

I'm an assistant professor at EPFL, heading the SYSTEMF lab. Previously, I was a PhD candidate at MIT with Adam Chlipala and then a senior applied scientist at Amazon AWS. My research focuses on programming languages, compilers, and formal verification; my broader interests include systems engineering, hardware design languages, security, performance engineering, and type theory. I built the first end-to-end verified compilation pipeline from high-level specifications to assembly language (POPL'15, SNAPL'17, IJCAR'20, PLDI'22, PLDI'24), the first verified compiler for a rule-based hardware design language with EHRs (PLDI'20, ASPLOS'21), and multiple widely-used IDEs and tools for automated and interactive theorem proving (CoqPL'16, SLE'20). My lab was the first to show that regex lookarounds can be matched in linear time (PLDI'24 — the resulting algorithms have been merged in Google Chrome) and to develop faithful specifications for a real-world regex language (ICFP'24, PLDI'24 SRC gold medal).

## Education

- 2014 – 22 **Massachusetts Institute of Technology** (Cambridge, USA), **MS 2016, PhD 2022**  
PhD thesis: *Relational compilation: functional-to-imperative code generation for performance-critical applications*.  
MS thesis: *Compilation using Correct-by-Construction Program Synthesis*, awarded a 🏆 **William A. Martin Memorial Thesis Award for Outstanding Thesis in CS**. Advised by Adam Chlipala.
- 2011 – 14 **École Polytechnique** (Palaiseau, France), **Diplôme d'ingénieur 2014, MSE 2016**  
MSE specializing in computer science; undergraduate in mathematics and physics.

## Selected publications

- 2024 **ICFP A Coq Mechanization of JavaScript Regular Expression Semantics**  
Noé De Santo, Aurèle Barrière, Clément Pit-Claudel.  
*Proceedings of the ACM on Programming Languages* (ICFP 2024).
- PLDI Linear Matching of JavaScript Regular Expressions**  
Aurèle Barrière, Clément Pit-Claudel.  
*Proceedings of the ACM on Programming Languages* (PLDI 2024).
- PLDI Foundational Integration Verification of a Cryptographic Server**  
Andres Erbsen, Jade Philipoom, Dustin Jamner, Ashley Lin, Samuel Gruetter, Clément Pit-Claudel, Adam Chlipala.  
*Proceedings of the ACM on Programming Languages* (PLDI 2024).
- 2022 **PLDI Relational Compilation for Performance-Critical Applications: Extensible Proof-Producing Translation of Functional Models into Low-Level Code**  
Clément Pit-Claudel, Jade Philipoom, Dustin Jamner, Andres Erbsen, Adam Chlipala.  
*Proceedings of the ACM on Programming Languages* (PLDI 2022).
- 2021 **ASPLOS Effective Simulation and Debugging for a High-Level Hardware Language using Software Compilers**  
Clément Pit-Claudel, Thomas Bourgeat, Stella Lau, Arvind, Adam Chlipala.  
*Proc. 26th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems* (ASPLOS 2021).
- 2020 **SLE Untangling Mechanized Proofs**  
Clément Pit-Claudel.  
*Proc. 13th ACM SIGPLAN Intl. Conf. on Software Language Engineering* (SLE 2020). 🏆 **Distinguished artifact**.
- PLDI The Essence of Bluespec: A Core Language for Rule-Based Hardware Design**  
Thomas Bourgeat, Clément Pit-Claudel, Adam Chlipala, Arvind.  
*Proceedings of the ACM on Programming Languages* (PLDI 2020).
- 2016 **CAV Trigger Selection Strategies to Stabilize Program Verifiers**  
K. Rustan M. Leino and Clément Pit-Claudel.  
*Computer Aided Verification: 28th Intl. Conf.* (CAV 2016).
- 2015 **POPL Fiat: Deductive Synthesis of Abstract Data Types in a Proof Assistant**  
Benjamin Delaware, Clément Pit-Claudel, Jason Gross, Adam Chlipala.  
*Proc. 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (POPL 2015).

