

Stanislav Nabatnikov — AI Integration & Data Engineer

Cherkasy, Ukraine | [LinkedIn](#) | [GitHub](#) | [Medium](#) | email: emulsion9000@gmail.com

Professional Summary

AI Integration & Data Engineer with a strong foundation in Async Python, serverless GCP infrastructure, and database systems. Specialized in building code-first AI automation workflows, agentic multi-model pipelines, and resilient event-driven ETL architectures.

Unlike prompt-only or low-code builders, I leverage a production-grade Data Engineering background (async pipeline design, idempotency patterns, PostgreSQL/Firestore, Docker, and GCP Serverless) to architect reliable, secure, and cost-efficient LLM integrations. Active technical writer on Medium and Dev.to analyzing AI agent architectures and cloud software design.

Core Competencies

Domain	Skills
AI & Agentic Engineering	Custom LLM API integration (DeepSeek, Mistral), Multi-Agent Systems (OpenCode, CrewAI), RAG architectures, prompt tuning, hybrid PDF/OCR processing
Data Engineering	Async Python (asyncio, aiohttp), PostgreSQL / PostGIS, Firestore, ETL/ELT pipeline design, streaming generators (yield), idempotency & CDC patterns, data validation
Cloud Platform (GCP)	Cloud Run, Pub/Sub, Cloud Functions, Firestore, Cloud Storage (incl. Object Composition), Secret Manager, Cloud Build CI/CD, Cloud Logging, IAM, Artifact Registry
Systems, Containers & OS	Native Linux (POSIX CLI), Bash scripting, systemd units, process signals & automation, Docker/ Podman (multi-stage builds)
Security & Reliability	Secret Manager credential management, path traversal prevention (is_safe_tmp_path), WSGI host-header middleware, least-privilege IAM
Technical Writing	Developer documentation, knowledge base & wiki ownership, system audits, published technical articles on Medium & Dev.to

Independent Engineering Projects

Agentic OpenCode Factory — Autonomous Multi-Role Code Factory

Python, OpenCode, LLM API Integration, Linux (POSIX), Bash, Agentic Workflows

github.com/pathexplorer/agentic-opencode-factory

- Designed an autonomous isolated code factory orchestrating 5 specialized LLM agent roles (Architect, Developer, Reviewer, Tester, Ops) with strict context separation
- Implemented environment isolation using XDG_CONFIG_HOME and custom CLI profiles for deterministic agent behavior and prompt execution
- Built automated workflow execution loops with structured output parsing, error handling, and file-based state persistence

Telegram Keyword Monitor — Serverless Real-Time ETL Pipeline

Python 3.11, asyncio, Telethon (MTPROTO), GCP (Cloud Functions, Firestore, Secret Manager, Cloud Build), aiohttp

github.com/pathexplorer/telegram_parser

- Designed a serverless async ETL pipeline: Extract via MTPROTO (Telethon), Transform via string normalization/keyword matching, Load via Telegram Bot API with formatted alerts
- Implemented idempotent cursor-based incremental processing (min_id) — cursor advances in Firestore only after successful delivery, guaranteeing zero duplicate alerts
- Built a hot-reloadable configuration store in Firestore for keywords and target channels without code redeployment
- Handled Telegram API rate limits via automatic backoff (FloodWaitError) and robust error degradation

BigBikeData — Event-Driven Cloud Pipeline & Telemetry Processing

Python, Flask, GCP (Cloud Run, Pub/Sub, Firestore, GCS, Secret Manager, Cloud Build), PostgreSQL, Docker

github.com/pathexplorer/BigBikeData

- Architected a two-service event-driven pipeline on GCP processing cycling telemetry data: detects and removes GPS anomalies, re-encodes clean .FIT files, and integrates with external APIs
- Implemented generator-based streaming CSV processing (yield pattern) handling multi-gigabyte files entirely in constant memory space without Pandas overhead
- Applied GCS Object Composition for server-side heatmap blob merging and designed Firestore-backed Pub/Sub idempotency with TTL document expiration
- Enforced a multi-layered security model: 5-stage path validation against path traversal, WSGI host-header middleware, and Secret Manager IAM credentials
- Automated 100% of GCP infrastructure provisioning via a 12-stage modular Bash IaC setup script

Medical Consultant — Local Multi-Model Agentic Pipeline

Python, LLM API, Local Model Integration, Structured Prompt Engineering

github.com/pathexplorer/medical-consultant

- Built a local multi-agent consultative pipeline orchestrating two independent LLM models to validate, cross-examine, and synthesize domain-specific queries
- Designed structured prompt workflows and response validation logic to minimize hallucination risks and ensure reliable outputs

Technical Writing

Published analytical articles on software architecture, AI agent internals, and cloud-native design patterns:

- “The Myth of ‘Lightweight’ AI Agents” — [Dev.to](#)
- “Why the Cline Extension Fails as an ‘Autonomous Agent’ Under Linux” — [Dev.to](#)
- “Elegant Profile Isolation in OpenCode via XDG_CONFIG_HOME” — [Medium](#)
- “Tracking DeepSeek’s API Dual Pricing in OpenCode via tokscale” — [Dev.to](#)
- “How to Save Chat from Gemini Web” — [Medium](#)

Supporting Work & Custom Utilities

- **gcp_actions** — Reusable GCP integration library (Pub/Sub, Firestore, GCS, Secret Manager) shared across production services
- **Mistral OCR Processing** — Hybrid PDF pipeline handling both digital text extraction and scanned documents via Mistral AI API
- **tokscale** — LLM API pricing config utility with hot-swappable cost adaptation

Education

- **Cherkasy State Technological University**
- **Cherkasky Polytechnical Technical School**

Languages

English (B1-B2, fluent written), Ukrainian (native)

Engineering Philosophy

- **Documentation-driven development:** Every solution backed by structured docs and comprehensive READMEs for maintainability and seamless team knowledge sharing
- **Production-first mindset:** Security, observability, idempotency, and failure handling designed in from day one — whether building AI workflows or data pipelines

Open to remote positions or hybrid roles in Cherkasy.