Project Armour — Executive Summary

One-page brief · v0.3 · May 2026

What it is

A **deterministic verification layer** for AI-generated financial analysis. Given a source document (earnings release, transcript, filing) and any LLM's written output, Armour returns a per-claim ledger: which source sentence supports each assertion, and how confident the match is. **No generative AI in the audit loop** — the verifier cannot hallucinate by construction.

Why it matters

Frontier LLMs produce confident, plausible, factually wrong claims at a non-zero rate that no prompt or model upgrade eliminates. The two industry-standard fixes fail in predictable ways:

Fix	Failure mode
"Use a better model"	Reasoning models *editorialise more*, not less
"Use an LLM as a judge"	Judge shares the same priors, agrees with the wrong answer
"Use RAG"	Grounds the input, does not constrain the output

A compliance officer cannot defend a sign-off on "the analyst skim-read it." Armour produces the **claim-by-claim audit trail** that defence requires.

Headline result (v0.3 evaluation)

203 (model, ticker, task) pairs · 7,372 audited claims · 30 sector-diversified US large-cap earnings releases · 4 frontier models · zero LLMs in the audit loop.

Model	Pairs	Claims	Accuracy	Fail	Fabrication
GPT-5.5 Pro	29	742	**74.3 %**	8.2 %	5.1 %
OpenAI o3	55	2,038	**71.5 %**	11.0 %	5.3 %
Claude Opus 4.7	59	2,220	60.0 %	15.5 %	**10.6 %**
Claude Sonnet 4.6	60	2,372	59.6 %	20.5 %	8.5 %

> *Fabrication = claim with no traceable anchor in the source. The marketed "premium reasoning" model (Claude Opus 4.7) posts the highest fabrication rate in the panel — every claim is a defensible audit failure waiting to happen.*

Three findings a buyer should care about

1. **Every frontier model fabricates.** Even the cleanest (GPT-5.5 Pro) invents ~5% of claims. The weakest invents ~11%. *That is the market.*
2. **Summaries fabricate more than analyses.** Counter-intuitive, audit-relevant: the one-page summary that gets emailed to the PM is the riskier artifact.
3. **Different models fail differently.** Opus *invents*; Sonnet *misstates*; o3 *drifts*. The correct model choice depends on which failure mode your reviewer can catch by hand. No LLM-as-judge can produce this signal.

Product surface

- **Deterministic auditor** — `src/auditor.py`. Regex extraction + arithmetic recompute + period alignment + scope fingerprint + token/embedding anchoring. Identical input → identical verdict, every time.
- **Unified CLI** — `armour audit | generate | batch | serve | report`. Full v2 evaluation (240 jobs queued, 203 completed) ran end-to-end in **<35 minutes** on a laptop.
- **HTML & JSON outputs** — per-claim ledger with verdict, source anchor, and confidence. Compliance-ready.
- **Three operating profiles** — `compliance` (strictest), `balanced`, `research`. Same engine, different verdict thresholds. Surfaced explicitly so the trade-off is auditable.

Market

Underserved tiers where an LLM-judge is unacceptable:

- **Boutique asset managers** — need defensible LLM workflows without an in-house ML team
- **Independent research / sell-side desks** — need an audit trail attached to every published note
- **Mid-tier bank compliance** — need to sign off on AI-assisted research without owning the model risk

What the auditor cannot do

It verifies **consistency with a supplied source**, not truth in the world. It flags inferences for human review rather than judging them. It is sensitive to source-extraction quality. These limits are scoped, enumerable, and exactly what a compliance buyer expects from a deterministic system.

Full whitepaper: WHITEPAPER.md · Source data: batch_results_v2/ · Reproduce: see Appendix B.