

[LinkedIn](#) [GitHub](#) [WhatsApp](#)

Phone: +48 576 519 370
Email: info@yusypenko.com

Location: Poland, Wroclaw. Open to remote work
Work CET (UTC+2), comfortable overlap with
Timezone: US hours

Languages: English, Polish
Education: Master in Computer Science, 2021

SUMMARY

- Senior Full-Stack & AI Engineer - 8 years building production web apps in TypeScript/Node: React/Next.js frontends, NestJS/Express services, PostgreSQL/Redis data layers, with end-to-end ownership from architecture to CI/CD and performance work at 300M+ record scale. Last 2 years also shipping applied-LLM systems in TS and Python (FastAPI): RAG pipelines on pgvector, agent orchestration (LangGraph, MCP servers), tool calling, structured outputs. Treats LLM features like any production system — evals in CI (RAGAS, golden datasets), tracing (Langfuse, OpenTelemetry), cost/latency control via prompt caching and model routing.

TECHNICAL SKILLS

AI Engineering:	LLM apps in production, RAG, agent orchestration, tool/function calling, structured outputs, prompt caching, cost/latency optimization, human-in-the-loop
Agents & Orchestration:	LangGraph (StateGraph, ReAct, checkpointing), LangChain, MCP servers, Claude Code, multi-model routing
Retrieval & Vector Search:	RAG, pgvector (HNSW), Postgres FTS, hybrid search + RRF, HyDE, LLM reranking, embeddings, Qdrant
Eval & Observability:	RAGAS, LLM-as-judge, golden datasets, CI eval gates, output validation & guardrails, Langfuse, OpenTelemetry (GenAI), Arize Phoenix
LLM APIs & Serving:	Anthropic API, OpenAI API, FastAPI, Pydantic, Instructor, SSE streaming, Vercel AI SDK, Zod
Back-End & Data:	Python, Node.js, NestJS, PostgreSQL, ClickHouse, Redis, BullMQ, idempotent ETL, REST, GraphQL, WebSocket
Front-End & Infra:	React, Next.js (SSR/SSG/ISR), TypeScript, AWS Bedrock, Docker, Cloudflare, GitHub Actions CI/CD, Vercel, Railway

PROFESSIONAL EXPERIENCE

Senior AI / Full-Stack Engineer | *Company: BinaryBuilders (Germany)*

Jan 2025 – Present

- Melio MealPlan AI

Project: AI-powered meal planning platform — LangGraph agent over dual-retrieval RAG, async queue processing, real-time SSE streaming.

Key Achievements:

 - Built the meal-generation pipeline as a LangGraph StateGraph agent (single tool-using ReAct loop) in a Python FastAPI service, orchestrated from NestJS over a BullMQ + Redis queue and SSE-streams each day to Next.js as the agent runs.
 - Architected dual-retrieval RAG: lexical (Postgres FTS + an LLM reranker) for nutrition grounding and semantic (pgvector HNSW + HyDE query expansion, fused by Reciprocal Rank Fusion) for recipe ideas.
 - Integrated observability layer: structured-output validation with Instructor and Zod, RAGAS retrieval evals over a golden query set, Anthropic ephemeral prompt caching to cut retry token cost, and provider fallback across Anthropic, OpenAI, and local LM Studio.
 - Made the pipeline operable for the team: OpenTelemetry GenAI spans into Langfuse v3 for per-generation trace debugging and Postgres-checkpointed graph state (AsyncPostgresSaver) for crash-resume.

Technologies Used: Python, FastAPI, NestJS, NextJS, LangGraph StateGraph/ReAct, Anthropic API, OpenAI API, RAG, pgvector, Postgres FTS, HyDE, RRF, LLM reranking, structured outputs, Instructor, RAGAS, Langfuse, OpenTelemetry, AsyncPostgresSaver, prompt caching, BullMQ, Redis, PostgreSQL, SSE, Zod
- GovChime

Project: Federal contract-intelligence platform — Python AI service (LangGraph + RAG) over OLAP analytics across 300M+ records from SAM.gov and USAspending.gov.

