

FULL-STACK ENGINEERING + APPLIED AI RESEARCH / 2026

AI ENGINEER & BUILDER

From model experiments to production systems.

BACKGROUND / SERVICES / CASE STUDY / RESEARCH

00 / AGENTIC AI SERVICES

A full-stack background. An AI engineering practice.

I am an independent AI engineer with a background in full-stack software development. I work across interfaces, backend services, data, and infrastructure, connecting model behavior to the software people actually use. My experience includes working with Mercedes-Benz, Société Générale, and Illuvium.

That background shapes how I approach AI. A promising model still needs a useful product, a sound architecture, and a way to keep working when a tool fails or an input changes. I work with companies, teams, and individuals from the first prototype through production and ongoing operation.

My work spans agent harnesses, workflow automation, LLM architecture engineering, inference systems, and custom fine-tuning. I also study frontier AI research and test ideas through focused experiments, including adapter training, model representations, and distributed inference architectures.

For agent projects, I build the harness around the model: tool access, context management, state, checkpoints, retries, evaluation, and human approval. For model-focused work, I investigate architecture choices, adaptation methods, and inference constraints against the project's data, quality targets, and compute budget.

DESIGN AND BUILD

Define the problem, choose the architecture, and build the application, agent harness, and supporting services. Where the task calls for model adaptation, prepare the data, fine-tuning experiments, and evaluation.

OPERATE

Monitor runs, manage model and tool changes, investigate failures, control cost, maintain quality, and keep human review and release work moving.

IMPROVE OR HAND OVER

Use logs, failures, review results, and cost data to improve the workflow. For handover, document the system and train its owners.

I publish case studies and engineering notes when the material can be shared. They include failures, costs, evaluation methods, and implementation decisions. Confidential client material stays private, and experiments are labelled with their limitations.

01 / SUITABLE PROJECTS

Projects that fit this service.

I work with founders, specialists, and teams building a new AI product or improving an existing system. Suitable projects include agent harnesses, research and publishing workflows, specialist knowledge products, document operations, internal tools, and customer-facing applications.

I also work on the model and infrastructure side: custom fine-tuning, adapter experiments, inference architecture, and evaluation. These projects start with a concrete behavior or constraint to improve, such as domain-specific responses, latency, memory use, or deployment cost.

The service is useful when work crosses several tools, depends on changing sources, contains repeated handoffs, requires human approval, handles sensitive context, or needs an audit trail. These requirements usually need software and operating controls around the model.

I recommend standard automation, search, or a single model call when it can meet the acceptance criteria. The architecture should remain as small as the work allows.

02 / HOW A PROJECT WORKS

How a project runs.

01 / Scope the problem. We define the users, current process, desired output, source of truth, constraints, and measurable acceptance criteria.

02 / Design the system. I map inputs, tools, permissions, model choices, inference constraints, human review points, and failure modes. If fine-tuning is appropriate, we define the training data, a baseline, and held-out evaluation cases.

03 / Build a working version. I implement the smallest complete system using the simplest suitable mix of agents, deterministic software, APIs, data stores, and interfaces.

04 / Test it on representative work. We record completeness, quality, speed, cost, safety, recovery, and usability across real cases.

05 / Operate, improve, or hand over. I can run and monitor the system as a managed operation, work alongside your team, or train an internal owner and transfer it with clear documentation and controls.

A project can end after setup or continue into managed operation. The scope states how results are judged, which decisions stay human, who owns each action, and what the team receives at handover.

Private inputs and project details stay outside public case studies unless publication is explicitly agreed.

03 / DETAILED CLIENT CASE STUDY

219 source notes, one book, and fourteen localized editions.

The client problem. A client brought 219 notes collected while building a house. They needed those fragmented observations to become a coherent, illustrated book for an analytical first-time home builder. Every requested comparison, checklist, example, visual, and personal experience had to survive the edit and remain traceable to its source.

What I built. I designed a content system around stable source IDs, structured content cards, chapter briefs, model-assisted drafting, fact and completeness checks, image tracking, and PDF inspection. When the brief expanded, I added resumable localization, separate native-language copy-editing, language-specific storefront generation, Stripe Checkout, checksums, and immutable edition records.

The production result. The system produced a 227-page Croatian source edition, fourteen localized books and storefronts, and zero unmapped notes at release. New notes can be processed without rebuilding unchanged work, and purchase-time edition metadata remains available for audit while buyers receive the active book for their language.

219 / SOURCE REQUIREMENTS

Stable IDs carried every author note through chapter planning, manuscript, audit, and release.

14 / LOCALIZED EDITIONS

Source hashes, checkpoints, retries, and review gates kept long-running language work controlled.

0 / UNMAPPED NOTES

The system made omissions visible before fluent prose or a polished PDF could hide them.

The controls make omissions visible, limit what a model may change, preserve human editorial review, recover interrupted jobs, record cost, and give each release a verifiable history.

Read A Book Is a Dependency Graph for how source notes, editorial review, localized editions, and releases fit together, or inspect the live product at buildyourho.me/.

04 / OTHER PRODUCTS I BUILT

Two products I built and shipped.

These are self-directed products. Cartoonio coordinates a multi-stage generative media workflow; Symmetry Studio performs image processing locally in the browser.

CARTOONIO / GENERATIVE MEDIA PRODUCT

What it does: turns a written story into reusable characters, planned scenes, generated voices, motion, and a finished animated video. What it demonstrates: orchestration across several generative-media stages while maintaining a consistent project and editable result. Use it at cartoonio.net.

SYMMETRY STUDIO / COMPUTATIONAL DESIGN TOOL

What it does: turns a source image into repeatable designs across all seventeen wallpaper symmetry groups. What it demonstrates: responsive worker-based image processing that keeps source images on the user's device. Open it at operator0x.com/symmetry/.

05 / LEARNINGS AND R&D

Research that feeds the engineering.

I study frontier AI research, focusing on sparse architectures, model adaptation, learned representations, and distributed inference. I turn questions from papers into focused experiments and engineering notes, documenting what I tested and what remains uncertain.

How Small Can We Slice an AI Model? explores sub-layer sharding, granular experts, compact attention, and the possibility of running model computation across everyday devices. It is an architectural argument informed by research, rather than a deployed computing swarm.

Teaching a Model to Be Sarcastic. I rented a GPU and tested small adapters and LoRA placements on a 7B model. The experiment examines what the model learned, what failed to generalize, and how the results changed with more examples. The local draft includes saved outputs and evaluation limits.

Understanding LLMs by seeing language as a shadow of a higher-dimensional state. A companion research note uses shadow-casting sculpture to think through internal representations, changing states, and learned computation. It connects the model-adaptation experiment with questions about what happens between a prompt and a reply. Local draft.

PolyMind explores agent orchestration by sending one brief to two configured AI perspectives and synthesizing their responses. The published $0.09 website critique is an early experiment, with no single-agent baseline or blinded human evaluation.

New case studies and engineering notes are listed at operator0x.com/feed.xml.

06 / DISCUSS A PROJECT

Tell me what you want to build or understand.

Bring an AI product, an agent workflow, a fine-tuning problem, or an architecture question. Explain the current setup, the behavior you want to improve, and any data, privacy, compute, budget, or timing constraints. You do not need to choose a model or framework first.

I will reply with a proposed starting scope or explain why standard automation, search, or a smaller model workflow would be sufficient.

Send the brief to contact@digital-distributed.hr.

> *Start with the problem, the people it serves, and what a useful result looks like.*

FREE TO READ, COPY, AND SHARE WITH CREDIT.