



CURRICULUM VITAE

Jyri Gromyko Lintunen

Full-Stack Developer · Financial & Industrial Automation

I build software. I love hard problems. I ship things that work.

Open to new projects

9+ years self-employed

Helsinki · remote / hybrid

Results-driven full-stack developer with 9+ years of entrepreneurial coding experience. Specialised in financial and industrial automation, banking integrations, ERP and CRM systems, and LLM-backed production pipelines. Equally at home in UI architecture, Python and PHP backends, database design and server operations. I take a project from spec to production and then keep it alive for years.

✉ jyri.gromyko2002@gmail.com

📍 Helsinki, Finland

in linkedin.com/in/jyri-gromyko

🌐 cv.gromi.fi

9+

years building software

80+

delivered projects

6

years on one ERP system

25+

active code repositories

Profile

Who I am and what I do

I started coding as an entrepreneur in 2017 and have delivered 80+ engagements since: online stores, ERP and CRM systems, banking integrations, computer-vision pipelines and marketing automation. Most of them are still in production.

My focus is where business rules are messy and data is dirty: VAT handling for two company forms, automatic reconciliation of bank statements against receipts, facade-panel geometry on a canvas, lead enrichment across ten different APIs. Systems like these are not flashy — they are what keeps a company standing.

I work with a paper trail: architectural decisions go into ADR files, every module has its own spec, and tests run before anything is declared done. I have built my own AI-agent operating contract (AGENTS.md) that every tool follows, so the work stays repeatable and reviewable from one session to the next.

Work experience

Entrepreneurship, client projects and national service



Entrepreneur & Full-Stack Developer

2017 — Present

[Gromyko Jyri Tmi. / GG Solutions Oy](#) · Helsinki, Finland

My own sole trader business and limited company. I own client projects end to end: scoping, architecture, implementation, production rollout, maintenance and invoicing.

- Designed and built financial automation systems: bookkeeping, VAT and pension-expense handling, financial statements, receivables, banking integrations (Nordea, Holvi).
- Full-stack applications and SaaS products: React / Vue.js front ends, Python, Django, Laravel and Flask back ends, PostgreSQL and MySQL databases.
- API and database architecture: REST APIs, multi-step migrations, bulk SQL processing and synchronisation with POS and ERP systems.
- Server operations and deployment: Docker / Podman Compose, systemd + Nginx, Redis, Celery, shared development infrastructure across several projects.
- LLM-backed production systems: Claude and Gemini integrations, vision analysis, a versioned prompt registry with eval runs, per-call cost tracking.
- Team coordination: developers, designers and marketers; splitting work into parallel work packages and reviewing the results.
- Multi-channel marketing campaigns (Google Ads, Meta, LinkedIn, SEO) and conversion optimisation with measurable results.

Python

Django

Flask

Laravel

React

PostgreSQL

MySQL

Docker

Redis

Celery

REST API

LLM

Military service

6 months

Finnish Defence Forces — Jaeger Brigade · Sodankylä, Northern Lapland

Completed national service in arctic conditions. Endurance, teamwork and disciplined execution under pressure.

Skills & technologies

What I use daily, what I command, and what is familiar

BACKEND

Python

Django

Flask

FastAPI

PHP

Laravel

Node.js

REST API

SQLAlchemy

Celery

WordPress / WooCommerce

FRONTEND

React

Redux Toolkit

TypeScript

Vite

Konva / Canvas

MUI

Tailwind

Vue.js

i18next

jQuery

HTML / CSS

DATABASES

PostgreSQL

MySQL

MariaDB

SQLite

Redis

ChromaDB

SQL-massakäsittely

BANKING & FINANCE

Nordea

Holvi

Firefly III

ALV / VAT

YEL

Tilinpäätös (KPL)

Reskontra

Sähköinen laskutus

Stripe

Visma Nova

INFRASTRUCTURE & DEPLOYMENT

Docker / Compose

Podman

systemd + Nginx

Linux

Git

Google Cloud

AWS

CI/CD

Playwright

Selenium

AI & DATA

Claude API

Gemini API

OpenAI API

openai-agents SDK

LiteLLM

Vision-analyysi

Prompt-versiointi & evalit

NLP

RAG



B2B CRM & lead automation

2026 — Present

Lead acquisition, enrichment, campaign sequences and a sales pipeline on one platform

Architect & sole developer

In production · v8.4

A GDPR-compliant sales-outreach platform that combines an email log, sequence-based campaign automation, lead enrichment and a React CRM. Two back ends live side by side: the Flask production app and a Django rewrite, sharing one PostgreSQL schema and matching each other endpoint for endpoint.

