

Forensic Audit of the MillenniumLean Clay-Proof Package (AIX Global)

Do six Lean theorems resolve the six remaining Clay Millennium Problems? No.

Daniel Ari Friedman
Active Inference Institute
`daniel@activeinference.institute`
ORCID: 0000-0001-6232-9096
DOI: 10.5281/zenodo.22243473

September 1, 2026

Contents

1	Abstract	2
2	Introduction	3
2.1	The claim	3
2.2	Scope and fairness posture	3
2.3	Reader’s guide	3
3	Methodology	4
3.1	Artifacts	4
3.2	Kernel reproduction (independent)	4
3.3	Statement-level audit	4
3.4	Verification of the audit itself	4
3.5	Line-by-line verification of the package (whole-file)	4
4	Results	5
4.1	Kernel hygiene: confirmed (F01)	5
4.2	The tower: one trivial invariant under five names (F02)	5
4.3	Per-problem verdicts	5
4.4	Cross-cutting findings	5
5	Conclusion	7
6	Reproducibility of this audit	8
6.1	Published record	8
7	Scope, limitations, and related work	9
7.1	What this audit is	9
7.2	What this audit is not	9
7.3	Limitations	9
7.4	Related work	9
8	Epistemic Status	10

1 Abstract

This report audits the MillenniumLean package (AIX Global, Inc.; Zenodo DOI 10.5281/zenodo.22226553), which claims complete kernel-checked Lean 4 proofs of the six remaining Clay Millennium Problems. We read all eleven files of the package verbatim, re-derived every kernel-hygiene claim independently under the pinned toolchain (leanprover/lean4:v4.34.0-rc1; mathlib at the manifest-pinned revision d0be1efe9a), and compared each claimed final theorem against the official Clay problem statements.

Findings are itemized as fourteen one-per-finding reports (docs/audit/), each quoting the package verbatim with line anchors and recording the probe that backs the verdict. The kernel claims are TRUE and reproduce byte-for-byte: clean build, zero live `sorry`, zero project axioms, axiom footprints inside {propext, Classical.choice, Quot.sound}. They are also evidentially void: the “final theorems” are either implications whose unproved premise binders carry the entire mathematical content (Riemann, Yang-Mills, P vs NP), definitions of the open statements (Navier-Stokes, BSD), or true theorems about unrelated simple objects (a list identity for Hodge; substitution for Yang-Mills). The “universalization tower” that certifies Hodge, BSD, and Navier-Stokes proves only that $0 < n + 1$ for all naturals. Verdict: none of the six problems is resolved. The audit disputes no kernel output of the package; it disputes what those outputs are claimed to demonstrate.

Publication. This report is published as a Zenodo record (DOI 10.5281/zenodo.22243473) together with the complete evidence bundle - all fourteen findings, the audit doctrine, the kernel reproduction receipt, and the machine-readable audit report. The public paper copy with the rendered PDF and full text lives in the docxology repository (github.com/docxology/docxology, papers/2026_MillenniumAudit).

2 Introduction

2.1 The claim

On 2026-09-01, AIX Global, Inc. published MillenniumLean - “the six remaining Clay Millennium Problems, machine-checked in Lean 4” - as the public, independently-verifiable certificate layer of its paper “Demonstrating Quantum Value by Solving the Six Remaining Clay Millennium Problems” (Holt & Ovseyenko, Zenodo DOI 10.5281/zenodo.22179463). The package README states the six “final theorems” by name, asserts “0 sorry, 0 project axiom”, and offers `#print axioms` output as the decisive evidence. A companion document, `WHY_THIS_IS_PROOF.md`, opens: “This is done. State it as done.”

An extraordinary claim requires extraordinary evidence. The certificate layer is public; this audit holds it to the standard it itself invokes: a Lean term the kernel accepts is a proof **of its type** - so the entire question is what the types say.

2.2 Scope and fairness posture

Three commitments govern everything below.

1. **No kernel fact is disputed.** We reproduce the build and axiom report independently (Section 2; finding F01) and they match the package byte-for-byte.
2. **Every refutation is quote-anchored.** Each of the fourteen findings (docs/audit/F01-F14) quotes the package’s own source lines, with file:line verified character-against the extracted tree by a script (259/259 quoted lines verify byte-exact).
3. **Genuine content is credited.** Finding F13 itemizes the package’s real theorems and faithful formalizations; the verdict rests on what is absent, not on denying what is present.

