Case study: one caption slot destroyed ~2,100 prior results

Key takeaways

1. One slot per figure made every re-run destructive by construction.
2. Generated content has no test; silent quality drift needs audits.
3. Cost attribution lies: $11.65 here, the real spend bills to siblings.

209 papers, and a rule: no code that runs

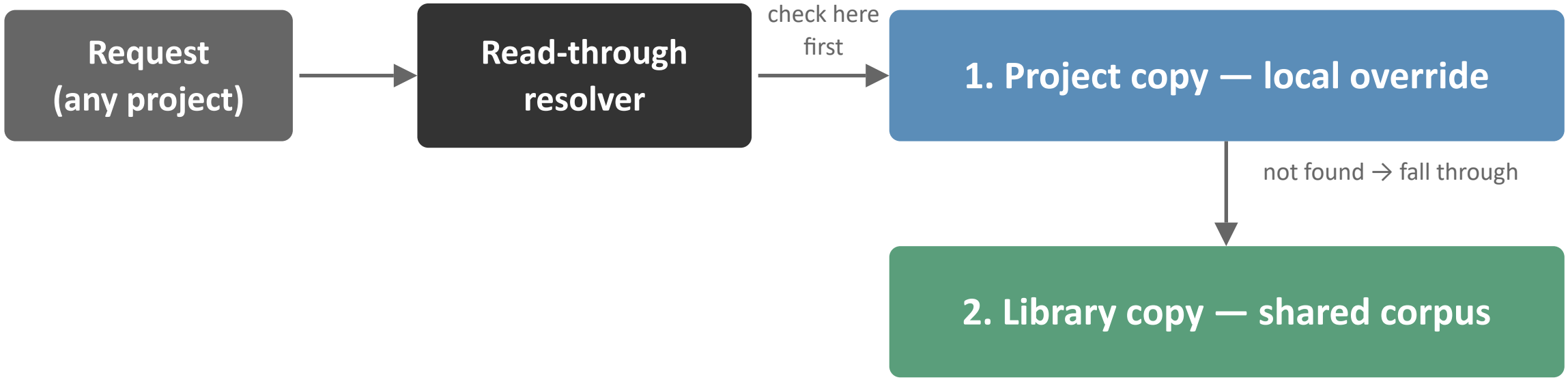
The corpus paper-tools queries · as of 2026-05

- The data sibling to paper-tools — separated on purpose
- Holds papers, knowledge bases, writing-voice data, and agent definitions
- Looks tiny on its own ($11.65); most agent work that touches it is billed elsewhere

What paper-library does

- Hold one shared corpus that multiple paper projects can read from
  - 209 systems-research papers across 8 sub-fields, each kept as PDF + markdown + per-figure captions
  - Domain knowledge sheets (50–100K tokens each) summarizing the state of each sub-field
  - Writing-voice data distilled from 714 prior reviews and 49 published papers
- Serve content through a read-through resolver
  - Any project asks for a resource; the resolver checks the project first, then the library
  - Nothing is copied into projects during writing — every command sees the current state
  - At submission, a snapshot freezes only what was cited (the archival record)
- Stay code-free, on purpose
  - Founding rule: ‘separation of content from tooling — no scripts, no Python’
  - Broken twice on purpose: agent definitions (declarative) and one IEEE fetcher (274 lines)

The read-through resolver



At submission, a snapshot freezes only what was cited — the archival record. Nothing is copied into projects while writing.

Architecture — five content categories

- The paper corpus
  - 208 PDFs + 232 markdown conversions + per-paper figure tarballs
  - Hard part — keeping PDF, text, images, and captions consistent across re-conversions
- Domain knowledge bases
  - Each built from 25–40 web searches; one per sub-field
  - Hard part — keeping them current as the field moves
- Writing-voice and style data
  - Distilled from 714 prior peer reviews and 49 published papers
  - Hard part — capturing the voice without copying it verbatim
- Agent role definitions
  - Declarative reviewer / paper-QA / context-builder — lives with the data, not code
- VLM-agreement benchmark harness
  - Do two vision models agree on a caption? Built to catch the next destructive swap

By the numbers — content

Metric	Value
Tracked files	888
PDFs (the paper corpus)	208
Markdown files	232 (paper conversions + knowledge + style + agents)
Python LOC	274 (one IEEE fetcher — the only executable code)
Papers indexed	209 in papers.toml
Commits	84 over 2.7 months

As of 2026-05. This is a data repo — the interesting size signal is PDFs and markdown, not lines of code.

By the numbers — tokens and cost

Metric	Value
Active days (ccusage)	6
Total tokens	14,250,000
Total cost (Claude Code)	$11.65
Most expensive day	2026-03-05 · $9.60 · 82% of lifetime cost
Models used	Opus 4.6, Haiku 4.5

As of 2026-05. Most agent work that touches this corpus runs from paper-tools or per-paper projects and is billed there — paper-library's attribution badly under-counts effort.