WHAT I BUILT

- A campaign engine: multi-step sequences, message variants with A/B testing (multi-armed bandit), condition-based branches, automatic actions and recipient enrollments.
- Orchestration of 17 lead sources under one scheduler: Finder.fi, PRH/YTJ, Jooble, LinkedIn Jobs and Messages, People Data Labs, Hunter, ZeroBounce, QuickEmail, Apify, Apilayer, Abstract, tech-stack detection and PageSpeed scoring.
- PageSpeed and ad-gap scoring: Google PageSpeed Insights, SSL detection, Google Ads Transparency and the Meta Ads Library are combined into a 0–100 score and a hot / warm / cold priority that refreshes automatically when a company's website changes.
- An enrichment chain where every contact passes through paid and free sources (PDL → Hunter → ZeroBounce → Gemini) until it has a verified, deliverable email address and a reasoned company context.
- A sales pipeline: deals, stages and stage history, tasks and task templates, an activity feed, saved filters and per-user column configurations.
- A GDPR layer: per-project blacklists, a compliance module, unsubscribe handling, bounce events and OAuth rotation for email accounts.
- A proxy layer with sticky sessions: one exit IP is pinned for a block of requests, scoped by a ContextVar, and a dropped session is retried on a fresh session id. Measured: 12 sequential requests gave 11 successes on 11 distinct Finnish IPs, median 1.6 s.
- Container-free production deployment: systemd + Nginx + native PostgreSQL. Development runs on Podman Compose with local overrides.

SCALE

39+ domain models	17 lead sources	17 Django apps	v8.4 released version	78 commits in 6 months
Python	Flask 3.1	Django 6.0	SQLAlchemy 2.0	PostgreSQL 16
Gemini	systemd + Nginx	React 19	Vite 8	Celery
				Redis
				Telegram API



Automated accounting system

2026 — Present

From bank statement to financial statements for two company forms — with no human in the loop

Architect & developer

In daily use

End-to-end bookkeeping automation for a sole trader, a limited company and a personal account. The system pulls bank statements, fetches receipts from email, reconciles them against transactions, computes VAT and pension expenses, publishes the ledger into Firefly III and generates a finished set of financial statements.

WHAT I BUILT

- Bank-statement retrieval and parsing: Nordea (CSV / JSON) and Holvi (XLSX / JSON). The limited company has two Holvi accounts — main and tax reserve — of which only the main one is published to the ledger while the reserve is tracked as a budget.
- Automatic receipt collection through the Gmail API (OAuth2), email-to-PDF conversion, and matching each receipt to the right transaction by amount, date and counterparty.
- A VAT engine: rates and tax classes, reverse-charge VAT, pension-scheme expenses, a counterparty map and rules that live in configuration rather than in code.
- Idempotent publishing into Firefly III: transactions are keyed by external_id, so re-running never creates duplicates. The Finnish chart of accounts is mapped to Firefly categories, budgets are read from a Google Sheets plan, PDF receipts are attached to their transactions.
- A financial-statement generator: a balance sheet and income statement compliant with Finnish accounting law for both company forms, output as XLSX and PDF. Plus a Vero.fi-ready HTML VAT return.
- A planning module: budget versus actuals across three accounts, the source plan coming from Google Sheets and the report published back into a second sheet.
- 20+ runner scripts (.sh and .bat) that drive the whole pipeline with one command; Chrome and fonts are auto-detected on Linux, macOS and Windows.

SCALE

3 ledgers (OY / TMI / personal)	2 banks integrated	20+ pipeline scripts	0 duplicates (idempotent)
Python	Gmail API	Google Sheets API	Nordea
Holvi	Firefly III	SQLite	Gemini
openpyxl	Chrome headless		
Docker			



SKT quoting software — computer vision for job costing

2026 — Present

Building assessment and window-cleaning cost estimation for the Helsinki metro area

Architect & developer · client Suomen Kiipeilytekniikka Oy

Active client project

Replaces the client's Visma Nova based quote calculation with a web application. The pipeline scrapes Oikotie listings, geocodes the buildings, fetches street-level imagery, runs Claude Vision over it, computes man-hours and produces a proposal. It is being extended into per-postal-code prospecting of housing companies and export back to Nova.