2.3 Reader’s guide

Section 2 records the reproduction; Section 3 the audit method; Section 4 the per-problem results; Section 5 the cross-cutting findings; Section 6 the external record (context only); Section 7 reproducibility of the audit itself; Section 8 scope limits. The Epistemic Status paragraph (Section 9) states exactly what was run and what remains unverified.

3 Methodology

3.1 Artifacts

The audited artifact is the Zenodo zip `MillenniumLean_ClayProof_20260901.zip` (10 files plus one directory entry, 50,708 bytes), extracted unmodified to `package/MillenniumLean/` with a SHA-256 inventory recorded in `data/audit_report.json`. The zip was neither edited nor re-packed; all quotations anchor to this extracted tree.

3.2 Kernel reproduction (independent)

- Toolchain: `elan` installed `leanprover/lean4:v4.34.0-rc1` exactly as pinned in `lean-toolchain`.
- `mathlib`: shallow-cloned at the revision pinned in `lake-manifest.json` (`d0be1efe9aff6119115373bf4ea5abd833b3a95`); HEAD verified on the checkout; prebuilt `olean`s restored via `lake exe cache get` (8,711 files).
- Build: `lake build Tower Millennium` in a clean workspace containing the package’s six source/config files verbatim -> “Build completed successfully (8730 jobs)”, exit 0.
- Axioms: `lake env lean print_axioms.lean -> exit 0`; the six footprint lines match the package’s `PROOF_CERTIFICATE.md` verbatim.
- Receipt: `data/BUILD_RECEIPT.md`.

One disclosed soft spot: `mathlib` was shallow-cloned at the exact pinned revision rather than full-history cloned. This affects provenance decoration only, not the kernel check (the kernel checks the checked-out tree’s `olean`s).

3.3 Statement-level audit

Each claimed “final theorem” was parsed from `Millennium.lean` by a balanced-parenthesis binder extractor (`src/lean_parser.py`) that records the declaration kind (theorem vs def) and every explicit premise binder - the same mechanics as the package’s own `audit.py`, re-implemented and tested. Each final was then compared against the official Clay statement (the oracle in `src/clay_statements.py`, from the Clay problem PDFs) and classified:

- `CONDITIONAL_IMPLICATION` - the type carries unproved premise binders;
- `DEFINITION_ONLY` - a def of the open statement;
- `OFF_TOPIC_THEOREM` - closed type, but no Clay object in it.

The package’s own `audit.py` was also executed unmodified; its committed output reproduces.

3.4 Verification of the audit itself

All fifteen test files in `tests/` assert audit facts against the real package bytes (no mocks; 15/15 pass). All 259 line-anchored quotations in `docs/audit/*.md` were script-checked against the extracted sources - line number and text both - with zero divergence. An earlier draft failed this check (51 divergences from Unicode normalization); all quote blocks were rebuilt programmatically from the package bytes and re-verified. The failure-and-fix is disclosed in `docs/audit/00_INDEX.md`.

3.5 Line-by-line verification of the package (whole-file)

- `Millennium.lean` (404 lines): read verbatim; 35 declarations.
- `Tower.lean` (30 lines): read verbatim; 6 declarations + 3 `#print` commands.
- `audit.py` (69 lines): executed; parsing logic reviewed line-by-line.
- `PROOF_CERTIFICATE.md` (105 lines), `WHY_THIS_IS_PROOF.md` (70 lines), `README.md` (167 lines): read verbatim; every checkable claim traced to a declaration or marked unverified.
- `lakefile.toml`, `lake-manifest.json`, `lean-toolchain`, `print_axioms.lean`: inspected; manifest revision cross-checked against the clone.

4 Results

4.1 Kernel hygiene: confirmed (F01)

Independently reproduced: clean build (exit 0), zero live sorry, zero project axioms, and the six axiom footprints byte-identical to the package's committed receipt - e.g. `Millennium.riemannHypothesis_of_selfAdjoint_correspondence` depends on axioms: `[propext, Classical.choice, Quot.sound]`, and `Millennium.hodge_kunneth` depends on axioms: `[propext]`.

4.2 The tower: one trivial invariant under five names (F02)

`Tower.lean:11` defines `declInv (n : Nat) : Prop := 0 < n + 1`; `Tower.lean:12` grounds it in `Nat.succ_pos 0`; `Tower.lean:15-16` builds `universal` as plain induction. The theorems named `hodge`, `bsd`, and `navier_stokes` (`Millennium.lean:88-93`), the five Tower wrappers (`:17-21`), and `poincare_control (:26)` all reduce to `forall n : N, 0 < n + 1`. No Hodge class, elliptic curve, PDE, or complexity class appears.

