

**Title** A Multi Debugger for Hop.js

**Keywords** semantics, ocaml, javascript, hop, multitier

**Contact** alan.schmitt@inria.fr

**Location** Celtique team, Inria Rennes

## 1 Description

It is often said that the "S" in "IOT" ("Internet of Things") stands for security. Indeed, security is usually a secondary concern if considered at all. The ANR project CISC aims at certifying IOT applications by leveraging compilation techniques and a formalization of the protocols used. This internship takes place in that setting.

The compilation target for the protocols considered in the CISC project is an extension of the multitier language Hop.js [3, 4]. Hop.js is an extension of JavaScript [2] enabling the development of distributed applications using a single program that contains the code for every node, such as a client and a server. The purpose of this internship is to provide a tool to explore the behavior of Hop.js programs. More precisely, we propose to develop a *multi debugger* for Hop.js. A double debugger is a step-by-step interpreter that allows to inspect not only the state of the interpreted program but also the state of the interpreter, a multi debugger is a double debugger for distributed programs, where several interpreters interact. The starting point of the internship is a double debugger for JavaScript [1].

The work program of this internship are as follows.

- Define a semantics for the distributed part of Hop.js.
- Develop an OCaml interpreter for Hop.js.
- Extend the notion of double debuggers to multi debuggers for distributed programs.
- Provide a multi debugger for Hop.js.

The student will be remunerated. There is also an opportunity to continue this work as a PhD student in the setting of the CISC project.

The internship will be supervised by Alan Schmitt and will take place in the Celtique team at Inria Rennes. The student will have the opportunity to interact with the other researchers of the CISC project, including Tamara Rezk (Indes team, Inria Sophia Antipolis), Gérard Berry (Collège de France), Claude Castelluccia and Daniel Le Métayer (Privatics team, Inria Grenoble).

## References

- [1] Arthur Charguéraud, Alan Schmitt, and Thomas Wood. *JSExplain: an interactive debugger for the JavaScript specification*. 2017. URL: <https://github.com/jscert/jsexplain>.
- [2] ECMA. *ECMAScript 2017 Language Specification (ECMA-262, 8th edition)*. June 2017. URL: <https://www.ecma-international.org/ecma-262/8.0/index.html>.
- [3] Manuel Serrano, Erick Gallesio, and Florian Loitsch. “Hop: a language for programming the web 2.0”. In: *Proceedings of the First Dynamic Languages Symposium*. Portland, OR, USA: ACM, Oct. 2006, pp. 975–985.
- [4] Manuel Serrano and Vincent Prunet. “A Glimpse of Hopjs”. In: *International Conference on Functional Programming (ICFP)*. ACM. 2016.