Hyperlogistics: Smart Supply Chain System Optimization System

CS 5542 Big Data Analytics & Applications – Project Proposal

Daniel Evans (Data/Back-End Engineer)
Joel Vinas (Data/Engineer Engineer)
Tony Nguyen (ML/Full-Stack Engineer)

February 9th, 2026

Abstract

This proposal outlines the design and implementation of an **Intelligent Middle-Mile Optimization System**, an AI-driven framework designed to mitigate shipment delays and disruptions in the most complex segment of the modern supply chain. Middle-mile logistics currently suffer from fragmented data sources and a reliance on static routing algorithms that fail to adapt to emergent real-time conditions like inclement weather or infrastructure failures. Leveraging **Snowflake** as the core data engine and **Retrieval-Augmented Generation (RAG)** for historical context, our system translates real-time disruption signals into proactive, explainable re-routing recommendations. By integrating recent advancements from **NeurIPS 2025**, including semantic abstraction and expert schema linking, the solution addresses the "lack of trust" bottleneck where dispatchers often override algorithmic suggestions due to a lack of context. The project features a **Streamlit in Snowflake** interface for real-time interaction, **Snowflake Cortex** for secure LLM inference, and a rigorous evaluation framework utilizing a synthetic "Golden Dataset" of disruption scenarios. This system transforms middle-mile management from a reactive, manual process into a proactive, grounded, and transparent business utility.

Contents

1. Introduction

The modern supply chain is currently facing unprecedented pressure to increase demand for one-hour delivery, same-day delivery, and rapid fulfillment cycles. While significant investment has been poured into last-mile delivery, the middle-mile delivery remains a significant fragility point. Delays and disruptions in these segments, ranging from inclement weather to infrastructure failures, accounted for up to $95 billion in losses per year McKinsey & Company (2024).

A critical barrier in current middle-mile management is the reliance on static routing algorithms and fragmented data sources. When a disruption occurs, logistics managers are often forced to manually re-route hundreds of shipments using "disconnected" information that is late or incomplete. Furthermore, even when advanced algorithmic suggestions are available, they often lack **explainability**, leading dispatchers to override recommendations they do not fully trust.

This project proposes an **Intelligent Middle-Mile Optimization System**, a robust, Snowflake-native solution designed to transform reactive logistics into a proactive, grounded business utility. By leveraging a **Retrieval-Augmented Generation (RAG)** architecture specialized for logistics metadata, our system allows dispatchers to interact with complex supply chain data using natural language.

2. Objectives

2.1 Real-World Problem

In modern, data-driven logistics, the ability to maintain middle-mile fluidity is hindered by a critical gap between raw data availability and actionable intelligence. While organizations collect vast amounts of telematics and environmental data, the "intelligence" remains concentrated in manual intervention, creating a significant bottleneck during disruptions. This problem is compounded by three specific systemic failures

1. **Fragmented Data Silos (The Information Gap):** Weather data, traffic incidents, and delivery performance metrics are not unified. We need a way to correlate disruptive events with their impact on downstream routes.

2. **Slow Reaction Time (The Time Gap):** Current systems respond to disruptions after they occur (i.e., a disruptive event causes a tangible disruption). Dispatchers must react to this instead of taking proactive/precautionary measures. Because they cannot easily provide models with secure access to the full database schema in real-time.

3. **Lack of Explainability (The Explainability Gap):** Algorithmic routing decisions lack explainability and context. If a system suggests rerouting a shipment through a longer path without a natural language explanation grounded in business logic, dispatchers may not trust the recommendation and can manually override it, which may be less efficient.

2.2 Target Users

- **Primary Users:** Logistics Network Controllers and Area Managers who need to make immediate decisions on hub-to-hub transfers without manually querying fragmented data.

- **Secondary Users:** Supply Chain Analysts who can use the system to draft complex simulation scenarios.

2.3 Innovation Layer

The system introduces a **neuro-symbolic agentic architecture** to bridge the gap between AI reasoning and industrial safety requirements:

- **Neuro-Symbolic Guardrails:** Implements a dual-check mechanism where LLM-suggested detours are cross-referenced against a "Symbolic" layer of **Spatial SQL** queries. This ensures 100% compliance with **US DOT bridge and height clearances,** acting as a hard-coded safety veto.

- **Spatio-Temporal Disruption Patching:** Leverages **SRSNet** (NeurIPS 2025) to adaptively "patch" historical accident and weather data. This allows the system to

predict how current disruptions will propagate across the middle-mile network over a 4–8 hour window.

