# Nick Lamb, PhD

**Applied AI Engineer | Technical Founder | Open-source eval, MCP and agent infrastructure for healthcare and scientific research**

Oxford, UK · +44 7941 892 694 · nick@pharmatools.ai · github.com/nickjlamb · linkedin.com/in/medcopywriter

## PROFESSIONAL SUMMARY

Technical founder building and shipping AI products from 0→1, with a focus on open-source evaluation infrastructure, MCP tooling, privacy-preserving clinical AI and scientifically grounded agent systems. I design Claude-native tool-use architectures, custom MCP servers and deterministic evals that gate releases, using Claude Code as my primary development environment. A PhD in Biochemistry and 20 years translating complex science allow me to work effectively with clinicians, researchers and technical teams, turning ambiguous needs and model failures into practical, testable systems.

## CORE SKILLS

**Core:** Rapid AI prototyping · Agentic AI · Claude Code · MCP (Model Context Protocol) · Agent Skills · Anthropic / OpenAI APIs · TypeScript · Python · AI evaluation & benchmarking · Product development (0→1)

**AI Engineering:** LLM apps, agents & assistants – tool-use, forced tool schemas, structured extraction · MCP server development and publishing (npm, official MCP Registry, Anthropic MCP Directory) · packaging for adoption (PyPI, Docker, GitHub Actions) · prompt & context engineering · controlled generation, structured outputs, prompt caching · RAG – LangChain (LCEL), FAISS / Pinecone, embeddings · multi-modal (vision), long-context handling

**Evaluation & Grounding:** deterministic gold-anchored eval design · citation-faithfulness & groundedness checks · LLM-as-judge evaluation and deterministic alternatives · hand-labelled gold sets and labelling guides · regression gates against a baseline in CI · benchmark and adapter design · clinical validation study design (clinician + patient panels)

**Engineering & Delivery:** TypeScript / Node.js, React, Python, Swift (SwiftUI) · Firebase, Azure OpenAI, Cloud Functions · GitHub Actions CI, Docker, Railway · Microsoft Word add-in (Office.js), Streamlit, iOS · testing, code review, modular design, open-source release engineering and documentation

## SELECTED OPEN-SOURCE & PUBLIC-GOOD INFRASTRUCTURE

**OpenGATE** – Open Grounded AI Testing & Evaluation · github.com/nickjlamb/opengate · Node.js + Python · MIT · Zenodo DOI

An open evaluation framework for AI systems that must justify every answer from source material – RAG pipelines, document QA, scientific and clinical assistants. The check is **deterministic and gold-anchored: no LLM judge, no grader model.** Required facts must be present, every number must trace back to the source, and the system must abstain rather than fabricate when the context can't answer. Scores are reproducible, free, and fast enough to gate every commit.

- Designed the methodology: **40+ hand-labelled gold cases** with a published schema and labelling guide, 7 scorers (one per metric family), versioned JSON/HTML scorecards, and a regression gate that diffs each run against a per-adapter baseline and fails the build on a drop.
- Used it to drive model and system decisions using measured performance: **halved passage hallucination in a production system (5.8% → 2.4%)** via a model change, eliminated a parse-failure mode affecting ~50% of multi-claim verdicts, and holds claim extraction at **~0.95 F1** with near-full recall.
- Shipped as ecosystem infrastructure across **five surfaces from one core**: CLI/framework (npm), a drop-in GitHub Action, a Python package with a pytest gate and a DeepEval metric, an MCP server so agents can verify their own answers inline, and a CPU-only Docker image.
- Built an adapter architecture supporting retrieval, citation-verification, redaction and simplification systems, allowing teams to reuse the same scoring framework while keeping their own gold sets.

**PubCrawl MCP** – github.com/nickjlamb/pubcrawl · TypeScript · MIT · npm + official MCP Registry

Public infrastructure giving any LLM client reliable, live access to the biomedical evidence base – **PubMed, ClinicalTrials.gov and FDA/UK drug labelling** – through **13 MCP tools**, because AI systems reasoning about medicine should retrieve the underlying evidence rather than rely on model memory: search and abstracts, full text via PMC, related-article discovery, trial detail with eligibility and outcomes, SmPC/USPI retrieval and comparison, and formatted citations.

