

eric fritz

(608) 774-1120 · eric@eric-fritz.com · eric-fritz.com

software engineer

languages and tools

Go · Python · Rust · Scala · SQL
Postgres · Redis (and the module API) · RabbitMQ · Cassandra · DynamoDB
Git · Docker · Kubernetes · Mesos · AWS

work history

2015 - Now **Mitel** Milwaukee, WI
Currently developing Mitel's IoT infrastructure and IoT Collaboration strategy. Previously developed *Summit* and the surrounding infrastructure for Mitel's communications platform as a service that allows users to build voice and SMS applications with Lua code that runs in a containerized sandbox.

education

2014 - 2018 **Ph.D. Engineering, Computer Science** Milwaukee, WI
University of Wisconsin - Milwaukee
'Waddle - Always-Canonical Intermediate Representation': an optimizing compiler and a supporting set of algorithms whose internal representation never *goes stale*. Local updates to internal structures reduces compilation time while yielding the same output.

2011 - 2013 **M.S. Computer Science** Milwaukee, WI
University of Wisconsin - Milwaukee
'Optimizing the RedPrairie Distance Cache': implemented and evaluated a number of caching solutions for RedPrairie's vehicular route solver using production query data given hard runtime and space constraints. Applying the chosen caching strategy provided a marked improvement in the solver's throughput.

publications

2018	Waddle - Always-Canonical Intermediate Representation	Dissertation
2018	Maintaining Canonical Form After Edge Deletion	ICOOOLPS
2017	Charon: The Design of a Limiting Microservice	Whitepaper, Mitel
2017	Typing and Semantics of Asynchronous Arrows in JavaScript	The Science of Computer Programming
2016	Arrows in Commercial Web Applications	HotWeb
2015	Type Inference of Asynchronous Arrows in JavaScript	REBLS

teaching history

2011 - 2016 **University of Wisconsin - Milwaukee** Milwaukee, WI
Designer and primary instructor of a mobile application programming course using iOS, a compiler implementation laborator, intermediate programming using Java (second course of a three-part series), and a pair of courses focusing on server-side and client-side application development. Ran laboratories and graded for seven additional courses including text retrieval, data structures and algorithms, and assembly language programming (using MIPS).