- **Explainable Multi-Agent Consensus (CPP):** Utilizes the **Consensus Planning Protocol** where three specialized agents (Context, Efficiency, and Compliance) must negotiate a final plan. This eliminates the "lack of trust" bottleneck by providing dispatchers with transparent, grounded reasoning for every re-route.

- **Snowflake-Native RAG Orchestration:** Executes all vectorization and LLM inference directly within **Snowflake Cortex**. This preserves enterprise-grade data governance while providing a "Reasoning Path" grounded in real-time environmental and infrastructure metadata.

3. Related Work (NeurIPS 2025 & Logistics Foundations)

Our system design is directly informed by three cutting-edge papers that address the specific challenges of fragmented data silos, slow reaction time, and lack of explainability in middle-mile logistics

3.1 Selective Representation for Forecasting (SRS)

- **Paper:** [Enhancing Time Series Forecasting through Selective Representation Spaces: A Patch Perspective](#).

- **Code:** [github.com/decisionintelligence/SRSNet](#).

- **Summary:** This research proposes SRS, a technique that adaptively selects and reshuffles useful patches of data in a time series model. This technique boosts forecasting accuracy in the long-term pattern detection system, improving existing patch-based models.

- **Project Integration/Explanation:** Reliably predicting weather events as they relate to logistics become more difficult as the scope of time increases. By leveraging the remarkable work in adaptively selecting and reshuffling useful patches of data, we can enable our time series model to remain flexible with the adjacent patch while being able to extend to longer-term weather pattern forecasts for middle-mile planning. We adopt the "SRS" philosophy to enable our Snowflake-native models to remain flexible with real-time telematics.

3.2 Knowledge-Guided Retrieval (ReMindRAG)

- **Paper:** [ReMindRAG: Low-Cost LLM-Guided Knowledge Graph Traversal for Efficient RAG](#).

- **Code:** github.com/kilgrims/ReMindRAG.

- **Summary:** This research introduces REMINDRAG, a method which uses LLM-guided knowledge graph traversal with memory replay in order to boost performance, reducing token usage and inference time while retaining accuracy.

- **Project Integration/Explanation:** To search the numerous data sources on current weather events, our tool will need to have improved performance and reduced token usage while retaining accuracy. REMINDRAG could be used as a part of the RAG process to ensure quality with less computation and to provide the dispatcher with high-quality, explainable justification, solving the "lack of explainability"

3.3 Multi-Agent Consensus Planning (CPP)

- **Paper:** Consensus Planning with Primal, Dual, and Proximal Agents (Maggiar et al., 2024).

- **Code/Proxy**: github.com/hubbs5/or-gym

- **Summary:** This research details the **Consensus Planning Protocol (CPP)**, defining how different supply chain agents (e.g., a Buying System vs. a Placement System) negotiate to agree on a single plan. This is particularly effective for handling large-scale disruptions where capacity constraints must be balanced across multiple supply chain nodes.

- **Project Integration/Explanation:** Our architecture adopts this "multi-agentic" flow. The system utilizes a "Validation Agent" to negotiate between the LLM's suggested route and physical constraints (bridge weight, driver hours) before the final recommendation is rendered in Streamlit UI.

4. Data Sources

4.1 Supply Chain Logistics Dataset (Ground Truth)

- **Source (Kaggle):** DataCo Smart Supply Chain for Big Data Analysis

- **Description:** This dataset contains 180k+ rows of structured logistics data, including scheduled vs. actual delivery times, shipping modes, and order status (late delivery risk). The modality is tabular (CSV), representing a static historical snapshot of supply chain operations. It is ideal for training the "Selective Representation" (SRS) model to detect patterns in middle-mile delays.

- **Snowflake Integration:**

  - Ingestion: Uploaded via a Snowflake Internal Stage (@LOGISTICS_STAGE).

- o Storage: Loaded into a structured RAW_LOGISTICS_STRIPS table using the COPY_INTO command

- o Processing: We will Snowpark (Python) to perform feature engineering, specifically calculating "Delay Propensity" scores that will be stored as materialized view for the RAG historical context

.

4.2 Global Weather & Natural Disaster Feed (Disruption Signals)

- **Source (AWS Open Data) :** [NOAA Global Surface Summary of the Day (GSOD)](#) / [AWS Open Data](#)

- **Description:** A massive multi-terabyte environment dataset. The modality is time-series geospatial data. It is updated daily, providing real-time weather signals (winds, precipitation, visibility) that acts a primary disruption trigger in our system.

