

Krzysztof Kutniowski

AI Engineer (LLM, agenci AI, RAG, MCP) · Full-stack Python / TypeScript · Automatyzacja procesów

Zabrze, Polska krzysztofkutniowski@gmail.com +48 572 042 069
github.com/GenorTG linkedin.com/in/krzysztofkutniowski
Portfolio: smart-samurai.pl Smart Knife: smart-knife.smart-samurai.pl
Finetune Studio (GitHub Pages): genortg.github.io/finetune-studio

PROFIL

Inżynier AI i full-stack developer z ponad 8 latami komercyjnego doświadczenia w tworzeniu oprogramowania; od 2 lat buduję systemy AI na produkcji. Projektuję i wdrażam rozwiązania oparte na LLM: agentów AI z własną pętlą agentową i tool calling, pipeline'y RAG, integracje przez MCP oraz automatyzacje procesów w n8n — od prototypu po działające wdrożenie na własnej infrastrukturze (Linux, Docker, PostgreSQL/PostGIS, Tailscale). Jako Lead IT Developer i Team Leader IT w Rentiers P.S.A. buduję systemy firmy i wdrażam AI w pracy innych działów. Rozwijam publiczne projekty: Finetune Studio (RAG, fine-tuning LoRA/QLoRA, ewaluacja modeli, 2 725 testów automatycznych) i Smart Knife (25 narzędzi, asystent AI działający w każdym z nich, zbudowany przeze mnie serwer MCP ze 113 narzędziami dla zewnętrznych agentów, publiczna instancja online), a na własnym serwerze utrzymuję skonfigurowany przeze mnie open-source'owy OpenClaw — centrum zarządzania agentami CLI, aplikacjami do kodowania i automatyzacjami. Programuję z agentami AI (Claude Code, OpenAI, OpenCode) i odpowiadam za jakość tego, co trafia do produkcji: architektura, code review i testy.

DOŚWIADCZENIE

Właściciel · AI Systems Engineer, Krzysztof Kutniowski Smart Samurai — 11.2024 — obecnie

Własna firma B2B

- Projektuję i wdrażam lokalne oraz chmurowe systemy AI i automatyzacje agentowe dla procesów biznesowych.
- Łączę LLM inference, RAG, fine-tuning oraz generowanie obrazu, wideo i głosu.

Lead IT Developer · Team Leader IT (wcześniej Web Developer / Specjalista IT), Rentiers P.S.A. — 05.2024 — obecnie

wcześniej ASI Deweloperskie / Deweloperskie P.S.A.

- Zaprojektowałem agenta AI do analizy lokalizacji: własna pętla agentowa, tool calling (OpenAI, Anthropic), 20 narzędzi i scoring z flagami ryzyka.
- Pipeline danych przestrzennych: import ponad 2 mln obiektów OpenStreetMap, PostgreSQL/PostGIS w Dockerze, izochrony dojazdu 5/10/15 min i raporty Excel.

Senior Interactive Courses Developer · Team Lead, Weblink Software — 02.2018 — 05.2024

- Prowadziłem projekt i koordynowałem pracę zespołu ponad 10 osób, uczestnicząc w spotkaniach z klientami i przekładając wymagania na rozwiązania techniczne.
- Przeszkoliłem ok. 90% zespołu developerskiego (w szczycie ponad 30 osób) i wdrażałem nowych pracowników.

WYBRANE PROJEKTY

Finetune Studio

genortg.github.io/finetune-studio

Publiczny projekt na GitHubie i moje główne laboratorium inżynierskie: self-hosted WebUI (Python, FastAPI) do budowy korpusów RAG, fine-tuningu LoRA/QLoRA, lokalnego inference, ewaluacji i benchmarków modeli oraz eksportu GGUF. Prezentacja projektu: genortg.github.io/finetune-studio; kod: github.com/GenorTG/finetune-studio. Sama aplikacja nie ma publicznego live — uruchamia się ją lokalnie.

Python, FastAPI, Jinja2, LoRA, QLoRA, RAG, SentenceTransformers, Tesseract OCR, MCP, GGUF, CUDA

- Pipeline RAG do dokumentów: import wielu formatów z OCR (Tesseract), konfigurowalny chunking, embeddingi SentenceTransformers i postęp na żywo przez SSE.
- Eksport korpusu jako samoinstalujący się pakiet z wyszukiwaniem przez MCP i HTTP API — także w pełni offline, z dołączonymi modelami.
- Fine-tuning LoRA/QLoRA z podglądem loss, gradientu i learning rate na żywo, łączeniem adapterów i eksportem GGUF.
- 981 commitów w 7 tygodni, CI w GitHub Actions i zestaw 2725 testów automatycznych.

Smart Knife

smart-knife.smart-samurai.pl

Smart Knife: mój serwis online z zestawem narzędzi i asystentem AI, domyślnie po polsku, działający bez instalacji pod adresem smart-knife.smart-samurai.pl; wydania przez systemd z auto-aktualizacją i rollbackiem.

