Xiangzhe Zeng

xzos.net | github.com/xz-dev | xz@xzos.net | +86 15530859511

Professional Skills

- Cloud Native & DevOps: Azure (Container, Functions, Serverless), Kubernetes (Helm, KEDA), Docker multi-stage builds, GitHub Actions (CI/CD), Google Cloud Platform (Cloud Run), Private/hybrid cloud deployment architecture, Zero-downtime deployment strategies
- Backend Architecture: Microservice architecture, Distributed systems design, OAuth2.0/OIDC unified authentication gateway, Python (FastAPI, asyncio), TypeScript (Node.js/Express), gRPC, Redis caching strategies, PostgreSQL
- AI Infrastructure & Protocols: Model Context Protocol (MCP) core development, OpenAPI/Swagger parsing and conversion engine (Strata), AI Agent toolchain integration, LLM context management, BM25+ vector retrieval algorithms
- Systems Programming & Low-level: Rust, C/C++, GNU/Linux distribution building (Gentoo), Kernel module debugging, WebAssembly, Embedded development
- Frontend Technology: Next.js (SSR), React, Astro, Material Design
- Programming Languages: Kotlin, Java, Python3, C#, Rust, C, PowerShell, Bash Script, JavaScript, TypeScript, SQL
- 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

Open Source Contributions and Key Skills

- Klavis AI Open Source Ecosystem (AI Agent & MCP Protocol):
  1. Code Owner: 349 commits, contributed 1.65M+ lines of code. Led the Open-Strata open source project, contributing the enterprise-grade OpenAPI conversion engine back to the community
  2. PR#788: Implemented process-isolated Playwright MCP Server, solving security and resource isolation issues for multi-tenant browser automation
  3. PR#833 / PR#836: Extended MCP ecosystem, integrating OAuth authentication and tool definitions for 10+ services including Azure AD, Sentry, HuggingFace
- Personal Open Source Projects:
  1. UpgradeAll (1k+★): Cross-platform application update manager. Built with Kotlin (Android) + Rust (Core) architecture, designed modular update source plugin system
  2. distrobox-boost / numlockw: Rust container runtime optimization tool and Wayland protocol low-level input control tool
- Low-level Systems & Community Contributions:
  1. Gentoo Linux: Maintainer, contributed patches for sys-fs/zfs (kernel experimental features PR#44885), kde-plasma, and other critical packages
  2. CachyOS-kernels: Maintainer of Gentoo overlay for CachyOS optimized kernels, actively tracking upstream kernel releases and maintaining patch compatibility
  3. virtio-win (Windows Paravirtualization Drivers):
     - viogpu Driver: Implemented dynamic framebuffer resizing with indirect descriptor support for 8K+ resolutions (PR#1479), fixed BSOD crash in error path (PR#1473), and resource leak on init failure (PR#1475)
     - Installer: Fixed driver upgrade failure when drivers are in use, preventing error 1603 during system upgrades (PR#85)
  4. GradleAndroidRustPlugin: Resolved Rust/Android cross-compilation ABI compatibility and Gradle 9 upgrade adaptation (PR#10, PR#11)
  5. InfiniTime: Contributed CMake build fixes and Chinese support for smartwatch firmware (PR#2143)
- Selected for Linux Foundation Talent Incentive Program, adaptable to cross-timezone collaboration

Power by LaTeX and Emacs. Written by Xiangzhe and AI

Work Experience

Klavis AI - Full Stack Engineer / MCP Platform Architect 2025.7 - 2025.12

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
- 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 MCP Runtime Architecture: Designed process isolation model ensuring high-risk tools like Playwright run in independent sandboxes; built unified OAuth authentication gateway supporting 10+ platforms including Microsoft Entra (Azure AD), GitHub, PayPal, Sentry

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 (Vendor) - 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.

- Cloud-Native Architecture Transformation: Decoupled legacy .NET Framework monolithic applications into microservice architecture based on Azure Container and Azure Functions, replaced traditional credentials with Azure Managed Identity, reducing resource costs by 85% and eliminating security compliance risks
- Automated Testing Platform: Designed ”code as configuration” Python automation testing framework, combined OCR visual recognition technology and GPT models, implemented end-to-end automated testing for Office Word character rendering compatibility, reducing manual testing workload by 80%
- Document Engineering Intelligence: Deep research into Office Open XML (OOXML) underlying structure, developed LLM-based document terminology consistency verification tool, improving GB certification document correction efficiency by 96% (from 2.5 hours to 5 minutes)

Talkweb Information - China Unicom A-Share Portal Website - JAVA Backend Developer 2022.2 - 2022.7

The Unicom A-Share portal website, due to years of neglect and outdated technology, was difficult to deploy and maintain. Responsible for the microservice design and code rewrite of the project backend, implementing k8s pipeline deployment and automated scaling.

- Restructured the project using microservice architecture
- Adopted agile development methods
- Implemented dynamically scalable k8s deployment

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 (including uninstalled apps), Magisk modules, etc. The project aims to provide a fast and user-friendly application update experience. The client has received 1k+ Stars.

