# Robin Lasserye

Agentic AI Architect · CTO

robin.lasserye@elytre.dev 06 01 64 64 98 elytre.dev
Croix (59170) — Hauts-de-France linkedin.com/in/robin-lasserye github.com/RobinLasserye

I help development teams transition to agentic AI through hybrid AI and human-fallback architectures, backed by an established medtech case. Contract Technical Director for a digital medical device from April 2024 to August 2026, pursuing CE marking.

## Skills Dossier

Agentic AI Architect · CTO — delivery, technologies and measured outcomes

Prepared on 16 September 2026. This dossier details what was delivered in each role and project, the technologies used and measured outcomes. Personal projects are presented alongside client engagements, with each status identified: running in production, prototype, or still experimental.

### AREAS OF EXPERTISE

**Advanced Python, FastAPI and AI services** · Python and FastAPI across the stack: 3D processing microservices, human-review APIs, MCP servers, and data-collection and qualification tools. Systematic pytest coverage and retrofitted TDD on an existing codebase of more than 400,000 lines.

**Agentic AI, multi-agent systems and orchestration** · The core of the work today: designing agent architectures, defining roles, coordination and recovery, and especially governing tool and data access through MCP servers. Hybrid workflows deployed in a medical-device context, where the agent proposes and the human fallback decides.

**LLMs, orchestration and tool calling** · Integration of proprietary models and open models running locally. Tool calling is exactly what an MCP server formalises: declaring tools, controlling what the agent may call, and logging what it called.

**RAG and document retrieval** · Large-scale document ingestion and classification in production in the opportunity collector, the first half of a RAG pipeline. End-to-end enterprise RAG architecture designed and documented: vector store and fallback tier, embeddings, reranker, retrieval evaluation set, mandatory citations and cost by tier. Production deployment of retrieval itself remains to be done, and is described as such.

**AI system evaluation** · Evaluation pipelines built and maintained: the email assistant measures categories, urgency and abstention against a deterministic reference and held-out corpus. Regression measurement is a constant practice, not a tool.

**Engineering delivery, MLOps and LLMOps** · Docker containerisation and continuous-integration pipelines across projects; API services, reusable components, versioning and recovery after interruption. Production services run under systemd with automatic restart.

**Enterprise integration, security and traceability** · Production integrations with third-party systems: shipping, CRM, team chat and business software. Access separation, secrets management, logging and traceability — the WhatsApp connector is an intentionally extreme case: read-only database, no write-capable library loaded, access scope covered by tests.

**Machine learning and data** · Neural-network segmentation on 3D meshes in the orthodontic pipeline — PyTorch and scikit-learn, 17,169 lines, 173 commits over six months. Six-source ingestion and reconciliation pipelines running nightly in production, and more than three years designing company KPIs across finance, production, operations and sales.

**Compliance and responsible AI** · EU AI Act and GDPR review integrated into migration scoping, data anonymisation and access governance. Experience with a product pursuing CE marking, where traceability of an automated decision is essential.

**POC → MVP → production lifecycle** · Completed end to end several times: the 3D viewer moved from prototype to the centre of the production workflow and has been there for nearly four years; the opportunity collector runs continuously; the orthodontic pipeline remains experimental, as stated.

## PROFESSIONAL EXPERIENCE, IN DETAIL

### Aligneurs Français — technical leadership for a digital medical device

Contract Technical Director · April 2024 – August 2026 · Paris, remote

Publisher of a digital medical device for orthodontic treatment with custom aligners, pursuing CE marking. A team of five or more developers; the product runs continuously in production at clinics and dental laboratories.

- Technical leadership and tech lead: architecture, code review, technical-debt decisions, technical hiring and change management.
- Agentic-AI-assisted refactoring of a back end of more than 400,000 lines, with retrofitted TDD and generated, reviewed end-to-end tests.
- Hybrid agentic workflows in production: the agent proposes; the human fallback decides. Human fallback is a domain constraint, not a precautionary principle.
- Billing and support data-ingestion pipelines: six heterogeneous sources reconciled nightly in continuous production.
- Designed dashboards and produced company KPIs for more than three years: finance, production, operations and sales-representative tracking.
- Logistics automation: grouped orders by recipient, generated shipping labels and returned them through the team's chat tool.

**Metrics** · Refactored back end of more than 400,000 lines. • 38% reduction in shipping costs across 1,432 shipments. • Six data sources reconciled every night. • Team of five or more developers.

**Technologies** · Python, Django, PostgreSQL, Redis, pytest, end-to-end tests, Docker, CI/CD, Metabase, HubSpot, FedEx API, Slack, Three.js, WebGL, React, TypeScript

### Elytre (EURL) — agentic AI architecture consulting

Managing director, 100% owner · since May 2025 · Croix (59), SIREN 945 053 023

Independent consulting firm. Scoping migrations to agentic AI, designing agent architectures, and supporting and training development teams.

