

Xiangzhe Zeng

xzos.net | github.com/xz-dev | xz@xzos.net | [+86 15530859511](tel:+8615530859511)

Senior Software Engineer (SDE) | 9 Years Technical Development & Open Source Experience | Python, TypeScript, Rust, Go, C/C++
Distributed Systems | High-Concurrency Architecture | 300x Performance Optimization | MCP Protocol Core Contributor | 120,000+ Stars Impact

Professional Summary: 9 years of software development experience with a technology stack spanning from kernel drivers to AI Agent infrastructure. Proficient in Rust and C/C++ systems programming with deep understanding of OS internals (Gentoo Linux maintainer; independently fixed hibernate/resume BSOD and implemented 8K+ resolution support for Red Hat virtio-win), demonstrating strong capability in exploring low-level technologies and quickly adapting to large C/C++ codebases. As a core developer of AI Agent projects, deeply involved in the MCP official open-source ecosystem (77k+ Stars), with hands-on experience in Agent Runtime, Function Calling, and Tool Use orchestration. Skilled in building application-layer protocols and high-concurrency architectures using Python/Go/TypeScript, and led the design of an OpenAPI-to-MCP-Tool conversion engine that achieved a 300x performance improvement.

Professional Skills

- AI Infrastructure & Protocols: Model Context Protocol (MCP) core development, **Agent Runtime, Function Calling / Tool Use orchestration**, OpenAPI/Swagger parsing and conversion engine (Strata), AI Agent toolchain integration, LLM context management, BM25+ vector retrieval algorithms
- Cloud Native & DevOps: Kubernetes (Helm, KEDA), Docker multi-stage builds, GitHub Actions (CI/CD), Google Cloud Platform (Cloud Run), Azure Serverless, Private/hybrid cloud deployment architecture, Zero-downtime deployment strategies
- Backend Architecture: Python (FastAPI, asyncio), TypeScript (Node.js/Express), Go, OAuth2.0/OIDC unified authentication gateway design, gRPC, Redis caching strategies, Microservice architecture, PostgreSQL, **Distributed Systems, High-Concurrency Programming**, JSON-RPC, WebSocket
- Systems Programming & Low-level: Rust, C/C++, GNU/Linux distribution building (Gentoo), Kernel module debugging, WebAssembly, Embedded development, cgroup/namespace, **Inter-Process Communication (IPC)**, Cross-process File Locking, **Performance Tuning**
- Frontend Technology: Next.js (SSR), React, Astro, Material Design
- Programming Languages: Python3, TypeScript, Rust, Go, C#, Kotlin, C/C++, PowerShell, Bash Script, Java, SQL, JavaScript
- Development Tools: Emacs, (Neo)Vim
- Certifications: RISC-V Foundational Associate (RVFA), Red Hat Certified Engineer (RHCE) - 285/300, Red Hat Certified System Administrator (RHCSA) - Perfect Score

Core Open Source Contributions

200+ PRs | 85% Merge Rate | Across 100+ Projects | 55+ Technical Blog Posts

- AI Infrastructure & MCP Ecosystem:
 1. **MCP Official Servers (77k+★)**: Implemented cross-process file locking mechanism in [PR#3286](#), solving data corruption issues caused by concurrent writes from multiple clients. Verified with zero data loss in 10k+ concurrent write tests
 2. Klavis AI Code Owner: Contributed 1.65M+ lines of code. Developed application-layer logic for MCP tools (Function Calling, Tool Use) using Python and Go. Led the Open-Strata open source project, contributing the enterprise-grade OpenAPI-to-MCP-Tool conversion engine back to the community
 3. **SillyTavern** Function Calling: Re-implemented the Function Calling subsystem by referencing LibreChat's architecture, replacing the legacy text-prompt-based approach with a standards-compliant Function Calling implementation that correctly handles tool definitions, invocations, and response routing
- Personal Open Source Projects:

