Ugo Gosso

Systems Software Engineer | High-Performance & High-Assurance Systems

contact@buzzybis.com | buzzybis.com | github.com/BuzzYbis | linkedin.com/in/ugo-gosso

Summary

Systems software engineer focused on high-performance and high-assurance software. Works primarily in Rust, C++ and OCaml, with particular interests in correctness, concurrency, low latency and memory efficiency. Experienced in translating theoretical concepts into practical implementations.

Experience

Airbus Defence & Space 2024-09 — Present
Software Engineer (Apprenticeship) Toulouse, France

- Developed Rust cryptographic components integrating PKCS#11 HSMs to protect satellite ground-segment telecommand and telemetry flows.
- Designed the system for correctness and high throughput, with assurance objectives intended to support a future Common Criteria EAL4+ evaluation.
- Conducted research and development on post-quantum cryptography for next-generation satellite systems.

Projects

Custom 3D Game Engine
C++23 | Vulkan

- Architecting a Nanite-inspired virtualised-geometry engine targeting GPU-driven LOD selection, visibility processing and indirect submission with minimal CPU overhead.
- Engineering demand-paged asset streaming with asynchronous I/O, bounded memory residency and explicit latency and bandwidth budgets.
- Designing portable abstraction boundaries and explicit resource-ownership and CPU/GPU synchronisation protocols for concurrent streaming, transfer and rendering workloads.

phy-trng
Rust (no_std) | Creusot | Verus | seL4

- Developing a physical entropy-source/TRNG pipeline in Rust, using Creusot to verify functional invariants and safety properties and designing its health-test architecture against BSI AIS 31 and NIST SP 800-90B requirements.
- Designing an isolated seL4 acquisition component and camera driver for the Raspberry Pi 4 and IMX296 sensor, using Verus to prove invariants over the acquisition state-machine, frame-buffer ownership, bounds and fail-closed recovery under explicit MMIO, DMA and hardware assumptions.

Algorithms and Data Structures Curriculum
Author | C++23 | OCaml

- Building a four-semester algorithms and data structures curriculum covering complexity analysis, classical structures, dynamic programming, hashing and advanced amortised techniques. Paired with implementations and exercises.

Skills

- **Languages:** C++23, Rust, OCaml, Python.
- **Systems & Graphics:** Linux, `no_std` Rust, Vulkan, asynchronous I/O, virtual memory, cache-aware data layouts, explicit CPU/GPU synchronisation, multithreading and atomics.
- **Formal methods & Security:** Creusot, executable specifications, invariant-based verification, static analysis, cryptography, HSM integration, PKCS#11, ASan, TSan, UBSan.
- **Spoken Languages:** French (Native), English (Full Professional Proficiency).

Education

École 2600 2024-09 — 2027-07
M.En. – **Cybersecurity**, **Software Engineering**, **Safety** — and **Performance** — critical software. Paris, France

Telecom Nancy 2022-09 — 2024-07
Coursework towards M.En. (Transferred) – **CS**, **Software Engineering**, **Formal Proof**. Nancy, France

Victor Hugo High School 2020-09 — 2022-07
Preparatory Classes (CPGE) – intensive **Mathematics** and **Physics** coursework. Besançon, France