

John Kelliher

Senior Software Engineer

jackwkelliher@gmail.com | GitHub: jack-work | john.kelliher.info

PROFESSIONAL SUMMARY

Senior Software Engineer at Microsoft building distributed infrastructure for agentic development platforms and real-time collaborative state systems at global scale. Currently making GitHub Cowork a coding agent on Microsoft's first-party managed application platform, focused on sandboxing and isolation boundaries for untrusted agent-authored code. End-to-end engineer spanning backend architecture, cloud deployment, observability, container orchestration, and Linux systems development, equally at home designing persistent data structures, wiring deployments, or fighting livesite fires with panache. Compiler design background. Deployed across 30+ regions including sovereign and air-gapped clouds, serving products within the Microsoft 365 Copilot ecosystem (100M+ MAU) and Power Platform (56M+ MAU).

PROFESSIONAL EXPERIENCE

Senior Software Engineer | *Microsoft, Power Platform & M365 Copilot*

July 2019 – Present

Local-First Replicated Coding Agent Infrastructure

Fall 2025 – Present

- Architected the backend of an event-sourced, local-first replicated state service on the virtual actor model (Orleans), serving as the state kernel for Power Apps and M365 Copilot products, enabling real-time collaborative editing for both human developers and autonomous agents
- Porting this infrastructure into **GitHub Cowork** to make it a coding agent, integrating with **Managed Apps**, Microsoft's first-party managed application platform
- Engineered **container sandboxing** for untrusted agent-authored code: Linux namespace and cgroup isolation, controlled network egress, and systemd-supervised agent processes with on-container and off-container execution
- Contributed **Rust** middleware to the platform's service layer
- Designed a persistent immutable trie with structural sharing for versioned state, synchronized via server-authoritative optimistic concurrency over WebSocket and HTTP; client replicas merge structural and text patches with server-arbitrated conflict resolution
- Engineered a write-ahead log with segmented compaction and snapshot-based watermarking for recovery, backed by hot-tier Redis and tiered object storage; designed log compaction with consistency reconciliation through scalable job systems
- Built FUSE virtual filesystem with write-through local caching and debounced network propagation, giving agents a native filesystem interface to replicated state; materializes Git-compatible change history via a virtual .git
- Delivered first-party client libraries (TypeScript, C#) and React shell implementing local-first replication with optimistic writes and rollback; code-generated schema bindings for filesystem-native (agentic) and Power Apps state models
- Deployed globally across 30+ regions and sovereign clouds using Bicep, ARM, and Ev2 rollouts orchestrated via Azure DevOps; built observability on Geneva and Grafana with time-series telemetry, distributed tracing, and alerting under Microsoft safe deployment practices

Compiler & Platform Engineering

2019 – 2025

- Co-led three-person senior team to rewrite the Power Apps formula compiler, achieving 500% throughput improvement and orders-of-magnitude gains on pathological inputs
- Secured enterprise funding for an experimental AI-powered development platform with sandboxed execution; built novel LLM streaming framework that evolved into current agentic infrastructure
- Architected telemetry platform on OpenTelemetry and built distributed conflict-free data structures for real-time collaboration
- Designed authentication middleware enabling entry into compliance-heavy enterprise markets while reducing operational costs

Software Engineering Intern | *Microsoft*

Summer 2018

- Built interactive debugging system for Power Apps that became the production-standard development tool

Software Intern | *Hankook Tire*

2016 – 2017

- Built internal systems, employee management platforms, and the public website

INDEPENDENT WORK

Figaro | *coding-agent harness, Go*

2026 – Present

- Author and daily user of a coding-agent harness: one static Go binary in three roles (CLI, supervisor daemon, and a transient agent per conversation) over an append-only log with forkable conversation history
- Profiled and tuned for a low memory footprint: peak retained transcript memory cut 3× (2.4 MB → 816 KB) via viewport-windowed paging, with allocation-free hot paths and a fixed-cost render loop
- Operate a self-hosted NixOS service platform behind Cloudflare tunnels, Caddy, and Authelia 2FA with LDAP-backed authorization and sops-encrypted secrets; declarative, reproducible, deployed with automated rollback

EDUCATION

Master of Science in Computer Science, Specialization in AI Systems

Fall 2025 – Present

Georgia Institute of Technology

Bachelor of Science in Computer Science and Engineering

2015 – 2019

The Ohio State University

TECHNICAL PROFICIENCIES

Programming: C# and .NET, Go, TypeScript, Python, Rust, React

Containers: Docker, Kubernetes, systemd, Service Fabric

Networking: eBPF and Cilium, service mesh, network policy, WebSocket/HTTP

Observability: OpenTelemetry, Geneva, Grafana, KQL, distributed tracing

Cloud & IaC: Azure, Bicep, ARM, Terraform, Nix, Azure DevOps

Security & Isolation: Container sandboxing, Linux namespaces and cgroups, egress control, OIDC/2FA, LDAP

Architecture: Event sourcing, actor model, local-first replication, microservices, compiler design

Specialized: Agentic systems and MCP, FUSE, Redis, Linux/Unix systems