

Daniel Nguyen

Senior Full-Stack & AI Engineer

Houston, TX | (325) 222-9022 | hello@danielhnguyen.net | <https://www.linkedin.com/in/danielnguyen714>

PROFESSIONAL SUMMARY

Senior Full-Stack & AI Engineer with over 11 years of experience architecting high-scale SaaS platforms and autonomous agent systems. Expert in bridging traditional distributed systems with modern generative AI, building robust infrastructure for real-time Voice AI, conversational agent platforms, and financial automation. Proven track record of optimizing low-latency data pipelines, designing resilient event-driven architectures, and shipping production-grade applications that integrate with legacy enterprise systems. Specialized in system reliability, cloud infrastructure, developer tooling, and delivering autonomous business logic that drives operational efficiency at scale.

TECHNICAL SKILLS

Languages: Python, JavaScript, TypeScript, Java, SQL, Bash, HTML, CSS

Frontend: React, Next.js, Angular, Redux Toolkit, React Query, Tailwind CSS, SPA/SSR, Virtualized Rendering, Performance Optimization

Backend & APIs: FastAPI, Django, Node.js, Express.js, Nest.js, Spring Boot, REST, OpenAPI/Swagger, GraphQL, gRPC, WebSockets, Server-Sent Events (SSE), Celery, OAuth2/JWT, Middleware Pipelines

AI, LLM & Voice: OpenAI API, Anthropic API, Function Calling/Tooling, ReAct Agent Loops, LangChain, RAG, Embeddings, Vector Search, Pinecone, pgvector, Semantic Caching, Prompt Engineering, Evals Pipelines, Sentiment Analysis, Whisper (STT), Twilio Voice/Media Streams, NLP, Multi-Model Orchestration

Data & Messaging: PostgreSQL, Redis, Apache Kafka, MongoDB, SQL Tuning, Indexing/Partitioning, Background Jobs, Idempotency Patterns, Distributed Locking, Event-Driven Architecture, Pub/Sub

Cloud & DevOps: AWS (ECS, Lambda, RDS, ElastiCache, S3, CloudWatch, IAM, VPC), Docker, Kubernetes, Terraform (IaC), CI/CD (Jenkins, GitHub Actions), Linux, Git, Ephemeral Environments, Canary Rollouts

Payments & Integrations: Stripe Connect, Split-Payment Orchestration, PCI Compliance, PII Redaction, FMS Integrations (StorEdge, SiteLink, Storage Commander)

Testing & Reliability: Pytest, JUnit, Mockito, Jasmine, Karma, Integration/Regression/E2E Testing, Load/Latency Testing, Feature Flags, Incident Response, Monitoring/Alerting, Distributed Tracing, SOC 2 Compliance

PROFESSIONAL EXPERIENCE

Swivl | Sr. Full Stack & AI Engineer

Jan 2025 - Present

Led development of swivl's real-time Voice AI Agent and full supporting infrastructure for the self-storage conversational AI platform. Owned the voice pipeline end-to-end from Twilio telephony ingestion to LLM response, built bi-directional FMS integrations, and stood up the AWS cloud infrastructure and CI/CD workflows serving 4,500+ facilities.