See page 5 for a complete list of publications.

## Research projects

### Formal methods for real-world DSLs

**Elk** is a linear-time engine for a subset of JavaScript regular expressions (regexes). Some of the algorithms pioneered in Elk have been merged into V8 (and hence Chrome and Node.js). The project was led by my post-doc Aurèle Barrière.

**Warblre** is a near-complete Coq translation of the JS regex specification. Developing this spec helped us find mistakes in previous paper-based specifications and disprove longstanding assumptions about JS regexes. The project was led by Noé De Santo, an MS thesis student, and co-supervised by my post-doc Aurèle Barrière. 🏆 **SRC Gold medal (1st place, graduate division)**

### Correct-by-construction refinement

**Fiat** is a library for the Coq proof assistant that lets users automatically refine declarative specifications into efficient functional programs. I wrote parts of the framework and examples and did the evaluation.

**Narcissus** is an extensible library of context-sensitive parser combinators, general enough to specify and automatically derive verified encoders and decoders for a wide range of binary formats like Ethernet, ARP, TCP, IP, etc. I wrote bit-manipulation libraries and data structures and did the integration and evaluation.

### Extensible proof-producing compilers and binary code extraction

**F2F** is a program extraction framework for Coq that uses syntax-driven automation to derive correct-by-construction imperative programs from nondeterministic functional source code. I am the sole author.

**Rupicola** is a compiler-construction toolkit that lets users assemble verified domain-specific compilers from reusable translation lemmas, producing high-performance low-level code from unoptimized domain-specific functional models. I am the lead author.

### Hardware design languages: semantics and verification

**Kôika** is a rule-based hardware design language with cycle-accurate semantics formalized in Coq. It features high-level abstractions inherited from Bluespec, executable semantics proven to refine one-rule-at-a-time execution, and a formally-verified compiler that generates circuits with good performance. I am one of two lead authors; I formalized the semantics in Coq and designed and verified the compiler to circuits.

**Cuttlesim** is a fast cycle-accurate simulator for Kôika that beats state-of-the-art RTL simulators by a factor 2 to 5 by leveraging high-level information to minimize redundant work. Cuttlesim generates C++ models that are readable enough to enable hardware debugging and testing using traditional software tools. I am the lead author.

### Tooling for interactive theorem provers

**Alectryon** is a literate-programming system for Coq that produces interactive visualizations of Coq proofs. Alectryon offers a new way to write, communicate, and preserve proofs, combining the flexibility of procedural proof scripts and the intelligibility of declarative proofs. I am the sole author. 🏆 **Distinguished artifact**

## Teaching experience

- |      |  |
|------|--|
| 2024 | <b><i>Interactive Theorem Proving (CS-628)</i></b> — EPFL<br><b>Created and taught a hands-on introduction to formal verification and computer-checked mathematics</b> , including compiler verification, proofs as programs, and proof automation. <b>4-month</b> instructorship  |
| 2023 | <b><i>Software Construction (CS-214)</i></b> — EPFL<br><b>Designed and taught a new software engineering curriculum to 400 students; managed a team of 30 people.</b> Introduced a new lecture series on debugging; a software-correctness track; refactoring-focused lab callbacks; hands-on exercises; live-coding demos; and a career-advice lecture. The course received 98% positive ratings. <b>4-month</b> instructorship |
| 2021 | <b><i>Kaufman Teaching Certificate Program</i></b> — Teaching+Learning Lab, MIT<br><b>Successfully completed the KTCP, MIT's flagship teaching development program</b> , with 15-hours of in-class instruction and 30 hours of pre- and post-workshop assignments aimed at developing my teaching skills. <b>3 month</b> teaching certification program  |