1. [UpgradeAll \(1.3k+★\)](#): Led a 6-person team for 7 years of continuous development. Designed Kotlin(Android) + Rust(Core) cross-language architecture; led major 2026 modernization: AGP 9.0 + Gradle 9 upgrade, designed Kotlin ↔ Rust WebSocket JSON-RPC cross-language communication architecture. Reduced user update time from 30 minutes to 2 minutes
- Low-level Systems & Community Contributions:
 1. [virtio-win](#) (Red Hat, Windows Paravirtualization Drivers):
 - viogpu Driver: Demonstrated the ability to quickly understand large low-level C/C++ codebases from scratch and pinpoint complex kernel-level bugs. 6 merged core PRs. Fixed hibernate/resume BSOD ([PR#732](#)), passed 100-cycle zero-failure test; Implemented resolution overflow protection ([PR#1474](#)); Designed multi-block contiguous memory allocation with indirect descriptors to overcome memory fragmentation, enabling 8K/24K+ ultra-high resolution support ([PR#1536](#)); Implemented HDR10/Dolby Vision support ([PR#735](#))
 - Installer: Fixed upgrade error 1603 ([PR#85](#)), fixed GUI maintenance mode ([PR#87](#))
 2. [distrobox \(12k+★\)](#): Fixed rootless container cgroup v2 delegation issues, enhanced PID namespace isolation ([PR#985](#))
 3. [ansible-runner](#) (Red Hat): Refactored TTY detection logic ([PR#1306](#)), resolving pexpect hang issues in containerized environments
 4. Gentoo Linux: Maintainer, 90+ total PRs, maintaining 7 software categories. Contributed critical patches for ZFS, kde-plasma. CachyOS optimized kernel maintainer (6.6 LTS - 6.19)
 5. GradleAndroidRustPlugin: Submitted 4 PRs improving Android-Rust cross-compilation build system, fixed ABI matching logic ([PR#11](#)), adapted for Gradle 9 ([PR#9/#10](#)), completed AGP 9.0 DSL migration ([PR#14](#) merged)
 - Selected for Linux Foundation Talent Incentive Program
 - Experienced in asynchronous cross-timezone collaboration
 - Technical Community Impact: Technical blog [xzos.net](#) with 55+ original articles; [Arch Wiki](#) 29 edits + [Gentoo Wiki](#) 35 edits

Work Experience

Klavis AI - Full Stack Engineer / MCP Platform Architect

2025.7 - 2026.01

As a core founding engineer, led the infrastructure development for the AI Agent integration platform. Responsible for full-stack architecture design from low-level protocol implementation (MCP) to the upper-layer OpenAPI conversion engine (Strata), solving the standardization challenges of AI model interoperability with 20+ external SaaS services.

Core Architecture & Platform Development:

- Led Strata OpenAPI Engine Development: Designed the core parsing engine capable of automatically converting any OpenAPI v2/v3 specification into standardized MCP Tool definitions for Agent Runtimes with Function Calling support
- Extreme Performance Optimization: Through OpenAPI object pre-build caching and jsonref deep parsing optimization, reduced complex specification parsing time from 15 minutes (single-threaded) to 3 seconds, a 300x performance improvement
- Production Value: Completely resolved Google Cloud Run performance bottlenecks during Serverless cold starts and high-concurrency scaling, achieving zero-downtime deployment
- Enterprise Agent Runtime Architecture: Designed process isolation model ensuring high-risk tools like Playwright run in independent sandboxes; Optimized IPC mechanism for Tool Use orchestration, reducing tool synchronization latency from 200ms to 5ms

DevOps & Private Cloud Delivery:

- Built On-Premise Deployment Architecture from Scratch: Designed and implemented complete private cloud delivery solution based on Kubernetes + Helm, implementing on-demand loading and auto-hibernation mechanisms for MCP services
- Elastic Scaling: Integrated KEDA HTTP Add-on, implementing Pod auto-scaling based on real-time HTTP traffic
- Full-Stack CI/CD Pipeline: Built GitHub Actions automation system, implementing multi-architecture (amd64/arm64) Docker image builds, security scanning (Trivy), and automatic distribution

Microsoft - Office365 Interoperability and Compliance Project - Full Stack Engineer

2023.10 - 2025.7

Responsible for toolchain development and compliance engineering for the Office document ecosystem, led cloud-native transformation of internal systems and automated testing infrastructure development. Built AI-assisted developer tooling: achieved automated testing coverage for 400 Microsoft Graph API endpoints in 20 days by integrating GitHub Copilot with MCP Function Calling to auto-generate test cases and validate API responses.

Talkweb Information - China Unicom A-Share Portal and Work Order System - JAVA Backend Developer 2022.6 - 2023.9

Responsible for the microservice refactoring of the A-Share portal backend, implemented automated deployment and elastic scaling based on Spring Boot + Kubernetes, solving maintenance and scalability issues of the legacy system. Responsible for the backend development and maintenance of the Unicom work order system website, implementing a file storage system based on the IPFS distributed protocol.

Project Experience

Open Source Project - UpgradeAll Full Stack Application Updater - Project Initiator

2019.4 - Present

Project Links: [UpgradeAll](#) (Kotlin/Rust client), [Server](#) (Python server, 2020.3-2022.6.5)

Led a six-person team in collaborative development of the free open-source software UpgradeAll, solving the fragmentation problem in traditional software updates. Simplified the update search process for Android applications. The client has received 1.3k+ Stars.