- Architected swivl's Voice AI Agent in Python/FastAPI that ingests Twilio Media Streams as base64 mulaw audio, runs Whisper STT, and feeds transcripts into an LLM response loop with barge-in handling, cutting live-agent routing by 40% QoQ at sub-500ms end-to-end latency
- Designed a Redis Semantic Caching layer that hashes caller intent embeddings and returns stored responses for high-frequency tenant questions (gate codes, balances, office hours), bypassing OpenAI inference on 35% of inbound calls and shaving 600ms off p95 latency
- Built the intent classification engine using LLM function calling that routes caller utterances to autonomous actions (payments, access code lookups, autopay enrollment), knowledge-base retrieval (facility FAQs, move-in instructions), or live-agent escalation with full context summaries
- Assembled the RAG knowledge retrieval system using pgvector that chunks and indexes facility-specific FAQs, storage policies, and unit pricing, grounding every agent response in each property's actual data rather than generic answers
- Shipped the Next.js Live Monitor dashboard with React Query and SSE that streams real-time call transcripts, displays LLM-powered sentiment scores to flag negative tonal shifts, and gives property managers toggles to control which intents the AI handles versus escalates
- Crafted swivl's unified CRM inbox in React and Tailwind CSS that aggregates calls, SMS, emails, and webchat into a single timeline per tenant, with AI-generated conversation summaries, suggested replies, and one-click escalation to route unresolved threads to the next available agent
- Developed the call analytics dashboard in Next.js with SSR rendering automation rate, reservation conversion, average handle time, and call volume per facility, with date-range filters and facility-level drill-downs backed by PostgreSQL aggregate queries
- Created the facility onboarding configuration UI in React where operators set up per-property AI behavior: toggling which intents the agent handles autonomously, uploading custom FAQ documents for the RAG pipeline, and mapping FMS credentials for StorEdge or SiteLink sync
- Engineered the bi-directional FMS sync engine connecting StorEdge and SiteLink REST APIs with distributed locking, synchronizing real-time inventory, tenant balances, and access codes across 50,000+ storage units so the AI Agent can answer unit status questions mid-call without human help
- Integrated Stripe Connect for autonomous split-payment processing where the Voice AI Agent accepts a tenant's card payment mid-call, posts it to the FMS account, and triggers a receipt via SMS through OAuth2/JWT-secured endpoints documented with OpenAPI/Swagger
- Implemented a PII Redaction middleware that strips credit card numbers and personal identifiers before they reach external LLM providers, with Celery/Redis workers handling post-call transcript generation, call scoring, and CRM updates asynchronously off the live voice path

- Provisioned the full AWS stack (ECS, RDS, ElastiCache, S3, CloudWatch, IAM, VPC) with Terraform, configured custom CloudWatch metrics and alerting for voice pipeline health (WebSocket latency, LLM response time, FMS sync failures), and wired up GitHub Actions CI/CD with zero-downtime ECS deployments
- Established feature flags gated by facility ID and canary rollout strategies, letting the team ship new AI behaviors to a single storage operator's 50 locations before expanding to the full 4,500+ facility fleet
- Wrote a telephony emulation suite mocking Twilio Media Streams and FMS state, with Pytest integration and load tests replaying recorded call audio against the full pipeline, validating sub-500ms p95 under 200 concurrent simulated calls

Tech Stack: Python, FastAPI, WebSockets, Next.js, React, TypeScript, Tailwind CSS, React Query, Redis, PostgreSQL, pgvector, Celery, OpenAI API, Twilio Voice/Media Streams, Whisper (STT), Stripe Connect, SSE, OAuth2/JWT, OpenAPI/Swagger, Docker, AWS (ECS, RDS, ElastiCache, S3, CloudWatch, IAM, VPC), Terraform, GitHub Actions, Pytest

Sierra | Software Engineer, Agent

Jan 2023 - Dec 2024

Built core agent infrastructure for Sierra's conversational AI platform serving enterprise brands like WeightWatchers, SiriusXM, Sonos, and OluKai. Owned the ReAct agent runtime, vector search retrieval layer, multi-model orchestration, and developer tooling powering multi-step customer workflows across chat, voice, and messaging channels.

