

ARTEOM TRIFANIUC

AI PRODUCT ENGINEER | FULL-STACK | LLM AGENTS & EVAL

arteom.trifaniuc@gmail.com | linkedin.com/in/percifali | arteom-trifaniuc.com



SUMMARY

Senior Full-Stack Engineer with a product mindset. I've built and shipped monetized LLM chat experiences end-to-end—frontend, backend, streaming, safety layers, and production hardening. I move fast, own outcomes, and translate ambiguous problems into shipped features.

TECHNICAL SKILLS

Languages: TypeScript, Python, JavaScript (ES6+), Go, SQL, React Native (Expo)

LLM / Agents: OpenAI, Anthropic, tool/function calling, prompt/versioning, structured outputs (JSON schema), agent orchestration (router/supervisor patterns), eval harnesses & regression testing, safety/guardrails

Backend / APIs: FastAPI, Node.js, REST, GraphQL (Apollo), WebSockets, streaming (SSE)

Data: PostgreSQL, MySQL, MongoDB, Redis (queues/caching), SQLAlchemy, Alembic

Frontend: React, Next.js, Tailwind CSS, Radix UI, Storybook, D3.js

Testing: Jest, Vitest, Playwright, React Testing Library

Observability: OpenTelemetry, Honeycomb, Sentry, Datadog

DevOps: Docker, AWS, Vercel, Railway, GitHub Actions (CI/CD)

Product / Analytics: Segment, Amplitude

Collaboration: Linear, Jira, Slack, Notion, Figma

RAG tooling: LangChain / LlamaIndex

Async: Celery/RQ, RabbitMQ/Kafka, Redis

Languages: English, Russian, Romanian

PROFESSIONAL EXPERIENCE

Founder / Full-Stack & AI Engineer — Multimodal AI Companion (LLM Chat + Image + Voice) Web + Telegram Bot

Jan 2024 - Present

- Built and shipped a monetized multimodal AI product end-to-end across Web + Telegram: onboarding, chat UX, image generation, voice generation, memory/personalization, credits/billing, admin tooling, and analytics.
- Implemented a scalable image generation pipeline: prompt templating, async GPU job execution, progress/status updates, and media lifecycle (store/serve via object storage + CDN-friendly URLs).
- Shipped voice generation (TTS) flows for chat responses: background rendering, caching/deduplication, and delivery optimized for both Web playback and Telegram voice messages.
- Built a memory system for personalization: durable conversation history + summarized/structured user context, with guardrails (reset/forget controls) and careful latency/cost tradeoffs.
- Implemented Telegram Bot API webhooks with idempotent processing, stateful flows (inline keyboards), and anti-abuse controls (rate limits, cooldowns).
- Delivered real-time chat via SSE streaming (and WebSockets where needed) for responsive multi-turn UX at scale.
- Designed agent/tool workflows using OpenAI tool calling + Structured Outputs (JSON Schema) to enforce strict, parseable tool outputs and reduce runtime failures.
- Used Redis queues for async execution (retries, enrichment, background jobs) and Cloudflare Workers + R2 for edge routing and media/storage workflows.
- Operated the platform at 99.9% uptime, handling ~15K messages/day and ~3K DAU, with p95 time-to-first-token ~50–150ms.
- Cut infrastructure costs by ~3× over 6 months (~67% reduction) through caching, queue-based execution, and pragmatic infra optimization.

Tech: TypeScript, React/Next.js, Python (FastAPI), PostgreSQL/MySQL, Redis (queues/caching), OpenAI (tool calling, Structured Outputs/JSON Schema), image generation (diffusion/GPU workers), TTS/voice generation, SSE, Cloudflare Workers/R2, Docker, Telegram Bot API.

Senior Full Stack / AI Engineer — Siena AI

Jan 2023 - Present

- Built end-to-end AI product features across frontend + backend: user-facing workflows, internal tooling, and API surfaces that expose LLM capabilities safely and reliably.
- Developed and maintained backend services (FastAPI/Node) and streaming endpoints (SSE/WebSockets) to support real-time AI interactions; reduced p95 latency from ~1.2s to ~300ms by optimizing execution paths and streaming.
- Implemented LLM agent/tooling with strict interfaces using Structured Outputs (JSON Schema) and tool calling; reduced tool-call failure rate to 0.4–0.6% through schema enforcement + guardrails.
- Built RAG features using LangChain/LlamaIndex and integrated retrieval into product flows to improve consistency and task completion.
- Designed and shipped evaluation & regression pipelines (golden sets, automated checks, failure clustering), increasing workflow success/accuracy to ~92–94%.
- Scaled async processing with Redis queues, supporting ~200K workflows/day and ~1.2M tool calls/day with reliability targets and safe execution controls.
- Reduced cost per 1K messages by ~3× (~67% reduction) via batching, caching, and infra-level optimization.

Tech: TypeScript, React/Next.js, Python/FastAPI, Node.js, SSE/WebSockets, Redis (queues), LangChain/LlamaIndex (RAG), OpenAI tool calling, Structured Outputs/JSON Schema

Middle Full Stack Developer — Material Bank / Compire

2021 – 2023

- Improved front-end performance for a high-volume product: TTI ↓28% and JS bundle ↓35% through profiling, refactors, and smarter rendering patterns.
- Raised delivery reliability by standardizing implementation patterns and mentoring: regressions ↓20% and onboarding ramp-up ↓30% via docs + code reviews.

Tech: React, TypeScript, performance profiling, documentation, mentoring

Full Stack Developer — Freelance

2016 - 2021

- Built React-based web apps end-to-end: landing pages, multi-step onboarding flows, dashboards, and complex form-heavy interfaces.
- Developed admin panels and back-office tools (RBAC/roles, CRUD at scale, audit logs, approvals), tailored to real operational workflows.
- Implemented e-commerce features: product catalogs, search/filtering, cart/checkout flows, promo codes, and payment integrations (where applicable).
- Built and integrated REST APIs (Laravel) for auth, user management, orders, inventory, notifications, and reporting; handled migrations and data modeling.
- Delivered data-heavy UI: tables with server-side pagination/sorting/filtering, exports (CSV/PDF), charts, and real-time status updates.
- Set up common “production basics”: error handling, logging, basic monitoring, CI/CD pipelines, and Dockerized deployments when needed.
- Owned full delivery cycle: scoping/estimation, implementation, QA, documentation, client demos, and post-launch support; averaged ~8–10 modules/features per quarter with ~93% on-time delivery.