation deliveries then piled up more sleeping workers; the pool starved; and once there were no free workers, every request — including a plain homepage — had to wait, and Cloudflare gave up with a 524. Using visitor page-loads as the trigger for slow work had the same shape: a real person’s page load was doing the waiting.

4. EVIDENCE AT DIAGNOSIS

The 524 appeared on requests that did no slow work themselves, which points at worker-pool starvation rather than any single slow page; and inbound federation traffic correlated with the pile-up of sleeping workers.

5. WHAT WE DID

The fix was not to patch the pause — it was to **remove the feature that leaned on page loads**. All paced, measured delivery now happens only in a command-line scheduled job that has no web timeout and refuses to run from the web at all. The visitor-triggered background jobs were removed entirely; buttons and the hub's own scheduler do that work now. File backups moved to the desktop side, and the site owner can always pull files straight off the server by FTP or SFTP.

6. VERIFIED LIVE

The outages stopped once the feature was pulled. Confirmed by Sean on live sites.

7. EYEBALL NEXT

Any remaining place that still assumes a page view is a safe time to do slow work.

8. THE DOCTRINE THIS INCIDENT CREATED

Web requests never sleep. If a job can be slow, it runs somewhere that can fail on its own without a visitor noticing — never inside the request a person is waiting on. A visitor is not a worker.