Adrien **Ghosn**

Post Doc Researcher in Computer Science, Microsoft Research Cambridge

✉ ghosn.adrien@gmail.com | aghosn | aghosn

Education

Ecole Polytechnique Federale de Lausanne(EPFL)
Lausanne, Switzerland

Computer Science Engineering — *Sep. 2010 - 2021*

- 2016 – 2021: PhD in Datacenter System Laboratory, Prof. Edouard Bugnion and Prof. James Larus
- 2013 – 2016: Master Degree, Foundations of Software specialization (avg 5.75/6)
- 2010 – 2013: Bachelor Degree

Northeastern University(NEU)
Boston, U.S.A.

Master Thesis — *Sep. 2015 - Mar. 2016*

- Supervised by Prof. Jan Vitek in the Programming Languages Laboratory

Carnegie Mellon University(CMU)
Pittsburgh, U.S.A.

Exchange Year, Bachelor Degree in Computer Science — *Aug. 2012 - Jul. 2013*

- Dean's list School of Computer Science for QPA > 3.75/4

Industry

Microsoft Research
Cambridge UK

Researcher - Post Doc — *November 2021 - present*

- Trusted Execution Environment on legacy hardware
- Verona: Infrastructure programming language

Google Asylo team
Kirkland, USA

Summer Internship - supervisor: Matt Gingell — *June - August 2019*

- Asylo team, Trusted Execution environments, SGX
- Explored potential designs to support higher-level programming languages in SGX enclaves
- Delivered a prototype that allowed HLPL code to run inside SGX

ABB Corporate Research
Baden, Switzerland

Master Internship - supervisor: Dr. Manuel Oriol — *Feb. 2015 - Aug. 2015*

- Aperiodic-Event Support in FASA
- Fixed-priority servers, data-driven events, real-time control applications
- kernel design, dynamic linking/loading & software updates, pi-calculus

Skills

Programming	Go, C/C++, Java, Rust, Shell scripting, Assembly, Python
Knowledge in	Operating System design, Virtualization, KVM, Intel VT-x, Intel MPK Software security, Hardware Security extensions, Trusted Execution Environments Compilers, Language runtimes & virtual machines Theoretical CS, Concurrent & Distributed Algorithms

Research & Publications

Focus Areas	Systems, Virtualization, Security, Programming Abstractions Isolation of mutually distrustful software components Hardware-enforced isolation

Ongoing Research
Cambridge, UK

Imperial College London: Marios Kogias, EPFL: Prof. Edouard Bugnion, Prof. Mathias Payer *Nov. 2021 - Present*

- Tyche: Isolation Monitor
- Hardware-independent support for compartmentalization & Confidential Computing.
- Written in Rust, runs on x86 & RISC-V

PhD Thesis: Trust as a Programming Primitive
Lausanne, Switzerland

EPFL - Prof. Edouard Bugnion, Prof. James Larus *Sep. 2016 - Sep. 2021*

- Programming Language extensions for compartmentalization and confidential computing.
- Programming languages, isolation, security, confidentiality, integrity, virtualization, hardware security extensions

Enclosures: Language-based restriction of untrusted libraries [ASPLOS21]
Lausanne, Switzerland

EPFL - Prof. Edouard Bugnion, Prof. Mathias Payer *Sep. 2019 - Oct. 2020*

- New fine-grain programming abstraction to restrict public libraries access to program resources
- Frontend extensions to Go and Python PLs, backend hardware isolation enforcement (Intel VT-x & Intel MPK)
- Intra-address-space isolation, Sandboxing, Compiler, Linker, Runtime

Secured Routines: Language-based construction of TEEs [ATC19]
Lausanne, Switzerland

EPFL - Prof. Edouard Bugnion, Prof. James Larus *Jun. 2018 - May 2019*

- Extended Go programming language to support executing goroutines inside Intel SGX.
- Intel SGX, Confidentiality, Intergrity, Go, Compilers, Code partitioning, Hardware Extensions

Light-Weight Contexts in Dune
Lausanne, Switzerland

EPFL - Prof. Edouard Bugnion *Sep. 2016 - Jul. 2017*

- Process virtualization with Dune
- Intra-address space isolation, protecting secrets, memory snapshots, 5x faster than fork
- Intel VTX, Dune, Virtualization, Kernel module, Virtual Memory Management

Efficient Runtime Deoptimization for R(Master Thesis)
Boston, U.S.A.

Northeastern University - Prof. Jan Vitek *Sep. 2015 - Mar. 2016*

- Speculative optimizer for an R JIT compiler
- Removes performance bottlenecks due to the language semantics
- On-stack replacement, speculative optimizations, runtime de-optimization, R, LLVM, JIT compiler

Scalameta: AST Persistence & Obey: Code Health
Lausanne, Switzerland

EPFL, LAMP - Prof. Martin Odersky & Dr. Eugene Burmako *Jan. 2014 - Feb. 2015*

- Obey: Scala-linter for user-defined rules enforced at compile-time
- AST Persistence: typed-AST format for Scala
- Resolves compiler version incompatibilities and provides IDE macros expansion support

Operating Systems & Design 15-410
CMU

Undergraduate *Jan. 2013 - Jul. 2013*

- Implementation of a x86 Unix like Kernel in C and ASM
- Design and implementation of thread library, scheduler, virtual memory, various drivers, system calls

Management & Teaching

Grants	Swiss Joint Research Grant: Confidential Computing solutions for legacy hardware. Joint program with Microsoft Research, EPFL, Imperial College London. Involves three PhD Students.
Semester Projects	Go Intel MPK library (Charly Castes) System call interposition in Go & Python runtimes (Elsa weber)
Teaching Assistant	Functional Programming (2020), Introduction to Operating Systems (2019) Introduction to Java Programming (2018), Systems for Data Science (2017-2020) Introduction to C Programming (2016-2017), Concurrent Programming (2015) Student Volunteer at ECOOP (2016)

Personnal

Languages Fluent in French & English