Project-High Level Design
on

Muti –Agent Manufacturing System

Course Name: Agentic AI

Institution Name:Medicaps University – Datagami Skill Based Course

Student Name(s) & Enrolment Number(s):

Sr no	Student Name	Enrolment Number
1	Pratik Patel	EN22CS301748
2	Pooja Dangi	EN22CS301708
3	Rahul Negi	EN22CS301778
4	Rupesh Sirvi	EN22CS301834
5	Pransu Singh	EN22CS301735

Group Name:09D7

Project Number:AAI-32

Industry Mentor Name: Aashruti Shah

University Mentor Name:Ajeet Singh Rajput

Academic Year:2025-2026

Table of Contents

1.Introduction

The **Collaborative Manufacturing Multi-Agent System** is an AI-powered platform designed to automate manufacturing decision-support tasks such as supplier sourcing and report preparation. Instead of using a single AI model, the system uses multiple specialized agents that collaborate through a structured hand-off process.

The **Researcher Agent** collects manufacturing information like suppliers, pricing, and market data, while the **Writer Agent** analyzes this data and generates a well-structured report. A coordinator manages the workflow, ensuring proper communication and state tracking between agents.

By mimicking real-world team collaboration, the system improves accuracy, reduces manual effort, and produces consistent, high-quality outputs, making it suitable for smart manufacturing and industrial automation environments.

1.1 Scope of the Document

This document defines the **High-Level Design (HLD)** of the *Collaborative Manufacturing Multi-Agent System* and explains how the system is architected to automate manufacturing decision-support workflows through cooperating AI agents.

The scope of this document includes:

1. Project Overview

- Describes the purpose and objectives of building a collaborative multi-agent manufacturing assistant
- Explains the importance of automating supplier sourcing and report preparation in industrial workflows
- Introduces the concept of agent specialization and structured hand-off communication

2. System Design

- Presents the overall architecture and module interaction
- Explains the role of the Coordinator, Researcher Agent, and Writer Agent
- Describes workflow orchestration and execution sequence
- Shows how agents collaborate to transform raw queries into structured reports

3. Communication & Workflow Logic

- Defines the inter-agent communication protocol
- Describes the hand-off mechanism between Researcher and Writer agents
- Explains how system state and context are preserved during execution

4. Data & State Management

- Overview of stored data such as queries, research results, and generated reports
- Session tracking and workflow state maintenance
- Logging and traceability of agent actions

5. Interfaces

- User interaction with the system through query submission and report viewing
- Backend APIs used for agent orchestration
- Integration with external tools or data sources used for research

6. Performance, Security, and Reliability Considerations

- Handling failures and retries during agent collaboration
- Ensuring secure storage and controlled access to data
- Maintaining scalability for multiple user queries

1.2 Intended Audience

This document is intended for individuals involved in the development, evaluation, and maintenance of the **Collaborative Manufacturing Multi-Agent System**. It helps different stakeholders understand the system architecture and the interaction between specialized agents.

1. Developers / AI Engineers

To understand the overall architecture, agent roles (Researcher and Writer), workflow orchestration, and communication protocols required for implementation and integration.

2. Testers & QA Team

To verify system behavior, validate agent collaboration, and ensure the workflow produces accurate and structured manufacturing reports.

3. Project Mentors & Stakeholders

To gain a clear overview of the system's objective, functionality, and how collaborative AI agents automate manufacturing decision-support tasks.

4. Future Maintainers / Contributors

To support future enhancements such as adding new agents, integrating tools, improving workflows, or scaling the system.

5. End Users (Engineers / Procurement Teams)

To understand how the system assists in supplier sourcing and report generation and how they interact with the platform for their manufacturing requirements.

1.4 System Overview

The **Collaborative Manufacturing Multi-Agent System** is an AI-based platform designed to automate manufacturing decision-support activities such as supplier sourcing and structured report generation. The system uses a multi-agent architecture in which specialized agents collaborate to complete complex tasks, similar to how different teams work together in a real manufacturing organization.