- **Snowflake Integration:**

  - o **Ingestion:** Connected as a **Snowflake External Table** pointing directly to the public S3 bucket, avoiding unnecessary storage costs.

  - o **Storage:** Metadata is indexed in Snowflake to allow rapid partitioning by data and geospatial coordinates.

  - o **Processing: Snowflake Cortex** functions will be used to extract semantic "weather alerts" from these logs, which are passed to the LLM to justify re-routing decisions.

4.3 US DOT National Transportation Atlas (Infrastructure Constraints)

- **Source (:** [Bureau of Transportation Statistics - National Tunnel & Bridge Inventory](#)

- **Description:** This dataset contains records for over 600,000 bridges in the US. The modality is Geospatial/Tabular, updated **annually**. It includes critical physical constraints like load limits (bridge weight) and vertical clearances.

- **Snowflake Integration:** This data will be stored in a **Geometry/Geography data type** within Snowflake. It will serve as the "Validation Agent" layer (CPP model) to ensure that when the LLM suggests a re-route due to weather, the new path is

  - o **Ingestion:** Ingested via **Snowpipe** from an S3 bucket containing GeoJSON/CSV exports.

  - o **Storage:** Stored in a table utilizing the **GEOGRAPHY** data type to enable high-performance spatial joins.

- o **Processing:** During the "Consensus Planning" phase, the system runs a **Spatial SQL query** to validate that a suggested re-route is physically compatible with the vehicle's dimensions, acting as a hard-coded safety guardrail.

4.4 US Accidents (2016 - 2023)

- **Direct Dataset Link:** US Accidents (2016 - 2023)

- **Brief Description:** This is a countrywide traffic accident dataset covering the contiguous United States. It contains **7.7 million records** (**3.06 GB**) in CSV format. Each record includes precise GPS coordinates, start/end times, and severity levels (1-4), along with proximity to "Points of Interest" like crossings or junctions.

- **Snowflake Integration:**

  - o **Ingestion:** Uploaded via **Snowflake Internal Stage** using the Python-based snowflake-ingest SDK for efficient chunking.

  - o **Storage:** Stored in a TRAFFIC_INCIDENTS table. We will apply **Clustering Keys** on the Start_Time and State columns to ensure rapid temporal and spatial filtering during real-time inference.

  - o **Processing:** We will utilize **Snowpark** to create a "Risk Heatmap" view. This allows the **Validation Agent** to cross-reference a suggested middle-mile route against historical accident "blackspots" during specific times of day.

5. Methods, Technologies & Tools

5.1 System Architecture

The architecture follows a sequential flow from raw data ingestion to an interactive dispatcher dashboard:

- **Backend:** RAG pipeline for weather/logistics metrics extraction.

- **Middleware:** Traffic condition forecasting using **DeepAR/SRSNet**.

