

Compiler Fuzzing Course Introduction

Guillermo Polito

guillermo.polito@inria.fr
@guillep



inria



**Université
de Lille**

First: About Me

guillermo.polito@inria.fr
@guillep



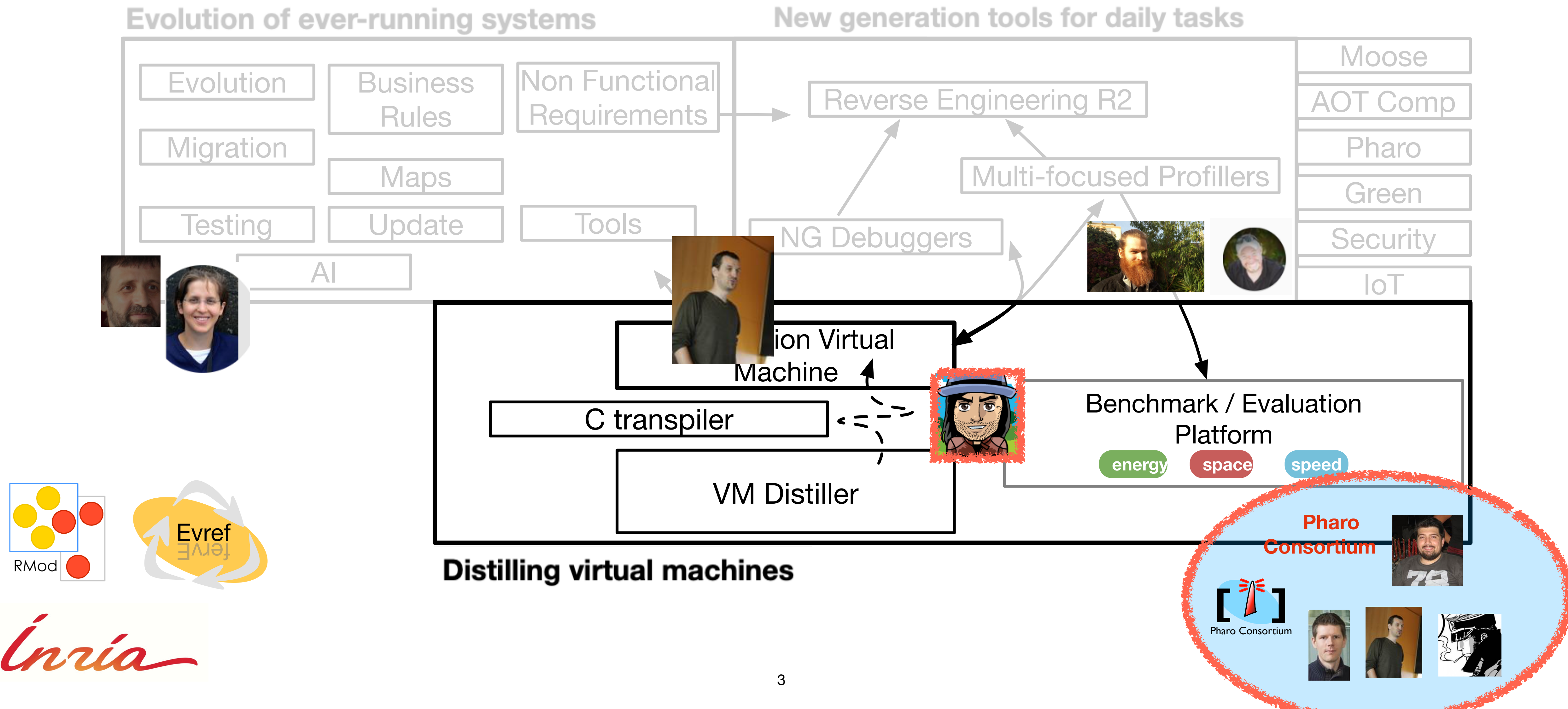
- **Now:** Permanent researcher (CRCN) at Inria - Lille since 2022
- **Ph.D.:** Reflection, debloating, dynamic updates
- **Keywords:** compilers, testing, test generation, performance
- **Interests:** tooling, benchmarking, 日本語, board games, batman, concurrency