A user submits a manufacturing requirement (for example, sourcing a component or identifying suppliers). The system then processes this request through a coordinated workflow managed by a **Coordinator (Orchestrator)**. The coordinator interprets the request, assigns tasks to appropriate agents, and maintains the execution state throughout the process.

The workflow involves two primary specialized agents:

- **Researcher Agent:** Collects relevant manufacturing information such as supplier details, availability, pricing trends, and market insights using tools or external sources. It converts raw information into structured data.

- **Writer Agent:** Uses the structured data produced by the Researcher Agent to generate a clear, organized, and human-readable report including summaries and recommendations.

The agents communicate using a structured hand-off protocol, ensuring that information is transferred accurately and context is preserved between stages. The final output is a well-formatted manufacturing report that can assist engineers, procurement teams, or decision-makers.

The system also maintains session data, logs, and generated reports in a database for traceability, reuse, and analysis. Its modular design allows additional agents (such as analysis or cost optimization agents) to be added in the future, making the platform scalable and adaptable for smart manufacturing environments.

Overall, the system improves efficiency, reduces manual effort, and ensures consistent, high-quality outputs by combining specialization, workflow orchestration, and collaborative AI processing.

Benefits

The Collaborative Manufacturing Multi-Agent System provides several advantages by combining agent specialization, structured workflows, and AI-driven automation.

1. Reduced Manual Effort

The system automates supplier research and report preparation, reducing the need for engineers and procurement teams to manually collect and organize information.

2. Faster Decision-Making

By automatically gathering and structuring relevant data, the system significantly shortens the time required to evaluate suppliers and generate reports.

3. Improved Accuracy and Consistency

Specialized agents handle specific tasks (research and report writing), reducing errors and inconsistencies that may occur in manual processes.

4. High-Quality Structured Outputs

The Writer Agent ensures that the final output is professionally formatted, well-organized, and suitable for management or technical review.

5. Clear Task Specialization

The separation of responsibilities between Researcher and Writer agents improves clarity, reduces hallucinations, and enhances system reliability.

6. Scalable Architecture

The modular multi-agent design allows easy addition of new agents (e.g., Analyst Agent, Cost Optimization Agent) without redesigning the entire system.

7. Traceability and Auditability

All queries, research data, and generated reports are stored in the system, enabling tracking, auditing, and performance monitoring.

8. Enhanced Workflow Automation

The structured hand-off protocol ensures smooth collaboration between agents, mimicking real-world industrial team processes.

9. Flexibility and Extensibility

The system can integrate new tools, APIs, or data sources, making it adaptable for future smart manufacturing requirements.

10. Foundation for Smart Manufacturing

By combining AI collaboration, workflow orchestration, and structured state management, the system supports digital transformation and intelligent automation in manufacturing environments.

2 System Design

The Collaborative Manufacturing Multi-Agent System follows a **layered, modular architecture** designed for agent specialization and workflow orchestration.

1. Presentation Layer

Provides a user interface where users submit manufacturing queries and view generated reports.

2. Backend Layer

Built using FastAPI/Flask, it handles API requests, session management, business logic, and agent orchestration.

3. Agent Layer

This is the core of the system and includes:

- **Coordinator Agent** – Manages workflow, assigns tasks, and controls hand-offs.
- **Researcher Agent** – Collects supplier and market data.
- **Writer Agent** – Synthesizes structured data into formatted reports.

Agents communicate through a structured hand-off protocol using validated data payloads.

4. Data Layer

Stores queries, research results, reports, logs, and session information in a database (SQLite/PostgreSQL).

Overall, the system design ensures modularity, scalability, clear role separation, and efficient collaboration between specialized AI agents.

2.1 Application Design

The **Collaborative Manufacturing Multi-Agent System** is designed as a modular and scalable application that automates manufacturing research and documentation using cooperating AI agents. The architecture follows a layered approach where each module performs a specific responsibility and interacts through well-defined interfaces.