- 2021 **Formal Reasoning about Programs — MIT**  
Created problem sets, developed automated grading technology, held debriefings, and taught students one-on-one and in small groups; received an average rating of 6.8/7 with 80% perfect scores.  
3 month teaching assistantship
- 2016 **Fundamentals of Programming — MIT**  
Designed labs, taught recitations, prepared debriefings; received 65 student reviews with an average teaching rating of 6.8/7 and 55 perfect teaching scores, and was awarded a 🏆 **Frederick C. Hennie III Teaching Award in Recognition of Outstanding Contributions to Departmental Teaching.**  
4-month teaching assistantship
- 2011–12 **Association Tremplin**  
Designed AP-level mathematics, physics, and computer science classes and taught over 60 motivated teenagers across three high schools from low-income neighborhoods. Classes were intended to encourage college applications, boost student confidence, and increase preparedness. Over 95% of the students completed the program successfully.  
7-month civil service in Seine-Saint-Denis, France

## Industry experience

- 2021–22 **Amazon AWS (Senior Applied Scientist, Automated Reasoning Group, Dafny team)**  
Launched Dafny-in-Dafny, a project to bootstrap and verify parts of Dafny's compilation pipeline; designed and implemented multiple new languages features, including changes in pattern matching, name resolution, and type-checking; audited parts of Dafny for soundness, discovering and fixing several soundness bugs; developed freely available educational materials and verified libraries for Dafny; provided technical and scientific support to existing customers; helped new customers adopt and get proficient in Dafny; participated in recruiting; mentored interns and colleagues; contributed to the day to day maintenance and evolution of Dafny.  
1 year pre-battical in Vancouver, WA
- 2017 **INRIA (Prosecco team, supervised by Cătălin Hrițcu)**  
Implemented reflective pattern-matching tactics for F\*; contributed to research on effect erasure; rewrote F\*'s IDE protocol and state machine to implement a full-fledged IDE; built a literate programming system for F\*; assembled web-based builds of F\* and Z3.  
3-month internship in Paris, France
- 2015 **Microsoft Research (RISE group, supervised by K. Rustan M. Leino)**  
Improved the predictability and robustness of the Dafny program verifier, generating custom triggers to prevent spurious quantifier instantiations and matching loops.  
3-month internship in Redmond, WA
- 2013 **华为 | Huawei**  
Designed and implemented a fast upper body limb detection and tracking engine based on Bayesian inference in graphical models.  
6 weeks internship in Shenzhen, China
- 2009 **CSSI Communication & Systèmes**  
Built a Fortran static analyzer to analyze control flow, build call graphs, and locate dead code.  
1 month internship in Le Plessis Robinson, France

## Entrepreneurship

- 2013–17 **Launched YiXué Chinese Dictionary, a paid English-Chinese dictionary for Windows Phone. YiXué sold over 1200 copies, achieved a 4.7★ rating with over 200 reviews, and was featured three times on Microsoft's app store.**
- 2009–now **Launched and maintained Create Synchronicity, an open source backup & synchronization app (450k downloads, 2k daily users, translated to 30 languages). Featured in PC Magazine's Best free software of 2011 and Computer Bild's Open Source DVD.**

## Service

**EPFL:** Working group on science & engineering of data (2023).

**IC:** Teaching committee (since Fall 2023), Curriculum committee (since Fall 2023), Faculty candidate interviews (2022, 2023), PhD recruiting committee (since Fall 2021, area chair for PLFM since Fall 2023), Instructor recruiting panel (Spring 2023), MS reform committee (2024).

**Events:** PLDI'24-25 (Workshops and tutorials chair), Coq Workshop'24 (PC chair), Proof Systems'24 (Organizer), POPL'24 (Session previews chair), Swiss Verification Day'24 (Organizer), CAV'21 (AEC chair).

**Reviewing:** OOPSLA'25, OSDI'25, CPP'25, POPL'25, HATRA'24, SEFI'24, ICFP'24, ESOP'24, PLDI'24, SPLASH'23 SRC, APLAS'23, CPP'23, CoqPL'23, LATTE'23, POPL'23, PriSC'23, PLDI'22, Coq Workshop'22, LATTE'21, PLOS'21, Formal Methods Symposium, Journal of Statistical Computation and Simulation, Journal of Functional Programming.