- Handles the unglamorous parts that make public data usable by agents: rate-limited clients respecting each API's etiquette, an LRU cache with per-resource TTLs, resilient PubMed and JATS XML parsing, and typed returns designed to be read by a model.
- Published to npm and listed in the official MCP Registry; deployed with GitHub Actions CI, powering retrieval in a shipped product, and evaluated as a system under test in OpenGATE – open source → production → measured.

**StudyDiff** – studydiff.pharmatools.ai · github.com/nickjlamb/studydiff · Claude-native · MIT

Built for the *Built with Claude: Life Sciences* hackathon. Two well-run papers often reach opposite conclusions, and the reason is usually a methodological difference buried in the methods sections. StudyDiff does that digging: it extracts each study's design, ranks the divergences that most plausibly explain the disagreement, and grounds every statement in the source.

- **Claude-native agent design:** each paper becomes a validated study card through a **forced Claude tool schema**, so every field returns structured and carries a verbatim supporting quote – no free-text parsing, no "mostly-JSON" failures. Absent fields default to
- Claude performs structured extraction; a deterministic verification pass then checks every supporting quote and numerical claim before comparison. Any value that fails is downgraded. Comparison and driver ranking involve no model judgment, so the same verified evidence always yields the same answer.
- Ships an MCP server (npm) alongside the web app, plus a PubMed/PMC client with a full-text-to-abstract fallback that tags how deep it read. Offline demos of real scientific contradictions run with no API key.

**Redacta** – github.com/nickjlamb/redacta · Zenodo DOI · MIT-0 · Anthropic MCP Directory

Open-source privacy infrastructure for clinical text. Clinicians and researchers routinely paste real patient data into AI tools; Redacta replaces identifiers with labelled tokens before the text reaches any downstream model, leaving the clinical meaning intact and returning a report of every replacement. Re-identification is first-class, so redact → process → restore is a complete round trip and identifiers never leave the user's machine.

- **One detection engine, seven installable surfaces**, so it meets people where they already work: an Agent Skill distributed through ClawHub, an MCP server (npm; in the Anthropic MCP Directory and the MCP Registry), TypeScript and Python libraries, a CLI, and FigJam/Miro plugins – plus a native iOS build (app, Share Extension, widget) running the same engine on-device via JavaScriptCore.

- **Benchmarked the deterministic engine against Microsoft Presidio** on 300 synthetic UK clinical notes (1,713 labelled identifiers): **100% strict recall with zero false positives** within its pattern scope, against 77.3% and 1,643 false positives for Presidio. Reproducible from the repo; free-text names are out of scope by design.
- Deliberate hybrid architecture: deterministic patterns for fixed-format identifiers (NHS numbers with Modulus-11 validation, NI numbers, postcodes, MRNs), and – in the Agent Skill – LLM reasoning for what patterns can't catch (patient names distinguished from treating clinicians, addresses, identifying ages), followed by a self-check pass. Includes a HIPAA Safe Harbor mode.
- Design decisions carry the judgment: a date of birth is removed but an appointment date is kept, because blanket date-stripping destroys the meaning the tool exists to preserve. Positioned as a strong first line of defence, not a guarantee.

## DEPLOYED BENEFICIAL-USE PRODUCTS

**Patiently AI** – getpatiently.ai · clinically validated · eval-backed · award-winning

A generative system using controlled generation and safety constraints to adapt complex medical documents for the audiences who have to act on them. Shipped behind a three-phase validation study (210 outputs, 15 clinicians, a 54-patient survey; medRxiv preprint, under peer review): **87.3% of outputs rated clinically safe, 70% preferred over the baseline.** Demonstrates that beneficial deployment depends as much on the evaluation and safety framework as on the underlying model.

