

# Podman in Production: The Book, and the Pipeline Behind It

Wolfgang Kerschbaumer ([hello@sysop.cafe](mailto:hello@sysop.cafe))

July 11, 2026

*Releasing a Podman book with an AI publishing team.*

**Human in the Loop**

sysop.cafe

<https://sysop.cafe/2026/07/podman-book-toolchain/>

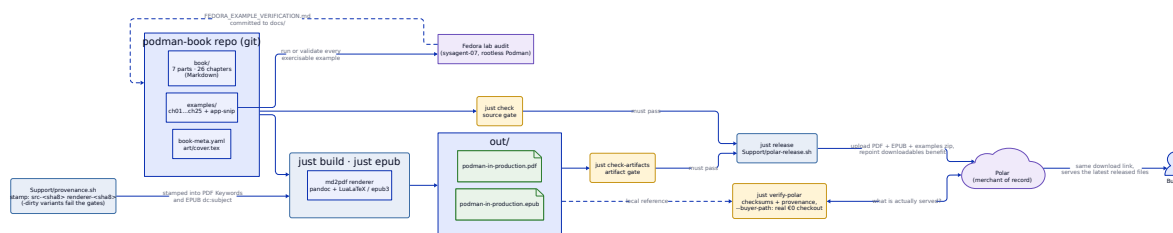


Figure 1: The whole toolchain, from Markdown sources to the buyer’s download

My first encounter with Podman back in 2020 was involuntary. A customer with air-gapped RHEL 7 machines that we needed to deploy a containerized microservices stack on. Installing third-party software like Docker on them would have been a major bureaucratic hurdle, so I looked at what we could use that came built in.

The tooling had its quirks — at least the way we were using it — but the general daemonless architecture of the Podman ecosystem just felt right from the beginning. And needless to say, the current version of said customer project is still running on Podman today. Not only that, but many more instances followed.

Being a big fan of good technical books, my first thought was: someone should write a book about this fine piece of technology. So I started with outlines, wrote the first chapters, bought a domain, started over again, life (and work) got in the way, and so on. Of course in the meantime quite a few Podman books were released, and I think I’ve bought most of them.

Two things gave my book project enough momentum to finally reach a release version now: my personal interest in the upcoming (now already released) version 6 of Podman, and the advances in frontier LLMs.

AI was part of every step of the publishing process. The diagram below shows the whole toolchain: Markdown sources and runnable examples pass a source gate (`just check`), the build produces a provenance-stamped PDF/EPUB, an artifact gate (`just check-artifacts`) checks the result, and `just release` ships everything to Polar; `just verify-polar` checks what Polar actually serves, down to a real buyer checkout with `--buyer-path`.

Verification of examples consists of a fast local gate on every `just check`, plus a live audit on a Fedora lab host (136 recorded executions, failures triaged into real defects vs. lab artifacts) whose report ships inside the buyer zip.

For the build pipeline, `md2pdf` (a CLI Markdown-to-PDF converter that “I” wrote independently of the book) discovers chapters, merges metadata, pre-renders diagram fences, then runs `pandoc` once: LuaLaTeX with a KOMA book template for print (cover included in the same run so metadata and outline survive), `epub3` for the reflowable edition.

The release flow: one pinned commit, no flags. Both gates green, provenance stamps verified, PDF + EPUB + a freshly rebuilt examples zip uploaded, then the downloadables benefit is reposted before superseded files are deleted, so existing buyers’ links keep working.

And yes, it feels like cheating. But creating the publishing pipeline and tweaking the