Modules Overview

1. **Query Input Module**

- Accepts user manufacturing requirements in text form
- Performs validation and preprocessing of the query
- Sends the request to the Coordinator for interpretation

2. Coordinator (Orchestrator) Module

- Analyzes the user request and identifies required workflow steps
- Initializes session and task identifiers
- Assigns tasks to the Researcher Agent and Writer Agent
- Maintains execution state and controls agent hand-offs
- Handles retries and failure conditions

3. Researcher Agent Module

- Gathers supplier information and manufacturing-related data
- Interacts with external sources/tools for data collection
- Organizes raw information into structured format (JSON)
- Passes validated structured output to the Coordinator

4. Validation & Context Management Module

- Verifies completeness of research data
- Maintains context and workflow state
- Ensures proper structured payload before moving to the next stage

5. Writer Agent Module

- Interprets structured research data
- Generates professional, human-readable manufacturing reports
- Produces summaries, recommendations, and formatted output

6. Database Module

- Stores queries, research outputs, final reports, and logs
- Maintains session history and traceability
- Supports retrieval for future reference and auditing

7. User Interface Module (Optional Dashboard)

- Displays submitted queries and generated reports
- Allows users to view history and download results

Architecture Highlights

- **Modular & Loosely Coupled:** Each agent and module works independently
- **Specialized Agents:** Clear separation between data collection and report generation
- **State-Aware Workflow:** Maintains context across agent interactions
- **Extensible:** New agents (e.g., Analyst Agent) can be added easily
- **Fault Tolerant:** Coordinator manages retries and validation before progression

2.2 Process Flow

The **Collaborative Manufacturing Multi-Agent System** follows a structured and sequential workflow where specialized agents collaborate through controlled hand-offs to generate a final manufacturing report.

Step-by-Step Process Flow

1 Query Submission

- User submits a manufacturing requirement (e.g., supplier sourcing request).
- Backend validates input and generates a unique session_id and task_id.

2 Coordinator Initialization

- Coordinator Agent interprets the user intent.
- Determines required workflow steps.

- Activates the Researcher Agent.

3 Research Phase (Researcher Agent)

- Collects supplier information, pricing, availability, and market insights.
- Organizes gathered data into structured JSON format.
- Sends structured output back to the Coordinator.

4 Validation & State Update

- Coordinator validates research payload (completeness & format).
- If incomplete → triggers retry.
- If valid → proceeds to Writer Agent.

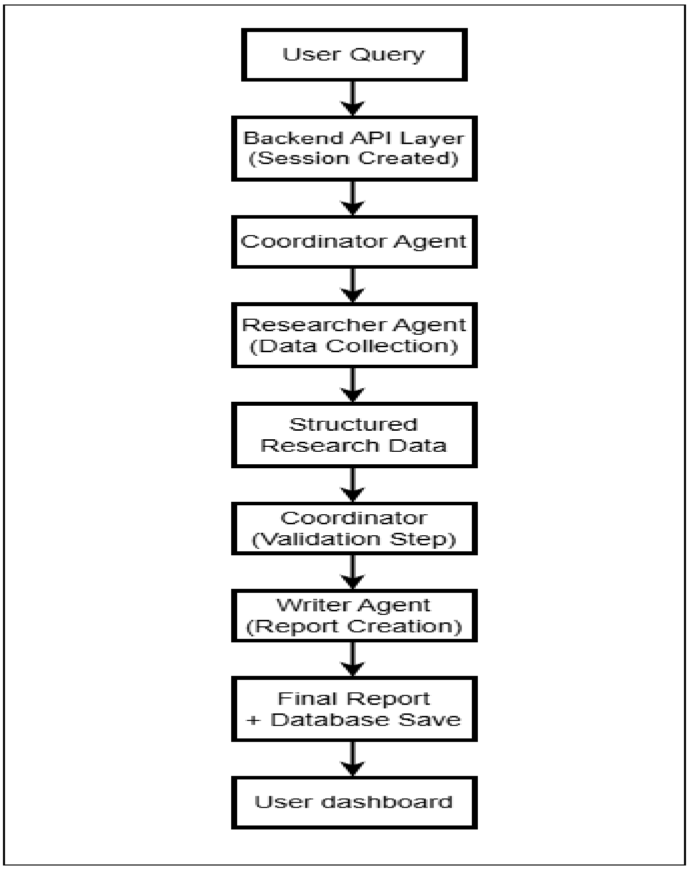
5 Report Generation (Writer Agent)

