

I'm a senior applied scientist at Amazon AWS, soon to join EPFL as an assistant professor. Previously, I was a PhD candidate at MIT with Adam Chlipala. My research focuses on programming languages, compilers, and formal verification; my broader interests include systems engineering, hardware design languages, security, performance engineering, databases, 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), the first verified compiler for a rule-based hardware design language with EHRs (PLDI'20), and multiple widely-used Coq tools (CoqPL'16, SLE'20).

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

- 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). **Dist** 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).
- 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.
- 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

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.

Teaching experience

- 2021 **Kaufman Teaching Certificate Program — Teaching+Learning Lab, MIT**
Successfully completed the KTCP, MIT's flagship teaching development program, with 15-hours of in-class instruction and 30 hours of pre- and post-workshop assignments aimed at developing my teaching skills.
3 month 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 months** civil service in Seine-Saint-Denis, France

Research & 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 WA, USA
- 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 months** 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 months** 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
- École Polytechnique (LIX, supervised by Stéphane Graham-Lengrand)**
Improved the performance of the Psyche theorem prover by extending its DPLL module to allow restarts. **3 months** research project in Palaiseau, France

Entrepreneurship

- 2013–17 **Launched YiXué Chinese Dictionary, a paid English-Chinese dictionary for Windows Phone. YiXué sold over 1000 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.**

Community service, mentorship, and outreach

Co-chaired CAV 2021's artifact evaluation committee.

Served on the program committee of **POPL'23, CPP'23, PLDI'22, PriSC'23, LATTE'21, and PLOS'21**; on the artifact evaluation committee of **POPL'16 and POPL'18**; on the student program committee of **IEEE S&P'19**; and reviewed for **NASA's Formal Methods Symposium, the Journal of Statistical Computation and Simulation, and the Journal of Functional Programming.**

Volunteered at **PLDI'18** and **SPLASH'18**.

Mentored **six undergraduate and two MS-level MIT students** for periods of six to eighteen months, providing guidance and hands-on software-design and verification experience (2014-2021).

Tutored early-undergraduate alumni of *Association Tremplin*, a French non-profit working to encourage high-school students from disadvantaged neighborhoods to pursue a higher education (2012; 4 hours per week for 7 months).

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).

Programming Languages Mentoring Workshop travel grant, POPL (2015).

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

2nd place, Microsoft France's **App Awards**, a mobile development contest, for *YìXué Chinese Dictionary* (2013).

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

4th place, **French Olympiads in Mathematics, Paris division** (2008).

Extra-curricular activities

Blogged about mathematics, programming, and verification; got featured in the *Code Project's Insider* daily newsletter.

Launched a twitter feed about Chinese linguistics and etymology, with over 500 subscribers.

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 and F*, and Emacs packages for bibliography management, 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*).

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

Invited talks, site visits, and conference and workshop presentations

Building Systems That We Can Trust: Compilers, Semantics, and Tooling for End-to-End Verified Systems

- 2021 June Online tech talk, Automated Reasoning Group (ARG) at Amazon AWS.
- 2021 Mar. Online seminar presentation, Carnegie Mellon University (CMU).
- 2021 Mar. Online talk, SRI International.
- 2021 Mar. Online presentation, *IC Colloquium* at EPFL.
- 2021 Mar. Online talk, RiSE group at Microsoft Research.
- 2021 Mar. Online seminar presentation, ETH.

Effective Simulation and Debugging for a High-Level Hardware Language using Software Compilers

- 2021 April Online conference presentation, *ASPLOS 2021*.

Coq meets literate programming: Tools for documenting, preserving, and sharing mechanized proofs

- 2021 Oct. Keynote, *CoqPL 2022*.

Untangling Mechanized Proofs

- 2021 Oct. Conference presentation, *SPLASH 2021*.
- 2021 May Invited talk, *Coq Working group* at LABRI.
- 2020 Dec. Plenary talk, *6th Coq Users and Developers Workshop*.
- 2020 Nov. Conference presentation, *SLE 2020 (Online)*.

An Experience Report on Writing Usable DSLs in Coq

- 2021 Jan. Workshop presentation, *CoqPL workshop at POPL 2021*.

Extensible Extraction of Efficient Imperative Programs

- 2021 Jan. Invited talk, *Cambium seminar* at INRIA.
- 2020 Dec. Invited talk, *Principles of programming & verification seminar* at Boston University.
- 2020 July Conference presentation, *IJCAR 2020*.

The Essence of Bluespec — A Core Language for Rule Based Hardware Design

- 2021 April Talk, *Programming languages and systems engineering seminar* at MIT.
- 2021 April Online invited talk, Google.
- 2020 Dec. Online invited talk, *Circuit IR compilers and tools seminar*.
- 2020 Sept. Online progress report to Sandia National Labs, MIT.
- 2020 July Online invited talk, *Computer architecture & programming abstractions seminar* at Cornell.
- 2020 July Online discussion, *PRL reading group* at Northeastern University.
- 2020 June Online conference presentation, *PLDI 2020*.

Kôika: A Core Hardware Language with Cycle-Accurate Semantics

- 2019 Oct. Technology demonstration to Sandia National Labs, MIT.
- 2019 Oct. Workshop presentation, *New England Systems Verification Day* at MIT.

Narcissus: Correct-by-Construction Derivation of Decoders and Encoders from Binary Formats

- 2019 Aug. Conference presentation, *ICFP 2019*.
- 2019 Aug. Invited talk, INRIA.
- 2018 Oct. Workshop presentation, *New England Systems Verification Day* at MIT.
- 2018 June Lightning talk, *DeepSpec workshop at PLDI 2018*.

Proof-Producing Extraction of Binary Encoders/Decoders

- 2018 June Invited talk, BedRock Systems.
- 2018 May Technology demonstration to DARPA's Information Innovation Office, MIT.

Deductive Synthesis of Abstract Data Types in a Proof Assistant

- 2017 May Guest lecture, *Formal Reasoning About Programs* at MIT.

Compilation Using Correct-by-Construction Program Synthesis

- 2016 June Site visit for DARPA's High assurance cyber military systems (HACMS), MIT.
- 2016 May Lightning talk, *Programming languages offsite* at MIT.
- 2016 April Lightning talk, *Specialist meeting on verified trustworthy software systems* at Imperial College.

Trigger Selection Strategies to Stabilize Program Verifiers

- 2016 July Conference presentation, *CAV 2016*.
- 2016 July Invited talk, INRIA.

Company-Coq: Taking Proof General one step closer to a real IDE


- 2016 Jan. Workshop presentation, *CoqPL workshop at POPL 2016*.

Quantifiers meet their match(ing loop): new techniques and tools for dealing with unpredictable performance in Dafny

- 2015 Sept. End-of-internship talk, Microsoft Research.

All publications

2022 **PLDI Relational Compilation for Performance-Critical Applications:**


 **Extensible Proof-Producing Translation of Functional Models into Low-Level Code**

Clément Pit-Claudél, 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-Claudél.
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).