MULTI-AGENT Architecture

This is a conceptual multi-agent system designed to show how various components could interact in a production-level setup.

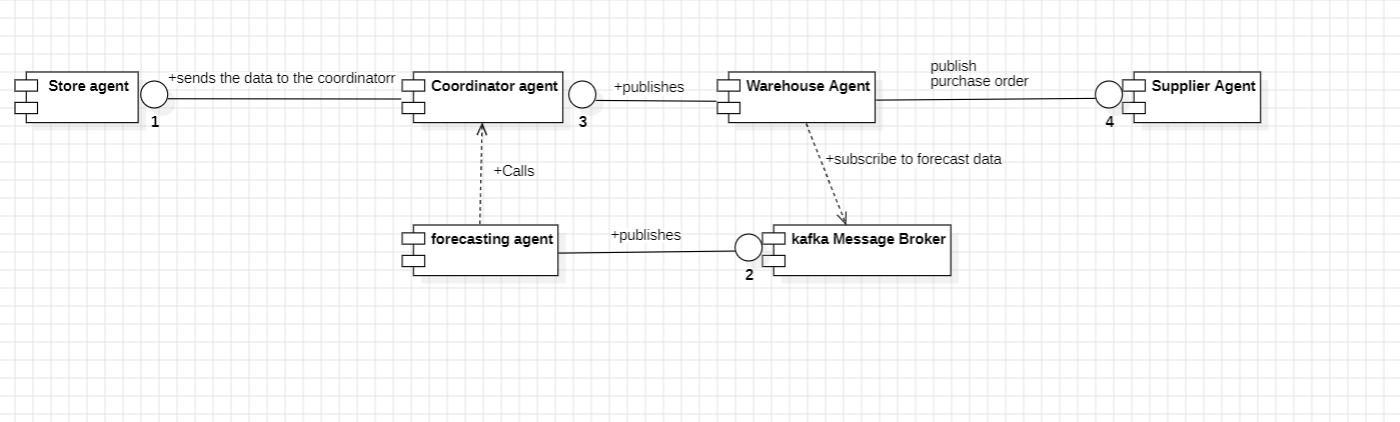
The proposed system is created to address the challenges of real-time demand forecasting and efficient inventory management in a retail environment. Each agent performs a dedicated function and communicates through a central messaging system **KAFKA** to ensure scalability and modularity

**Agents and Roles**

1. Store Agent: Responsible for collecting daily sales data from retail outlets and sending it to the Coordinator agent
2. Coordinator Agent: Coordinator agent is the central controller that ensures the smooth flow of data and coordinates the action between all other agents; it collects data from store agent and sends it to forecasting agent for prediction.
3. Forecasting Agent: Uses a time-series models like ARIMA or LSTM to predict future demand based on historical sales data. Publishes forecasts to Kafka
4. Warehouse Agent: Consumes forecast data and evaluates current stock levels. Alerts the supplier Agent if restocking is needed.
5. Supplier Agent: Receives restocking requests and processes them to fulfill supply needs. It may simulate ordering or interact with external vendor APIs.

COMMUNICATION PROTOCOL

* The system uses APACHE Kafka as the message broker to ensure asynchronous and decoupled communication between agents.
* Each agent either produces to or consumes from Kafka topics (e.g. sales data, demand forecasting , restock request)



Technology stack

* Language: Python
* Model: ARIMA
* Message Broker: Apache Kafka
* Visualization: StarUML