- Receives structured research data.
- Synthesizes data into:
  - Executive summary
  - Comparative analysis
  - Recommendations
- Produces formatted manufacturing report.

6 Final Output & Storage

- Final report returned to user dashboard.
- Query, research data, and report stored in database.
- Agent logs saved for traceability.

Process Flow Diagram


User Query
Backend API Layer (Session Created)
Coordinator Agent
Researcher Agent (Data Collection)
Structured Research Data
Coordinator (Validation Step)
Writer Agent (Report Creation)
Final Report + Database Save
User dashboard

2.3 Information Flow

The **information flow** describes how data moves through the Collaborative Manufacturing Multi-Agent System from the moment a user submits a request until the final structured report is generated and stored.

Step-by-Step Information Flow

1. User Submission

- The user enters a manufacturing requirement (e.g., supplier sourcing).
- The system temporarily stores the request and creates a unique session_id.

2. Coordinator Processing

- The Coordinator Agent analyzes the query to understand intent and required data.
- It prepares task metadata and forwards instructions to the Researcher Agent.

3. Research Data Collection

- The Researcher Agent gathers supplier details, pricing, availability, and market information from tools or sources.
- Raw data is cleaned and converted into structured format (JSON).
- The structured payload is sent back to the Coordinator.

4. Validation & Context Enrichment

- The Coordinator validates completeness and correctness of the data.
- Workflow state is updated and context is preserved for the next stage.

5. Report Generation

- The validated structured data is transferred to the Writer Agent.
- The Writer Agent synthesizes the information into a formatted report containing summaries and recommendations.

6. Storage & Output

- Final report is returned to the user interface.
- Query, research data, and report are saved in the database.
- Logs are recorded for traceability and auditing.

Simplified Information Flow

User Input
↓
Coordinator (Intent + Metadata)
↓

Researcher Agent (Data Collection)
↓
Structured Data (JSON)
↓
Coordinator (Validation & Context)
↓
Writer Agent (Report Generation)
↓
Final Report + Database Storage
↓
User Dashboard

2.4 Components Design

The **Collaborative Manufacturing Multi-Agent System** is built using modular components where each module performs a specific responsibility. This separation ensures clear agent specialization, easier maintenance, and scalable expansion.

1. User Input Module

- Accepts manufacturing queries from the user
- Validates and preprocesses the input
- Sends request to the backend workflow

2. API & Backend Module

- Handles HTTP requests and responses
- Manages sessions and task identifiers
- Connects UI with the agent orchestration system

3. Coordinator (Orchestrator) Module

- Interprets user intent
- Decides workflow sequence
- Assigns tasks to agents
- Maintains execution state
- Handles retries and error recovery

4. Researcher Agent Module

- Collects supplier and market information
- Uses external tools/APIs if required
- Converts gathered data into structured format
- Sends structured data to coordinator

5. Validation & Context Manager

- Verifies completeness and correctness of research output
- Maintains context between agents
- Prepares validated payload for next stage

6. Writer Agent Module

- Receives structured data from coordinator
- Generates formatted manufacturing report
- Produces summaries and recommendations

7. Database Module

- Stores queries, structured data, reports, and logs
- Maintains session history
- Enables auditing and retrieval

