RÉSUMÉ

GIORGI GIORGOBIANI

Senior AI Platform Engineer · New York, NY · Remote (US)

Senior AI platform engineer with nine years building production systems, specialised in LLM and agent infrastructure. Architect of multi-tenant conversation-intelligence and multi-agent orchestration platforms, and of a Rust-based cloud backend for real-time voice AI. Depth in agent runtimes, retrieval-augmented generation over vector search, streaming voice pipelines, multi-tenant data isolation, and infrastructure-as-code on AWS.

EXPERIENCE

Sunny Labs Inc.

Founder & Principal Engineer · Aug 2025 – Present

- Architected an AI companion platform end to end in Rust (tokio, axum, tonic): custom ESP32-S3 hardware, a three-tier memory system, and a multi-tenant cloud backend licensed as a white-label platform to third-party brands.
- Built three memory tiers queried in parallel every turn — Redis for session state and hot facts (<5ms), Qdrant for episodic, semantic and emotional memory ranked by similarity (<50ms), and an Amazon Neptune knowledge graph for entities and relationships (<150ms). Episodic memories decay on a 30/90/180-day schedule by domain; anything accessed repeatedly and high-confidence is promoted into the permanent graph.
- Built a streaming voice pipeline (streaming STT → LLM → custom TTS) tuned to a 520ms latency target against an 800ms hard budget, with self-hosted LLM, embedding and emotion-model inference on AWS SageMaker (Inferentia/Trainium) behind a trait-based provider abstraction with multi-AZ failover.
- Enforced per-child memory isolation across licensed brands with a separate Qdrant collection and a separate Neptune subgraph per child. Internal services communicate over gRPC/tonic, with WebSocket binary framing for the real-time audio protocol and OpenTelemetry/X-Ray tracing across the critical path.

Builders Studio

Senior Platform Engineer · Nov 2025 – Jul 2026

- Architected a multi-tenant conversation-intelligence platform converting ~100 hours of calls a week into structured business intelligence for several teams.
- Built a 9-worker, queue-driven pipeline across 7 queues with 7 dead-letter queues, bounded retries, and a shared concurrent message-pool library as the reliability backbone.
- Designed a 4-stage signal promotion ladder that never demotes, lifting per-call learnings into cross-call signals and 31 prerequisite-chained artifact types evaluated in 2 database queries.
- Built retrieval over 3,072-dimension embeddings in pgvector across transcript segments, summaries and reflection insights.
- Shipped a chat assistant with a 24-tool function-calling registry using thinking-token streaming and parallel tool dispatch over SSE, plus an MCP server exposing the intelligence layer to external agents.
- Guaranteed tenant isolation via PostgreSQL row-level security over a 50+ model Prisma schema in a single-deployment architecture, with all infrastructure as code at full dev/prod parity.
- Cut compute cost ~94% by moving off per-task serverless containers to 3 EC2 capacity providers tuned per workload shape, autoscaled on queue backlog per task rather than CPU.

Cere Network Senior Software Engineer · Apr 2025 – Oct 2025

- Built the NLP vertical on a decentralized data platform: a 10-agent pipeline where a meta-orchestrator fans batched events out to 7 model-backed agents and 3 coordination agents in parallel, merging into a social graph through a single writer.
- Implemented content-hash idempotency making the pipeline fully replayable — Postgres can be wiped and rebuilt from source events with identical output.
- Built semantic retrieval over 384-dimension embeddings in pgvector with dynamic topic clustering at a 70% similarity threshold, over a 27-table schema with 49 indexes.
- Exposed 5 query operations as MCP tools through the platform gateway, making the social graph directly queryable by external LLMs.
- Declared pipelines as infrastructure-as-code (CDKTF, custom resource types), so a new pipeline ships as a stack file and deploys without recompiling any backend service.
- Integrated the NLP vertical against a Substrate-based chain (five custom runtime pallets) and a signed-event ingestion pipeline — ed25519/sr25519 client signing, content-addressed storage — consuming from the same Kafka Streams topics and RocksDB-backed state store the platform's stream-ETL layer runs on.

