Vojtěch Špaček

Links: GitHub | LinkedIn | X/Twitter **Contact:** sparesparrow@protonmail.ch | +420776168749 | Schedule a call

sparesparrow - Whoami

- Pragmatic **Programmer**, Creative **Problem Solver**,
- **AI/LLM Developer** and Researcher
- **Cybersecurity** & **Linux** specialist, `

Preferred Roles

Software Developer

`(Linux, Embedded, TCP/IP, Cybersecurity, High-availability, RTOS, Communication Protocols, Backend, Data Processing, Hardware Abstraction Layers, Inter-process Communication)`

- **C++** (*Senior*)
- **C#** (*Medior*)
- **Python** (*Medior*)
- **JavaScript** (*Medior*)
- **Rust** (*Junior*)
- **Qt/QML Developer** (*Junior*)

AI/LLM Developer & Integration Engineer

- Although **C++** is my priority language to work with, I expect a significant programmer's workload transformation in the future, so I have decided to transform myself into **AI Developer**, mastering high-level programming languages (**Python, JavaScript**), Advanced Prompt Engineering (RAG, Zero-shot, Few-shot, Prompt caching, Structured Outputs, Tool Calling, encapsulating agents, Designing multimodal routing and collaborative workflows with multiple agents and the human in the loop, etc.)
- I seek positions fitting these skills I have been improving for almost 3 years (I am **all-in AI since GPT-3**)
- Working extensively with all major LLM providers APIs, contributing to open-source frameworks & SDKs, integrating them into traditional tooling and applications, focusing on scalability, modularity, cost optimization, reusability, DRY and automation. Extending traditional tooling and applications, building robust CI/CD pipelines that enable LLM-powered workflows at scale. I excel in natural language defined instructions and system prompts to achieve the best possible results from LLM agents.

Relevant Professional Experience - Overview:

- **Thermo Fisher Scientific** (2023-2024): `C++` server-side software for high-resolution detector hardware operation and image acquisition, C# WPF application for image processing and display, Python (tests), HAL/IPC (`COM/MIDL`), FPGA embedded systems, `gRPC` , image acquisition & processing.
- **Honeywell** (2022-2023): `C++` real-time communication APIs, Python (tests), ARM/RTOS embedded software, aviation standards compliance, Enterprise Architect (UML diagrams).
- **Trusted Network Solutions** (2020-2022): Kernun Adaptive Firewall, `C++` , `Linux` , `Netlink` , `nftables` , `TLS inspection` , `Perl` (tests), `JavaScript` (proxy configuration backend), VirtualBox, `GDB` .

Education

- **Masaryk University - Faculty of Informatics**
- `Computer Science, Learned programming, Linux, Networking`
- **Coursera.org, DeepLearning.ai**
- `Deep Learning Specialization` (four Machine Learning courses from Andrew Ng), Data Analysis, Prompt Engineering, Building LLM agents.
- **Design Patterns, Hack The Box**
- **Software engineering at Google, The Pragmatic Programmer, Clean Architecture**

Professional Experience - Detailed

Thermo Fisher Scientific - Ixperta Contractor

`Software Developer (C++)`: Electron Microscopy software development for high-resolution detector hardware operation and image acquisition. - Implemented server-side software in `C++`, including `hardware abstraction layers (HAL)` and `inter-process communication (IPC)` using `COM and MIDL` on Windows. - Developed embedded software for `FPGA` hardware, utilizing network communication protocols (`gRPC`) in C++, Python, and Linux environments. - Contributed to `WPF` application development on the backend using C# and `COM/MIDL` IPC with HAL, and ensured graceful degradation of the application. - Developed `vscode-automation`, an enterprise-grade VSCode extension integrating TICS code quality analysis with OpenAI API for automated violation detection and AI-powered remediation, streamlining code quality workflows for the development team using TypeScript, VSCode Extension API, and Git API integration. - Led 3 knowledge sharing sessions on `GitHub Copilot` best practices for the development team and presented bi-weekly stakeholder demos showcasing sprint deliverables via GitHub Teams, including final presentation of `vscode-automation` innovation sprint outcomes.

