

Emanuele Czofei

AI Engineer · Python Backend

Gipuzkoa, Spain (CET) · EU work authorization · Open to remote (EU / UK / US-overlap) and relocation within the EU
emanuelec06@gmail.com · +34 688 774 858 · emanuelec.dev · github.com/emanuelec · linkedin.com/in/emanuele-czofei

SUMMARY

AI engineer who builds LLM systems together with the evaluation and cost infrastructure that makes them safe to run in production. Shipped a Spanish regulatory RAG assistant gated in CI at Hit@8 0.98 / MRR 0.955 against a 50-question human-labeled golden set — and caught its LLM judge measuring nothing at all before rebuilding it against adversarial negatives. Backend foundation in Python, FastAPI and PostgreSQL; comfortable owning the TypeScript/Next.js surface on top.

EXPERIENCE

Independent AI Engineer

Remote · Gipuzkoa, Spain | 2024 — Present

Design and ship production-grade LLM systems end to end: retrieval architecture, agent orchestration, evaluation harnesses, cost and safety controls, and the deployed interface. Selected work below.

AI & Full-Stack Engineer Intern — Gaddr

Stockholm, Sweden · Remote | Sep — Dec 2025

Shipped authentication pages for an AI-driven web application in Next.js / TypeScript; implemented Zod request/response validation for an external API integration.

Technical Challenge — SlideSpeak

AI presentation startup | 2025

Built an async PowerPoint-to-PDF conversion service under time constraint — FastAPI upload endpoint dispatching Celery workers via Redis, server-side LibreOffice conversion, S3 storage, client-side job polling. Fast-tracked past intermediate interview rounds.

SELECTED PROJECTS

Spanish Occupational-Safety RAG Assistant

github.com/emanuelec/rag_project

Python · FastAPI · LangChain · ChromaDB · VoyageAI voyage-3 + rerank-2.5 · Claude API · Next.js · GitHub Actions

- End-to-end RAG over the Spanish INSHT/NTP safety corpus — hybrid BM25 + dense retrieval, cross-encoder reranking, grounded generation with citations. Validated on a 50-question human-labeled golden set: **Hit@8 0.98, MRR 0.955, Precision@8 0.69**.
- Three-layer evaluation harness — retrieval metrics, an LLM judge scoring faithfulness / relevance / completeness, and meta-evaluation of the judge against human labels — wired into GitHub Actions as a pull-request gate that fails the build if judge TPR or TNR falls below 0.85.
- Found the judge was decorative:** its labeled set contained no negative cases, so it had never rejected an answer and its true-negative rate was undefined rather than merely low. Rebuilt it with reference-aware prompting and nine purpose-written adversarial negatives; revalidated on a 14-item meta-labeled set at faithfulness TPR/TNR 1.00 / 1.00. A small set, and I say so.

Docent — Autonomous OSS Support Agent IN PROGRESS

github.com/emanuelec/Docent-Backend

Support agent for pydantic/pydantic | Python · FastAPI · LangGraph · Celery · Postgres/pgvector · Redis · MCP · Next.js

- Running today:** ~1,500 resolved issues ingested as a hybrid RAG corpus via GitHub GraphQL fetchers and Celery pollers on a five-minute cycle, Alembic-versioned schema, hybrid keyword + vector retrieval with reranking and score merging, and three of five agent phases live (intake → research → draft) — research calling the GitHub MCP server as a client.
- Designed, not yet built:** the LLM-judge quality gate and escalation phase, prompt and semantic caching, per-request cost metering, circuit breaker with degraded mode, prompt-injection defence.

Triage (RepoDoctor) — Automated Code-Diagnosis Agent

github.com/emanuelec/Triage

LangGraph state machine that clones a target repository, auto-detects its stack, installs dependencies and runs the project's own test / build / lint suites to explain what is failing — every claim grounded in real command output. Isolated sandbox and throwaway virtualenv per repo, async subprocess execution with timeouts, Next.js UI streaming progress over server-sent events.

TECHNICAL SKILLS

Backend: Python, FastAPI, PostgreSQL / pgvector, SQLAlchemy, Alembic, Celery, Redis, REST API design, async Python, Docker & Docker Compose, GitHub Actions, Railway, Vercel, AWS S3. Reading and writing Go and Rust (async runtimes, axum, sqlx).

LLM systems: RAG pipelines (hybrid BM25 + dense, cross-encoder reranking), LangChain, LangGraph (agent orchestration, state machines), MCP clients, Claude API, OpenAI API, VoyageAI embeddings & reranking, ChromaDB, structured outputs, prompt engineering.

Evaluation & reliability: golden-set construction, retrieval metrics (Hit@k, MRR, Precision@k), LLM-as-judge design, judge meta-evaluation (TPR/TNR), eval gates in CI, token cost metering and prompt-caching strategy, prompt-injection defence, Langfuse tracing.

Frontend: TypeScript, Next.js, React, Tailwind CSS, streaming UI (server-sent events), internationalization (ES/EN). | **Spoken:** Italian & Romanian (native), Spanish (fluent), English (professional working proficiency).

EDUCATION & WRITING

Self-Directed Software Engineering, 2024 — Present. Structured curriculum in production LLM systems, backend engineering and systems programming. Maintain a technical vault of 226 notes across eight tracks under a standing rule that no claim ships unmeasured — every number and program output quoted was produced by running it. Authored a 13-module hands-on course in LLM evaluation built around a deliberately broken assistant whose failure mechanisms are never named in the text, with a grader that scores a learner's taxonomy by trace-set overlap rather than category names. AWS Certified AI Practitioner (AIF-C01) — in progress.