- **Frontend:** Natural language query processor and route weighting engine.
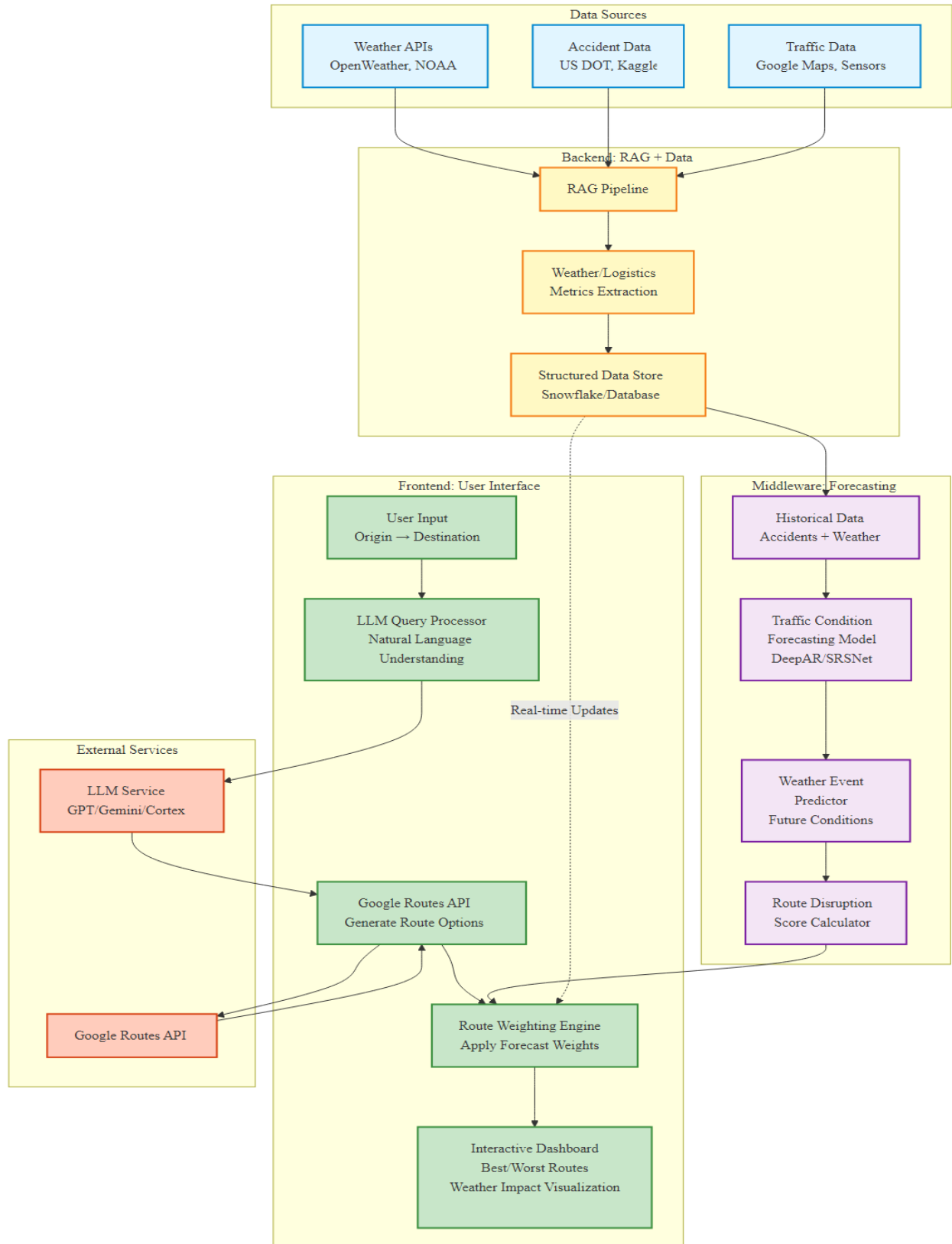
Figure 1: System Architecture

5.2 Data Ingestion & Storage

The system employs a hybrid ingestion strategy to balance the massive scale of historical datasets with the urgency of real-time disruption signals.

- **Ingestion Mechanisms:**

    - **Snowpipe & Streams:** Used for the US Weather Events and Traffic Data. As new files land in the S3 staging area, **Snowpipe** automatically triggers ingestion, while **Snowflake Streams** track change data capture (CDC) for downstream processing.

    - **External Tables:** Large-scale, frequently updated datasets like **NOAA GSOD** are accessed via External Tables to minimize storage costs while maintaining high-performance query capabilities.

    - **Batch Ingestion:** Historical datasets such as the DataCo Supply Chain and US Accidents are bulk-loaded via Snowflake Internal Stages using optimized COPY INTO commands.

- **Data Modeling & Governance:**

    - **Medallion Architecture:** Data is organized into **Bronze** (Raw Landing), **Silver** (Cleaned/Standardized), and **Gold** (Analytics/LLM-Ready) zones.

    - **Governance:** We implement **Row-Level Security (RLS)** and **Column-Level Masking** to simulate enterprise-grade data privacy, ensuring the LLM only accesses data relevant to the specific user's authorized region or role.

5.3 Analytics, ML & GenAI

The core "intelligence" of the system is built using the **Snowflake Cortex** and **Snowpark** ecosystem.

- **Processing Framework (Snowpark):**

    - **Python/SQL:** We use Snowpark to execute complex feature engineering and the SRSNet (Selective Representation) logic directly within Snowflake's elastic compute. This eliminates the need for data regression, significantly reducing latency for time-series forecasting.

- **LLM & RAG Pipelines:**

    - **Snowflake Cortex:** Utilized for secure LLM inference (e.g., Llama 3 or Mistral) to translate natural language dispatcher queries into SQL/Vector searches.