- Scoped migration to agentic AI, integrating EU AI Act and GDPR reviews into the design rather than adding them at the end.
- Trained and supported development teams on Claude Code and the MCP protocol: governance of tool and data access, guardrails and regression measurement.
- Custom MCP servers: controlled exposure of tools and data sources to an agent, with strict separation between read and write access.
- Designed and documented an enterprise RAG architecture for an AI factory: vector-store choice with a fallback tier, embedding model, reranker, retrieval evaluation set, mandatory citations and cost by tier. A reasoned, quantified design document.

**Technologies** · Python, FastAPI, MCP (Model Context Protocol), Claude Code, local LLMs, multi-agent orchestration, evaluation and guardrails, EU AI Act, GDPR, Docker

### Independent work — 2D and 3D web development engineer

Sole proprietorship · August 2022 – September 2025 · Lille

Three years of independent work in web development and real-time 3D, including the design of the viewer that became central to Aligneurs Français' production workflow.

- Independently designed the orthodontic 3D viewer from 2022.
- Web and real-time 3D applications for several clients.

**Technologies** · React.js, Next.js, TypeScript, Three.js, WebGL, Python, Django

## CLIENT PROJECT DELIVERIES

### Ask business questions in natural language from the team chat tool

Documented client delivery · medtech

A question-and-answer command backed by a language model and the business database, called from the team chat tool. The model receives a natural-language question, generates a database query, and returns the answer in the thread. This translates natural language into a query; it is not document-based retrieval-augmented generation.

**Technologies** · Claude, Slack, LLM integration, business database

### Medtech back-end modernisation

Client engagement · Technical Director mandate, 2024–2026

AI-assisted refactoring of a back end of more than 400,000 lines, with retrofitted TDD and end-to-end tests. Architecture and change management within the team.

**Metrics** · More than 400,000 lines refactored.

**Technologies** · Python, Django, PostgreSQL, Redis, pytest, end-to-end tests

### Billing and support pipelines

Client engagement · medtech · continuous production

Nightly reconciliation of six heterogeneous sources to improve the reliability of management data: ingestion, normalisation, matching and discrepancy detection.

**Metrics** · Six sources reconciled every night.

**Technologies** · Python, data ingestion and reconciliation, scheduling

### Automated shipping

Client engagement · medtech

Grouped orders by recipient, generated shipping labels and returned them through the team's chat tool.

**Metrics** · 38% reduction in shipping costs across 1,432 shipments.

**Technologies** · FedEx API, Slack, integrations and traceability

### Management dashboards and KPIs

Client engagement · company operations · more than three years

Designed dashboards and produced KPIs for finance, production, operations and sales-representative tracking. Bots pulled data from the CRM, which was centralised and presented.

**Technologies** · Metabase, HubSpot, KPIs, dashboards

### PureControl · orthodontic 3D viewer

Client engagement · in use for nearly four years

Visualisation and editing of treatment plans: step-by-step tooth animation, translations and rotations, IPR, attachments and comparisons. Occlusal contacts, gum simulation and CBCT display; streamed models and mesh-preparation and validation scripts. Setups shared between technicians and clinicians.

**Technologies** · React, TypeScript, Three.js, WebGL, Python, Django, glTF/STL

### Orthodontic pipeline

Business-software development · experimental automatic generation

Python microservice: STL scans and prescription as input, tooth segmentation, then preparation and export of a package usable by the business software. Task-launch API, progress tracking and result retrieval; mesh checks and quality reports. Export is technically integrated; clinical validation of automatic generation has not been established.

**Metrics** · 17,169 lines of code, 57 modules • 173 commits from 16 March to 29 August 2026 • Neural-network tooth segmentation on 3D meshes, with boundary refinement, tooth classification and segmentation diagnostics

**Technologies** · Python, FastAPI, PyTorch, scikit-learn, NumPy/SciPy, trimesh, uvicorn, pytest, CLI

## PERSONAL PROJECTS

Projects undertaken independently, using personal time and resources. The figures below were measured from the repositories on 16 September 2026.

### Email assistant — functional inbox triage using a local model, with an evaluation pipeline

Personal project · September 2026 · in progress

A service continuously reads a functional inbox, applies rules written with the client, labels messages, flags urgent items, summarises, prepares drafts, and moves anything uncertain to "to review". It never sends anything. The decision graph is orchestrated with LangGraph and a SQLite checkpoint; model outputs are schema-validated. Three evaluators measure category, urgency and "to review" routing against ground truth and produce a dated report. External tracing is wired up but disabled by default: client messages are not sent to a third party just to look good on a dashboard.

**Metrics** · 16,651 lines of code • 32 commits from 7 to 9 September 2026 • 5 containerised services (dashboard, engine, local inference, test mail server, webmail) • Ground truth of 120 messages across 9 categories

**Technologies** · Python 3.12, LangGraph, LangChain, Ollama, pydantic 2, FastAPI, Docker Compose, IMAP, SQLite, pytest

### Multi-source opportunity collector, exposed as an MCP server

Personal project · in continuous production since July 2026

End-to-end data-ingestion pipeline: nineteen connectors to heterogeneous public sources, normalisation, fingerprint-based deduplication, configurable qualification, storage, and exposure to an agent through an MCP server with six declared tools. Two collections per day, maintained by a system service unit. Qualification was custom-corrected: on a sample of 652 listings, the share of false positives fell from about 69% to about 4%. This is the project's key result because it was measured both before and after.

