Patrick Selamy

New York, NY | (210) 704-7370 | pselamy@gmail.com | selamy.dev | linkedin.com/in/patrickselamy | github.com/pselamy

Software Engineer | AI Code Generation, Agent Engineering, Observability, and Distributed Systems

Summary

Software engineer who builds AI-codegen and agent-engineering systems with strong observability and distributed-systems rigor. Former Meta and Google engineer who built one of the first Gemini-based software-engineering agents at Google Cloud Office of the CTO. Instruments multi-agent systems end to end with OpenTelemetry (traces, metrics, logs, and tool-call, token, latency, and trajectory telemetry) and ships CI at scale with live telemetry dashboards.

Selected Systems Portfolio

Autonomous Agent Harnesses & Loops

- Self-healing orchestrator coordinating six always-on agents through a SQLite priority queue, with multi-consumer leases, parallel workers, and dispatch/fan-out for team parallelism; tens of thousands of tasks executed over months of continuous operation.
- Watchdog that detects parked, stalled, and idle-while-work-pending agents and recovers them or enforces continued execution; reliable inter-agent messaging and resolve-or-decompose loops.
- Agents doing real work: credentialed document retrieval and prep, risk-capped live-capital market-making, CI repair, repo changes, and long-running task decomposition.

Fleet, Observability, Security & Production Rigor

- Modeled each agent as a reusable Terraform module (identity, runtime, skills, schedule, secrets), so adding or reconfiguring an agent is a declarative change rather than a bespoke rebuild.
- Fleet instrumented with OpenTelemetry (traces, metrics, logs) including agent tool-call, token, latency, and trajectory telemetry, with a live dashboard.
- Built CI-backed verification loops for autonomous engineering work, including multi-substrate testing, runner reliability, and cost-aware execution patterns.
- Built a production multi-agent fleet on Nous Research's open-source Hermes, choosing compose-over-fork so upstream improvements arrive via version bumps rather than fork maintenance.
- Per-agent OS-user and namespace isolation; GitOps and IaC with OpenTofu and Argo CD; runtime controls via flagd/OpenFeature; keyless secrets-broker custody with PAKE.

Agent Skills, MCP & Tooling

- Agent-architecture framework where MCP servers expose callable capabilities and skills encode reusable judgment and procedures, including a skill for choosing MCP vs skills.
- Public and internal skills/MCP authoring workflows with privacy scanning, security-audit gates, and broker-held credentials; agents build their own MCP servers and contribute skills back.

Experience

Meta, Software Engineer — New York, NY | 2025 to 2026

- Expanded Facebook Marketplace third-party partner APIs and shipped-listing workflows for partner and seller-facing commerce.
- Shipped an ML-assisted listing flow that removed manual item-weight entry for shipped listings; predictions were evaluated against user-entered weights and carrier actuals, met the launch threshold near 80% accuracy, and lifted new-listing creation about 8%.

Google, Software Engineer — Pittsburgh / Austin / Remote / Atlanta | 2018 to 2025

- Worked across Google Shopping (Engineering Productivity), Google Play, Google Cloud Office of the CTO, and Internal Identity and Access Management.

- In Google Cloud Office of the CTO, built early agentic-coding systems: a Gemini-based software-engineering agent, SWE-bench eval harnesses, and planning-vs-execution metrics that guided funding and staffing; benchmarked vector-search frameworks at scale (ScaNN, FAISS, pgvector) to position GCP's AI competitively.
- In Google Play, helped build Play Games Leagues from the ground up on a five-engineer team and designed a configurable fuzzer for third-party Play APIs.
- Built Insider Risk systems for internal security, enterprise identity, and trust-sensitive access.

Earlier, Backend / Platform / Healthcare Engineering — Citi, Lumeris, ClearDATA, and others

- Built backend, platform, cloud, and healthcare systems in Python and Java across APIs, data workflows, and regulated environments.

Selected Systems & Public Work

- Public agent-skills: reusable agent procedures, authoring patterns, and an MCP-vs-skills framework.
- Operating Agent Fleets: an evidence-backed field guide for durable agent systems and bounded authority.
- polymarket-insider-tracker: a Python tool that flags suspicious wallet activity on Polymarket prediction markets.
- AI instructor and speaker: led an Ayiti AI bootcamp session on MCP, AI-assisted coding, and deployment, and addressed lawmakers on AI policy at the NHAEON National Leadership Summit.

Technical Skills

Languages:	Python, Java, Kotlin, JavaScript/TypeScript, Hack, PHP, Bash, SQL
AI / Agent Systems:	AI agents, skills/playbooks, MCP-style tools, tool calling, evals, agent harnesses, autonomous coding loops, context engineering, RAG, model routing, secure tool execution, state management
Product & Systems:	API design, backend platforms, distributed systems, workflow orchestration, product metrics, experimentation, launch analysis, developer infrastructure
Security & Enterprise:	internal IAM, auth boundaries, insider-risk systems, credential brokering, trust-sensitive workflows, privacy/security review gates
Infrastructure:	Kubernetes, OpenTofu/Terraform, Argo CD, Docker, Linux, GitHub Actions, Actions Runner Controller, CI/CD, observability, multi-cloud
Data & Cloud:	PostgreSQL, Redis, Kafka, InfluxDB, Google Cloud, AWS