WHAT I BUILT

- A 10-phase pipeline: Oikotie scraper (cards, details and photos) → Digitransit geocoding → Google Street View (URL-signed) and Mapillary → Claude Sonnet Vision → man-hour formula v2.0 with a v3.0 refinement → a proposal from a single command.
- LLM extraction of property-manager details from free-form text, import of a list of 584 property-management offices and matching against the existing quote database.
- Cost tracking for every LLM call with its own self-test — the price of vision analysis stays visible before a run is scaled to hundreds of targets.
- A 5-page Streamlit dashboard over the results, an 11-table database and importers for the client's Excel dumps.
- The work is split into six parallel stages (S0–S5) that can run in up to nine simultaneous sessions — each with its own file, ownership and acceptance criteria.

SCALE

10 pipeline phases	264 listings in the pilot area	1,7 GB of imagery to analyse	584 management offices	84 commits
------------------------------	--	--	----------------------------------	----------------------

Python 3.11

Claude Sonnet Vision

Google Street View

Mapillary

Digitransit

SQLite

Streamlit

Docker

Visma Nova



Personal Advisory Council

2026

A self-hosted multi-AI advisory board that evaluates decisions in parallel

Architect & developer

Working prototype

The platform ingests bank statements (XLSX), Obsidian vault notes and GitHub repo READMEs, runs eight specialised agent personas in parallel against the user's question, peer-reviews each response and synthesises a chairman's verdict with an antifragility score (0–100) and a top-three action list.

WHAT I BUILT

- A five-stage workflow: ingest → an architect agent proposes the council → interactive editing → parallel execution → peer review and synthesis.
- Multi-user support with session isolation, URL-based routing, shareable session links and caching that prevents accidental reruns of expensive operations.
- Real-time token and cost tracking, configurable reasoning depth and live progress tracking during a run.
- A Docker Compose stack: FastAPI back end, React front end, a LiteLLM proxy, ChromaDB for vector search and an optional Telegram bot.

SCALE

8 agent personas in parallel	5 workflow stages	0–100 antifragility score
--	-----------------------------	-------------------------------------

Python 3.12

FastAPI

SQLAlchemy 2.0

openai-agents SDK

LiteLLM

ChromaDB

React 19

Tailwind 4

Docker Compose

Telegram



Trading automation & market intelligence

2022 — 2023

Copy-trading bots, strategy statistics and news sentiment analysis across three markets

Architect & developer

Delivered

A two-year effort aimed at measuring what part of successful investors' results is actually repeatable. Public top-trader profiles are read programmatically, their open positions are mirrored onto a demo account, and every trade is logged for analysis. Around that grew news collection, headline sentiment analysis and strategy comparison.

WHAT I BUILT

- A copy-trading bot for three markets: forex (cTrader / IC Markets), stocks and crypto (Binance). Profiles are polled every ten seconds; a new open position is mirrored and closed when the original closes.
- A parser for traders and strategies from public performance profiles (FXBlue): extracting the users, collecting each strategy's history and comparing them statistically — which results survive scrutiny and which are noise.

- A news pipeline with sentiment analysis: collecting foreign and crypto news (Cointelegraph among others), classifying headlines and mapping the sources behind media indices — signal was separated from noise before anything was traded on it.
- Strategy modelling and testing: trade-history export and PnL analysis, pattern search, demo trades written to a log and results compared across periods.
- Deployment to a dedicated server as an endless loop so the watching continues around the clock unattended; every trade is stored with its date, direction, instrument and size.

SCALE

3 markets covered	10 s polling interval	24/7 unattended runtime	2 years of development
Python	cTrader	IC Markets	Binance API
FXBlue	Sentiment analysis	NLP	SQLite / CSV
VPS	cron		

Client work & e-commerce

Commercial engagements, most of them still in production



camu.fi

2020 — 2023

Online store with 8,000+ products and continuous POS synchronisation

Developer & maintainer

- XML-based product and stock synchronisation with the POS every five minutes — millions of SQL statements a day without slowing the storefront down.
- Conversion lifted from 0.2 % to 1 %: SEO, page speed, Redis object cache, image optimisation and a redesigned purchase funnel.
- 14 separate background services: product sync, order export, stock updates, image repair and optimisation, SEO scripts, a Shopping feed, database cleaning and cache warming.