**Other roles:** Invited speaker (CoqPL'22); Session chair (ICFP'24 (Verification and Cost Analysis), PLDI'24 (Verification II), POPL'24 (Session preview), PLDI'22 (Hardware I, Memory), SPLASH'21 (SIGPLAN)); SRC judge (PLDI'24); AEC member (POPL'18, POPL'16); Mentor (PLATEAU'24, PLATEAU'23); Student PC member (IEEE S&P'19); Conference volunteer (PLDI'18, SPLASH'18).

## Awards

**Distinguished artifact**, *Untangling Mechanized Proofs*, ACM SIGPLAN International Conference on Software Language Engineering (2020).

**William A. Martin Memorial Thesis Award for Outstanding Thesis in CS**, *Compilation using Correct-by-Construction Program Synthesis*, MIT (2016).

**Frederick C. Hennie III Teaching Award in Recognition of Outstanding Contributions to Departmental Teaching**, 6.009 *Fundamentals of Programming*, MIT (2016).

**Robert B. Guenassia Award**, MIT (2015).

**2<sup>nd</sup> place**, *École Polytechnique's Best Group Research Project*, for research on webcam-based gaze tracking (2013).

**2<sup>nd</sup> place**, Microsoft France's *App Awards*, a mobile development contest, for *YiXué Chinese Dictionary* (2013).

**3<sup>rd</sup> place**, Computer Science **competitive entrance exam** of the *École Normale Supérieure (ENS Ulm)* (2011).

**4<sup>th</sup> place**, **French Olympiads in Mathematics, Paris division** (2008).

## Other activities

**Launched the proofs.swiss initiative** to strengthen the Swiss formal-methods community, and jointly organized the first *Swiss Verification Day* (68 participants, 5 universities, 3 companies) at EPFL and the 2024 *Proof Systems for Mathematics & Verification* workshop (50 participants, 9 countries) at Unil.

**Launched or contributed to many free software projects** since 2012, including plugins and translations for Rockbox (a free audio player firmware), documentation tools for Coq, Lean and F\*, and Emacs packages for bibliography management, on-the-fly syntax checking, and various research languages (I wrote biblio.el, ESH, company-coq, F\*-mode, and boogie-friends; I co-maintain Flycheck and Proof-General).

## Lab highlights (SYSTEMF)

**We hosted 4 MS theses, 6 semester projects, 3 interns, and 3 assistants** from 2023 to 2024.

**One of our MS thesis students won first place at PLDI'24's ACM student research competition** (gold medal, graduate division).

**Our linear-time regex-matching algorithms were merged in Google's V8 JavaScript engine** (Google Chrome & Node.js).











## Skills

**Software design & development:** Wrote professional software in C#, Emacs Lisp, Python, TypeScript, JavaScript, VB.Net, and C++. Other languages include OCaml, Coq, HTML+CSS, C, Java, Dafny, F\*, Racket, PHP.


