

# Guillaume Petiot

## Software Engineer

(+44) 7565 986 173 • [hello@petiot.dev](mailto:hello@petiot.dev)

### PROFESSIONAL SUMMARY

Software Engineer with over a decade of expertise in functional programming, formal methods, and safety-critical systems. Adept at crafting robust software solutions in OCaml and Ada, enhancing development processes through innovative CI/CD automation. Passionate about advancing software reliability and developer experience.

### WORK EXPERIENCE

#### Senior Software Engineer | Jul 2018 - Feb 2025

*Tarides, Paris (France) and Cambridge (UK)*

Delivered robust software solutions in OCaml for data-driven applications and internal tools.

- Designed and implemented key backend components in OCaml.
- Automated build, test, and deployment pipelines (GitHub Actions), cutting integration time and improving release reliability.
- Improved development speed through better CI/CD tooling and process automation.

#### Software Engineer | Sep 2016 - Jul 2018

*Systerel, Malakoff (France)*

Developed safety-critical software for Communication-Based Train Control (CBTC) systems.

- Implemented and maintained distributed systems in Ada.
- Analyzed and resolved field incidents using trace analysis and diagnostic tooling (Perl, Shell).
- Collaborated with QA and systems teams to ensure compliance with strict safety standards.

#### Postdoctoral Researcher | Nov 2015 - Aug 2016

*Verimag, Grenoble (France)*

Investigated software protection and verification techniques for embedded systems and compilers.

- Built C++/LLVM tooling for software-on-chip security analysis, enabling automated detection of code integrity issues.
- Authored peer-reviewed papers and technical reports on software integrity and verification.

#### PhD Candidate | Oct 2012 - Sep 2015

*CEA, Saclay (France)*

Researched formal verification and programming languages, combining OCaml, C, and Prolog.

- Applied formal verification techniques to improve reliability and safety.
- Published and presented results at international conferences.
- Developed verified software prototypes and contributed to open research tooling.
- Managed project milestones, supervision, and dissemination of research outcomes.

### SKILLS

**Languages:** OCaml, C, C++, Ada, Python, Java

**Tooling:** Git, Docker, GitHub Actions, Kubernetes, CI/CD, Shell

**Practices:** TDD, Continuous Integration, Code Review, Agile

### EDUCATION

- PhD in Computer Science
- MSc in Computer Science
- BSc in Computer Science