

GABRIEL WINDLIN

☎ +41 79 866 7358 ✉ gawindlin@gmail.com  [gabriel-windlin](https://www.linkedin.com/in/gabriel-windlin)  [0xShred](https://github.com/0xShred)  gawindlin.com

Education

ZHAW – Zurich University of Applied Sciences

Expected Graduation: Sept 2029

Bachelor of Science in Computer Science

Winterthur

- Starting Sept 2026; core curriculum includes Algorithms & Data Structures, Linear Algebra, Discrete Mathematics, and Software Engineering fundamentals.

Experience

Tokio

Jul 2026 – Present

Co-Founder & Backend Engineer (Pre-Seed / MVP)

Zürich

- Co-founded **Tokio**, an AI-powered brand-compliance platform pairing a native SwiftUI client with a custom backend, currently at MVP stage ahead of incorporation.
- Own the backend: architected a Rust GraphQL API on **Axum** and **async-graphql**, with **SQLx/PostgreSQL** persistence, JWT session auth backed by **Supabase**, and **tower-governor** rate limiting.
- Built the document-ingestion pipeline that turns uploaded brand guidelines into approved, embedded "nodes," and a **RAG**-based compliance engine that scores submitted copy against those approved nodes.
- Shipped supporting infrastructure: transactional email invites via **Resend**, workspace/role management, and an AI image-generation endpoint handling long-running (5-minute) requests.

Belimo Automation AG

Mar 2026 – Present

Software Developer

Hinwil

- Designing and building a multi-tool engineering platform from scratch: reusable Python toolkits for **iQuavis** and **Codebeamer**, a **FastAPI** backend-for-frontend exposing tool-namespaced REST routes, and a **Next.js/TypeScript** frontend, unifying internal engineering tools for the whole company behind a shared HMAC-signed session-auth layer and a background job runner for long-running operations.
- Shipped the iQuavis integration end-to-end: read-only browsing of projects/datasets/REMs/classes/diagrams, project-scoped and global fuzzy search, and a full admin write surface (user/organization/role management) gated behind a custom permission system with audit logging.
- Built the Codebeamer integration: client/resource wrappers plus write and background-job routes for stream and baseline updates, including fixes for real upstream API bugs (e.g. working-set vs. real stream ID confusion).
- Implemented a tiered authorization system (default/write/admin per tool) driven by a hot-reloading YAML config, enforced via **FastAPI** dependencies, fail-closed by design, and surfaced to the frontend so pages and navigation hide themselves for unauthorized users.
- Automated two cross-tool workflows previously done by hand (a REM-lifecycle automation, and a Codebeamer↔iQuavis diff/sync tool replacing manual CSV exports), backed by CI/CD (**GitHub Actions**, Conventional Commits, automatic semver) and a full test suite (pytest, mypy, ruff, tsc); deployed on an Ubuntu VM under **systemd**.

VIS ETHZ

Oct 2025 – Feb 2026

Computer Infrastructure Team Member

Zürich

- Deployed **Rally**, an open-source collaborative scheduling tool, on the association's local cluster using **Kubernetes**, enabling easier event coordination for the student board.
- Configured container orchestration and networking to ensure reliable, high-availability self-hosted services for the students.
- Collaborated with the infrastructure team to maintain, patch, and monitor on-premise server resources, gaining hands-on experience with bare-metal Linux administration.

PureGym / Skills Park

Mar 2022 – Present

Customer Service & Operations

Zürich Area

- Developed strong interpersonal and communication skills by serving a diverse client base in fast-paced environments, ensuring efficient daily operations and conflict resolution.

Projects

TCP Port Scanner | [Website](#) | [Codeberg](#)

C | POSIX Sockets | Non-blocking I/O

- Developed a highly efficient single-file TCP `connect()` port scanner in C, demonstrating deep understanding of **POSIX sockets** and **low-level networking**.
- Implemented **non-blocking I/O** with `select()` to manage per-port timeout control, significantly increasing scanning speed compared to blocking implementations.
- Engineered asynchronous connection handling using **EINPROGRESS** and **SO_ERROR** to correctly classify ports as open, closed, or filtered.
- Designed a minimal, signal-focused output system that emphasizes clarity and correctness without relying on any external libraries.

