Introduction

"Hello, my name is [Your Name], and I'm presenting our project titled *'Reimplementing a Multi-Agent Framework for Modeling Supply Chain Dynamics'*. This work is based on the influential framework proposed by Swaminathan, Smith, and Sadeh in 1996."

Motivation & Background

"Modern supply chains are under increasing pressure. Events like the COVID-19 pandemic, material shortages, and port congestion have highlighted how vulnerable global logistics can be.

Traditional, centralized decision-making systems often fail in such dynamic environments. This is where **Multi-Agent Systems**, or MAS, come in. These systems model supply chains as networks of independent agents—like manufacturers, distributors, and retailers—each with local goals and decision-making abilities. These agents interact, share information, and adapt over time.

In 1996, Swaminathan et al. introduced one of the earliest reusable MAS frameworks for supply chain modeling. It allowed researchers to simulate and evaluate trade-offs such as **inventory vs. serviceability** or **standard distribution vs. cross-docking** using modular agents and control policies.

Despite its practical relevance—IBM even used elements of it—the original implementation details are sparse, making reproduction difficult. That's the motivation for our work."

Project Goals

"Our project has three core goals:

1. **Rebuild the original multi-agent framework**, implementing agents like retailers, manufacturers, and transportation.
2. **Replicate two key experiments** from the paper: one on cross-docking and another on inventory-service trade-offs.
3. **Compare results**, analyze deviations from the original findings, and explore sources of discrepancy such as modeling assumptions or missing parameters."

System Design Overview

"The original paper defines two types of framework elements:

- **Structural agents**: these include retailers, manufacturers, distributors, and transportation vehicles.
- **Control elements**: these define policies for inventory, supply, demand, routing, and information flow.

Each agent communicates via discrete-event messages. These messages fall into three categories:

- **Material flows** (e.g., shipping goods),
- **Information flows** (e.g., forecast updates or order requests), and
- **Cash flows** (e.g., payment confirmations).

The agents are simulated over time using a discrete-event simulation engine. Each agent has its own performance metrics and can update its internal state based on incoming messages and local policies.

This modular setup allows us to reconfigure the supply chain and test various scenarios by just plugging in or modifying agent behavior."

Agent Implementation

"We've implemented the following agent types, closely following the original architecture:

- **Retailers**: fulfill customer demand using local inventory, reorder based on control policies.
- **Distributors**: in both standard and cross-docking variants.
- **Manufacturers**: produce goods in batches with lead times.
- **Transportation agents**: manage logistics between locations.
- **Customer agent**: generates demand stochastically or periodically.

Each agent's decisions are governed by a set of control elements—for example, when inventory drops below a threshold, a base-stock policy may trigger a new order. These interactions are modeled using asynchronous message passing."

Control Elements

"The paper defines several types of control elements:

- **Inventory control**: includes base-stock, (s, S), and MRP policies, centralized or decentralized.
- **Demand control**: allows demand to be random or forecasted.
- **Supply contracts**: define lead times, flexibility, and cost penalties.
- **Routing and loading policies**: determine logistics behavior.
- **Information control**: agents can share data in real-time or through periodic updates.

These policies let us replicate the exact configurations used in the original paper—and also try alternative designs."

Evaluation Metrics

"To compare our results with the original, we use four quantitative metrics:

1. **Fill Rate** – percentage of customer demand fulfilled on time.
2. **Average Inventory** – the mean inventory across all agents.
3. **Average Lead Time** – the time between order and delivery.
4. **Total Cost** – the sum of holding and transportation costs.

Each experiment will be run **50 times** to ensure statistical reliability, and results will include **95% confidence intervals**."

Cross-Docking vs. Standard Distribution

"The first experiment evaluates whether using a **cross-docking strategy** improves supply chain performance.

In standard distribution, inventory is stored at the distribution center. In **cross-docking**, goods arrive and are shipped out immediately—without storage—based on destination.

The goal is to reduce inventory costs, but it introduces risks: higher reliance on timing and potentially longer lead times.

We're replicating this experiment by simulating both configurations and comparing the impact on inventory levels and fill rates."

Inventory–Serviceability Trade-Off

“The second experiment focuses on the trade-off between **inventory levels** and **customer service**.

By adjusting base-stock levels, we analyze how increasing inventory improves the fill rate—until diminishing returns set in.

This is key for decision-makers: How much inventory is ‘enough’ to balance cost and performance?

We’ll run this scenario with varying base-stock levels and plot curves for fill rate vs. inventory to mirror the results in the original paper.”

Next Steps

“Over the coming week, our focus will be on:

- Finalizing the agent behaviors and control elements.
- Running both experiments with consistent configurations.
- Comparing our outputs—fill rates, inventory, lead times—to those in the original study.
- Analyzing any mismatches in terms of underlying assumptions or randomness.
- Preparing a detailed final report and demonstration video showcasing the simulator in action.”

Conclusion

“To wrap up:

Our project revisits a foundational multi-agent framework and tests its **reproducibility**—which is a core issue in modern computational science.

We aim not only to replicate key findings but also to make the framework easier to reuse and extend for future supply chain research.

Ultimately, this project reinforces the value of modular, agent-based modeling in designing resilient and adaptive supply chains—something that has never been more important than today.

Thank you for listening.”