Nguyen Duong Hong Giang

Senior AI Engineer, Agentic Systems

Agentic pipelines · AI-first SDLC · Agent evaluation and governance · LLM cost control · MCP and knowledge platforms

BASED IN Hanoi, Vietnam (GMT+7) EMAIL giangndh1810@gmail.com PHONE +84 364 474 416
PORTFOLIO portfolio.gigibrew.dev

SUMMARY

Senior AI engineer with **12+ years** of software engineering, specialising in **production-grade agentic AI systems** and **AI-native software delivery**. I architect multi-agent orchestration, **MCP-based tool ecosystems** and **AI-first SDLC automation**, with a strong focus on deterministic execution, **human-in-the-loop governance**, agent evaluation, reliability and **cost optimisation**.

Hands-on delivery of enterprise and government AI platforms (including on-premise LLMs for the State Audit Office of Vietnam), production LLM applications, knowledge systems and developer-productivity automation, mainly in **Python** (FastAPI, asyncio, background workers) and also TypeScript, with LangGraph, LangChain and MCP. Previously **Co-founder and CTO of a 40-person AI company**. I write and ship the code myself; my leadership experience goes into enabling teams, not managing them. Fluent English with EU and Korean teams.

AGENTIC ENGINEERING

Multi-agent coding pipeline over the Agent Client Protocol `self-built` 2026

`TypeScript, ACP (JSON-RPC over stdio), Claude Code (Opus, Sonnet), DeepSeek Harness, zod, macOS sandbox, Vitest`

- A **code orchestrator owns the flow**: it is the ACP client, opens one session per role and drives agents from different vendors; every reply is a JSON block validated against a schema before the next step.
- **Cross-vendor roles**: Opus plans a task DAG (or asks the user), DeepSeek plans and implements, Opus reviews, Sonnet rewrites failing plans and checks the work, Opus reviews the integration branch; independent tasks run in parallel waves.
- **Decision gates in code**: every agent permission request goes through a policy function; readers get read-only tools in their worktree, writers can edit only inside it, only the implementer gets a shell and it runs in a sandbox; acceptance commands are run by the orchestrator, never by Claude agents. A blocked run waits for a human unblock with a note.
- **Audit trail and cost**: run state, plans and an append-only event log make every decision replayable; numbered decision records; token usage per model and session; proven by fault-injection and kill-and-resume end-to-end runs.

AI-PM: project management with an MCP server for agents 2026

`TypeScript, Fastify, Prisma, PostgreSQL (pgvector), Redis, MCP SDK, React, Vitest, GitHub Actions`

- **40+ MCP tools** (create, claim, request review, approve, request changes, report blocker, daily report) so agents take tickets and move them through review. Agent writes run under an **autonomy level**: in approval mode they become proposals that wait for a human, completion requires review, and self-approval is blocked on the server.
- **Rules first, LLM on misses** for intent routing: 92.3% accuracy at zero cost with rules alone, **97.3% at about $0.11 per 1,000 messages** with the classifier fallback, on a 300-message eval over 27 routes. Test coverage on tools and approval rules, run in CI.

AI-first delivery harness for agent teams `self-built` 2026

`Python (hooks, gates, routers), Claude Code, subagents, MCP`

- **20 role-scoped agents** (explorer, researcher, code worker, planner, verifier), each pinned to a model and a minimal tool set; a **cost-aware model router** written as a Python hook tiers each delegated task and routes it to Haiku, Sonnet or Opus with a logged decision.
- **Five deterministic gates**, mostly Python scripts, that fail the run (output contracts, pre-publish sensitive-data scan, headless browser check, schema validation, release check); permission and stop events reach me in chat and replies route back to the session; one policy source synced to three agent harnesses with a drift check.

Ontology-based knowledge platform | GTV (Viec3Mien) 04/2026 - Present

`Python, LinkML, OWL, SHACL, SPARQL, JSON-LD, Oxigraph, OCR, Telegram and Zalo intake`

- Digitises the processes and know-how of a blue-collar recruitment group that lived only in **dozens of operational chat groups** and scattered files: a live listener captures the groups; an intake pipeline parses Office, PDF and scanned documents, redacts credentials, deduplicates by content hash and classifies with a confidence threshold that routes unsure items to **human review**.
- Designed the **domain ontology in LinkML** (24 classes: factories, roles, processes, decisions, meetings, documents) and generated OWL, SHACL shapes and a JSON-LD context from one schema.
- **Code decides what enters the graph, not the LLM**: agents propose facts as JSON-LD; a SHACL ingest gate rejects unknown classes, undeclared properties, missing provenance and duplicates; a SPARQL guard lets agents run only bounded read queries; ambiguous identities go to a warning queue instead of being guessed. Replayed the legacy knowledge log into the new model with zero violations.