Next.js, TypeScript, React, Tailwind CSS, MCP, Python, Playwright, Docker

- Asystent AI działa w każdym narzędziu: kilka czatów naraz, trwałe zadania, analiza obrazu, tryby zaufania i potwierdzanie destrukcyjnych akcji.
- 25 narzędzi, m.in. CV studio (wersje pod ofertę, ATS, tracker aplikacji), Website studio (audyty, PR-y GitHub, Vercel), biuro, workflowy, 13 aplikacji Craft, SQL, OCR PDF.
- Zbudowałem serwer MCP Smart Knife: 113 narzędzi, przez które zewnętrzni agenci (Claude Code, Cursor, OpenClaw) pracują na danych użytkownika; osobny adres URL konta i klucze tylko do odczytu.
- Sandbox Pythona, ochrona SSRF, szyfrowane pliki kont; 105 plików testów jednostkowych, 73 suity e2e (62 Playwright), sonda bezpieczeństwa.

OpenClaw

Moje własne wdrożenie OpenClaw: to otwartoźródłowy (MIT) asystent agentowy OpenClaw Foundation, nie mój kod — ręcznie skonfigurowałem go pod cały swój system i używam codziennie w pracy. Prywatna warstwa na własnym, zawsze włączonym serwerze spina agentów CLI i aplikacje do kodowania (Claude Code, OpenCode, Cursor), modele lokalne i chmurowe oraz automatyzacje w jedną sieć opartą na Tailscale. Wspólna konfiguracja osobista — persony, prompty, skille i pluginy — działa w każdym projekcie, a agenci pracują w pętli: zbierają dane, monitorują, wdrażają, testują i naprawiają, po czym raportują wynik.

AI agents / agentic workflows, Claude Code, OpenCode, Cursor, n8n, LLM inference, llama.cpp, Ollama, Linux, Docker, Tailscale, self-hosting

- Jedno miejsce zarządzania agentami: agenci CLI i aplikacje do kodowania (Claude Code, OpenCode, Cursor) dzielą konfigurację, prompty, skille i pluginy.
- Zawsze włączony własny serwer i kilka maszyn połączonych Tailscale.
- Zadania trafiają do modeli lokalnych (llama.cpp, Ollama, LM Studio) lub chmurowych (OpenRouter, OpenAI, Anthropic).
- Automatyzacje uruchamiane triggerami, harmonogramami i webhookami (m.in. n8n): scrape, monitoring, deploy, testy i naprawa bez ręcznego odpalania każdego kroku.
- Agenci sięgają do moich usług, danych i urządzeń sieciowych wewnątrz prywatnej sieci Tailscale, bez wysyłania ich na zewnątrz.

UMIEJĘTNOŚCI

Web i aplikacje: TypeScript, JavaScript, React, Next.js, FastAPI, REST API, JSON, Server-Sent Events, OAuth (Google), D3 (d3-geo), HTML/CSS, Tailwind CSS, Playwright, pytest, mobile apps, Git, SEO

AI / ML: Python, LLM inference, AI agents / agentic workflows, tool calling, MCP (Model Context Protocol), OpenClaw (agent management hub), RAG, embeddings, OpenAI API, Anthropic API (Claude), LoRA / QLoRA fine-tuning, LLM evaluation & benchmarks, n8n, llama.cpp, Ollama, LM Studio, GGUF, ComfyUI, TTS, Claude Code, Cursor, OpenCode

Infrastruktura i sprzęt: Linux, Docker / Docker Compose, PostgreSQL / PostGIS (Docker), SQLite, self-hosting, Tailscale, reverse proxy, DNS, tunneling, Nextcloud, Home Assistant, CI/CD (GitHub Actions), PM2, cloud deployment, custom PC/server builds, benchmarking

Inne: system architecture, automation, location analytics, report generation, mentoring, leadership, client communication

EDUKACJA

Liceum Ogólnokształcące nr 4 w Tychach

Wykształcenie średnie ogólnokształcące. Kurs Udemy: Complete Blender Creator — Learn 3D Modelling for Beginners (11.2021). Stale się dokształcam: Python, backend, LLM i inference, trenowanie modeli (Boot.dev).

JĘZYKI

Polski — ojczysty Angielski — biegły, poziom zbliżony do ojczystego (C2)

SPOSÓB PRACY

Programuję z agentami AI (na co dzień Claude Code, OpenAI i OpenCode, wcześniej Cursor) i nie ukrywam tego — to mój mnożnik produktywności. Odpowiadam za to, co agent dostarcza: rozumiem kod i architekturę, oceniam kompromisy, precyzyjnie deleguję zadania, robię code review i pilnuję jakości testami oraz CI, żeby nie produkować bylejakiego kodu.

Wyrażam zgodę na przetwarzanie moich danych osobowych zawartych w niniejszym dokumencie w celu realizacji procesu rekrutacji, zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. (RODO).