**Languages:** French (mother tongue, French citizen), English (fluent), Spanish, Chinese, Japanese (basic).


## All publications


- 2024 **SOSP Practical Verification of System-Software Components Written in Standard C**  
🏆🔒 Can Cebeci, Yonghao Zou, Diyu Zhou, George Candea, Clément Pit-Claudel.  
*Proc. 30th Symposium on Operating Systems Principles (SOSP 2024).*
- CCS Specification and Verification of Strong Timing Isolation of Hardware Enclaves**  
🔒🔒🔒 Stella Lau, Thomas Bourgeat, Clément Pit-Claudel, Adam Chlipala.  
*Proc. 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS 2024).* 🏆 Distinguished artifact.
- OOPSLA Gradient: Gradual Compartmentalization via Object Capabilities Tracked in Types**  
Aleksander Boruch-Gruszecki, Adrien Ghosn, Mathias Payer, Clément Pit-Claudel.  
*Proceedings of the ACM on Programming Languages (OOPSLA 2024).*
- ICFP A Coq Mechanization of JavaScript Regular Expression Semantics**  
🔒 Noé De Santo, Aurèle Barrière, Clément Pit-Claudel.  
*Proceedings of the ACM on Programming Languages (ICFP 2024).*
- PLDI Linear Matching of JavaScript Regular Expressions**  
🔒 Aurèle Barrière, Clément Pit-Claudel.  
*Proceedings of the ACM on Programming Languages (PLDI 2024).*
- PLDI Foundational Integration Verification of a Cryptographic Server**  
🔒 Andres Erbsen, Jade Philipoom, Dustin Jamner, Ashley Lin, Samuel Gruetter, Clément Pit-Claudel, Adam Chlipala.  
*Proceedings of the ACM on Programming Languages (PLDI 2024).*
- Dafny Incremental Proof Development in Dafny with Module-Based Induction**  
Son Ho, Clément Pit-Claudel.  
*Dafny 2024 (Dafny 2024).*
- 2023 **HATRA Diagrammatic notations for interactive theorem proving**  
Shardul Chiplunkar, Clément Pit-Claudel.  
*4th International Workshop on Human Aspects of Types and Reasoning Assistants (HATRA 2023).*
- PLARCH Hardware Verification of Timing Side Channel Freedom in the Spectre Era**  
Stella Lau, Thomas Bourgeat, Clément Pit-Claudel, Adam Chlipala.  
*Workshop on Programming Languages for Architecture (PLARCH 2023).*
- 2022 **PLDI Relational Compilation for Performance-Critical Applications:**  
🔒 **Extensible Proof-Producing Translation of Functional Models into Low-Level Code**  
Clément Pit-Claudel, Jade Philipoom, Dustin Jamner, Andres Erbsen, Adam Chlipala.  
*Proceedings of the ACM on Programming Languages (PLDI 2022).*
- (PhD thesis) **Relational compilation: functional-to-imperative code generation for performance-critical applications**  
Clément Pit-Claudel.  
PhD thesis at MIT.
- 2021 **ASPLOS Effective Simulation and Debugging for a High-Level Hardware Language using Software Compilers**  
🔒🔒 Clément Pit-Claudel, Thomas Bourgeat, Stella Lau, Arvind, Adam Chlipala.  
*Proc. 26th Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2021).*
- CoqPL Automated Synthesis of Verified Firewalls**  
Shardul Chiplunkar, Clément Pit-Claudel, Adam Chlipala.  
*The Seventh International Workshop on Coq for PL (CoqPL 2021).* Workshop paper.
- CoqPL An experience report on writing usable DSLs in Coq**  
Clément Pit-Claudel, Thomas Bourgeat.  
*The Seventh International Workshop on Coq for PL (CoqPL 2021).* Workshop paper.