8. Logging & Monitoring Module

- Tracks agent activities and workflow states
- Records errors and execution details
- Supports debugging and performance analysis

9. User Interface / Dashboard Module

- Displays generated reports
- Shows query history

- Allows report download

Design Highlights

- **Loosely Coupled Modules:** Independent components
- **Agent Specialization:** Dedicated responsibilities
- **Controlled Hand-offs:** Coordinator-managed communication
- **Extensible Architecture:** New agents can be added easily

2.5 Key Design Considerations

While designing the **Collaborative Manufacturing Multi-Agent System**, several important factors were considered to ensure reliability, scalability, and effective collaboration between agents.

1. Agent Specialization

Each agent is assigned a clearly defined responsibility (Researcher for data collection, Writer for report generation).
This reduces confusion, improves output quality, and minimizes AI hallucinations.

2. Structured Hand-Off Communication

Agents communicate using a standardized structured payload instead of free text.
This ensures accurate context transfer and prevents data loss between workflow stages.

3. State Management

The system maintains session and workflow states (task status, research data, report output) to support multi-step processing and traceability.

4. Scalability

The modular architecture allows new agents (e.g., Analyst or Cost Optimizer) to be added without modifying existing components.
The backend can also support multiple concurrent user requests.

5. Reliability & Fault Tolerance

- Coordinator validates outputs before passing to next agent
- Retry mechanisms handle incomplete research
- Fallback report generation ensures continuity

6. Maintainability

Each module is loosely coupled, allowing independent updates, testing, and debugging without affecting the whole system.

7. Security

Sensitive data such as queries and reports are securely stored.
API keys and configuration values are protected using environment variables and controlled access.

8. Performance Optimization

Structured data exchange reduces unnecessary AI processing and improves response time.
Caching and logging help monitor and optimize performance.

2.6 API Catalogue

The Collaborative Manufacturing Multi-Agent System exposes REST and streaming APIs that enable users to execute sourcing workflows, monitor agent activity, and download generated reports.

1. Workflow Execution APIs

Method	Endpoint	Description	Input	Output
POST	`/api/query`	Initiates the manufacturing research pipeline and streams execution results	`{ "query": "manufacturing requirement" }`	Server-Sent Events (logs, supplier data, final report)
POST	`/api/stop`	Stops an ongoing agent workflow	`{ "session_id": "MFG-xxxx" }`	Stop confirmation
GET	`/api/health`	Checks server and	None	Server status

Method	Endpoint	Description	Input	Output
		model status		information

2. Report Retrieval APIs

Method	Endpoint	Description	Input	Output
GET	`/api/download/<session_id>`	Downloads generated sourcing report	session_id	Text report (.txt)
GET	`/api/download-json/<session_id>`	Downloads structured supplier dataset	session_id	JSON structured data

3. Streaming Response Events

The /api/query endpoint returns real-time execution updates using Server-Sent Events (SSE).

Event Type	Description
`session`	Provides unique session identifier
`log`	Displays agent processing steps
`suppliers`	Shows extracted supplier information
`done`	Returns final report and metadata
`stopped`	Indicates manual cancellation

4. Internal Agent Interfaces

Component	Function
Manufacturing Orchestrator	Controls workflow and agent sequencing
Researcher Agent	Collects and structures supplier data
Writer Agent	Generates formatted sourcing report

Component	Function
Pipeline State	Maintains hand-off context between agents

3. Data Design

The Collaborative Manufacturing Multi-Agent System manages and stores workflow information generated during the sourcing process. The data design focuses on maintaining query context, agent outputs, and generated reports while ensuring traceability between workflow stages.

Each user request creates a unique **session** that holds the query, intermediate research data, and final report. The Researcher Agent produces structured supplier data, which is passed to the Writer Agent and transformed into a formatted sourcing report. All outputs are temporarily stored in an in-memory store and can be downloaded as text or JSON.

The system also records execution logs and metadata such as processing time and workflow status to support monitoring and debugging. This design ensures clear linkage between user queries, agent collaboration, and generated results while keeping storage lightweight and efficient.

