

Marc Pardo

Software Tech Lead • Rust • Linux • Networking • Distributed Systems • Arles, France • p13marc@gmail.com • marcpardo.eu

Profile

Software tech lead with 10+ years building Linux-based industrial systems — medical imaging, network security, and today autonomous maritime platforms. I specialize in systems software, networking, resilient communications and distributed systems, in Rust and C++. I design and build communication software for environments with unreliable RF, SATCOM, Wi-Fi and constrained links, and lead the technical development of a small team.

Core expertise

- **Languages** — Rust, modern C++; working knowledge of Python
- **Linux** — netlink, namespaces/VRFs, traffic control, nftables, systemd, eBPF; Debian, Yocto
- **Networking** — TCP/IP, routing and failover, WireGuard/VPNs, packet capture and zero-copy processing (AF_PACKET, AF_XDP), radio/SATCOM links
- **Distributed systems** — Zenoh, pub/sub and query architectures, DDS, multi-link WAN communication
- **Leadership** — technical design, roadmap, code review, mentoring

Experience

Exail — Mine Warfare division

Software Tech Lead • April 2024 - present

Exail builds autonomous maritime systems, including uncrewed mine-countermeasure solutions.

- Lead the technical design and development of the **Connectivity Solution**, the communication backbone linking every asset of a mission: surface drones (USV), aerial drones (UAV), underwater vehicles (AUV) and ground stations
- Designed the multi-transport communication architecture — RF, fiber-optic, SATCOM, Wi-Fi — with link supervision and failover across transports, and mesh routing between mobile assets
- Lead a team of 2-5 developers: design decisions, technical roadmap, code reviews and mentoring
- Built Rust-centric services on the **Zenoh** middleware, with C++ components where appropriate

Rope-access technician (cordiste)

Provence region • 2023 - 2024

A deliberate career break between software positions: industrial rope-access work, with the safety and risk-assessment discipline it demands.

Pellenc ST — Pertuis

C++ Developer • December 2022 - July 2023

Waste-sorting machines: developed camera drivers and their integration into the real-time processing pipeline; built Docker-based test and data-replay tooling that let the team develop against recorded production data.

NANO Corp — Paris

Rust Developer · July 2021 – December 2022

Network-monitoring startup; one of the first employees, working across probe development, security and infrastructure.

- Developed a **100 Gb/s userspace NIC driver** in Rust (Intel E810, VFIO) for the company's network probe
- Evaluated the packet-capture design space — `PACKET_MMAP`, `AF_XDP`, `DPDK`, custom driver — and turned the findings into the product's capture engine
- Implemented protocol detection and traffic analysis; rolled out mTLS across the product
- Carried the product through **CSPN security-certification** work: licensing, anti-reverse-engineering, security hardening
- Set up the entire company infrastructure: Proxmox, GitLab CI, Harbor registry, WireGuard, reverse proxies, backups

SuperSonic Imagine — Aix-en-Provence

C++ Developer · June 2017 – January 2021

Ultrasound imaging systems — a regulated medical product spanning kernel, imaging, UI and its own Linux distribution.

- **Introduced Rust to the team**, including a code-refactoring tool that unified logging across the decade-old C++ codebase
- Built the remote-update feature on apt and the fleet-monitoring tooling
- Implemented the DICOM **ATNA audit trail** and RFC 5424 syslog for regulatory compliance
- Prepared the Debian 9 → 10 platform migration (systemd transition, packaging); evaluated logging stacks and pub/sub-RPC middleware (gRPC, D-Bus, ZeroMQ, DDS...)

Xerox Impika — Aubagne

C++ Developer · July 2015 – October 2016

Prototyped an industrial 7-colour B1-format printer for a technology demo at Drupa: Qt UI, CUDA/OpenMP-accelerated data transformation, and migration of the in-house TCP/IP stack to boost.asio for Windows/Linux portability.

Open source

A Rust-based open-source toolkit for Linux networking, packet processing, network simulation, distributed communication and observability — developed on my self-hosted forge, mirrored to GitHub.

- **nlink** — a native async Rust API for Linux network configuration via netlink
- **nlink-lab** — reproducible Linux network labs from a typed topology DSL, no Docker needed
- **netring** — zero-copy packet I/O (`AF_PACKET`/`AF_XDP`) with a CI-enforced zero-alloc hot path
- **zensight** — distributed observability platform on Zenoh: 11 protocol sensors, one keypace

Education

CERI — Avignon University

Two-year university diploma (DEUG) in Computer Science · 2013

Lycée Montmajour — Arles

Baccalauréat S · 2011

Languages

- **French** — native
- **English** — professional working proficiency