guillermo.polito@inria.fr



Inria

Virtual Machines



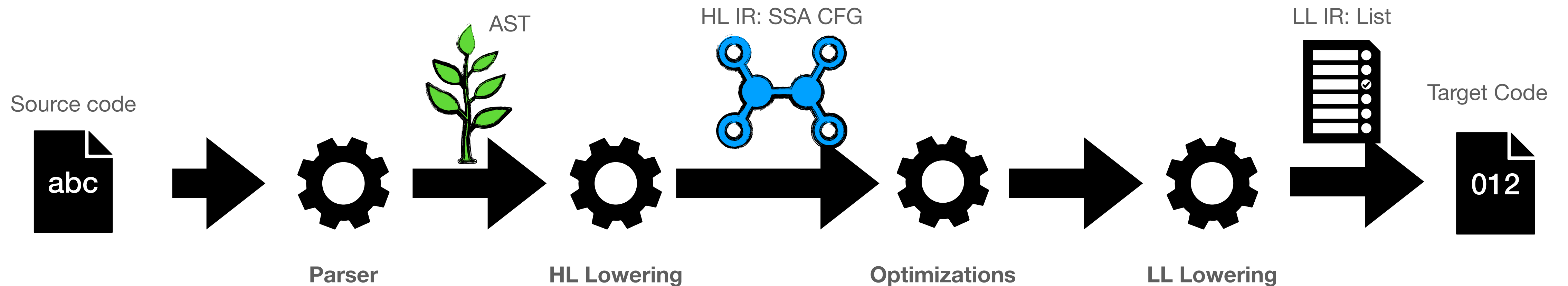
Ongoing Projects

- Ahead-of-time Language Virtual Machine optimizations
 - Benchmark generation using test generation techniques
 - Allocation profilers
-
- We are looking for PhD students, interns and young engineers!

Compiler Fuzzing Course

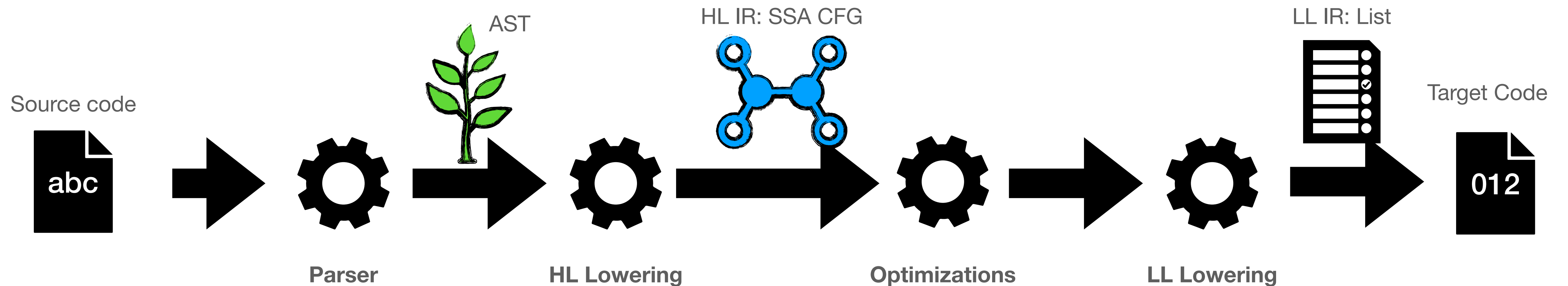
- 10 + 2 modules
 - 2 “leveling” modules: testing and compiler architecture
 - + fuzzing, oracles, syntactic fuzzing, differential testing, advanced fuzzing
- Extra citations and material: blogs, books, papers
- Practice module based on a *buggy compiler*

The SMLC compiler



- Autogenerated Parser — years mature
- High-level Lowering — custom
- Optimizations — >1 year mature
- Low-level Lowering - custom
- LL IR and code gen — >10 years mature

The SMIC compiler + bugs



- Potential bugs present
 - Left in purpose
 - Recently introduced
 - Architecture dependent
 - Maybe there for years and never hit

Course Evaluation

- Report PDF of course practice — in groups
 - What testing strategies have you tried
 - What bugs have you found
 - What was difficult to achieve, what was difficult to apply