3.1 Data Model

The Collaborative Manufacturing Multi-Agent System stores workflow information using a session-based data structure.
Each user request creates a unique session that links the query, extracted supplier data, generated report, and execution logs.
The model ensures traceability between the Researcher Agent and Writer Agent hand-off process.

1. Session

Represents a single sourcing workflow execution.

Attribute	Description
session_id (Primary Key)	Unique identifier for each workflow run
query_text	User manufacturing requirement

Attribute	Description
status	Running / Completed / Stopped / Failed
created_at	Session start timestamp
completed_at	Session end timestamp
elapsed_time	Total processing duration

2. SupplierData

Stores structured supplier information collected by the Researcher Agent.

Attribute	Description
supplier_id (Primary Key)	Unique supplier identifier
session_id (Foreign Key)	Links to session
supplier_name	Company name
location	Supplier location
product	Manufactured product
price_range	Estimated pricing
contact_info	Email/phone/website
source	Directory or website name
confidence_score	Extraction reliability score

3. Generated Report

Contains the final report created by the Writer Agent.

Attribute	Description
report_id (Primary Key)	Unique report identifier
session_id (Foreign Key)	Associated session
report_text	Final formatted sourcing report

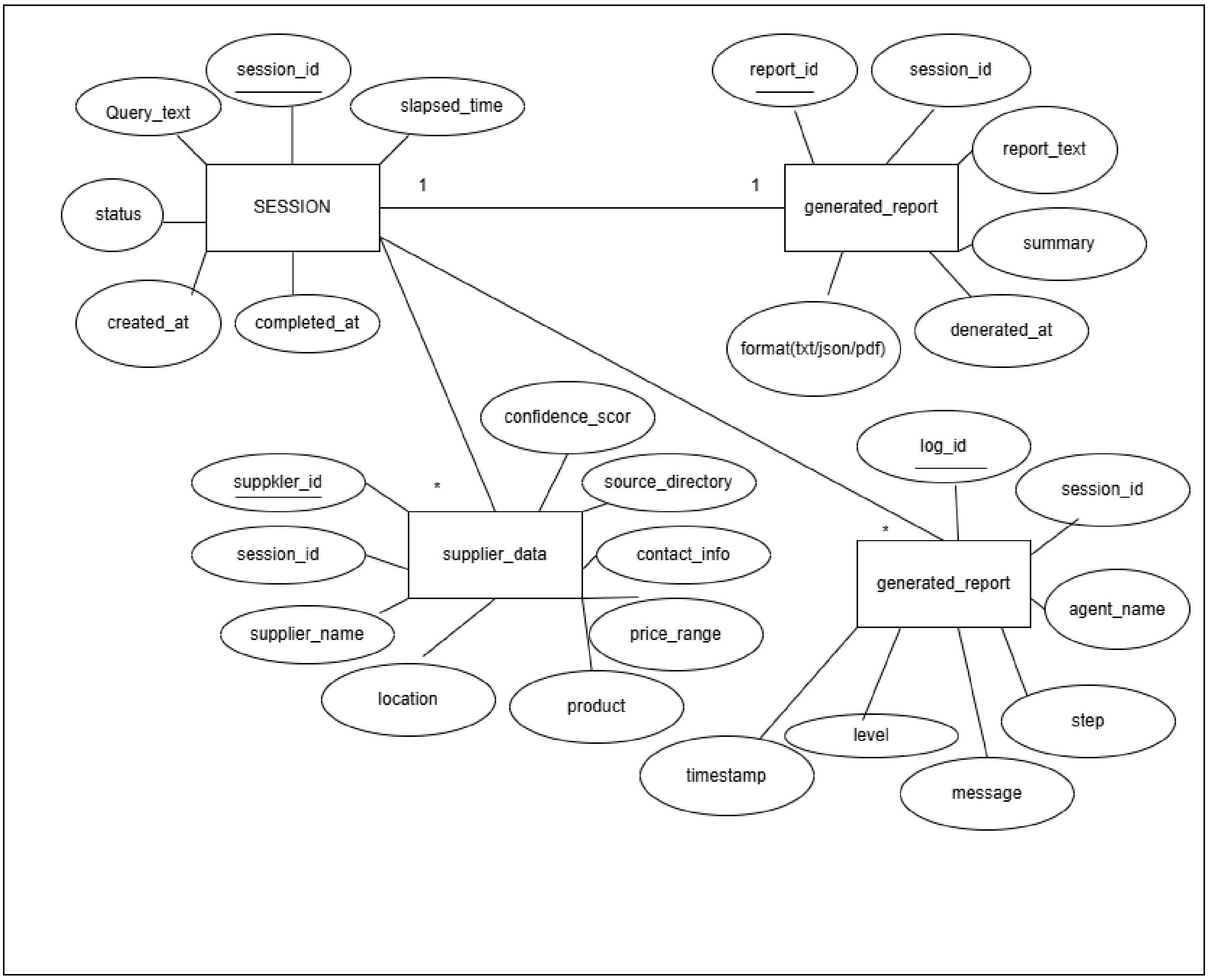
Attribute	Description
summary	Executive summary
generated_at	Report generation timestamp
format	TXT / JSON / PDF