Key Achievements:

 - Wired a Python FastAPI AI service (LangChain + RAG over solicitation embeddings on pgvector) into the platform, surfacing AI-matched contract opportunities and classifications to users over real-time streaming.
 - Owned end-to-end architecture of the Next.js + NestJS app and the ETL pipeline from the SAM.gov and USAspending.gov APIs into PostgreSQL, applying CQRS with a ClickHouse OLAP read projection: dedup on natural keys, retry-with-backoff on source flakiness, and per-award-type schema-normalization adapters as government feeds drift.
 - Modeled the analytics layer in PostgreSQL and ClickHouse with materialized views pre-aggregating by agency, NAICS, geography, and award type behind a 4-level query cascade, cutting headline dashboard queries from ~5s to under 200ms (p95) with no new infrastructure.
 - Established the team's agentic dev workflow: Claude Code plus custom MCP integrations plus TDD validation loops (test specs written first, AI-generated code checked against them), compressing multi-week features into days without quality regressions.
 - Also shipped an internal opportunity-discovery agent (LangGraph + Qdrant) with human-in-the-loop approval.

Technologies Used: Python AI Service, FastAPI, LangGraph, RAG, pgvector, Langfuse, Next.js, NestJS, PostgreSQL, ClickHouse, Materialized Views, OLAP, idempotent ingestion/ETL, ParadeDB BM25, Cloudflare Workers, OpenNext, Komodo, Docker, Hetzner bare metal, OpenAI API, Claude Code, MCP, Qdrant, human-in-the-loop approval
- Feels Protocol

Project: Solana DEX trading platform — Next.js 16 trading app with a Rust off-chain indexer and end-to-end type-safe contracts.

Key Achievements:

 - Owned both the data pipeline and the frontend for a Solana CLMM DEX: built the Rust (tokio) off-chain indexer that streams on-chain state from a validator via a Geyser gRPC plugin — updates land at transaction-processing time with zero polling — and fans it into a 4-tier store (PostgreSQL for OHLCV/markets/swaps, RocksDB for raw snapshots, Redis hot cache, Tantivy full-text search).
 - Led architecture and end-to-end delivery of the feels-web trading app on Next.js 16 App Router, React 19, and TypeScript strict with a Bun-first toolchain.

Technologies Used: Rust, Tokio, Solana, Anchor, Yellowstone gRPC (tonic/prost), Axum, PostgreSQL, SQLx, RocksDB, Redis, Tantivy, Prometheus, Next.js 16, React 19, TypeScript, OpenAPI, openapi-typescript, schemars, end-to-end type-safe codegen, Playwright, LiteSVM, Solana wallet adapters

Senior Full-Stack Developer | *Company: Filament (Germany)*

Nov 2023 – Dec 2024

Project: B2B Web3 airdrop governance platform with multi-phase voting and automated on-chain token distribution.

Key Achievements:

- Designed the NestJS backend for B2B Web3 airdrop governance: campaign lifecycle, multi-phase voting logic, and automated on-chain token distribution via Ethers.js.
- Owned the Next.js frontend for campaign management and voting flows; cut bundle size 45% and First Contentful Paint by 1.2s through route-level code splitting, lazy loading, and image optimization.

- Built real-time transaction tracking over WebSocket with event-driven architecture and an Adapter pattern unifying Ethereum + Hub blockchain handling behind a single client interface.

Technologies Used: NestJS, Node.js, PostgreSQL, Redis, TypeScript, Ethers.js, WebSocket, Clerk, REST API, Next.js, Tailwind CSS

Senior Full-Stack Developer | *Company: Primsell (Estonia)*

Nov 2022 – Nov 2023

Project: B2B Web3 e-commerce platform enabling brands to launch and manage NFT-based campaigns.

Key Achievements:

- Built the B2B NFT campaign creation flow for catalogs of 10K+ items on React/Next.js + Redux, with IPFS storage for assets and wallet-connected checkout via WalletConnect and MetaMask.
- Mentored 3 junior engineers and authored the team's TypeScript + React component conventions, adopted across the codebase.