Client Highlights (Kotlin+Rust):

- Developed frontend with Kotlin, implementing Material Design interface and related components
- Developed high-performance backend library using Rust, with modular code design; the kernel can be used independently
- Implemented highly customizable settings, supporting update sources configuration via Json. Embedded JavaScript engine enabling application hot update capability

Server Highlights (Python):

- Provided client gRPC and REST interfaces, supporting application updates from multiple sources including GitHub, GitLab, F-Droid, Play Store, etc.
- Used ZeroMQ to implement microservice architecture and service discovery, designed a horizontally scalable multi-layer cache service architecture
- Used Redis to implement distributed data caching, deployed services using Docker containerization technology

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 integrating multiple update sources

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 and consume significant resources from the testing team. To address this issue, I was responsible for developing an automated testing platform to execute hundreds of test cases weekly.

Project Contributions:

- Designed and implemented a Python automation testing framework specifically for Office Word software compatibility verification
- Innovatively adopted a ”code as configuration” test system architecture, allowing testers to efficiently define test rules through AI assistance and IDE environment
- Integrated Windows API interfaces to simulate user operations, building end-to-end automated testing processes
- Developed an OCR-based rendering result recognition system to improve the accuracy of character compatibility testing
- Implemented batch test scheduling in virtual machine environments to support large-scale parallel test 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 result documents from the testing team. When merging documents, issues such as inconsistent terminology and format disruption often occur. To solve these problems, I developed an automated tool to verify and correct documents, ensuring they meet standard requirements.

Project Contributions:

- Thoroughly researched the internal structure of Word document Open XML, mastering document format control mechanisms
- Implemented cut, copy, and paste functionalities not supported by python-docx
- Utilized LLM for intelligent replacement of professional terminology, ensuring document terminology consistency
- Used python-docx and oxml libraries for OXML low-level operations, solving formatting errors that occurred after merging multiple documents

Project Outcomes:

- Reduced document verification and correction time by 96.67%, from an average of 2.5 hours to 5 minutes per person
- Ensured terminology consistency and format standardization for all GB certification documents, improving certification pass rates
- Reduced the testing team's workload on document formatting adjustments, allowing them to focus on the test content itself
- Wrote a technical blog "Cut and move Runs via python-docx", contributing solutions to the open source community

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

Project Background: Gendox is Microsoft's internal document management tool, which automatically converts to wiki in the form of a Word plugin. It provides product managers with a structured document writing environment, adopting the "build menu first, then fill in content" method, supporting cross-document sharing and synchronous modification of document fragments. The project covers the entire process from editing to publishing, integrating core functions such as version control, automated building, and security assurance.

Project Contributions:

- Researched GenDox plugin loading and running efficiency, developed Python-based and image recognition-based automated testing tools
- Built an automatic publishing system based on Azure Pipeline, achieving continuous delivery of new versions. Refactored the Release tool
- Designed an Azure Function automatic archiving solution, integrated PowerBI automated telemetry data collection to generate real-time dashboards and email alert systems
- Upgraded the security model, migrating from password-based authentication to Azure Managed Identity, and wrote standardized migration documentation
- Developed Azure Serverless-based automation tools to automatically apply Patch Tuesday updates for VMs

Project Outcomes:

- Efficiently processed up to 30GB and 400,000 files weekly for releases
- Simplified team collaboration processes, reduced new version testing time, eliminated cross-team manual handovers, saving approximately 3 person-days of work per release
- Implemented automatic log checking, eliminating oversight risks from manual reviews, reducing potential delays and losses
- Automated monthly maintenance work, saving 1 person-day per month for system inspection and updates
- Completed Microsoft Q3 quarter security requirements, enhancing overall system security

Microsoft - Interop Department Data Synchronization and Training Management System Improvement - C# Development 2023.10 - 2025.7

Project Background: This system serves as a core tool for the Office Interop department, undertaking two key functions: cross-project personnel and document data warehouse synchronization, and employee training management. The old system had issues such as slow operation, outdated technology, and security architecture.

Project Contributions:

- The original Task Scheduler-based fixed-time execution mode ran for over 12 hours daily; implemented service dependency scripts and fragmented execution with PowerShell

- Upgraded the project security architecture to meet the latest security standards, integrated CodeQL into Azure Pipeline to achieve automated detection and continuous integration of code security
- Led the project's microservice transformation, decoupling the monolithic application into independent service components, implementing migration from .NET Framework to .NET through Azure Container technology

Project Outcomes:

- Reduced Azure resource costs by 85%, decreased scheduling system runtime by 50%, improved throughput and stability
- Fully met Microsoft's latest security compliance requirements; microservice architecture completely eliminated VM maintenance-related security risks
- The microservice gray migration solution achieved zero-downtime system upgrades, ensuring a continuously smooth user experience

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

Project Background: Traditional file transfer methods face bandwidth limitations and high server dependency. This project aimed to leverage the distributed nature of IPFS (InterPlanetary File System) to build a file sharing application that supports both face-to-face high-speed transmission and stable long-distance sharing.

Supervisor: Li Zhao

Project Contributions:

- Designed and developed Android client prototype, implementing core functionalities and user interface
- Integrated IPFS protocol, building an efficient P2P file transfer network
- Implemented Wi-Fi Direct-based face-to-face transfer functionality, significantly improving short-distance transfer speed
- Developed end-to-end encryption system, ensuring file transfer security
- Designed intuitive file preview interface, optimizing user experience

Project Outcomes:

- Short-distance transfer speed reached 3-5 times that of traditional cloud storage solutions, achieving 1GB/S
- Successfully implemented a P2P file sharing system without relying on central servers, improving performance and stability
- Served as the core implementation part of a graduate thesis, highly praised by the supervisor

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