Academic Writing Protocol — Terra / FMP Edition

Purpose: практическое руководство «от препринта до монографии и диссертации» для исследователей AIUZ Terra, выстроенное в духе Nullo / PLT / Fractal Metascience Paradigm. Полностью совместимо с Zenodo, arXiv, Springer, Elsevier, IEEE, Scopus, Google Scholar, Zotero, Mendeley и основными стилями цитирования (APA7, IEEE, Chicago).

---

0. Введение — почему это важно

Этот протокол — рабочая карта. Он превращает фрагменты исследования (чистые идеи, симуляции, код) в публикационные артефакты, пригодные для индексирования и цитирования. Принцип — минимизация ручных шагов у Оператора: весь рабочий поток описан как воспроизводимая машина, которую можно упаковать и архивировать.

---

1. Ключевые концепции (Nullo / PLT / FMP)

- **Nullo:** каждая публикация должна содержать свою нулевую точку — чёткое, формально оформленное утверждение того, что считается новым знанием.
- **PLT (Post-Lingua Trace):** фиксирует мета-слои (аннотации, слепки экспериментальной среды, FPG), которые остаются «после языка» — машинно-читаемые артефакты (JSON-LD, provenance).
- **Fractal:** всё публикуемое должно иметь многомасштабный representation: краткая нотация (abstract), статью (paper), монографию (book), и материал-репозиторий (code+data).

---

2. Универсальная структура: от идеи к DOI

1. **Problem Ingestion (Canonical Statement)**
2. `ingestion.json`: title, canonical_text, source_url, problemscope, deadline, eligibility.
3. **Fractal Problem Graph (FPG)**
4. `FPG_<id>.jsonld` — nodes/edges with `supported_by` links and checksums.
5. **Prototype & Evidence**
6. Code + raw data + container image or environment spec.
7. **Claim Formulation**
8. Template `claim.md`: statement, scope, failure modes, evidence links, 3 supporting refs.
9. **Artifact Assembly**
10. `article.tex` (IMRaD) + `supplementary.zip` (data + code + Dockerfile) + `.zenodo.json`.
11. **Registration & Archival**
12. Deposit on Zenodo (or institutional repo) — mint DOI.
13. **Sync to GitHub & Ledger**
14. Create GitHub release with DOI in body; record in Terra Proof Ledger (JSON).
15. **Outreach**
16. Executive Summary (1 page), Press Brief, ORCID update, ResearchGate/OSF mirrors.

---

3. File formats & metadata (practical cheatsheet)

- **Primary text:** LaTeX (`.tex`) with `bibtex` (`.bib`). Use `apalike` or `apalike2` for APA7-like, `ieeetr` for IEEE.
- **Data:** CSV / Parquet / HDF5 + `README_DATA.md` describing schema.
- **Code:** GitHub repo with `Dockerfile` or `environment.yml` + `run.sh`.
- **Provenance:** `provenance.json` containing hardware, software, seeds, checksums.
- **Zenodo manifest:** `.zenodo.json` containing metadata (creators, keywords, license).
- **Ledger:** `terra_proof_ledger_TERRA-PROOF-YYYY-NNN.json` (doi, sha256, authors).

---

4. Citation styles quick-matrix

- **APA7** — default for cross-disciplinary and Zenodo descriptions.
- **IEEE** — for engineering/computer science journals.
- **Springer/LNCS** — conferences/books; use `splncs04` class.
- **Elsevier** — use `elsarticle.cls`.

Provide BibTeX entries for each source; include DOI fields.

---

5. Templates (ready-to-use)

Inside this protocol are templates for: `ingestion.json`, `FPG.jsonld`, `claim.md`, `article.tex (IMRaD)`, `.zenodo.json`, `terra_proof_ledger.json`, `github_release_manifest.json`, `README_DATA.md`, and `Dockerfile`.

(These templates are available as separate files in the package.)

---

6. Submission checklist (operational)

- [ ] ingestion.json present
- [ ] FPG JSON-LD created
- [ ] Code & Dockerfile included
- [ ] Raw data + README_DATA.md included
- [ ] Article.tex + references.bib (≥3 refs per main claim)
- [ ] Supplementary.zip (all artifacts) included
- [ ] Zenodo deposit completed (.zenodo.json used)
- [ ] Terra Proof Ledger updated with DOI and checksum
- [ ] GitHub release created with DOI
- [ ] ORCID updated (add work)

---

7. Automation (CI) — practical recipes

- **GitHub Actions** to: run tests, build container, create release draft, and call Zenodo API (if token stored in GH secrets).

- **CI job skeleton** included: `.github/workflows/publish.yml` with steps for building `supplementary.zip`, generating checksums, and creating draft releases.

---

8. Indexing (Scopus / Google Scholar / Zotero)

- Zenodo DOI → provides indexability in Google Scholar automatically.
- For Scopus: ensure journal or book publisher is indexed; for books, request ISBN and publisher metadata.
- Provide metadata (title, authors, abstract, keywords) in English and Russian for broader indexing.

---

9. Ethics & Licensing

- Default: CC-BY-4.0 for documents; MIT for code; data licensing must consider privacy.
- Human/animal studies: include ethics approval docs in supplementary.

---

10. Appendix — Rapid collection of useful commands

- `pdflatex fmp_monograph.tex && bibtex fmp_monograph && pdflatex fmp_monograph.tex && pdflatex fmp_monograph.tex`
- `sha256sum TCPP_v1.0_release.zip > checksums.sha256`
- `curl -H "Authorization: Bearer $ZENODO_TOKEN" -H "Content-Type: application/json" -X POST "https://zenodo.org/api/deposit/depositions" --data-binary @zenodo_upload.json`

---

11. Fractal checklist (Nullo-style)

For each artifact, confirm: - Zero point: explicit new claim (1 sentence) - Traceable evidence: links to raw logs / datasets - Reproducible environment: container or environment.spec - Ledger entry: DOI + SHA256

End of Academic Writing Protocol — Terra / FMP Edition