WordPress WooCommerce PHP MySQL Redis Python XML SEO



westface.fi

2022 — 2026

Facade-panel store integrated with the Visma Nova / Loocos ERP

Developer & integration owner

- Two Python microservices connect the store to the ERP: an importer unpacks the Nova archive and upserts the catalogue through the WooCommerce REST API (stock, price, status, categories, brand, facade attributes); an order exporter runs the other way.
- Multilingual with WPML, SEO with Rank Math and WooCommerce HPOS order storage. Product JSON-LD was fixed on a single-source principle so search engines never see two competing Product entities.
- A full local Docker replica of the production store so changes can be tested without touching sales — including cache purging and URL rewriting.

WordPress WooCommerce PHP 8.2 MySQL Python WPML Rank Math Visma Nova Docker



whitebalance.fi

2022 — 2026

Climbing and outdoor gear store (operated by Kiipeilytekniikka)

Developer

- The same Nova / Loocos integration pair as westface, adapted to a different attribute model (brand, colour, size) and a different order-storage mode.
- A runtime translation layer with TranslatePress — products are stored in Finnish only, so no duplicate post is created per language.
- A local development stack with its own shims (REST authentication without HTTPS) clearly marked local-only so they can never reach production.

WordPress

WooCommerce 8.4

PHP 8.2

MySQL

Python

ACF

TranslatePress

Rank Math

Docker



tilasa.fi

2018

Website and store for Italian space-saving furniture

Design & build

- Wall beds and murphy beds as product categories, brochure downloads and a financing application straight from the site.
- Built to last: the site has been in uninterrupted production since 2018 and still runs in 2026 without a rebuild.

WordPress

WooCommerce

PHP

SEO



Finnish National Opera — campaign

2019

A targeted digital advertising campaign and ticket-sales funnel

Campaign design & optimisation

- Audience segmentation and per-channel messaging on LinkedIn Ads and Google Ads, with conversion optimisation at every step of the ticket-sales funnel.

LinkedIn Ads

Google Ads

Analytics

CRO



Palsbo — space management and form protection

2021 — 2026

Excel import of premises, space groups and spam protection

Developer

- Recurring Excel imports of premises lists are cleaned and normalised; the space-group data model was designed as a written spec before implementation.
- Spam protection on contact forms with Cloudflare Turnstile, with the plan and spec written down before any code.

PHP

WordPress

Python

Cloudflare Turnstile

XLSX



accounta.fi

2021 — 2023

SEO in the accounting industry — one of the most contested niches in Finland

SEO strategy & execution

- A cluster tree of roughly 400 keywords covering the whole industry: bookkeeping, accounting firms, payroll, financial statements, VAT and company formation — each branch mapped to its own page intent, from service pages to the blog.
- A competitor analysis of some 50 industry domains (Accountor, Azets, Rantalainen, Talenom, Netvisor, Visma, Zervant among them): organic traffic, referring domains and backlink depth — what is worth attacking and what is not.
- Results tracked against Search Console data rather than estimates — ranking and click development was recorded and the keyword list revised accordingly.

SEO

Search Console

Keyword research

Backlink analysis

Content strategy



Betonikoneet.fi & Liikuntasaumat.fi

2019 — 2021

Search and Google Ads visibility for construction machinery and joint contracting

SEO & Google Ads

- Keyword analysis and campaign structure for both niches, technical SEO fixes and expanding the product pages to hundreds of items.
- A walk-through of competitors' backlink profiles on the expansion-joint side: where the industry's links actually come from and which of them are reproducible.

SEO

Google Ads

WordPress

Analytics



The lead machine — from handwork to a platform in six years

2019 — 2026

The same problem solved again and again until it became a product

Developer

- 2019–2020: the first parsers and a draft lead CRM, target-group analysis, call scripts and lead-tracking sheets; a Helsinki events parser and a niche survey of the Nordic company registries (Proff.no/dk/se).
- 2021–2022: Fonecta and Finder parsers and a full construction-industry company database — 400+ companies with names, domains, decision-makers, revenue brackets and business IDs. On top of that, automated vacancy mailing.
- 2026: the same need, productised — 17 lead sources, an enrichment chain, campaign sequences and a sales pipeline in one platform. Six years of handwork crystallised into a system.