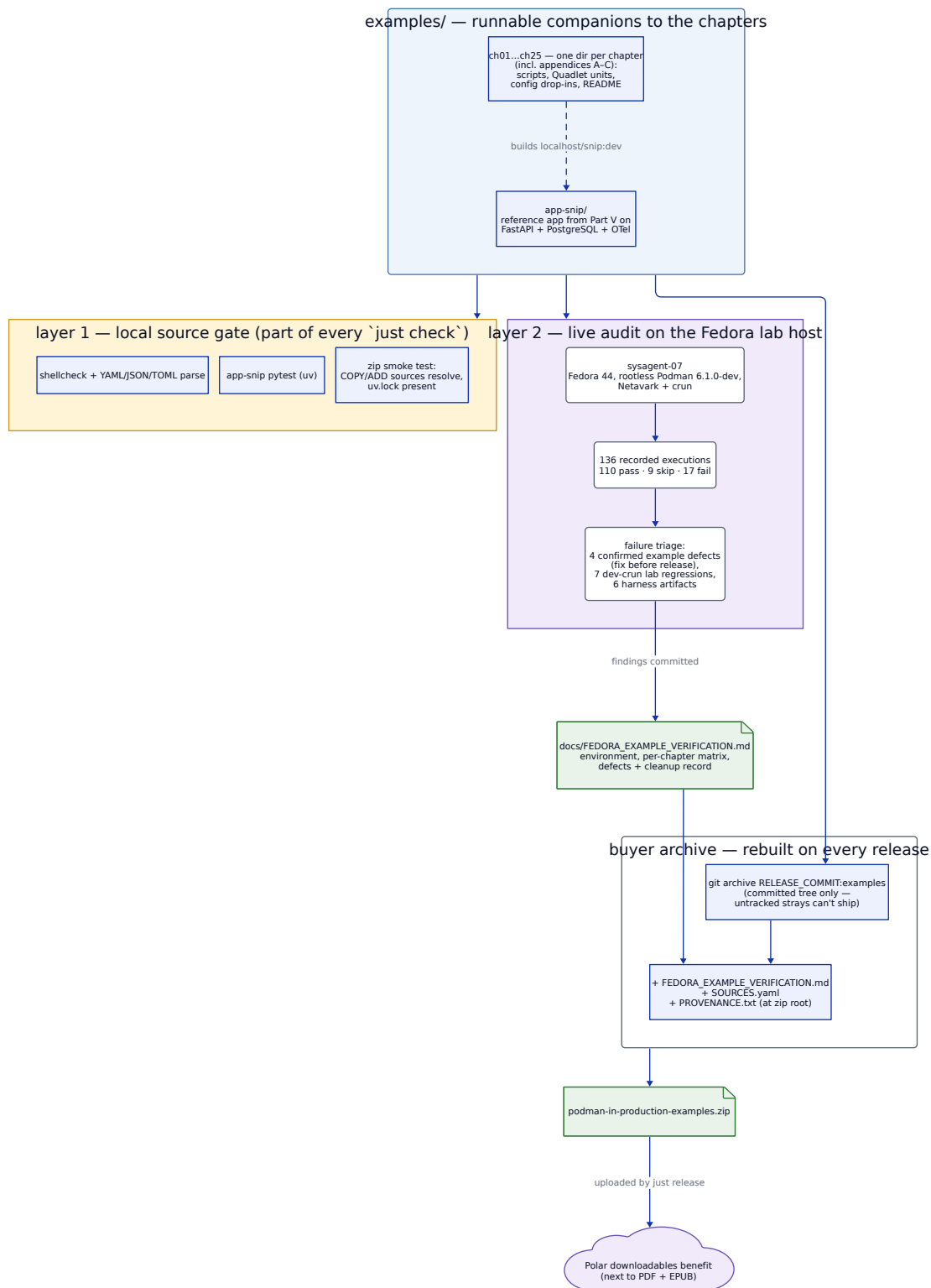


Figure 2: How the runnable examples are verified and shipped

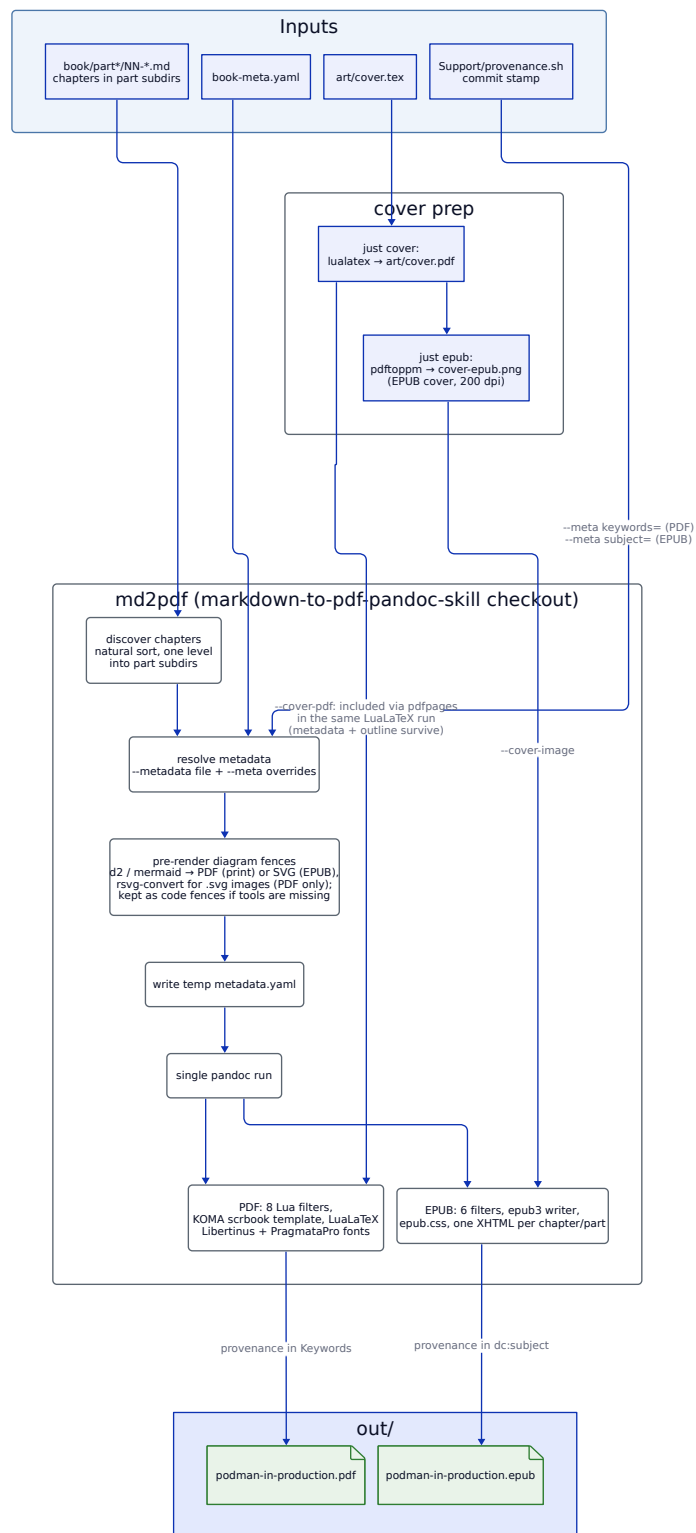


Figure 3: Inside the md2pdf build pipeline

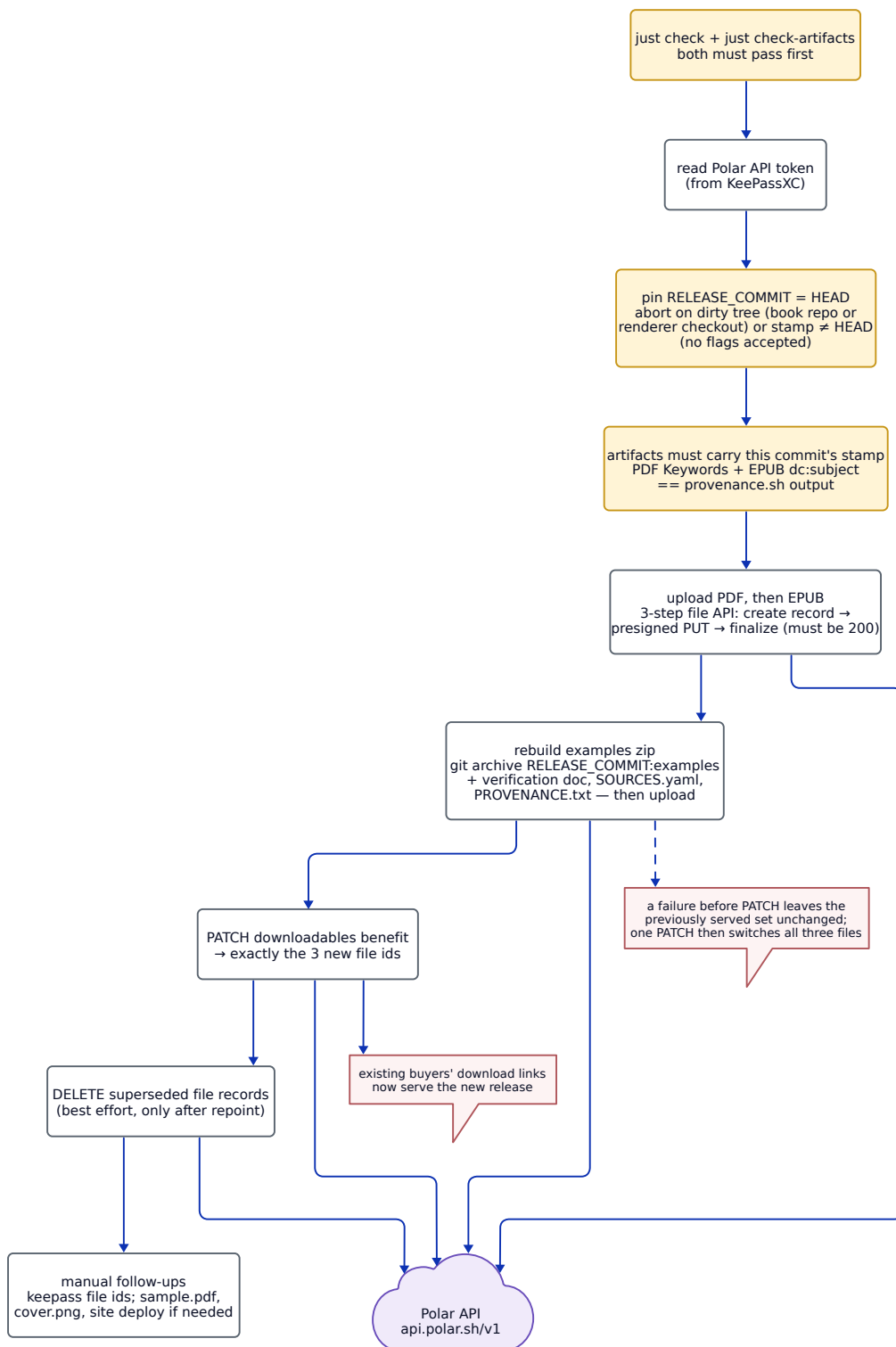


Figure 4: The release flow to Polar

typography alone would have cost me weeks instead of days.

If you are interested in publishing high-quality technical writing (let's postpone the discussion of whether anyone still needs that in an AI age), feel free to ping me.

And of course, if you want to take a look at sample chapters of the book, you can find them on the book's landing page: [podman-book.com](https://podman-book.com)