- Built the core ReAct event loop in Python with WAL-style transactional state management, so failed mid-workflow actions (order modifications, subscription changes) automatically roll back to the last consistent state without leaving partial mutations in the customer's record
- Designed multi-tenant PostgreSQL + pgvector schemas for enterprise knowledge retrieval, tuning partial indexes and HNSW parameters so agents query millions of product pages and policy documents at single-digit millisecond p95 latency per brand namespace
- Assembled the RAG pipeline with document-type-aware chunking (product specs vs. return policies vs. troubleshooting guides) and hybrid keyword + vector search, preventing cross-tenant answer contamination across Sierra's brand clients
- Devised the Tool Registry that introspects Python type hints at runtime and auto-generates OpenAI-compatible function calling schemas with strict input/output validation, blocking agents from passing malformed data to CRM and order management systems
- Shipped multi-model orchestration that routes prompts between OpenAI, Anthropic, and open-source LLMs based on task complexity scoring and per-token cost constraints, reserving frontier models for multi-step reasoning and falling back to cheaper models for FAQ lookups
- Developed the React Trace Visualizer in TypeScript rendering step-level agent execution graphs with tool calls, retrieval context, reasoning chains, and per-step latency, giving prompt engineers a visual debugger to pinpoint hallucinations and slow responses
- Formalized the deterministic action execution layer with gRPC that wraps every system-of-record mutation in an audit envelope (caller identity, timestamp, pre/post state diff), ensuring compliance with enterprise security policies across brands like SiriusXM and Sonos
- Created the automated Evals pipeline running 500+ golden conversation scenarios before every deployment, catching prompt regressions and hallucination spikes by comparing agent outputs to expected actions and flagging accuracy drops above a configurable threshold
- Integrated PII redaction middleware that scrubs SSNs, credit card numbers, and emails from prompts before they leave Sierra's infrastructure for external LLM providers, with audit logging for data governance reviews
- Drove prompt engineering workflows with few-shot examples, in-context learning, and self-consistency techniques across brand-specific agent configurations, systematically reducing hallucination rates in production deployments for WeightWatchers and OluKai
- Stood up AWS infrastructure (ECS, RDS, Lambda, S3, VPC) with Terraform and Kubernetes, building ephemeral preview environments that spin up a full agent stack per pull request so engineers test brand-specific configurations in isolation before merging
- Owned conversation observability tooling with distributed tracing tracking resolution rate, CSAT, hallucination rate, and tool-call latency on real-time dashboards, enabling the CX team to compare agent performance across brands
- Introduced feature flags and canary rollouts gated by enterprise cohort, letting Sierra ship experimental agent behaviors to a single brand's traffic before exposing changes to the full multi-brand production fleet

Tech Stack: Python, React, TypeScript, PostgreSQL, pgvector, OpenAI API, Anthropic API, RAG, Embeddings, gRPC, AWS (ECS, RDS, Lambda, S3, CloudWatch, VPC), Terraform, Docker, Kubernetes, Redis, Pytest, GitHub Actions

HighRadius | Software Development Engineer II

Jan 2018 - Dec 2022

Worked across the full stack of HighRadius's Integrated Receivables platform automating AR and Treasury processes for 600+ enterprise clients including 3M, Unilever, and P&G. Decomposed monolithic systems into event-driven microservices, built high-throughput payment pipelines, and delivered front-end tools for AR analysts.

- Decomposed the monolithic payment processor into event-driven microservices using Apache Kafka with topic partitioning per payment type (checks, ACH, wire, credit card), isolating failures so a broken check parser never blocks wire transfer processing
- Constructed a high-concurrency Java Spring Boot service that parses unstructured payment data from PDFs, EDI files (BAI2, MT940, CAMT.053), and bank remittances at 10,000+ documents/hour, normalizing 50+ banking formats into HighRadius's unified receivables schema
- Replaced a legacy JSP interface with a React virtualized grid using Redux Toolkit, rendering 100,000+ invoice rows with column-level sorting, inline editing, and real-time aging calculations without scroll jank, even on datasets that previously froze above 5,000 rows
- Optimized multi-terabyte queries on the PostgreSQL data warehouse with date-range table partitioning and covering indexes for common AR report filters (customer, aging bucket, dispute status), cutting report generation from 4+ minutes to under 2 seconds
- Implemented Redis distributed locking with lease-based expiration for concurrent invoice editing, so an analyst's browser crash never permanently blocks other collectors from working the same account
- Delivered the REST inference API for HighRadius's Propensity-to-Pay predictive model with OpenAPI/Swagger docs, serving real-time credit risk scores to the collections workflow so analysts prioritize accounts most likely to pay within 30 days

