

SUMMARY

iOS/macOS systems engineer: sandboxed system extensions, XPC and IPC across trust boundaries, low-level networking, and privacy-preserving request paths in modern Swift. Re-architected Proton VPN's packet tunnel around Swift Concurrency with a shared Rust WireGuard data path; OSS work merged to swift-foundation.

EDUCATION

- **Paris Diderot-Paris 7 University** Paris
Master's Degree in Computer Science - Programming Languages 2018

EXPERIENCE

- **Proton VPN** Paris Sentier
macOS / iOS / tvOS Engineer February 2024 - Present
Privacy-first VPN across iOS, macOS, and tvOS
 - **Concurrency & connection lifetime:** Re-architected the `NetworkExtension` packet-tunnel provider around `async/await` and `AsyncStream` — explicit cancellation, back-pressure, and lifetime management of the long-lived VPN connection; a Rust WireGuard data path
 - **Sandboxed IPC across the extension boundary:** XPC for app-to-extension commands; lighter primitives for extension-to-app and inter-extension broadcast — `notify_post` + `mmap`'d shared memory (iOS) and zero-copy `IOSurface` (macOS) — where XPC's bootstrap overhead was the wrong fit
 - **Low-level networking:** Noncopyable typestate socket API (phantom TCP/UDP, Closed/Open consuming transitions) over Darwin socket syscalls with typed errors and per-interface binding, for a `NetworkExtension` transparent-proxy split-tunnel — public sources
 - **Telemetry, profiling & security:** `os_log`/signpost tracing and connectivity diagnostics across device, network, and server tiers; `Instruments` on real hardware; cross-language (Go/Rust) leak work
- **Beam** Remote - Paris
macOS / iOS Engineer May 2022 - February 2024
WebKit-based macOS browser in SwiftUI, with note editing and on-device ML features.
 - **WebKit, ML integration & memory:** SwiftUI browser features on WebKit message handlers; studied WebKit source for private APIs (macOS PIP); integrated on-device ML (`CoreML`); fixed multiple memory leaks
- **Devialet** Paris Opéra-Paris Miromesnil
iOS Software Engineer October 2018 - May 2022
iOS app controlling speakers and accessories on the local network; Swift over a C++/Objective-C stack.
 - **Cross-language integration & lead:** Bridged Swift to a C++/Objective-C stack via `protobuf` on the wire; led the companion Gemini headset app — architecture and modular Swift Package structure

SKILLS

- **Languages:** Swift, Objective-C, C, C++, Rust
- **Concurrency & lifecycle:** Swift Concurrency (`async/await`, `AsyncStream`, typed throws, `swift-async-algorithms`), cancellation/back-pressure, lifetime management, custom `AsyncStream` primitives for sharing/processing, GCD
- **System-level & IPC:** XPC, `NetworkExtension`/system extensions, `notify_post`+`mmap`/`IOSurface`, Darwin sockets, Network, process isolation
- **Lower-level integration & tooling:** Swift to C/C++/Obj-C interop, Rust FFI, `protobuf`; `Instruments`, LLDB, Xcode, SPM
- **Interests:** OS internals (Mach-O, `dyld`), wire protocols, privacy-preserving systems, Swift Evolution, embedded systems

PROJECTS

- **swift-foundation (merged PR #1893):** Pure-Swift `Data.range(of:options:in:)` for non-Darwin platforms — Boyer-Moore search via `Span`, benchmarked on-par or better than the C implementation; part of the Data modernization at WWDC26
- **Besogne:** Swift interface over `os_log`/OSActivities — signposts and activity tracing as first-class Swift; source
- **CocoaHeads Paris talks (English slides):** LLDB, Logging, Virtual Machines (virtualization: `tart`, `QEMU`, `libkrnl`)
- **Reverse engineering & SpringBoard extensions (Cydia):** Objective-C runtime, method swizzling, Mach-O/`dyld`; SpringBoard tweaks, 300K+ downloads