    - **Vector Search:** We use Snowflake's native **VECTOR** data type and **VECTOR_L2_DISTANCE** functions. The system creates embeddings for

historical "Disruption Reports" to allow the RAG pipeline to retrieve "similar past scenarios" during an active event.

- **Innovation Layer: Neuro-Symbolic Validation:**

  o A custom validation layer combines the LLM's "creative" routing suggestions with "symbolic" rule-checking. For example, if the LLM suggests a detour, a **Snowpark Python UDF** cross-references the **DOT Bridge Inventory** to ensure the path is physically viable for the vehicle's height and weight constraints.

5.4 Infrastructure & Deployment

The system is designed to be a "single-pane-of-glass" utility for logistics managers.

- **Dashboard & Frontend:**

  o **Streamlit in Snowflake:** The primary interface is a Streamlit application hosted directly within the Snowflake environment. This provides a low-latency, interactive dashboard where users can visualize best/worst routes and weather impact overlays.

- **Deployment & Cloud Services:**

  o **Cloud Provider:** The infrastructure is hosted on **AWS**, leveraging S3 for raw data staging and Snowflake for the unified data and AI platform.

- **Monitoring & Evaluation:**

  o **Evaluation Pipeline:** We utilize a synthetic **"Golden Dataset"** of known disruption scenarios to measure the RAG system's accuracy and the LLM's adherence to safety constraints.

  o **Logging:** All dispatcher interactions and system recommendations are logged into a **Snowflake Event Table** for auditability and continuous model improvement.

6. Implementation Plan (4-Week Sprint)

We have structured the implementation into four one-week sprints, aligning with a **Sequential Agentic Flow** that transitions from raw data perception to intelligent planning and execution.

Four-Week Implementation Plan

- **Week 1: Data Foundations & Perception Agent.**

  o Set up the Snowflake account, Medallion schemas (Bronze, Silver, Gold), and GitHub repository for team collaboration.

- Ingest the **DataCo Logistics**, **US Accidents**, and **US Weather** datasets using Snowflake Internal Stages and Snowpipe.

- Configure **External Tables** for the NOAA GSOD multi-terabyte feed to enable cost-effective, real-time "Perception Agent" monitoring.

- Define "Semantic Views" and table descriptions to prepare metadata for the RAG-driven Schema Linker.

- **Week 2: RAG Pipeline & Reasoning Agent.**

  - Implement a **Vector Store** within Snowflake using the VECTOR data type to store historical weather impact reports.

  - Develop the **"Reasoning Agent"** utilizing **Snowflake Cortex** to translate dispatcher queries into grounded re-routing requests.

  - Build the core **Knowledge-Guided Retrieval (ReMindRAG)** architecture to search schema metadata and provide explainable justifications for suggested route changes.

- **Week 3: Frontend & Predictive Analyst Agent.**

  - Implement the **"Predictive Analyst Agent"** using **Snowpark (Python)** to deploy the **SRSNet** time-series forecasting model.

  - Build the primary **Streamlit in Snowflake** interface, featuring interactive maps and a natural language "Ask the Agent" sidebar.

  - Integrate visualization logic to render real-time **Route Disruption Scores** and weather impact overlays for dispatchers.

- **Week 4: Safety Guardrails & Final Evaluation.**

  - Develop the **"Planning Agent"** (Consensus Planning Protocol) to run Neuro-symbolic validation checks against the **US DOT Bridge Inventory**.

  - Generate a synthetic **"Golden Dataset"** of middle-mile disruption scenarios to benchmark system accuracy, latency, and trust metrics.

  - Finalize the PDF proposal, technical documentation, and the live dashboard demo for deployment.

7. Risk Analysis

Given the complexity of a multi-agentic logistics system, we have identified four primary risk categories. Each risk is evaluated based on its impact on the "Middle-Mile" operation and paired with a specific mitigation strategy leveraging our Snowflake-native stack.

7.1 Data Integrity and Latency Risk

- **Risk:** Real-time disruption signals (weather/accidents) may suffer from ingestion latency, leading to "stale" routing recommendations that send drivers into already closed corridors.

- **Impact:** High. Operational delays and potential safety hazards.

- **Mitigation:** We utilize **Snowpipe** for continuous micro-batch ingestion of traffic feeds and **Snowflake Dynamic Tables** to ensure that the "Gold" analytics layer is refreshed with sub-minute latency.