4. Agent Logs

Tracks workflow execution and agent activities.

Attribute	Description
log_id (Primary Key)	Unique log entry
session_id (Foreign Key)	Associated workflow
agent_name	Coordinator / Researcher / Writer
step	Current operation performed
message	Execution message
level	info / warning / error
timestamp	Log time

ER Diagram



3.2 Data Access Mechanism

The Collaborative Manufacturing Multi-Agent System uses a session-based data access approach to manage workflow data generated by the Researcher and Writer agents. The mechanism separates application logic from storage operations to ensure maintainability and reliable agent communication.

1. Access Layer (Service Layer)

All data operations are handled through a dedicated service layer rather than direct database manipulation.
This layer:

- Creates and retrieves workflow sessions
- Stores structured supplier data
- Saves generated reports
- Records agent execution logs

It ensures that agents interact only with controlled interfaces, preventing inconsistent data updates.

2. Storage Strategy

The system primarily uses an **in-memory store** for active workflows:

- Stores session state during execution
- Holds generated reports temporarily
- Enables fast real-time streaming updates

Older sessions may be removed automatically after a limit is reached to optimize memory usage.

3. CRUD Operations

Operation	Purpose
create_session()	Initializes a new workflow session
save_suppliers()	Stores structured supplier data from Researcher Agent
save_report()	Saves final report generated by Writer Agent
get_report()	Retrieves report for download
log_event()	Records agent actions and status messages
stop_session()	Updates session state when cancelled

4. Data Flow Control

- Coordinator manages write operations
- Researcher Agent can only add supplier data
- Writer Agent reads supplier data and writes report
- Logs are append-only to maintain traceability

5. Advantages

- Prevents direct uncontrolled access by agents
- Maintains consistency across hand-offs
- Enables real-time updates during execution

- Improves debugging and monitoring

3.3 Data Retention Policies

The Collaborative Manufacturing Multi-Agent System follows lightweight retention rules since workflow data is stored temporarily in memory during execution. The policy ensures efficient memory usage while keeping sufficient information for user access and debugging.

Retention Rules

Data Type	Retention Period	Policy
Active Session Data	During execution	Stored in memory until workflow completes or stops
Generated Reports	Until server memory limit reached	Oldest reports automatically removed when limit exceeded
Supplier Structured Data	Same as report session	Deleted along with associated session
Agent Logs	Short-term (runtime only)	Cleared when session expires or server restarts
Cancelled Sessions	Immediate cleanup	Removed after cancellation confirmation
Downloaded Reports	Not stored permanently	User downloads locally; server does not keep files

Retention Strategy

- The system keeps a limited number of recent sessions in memory.
- No long-term disk storage is used to prevent unnecessary data accumulation.
- Automatic eviction removes the oldest workflow when capacity is exceeded.
- Server restart clears all temporary records.