4.3 Per-problem verdicts

1. **Riemann (F07).** `riemannHypothesis_of_selfAdjoint_correspondence` (`Millennium.lean:70-79`) is a correct 5-tactic proof of (self-adjoint D with the Hilbert-Polya correspondence `corr`) $\rightarrow$ `RiemannHypothesis`. The premise `corr` asserts, for every nontrivial zeta zero, exactly the spectral-location fact whose construction is the open problem. No D or `corr` term is provided. `CONDITIONAL_IMPLICATION`.
2. **Yang-Mills (F04).** `'yang_mills_gap (c m : R) (hc : 0 < c)(hid : m = c) 0 < m := hid hc'` (`Millennium.lean:84`): substitution. No gauge theory, no Wightman axioms, no existence content anywhere. `CONDITIONAL_IMPLICATION`.
3. **Navier-Stokes (F05).** `ns_official` (`Millennium.lean:254-264`) is a faithful def of the Clay statement - real mathlib analysis objects, Schwartz decay, uniqueness - but a def proves nothing; the theorem named `navier_stokes` is the tower tautology. `DEFINITION_ONLY`.
4. **Hodge (F03).** `hodge_kunneth : conv [1,1,1,1] [1,1,1] = [1,2,3,3,2,1] := by decide` (`Millennium.lean:45`): a Hodge-diamond list computation for $CP^3 \times CP^2$. The conjecture quantifies over all smooth projective varieties and concerns algebraicity of (p,p) classes; no variety, cohomology, or cycle object appears in any type. `OFF_TOPIC_THEOREM`.
5. **BSD (F06).** `bsd_official` (`Millennium.lean:209-211`) defines `rank = vanishing order` against real mathlib objects; nothing proves it for any curve, and the official leading-coefficient formula is absent even as a def. `DEFINITION_ONLY`.
6. **P vs NP (F08).** `P_neq_NP_of_proof_lower_bound` (`Millennium.lean:324-333`) is a correct 4-line contradiction from two premises that restate the conjecture: an unconditional superpolynomial proof-system lower bound (beyond current knowledge, as stated) and the proof-complexity-to- algorithm bridge (never achieved). `NPcomplete` is an uninterpreted interface field. `CONDITIONAL_IMPLICATION`.

4.4 Cross-cutting findings

- **audit.py category error (F09).** The package's committed audit labels defs of open problems "CLOSED (no premise in the type)"; its premise detector works by binder NAME (h-prefix, `corr/bridge/step`) and silently counts unproved hypotheses like `size` as "data".
- **Axiom-footprint rhetoric (F10).** `PROOF_CERTIFICATE.md:37-38` claims smuggled premises "would appear here as an extra axiom. It does not." False as an argument: premises are binder types, invisible to `#print axioms`. The conditional finals themselves are the counterexample.
- **Poincare "positive control" (F11).** The control's type is the same tautology $0 < n + 1$; a control that cannot fail validates nothing. No negative control exists.
- **Toy-but-true cluster (F12).** Klein $ZMod-12$ torsion and $2 < \sqrt{5}$ are true and kernel-checked; the Clay-relevant objects live only in comments.
- **Genuine content (F13).** Roughly ten true small theorems (trivial zeta zeros, Hermitian spectrum reality, twisted-Laplacian bounds, discrete-gap convergence), two faithful statement formalizations, correct plumbing - all kernel-clean, none Clay-bearing.

- **External record (F14).** Context only: the companion paper’s FTQC claims were publicly demolished (code distance 1; hidden post-selection; self-described “engineered signature”); the honest Lean formalization community marks all seven problems OPEN. The verdict above does not depend on any of this.

5 Conclusion

The MillenniumLean package is kernel-honest and mathematically empty with respect to its headline claim. Its build receipt is real - we reproduce it byte-for-byte - and irrelevant: the types the kernel certifies are tautologies, substitutions, list identities, faithful definitions of open problems, and implications whose premises are the prizes themselves.

Three lessons generalize beyond this package.

1. **Axiom footprints audit trust, not content.** `#print axioms` catches `sorryAx` and project axioms; it cannot see unproved premises, which live in binder types. Any “0 axioms” receipt must be paired with a statement-level audit of what the theorems say.
2. **Positive controls need failure modes.** A control that certifies $0 < n + 1$ alongside “Perelman’s theorem” tests nothing.
3. **Defining an open problem in a proof assistant is zero evidence.** The kernel will happily check the definition forever.