Agent-delivered production systems | GTV (Viec3Mien) 04/2026 - Present

`Python, FastAPI, Pydantic, PostgreSQL, SQLite, MCP, OpenRouter, Claude, Node.js, React, Remotion, Playwright`

- **Video production studio** (Python, FastAPI): idea to finished 9:16 video through a multi-step graph with human approval gates; a **per-flow budget engine** reserves cost before each paid model call and settles on the real cost; Pydantic-validated structured outputs with bounded repair; an editor agent with a single read tool returns JSON that code applies. Delivered ticket by ticket by coding agents.
- **Domain-clustered testing for agent delivery**: the suite is split into domain clusters mapped to source paths, with a self-test that fails on orphan or duplicate test files. An agent ships after lint, typecheck and its own TDD tests, then only the clusters its diff touches run, under memory and time limits with a JSON report per run: full test coverage on every change without paying for the full suite on every ticket.
- **Intake, data and platform**: a chat bot turns team requests into tickets, gated by a golden-set eval that rejected two candidate models; warehouse agents reach Postgres only via a **read-only MCP server** with personal data masked; a telemetry contract ships LLM cost per call.

PRODUCTION AI PRODUCTS

On-premise meeting AI for the State Audit Office of Vietnam | PTHub (VAI-M) 2025 - 2026

`PyTorch, Whisper, PhoWhisper, pyannote, Qwen 2.5 72B, vLLM, Ollama, FastAPI, Redis Streams, GPU servers`

- Audit meetings to speaker-attributed transcripts and the official conclusion notice, fully on-premise; LLM chosen by benchmarking 5 models x 9 prompts. **Event-driven Python services**: FastAPI API and GPU inference workers linked by Redis Streams and work queues, with autoscaling workers. **Deterministic correction** of names, titles, state bodies and legal references; the LLM only drafts into a fixed template with no inferred facts; political-risk flagging with evidence review.

E-Office AI Assistant | PTHub 02/2026 - 04/2026

`Agno AgentOS, FastAPI, PostgreSQL + pgvector, Redis, Claude, Gemini, OpenTelemetry, Grafana, Kubernetes`

- A coordinator routes staff requests to **11 specialised agents**, including text-to-SQL and department-scoped hybrid RAG with RBAC; layered guardrails, SSO and OpenTelemetry tracing. Led the team.

Also: LangGraph RAG with golden-set eval (hit@5 1.00, MRR 0.98); scoped SQL agent with a deny-by-default query guard; on-premise voice agent; production computer vision.

EXPERIENCE

Senior AI Engineer, platform | GTV, Viec3Mien (AI & Technology Lead) 04/2026 - Present

- Built the group's AI platform hands-on (knowledge platform and agent systems above); set the AI roadmap from 95 mapped operational bottlenecks.

Lead AI Engineer | PTHub, AI technology company, Hanoi (Co-founder, CTO) 11/2024 - 04/2026

- Designed and built the Core AI platform (speech-to-text, assistant, RAG, OCR) and its products while growing the company to **40 people in 17 months**; enabled AI adoption across delivery teams.

Technical Consultant | GIZ AgriCRF project (BMZ-funded), Hanoi 03/2025 - 06/2025

- Microfinance app for 100,000+ clients: security assessment, core-banking integration, credit scoring.

Senior Software Engineer, remote | Dreams Quest (EU) and Kong Studios (South Korea) 05/2023 - 11/2024

- Architecture and process for a 30+ person team in English; reviews cut technical debt by about 30%.

Lead Software Engineer | Brew Monster Studio, Hanoi (Technical Director) 2020 - 2023

- Built and shipped 10+ games and real-time AR for VinGroup, BIDV and DOJI, leading a 15+ person team.

Software Engineer | BIDV, Digital Banking Center, Hanoi 12/2015 - 11/2019

- Built an **interbank transfer system at 200 TPS** (NAPAS) and core-banking integrations (**.NET**, Java, Oracle, IBM MQ).

Earlier roles | VGC Game (Technical Director), CMC Software (Programmer), Hanoi

SKILLS, EDUCATION AND LANGUAGES

Agentic and SDLC	Agent Client Protocol, MCP (servers and tools), Claude Code and subagents, LangGraph, LangChain, Agno; multi-agent orchestration, policy and approval gates, sandboxed tool use, structured outputs with repair, ticket-driven agent delivery, domain-clustered test selection
Evals, safety, cost	Golden-set evals with go/no-go thresholds, DeepEval, RAGAS, model benchmarking; guardrails (PII, SQL injection); cost-aware routing, per-flow budgets, LLM cost telemetry
Knowledge and retrieval	LinkML, OWL, SHACL, SPARQL, JSON-LD; hybrid RAG, reranking, graph-RAG; pgvector, FAISS; vLLM, Ollama, liteLLM; Claude, GPT, Gemini, Qwen, DeepSeek
Engineering	**Python** (FastAPI, asyncio, Pydantic, background workers, pytest), TypeScript (Fastify, React), .NET; event-driven design (Redis Streams, queues); PostgreSQL; Docker, Kubernetes, GitHub Actions, GitLab CI; OpenTelemetry; PyTorch (ASR, TTS), YOLOv8
Education, languages	B.Eng., Applied Mathematics and Informatics, Hanoi University of Science and Technology, 2008 - 2013; English fluent