**Metrics** · 17,137 lines of code, 19 source connectors • 63 commits from 29 July to 7 September 2026 • 15,563 listings and 1,386 distinct companies in the database, 116 collections logged • 6 MCP tools exposed, 29 automated-test files

**Technologies** · TypeScript 5.7, Node 20, Model Context Protocol SDK, better-sqlite3, zod, systemd

### SARTOR — role-separated agentic pipeline

Personal project · August 2026

Multi-agent system where each role is a versioned prompt locked by a fingerprint: designer, critic, subject-matter specialist and writer. Model transport is injectable, and output is constrained by a JSON schema that the SDK validates and asks the model to correct if it does not conform. Execution sandbox, quality gates with thresholds locked in a file, and individually recorded architecture decisions. Exposed through CLI, API and MCP server, all three as thin layers over the same function.

**Metrics** · 11,891 lines in the package, 22,098 in the repository • 45 commits from 5 to 24 August 2026 • 32 recorded architecture decisions, 37 test files • 4 versioned and fingerprint-locked agent roles

**Technologies** · Python 3.12, pydantic 2, FastAPI, FastMCP, claude-agent-sdk, JSON Schema draft-07, PyTorch (CPU), NumPy, shapely, pytest, hypothesis, ruff, mypy

### Read-only WhatsApp connector through an MCP server

Personal project · August–September 2026 · in service

Bridge to a personal account, split deliberately into two processes: one connects and writes to a database; the other reads that database and exposes it to an agent. The database is opened read-only and no write-capable library is loaded on the agent side. Access scope is covered by tests. This is an access-design exercise: make a dangerous capability structurally impossible rather than merely forbidden.

**Metrics** · 3,900 lines of code • 11 commits from 20 August to 15 September 2026 • 7 MCP tools exposed, all read-only

**Technologies** · TypeScript, Model Context Protocol SDK, SQLite, Baileys, vitest

**Local voice dictation, with a measurement bench**
Personal project · since February 2026 · public repository

Transcription through local inference on a graphics card, then formatting by a model that also runs locally: nothing leaves the machine. The project's value for an AI engineering engagement is not dictation itself, but its measurement bench: eleven model-and-precision combinations compared on the same corpus, measuring memory use, load time, real-time factor and word error rate, with and without vocabulary bias. Generative scores are averaged over three runs because a model is not deterministic even at low temperature, and a single run can shift by one point.

**Metrics** · 8,394 lines of code, 23 commits since 9 February 2026 • Measurement corpus of 30 clips, 166 seconds of audio • 11 model-and-precision combinations compared • 3 runs per case for the generative component

**Technologies** · Python, faster-whisper, CTranslate2, CUDA 12, Ollama, SQLite

**Majordome — agent operations cockpit**
Personal tool · used daily since August 2026

Orchestration of agent sessions working in parallel on scoped tasks, messaging one another and reporting results. Each agent receives a scope, explicit prohibitions and an output format; rules learned from real errors are recorded in a doctrine read by all agents, together with the measurement that produced them. This is the daily practice of what an agent platform requires: isolation, logging, recovery, and refusing to issue a verdict when the tool could not measure.

**Metrics** · 84 custom-written Python files, 13,797 lines • Agents can address one another, with scope and logs

**Technologies** · Python, MCP, session orchestration, systemd, tmux, SQLite

**Self-hosted inference stack**
Personal installation · in service

Local language-model stack: inference service, interface, models customised through a definition file, self-hosted search engine, all isolated on an internal network, with watchdogs that restart the service and monitor the graphics card.

**Technologies** · Docker Compose, Ollama, Open WebUI, SearXNG, monitoring scripts

**Java engineering and delivery lab**
Training project · August 2026

API with hexagonal architecture and web interface, multi-stage containerisation, continuous integration pipeline and integration tests run against real services started in containers. The project was built specifically to practise engineering delivery and deployment, not AI.

**Metrics** · 4,923 lines of code, 52 Java files • 17 commits from 13 to 18 August 2026

**Technologies** · Java 21, Spring Boot, Vue 3, multi-stage Docker, JUnit 5, Mockito, Testcontainers, continuous integration

## EDUCATION, LANGUAGES AND ENGAGEMENT TERMS

Engineering degree, Polytech Lille — computer science, microelectronics and automation, 2017–2022. Scientific baccalaureate, highest honours.

French: native. English: B2, TOEIC 850 in 2022.

Contract work through Elytre (EURL), SIREN 945 053 023; sole managing director and 100% owner. Based in Croix (59170), Lille metropolitan area.

Some of the work described is covered by confidentiality commitments: the names and figures included here are those that may be disclosed. Personal repositories can be opened and discussed in a technical interview.