SCIENCES & BIOENGINEERING SCIENCES

The Research Group
Software Languages Lab

has the honor to invite you to the public defence of the PhD thesis of

Laurent Christophe

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

Provenance-Aware Dynamic Analysis of JavaScript

Supervisor:
Prof. dr. Coen De Roover (VUB)
Co-supervisor:
Prof. dr. Wolfgang De Meuter (VUB)

Tuesday, January 27, 2026 at 5 p.m.

VUB Etterbeek campus, Pleinlaan 2, Elsene, Promotiezaal D.2.01

Live stream:
https://lachrist.github.io/defence

Members of the jury:
Prof. dr. Ann Nowé (VUB, chair)
Prof. dr. Jens Nicolay (VUB)
Prof. dr. Elisa Gonzalez Boix (VUB)
Prof. dr. Michael Pradel (Uni-Stuttgart, DE)
Prof. dr. Tom Van Cutsem (KULeuven)

Curriculum vitae

I'm an enthusiastic programmer and active contributor to open-source projects. My expertise includes the implementation of programming languages, functional programming, reflective programming, distributed systems, and, more recently, generative AI.

Personal homepage:
https://lachrist.github.io

LinkedIn page:
https://linkedin.com/in/lachrist

Abstract of the PhD research

Dynamic program analysis, which examines a program's behavior during execution, is essential in domains such as security and software maintenance. This dissertation focuses on provenance-aware analysis, such as dynamic taint analysis, which requires tracking how data propagates during execution. To implement analyses independently from the execution environment, a common approach is to instrument the source code of the target program. However, developing robust instrumentation-based analysis tools is hindered by three key challenges in most managed languages: the syntactic complexity of the language complicates instrumentation; optimizations in the representation of immutable values make provenance tracking difficult; and the distributed nature of modern applications requires analysis orchestration.

This dissertation presents a novel modular approach to overcome these challenges for JavaScript. To simplify instrumentation, we transpile programs into a core variant. Compared to JavaScript, this core variant employs simpler constructs, reduces the number of syntactic node types by 57%, and still passes 99.8% of the official JavaScript conformance test suite. To track immutable values, we introduce a value-centric approach that promotes primitive values to transparent reference values, which requires an access-control system based on reflection. Owing to JavaScript's reflection API, this method yields high precision and transparency, albeit with a performance cost. Finally, to coordinate the analysis of distributed applications, we centralize the analysis logic in a dedicated process. This architecture provides high expressiveness and convenience for the analysis developer, but introduces additional communication overhead.

This research advances the state of the art by delivering a modular framework for building robust instrumentation-based tools for provenance-aware analyses in JavaScript.