Also read and integrated directly against, without owning: Go (the platform's unified ingestion/compute engine), Kotlin (Quarkus webhook service, Kafka Streams ETL topology).

YellowPad Founding Engineer · Apr 2024 – Feb 2025

- Founding engineer at a legal-AI startup; built the full platform from zero (AWS, Next.js, Node.js, TypeScript), serving 2,000 users.
- Built the document-intelligence pipeline behind contract due diligence: parsing and chunking, embeddings, and LLM-driven extraction, cutting the manual review hours attorneys spent per deal.
- Orchestrated that pipeline as coordinated Lambdas with SQS queueing, offloading heavier LLM processing to EC2 where Lambda limits bound throughput.
- Designed AppSync GraphQL endpoints for real-time updates and offline sync, cutting API response times 40%.
- Led the company through SOC 2 Type 2 certification.

Remotasks (Scale AI) NLP & Prompt Engineering, Contract · Sep 2023 – Apr 2024

- Developed prompt strategies and evaluation loops for LLM output quality, iterating on model behaviour against project-defined objectives.

Genentech Software Engineer · Aug 2021 – Aug 2023

- Built a model-federation dashboard letting researchers schedule, sequence and automate model-training jobs, removing most of the manual coordination from the training workflow.
- Visualised the full training-node topology and inter-job relationships as an interactive browser graph over Neo4j.
- Built full-stack applications (React/Redux, Java/Vert.x, Node.js, GraphQL) deployed on AWS with Docker and Kubernetes.

StoreTasker Software Engineer · Jul 2019 – Jul 2021

- Built a Ruby proxy layer for frontend request capture and third-party integration, and led the team's adoption of React and Node.js.

Biz2Credit Software Engineer · Jul 2017 – Jul 2019

- Contributed to a monolith-to-microservices migration (Node.js/Express, Ruby on Rails), built React/Redux/TypeScript frontends, and deployed serverless services on AWS Lambda.

SKILLS

AI & LLM SYSTEMS

Multi-agent orchestration, agentic workflows, agent runtimes, RAG pipelines, vector search (pgvector, Qdrant), embeddings, semantic clustering, MCP servers (client + server), streaming STT/TTS, self-hosted inference (AWS SageMaker, Inferentia/Trainium, AWS Neuron SDK), prompt versioning, LLM observability (OpenTelemetry, AWS X-Ray), cost and latency optimization

CLOUD & INFRASTRUCTURE

AWS (ECS on EC2, SQS, Lambda, S3, RDS, SageMaker, ElastiCache, Neptune, AppSync, CloudWatch, X-Ray, Route 53, WAF, Secrets Manager, ECR, EC2, VPC, CloudFront), CDKTF (incl. authoring custom Terraform providers), Pulumi, Docker, Kubernetes, GitHub Actions, GCP

DATA

PostgreSQL (row-level security, pgvector), Redis, Qdrant, Neo4j, Amazon Neptune, DynamoDB, MongoDB, Elasticsearch, Kafka (incl. Kafka Streams), RocksDB, Prisma, TypeORM

LANGUAGES

Rust (tokio, axum, tonic), TypeScript, JavaScript, Python, Go, Kotlin, Java, Ruby, SQL, Cypher/openCypher, C/C++ (embedded firmware)

BACKEND & API

Node.js, NestJS, Express, axum, tonic (gRPC), tokio, FastAPI, GraphQL, AppSync, Flask, Vert.x, WebSocket protocols, JSON-RPC

PROTOCOLS & CRYPTOGRAPHY

ed25519/sr25519 signing, Curve25519, content-addressed storage (CID), Substrate/Polkadot SDK (integration), HMAC request signing

FRONTEND

React, Next.js, React Native, Redux, Tailwind, D3.js

EDUCATION

APP ACADEMY

Full-Stack Web Development, New York, NY — 2017

SUNY PLATTSBURGH

B.S. Business Administration, New York — 2015