New Skills Learned: `Image Acquisition & Processing, C#, .NET, Component Object Model, Software Development for Windows, gRPC, FPGA development, VSCode Extension Development, TypeScript, OpenAI API Integration, Technical Presentation & Stakeholder Communication`

Honeywell

- *Project:* Onboard Maintenance System that monitors the condition of aircraft and diagnoses issues during flight.
- Architected and implemented `DiagTest framework`, a sophisticated `state machine-based diagnostic test engine` in `C++` managing 15+ operational states for aircraft member system testing during flight and maintenance operations.
- Designed `job-based execution system` with `DiagTestJob` and `DiagTestMsgQueProc` for `asynchronous test orchestration`, implementing message queue processing patterns for reliable real-time diagnostics.
- Engineered `test interference management system` preventing conflicting diagnostic tests from running simultaneously, ensuring aircraft system safety and data integrity.
- Implemented `four diagnostic test modes` (data collection, passive monitoring, interactive, non-interactive) with `timeout management`, precondition handling, and `aviation-standard command/response protocols`.
- Developed comprehensive integration tests using `Python` to validate communication protocols, state transitions, and system stability under various failure scenarios.
- Built `real-time parameter monitoring system` (`MonitorManager`) tracking aircraft system values during test execution with maintenance message correlation.

New Skills Learned: `Embedded Software, ARM, RTOS, Clang, Cross-platform Compilation, Flatbuffers (binary data serialization, C++ headers generation), Conan (Dependency management), Enterprise Architect (UML diagrams), Aviation Safety Standards, State Machine Design Patterns, TeamCity, GitLab CI/CD`

Trusted Network Solutions

- **Implemented a forward proxy** using `C++` on `Linux` , featuring `HTTP/HTTPS proxy with TLS/SSL inspection` , `OCSP integration` for certificate validation, `Kerberos authentication` , and `LDAP integration` for user/group lookup.
- Built comprehensive proxy infrastructure with `certificate generation and management` (CA, client certificates), `dynamic configuration builder/parser system` , and `manager/partition architecture` for modular daemon design.
- Integrated with security components including `Suricata (Intrusion Detection System)` and `nftables` `(Static Firewall)` , aggregating logs and managing connection information in `PostgreSQL` .
- Developed `DNS resolution and caching mechanisms` , domain categorization, and client authentication methods.
- `Developed and maintained ipmon` , a sophisticated network monitoring solution demonstrating deep expertise in:
- Implemented `real-time network interface monitoring` using `NETLINK sockets` , handling both IPv4 and IPv6 address management.
- Engineered robust `socket programming with Unix domain sockets` for inter-process communication.
- Designed and implemented integration with `nftables` for dynamic firewall rule management.
- Created `JSON-based configuration management` for network settings and proxy configurations.
- Implemented `thread-safe networking operations` using modern C++ features including mutex locks and `RAII patterns` .
- Designed `event-driven architecture` for real-time network changes with configurable delay mechanisms.
- Created `atomic file operations` for configuration management ensuring system stability.
- `Developed reporter_log` C++ replacement of Perl command, implementing `PostgreSQL database integration` , `structured log parsing and processing` , and `Perl integration` for legacy compatibility with `database schema management` .
- `Built configuration generation system` using `TypeScript/JavaScript` , implementing `YAML-based high-level configuration processing` , `certificate generation automation` , and `integration with web backend` for dynamic proxy configuration.

- `Developed comprehensive testing infrastructure` with `Perl test framework`, implementing `HTTP/HTTPS protocol testing`, `TLS/SSL inspection testing`, `Kerberos/LDAP authentication testing`, and `OCSP validation testing` for enterprise-grade quality assurance.

New Skills Learned: `TCP Proxy, HTTP Proxy, Encrypted Traffic Inspection, TLS 1.3, C++20, C++17, Boost Libraries, JSON-CPP, Kerberos/Heimdal, OCSP, LDAP, SQLite, Perl, JavaScript/TypeScript, YAML, Jenkins, Linux Networking, Netlink, UNIX Sockets, SSH Tunneling, Port Forwarding`

Self-employed

- **B2C**: *IT Support Freelance, Consumer Hardware Retail, Security Consulting.*
- **B2B**: *CI/CD Automation Pipelines, Linux Server Administration* **New Skills Learned:**: `Kali Linux, Wireshark, TeamCity, Docker. Self-studied for Certified Ethical Hacker, never took the exam.`

IBM - Automation Developer

Project: Enterprise automation solution for IT operations **Role:** Automation Developer

- Developed automation application that **replaced manual human workload** including ticketing operations and basic server administration processes.
- Implemented workflow automation using **Bash**, **PowerShell**, and **JavaScript** for cross-platform server management.
- Applied **Automata Theory** principles to design state-driven automation workflows for complex IT operations.

