Caner Derici, PhD

dericilab.live · caner@dericilab.live · canerderici · cderici · extended (CV) · UT, USA

Technical Skills

Areas of Expertise:	Compilers & Programming Languages · Distributed Systems · Machine Learning
Languages:	Go · C/C++ · Python · Racket/Scheme · LLVM · Java · SQL · JavaScript
Cloud:	Kubernetes · AWS · GCE · Terraform · LXD · Docker
Productivity & Workflows:	Obsidian · Toggl · Todoist · Linux · Neovim · VSCode · Copilot · ChatGPT · Git · GH Actions
API, DB & Misc:	REST · gRPC · OpenAPI · FastAPI · DQLite · MongoDB · PostgreSQL · CI/CD · Jenkins

Education

Ph.D., Indiana University, Bloomington, Computer Science, Compilers & Programming Languages — 2015 – 2025
Dissertation: Self-Hosting Functional Programming Languages on Meta-Tracing JIT Compilers

M.Sc., Boğaziçi University, Computer Science, Machine Learning & Natural Language Processing — 2012 – 2015

B.Sc., Bilgi University, Computer Science — 2005 – 2010

Experience

Canonical USA — Remote, US

Software Engineer II (L4), Enterprise Cloud Engineering, Juju team — *2021 – 2024*

- Developed and maintained Juju (see Projects below) as part of a 10-engineer team.
- Improved reliability and fault tolerance by implementing edge machine services on relational DQLite back-end, migrating from NoSQL MongoDB (e.g., sample PR).
- Owned client libraries for three years–python-libjuju, Terraform Juju Provider; doubled active users and maintained a steady release cadence.
- Took part in roadmap planning, coordinated cross-team efforts; mentored junior engineers, and improved hiring by creating a structured Juju-Bootcamp process that cut ramp-up from 6 months to 1 month for new engineers.

Indiana University — IN, US

Research Assistant, Course Instructor — *2015 – 2021*

- Independently took an ambiguous, uncharted problem from zero to working product; built the first-ever tracing JIT compiler that is a full-scale runtime for a self-hosting, production-grade language. Conducted a full performance investigation and designed new optimization algorithms (see Pycket below).
- Taught data structures & algorithms, compilers, virtual machines, and domain specific languages.

Asseco SEE Group

Software Engineer — *2010-2012*

- International software company developing virtual payment platforms for e-commerce platforms. I developed and delivered 3 virtual point-of-sale projects in 2 years. Used Java, Apache Tomcat, Spring, Mercurial, Jira.

Selected Projects

Juju: A large scale, eventually consistent distributed orchestration system
Used by ~200 companies globally for managing cloud workloads on any infrastructure (Kubernetes or otherwise) across various cloud providers (e.g., AWS, GCE), capable of handling 1000+-node workloads with 99.9% availability. See Canonical above for my contributions. In Go, and Python.

Pycket: A tracing JIT compiler for full-scale Racket
Developed and maintained for over five years. Designed the compiler to self-host a language on a meta-tracing JIT backend. Contributed designing a new IR (see publications). Built performance analysis tools, run-time optimizations, and formalisms to improve performance. Implemented code-gen for full FFI layer, and engines with meta-continuations for preemption for green threads. In Python, C, and Racket.

Rax: A full-stack nanopass compiler from Racket to x86_64
Implemented all the passes (e.g., closure conversion, register allocation, code-gen, etc.), along with garbage collection. Developed optimizations, such as inlining, loop-invariant code motion, and proper tail-calls. In Racket, and C.

FARS: Functional Automated Reasoning System
A resolution/refutation theorem prover, for expressions in first-order predicate logic with equality. Used binary paramodulation, and forward and backward subsumption for equational deduction. In Racket.

HazirCevap (Witty): A closed domain question answering system for high school students
Government funded large scale question answering system. M.Sc. thesis on NLP. Led the R&D team (3 faculties, 4 grad students). Developed a Hidden Markov Random Field model for question analysis, and relevance metrics for information retrieval and response generation (see publications). In Python, and JavaScript.

Last compiled on September 27, 2025

Selected Publications

- Flatt M., Derici C. Dybvig R. K., Keep A. et. al. "Rebuilding racket on chez scheme (experience report)", ICFP'19
- Derici C. et. al. "A closed-domain question answering framework using reliable resources to assist students" Natural Language Engineering'18
- Derici C. et. al. "Question analysis for a closed domain question answering system", CICLING'15
- Derici C. et. al. "Rule-based focus extraction in Turkish question answering systems", SIU'14
- Başar R. E., Derici C., and Şenol Ç. "World With Web: A compiler from world applications to JavaScript". Technical Report, Scheme and Functional Programming Workshop'09

Awards & Scholarships

- Scholarship and award for a project on teaching natural languages to hearing impaired, 2014.
- Full Scholarship for PhD, 2015-2020
- Full Scholarship for MSc, 2012
- Full Scholarship for BSc, 2005-2010