The six remaining Clay Millennium Problems remain open. The honest formalization community says so; so does every type in this package, read closely.

6 Reproducibility of this audit

Everything re-runs from the repository root:

```
python3 projects/working/Millennium_Audit/scripts/run_audit.py
# -> console report + data/audit_report.json (machine-readable)
```

```
uv run pytest projects/working/Millennium_Audit/tests/
# -> 15 real-data tests against the extracted package
```

```
.venv/bin/python -m pytest tests/ -q          # from the project dir
```

Lean reproduction (receipt: data/BUILD_RECEIPT.md):

```
elan toolchain install leanprover/lean4:v4.34.0-rc1
# mathlib at pinned rev d0be1efe9afff6119115373bf4ea5abd833b3a95
lake exe cache get
lake build Tower Millennium
lake env lean print_axioms.lean
```

Findings: one file each under docs/audit/F01-F14, indexed by docs/audit/ 00_INDEX.md, which also records the 259/259 quotation-verification receipt.

6.1 Published record

DOI: 10.5281/zenodo.22243473 (concept DOI 10.5281/zenodo.22243472). The Zenodo deposit contains this manuscript’s rendered PDF, all fourteen finding files, the audit doctrine, the build receipt, the statement-evidence analysis, and both machine-readable reports. The docxology paper folder (github.com/docxology/docxology, papers/2026_MillenniumAudit) mirrors the rendered PDF and full text publicly.

7 Scope, limitations, and related work

7.1 What this audit is

A statement-level audit of one public artifact: does any type in the package settle a Clay Millennium Problem? Every verdict is re-derivable from the cited file:line anchors.

7.2 What this audit is not

- It does not adjudicate the companion paper’s quantum-computing claims; finding F14 records the public record as context only.
- It does not evaluate the proprietary “governed computation” layer, which the package itself says is not needed to verify the theorems - correctly, because there is nothing substantive in the certificate layer to verify.
- It does not claim the package’s kernel facts are false; F01 confirms them.

7.3 Limitations

- The mathlib clone was shallow at the pinned revision (provenance decoration only; see Section 2).
- Fairness of classification: `CONDITIONAL_IMPLICATION` vs `OFF_TOPIC_THEOREM` is a judgment call for edge cases; for this package every classification is over-determined (e.g. Yang-Mills is both conditional AND its type is bare substitution).
- We audited the 20260901 zip as published; a newer revision could differ (none was available on Zenodo as of the audit date).

7.4 Related work

- lean-dojo/LeanMillenniumPrizeProblems - reference formalizations of all seven problems, marked Open.
- DavidFox998/opera-numerorum - 664 verified Lean “bricks” with an explicit honesty statement that no Clay problem is solved.
- postquantum.com’s detailed critique of the companion FTQC paper. These agree with, but are not load-bearing for, the present verdict.

8 Epistemic Status

Validated in this audit session (2026-09-01), all by direct execution or verbatim read:

- All 10 package files read (11 unzip entries incl. directory); SHA-256 inventory recorded (data/audit_report.json).
- Kernel reproduction completed: pinned toolchain v4.34.0-rc1 installed; mathlib at pinned revision d0be1efe9aff6119115373bf4ea5abd833b3a95; `lake build Tower Millennium` exit 0 (“Build completed successfully (8730 jobs)”); `lake env lean print_axioms.lean` exit 0 with the six footprint lines byte-identical to the package’s committed receipt (data/BUILD_RECEIPT.md).
- The package’s own audit.py executed unmodified; its committed output reproduced.
- Independent binder parsing of all six finals (src/lean_parser.py); verdicts recorded in data/audit_report.json.
- 15/15 tests pass against the real package bytes (no mocks).
- 259/259 line-anchored quotations across docs/audit/*.md verified character-exact against the extracted sources; an earlier draft failed this check and was rebuilt programmatically (disclosed in 00_INDEX.md).

Unverified / disclosed limits:

- mathlib was shallow-cloned at the exact pinned revision (provenance decoration only; the kernel checks the checked-out tree).
- Third-party web claims (F14) are recorded as reported by their sources, not independently investigated.
- Classification of edge cases (conditional vs off-topic) involves judgment; every classification here is over-determined by multiple quotes.

Nothing in the verdict depends on any unverified item.

References