- 2020 **SLE Untangling Mechanized Proofs**  
   Clément Pit-Claudel.  
*Proc. 13th ACM SIGPLAN Intl. Conf. on Software Language Engineering (SLE 2020)*.  **Distinguished artifact**.
- IJCAR Extensible Extraction of Efficient Imperative Programs with Foreign Functions, Manually Managed Memory, and Proofs**  
 Clément Pit-Claudel, Peng Wang, Benjamin Delaware, Jason Gross, Adam Chlipala.  
*International Joint Conference on Automated Reasoning (IJCAR 2020)*.
- PLDI The Essence of Bluespec: A Core Language for Rule-Based Hardware Design**  
  Thomas Bourgeat, Clément Pit-Claudel, Adam Chlipala, Arvind.  
*Proceedings of the ACM on Programming Languages (PLDI 2020)*.
- 2019 **ICFP Narcissus: Correct-by-Construction Derivation of Decoders and Encoders from Binary Formats**  
 Benjamin Delaware, Sorawit Suriyakarn, Clément Pit-Claudel, Qianchuan Ye, Adam Chlipala.  
*Proceedings of the ACM on Programming Languages (ICFP 2019)*.
- ESOP Meta-F\* : Proof Automation with SMT, Tactics, and Metaprograms**  
 Guido Martínez, Danel Ahman, Victor Dumitrescu, Nick Giannarakis, Chris Hawblitzel, Cătălin Hrițcu, Monal Narasimhamurthy, Zoe Paraskevopoulou, Clément Pit-Claudel, Jonathan Protzenko, Tahina Ramananandro, Aseem Rastogi, Nikhil Swamy.  
*Proc. 28th European Symposium on Programming (ESOP 2019)*.
- 2018 **SETTA Correct-by-Construction Implementation of Runtime Monitors Using Stepwise Refinement**  
 Teng Zhang, John Wiegley, Theophilos Giannakopoulos, Gregory Eakman, Clément Pit-Claudel, Insup Lee, Oleg Sokolsky.  
*Proc. 4th International Symposium on Dependable Software Engineering: Theories, Tools, and Applications (SETTA 2018)*.
- ML ML as a Tactic Language, Again**  
 Guido Martínez, Danel Ahman, Victor Dumitrescu, Nick Giannarakis, Chris Hawblitzel, Cătălin Hrițcu, Monal Narasimhamurthy, Zoe Paraskevopoulou, Clément Pit-Claudel, Jonathan Protzenko, Tahina Ramananandro, Aseem Rastogi, Nikhil Swamy.  
*ML family workshop at ICFP 2018 (ML 2018)*. Workshop paper.
- 2017 **SNAPL The End of History? Using a Proof Assistant to Replace Language Design with Library Design**  
 Adam Chlipala, Benjamin Delaware, Samuel Duchovni, Jason Gross, Clément Pit-Claudel, Sorawit Suriyakarn, Peng Wang, Katherine Ye.  
*2nd Summit on Advances in Programming Languages (SNAPL 2017)*.
- 2016 **CAV Trigger Selection Strategies to Stabilize Program Verifiers**  
 K. Rustan M. Leino and Clément Pit-Claudel.  
*Computer Aided Verification: 28th Intl. Conf. (CAV 2016)*.
- CoqPL Company-Coq: Taking Proof General one step closer to a real IDE**  
 Clément Pit-Claudel, Pierre Courtieu.  
*The Second International Workshop on Coq for PL (CoqPL 2016)*. Workshop paper.
- (MS thesis) **Compilation Using Correct-by-Construction Program Synthesis**  
 Clément Pit-Claudel.  
 Master's thesis at MIT.  **William A. Martin Memorial Thesis Award for Outstanding Thesis in CS**.
- 2015 **POPL Fiat: Deductive Synthesis of Abstract Data Types in a Proof Assistant**  
 Benjamin Delaware, Clément Pit-Claudel, Jason Gross, Adam Chlipala.  
*Proc. 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2015)*.
- (Tech report) **Outlier Detection in Heterogeneous Datasets using Automatic Tuple Expansion**  
 Clément Pit-Claudel, Zelda Mariet, Rachael Harding, Sam Madden.  
 Technical report.


## Key

 This publication received an award.

 This publication received the “Artifacts Evaluated” badge (used in some conferences that do not distinguish between “functional” and “reusable”).

 This publication received the “Artifacts Evaluated – Functional” badge: “The artifacts associated with the research are found to be documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation.”

 This publication received the “Artifacts Evaluated – Reusable” badge: *“The artifacts associated with the paper are of a quality that significantly exceeds minimal functionality. That is, they have all the qualities of the Artifacts Evaluated – Functional level, but, in addition, they are very carefully documented and well-structured to the extent that reuse and repurposing is facilitated. In particular, norms and standards of the research community for artifacts of this type are strictly adhered to.”* (not all conferences offer this badge; all my eligible papers received it).

 This publication received the “Artifacts Evaluated – Results reproduced” badge: *“The main results of the paper have been obtained in a subsequent study by a person or team other than the authors, using, in part, artifacts provided by the author.”* (not all conferences offer this badge; all my eligible papers received it).