

Caner Deric

dericilab.live @ caner@dericilab.live [in canerderici](https://www.linkedin.com/in/canerderici) [cderici](https://github.com/cderici) [extended \(CV\)](#) [UT, USA](#)

Technical Skills

Areas of Expertise: Distributed Systems · Compilers & Programming Languages · Machine Learning
Languages: Go · Python · C/C++ · Racket/Scheme · Java · SQL · JavaScript
Cloud: Kubernetes · AWS · GCE · Terraform · LXD · Docker
API, DB & Misc: REST · gRPC · DQLite · MongoDB · PostgreSQL · Git · CI/CD · GitHub Actions · Jenkins

Experience

Canonical USA REMOTE, US

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

Worked on distributed orchestration system handling 1000+ node cloud workloads across multiple regions with 99.9% availability. Primarily in Go, and Python.

- Architected full-stack distributed components, improving reliability, fault tolerance, back-pressure handling on an eventually consistent backend.
- Doubled the user base of client libraries [Python-libjuju](#), [Terraform Juju Provider](#)), owned all deliverables for 3 years, and maintained release cadence.
- Helped transition [Juju](#)'s data model from NoSQL MongoDB to relational DQLite (e.g., [sample PR](#)).
- Redesigned the facade-based RPC API and developed a REST API alternative with OpenAPI.
- Took part in roadmap planning, cross-team coordination, mentoring junior engineers, and hiring.

Indiana University IN, US

Course Instructor, Teaching & Research Assistant 2015 – 2021

Work during PhD. I taught data structures & algorithms, compilers, virtual machines, and domain specific languages. Researched runtime performance of JIT compiled VMs for functional languages.

Asseco SEE Group

Software Engineer 2012-2013

International software company developing virtual payment platforms for e-commerce platforms. I developed and delivered 3 virtual point-of-sale projects in 1 year. Used Java, Apache Tomcat, Spring, Mercurial, Jira.

Education

PhD (abd), [Indiana University, Bloomington](#), Computer Science, Programming Languages

Optimizing VM run-times for self-hosting dynamic languages on meta-tracing JIT compilers.

MSc, [Boğaziçi University](#), Computer Science, Machine Learning, Natural Language Processing

BSc, [Bilgi University](#), Computer Science

Selected Projects

Juju

A large scale distributed orchestration engine for managing cloud workloads on any infrastructure (Kubernetes or otherwise) across various cloud providers (e.g., AWS, GCE). Juju is written in Go, and has over a million lines of code. I was a core maintainer in a team of 14 engineers. See Canonical above for details of my contribution.

Terraform Juju Provider

A Terraform provider that enables integration with [Juju](#) while managing Terraform environments. I implemented new resources and features (e.g., [manual provisioning on AWS](#)), migrated the provider from the sdk2 to the provider framework (e.g., [sample PR](#)), and maintained release cadence of new versions. All in Go.

Pycket: A meta-tracing JIT compiler for self-hosting Racket

PhD thesis project. Developed and maintained Pycket for over five years, designing the compiler to bootstrap the entire Racket language on a meta-tracing JIT backend. Contributed to a new IR (linklets, see publications) for a more portable Racket run-time. Built [performance analysis tools](#) and [formalisms](#) to improve meta-tracing efficiency. Implemented run-time optimizations, data structures, and primitives.

Rax: A full-stack Racket to x86_64 nanopass compiler

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.

Hazırcevap (Witty): A closed domain question answering system for high school students

Government funded large scale question answering system. MSc thesis on NLP and Machine Learning. Led 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). Full stack in Python, and JavaScript.

Last compiled on February 28, 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
-