

Arnaldo De Lisio

AI SOLUTION ARCHITECT

delisioarnaldo@gmail.com | [+39 333 238 6826](tel:+393332386826) | arnaldodelisio.it | linkedin.com/in/arnaldodelisio | github.com/arnaldo-delisio

PROFILE

Ten years coordinating film productions taught me how complex systems fail: people, suppliers, deadlines that cannot move. I changed tools, not the job. The systems I build now are software, data and AI inside real businesses, and they have to work the next morning with the people who run them. Every one I have shipped went from nothing to production use and was handed over so it runs without me. Forward-deployed engineering, without a vendor behind me. People do not buy technology. They buy clarity, better decisions and less friction, so I start from how the work happens today and let the technical choices follow. I have done this alone. I want to do it with a team, at a scale one person cannot reach.

ENGAGEMENTS

2024 - 2026 | Three to four running concurrently: three built, put into production use and handed over, one advisory

AI & Automation Lead | De Lisio Costruzioni

2024 - 2026 | My family's construction company, public-works contractor | Delivered and handed off

Bidding taken from 1-2 to 10-20 tenders a month in-house, and its first win in the quality-scored category the company could not enter before, a 2 million EUR contract.

- Before this work the company bid on one or two public tenders a month with outside consultants preparing them. It now competes in tenders awarded on the most economically advantageous offer, scored on quality as well as price, which it could not enter before. The 2 million EUR contract won this year was its first of that type.
- I built the multi-agent tender-analysis platform behind that: Gemini extraction of visually hostile scanned bills of quantities and drawing sets, Supabase Postgres with pgvector, hybrid retrieval fusing keyword and embedding passes with Reciprocal Rank Fusion because tender work turns on exact article codes and norm references, and five specialist agents in parallel (safety, cost, compliance, technical, timeline) reviewing 30-50 documents in a few hours instead of a week, with human oversight on the bid decision.
- Beyond tenders, I built the company's automation for accounting, administration and tender participation, and handed all of it over so it runs without me.

Founding AI Solution Architect | Bureau of Technical Safety

2025 - 2026 | Workplace-safety compliance, regulated document production

3 hours to about 5 minutes per regulated compliance package, about 400 hours of inspector time freed a year.

- I architected and shipped the production platform that turns a site visit into a compliance-document package: voice, text and photographic evidence captured in the field, written to Postgres with a provenance stamp the moment it is heard, and deterministic Python generators producing the legal documents from that database, so the document is a view of the evidence rather than a thing someone typed.
- Professional review, traceability and accountability are preserved; one inspector at about three inspections a week is where the 400 hours come from.
- Claude Haiku extraction with a Claude Sonnet vision fallback, a custom MCP server inside the Next.js deployment as the scoped tool surface, state arbitration in per-table Postgres functions; transitioned to partner-led operation on self-hosted infrastructure.

Fractional CTO | Menaboh.com

2025 - 2026 | Direct-to-consumer luxury fashion, made-to-order production

About 50% less manual coordination across 8 operators, on a platform built from scratch and in production within 6 months.

- I designed and shipped a complete order-to-cash and production ERP from scratch: customer master data, a thirteen-state order lifecycle from intake through design, pricing, approval, payment, production and delivery, every order carrying explicit owners for customer, designer, tailor and coordinator, billing through Stripe, fulfilment through Shippo, document generation and role-based access across six roles, on one Supabase Postgres model.
- Next.js and TypeScript on the front, a separate Node service on the back; the operators gained shared visibility of every order and who owns it.

AI Production Workflow Consultant | Penzing Studios

2025 | Film technology and virtual production

- This was not about building production software; it was about where AI could realistically fit inside professional film production. The question was not what AI can do but what people would actually trust inside their workflow.
- I worked with production and technology stakeholders to find where AI created practical value in creative and operational workflows, and turned that into MVP concepts and integration approaches around existing infrastructure, controls and human decision points. The exciting ideas tended to add more complexity than value; the ones that looked boring removed friction, and those were the ones worth carrying into planning.

EARLIER

Founder | Sunside.club

2023 - 2025 | B2B marketplace for film and television production resources

2,500+ registered users and 120 paying customers, built alone in 6 months, still in daily use after I handed it over.

- Sourcing on a film production is a recurring, high-friction problem: a specific piece of equipment, crew capability, location or service, needed inside a window measured in days, found through a phone book of personal contacts. I had lived on the demand side of it for ten years, so I built the marketplace that replaces the phone book.
- Web, iOS and Android: Next.js, a Node service on Google Cloud Platform, Flutter, Supabase Postgres, Sentry from day one. Handed over to someone I trained, who maintains it without me.

Production Coordinator | International Film & Television Productions

2015 - 2025 | 20+ projects, including Men in Black: International, Sense8, The Equalizer and My Brilliant Friend

- Suppliers, schedules, logistics and cross-department dependencies on productions of 300-500 crew with immovable deadlines, directly coordinating daily teams of 10-20: where the ownership, traceability and contingency habits I apply to AI delivery come from.

OPEN SOURCE

Cockpit (github.com/arnaldo-delisio/cockpit-oss): an operating layer for AI-assisted work that moves memory, operating rules and architectural decisions out of vendor products into versioned files you own. A decision ledger records each choice with its rejected alternatives; retrieval runs on-device with embeddings and keyword search merged by rank fusion; releases run plan/apply with snapshot, verification and atomic rollback.

readwise-mcp-enhanced (github.com/arnaldo-delisio/readwise-mcp-enhanced): the most-starred community MCP server for Readwise, behind only the vendor's own. Treating retrieval as a database-query problem cut typical queries from roughly 25,600 tokens to 1,600, a 94% reduction.

TECHNICAL STACK

Languages: Python, TypeScript, Go, PHP, Ruby.

AI and agents: Claude Agent SDK, OpenAI SDK, LangChain, LangGraph, CrewAI, AutoGen, MCP, multi-provider abstraction layers, retrieval-augmented generation, multi-agent orchestration, evaluation harnesses, fine-tuning.

Data and retrieval: PostgreSQL, Supabase, pgvector, Pinecone, Redis, MongoDB, graph databases, hybrid retrieval, document ingestion.

Backend and web: FastAPI, Node.js, Next.js, React, Vue, Nuxt, Angular, React Native.

Cloud and infrastructure: Google Cloud Platform, AWS, Docker, microservices, CI/CD pipelines, Terraform, self-hosted model runtimes, VPS and bare-metal.

Observability and governance: Grafana, Prometheus, Datadog, Sentry, provenance, scoped permissions, human-in-the-loop controls.

EDUCATION & LANGUAGES

Bachelor's Degree in Business Economics, University of Naples Federico II

Financial Adviser Professional Qualification (OCF)

English: Native | Italian: Native