

## Jianxin Qiu

jianxin.qiu@outlook.com · github.com/imtsuki

<b>Education</b>	<b>University of Toronto</b> <i>Master of Engineering in Computer Engineering</i>	Toronto, Canada 2022 – 2023
	<b>Beijing University of Posts and Telecommunications</b> <i>Bachelor of Engineering in Computer Science</i>	Beijing, China 2017 – 2021

<b>Employment History</b>	<b>Software Engineer, TikTok</b> 2023 – Present	
	Launched the Model Context Protocol (MCP) gateway, where teams can register REST/gRPC APIs as MCP tools, host third-party servers, or implement custom handlers programmatically, abstracting away protocol complexity, authentication, and observability concerns. Scaled the platform to host 9,000+ MCP servers and process 2B+ requests daily, serving as the foundational infrastructure for AI tool integration across all engineering teams.	
	Engineered the AI Sandbox platform on the serverless infrastructure to enable on-demand, ephemeral environments for AI agents with configurable lifetimes and per-instance routing. Built a runtime daemon handling remote command execution, process lifecycle management, port forwarding, and browser automation. Scaled to serve 40,000+ cores daily.	
	Maintained the serverless platform, including its control plane, build system, and data plane components. Unified on-prem and cloud control planes into a single codebase and migrated metadata for 4,000+ functions, reducing maintenance overhead and accelerating new feature delivery. Engineered CronJob-on-Serverless to enable existing batch workloads to run serverlessly, solving challenges around custom container images, distributed scheduling across multiple data centers, and seamless integration with existing Kubernetes workloads.	

<b>Past Internships</b>	<b>ByteDance</b> · Worked on Lark Messenger's cross-platform Rust library. Introduced new product features to the calendar module, improved codebase modularity and binary footprint through systematic assembly-level profiling and code refactoring.
	<b>Alibaba Cloud</b> · Worked on the ClickHouse database team. Integrated ClickHouse and Apache Flink by developing a high-throughput database connector, achieved over 5x higher throughput over the default JDBC connector.
	<b>SmartX</b> · Worked on the distributed block storage team. Enhanced the long task execution module, implemented features like storage backup parallelization, QoS bandwidth limiting, and task status management.

<b>Skills</b>	<ul style="list-style-type: none"><li>• C, C++, Rust, Go, Python, Compiler, Linux Kernel, Serverless, Model Context Protocol (MCP), etc.</li><li>• Open-source contributions: contributed to @rust-lang, @rust-analyzer, @rust-osdev, etc.</li></ul>
---------------	--