

50M+

monthly users on marketplaces led

65%+

support queries resolved by AI agent

~68%

operational cost cut on brand platform

55→80%

first-pass acceptance of generated code

~30%

LLM API cost cut via context engineering

SUMMARY

Agentic AI engineer and technical founder with deep experience in software engineering — distributed systems, high-scale platforms, and 0-to-1 product delivery — now building production agentic systems where multi-agent orchestration, RAG, and LLM evaluation replace manual work. Practises eval-driven and spec-driven development: every system is instrumented for faithfulness, groundedness, latency, and cost, with automated evaluation gating releases. Currently building Vauris.ai, which measures how brands appear in AI-generated answers through agent orchestration and automated LLM evaluation. Eight years of technical leadership across teams of up to eight engineers, hands-on throughout.

TECHNICAL SKILLS

AGENTIC & LLM	LangGraph, LangChain, CrewAI, MCP server design and implementation, multi-agent orchestration, agentic design patterns, deep-research loop agents, RAG pipelines, pgvector, FAISS, Neo4j, knowledge graphs, GraphRAG, hybrid search, metadata filtering, semantic chunking, context engineering, multimodal document reasoning (VLMs), agent skills, human-in-the-loop design
EVALUATION & OBSERVABILITY	Langfuse, DeepEval, RAGAS, LLM-as-judge, faithfulness and groundedness scoring, claim decomposition, NLI verification, cross-LLM consistency, golden datasets, regression evals in CI, distributed tracing, token/cost/latency monitoring
SECURITY & GOVERNANCE	Prompt-injection defence (direct and indirect), system-prompt leakage prevention, instruction/data separation, least-privilege tool authorisation, output filtering, red-team eval suites, multi-tenant data isolation, DPIA, SCCs, IdP integration, SIEM
PRACTICE & VELOCITY	Spec-driven development, reusable agent skills, drift control for generated code, deterministic vs. agentic workflow design, event-driven orchestration, n8n, Make, Lovable rapid prototyping
CLOUD & INFRASTRUCTURE	Azure, AWS (Bedrock, Lambda, ECS, RDS, S3, DynamoDB), Docker, Kubernetes, Terraform, Serverless, GitHub Actions CI/CD
LANGUAGES & FRAMEWORKS	Python (FastAPI, async), TypeScript, Node.js, Next.js, Go, PHP
ARCHITECTURE	Microservices, event-driven systems, distributed systems, high-scale platform design, multi-tenancy
LEADERSHIP	0-to-1 product, technical strategy, engineering roadmaps, cross-functional team leadership (8+ engineers), mentoring

PROFESSIONAL EXPERIENCE

Founder & Technical Lead — Vauris.ai

07/2025 – Present

- AI Brand Visibility Platform · Berlin, Germany
- Building an AI brand-visibility platform that measures how brands are represented in AI-generated answers across ChatGPT, Gemini, and Perplexity — defining an emerging product category in AI search visibility.
 - Architected the production platform: three parallel retriever agents (ChatGPT, Gemini, Perplexity) feeding a synthesis layer that produces unified visibility and sentiment scores.
 - Built a two-signal evaluation layer combining cross-LLM consistency scoring with NLI verification against brand source documents to detect hallucination and measure positioning accuracy, plus a RAG faithfulness evaluator using claim decomposition and LLM-as-judge entailment.
 - Shipped an autonomous SDLC agent as a LangGraph state machine spanning planning, architecture, coding, testing, review, and deployment — human-approval gates at high-risk transitions plus automated quality gates (generated-test pass rates, review-agent acceptance criteria, regression checks) block promotion until thresholds are met.
 - Designed and built a deep-research loop agent (LangGraph + FastAPI) running multi-step reasoning over repeated research cycles: a planner decomposes the question, retrieval and synthesis agents gather evidence, and a critique agent scores confidence and grounding — below threshold the loop re-queries and re-synthesises, escalating to a human reviewer after three failed cycles. Grounded, citation-backed answers; cut manual research effort ~55%.
 - Moved the production index from FAISS to pgvector for durable storage, transactional consistency with application data, and metadata-filtered retrieval — kept FAISS for offline evaluation sweeps; HNSW parameters tuned against a recall-versus-latency budget.
 - Hardened the agent stack against direct and indirect prompt injection: retrieved content is delimited and handled as data rather than instructions, tool calls run under least-privilege allowlists with human approval on high-risk actions, and output filters block system-prompt and credential leakage — a red-team suite of ~60 injection and extraction probes runs in CI and gates releases.

- Adopted spec-driven development with reusable agent skills as the source of truth for code generation — sharply reduced model drift across long-running agent tasks, lifting first-pass acceptance of generated code from ~55% to ~80% and cutting review cycles per feature ~40%.
- Instrumented the agent stack with Langfuse for run-level tracing and cost attribution, and DeepEval for automated agentic evaluation — tool-selection accuracy, task completion, and hallucination checks run as a regression suite in CI against a curated golden set, blocking deploys on score regressions.
- Cut LLM API cost ~30% through context engineering — per-turn context budgets, conversation summarisation, retrieval-result deduplication, response caching, and prompt compression — with no measurable drop in faithfulness scores; per-engine latency and cost traced end to end for auditable runs.
- Cut concept-to-client-demo from ~2 weeks to ~2 days by prototyping in Lovable against FastAPI stubs, letting clients validate scope before engineering spend.