New Skills Learned: `Bash, PowerShell, JavaScript, Practical usage of Automata Theory`

IBM - IT Systems Specialist

Project: Enterprise IT infrastructure monitoring and incident management **Role:** IT Systems Specialist

- Monitored **Unix/Windows servers** and resolved incidents in production environments.
- Performed ticketing operations and developed **scripting solutions** for system administration tasks.
- Gained hands-on experience with **Active Directory**, **VMware** virtualization, and enterprise IT operations.

New Skills Learned: `RedHat Linux, Windows Server, Active Directory, VMware`

Open Source Contributions

MCP-Prompts *(97 ⭐| 21 forks)*:

- Comprehensive **Model Context Protocol server** for managing prompt templates and LLM interactions.
- **Four core MCP implementations:** filesystem operations, memory management, AWS integration, and generic MCP configurations.
- Features **Docker containerization** across multiple server types, **AWS deployment examples** (S3, Lambda, DynamoDB, CloudWatch, IAM, ECR), and **comprehensive documentation**.
- Supports development workflows, AWS infrastructure, data processing, and system administration through specialized prompt templates.
- *mcp-prompts-rs: Reimplementation in Rust, optimized for memory efficiency and high concurrency while maintaining full MCP protocol compatibility.* **Repositories & Marketplace:**
- GitHub
- npm package
- Docker Image
- **Community impact (MCP Server Hubs & Markets directories):** Glama.ai, Glama (Fly.dev), MCP Market, MagicSlides, MCPHub, npm, Docker Hub, Skywork, PulseMCP, AIBASE

mcp-project-orchestrator:

- **Template orchestration** with pattern recognition and code generation.
- **JSON-driven orchestration** analyzing user input to identify design patterns and create initial project files from templates.
- **Template management** with project/component templates and variable substitution, plus prompt template rendering.
- **Mermaid diagram generation** (flowcharts, sequence diagrams, class diagrams) with SVG/PNG rendering.

human-action: Audiobook generation pipeline with Elevenlabs Python SDK and voice (Python)

hard-coder:

- Demonstrates **early adoption of agentic design patterns** and multi-step reasoning in AI code generation
- **Reasoning Engine** AI Assistant with implemented **reasoning & planning capability**, designed when LLMs lacked advanced reasoning without agentic frameworks.
- Transforms ideas into working code including **tests and documentation**.

- Evolution path: **OpenAI GPT Store** → **Anthropic** migration → **CrewAI** refactoring → **Cursor IDE Composer Notepad**.

github-events: Python service monitoring 23+ GitHub event types with analytics (Python)

- Comprehensive monitoring service tracking GitHub activities with **repository health scoring**, **developer productivity analysis**, and **security anomaly detection**.
- **FastAPI REST API** with 15+ endpoints, **interactive dashboard** with live data integration, **MCP server** support.
- **Dual backend support:** SQLite and AWS DynamoDB with **ecosystem monitoring** for cross-domain cooperation.

sparetools: monorepo aggregating shared Python tools, Conan recipes, demonstrating C/C++ dependency management and cross-platform CI/CD.

cpy: Bundled CPython for sparetools with Python runtime

rtp-midi: Modular Rust architecture for real-time MIDI, audio, and LED control (Rust 66.5%, 2⭐)

- **Low-latency MIDI routing** connecting music production hardware to DAWs and addressable LED systems via network protocols.
- **Android Hub** (Kotlin + Rust NDK) with foreground service, AMidi NDK, mDNS discovery, plus **ESP32 visualizer** using FastLED and dual-core FreeRTOS.
- **Dual-protocol architecture:** RTP-MIDI for DAWs, OSC for LED devices with modern CI/CD pipeline and Docker containerization.

Podman-Desktop-Extension-MCP: combining multiple agent containers for collaboration and routing between them.

ipmon: Netlink socket monitoring daemon that detects real-time IPv4/IPv6 address changes, and updates firewall (`nftables`) rules accordingly. Built with modern C++20, applying patterns such as RAII, move semantics, thread-safe and atomic operations. Event-driven architecture, Unix domain sockets for IPC, JSON configuration management. Packaged for Debian-based systems with complete systemd hardening profiles for production deployment.

rust-network-mgr: Redesigned and reimplemented ipmon as a Rust daemon orchestrating dynamic firewall rules via Netlink with minimal memory footprint. Architected event-driven system for real-time network state changes, implementing atomic firewall updates to prevent race conditions and inconsistent rulesets during topology changes. Built with lock-free data structures and epoll-based multiplexing for high-throughput event processing. | crates.io/crates/rust-network-mgr