*A Sovereign Builder's Case That the Vector Store Is a Finding Aid, Not a Collection*

EVELYN CARO · SEPTEMBER 28, 2026 · POSITION-PAPER

DOI: 10.5281/zenodo.24356906

# The Cache Is Not the Corpus™

Download PDF

**Nineteen Deaths, Zero Losses, and the Asymmetry That Decides What an Archive Is**

*A Sovereign Builder's Case That the Vector Store Is a Finding Aid, Not a Collection*

**Evelyn Caro** September 28, 2026

---

## AI Collaboration Disclosure

This paper was developed in collaboration with AI. The author directed the research, structure, and argument. AI assisted with drafting, organization, and reference verification. All claims, decisions, and conclusions are the author's own. The events described in Section 2 occurred on the author's own hardware and are documented in the author's retained logs.

---

## Abstract

Between the night of September 27 and the morning of September 28, 2026, the author's data ingestion pipeline ran through nineteen complete cycles of start, wedge, kill, and relaunch. It ingested nothing. It lost nothing. This paper uses that night to argue a distinction most RAG practice blurs: the vector store is a cache, not a corpus. The cache is a derived, loss-

compressing, machine-bound representation of sources that live elsewhere. Nineteen abrupt deaths proved the asymmetry that makes this matter: nothing can be rebuilt from the cache, and everything can be rebuilt from the corpus. The direction that keeps everything is the archive.

---

## I. The Thesis

A vector store is not an archive. It is a cache. A derived, machine-bound representation of a corpus that lives somewhere else.

The conversion is **lossy**. The embedding step throws information away on purpose, and the throwaway is permanent. Exact wording goes. Document structure goes. Anything the model cannot represent goes. You cannot rebuild the corpus from the cache. The information is gone.

The reverse is not true. You can rebuild the cache from the corpus at any time, because the corpus keeps everything.

That asymmetry is the whole paper. One direction loses information. The other keeps it. The direction that keeps everything is the archive. The direction that loses things is a finding aid. Mistake one for the other and you have not built a memory. You have built a summary of a memory and called the summary the record.

The engineering literature says the same thing in its own vocabulary. Kleppmann and colleagues treat the server-held copy in cloud applications as "merely a cache that is subordinate" to the primary data, and argue the primary copy belongs with its owner (Kleppmann, Wiggins, van Hardenberg, & McGranaghan, 2019, DOI:10.1145/3359591.3359737). Gilbert and Lynch proved that a system under partition must choose between consistency and availability (DOI:10.1145/564585.564601). A single-machine RAG pipeline makes that choice silently, and usually makes it wrong: it treats its index as durable when the index is, at best, rebuildable.

Archival practice settled this decades ago. The OAIS reference model (ISO 14721:2025) requires that archival information survive independent of the systems that render it. The National Museum of African American History and Culture (NMAAHC) Freedmen's Bureau collection, 1.7 million images, holds because the collection and the finding aid are separate things. The collection survives the index. Every time.

**Where it stops:** the vector store is useful. It is fast, it is queryable, and it earns its place. But it is the finding aid. The corpus is the collection. Budget accordingly.

---

## II. Evidence: Nineteen Deaths, Zero Losses, Zero Gains

Between 21:33 on September 27 and 08:07 on September 28, the author's ingestion pipeline ran through nineteen complete cycles of start, wedge, kill, and relaunch. The interval was metronomic: roughly thirty minutes, every cycle, all night. The pipeline is version 5. It is subprocess-isolated, checkpointed through a sidecar file, and monitored by a watchdog process that kills any run whose log goes stale. The logs are retained and quotable; the evidence ledger at qaevelyn.github.io/site/evidence.json indexes the public receipts.

| Observation | Value |
|---|---|
| Cycles, 21:33 to 08:07 | 19 |
| Interval between restarts | ~30 minutes, consistent |
| Watchdog stale timeout | ~1798 seconds, every kill |
| Conversations in source corpus | 127 |
| Conversations ingested at first preflight | 72 |
| Conversations ingested at final preflight | 72 |
| Conversations lost or corrupted | 0 |

Three findings, each verifiable in the logs.

**First, the corpus held.** Nineteen abrupt kills, and the source corpus of 127 conversations was intact at every preflight. The checkpoint never moved and never lied. Nothing was lost, nothing was corrupted, nothing was written twice. This is not luck. It is the design: idempotent writes, meaning the same record written twice lands the same as written once, and a checkpoint written before the work it describes. The pipeline was built on the assumption that it would die, and the assumption was correct nineteen times.

**Second, the pipeline made no progress.** The same design that prevented loss failed to advance. No conversation beyond the 72 completed after the first cycle. The run wedged at a

consistent point, roughly thirty minutes in, and the watchdog killed it before any child process finished. This is a livelock. A crash fails loudly and stops the system. A livelock keeps the system available while it makes no progress. The distinction matters because livelock defeats the common defense of "but it's still running." **Running is not progress.** Nineteen restarts registered as nineteen relaunches in one log and as zero in the other.

**Third, the distinction between cache and corpus is what made nineteen deaths survivable.** Had the vector store been the system of record, nineteen kills would have been nineteen opportunities for a partial write to corrupt it. Instead, every death cost a day of progress on constrained hardware, and the pipeline still has not moved. The cache absorbed the failure. The corpus never noticed. The corpus does not pay the bill; the builder does.