- Devised automated cash application matching using fuzzy matching on remittance memo fields, check numbers, and partial amount combinations, reducing manual matching intervention across 600+ enterprise clients
- Orchestrated ETL pipelines to ingest and normalize payment data from 50+ banking formats (BAI2, MT940, CAMT.053) into a unified SQL schema, standardizing heterogeneous remittance structures so the cash application engine can match against a single canonical format
- Launched React SPA dashboards for monitoring DSO, aging buckets, and Collection Effectiveness Index using GraphQL with client-side caching, giving finance teams real-time receivables visibility without waiting for overnight batch reports
- Led the Redux migration to normalized entity state with createEntityAdapter, eliminating duplicate API calls across invoicing, collections, and cash application modules and reducing client-side memory usage by 40%
- Structured event-sourced audit trails with role-based access control (OAuth2/JWT) recording every invoice status change, deduction claim, and cash posting as immutable events, enabling full transaction replay for SOC 2 Type II certification and annual financial audits
- Wrote JUnit/Mockito integration tests covering the full payment path from remittance ingestion through cash posting at 90%+ coverage on the matching engine, catching regressions before they reach production serving clients like 3M and Unilever
- Configured event-driven notification workflows using Kafka that alert collectors when a high-value invoice ages past 60 days or when a customer's payment pattern deviates from their historical behavior, replacing a nightly batch job with near-real-time triggers

Tech Stack: Java, Spring Boot, Apache Kafka, React, Redux Toolkit, JavaScript, TypeScript, PostgreSQL, Redis, MongoDB, GraphQL, REST APIs, OpenAPI/Swagger, Python, JUnit, Mockito, Jenkins, AWS, Docker

BMC Software | Software Engineer

Jun 2015 - Dec 2017

Developed front-end frameworks and backend microservices for BMC's Remedy ITSM platform used by thousands of organizations worldwide. Migrated legacy services to RESTful architectures and established the team's first automated testing and CI/CD practices.

- Pioneered a custom Angular directive framework that reads JSON schema definitions and dynamically renders ITSM forms (approval chains, SLA indicators, escalation triggers), enabling IT admins to build new service request workflows through configuration alone, cutting form development time by 70%
- Migrated legacy SOAP/XML endpoints to RESTful Node.js/Express.js microservices with Hystrix-pattern circuit breakers that open after 5 consecutive failures, preventing a single misbehaving dependency from cascading across the Remedy platform under load
- Constructed a real-time log aggregation pipeline in MongoDB tracking automation success rates, MTTR, first-call resolution, and SLA compliance on Angular dashboards used by operations leadership to spot service degradation before it hits end users
- Wired up webhook integrations between Remedy and monitoring tools (Nagios, SNMP traps) that auto-create incident tickets from infrastructure alerts with deduplication logic checking for open incidents on the same CI and alert signature before creating duplicates
- Delivered a reusable Angular component library for ITSM form elements (approval chain widgets, SLA countdown timers, CMDB relationship pickers) standardizing UI patterns across 12+ modules in the Remedy suite
- Executed ITIL-aligned change management workflows with automated approval routing that checks the CMDB for CI dependencies, flags conflicting changes in the same maintenance window, and routes approvals to CAB members based on risk category
- Deployed server-side middleware for request validation, authentication, and rate limiting across the Node.js layer, throttling API consumers to 100 req/s per tenant to protect shared Remedy infrastructure from noisy neighbors in multi-tenant deployments
- Automated Jenkins CI/CD pipelines with Git-hook pre-commit validation running Jasmine/Karma test suites, enabling one-click staging deployments with rollback on test failures, cutting manual release effort by 30%
- Authored RESTful API endpoints in Node.js/Express.js for ITSM workflow automation, enabling third-party integrations with monitoring tools, ticketing systems, and CMDB data sources through documented REST contracts
- Mentored 3 junior developers on async JavaScript patterns (Promises, async/await, event loop behavior) through structured code reviews and pair programming, improving team velocity and code quality across the frontend codebase

Tech Stack: JavaScript, TypeScript, Angular, Node.js, Express.js, MongoDB, REST APIs, SOAP/XML, HTML/CSS, Jenkins, Jasmine, Karma, Hystrix, JSON Schema, Linux, Git, Bash

EDUCATION

University of Houston | Houston, TX

Bachelor of Science (B.S.), Computer Science | 2012 - 2015