7.2 The "Explainability Gap" & User Trust

- **Risk:** Dispatchers may ignore the system's "Intelligent" re-routing if the LLM provides generic advice or fails to cite specific historical data (the "Lack of Trust" bottleneck).

- **Impact:** Medium/High. System abandonment and continued reliance on manual, reactive processes.

- **Mitigation:** By integrating **ReMindRAG** (NeurIPS 2025), the system is forced to ground every recommendation in specific historical records from our **US Weather Events** and **Accident** datasets. The Streamlit UI will explicitly display the "Reasoning Path" (e.g., "Rerouting due to 85% probability of icing at I-80 based on 2022 patterns").

7.3 Physical Feasibility and Compliance

- **Risk:** The LLM Reasoning Agent might suggest an efficient path that is physically impossible for a heavy-duty hauler (e.g., a low-clearance bridge or a weight-restricted rural road).

- **Impact:** Critical. Risk of bridge strikes, vehicle damage, and legal liability.

- **Mitigation:** We implement a **Neuro-Symbolic Guardrail**. Before any route is finalized, the **Planning Agent** runs a mandatory **Spatial SQL join** against the **US DOT National Bridge Inventory**. This acts as a "hard-coded" veto that overrides the LLM if physical constraints are violated.

7.4 Model Hallucination in Big Data

- **Risk:** When querying millions of rows across fragmented schemas, the LLM may hallucinate non-existent correlations between weather patches and delivery performance.

- **Impact:** Medium. Inefficient routing decisions.

- **Mitigation:** We utilize **Selective Representation (SRS)** to limit the time-series forecasting to relevant "patches" of data, reducing the noise the LLM has to process. Additionally, the **Consensus Planning Protocol (CPP)** ensures that three different agents must "agree" on the data validity before a recommendation is rendered.

Risk Category	Probability	Severity	Primary Mitigation Tool	Impact Logic
Data Latency	🟢 Low	🔴 High	Snowpipe / Dynamic Tables	Ensures real-time disruption signals from NOAA and Traffic feeds are processed with sub-minute latency.
Trust / Explainability	🔴 High	🟡 Medium	ReMindRAG / Cortex Search	Grounding LLM outputs in historical weather/accident records to eliminate the "black box" algorithmic bottleneck.
Compliance / Safety	🟢 Low	🟣 Critical	DOT Bridge Inventory / Spatial SQL	Uses a "Hard-Veto" logic; reroutes are automatically discarded if vehicle clearances violate US DOT infrastructure limits.
LLM Hallucination	🟡 Medium	🟡 Medium	SRSNet / Multi-Agent Consensus	Prevents false correlations by limiting time-series forecasting to validated "patches" of logistics metadata.

Figure 2: Risk Heatmap Summary

8. Expected Outcomes & Evaluation Metrics

8.1 Evaluation & Metrics

To prove the project is a "research prototype," we will evaluate the system across three dimensions: **Technical Accuracy**, **Operational Safety**, and **Explainability Trust**.

8.1.1 Performance Benchmarking

We will utilize the **"Golden Dataset"**—a curated set of 50 extreme disruption scenarios (e.g., a "Level 4" accident on I-70 during a "Severe Snow" event)—to measure the system's effectiveness.

- **Routing Efficiency:** We compare the "System Suggested Route" against the "Static Original Route" and a "Random Reroute" to measure the percentage reduction in predicted delay time.

- **Latency (End-to-End):** Time taken from a disruption signal appearing in the **Bronze Layer** to a re-routing recommendation appearing in the **Streamlit Dashboard**.

- **Vector Retrieval Precision:** Measuring the relevancy of the historical context retrieved by the **ReMindRAG** agent during specific weather events.

8.1.2 Safety & Compliance

This metric ensures the system never compromises physical safety for speed.

- **Constraint Adherence:** A 100% success rate requirement for the **Planning Agent**. Any route that violates a US DOT bridge clearance or weight limit is logged as a "Critical Failure."

- **Near-Miss Prediction:** Assessing the system's ability to flag high-risk zones (e.g., yards with high historical accident density during specific visibility levels).

8.1.3 Quantitative Metrics Table

Metric Category	Specific KPI	Target Benchmark	Evaluation Method
Logistics	Delay Mitigation	> 15% Reduction	Comparison against Static Routing
Safety	Bridge Strike Risk	0% Violations	Spatial SQL vs. DOT Inventory
AI Quality	RAG Grounding	> 90% Accuracy	Manual review of "Golden Dataset" logs
System	End-to-End Latency	< 30 Seconds	Snowpipe → Cortex → UI Timestamp

Figure 3: Quantitative Metric Table

8.2 Expected Outcomes

Upon completion, **Hyperlogistics** will deliver a working prototype that demonstrates:

- **A Live Streamlit Dashboard:** Visualizing middle-mile routes with real-time risk overlays from the **US Accidents** and **Weather** datasets.