One correction belongs in this section, because a record that hides its own errors is publicity, not documentation. The author's session records from the night described the first four wedge-kill cycles as "resilience proven." That reading was wrong. It counted restarts instead of completions. The full log, reviewed the next morning, showed the sidecar pinned at 72 across nineteen cycles. The correction was appended to the author's dated records the same day. The method this paper asks of its reader is the method the correction demonstrates: **read the whole log before you characterize the system.** A cache tells you what it was last given. Only the corpus, read completely, tells the truth.

**Where it stops:** the checkpoint design prevented data loss and cannot solve a livelock. Isolation protects the records; it does not advance the work. The thirty-minute wedge is a specific, reproducible failure with nineteen logged samples, and diagnosing it is engineering, not archival theory. The archive survived the night. The pipeline did not move.

---

## III. What the Builder Owes the Record

The failure documented here is not exotic. It is the ordinary condition of sovereign AI on consumer hardware: a builder, one machine, an Intel quad-core i5 at 1.1 GHz, eight gigabytes of memory, and a pipeline that must be interrupted for class, for meetings, for life. The archival question is not whether the pipeline is fast. It is whether the record survives the interruptions. It does, because the builder put the record first and the cache second.

This inverts the usual priorities. Most RAG guidance optimizes retrieval quality and treats durability as an implementation detail. The events of September 27 and 28 suggest the reverse ordering for any archive of consequence: checkpoint before you write, write idempotently,

expect the kill, keep the corpus outside the process. The retrieval can be rebuilt overnight. The record cannot be rebuilt at all if it was never separate from the system that died.

The correction published in Section 2 is the third part of the practice. A builder who is also the archivist is subject to the same conflict the labs carry: the party that made the error is the party positioned to hide it. The resolution is the same for the sovereign builder as it would be for the lab. Date the correction. Append it. Let the record show the miss.

**Where it stops:** this paper does not diagnose the thirty-minute wedge. It documents that the wedge occurred nineteen times and cost the record nothing. The diagnosis will be published as an addendum to this paper under the author's appended-not-retro-edited rule. The archive holds. The finding aid is unfinished and rebuildable at any time. That is not the failure of the system described here. That is its design working.

---

## References


- Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2), 51-59. DOI:10.1145/564585.564601
- Kleppmann, M., Wiggins, A., van Hardenberg, P., & McGranaghan, M. (2019). Local-First Software: You Own Your Data, in spite of the Cloud. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward! '19)*, 154-178. ACM. DOI:10.1145/3359591.3359737
- ISO. (2025). *Space data and information transfer systems — Open archival information system (OAIS) reference model* (ISO 14721:2025).
- Garcia, A. sqlite-vec vector search extension documentation. https://github.com/asg017/sqlite-vec
- SQLite documentation: Write-Ahead Logging and the PRAGMA synchronous durability caveat.
- National Museum of African American History and Culture (NMAAHC). Freedmen's Bureau records, 1.7 million digitized images, with separate published finding aid.
- Caro, E. (2026). Session logs, ingest logs, and watchdog records, September 27-28. Private repository; cited excerpts reproduced in Section 2.


---

© 2026 Evelyn Caro. All rights reserved. A Mirror of My Becoming™ — https://qaevelyn.github.io For licensing inquiries: evelyn.caro.cloud@gmail.com



---

# ADDENDUM — 2026-09-29 — THE WEDGE DIAGNOSED: A LOGGING FAILURE, NOT A SYSTEMS FAILURE

*Appended per this paper's Section III commitment and the author's appended-not-retro-edited rule. The original text above stands as published.*

The thirty-minute wedge documented in Section II has been diagnosed and resolved. The result, stated first: **the same pipeline completed all 55 remaining conversations in a single unattended run, including a conversation of 5,163 chunks and 4,646,264 characters. The sidecar reads 127 of 127. Zero conversations were lost.**

The diagnosis is not what the nineteen deaths suggested. The pipeline was not failing. The pipeline was **working silently**. Each conversation's embedding phase — one model call per chunk, roughly one second each — produced no log output while it ran. The watchdog, built to kill any run whose log goes stale, read the silence as a wedge and killed a healthy process every thirty minutes. Conversation 73 alone held 3,615 chunks and 3,250,607 characters: roughly an hour of silent, legitimate work. The first remaining conversation was also the largest in the corpus. The livelock was a machine telling the truth too quietly for its own guardian to hear.

The fix was two lines of honesty: the log now announces each conversation ("chunking N chars"), its size ("embedding N chunks"), and its heartbeat ("chunk 11/3615") as it works. The watchdog has not killed a healthy process since. The wedge that consumed three days and nineteen cycles was solved not by rearchitecting but by making the record match the reality — which is this paper's argument, holding at a level the author did not expect when writing it. **A system that reports truthfully cannot be killed for working.**

Three corrections to Section II's evidence table, per the same standard that built it:

| Observation | Section II said | The complete record shows |
|---|---|---|
| Largest conversation | 3,615 chunks (assumed largest) | 5,163 chunks, 4,646,264 characters |
| Cycles before resolution | 19, unresolved at publication | 20th run completed all 55 |
| The thirty-minute mark | "a fingerprint" of the failure | The watchdog's staleness timeout — nothing failed at thirty minutes; silence reached its limit |

The paper's thesis is unchanged and reinforced: the cache process failed twenty times; the corpus of 127 conversations was intact at every preflight. The finding aid died and was rebuilt at will. The collection never noticed. The author's pipeline, published as a free and open artifact, carries this war story in its README — battle-tested, not beta-tested.

Where it stops: the addendum documents the fix; it does not claim permanence. The watchdog's timeout was calibrated to a chatty script and may require retuning for silent stretches longer than any yet observed. The next failure will be logged with the same honesty as this one.

← All White Papers