Gunnar Hedlund Staff Rust Engineer

Seattle, WA · gunnar.hedlund@gmail.com · +1 206-555-0158 · linkedin.com/in/gunnarhedlund

PROFILE SUMMARY

- **Staff Rust Engineer** with **15 years** in **systems and browser engineering**, now moving **C++ parsing code to Rust** in a web browser with **millions of daily users**, after 7 years writing **C++ for network appliances**.
- Works in **Rust** and **C++** with **cbindgen**, **bindgen** and **UniFFI** at the boundary, ships to **Windows, macOS, Linux and Android**, and uses **WebAssembly sandboxing** for third-party C libraries.
- Strong in **unsafe code review**, **incremental C++ to Rust migration** and **fuzzing at scale**, and picks rewrite targets from each component's **crash and security bug history**.
- Works with **security engineering**, **release management**, the **Android and desktop teams** and **open source contributors**, and writes the RFCs on how teams mix Rust and C++.
- Co-wrote the **unsafe Rust review rules** 3 teams follow, mentors **6 engineers** moving from C++ to Rust, and reviews **cross-team designs** for the platform group.

TECHNICAL SKILLS

Languages:	Rust, C++17, C, Python, JavaScript
Async & Networking Crates:	Tokio, serde, futures, hyper
Systems & Interop:	unsafe Rust, cbindgen, bindgen, UniFFI, cxx, Rayon
Wasm & Embedded Tooling:	WebAssembly, RLBox, wasm2c, Wasmtime, WASI
Testing & Verification:	cargo test, cargo-fuzz, libFuzzer, proptest, Miri, AddressSanitizer
Profiling & Observability:	Firefox Profiler, perf, criterion, heaptrack
Build, CI & Supply Chain:	Cargo, clippy, rustfmt, cargo-vet, cargo-deny, Taskcluster

EDUCATION

KTH Royal Institute of Technology — **M.Sc. in Computer Science** — Stockholm, Sweden · Aug 2006 to Jun 2011

WORK EXPERIENCE

Mozilla — **Staff Software Engineer** — Seattle, WA (remote) · Mar 2019 to Present

- Rewrote a C++ image metadata parser in Rust with 3 other engineers, keeping its C++ interface through **cbindgen** and shipping it behind a pref for 2 releases, which retired **11,000 lines** of C++.
- Reviewed new **unsafe** blocks in the media crates for 3 years and wrote the SAFETY comment rules teams now follow, cutting unsafe blocks per 10,000 lines from 31 to **9**.
- Set up **cargo-fuzz** and libFuzzer harnesses for the new parsers in continuous fuzzing, plus **Miri** runs on the FFI layer in CI, finding **40+ bugs** before any reached a release build.
- Moved 2 third-party C libraries into **WebAssembly** sandboxes with RLBox and wasm2c, so a memory bug in either stays inside the sandbox, at a CPU cost under **3%**.
- Designed a shared **UniFFI** crate that brings Rust sync code to the Android and iOS apps, replacing separate Kotlin and Swift copies and cutting sync bug reports by **45%**.

F5 Networks — **Senior Software Engineer** — Seattle, WA · Aug 2011 to Feb 2019

- Wrote packet inspection code in C++ for BIG-IP appliances handling **40 Gbps** of inline traffic per box.
- Cut per-packet CPU cost by **35%** with cache-line aligned structures and by removing an allocation from the hot path.
- Built a **Tokio** load generator in Rust for TLS termination tests, replacing a Python tool and raising test traffic from 2 to **18 Gbps**.