

Kamil Braun

kamilgbraun@gmail.com · github.com/kbr-scylla · github.com/kbr-scylla · linkedin.com/in/kamil-braun
· Warsaw, Poland (open to remote)

Summary

Software engineer with 6.5 years of full-time experience specialising in distributed systems. Strong background in designing and implementing distributed algorithms, database internals, and correctness testing infrastructure. Former team lead and open-source maintainer.

Experience

Software Engineer

poolside.ai

Remote

Feb 2025 – Nov 2025

Worked on the containerization system used for training and evaluating foundation models for programming with reinforcement learning.

Software Engineer — Core Database

ScyllaDB

Warsaw, Poland (remote)

Jan 2019 – Jun 2025

ScyllaDB is a high-performance distributed NoSQL database (Cassandra-compatible), written in C++ on the *Seastar* asynchronous framework.

- **Team lead.** Led a 5-person engineering team. The team was responsible for a large portion of the core database's control plane (mainly schema and topology management).
- **Maintainer.** Part of a small group of engineers with merge authority over the *ScyllaDB* core codebase, responsible for reviewing and gating contributions across multiple subsystems.

Notable contributions include:

- **Change Data Capture (CDC).** Designed and [implemented](#) CDC's support for cluster topology changes.
- **Query performance.** Implemented a [new query algorithm](#) for time-series tables, fixing a read amplification issue that caused unnecessary parallel disk access on every single-partition lookup. Took part in [implementing](#) native efficient handling of reverse-order queries in the storage layer, replacing an approach that required loading entire partitions into memory.
- **Strong consistency initiative (Raft).** Core contributor to *ScyllaDB*'s flagship multi-year project to replace eventually-consistent gossip protocol with the Raft consensus protocol for managing cluster metadata — schema changes, topology operations, authentication and more. Implemented foundational parts including [enforcing linearizability of control plane operations](#), [live zero-downtime cluster migration](#), the [failure detection mechanism](#) underpinning the new system. Designed a procedure for initializing and joining the control plane's Raft group by *ScyllaDB* nodes, [formalizing](#) it in TLA+ (it was later [implemented](#) from the spec).
- **Testing and correctness.** Built a [simulated-network chaos testing framework](#) for the Raft implementation, enabling “e2e” correctness testing without real I/O. This included a linearizability checker that caught multiple real bugs in the consensus logic.

Software Engineering Intern

CodiLime

Warsaw, Poland

Jul 2018 – Sep 2018

Contributed to an open-source Go implementation of [Contrail](#), a network virtualization platform.

Software Engineering Intern

Intel

Gdańsk, Poland

Jul 2015 – Sep 2015

Worked on a research project for analysing software crash reports from end users and automatically determining root causes of failure.

Education

University of Warsaw
M.Sc. Computer Science

Warsaw, Poland
2018 – 2020

Thesis (LPaxos): Designed a novel distributed algorithm for building linearizable services without replicated logs, and implemented it as a key-value store supporting strictly serializable multi-key transactions.

Notable projects include:

- **lattistore** — Key-value store with strictly serializable multi-key transactions, implementing the LPaxos algorithm described in the thesis above. Written in Rust.
- **harddoom2** (coursework) — Linux kernel driver for an artificial PCI graphics accelerator device designed around the Doom 2 game engine. Written in C.

University of Warsaw
B.Sc. Mathematics

Warsaw, Poland
2015 – 2018

University of Copenhagen — Erasmus Exchange
Mathematics (one semester)

Copenhagen, Denmark
2017 - 2018

University of Warsaw
B.Sc. Computer Science

Warsaw, Poland
2013 – 2016

Notable projects include:

- **Motions** (bachelor's project) — Markov Chain Monte Carlo simulation of chromatin movements within a cell nucleus, modelling DNA-protein interactions. Written in Haskell and C++.
- **SIO2** (2014/2015) — Contributed to an open-source platform for running algorithmic programming contests, used in national competitions in Poland. Written in Python (Django).

Skills

Languages (proficient): C++, Python

Languages (familiar): Go, Rust, Java, JavaScript, Haskell, OCaml, C#, Prolog

Programming paradigms: Object-oriented, functional, imperative, logic programming

Areas: Distributed systems, consensus algorithms, database internals, storage engines

Spoken languages: Polish (native), English (professional working proficiency)

Honours & Awards

- Finalist, 20th Polish Olympiad in Informatics — a prestigious national competitive programming competition.