Nils Matteson

linkedin.com/in/nilsmatteson nilsmatteson.com nilsmatteson@icloud.com Madison, WI

Education

University of Wisconsin–Madison Madison, WI

B.S. Data Science, Minor in Computer Science *Expected May 2026*

- **Relevant Coursework**: Artificial Intelligence (CS 540), Machine Organization (CS 354), Data Structures & Algorithms (CS 400), Discrete Mathematics (MATH 240), Linear Algebra (MATH 340), Statistical Modeling (STAT 340).

Technical Skills

Languages: Python, Rust, Go, C++, TypeScript/JavaScript, SQL
ML & AI: PyTorch, Transformers, LLMs, Stable Diffusion, Scikit-learn, Hugging Face, OpenCV
Systems & Cloud: Distributed Systems, gRPC, Docker, Kubernetes, AWS, GCP, Redis, Kafka, PostgreSQL
Web & Graphics: React, Next.js, WebSockets, Three.js, WebGL, GLSL Shaders

Research Experience

Research Cyberinfrastructure, UW–Madison DoIT Madison, WI

AI Workflows Research Collaborator *Jan 2026 – Present*

- Collaborating on **WattBot RAG Playground**: deploying KohakuRAG (1st place Kaggle solution, 0.861 score) as a Streamlit-based research service for AI energy estimation queries.
- Building interactive chat interface wrapping hierarchical RAG pipeline with **multi-query retrieval**, cross-query reranking, and ensemble inference with abstention-aware voting.
- Architecting AWS deployment using Bedrock for LLM inference, S3 for pre-built SQLite vector indices, with cost optimization and on-prem GPU benchmarking (GB10).

Selected Projects

Sentinel: Distributed Log Streaming Engine *Go, gRPC, Protobuf, LSM Trees, Raft Consensus*

Production-grade distributed message queue architected from scratch — 5,600+ lines of systems code.

- Engineered custom LSM-tree storage engine with skip list memtable achieving **1.7M writes/sec** and **3.9M reads/sec**; CRC32 checksums, bloom filters, and crash-safe write-ahead logging.
- Built Raft consensus layer for fault-tolerant leader election and log replication with randomized timeouts, AppendEntries RPC, and majority-quorum commit protocol.
- Designed gRPC streaming API with Kafka-style topic/partition semantics, consumer groups, offset tracking, and Prometheus-compatible metrics (p50/p95/p99 latency).

Madison Metro: Autonomous ML Bus Prediction System *Python, XGBoost, Flask, PostgreSQL, React, GitHub Actions*

End-to-end MLOps pipeline with automated retraining, model registry, and live inference serving.

- Architected data pipeline ingesting **50K+ daily GTFS-RT observations** via polling daemon with data validation, deduplication, and schema enforcement before PostgreSQL feature store ingestion.
- Built autonomous ML pipeline: nightly XGBoost on 14-day sliding window, **automated A/B comparison** (deploy if F1 >1% improvement), immutable model versioning with lineage tracking.
- Deployed Flask inference API with **sub-50ms P95 latency**; feature engineering at request time (temporal encoding, route embeddings, delay aggregations) with request logging for drift monitoring.
- Shipped React dashboard with real-time bus tracking, delay heatmaps by route/time, model performance metrics (accuracy, F1, precision/recall), and training run history; CI/CD via GitHub Actions.

Synapse: Real-time Collaborative Whiteboard *Rust, WebAssembly, WebSockets, CRDTs (Yjs), Redis Pub/Sub*

Lock-free distributed canvas supporting 50+ concurrent editors with sub-100ms sync latency.

- Built Rust WebSocket server (Actix) with CRDT state sync (Yjs) for conflict-free concurrent editing without operational transforms or central locking.
- Compiled rendering to WebAssembly achieving **60 FPS** with 10K+ vector objects; R-tree spatial indexing for viewport culling.
- Architected horizontal scaling via Redis Pub/Sub with connection draining and session affinity for zero-downtime deployments.