# JARED MAHOTIERE

Jared Mahotiere | Bear, DE | (302) 803-7673 | jmahotie@purdue.edu | LinkedIn: linkedin.com/in/jared-mahotiere | GitHub: github.com/jmahotiedu | Site: jmahotiedu.github.io

## SUMMARY

Data platform engineer focused on stream processing semantics and reliability, with 100+ events/sec ingest, checkpointed recovery, and production AWS infrastructure delivery.

## EDUCATION

**Purdue University** - B.S. Electrical Engineering Technology (Computer Engineering Technology)
Minor: Computer & IT | Certificate: Entrepreneurship & Innovation | Expected May 2026

## LEADERSHIP & ORGANIZATIONS

**Delta Tau Delta (Campus Chapter):** DEI Chair | **National Society of Black Engineers (NSBE):** Member

## SKILLS

Languages: Python, SQL | Stream Processing: Kafka, Spark Structured Streaming, Airflow, checkpointing, watermarking, DLQ handling, at-least-once semantics | Datastores: PostgreSQL, SQL Server, Redis | Cloud/Infra: AWS (MSK, S3, ECS, ALB, ECR, RDS, ElastiCache, EMR, MWAA, Redshift), Terraform, Docker, GitHub Actions

## EXPERIENCE

**Nucor Corporation - Software/Automation Engineering Intern | Darlington, SC | May-Aug 2024 and May-Aug 2025**

- Developed and tuned Oracle QMOS + SQL Server query workflows (multi-table joins, priority logic) used by quality, sales, shipping, and mill operations.
- Built the Hold Disposition Management system in Blazor Server (Telerik UI), integrating Oracle QMOS with a SQL Server-backed priority system to centralize hold-status decisions.
- Implemented Quartz.NET automation for weekly hold/priority reports to 4 departments (Mill 1, Mill 2, Saw Cut, Scrap), replacing manual report distribution.
- Led cross-functional validation and rollout planning (test cases, sign-offs, punch-list closure) with operations, maintenance, quality, sales, and shipping for on-time delivery.

## PROJECTS

**IoT Streaming ETL Pipeline - Kafka, PySpark, Airflow, Redshift**

**Links:** GitHub: github.com/jmahotiedu/streaming-etl-pipeline, Architecture: github.com/jmahotiedu/streaming-etl-pipeline/blob/main/docs/ARCHITECTURE.md, Demo: streaming-etl-dash-demo-1722592003.us-east-1.elb.amazonaws.com

- Designed a stream pipeline at 100+ events/sec with sensor_id partitioning, checkpoint-based recovery, watermark handling for late data, and at-least-once ingestion semantics; throughput scaled primarily with Kafka partition count and Spark executor parallelism.

**Event Stream Platform - C#, .NET, WebSocket, WAL, Materialized Views**

**Links:** GitHub: github.com/jmahotiedu/event-stream-platform

- Built WAL-backed ingest and deterministic replay flows with materialized views and backfill/reconciliation tooling for data correctness under retries and restarts.

**workflow-orchestrator - TypeScript, Redis Streams, Postgres**

**Links:** GitHub: github.com/jmahotiedu/workflow-orchestrator, Live: workflow-orc-demo-alb-1577468805.us-east-1.elb.amazonaws.com

- Implemented durable run/task state, idempotent retries, and dead-letter handling for queue-driven orchestration with benchmarked reliability (25/25 runs in 15.94s).

## OPEN SOURCE CONTRIBUTIONS

- PicoClaw (Go): contributed provider protocol-family refactor work (PR #213, related to roadmap #283) and device-code auth interval parsing fix (PR #56); PicoClaw + Bloomberg PR history: https://github.com/pulls?q=is%3Apr+author%3Ajmahotiedu+org%3Asipeed+org%3Abloomberg.
- Bloomberg comdb2 (C/C++/SQL): backported targeted SQLite security fixes in PR #5743 with source-build validation and verification matrix.
- Open contributions: Databricks CLI auth-resolution fix (#4504) and Google langextract cache-key hashing fix (#359).