**RefCheckr** – pharmatools.ai/refcheckr · custom eval framework

Verifies claims against their cited references in minutes, combining LLM-assisted verification with structured retrieval and vision-based table/figure analysis; ships as a Microsoft Word add-in (Office.js) with annotated-PDF and report exports. Gated by a two-stage citation-faithfulness eval – deterministic quote location, then LLM-as-judge support grading – that catches unsupported outputs before a user sees them. The eval design became the seed of OpenGATE.

## PROFESSIONAL EXPERIENCE

**Founder – PharmaTools.AI** 2022 – Present

- Ships AI products 0→1, end-to-end – scoping, prototyping, building, deploying and documenting agentic and generative systems across TypeScript, React, Node.js, Python and Swift codebases, using Claude Code daily. Multiple production apps and open-source packages live across web, mobile and the MCP ecosystem.
- Builds ecosystem-level public goods rather than one-off tools: OpenGATE (evaluation), PubCrawl (biomedical retrieval) and Redacta (de-identification) – released open-source and packaged for the surface each audience actually uses, from npm and PyPI to Docker, GitHub Actions, MCP servers and Agent Skills, with the documentation, schemas and contribution paths other teams need to adopt them without me.
- Designs evals that gate releases and drive prompt and model decisions – a deterministic gold-anchored grounding framework, a two-stage citation-faithfulness eval that blocks unsupported outputs, and a three-phase clinical validation run with practising clinicians and patients.
- Acts as a technical partner to clinicians, medical and life-science teams – translating user needs into product requirements, demonstrating prototypes, explaining system capabilities and limitations, and converting model failures into measurable evaluation cases.
- Supports adopters directly – technical writing, reference documentation, worked examples, offline demos, issue triage and code review on open-source repos, and quick-start guides that let teams evaluate the work independently – and champions engineering practice in AI work (tests, CI, reproducible builds, versioned scorecards, baseline diffs).

**Medical Writer & Scientific Communications Consultant** ~20 years to 2022

- Two decades translating complex clinical and scientific evidence for clinicians, patients, regulators and commercial teams – the same problem I now solve with models, and the source of the domain judgment behind Patiently AI, RefCheckr and Redacta.

## OTHER OPEN-SOURCE WORK

**LitRAG** – Python · MIT. Grounded RAG with a built-in citation-faithfulness eval; LangChain (LCEL), sentence-transformers, FAISS, and a two-stage groundedness check informing prompt and model selection before shipping.

**The RSI Loop** – Python · MIT. A self-improving computer-vision system that tunes its own detection logic against a benchmark suite, supervised by an Auditor that prevents reward hacking. Ships with tests and a narrated demo.

**HushMap** – SwiftUI, Firebase, OpenAI. iOS app helping neurodivergent users find and predict quiet, low-stimulation places, combining community reports with an AI noise-prediction engine; on the App Store. **PosterLens** – Swift. Multi-modal querying of scientific posters; a reference implementation in the Perplexity API Cookbook. **SideEffectViz** – Python · MIT. ML visualisation of adverse-event datasets (K-means, PCA); Streamlit app with CI. All at github.com/nickjlamb.

## EDUCATION & PUBLICATIONS

PhD – Biochemistry, Imperial College London

Lamb N. Translation, not Interpretation: Rethinking Language Model Design for Healthcare. SN Comprehensive Clinical Medicine. 2026;8:71 · doi.org/10.1007/s42399-026-02316-9

Lamb N. Validation of an AI-powered mobile application for personalizing medical note explanations. medRxiv (2025) – preprint, under peer review · doi.org/10.1101/2025.09.17.25335707

## AWARDS & RECOGNITION

Winner – Communiqué Awards 2025, Progress in Healthcare & Scientific Communication · Winner – PMEA Awards 2025, Innovation & Patient Education (both Patiently AI)

Redacta in the Anthropic MCP Directory · Redacta and PubCrawl in the official MCP Registry · PosterLens in the Perplexity API Cookbook