- Owned the storefront's Web3 wallet integration: Ethereum signing flow, transaction state machine, and Sentry-instrumented error tracking surfacing wallet UX regressions within a release cycle.

Technologies Used: NestJS, Node.js, PostgreSQL, Redis, React, Next.js, Redux, TypeScript, Web3.js, WalletConnect, Metamask, IPFS, AWS, Jest, Sentry

Senior Full-Stack Developer | *Company: LITSLINK (Ukraine)*

Feb 2020 – Nov 2022

Project: Gamification platform boosting industrial-employee productivity by turning workplace activity into rewards and leaderboard progress.

Key Achievements:

- Developed 20+ Chart.js modules handling complex KPI visualizations across operator and admin dashboards.
- Integrated real-time WebSocket connection for live leaderboard updates across plant-floor displays.

- Cut dashboard load time 35% via lazy loading and memoization of heavy chart components.

Technologies Used: React, Next.js, TypeScript, MobX, Chart.js, Chakra UI, Node.js, NestJS, PostgreSQL, Redis, Docker

Front-End Developer | *Company: GeeksForLess Inc (Canada)*

Mar 2018 – Feb 2020

Project: Healthcare analytics dashboard for Canadian hospitals with real-time metrics and benchmarking.

Key Achievements:

- Built 20+ interactive Chart.js modules across 100+ unique KPIs for a Canadian-hospital analytics dashboard, with responsive layouts for clinical tablet use.

Technologies Used: Svelte, React, TypeScript, MobX, Chart.js, Responsive Design, Performance Optimization

TECHNICAL SKILLS — FULL REFERENCE

AI Orchestration & Agents:	LangGraph, StateGraph (nodes, conditional edges, reducers), LangChain, ChatAnthropic, ReAct loops, function calling / tool use (bind_tools, tool_choice), supervisor-workers (Send), AsyncPostgresSaver checkpointing, human-in-the-loop approval, AI Agents, Claude Code, MCP servers
LLM Integration:	Anthropic API, OpenAI API, prompt caching (ephemeral cache_control), multi-model routing, structured outputs (Instructor, Zod), real-time AI streaming (SSE)
RAG & Retrieval:	RAG pipelines (lexical + hybrid semantic), pgvector (HNSW), Postgres full-text search (tsvector, ts_rank, GIN), embeddings (text-embedding-3-small), hybrid search + RRF fusion, HyDE query expansion, LLM reranking, Recall@k, Qdrant, LlamaIndex ingestion pipelines
LLM Evaluation & Observability:	LLM-as-judge, ground-truth fact-checking, RAGAS, golden datasets, agent-trajectory metrics, output validation / grounding & faithfulness checks, guardrail patterns, CI eval gates, Langfuse (self-hosted), OpenTelemetry (GenAI semconv / OTLP), Arize Phoenix
Python & AI Backend:	FastAPI, asyncio, Pydantic, Instructor, scipy
Languages:	JavaScript/TypeScript, Python, SQL
Back-End:	Node.js, NestJS, Fastify, Express.js, PostgreSQL, ClickHouse, Redis, RabbitMQ, MongoDB, Prisma ORM, BullMQ
APIs & Protocols:	REST, GraphQL, JSON-RPC, WebSocket, SSE, end-to-end type-safe contracts (OpenAPI/openapi-typescript codegen), shared Zod schemas, Swagger, OpenAPI
Front-End:	React, Next.js (SSR/SSG/ISR), Svelte, SvelteKit, SEO, Internationalization (i18n)
UI & Styling:	HTML5, CSS3, SCSS, Tailwind CSS, Shadcn/ui, Framer Motion, Chart.js, Recharts
DevOps & Infra:	Docker, AWS, Vercel, Cloudflare (Workers, R2, KV, Durable Objects, Zero Trust, Tunnels), GitHub Actions, CI/CD Pipelines, Railway, pnpm monorepo
Analytics:	PostHog (product analytics, conversion funnels), Sentry
Testing & Quality:	Jest, Vitest, Playwright, Unit Testing, Integration Testing, TDD