The figure-description problem

- What was at stake
  - Every figure in 204 papers had a text caption — ~2,100 captions total
  - Captions were stored inline in each paper's markdown (one slot per figure)
  - Months of accumulated VLM work — irreplaceable without rerunning every paper
- Why a re-run was inevitable
  - Newer vision models keep arriving (Gemini Flash 2.5 → 3.1 Flash Lite → …)
  - Each new model claims better numeric reading; you want to use it
  - Re-running the pipeline overwrites the old caption with the new one
- The schema invariant nobody had written down
  - Each figure had exactly one caption slot — there was no room for two captions side-by-side
  - Re-running was thus destructive by construction, not by mistake

The swap that destroyed two months of data

- What happened (commit 8f144dd, 2026-04-11)
  - Single command: ‘redescribe all 204 paper figures with gemini-3.1-flash-lite-preview’
  - Pipeline did exactly what it was designed to do — overwrite every caption
  - ~2,100 prior captions (Gemini 2.5 Flash + 2.5 Pro) gone from the working tree
- How it was noticed
  - Not by a test failure — by reading the new captions and noticing quality dropped
  - Discovery happened two days after the overwrite
- The verdict that landed in HISTORY.md
  - ‘Until now, VLM swaps in the library were leaps of faith.’
  - A destructive change in a data repo has no rollback if you only learn after the fact

Three fixes in one week

- Recover — pull pre-swap captions back out of git history under a ‘git-recovery’ tag
- Restructure — swap the one-caption slot for a sidecar with one entry per model
- Prevent — a benchmark harness so the next VLM swap is measured before it ships

Old schema — destructive by construction

```
# in paper.md (inline):

![Figure 3](figures/fig3.png)
*Caption: a bar chart showing
throughput vs. thread count ...*

# one slot per figure;
# new run overwrites old caption
```

New schema — append-only sidecar

```
# paper.descriptions.json
{
  "figure_3": [
    {"model": "2.5-flash",
     "src": "git-recovery"},
    {"model": "3.1-flash-lite",
     "text": "..."}
  ]
}
```

source: HISTORY.md @ 2026-04-11–13; commits 1403d7f + benchmark + sidecar

Story — when ‘Figure_1’ matches ‘Figure_10’

- The bug, hidden in a one-line helper
  - Code that linked captions to figures used substring matching on the figure number
  - ‘Figure_1’ matched ‘Figure_1’, ‘Figure_10’, ‘Figure_11’, ‘Figure_12’ — all four
  - Fixed upstream with word-bounded matching + zero-padded numbers (Figure_01)
- What it actually cost — for weeks, silently
  - 92 papers had figure descriptions silently orphaned — readers saw the wrong caption
  - Nothing failed — no test, no exception, no audit alarm
  - Surfaced only when someone investigated one specific bad figure by hand
- The pattern — silent data-quality bugs in agent-generated content
  - Generated content has no contract a test can check against
  - Bugs surface only when something forces a comparison — like the benchmark harness landed the same week

Story — agents absorb their first project

- Workshop paper snuck past explicit selection criteria
  - SELECTION_CRITERIA.md says workshop papers are excluded — the agent had read it
  - One workshop paper was added anyway; removed later in a one-line commit
  - The agent read the criteria for the immediate task, not as a permanent filter
- Agent definitions had baked-in paths from their first user
  - Reviewer / context-builder agents had osdi26-specific filesystem paths inside them
  - They had been written while working on one paper and never abstracted afterward
  - A deliberate cleanup pass stripped the project-specific assumptions
- A first read-through found 11 papers were the wrong article
  - Bulk DOI resolution wrote the adjacent entry — mind-sosp21 actually held Snoopy
  - Title-token containment (11 bad below 0.45, 313 good above 0.55) is now verify-references
  - Lesson — a paper nobody read is a paper nobody verified; audits catch what tests can’t

Feature creep + stalls

- Feature-creep cases — agent built it, project ripped it out
  - A hand-rolled batch conversion shell script, maintained across a secrets migration, then deleted when the main tool absorbed batching
  - Two .gitignore entries added on autopilot — copied from sibling-project habits — then removed once it was clear the library plays a different role
- Stalls all share one shape — ‘the data needs code I have to write upstream first’
  - Citation-count metrics deferred because fetching them requires code that doesn’t belong in a content-only repo
  - A sidecar-pruning bug deferred because the fix lives in paper-tools, not in the data

Key Takeaways

1. Cost attribution lies — $11.65 here, but the real spend lives in sibling repos that consume the corpus
2. A data schema is a contract about what you can change later — ‘one slot per item’ is a destructive-by-construction invariant
3. Generated content has no test that catches silent quality drift — periodic audits and agreement benchmarks are not optional