Python

Finder.fi

Fonecta

Proff

PRH / YTJ

Web scraping

CRM



kiipeilytekniikka.fi

2025 — 2026

Site and reference material for a rope-access company

Maintenance & local dev stack

- A local Docker clone of the production site with database import, so changes can be made safely before release.

WordPress

PHP

MySQL

Docker

+ 74+ further delivered projects over the years

Tooling & infrastructure

What I built so that everything else became possible



AutoCrawler — self-driving QA agent

Detects a web app's framework, statically extracts its routes, forms and selectors, boots it with Playwright, crawls it with LLM-assisted form filling and vision-based anomaly detection, and produces SQLite + HTML reports. Adapters for eight frameworks (Django, Flask, FastAPI, Laravel, React, Vue, Next.js, Nuxt), eight fixture apps and integration tests that assert at least 80 % route coverage.

Python 3.11

Playwright

SQLite

LLM vision

pytest



gromi-core — shared Python core library

A multi-provider LLM client (Claude, Gemini) with cost tracking, forced JSON completion and retry/fallback logic; structured JSON logging; layered configuration (env → .env → pyproject); tenacity decorators tuned for LLM APIs. Published as a private package that other projects install as a dependency.

Python

Claude API

Gemini API

tenacity

GitHub Packages



gromi-prompts — versioned prompt registry

Prompts are versioned artefacts rather than strings buried in code: each has YAML metadata, a version history and at least ten JSONL test cases. A CLI runs the evals, compares versions against each other and dry-runs a render without calling an LLM. The registry holds 19 prompt families — expense classification, receipt recognition, facade analysis, outreach personalisation, SEO content, product SKUs.

Python

JSONL evals

YAML

CLI

GitHub Packages



myaitools — a self-growing tool library

Implements the LATM loop ("LLMs as Tool Makers"): recognise → specify → build → test → document → register → use. At session start the tool catalogue is injected into context; at task end a cheap local heuristic decides for free whether some subtask was worth capturing as a tool — and if so, a detached builder agent writes it with tests and registers it. Reasoning is paid for once; the tool accrues.

Python

Bash

Claude Code hooks

JSON manifest

pytest



proxybroker — rotation coordinator

The only process that calls the providers' rotation endpoints. Providers rate-limit rotation (one per two minutes), so concurrent callers get 429s and the exit IP freezes — and one service rotating mid-request breaks another. The broker serialises, cooldown-gates and barrier-coordinates rotation by quorum: the IP changes only once every connected service has voted for it, or when a safety valve fires. A stuck service can never deadlock the others.

Python

HTTP daemon

flock

cron

curl_cffi



Shared development infrastructure

One Docker stack (MySQL 8, Redis 7, Mailpit) serving every project on a shared network: databases are created by script, each project joins the external network, and its launcher brings the infrastructure up automatically if it is not already running. Seven projects share the same stack with no port conflicts.

Docker Compose

MySQL 8

Redis 7

Mailpit

Bash



Firefly III finance stack

A local Firefly III deployment with the Data Importer, an AI categoriser and an email reporter. A one-command launcher handles database import, container startup and the health check — this is the target the accounting automation publishes transactions and receipts into.

Docker Compose

Firefly III

MySQL

Mailpit



AI-agent contract and learning pipeline

A single binding operating contract (AGENTS.md) mirrored to every AI coding tool — Claude Code, Copilot, OpenCode, Cursor, Windsurf, Codex, Aider — so the work stays repeatable regardless of which tool does it. On top of that a three-layer Obsidian knowledge vault (modules, sublists, specs) and architectural decisions as ADRs. The same repository hosts a learning pipeline: summaries, flashcards and NLP maps.

Markdown

Obsidian

Node.js

Python

ADR



autoreger — API quota orchestration

Orchestrates registrations and quotas across five enrichment providers (People Data Labs, Apilayer, Apify, Hunter, ZeroBounce) in two modes: burst drains one provider at a time, pulse cycles one key per provider for a flatter load profile. Two root orchestrators and five per-project bootstraps, all logged.

Python

Bash

Selenium

Proxy rotation

cron



MailFlow connector automation

A semi-automatic tool for enrolling mailboxes into a warmup service. One headful Chrome on a persistent profile through an authenticated proxy: you log in once, then the tool walks the mailboxes one at a time, drives Gmail's app-password wizard and prefills the form. The proxy IP is rotated once per session, respecting the cooldown.

