Jayakrishna Gandhamalla

Santa Clara, CA · +1 (813) 585-4395 · rareone.jk@gmail.com · jkzod.dev

Senior Software Engineer | Agent Platforms and Developer Tools

Senior software engineer with 5+ years building developer tools, AI agent platforms and full-stack products. Built Xpeng's internal AI coding-agent platform from 0 to 1 as lead engineer and scaled it to **1,000 users (600 daily)** and **50 billion tokens per workday**. Owned the Python agent runtime, LLM gateway, backend services, cross-platform clients and CI/release, with measured gains in latency, token use and build time. M.S. in Computer Science.

EXPERIENCE

Xpeng Motors (XMotors.ai) — Santa Clara, CA

Senior Software Engineer, AI Platform — *Sep 2026 – Present*

Software Engineer, AI Platform — *Jul 2025 – Aug 2026*

- Lead engineer and top contributor (**about 60% of commits**) for the platform's agent runtime, **8 backend services** (LLM gateway, auth, marketplace, scheduler and others) and its desktop, CLI and VS Code clients.

Agent runtime and orchestration

- Built the Python agent runtime with resumable event replay, tool approvals, interruption and crash recovery, configurable context compaction, permission-aware subagents, Git-worktree-isolated workspaces, Agent Client Protocol (ACP) support and delegation to Claude and Codex.
- Made the harness compatible with Claude Code setups: Agent Skills (SKILL.md), mcp.json, AGENTS.md/CLAUDE.md, plugins, and a .claude-style folder for skills, subagents, hooks and permissions.
- Built orchestration for long-running work: a persistent coordinator agent that delegates to worker chats and monitors them, Goal mode for multi-step goals, scheduled tasks, chat handoff and subagent tasks.
- Added execution controls: Auto mode, an AST-based shell-command safety classifier, automatic review of risky tool calls, lifecycle hooks and permission rules.
- Built persistent agent memory across conversations: global and per-workspace stores, bounded BM25 search, agent read/write tools, a desktop memory panel, and a daily idle-time job (Dreaming) that distills durable memories from past chats.
- Built SSH workspaces that run the agent on remote hosts, with retained native sessions, a connection pool, guided host setup and diagnostics; added mobile control through a Lark bridge with queued follow-ups.

Performance and infrastructure

- Built an OpenAI-compatible LLM gateway (llm-proxy) with provider routing, per-user rate limits, usage tracking, Kong auth caching and Lark alerts on model failures; replaced LiteLLM with an in-house client. Added differential request caching: the proxy retains conversation prefixes, and clients upload only new messages, cutting per-step upload time from **about 18 s to 0.03 s** (median) and long-conversation uploads from **over 1 MB to a few KB**.
- Cut time to first token from **4.4 s to under 0.6 s** by prewarming the agent while the user types, and made switching to large conversations **about 3× faster (2.5 s to 0.8 s)** with windowed history loading and an on-disk index.
- Replaced third-party file tools with in-house hash-line read and edit tools: edits address lines by content hash instead of re-sending the text they replace, which made up **about 36%** of each edit call's output tokens. Added search backed by a bundled ripgrep and rewrote the tool descriptions, cutting model steps by **11–20%** in A/B tests.
- Packaged the runtime as a Docker-free single binary for **6 platforms** with macOS, Windows and Linux release pipelines, staged rollout and auto-update. Parallelized CI test suites, cutting the runtime test gate from **49 to about 19 minutes**, and moved builds and tests to Turborepo with parallel tasks and cached results, reducing rebuild time for unchanged code **from almost 4 minutes to about 1 s**.

