Richard (Kai) Hallett

Applied AI Engineer

Production LLM systems | evaluation | safety

kai@oceanheart.ai | github.com/rickhallett | linkedin.com/in/richardhallett86 | oceanheart.ai

BASED / TRAVEL
United Kingdom
Remote, hybrid, willing to relocate

ENGINEERING
6.5 cumulative years
Professional since April 2019

CLINICAL
15 years CBT
NHS and private practice

Applied AI engineer who makes LLM systems safe to run commercially, with more than six years of professional software engineering since April 2019 across salaried roles, contracts, client delivery, and independent product development. Most recently, I architected and delivered a fail-closed conversational engine for Loans by MAL, a regulated UK lender. An untrusted LLM proposes each turn, a deterministic policy validator overrides unsafe output before it reaches a customer, and every turn lands in an audit trace. I build the full stack around the model: eval harnesses, guardrails, provider-agnostic adapters, cost ledgers, and the observability that shows what an agent system is actually doing.

I own delivery end to end - backend, APIs, front end, infrastructure, live support - and I run AI coding agents behind structured review gates as part of that delivery. Everything I build is provider-agnostic by default: no system I ship is locked to a single LLM vendor or its pricing.

Before engineering, I worked as a cognitive behavioural therapist across NHS and private practice. The core clinical skill - noticing when something is producing confident output that is completely wrong - is the same skill that makes evals, adversarial testing, and fail-safe design work.

AI ENGINEERING

Fail-closed conversational engine - Loans by MAL (regulated UK lender, contract). The reference build: an untrusted LLM proposes each turn; a deterministic policy validator enforces hard compliance rules and overrides unsafe output before it reaches a customer - no ungrounded answers, no account-specific promises, credentials and vulnerability cases routed to a human. Hardened against a persona-driven journey simulator (adversarial, confused, oversharing, vulnerable personas) before any happy-path testing, benchmarked across providers on accuracy, cost, and safety. Every turn audited. TypeScript, Zod, provider-agnostic model adapter.

Sortie: ensemble AI code review. Three LLMs (Claude, Codex, Gemini) review a diff in parallel, a fourth synthesises, and merge is gated on convergent severity - runnable as a CI or pre-merge hook. It surfaced a committed API key in one run. TypeScript, Bun.

The Pit: agent evaluation and cost visibility. A platform for evaluating multi-agent systems: structured contests with observable traces, rubric-based scoring, failure tagging, and a cost ledger with per-run microdollar accounting. Next.js, TypeScript, Postgres, Stripe.

Halo: agent infrastructure. A Node.js gateway over the Claude Agent SDK with containerised agent runners and a Python tool layer (27 modules) for memory, work tracking, briefing synthesis, and integrations - each module callable from an agent and testable in isolation. Event sourcing via NATS JetStream; Kubernetes with Grafana/Prometheus observability.

Also: **Arcana**, a multi-agent data pipeline (NATS JetStream, LangGraph, LLM-as-judge; Python, FastAPI, Kubernetes), and **Jeany**, a content-to-intelligence pipeline (Python services, Next.js dashboard, Go CLI; Kubernetes via Helm, Terraform, ArgoCD).

All repos public on GitHub except confidential client work.

EXPERIENCE

2026 (May-June) · Full Stack Software Developer · Loans by MAL (contract)

One-month contract building the conversational engine above, at senior scope from day two.

- Architected end to end: Vue/TypeScript front end, Node.js/TypeScript backend, shared Zod contracts, deterministic validator overriding unsafe model output.
- Defined scope across concurrent epics: turn-planning engine, retrieval, audit trail, compliance rules.
- Presented technical direction to C-suite stakeholders, translating regulatory constraints into engineering decisions.
- Built the AI layer provider-agnostic behind a swappable adapter, benchmarked with a persona-driven journey simulator across accuracy, cost, and safety.

- Infrastructure as code with Terraform on AWS. Spec-first under prototype pressure: typed contracts, per-turn audit traces.

2024-present · Software Engineer · Oceanheart.ai (independent)

The AI engineering projects above, plus paid client builds delivered solo end to end (marketing sites, gated member portals with AI chat and Stripe, community platforms - Next.js, Astro, React 19). Custom agentic harnesses for clients; AI content workflows tuned through disciplined evals and iteration.

2023-2024 · Software Engineer · EDITED · retail analytics

React, TypeScript, Python. Data-visualisation features for an AI-driven retail analytics SaaS; API integrations for enterprise clients with the backend team.

2021-2023 · Software Engineer · Brandwatch · social intelligence

Greenfield enterprise data-visualisation platform (Monitor, team of 5). Scalable React components, modernised a legacy Backbone codebase, mentored junior developers.

2020-2021 · Full Stack Engineer · Telesoft Technologies · network security

TypeScript, Angular, Node.js. Secure features for cybersecurity applications.

2019-2020 · Software Developer · School Business Services

Vue.js frontend for school financial-management software.

2004-2026 · Cognitive Behavioural Therapist · NHS / Private Practice

PGDip CBT (Royal Holloway, University of London). Small private caseload maintained.

EDUCATION AND TECHNICAL

EDUCATION

PGDip Cognitive Behavioural Therapy, Royal Holloway, University of London

PGCert Primary Mental Healthcare, University of Central Lancashire (UCLan)

BSc Psychology, UWE Bristol

TECHNICAL

AI engineering: agent and tool-layer architecture (Claude Agent SDK), eval harnesses and rubric-based evaluation, LLM-as-judge, guardrail and fail-closed safety design, provider-agnostic multi-model pipelines (Claude / Codex / Gemini), adversarial persona simulation, cost/token accounting, AI-assisted delivery behind review gates

Languages: TypeScript, Python, Go, Bash

Backend: Node.js, FastAPI, REST APIs, PostgreSQL, NATS

Frontend: React, Next.js, Vue, Tailwind

Cloud & infra: AWS, Terraform, Docker, Kubernetes, GitHub Actions, Grafana/Prometheus