Python

Selenium

Chrome profile

Proxy

Personal projects & experiments



krisha_hunter

An apartment-hunting pipeline for Almaty: crawl the listings → a text filter with alias and negation detection → Gemini vision over the photos → a knowledge base of complexes and districts → a 0–100 score → Telegram. Geodata from 2GIS and Google Routes. Roughly \$0.001–0.003 per listing.

Python

Gemini 2.5 Flash

2GIS API

Google Routes

SQLite

Telegram



pupu_bot

A browser bot that actually fixes headless detection instead of working around it: a persistent profile, a one-time manual login, session export and restore every 30 seconds, and a doctor command reporting Chrome version, viewport, User-Agent leakage and geolocation. A browser-free jsdom suite covers 34 checks.

Python

Selenium

Chrome

jsdom

Bash



bibi_bot

Telethon-based Telegram automation for a bot driven by a reply keyboard. Quota exhaustion is detected and the work resumes by itself after the wait; a preflight command validates configuration, session and card parsing before a run. Unit tests cover the whole decision logic.

Python

Telethon

unittest



iOS WebView app and App Store release

Wrapping a web service into a native iOS app around a WebView, and the whole release path to the App Store: certificates, provisioning, the review process's requirements and the part of the work nobody anticipates before their first rejection.

iOS

WebView

App Store Connect



Social-platform automation

A series of small browser and API automations for various platforms (Instagram and Tinder among them) — exercises in keeping a session alive, detecting quota exhaustion, and making an automation break loudly rather than silently when the UI changes.

Python

Selenium

REST APIs



crimsonalter

A XenForo forum parser and cloud-download manager: keyword topic search, size estimation before downloading, downloads from Mail.ru Cloud and Yandex.Disk, everything filed by topic. Two different forum engines, one of them behind DDoS-Guard and a captcha, which needed its own browser-based bootstrap handshake.

Python

curl_cffi

Playwright

XenForo

How I work

The principles visible in every project above



Architectural decisions get written down

Every non-obvious choice ends up in an ADR file: what was decided, which alternatives were weighed and why. Six months later nobody has to guess.



Evidence before assertions

Nothing is declared done until the tests have run and the output has been read. Health-check commands are part of every stack, not an afterthought.



Idempotency by default

Re-running an accounting job, a product sync or a lead import must never create duplicates. External identifiers and dry-runs are the rule, not the exception.



AI as a tool, not a shortcut

LLM calls are versioned, tested and cost-tracked. Prompts live in a registry with eval cases — like any other piece of production code.



A reproducible environment

Every project starts with one command on a clean machine. Shared infrastructure removes port conflicts and "works on my machine" conversations.



Work is split to run in parallel

Large efforts are split into independent work packages with their own file, owner and acceptance criteria — so several developers (or agents) can advance at once without colliding.

Education

Bachelor of Science — AI in Business

[IU International University of Applied Sciences](#)

2024 — Present · Distance learning

Artificial intelligence in business: data analytics, machine learning, digital business models and project management.

9+ years of production code

[Self-taught & practice](#)

2017 — Present

Most of the skill set was acquired by building systems for clients that had to work and then keep working for years.

Languages

Finnish		Native
English		Fluent
Russian		Advanced
Swedish		Beginner

References

Jyri built the tilasa.fi website for Tilasa back in 2018. It still runs in 2026. I recommend Jyri for his expertise and the quality of his work.

Jussi Jokinen

Tilasa Oy

Delivers what he promised — on time and with care. Thank you!

Miika Junttila

Peer Car Share Oy

Interests

 Combat sambo (unarmed self-defence)

 Boxing

 Kickboxing

 Building my own tooling

About me

I do my best work on hard problems and I enjoy untangling complicated ones. Automating a banking integration, architecting a scalable SaaS product, optimising bulk processing — I bring the same energy to all of it. I like building boring software, because boring software is what keeps society running smoothly.

What motivates me most is the moment a tangled whole turns into a clear, dependable system. Good software is not merely technically correct — it is understandable, maintainable and durable over the long run. That is why I invest in clean architecture, testability and automation.

Let's talk

Tell me what needs building — I answer the same day.

 jyri.gromyko2002@gmail.com

in linkedin.com/in/jyri-gromyko

 cv.gromi.fi

 Helsinki, Finland

This CV is generated from a single source in three languages. · cv.gromi.fi