Client Highlights (Kotlin+Rust):

- Developed frontend with Kotlin, implementing Material Design interface; Developed high-performance backend library using Rust with modular design
- 2026 Modernization Refactor: Upgraded to AGP 9.0 + Gradle 9, designed unified communication architecture based on WebSocket JSON-RPC, significantly improving cross-language call efficiency and stability

Server Highlights (Python):

- Provided client gRPC and REST interfaces, supporting application updates from multiple sources; Used ZeroMQ for microservice architecture and service discovery; Used Redis for distributed data caching, deployed services using Docker

Project Outcomes:

- Significantly reduced user application update time from 30 minutes to 2 minutes, automating what was previously a manual search process; Built a one-stop application update platform

Microsoft - Office Word Automated Testing Platform - Python Development

2024.3 - 2025.7

Project Background: Office Word requires extensive character compatibility and rendering tests to ensure product quality. Traditional manual testing methods are inefficient. I developed an automated testing platform to execute hundreds of test cases weekly.

Project Contributions:

- Designed and implemented a Python automation testing framework for Office Word compatibility verification; Adopted "code as configuration" architecture, enabling efficient test rule definition via AI assistance
- Integrated Windows API to simulate user operations for end-to-end testing; Developed OCR-based rendering recognition system to improve accuracy
- Implemented batch test scheduling in virtual machine environments to support large-scale parallel execution

Project Outcomes:

- Achieved automated execution of 800+ test cases weekly, significantly improving test coverage. Reduced manual testing time by approximately 50%
- Improved product quality through systematic testing, reducing user feedback rates for character rendering issues

Microsoft - GB Document Verification and Correction Tool - Python Development

2024.6 - 2025.7

Project Background: Microsoft Office suite needs to pass GB certification requiring standardized test documents. Issues like inconsistent terminology often occur when merging documents. I developed an automated tool to verify and correct documents.

Project Contributions:

- Deeply researched Word Open XML structure; Implemented cut, copy, and paste functionalities not supported by python-docx
- Utilized LLM for intelligent replacement of professional terminology, ensuring consistency
- Used python-docx and oxml libraries for low-level OXML operations, solving formatting errors in merged documents

Project Outcomes:

- Reduced document verification/correction time by 96.67%, from 2.5 hours to 5 minutes per person
- Ensured terminology consistency and format standardization, improving certification pass rates; Reduced formatting workload for testers
- Wrote technical blog "[Cut and move Runs via python-docx](#)", contributing solutions to the community

Microsoft - Gendox Document Management System - C# Development 2023.10 - 2025.7

Project Background: Gendox is Microsoft's internal document management tool (Word plugin) that converts docs to wiki. It supports cross-document sharing and version control.

Project Contributions:

- Developed Python/OCR-based automated testing tools for plugin efficiency research
- Built automatic publishing system on Azure Pipeline; Designed Azure Function archiving solution with PowerBI telemetry integration
- Upgraded security model to Azure Managed Identity; Developed Azure Serverless tools for automatic VM Patch Tuesday updates

Project Outcomes:

- Efficiently processed 30GB/400k files weekly; Saved approx. 3 person-days per release by simplifying collaboration
- Implemented automatic log checking, eliminating oversight risks; Automated monthly maintenance saving 1 person-day/month
- Completed Microsoft Q3 security requirements, enhancing overall system security

Microsoft - Interop Department Data Synchronization System - C# Development 2023.10 - 2025.7

Project Background: Core tool for Office Interop department for data warehouse synchronization and training management. Old system was slow and outdated.

Project Contributions:

- Replaced 12-hour Task Scheduler job with PowerShell-based fragmented execution and dependency scripts
- Integrated CodeQL into Azure Pipeline for automated security detection; Led microservice transformation from monolithic .NET Framework to .NET on Azure Container

Project Outcomes:

- Reduced Azure costs by 85%, runtime by 50%, improved stability
- Met latest security compliance; Microservice architecture eliminated VM maintenance risks; Achieved zero-downtime upgrades via gray migration

Tsinghua University Laboratory Project - IPFS-based File Sharing Application - Android Client Development 2021.4 - 2021.5

Developed Android P2P file sharing app based on IPFS protocol. Implemented Wi-Fi Direct short-distance transfer (1GB/s, 3-5x faster than cloud) and end-to-end encryption, solving bandwidth limitations and server dependency issues. Supervisor: Li Zhao

Education

North China University of Science and Technology - Computer Science and Technology - Bachelor's Degree 2023.6

Courses: Network Principles, Computer Principles, Software Engineering, Algorithm Design and Analysis, Object-oriented Programming, Database Principles, Operating Systems (Teaching Assistant)

Awards & Honors

ASC18 World University Supercomputer Competition - Second Prize