

# A template/ approach to Reproducible Generative Research

Research infrastructure that documents itself — a case study for meta-science  
and science-integrity teams



# The problem

---

- Research groups publish claims about their own tools and pipelines that no one outside the team can re-check.
- Manuscripts describing infrastructure go stale the moment the infrastructure changes underneath them.
- Reproducibility efforts usually stop at the data and code layer — the documentation layer is still hand-maintained and drifts silently.



# What `template_template` is

- A manuscript that is generated FROM the repository it describes, not written ABOUT it from memory.
- Every metric in the paper — module counts, test counts, pipeline stages — is computed live by `template_template`'s own introspection code.
- One of 19 public exemplar templates in the same monorepo, each independently tested and coverage-gated.

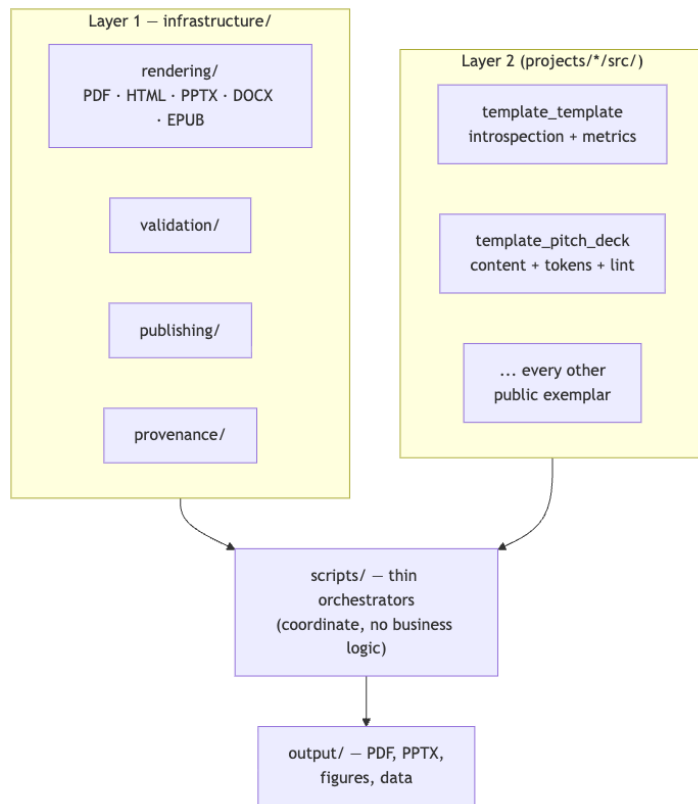


# How it works

- `src/template_template/introspection.py` scans `infrastructure/` modules, the pipeline DAG, and the public project roster.
- `metrics.py` turns that scan into a dictionary of `${variable}` values; `inject_metrics.py` substitutes them into the manuscript text.
- Re-running the pipeline regenerates the paper from current repository state — the citation and the artifact cannot drift apart.



# Two-layer architecture, at a glance



## Proof, not adjectives

# 130 tests

99.37% coverage on `template_template`'s own source — the same gate every exemplar in the repo must pass.



# Already public

- Concept DOI 10.5281/zenodo.20419007, 9 durable publication records already on file: Zenodo, GitHub, PyPI (sandbox), IPFS, Software Heritage, GitHub Pages, Netlify, Hugging Face, OSF.
- Licensed Apache-2.0 — the introspection code, not only the prose, is available for a science-integrity team to audit directly.



# Why this matters for science integrity

- A funder or reviewer can clone the repository and regenerate the exact figures and counts cited in the manuscript, on their own machine.
- The same no-mocks, coverage-gated testing discipline that verifies the code also verifies the claims made about the code.
- The pattern generalizes across the whole projects/templates/ folder — all 19 exemplars, not just one cherry-picked example — follow the identical thin-orchestrator, two-layer architecture.



# Cite this deck

- This pitch deck (`template_template_pitch_short.pdf`) is its own exemplar in the same monorepo — it has its own source, tests, and citable identity, separate from the DOI of the project it pitches.
- This deck's own DOI: `10.5281/zenodo.21281509`.
- Every slide also carries a QR code to its own standalone, citable page — see the deep-linking mechanism described later in the medium and long versions of this deck.



# The ask

- Adopt this pattern for your own methods papers: bind every reported number to the code that produced it, not to a hand-typed table.
- Pilot it on one existing manuscript in your group before committing to a full rewrite.
- Happy to pair with your team through a first regeneration cycle, if useful — no prior engagements to point to yet, this pattern is newly forkable.
- Open to funding conversations and scoping a consulting engagement for a deeper or custom integration — no prior engagements exist yet; this is an invitation to talk, not an established offering.



# Questions

Research infrastructure that documents itself — a case study for meta-science and science-integrity teams