Python FastAPI LangGraph CrewAI pgvector FAISS Langfuse DeepEval n8n Lovable TypeScript Next.js AWS CI/CD

Software Development Lead — MetaDesign

09/2023 – 06/2025

Publicis Groupe · Berlin, Germany

- Architected and led 0-to-1 delivery of a multi-tenant, multi-site Brand Management System on Ibexa DXP, owning technical direction across an 8-engineer cross-functional team — improved performance ~25% and cut operational cost ~68%.
- Built an internal multi-tenant RAG platform: one engine with fully isolated per-tenant knowledge bases (separate FAISS collections so corpora never mix) and grounded, citation-backed generation.
- Built and operated an internal MCP server exposing shared AI capabilities — per-tenant retrieval, document ingestion, log and ticket lookup — as versioned tools with scoped per-tool authorisation, so every new agent consumed one integration layer instead of rebuilding it — adopted across four internal agents, cutting new-agent integration from ~2 weeks to ~2 days.
- AI Buddy (onboarding) — knowledge base over all project documentation for new joiners and team-switchers, with multimodal document reasoning over diagrams, screenshots, and scanned PDFs so visual-only content stayed answerable; cut ramp-up from roughly a month to a couple of days.
- Validated AI Buddy with RAGAS before rollout — faithfulness, answer relevancy, and context precision/recall scored against a curated golden set, with releases gated on a faithfulness floor; context budgeting and chunk deduplication cut tokens per answer ~40%.
- Added prompt-injection and system-prompt-leakage defences across the tenant-facing agents — instruction/data separation on retrieved chunks, input screening on user-supplied text, and output scanning to block system-prompt disclosure and cross-tenant leakage, with per-tool MCP authorisation ensuring a compromised prompt could not widen an agent's access.
- Customer Support Agent — intent classification, RAG retrieval, and escalation routing to troubleshoot and resolve customer issues; handled 65%+ of incoming queries and cut resolution time from 5–6 hours to 2–3 minutes.
- Built an AI Ops agent that ingests structured log streams, detects anomalies, and runs multi-step root-cause analysis to produce reports with remediation steps — reduced production-incident resolution from hours or days to ~30 minutes.
- Built a cybersecurity detection agent correlating security events with rule-based pre-filters and LLM reasoning, adding severity scoring, MITRE ATT&CK tagging, and automated escalation — cut alert noise ~40%.
- Shipped Brandcheck AI, a customer-facing multimodal LLM tool that reasons over brand assets and visual material alongside guideline text to guide brand decisions; co-developed the operational roadmap with the Director, Product, and Delivery teams, contributing to a 20% YoY revenue increase.
- Designed and operated the AI workloads on Microsoft Azure — deployment, scaling, networking, and identity integration — to meet enterprise security and data-residency requirements across tenants.
- Stayed hands-on throughout — architecture, code review, and unblocking the team on hard technical calls.

Python FastAPI TypeScript Node.js MCP Azure Ibexa DXP FAISS RAGAS AWS Terraform MySQL Microservices

Tech Lead — Flaconi GmbH

11/2021 – 08/2023

High-traffic e-commerce · Berlin, Germany

- Tech Lead for core commerce on a high-traffic platform serving millions of users; owned delivery end-to-end — requirements gathering, architecture, cross-team communication, hands-on development, and code review.
- Designed event-driven microservices (Node/TypeScript, Python, DynamoDB, Redis, Kubernetes); built self-service returns, order cancellation, and QR-code returns as new services — order cancellation cut customer-service load 57%, with ~80% QR adoption.
- Shipped a gift-card system (+20% revenue) and back-in-stock and price-drop notifications (+10% conversion); trained TensorFlow churn-prediction models powering proactive retention (+20%) — production ML feeding live business workflows.
- Built a graph-based product recommendation system on Neo4j — modelling products, categories, and customer behaviour as a graph and serving traversal-based recommendations into the storefront.
- Cut infrastructure cost ~20% through a serverless / arm64 Lambda migration and performance tuning; raised flaconi.de Core Web Vitals from ~40% to 80%+; owned alerting, logging, and monitoring (New Relic); mentored engineers through 1:1s and performance reviews, driving 2 internal promotions, and established CI/CD that reduced post-release defects.

Python TypeScript Node.js AWS Serverless Kubernetes DynamoDB Neo4j Redis New Relic

Lead Engineer — iCarAsia Sdn Bhd

06/2018 – 04/2021

Automotive marketplaces · Kuala Lumpur, Malaysia

- Lead Engineer across high-traffic automotive marketplaces (carlist.my, mobil123.com, one2car.com) serving 50M+ users per month, plus autospinn.com and otospirit.com.
- Owned end-to-end delivery — planning, development, testing, and deployment — on a service-oriented architecture; built and trained TensorFlow models for car-price prediction in production.
- Led CRM migration and built dealership and accounts portals; decomposed a legacy monolith into services and modernised the CI/CD pipeline.
- Managed a 6-engineer team with mentoring and growth plans; enforced DRY/KISS/SOLID and code-review standards.

Python PHP Yii2 TypeScript React AWS Docker Kubernetes TensorFlow MySQL MongoDB Redis

EDUCATION

MSc Software Engineering

Universiti Putra Malaysia · 2019 – 2021

BSc (Hons) Information Technology

University of the Punjab, Lahore · 2006 – 2010

AI Software Engineering

Digital Leute School, Cologne (remote)