Developer tools and product surfaces

- Shipped the React 19/Electron desktop client: an MCP-controlled browser, integrated terminal, sandboxed HTML artifacts with a public sharing gallery, Daily Brief, and a diagnostics agent with scripted macOS and Windows runbooks.
- Built the platform's agent SDK for headless agents in CI: an automated merge-request code review that runs after lint, tests and typecheck pass and posts deduplicated, schema-validated review threads, plus agent-driven integration tests under Bazel.
- Built a GitLab merge-request review agent (MR Hub), a Lark/Feishu bot, and a marketplace for skills, MCP servers and agents with a skill-review pipeline and admin console.
- Designed a pooled Model Context Protocol (MCP) service with health checks, automatic reconnects, failure isolation, a server registry and desktop-rendered MCP apps; built a RAG service with pgvector embeddings, hybrid vector/full-text search and incremental re-indexing.

Tools and Infrastructure Intern — *Jan 2025 – Jun 2025*

- Migrated the autonomous-driving perception codebase's lint, type-check and test CI to Bazel (**700+ commits**) with test sharding, remote execution, caching and daily coverage; owned it through GPU-cluster migrations and extended it to the robotics, compiler and runtime repos.

- Forked and customized Aspect's rules_py for Bazel (environment-loading hooks, selective py_test runs, select() in dependencies), cutting integration-test runs from **a couple of hours to under 20 minutes** and freeing GPU capacity for other workloads.
- Built an LLM unit-test generator with an LLM-graded evaluation tool and used it to generate unit tests for the perception codebase; built a CI and coverage dashboard app (Code Quality Hub), whose multi-project merge-request view became the review agent (MR Hub).
- Built an Agent-to-Agent (A2A) multi-agent framework with web, CLI and VS Code interfaces and MLflow tracing; shipped agents for codebase questions, pipeline debugging and database queries.

Siemens EDA Hyderabad, India
Software Engineer II *Aug 2020 – Aug 2023*

- Led full-stack development of Capital Teamwork and Xcelerator Start from concept to beta for enterprise customers.
- Built a real-time infinite-canvas collaboration tool and a cross-platform Electron/Java launcher connecting web UIs to local EDA tools.
- Unified Mendix, React and Angular components in a shared UI library, set up GitLab CI/CD with Jest, Puppeteer and Selenium, and mentored junior engineers.

Being Zero, Software Engineer · Hyderabad, India Mar 2020 – Jul 2020

- Built a MEAN-stack coding-contest platform with an in-browser Monaco editor and external submissions.

TECHNICAL SKILLS

Engineering: System design, distributed systems, microservices, API design, developer productivity, performance optimization, observability, cross-platform desktop (macOS, Windows, Linux)

AI and agents: Agent harnesses, agentic AI, LLM applications, LLMOps, multi-agent orchestration, agent memory, tool calling, prompt and context engineering, Model Context Protocol (MCP), ACP, A2A, Agent Skills, agent SDKs, RAG, vector search, LLM gateways, evals (MLflow), LangChain, PyTorch

Languages: Python, TypeScript, JavaScript, Java, C++

Backend and data: FastAPI, asyncio, Node.js, NestJS, REST APIs, WebSockets, server-sent events, PostgreSQL, pgvector, MongoDB, Redis, SQLite, Kong

Frontend and clients: React, Next.js, Electron, VS Code extensions, CLI tools, Angular, Tailwind

Infrastructure and testing: Docker, Kubernetes, GCP, Linux, Bazel (rules_py), Turborepo, CI/CD (GitLab), Git, SSH, PostHog, Pytest, Playwright, Jest

PROJECTS

Chronoscribe (2024): real-time lecture transcription and RAG summarization microservices (FastAPI, Next.js, Redis, Pinecone, server-sent events); presented at UB CSE Demo Days.

ChessVision (2024): YOLO/DETR fine-tuned on a custom dataset with OpenCV homography and Stockfish to read chess positions and generate live LLM commentary.

EDUCATION

University at Buffalo, M.S. Computer Science Aug 2023 – Jan 2025

CMR College of Engineering & Technology, B.Tech. Computer Science (CGPA 8.8/10) Jun 2017 – Jun 2021

ACM ICPC Regionals qualifier (2020) · 2nd prize, Hyderabad's Best Coder (2019) · 5-star CodeChef (2066 rating)