Homelab: Proxmox + k3s Cluster

Proxmox VE | Kubernetes (k3s) | Rocky Linux | Infrastructure as Code

- Repurposed a salvaged PC (upgraded RAM) into a **Proxmox VE** hypervisor host and rebuilt a 3-node **k3s** cluster (1 control-plane, 2 workers) on **Rocky Linux 9** VMs, replacing a prior single-node Surface-based homelab.
- Bootstrapped k3s with **Traefik** ingress, metrics-server, and local-path-provisioner; validated cluster health and LoadBalancer service exposure via **svclb-traefik** across all nodes.
- Documented the rebuild (storage layout, VM sizing, k3s bootstrap and node-join process) as versioned runbooks, ahead of publishing a sanitized Infrastructure-as-Code repo (Kubernetes manifests, Ansible, Terraform).
- Hardening in progress: **Tailscale** mesh for remote SSH access, least-privilege node auth, and a GitOps-based application deployment workflow.

SEC-SUITE: Security Toolkit | [Website](#) | [Codeberg](#)

Python | Multi-threading | Security

- Engineered a comprehensive security toolkit featuring multiple password cracking techniques including **Markov Chain** probabilistic generation, dictionary attacks, and **multi-threaded brute force**.
- Implemented modern hash algorithms (bcrypt, scrypt, argon2) with automatic type detection and salt handling.
- Developed security utilities including a **network port scanner** and encoding/decoding tools wrapped in an intuitive interactive CLI interface.

rsh: POSIX Shell | [Codeberg](#)

Rust | POSIX | Parsing

- Built a POSIX-compliant shell in **Rust** from scratch: a REPL execution loop, command tokenization with single/double-quote and backslash-escape handling, and \$PATH resolution for external executables.
- Implemented core builtins (`cd`, `echo`, `pwd`, `exit`, `type`) with dedicated error handling.
- Structured the codebase around a dedicated parser and per-builtin command modules to keep tokenization, execution, and error paths independently testable.

Packet Sniffer | [Codeberg](#)

C | Raw Sockets | pcap

- Built a raw-socket packet sniffer in C that captures live **TCP** and **DNS** traffic and writes it to a standard `.pcap` file for analysis in tools like Wireshark.
- Parsed Ethernet/IP/TCP headers directly from raw socket buffers, requiring root privileges and low-level control over the network stack.

Kiroku-TUI | [Codeberg](#)

Rust | Ratatui | Git Integration

- Developed a high-performance terminal-based journaling tool in **Rust** using the **Ratatui** library for a responsive and keyboard-centric TUI.
- Integrated **Git-based synchronization** and file-watching functionality to ensure seamless note consistency across multiple devices.
- Implemented advanced search capabilities including **fuzzy matching**, deep content search, and tag-based filtering via YAML frontmatter.
- Engineered a robust configuration system and custom theming engine to support personalized workflows and visual aesthetics.

Living Memory: E-Ink IoT Display | [Codeberg](#)

C++ (Arduino) | ESP32-S3 | Embedded Systems

- Designed and built a custom low-power e-ink display (**LilyGo T5, ESP32-S3**) as a shared, app-free communication device, with firmware written in C++ on the Arduino framework.
- Implemented a multi-partition touch UI (calendar, live weather, countdown, and a daily-rotating photo panel read from an external microSD card).
- Built a **Google Apps Script** backend to sync remote content updates onto the device over Wi-Fi without requiring a reflash.

- Engineered a modular neural network from scratch in pure C, implementing **backpropagation** and **gradient descent** without external libraries.
- Implemented multiple activation functions (ReLU, Sigmoid, Tanh) and loss functions to solve complex classification and regression problems.
- Integrated advanced training features including **momentum**, **learning rate decay**, and **L2 regularization** to improve convergence and prevent overfitting.
- Developed a diagnostic suite featuring **gradient checking** to mathematically verify the correctness of the backpropagation implementation.

Technical Skills

Languages: Java, Python, C, Rust, SQL, Bash/Shell, JavaScript, TypeScript, Lua

Infrastructure & Tools: Linux, Docker, Kubernetes (k3s), Proxmox, Git, Supabase, Caddy, CrowdSec, systemd, Neovim

Libraries/Frameworks: Spring Boot, FastAPI, Axum, async-graphql, Next.js, React, Ratatui, PostgreSQL, SQLx, Redis, Astro, Tailwind CSS

Certifications

CompTIA Security+ (SY0-701) — issued Apr 2026, valid through Apr 2029

Linux Foundation Coursework: Introduction to Zero Trust, Secure AI/ML-Driven Software Development, Cybersecurity Essentials, OpenAPI Fundamentals, Fundamentals of Quantum Computing