Nathan Timmis

Heidenkellerweg, 8902 Urdorf, ZH, Switzerland
nattimmis@gmail.com · +41 78 208 84 62
linkedin.com/in/nathantimmis · hyperlog.agency/cv/

23 August 2026

Delta DML Team — Executive Engineering
Databricks
Zürich, Switzerland

Senior Staff Software Engineer — Delta (Req. 8303017002)

Dear Hiring Team,

I have spent **over a decade keeping data correct while it moves** — across Kafka clusters, z/OS batch estates and CDC pipelines inside Swiss tier-1 banks — and the recent years building a compiler and a formal verifier, because I wanted those guarantees to be provable rather than asserted. Delta DML is where those two halves meet: transactional correctness on a storage format, at cluster scale, in a place where a wrong answer is far worse than a slow one. That is the problem I would like to work on next.

The migration experience I would bring is attested rather than asserted. On distributed systems at scale, and on your ask for **a multi-year vision delivered incrementally**: at **Raiffeisen Schweiz** I ran Kafka platform engineering on Strimzi / Red Hat AMQ Streams over OpenShift, with named responsibility for **migrating from ZooKeeper to KRaft across every environment** — a consensus-layer replacement that has to land environment by environment, with no window in which the cluster is unavailable and no possibility of losing a committed offset. Alongside it I rotated the entire Kafka certificate estate, upgraded MirrorMaker 2, onboarded **Debezium CDC** with a cluster load assessment, and moved Kafka Streams internal topics off operator-managed CRs onto GitOps definitions. At **Bank Julius Bär** I built and maintained Kafka data flows — Schema Registry, MirrorMaker 2, topic filtering — inside an air-gapped high-security banking environment, and deputised for the team lead.

On **low-level debugging, performance measurement and optimisation on large production clusters**, and on a real foundation in algorithms and data structures: at **UBS** I worked inside Splunk's indexing and search path across a tier-1 monitoring estate, diagnosing ingestion, indexing and access-control failures where the symptom and the cause were almost never in the same subsystem — and more recently, as PL/I developer on z/OS market migrations, I worked the overnight batch estate: JCL job chains, VSAM datasets and their scheduling dependencies, where data quality is the deliverable. I also write the other kind of code. **OverCaml** is a 12,980-line OCaml→LLVM compiler I wrote by hand, with Hindley–Milner type inference: union-find, occurs-check, the actual data structures rather than a library call. Its companion ternary verifier gates my production releases on **idempotency contracts** — not far from the property a MERGE has to hold under retry.

Let me be straight about the gap: I have not shipped Delta Lake or Spark internals, and I would be learning that codebase rather than arriving fluent in it. What I bring instead is eleven years of being the person accountable when a distributed data platform has to change underneath live, regulated workloads and still be provably correct afterwards — plus the compiler and formal-methods background to work **inside** a transaction log rather than around it. On the Staff+ responsibilities beyond code: I led a team of seven engineers in a regulated environment at LKQ Group, led the firewall team at Inselspital Bern, and at UBS led an AI-strategy initiative for mainframe modernisation with an enterprise rollout plan for 80+ engineers. Cross-team projects and mentoring are familiar ground.

I am Swiss, based in Urdorf — twenty minutes from your Zürich office — a native English speaker, and available now. My CV and a live portfolio of production systems are at **hyperlog.agency/cv/**. The reference letters I can send you; the sites you can open from your browser while we talk; and I am happy to build the compiler live on a screen-share. I would welcome the chance to talk about where Delta DML is heading, and where I could carry weight.

Yours sincerely,

Nathan Timmis