- **An Autonomous Agentic Flow:** A system that "thinks" through disruptions, shifting from reactive manual re-routing to proactive, grounded suggestions.

- **Explainable AI:** A natural language interface that doesn't just give a route, but explains *why* (e.g., *"Re-routing to avoid visibility under 0.5 miles at the KCMO hub"*).

9. Final Deliverables

The project will culminate in a comprehensive suite of artifacts that demonstrate technical depth, operational safety, and research alignment.

9.1 Software Prototype & Production Dashboard

- **Live Streamlit Assistant:** A functional application hosted in Snowflake. This assistant will feature:

  - **Real-time Risk Map:** Interactive visualization of active shipments layered with NOAA weather alerts and accident heatmaps.

  - **Natural Language Interface:** A chat-based assistant powered by **Snowflake Cortex** that provides explainable re-routing justifications.

  - **Reroute Comparison Tool:** A side-by-side view comparing the original static route vs. the AI-optimized "Safety-First" route.

9.2 Technical Implementation Repository

- **Snowpark & SQL Pipelines:** A complete codebase for the Medallion architecture (Bronze, Silver, and Gold layers), including:

  - **The Ingestion Layer:** Snowpipe and Task scripts for automated data streaming.

  - **The ML Engine:** Snowpark Python notebooks implementing the **SRSNet** time-series forecasting and **CPP** multi-agent negotiation logic.

  - **The RAG Core:** Vector embedding generation and **Cortex Search** service definitions.

- **GitHub Repository:** A professionally documented repository containing the full source code, a reproducibility/ folder, and an environment setup guide.

9.3 Research & Evaluation Report

- **Technical Design Paper:** A formal report on the system architecture and the integration of **NeurIPS 2025** papers detailing our design decisions, challenges, and evaluation results.

- **Benchmark Results:** A performance audit based on the **"Golden Dataset,"** quantifying improvements in delay mitigation and safety compliance.

9.4 Data & AI Artifacts

- **Vectorized Knowledge Base:** The processed and embedded historical weather/accident reports used to ground the RAG pipeline.

- **Safety Logic Schema:** The "Symbolic" logic layer (SQL/Geography joins) that acts as the hard-coded compliance check against the US DOT Bridge Inventory.

References

Research Papers (NeurIPS 2025 & Logistics Foundations)

- **SRSNet:** Wu, X., Qiu, X., et al. (2025). *Enhancing Time Series Forecasting through Selective Representation Spaces: A Patch Perspective*. arXiv preprint arXiv:2510.14510. URL: https://arxiv.org/abs/2510.14510

- **ReMindRAG:** Hu, Y., Zhu, J., et.al. (2025). *ReMindRAG: Low-Cost LLM-Guided Knowledge Graph Traversal for Efficient RAG*. arXiv preprint arXiv:2510.13193. URL: https://arxiv.org/abs/2510.13193

- **CPP:** Maggiar, A., Dicker, L., & Mahoney, M. W. (2024). *Consensus Planning with Primal, Dual, and Proximal Agents*. Amazon Science. URL: https://www.amazon.science/publications/consensus-planning-with-primal-dual-and-proximal-agents

Datasets

- **Supply Chain Logistics:** Constante, F. (2019). *DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS* [Dataset]. Mendeley Data. URL: https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis

- **US Traffic Accidents:** Moosavi, S., Samavatian, M. H., Parthasarathy, S., & Ramnath, R. (2019). *A Countrywide Traffic Accident Dataset*. In Proceedings of the 27th ACM SIGSPATIAL. URL: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

- **Infrastructure Constraints:** Bureau of Transportation Statistics. (2025). *National Tunnel & Bridge Inventory* [Dataset]. U.S. Department of Transportation. URL: https://geodata.bts.gov/datasets/national-bridge-inventory/

- **Weather Disruptions:** National Oceanic and Atmospheric Administration (NOAA). (2025). *Global Surface Summary of the Day (GSOD)* [Dataset]. AWS Open Data Registry. URL: https://registry.opendata.aws/noaa-gsod/