

Staff Software Engineer at Timescale Inc, specialist in storage systems.

## WORK EXPERIENCE

Timescale Mid-2022 — Present  
*Staff Software Engineer*

### Technical

- Drove company strategy defining function telemetry project, on time, in budget, with negligible customer performance overhead, and 1/10<sup>th</sup> of projected overhead.
- Designed and prototyped columnar execution engine to speed up query performance, projected to provide 4x speedup in the near term with increased projected impact in the next 1-2 years.
- Designed and prototyped numerous successful projects, including cross-group query functionality, script-generation tools, and an embedded sql test runner.
- Improved numerous tech-debt and reliability issues, including fixing access control handling, improvements to build tool and continuous integration suite, workarounds for Postgres serialization issues.
- Built tools analyzing customer behavior, informing company strategic decisions.

### Leadership

- Instituted collaborations between management and engineering to optimize product planning.
- Analyzed customer data to ensure product department alignment with customers.
- Interviewed prospective hires and drove hiring decisions.
- Oriented new hires in product and management inculcating an understanding of our product and corporate positioning.
- Mentored engineers and team leads, hosting weekly, bi-weekly and walk-in meetings providing technical and career advice.
- Drove planning discussion for the company's next 5-year mission.

Timescale 2021 — Mid-2022  
*Software Engineer (Tech Lead, [Toolkit Team](#))*

### Technical

- Implemented libraries and bindings to assist with Rust development of Postgres extensions including, along with other improvements, tools for zero-copy deserialization.
- Designed and implemented function pipelines.
- Designed and implemented Toolkit's continuous integration, versioning, and build suite.
- Implemented and assisted in the implementation of various aggregates for customer use including versions of Hyperloglog, T-Digest, UDDSketch and various statistical functions.

### Leadership

- Founded and led team dedicated to improving DB query UX.
- Served as interim manager for the team, led hiring, salary discussions, performance reviews, and team building.
- Trained engineers on using Rust for software development, and how to interoperate with a large, mature C codebase.
- Mentored engineers and team leads.
- Helped a struggling report pivot into a new role as a company-leading product manager.

Timescale 2021 — 2022  
*Software Engineer ([Observability Team](#))*

- Implemented the IO layer connecting the Promscale data collector to the underlying database.
- Implemented a concurrent CLOCK LRU cache which serves as the basis for Promscale's in-memory caching.
- Implemented epoch-based concurrency control for ID deletion.
- Improved [Promscale](#) runtime and heap size by orders-of-magnitude.

Timescale 2018 — 2020  
*Software Engineer ([Core Database Team](#))*

- Designed and implemented initial Skip Scan support.

- Designed and implemented columnar compression.
- Designed and implemented automated materialized view (part of Continuous Aggregates).
- Designed and implemented license-key system.
- Implemented initial Postgres 12 support.

## EDUCATION

PhD in Computer Science 2013 — 2019

*Yale University*

- Thesis: *Alternative History: Multilog Systems in Theory and Practice*
- Focus in distributed shared log state machine replication.

MS in Computer Science 2013 — 2015

*Yale University*

Bachelor of Arts 2009 — 2013

*Yeshiva University*

- Major: Computer Science, Minor: Mathematics
- Professor Jekuthiel Ginsburg Memorial Award for Excellence in Mathematics and Computers
- Dean's List (all semesters)
- Wilf Family Distinguished Undergraduate Scholarship
- GPA: 3.896, Computer Science and Math GPA: 3.959

## RESEARCH

Distributed Storage Systems 2016 — 2019

- Created *The FuzzyLog, a partially-ordered distributed shared log abstraction*.
- Provides strong consistency, durability, failure atomicity and scalability, and partition-reliance over a simple API.
- Presented in [OSDI 2018](#)

Machine Learning and Distributed Systems. 2016 — 2017

- Internship at Microsoft Research NYC.

Driver Verification 2014 — 2016

- Contributed to the effort to verify the CertiKOS research OS driver stack
- *Toward Compositional Verification of Interruptible OS Kernels and Device Drivers*. PLDI 2016.

Owner of Research OS 2013 — 2016

- Basis of the ground team for the DARPA HACMS Phase II deliverable.
- Sufficiently implemented to run an autonomous vehicle controller.
- Developed CertiKOS Shared Memory communication; primary interprocess and inter-VM communication mechanism for CertiKOS.
- Ported CertiKOS to ARM.

Large-Corpus Text classification Summer 2013

- Bar-Ilan University

## ACADEMIC APOINTEMENTS

Yale Teaching Fellow 2014 — 2015

- Building Decentralized Systems
- Advanced Cloud Computing
- Datastructures and Programming Techniques
- Introduction to Programming

## SKILLS

Programming

- Protocol Design, Concurrency Control, Verification, Debugging, Benchmarking

Programming Languages

- Rust, C, Go, Java, Coq, Python, C#, F# C++