Rationale

- Protects system memory and performance
- Ensures user privacy (no permanent storage)
- Simplifies maintenance and data management
- Suitable for real-time AI workflow execution systems

3.4 Data Migration

Definition

Data Migration refers to the process of transferring existing procurement or supplier-related data from legacy systems (such as spreadsheets, CSV files, or older databases) into the Collaborative Manufacturing Multi-Agent System.

Since the current system primarily uses in-memory storage for active sessions, migration is mainly required when integrating historical supplier datasets or moving to a persistent database in future enhancements.

Migration Scenarios

1. Legacy Supplier Data Import

If an organization has existing supplier records in:

- Excel sheets
- CSV files
- ERP exports

- Previous procurement systems

These can be migrated into the system's structured data format.

2. Migration to Persistent Database (Future Upgrade)

If the system transitions from in-memory storage to:

- PostgreSQL
- MySQL
- MongoDB

Existing session data and reports can be exported and inserted into the new schema.

Migration Steps

Step 1: Data Assessment

- Identify data sources (CSV, Excel, legacy DB)
- Check for missing values, duplicates, formatting issues
- Validate supplier fields (name, location, product, contact info)

Step 2: Data Mapping

Map legacy fields to the new system schema:

Old Field	**New Field**
Company Name	supplier_name
Address	location
Product Category	product
Phone / Email	contact_info
Price	price_range
Source	source_directory

Step 3: Data Cleaning

- Remove duplicates
- Normalize formats (phone, email, currency)
- Validate mandatory fields

Step 4: Migration Execution

- Use Python ETL scripts or SQL scripts
- Convert data into system-compatible JSON or database format

- Insert into SupplierData table or persistent storage

Step 5: Post-Migration Validation

- Verify supplier counts
- Test report generation using migrated data
- Ensure no broken references in sessions

Tools for Migration

- Python (pandas, SQLAlchemy)
- SQL scripts
- CSV/Excel import utilities
- ETL pipelines

Migration Considerations

- Maintain data integrity during transfer
- Ensure consistent data formatting
- Backup original data before migration
- Log migration process for auditing

4 Interfaces

The Collaborative Manufacturing Multi-Agent System provides multiple interfaces that enable communication between users, internal agents, and external data sources. These interfaces ensure smooth execution of the sourcing workflow and proper coordination between system components.

4.1. User Interface (UI)

Type: Human–System Interface

Provides an interactive chat-based web interface where users can submit manufacturing sourcing queries and view real-time results.

Functions:

- Enter sourcing query
- View live agent progress (logs & supplier cards)
- Download generated reports (TXT/JSON/PDF)
- Stop running workflow

4.2. System / Internal Interfaces

Type: Internal API Communication

Used for communication between system modules and agents during workflow execution.

Interface	Purpose
Coordinator ↔ Researcher Agent	Sends query and receives structured supplier data
Coordinator ↔ Writer Agent	Sends validated data and receives final report
Pipeline State Interface	Maintains shared context between agents
Logging Interface	Records workflow progress and status

4.3. External Service Interfaces

Type: External API / Data Source Interface

Used by the Researcher Agent to collect supplier information.

External Source	Purpose
Web Search Engines	Discover supplier websites
B2B Directories (IndiaMART, Alibaba, etc.)	Retrieve supplier listings
LLM Processing Service	Extract structured data & generate report

4.4. Programmatic API Interface

Type: REST & Streaming API

Allows applications or frontend to interact with the backend workflow.

Key Operations:

- Start pipeline execution
- Cancel running workflow
- Download reports
- Monitor system status

Summary of Interface Types

Interface Category	Type
User Interface	Human–System
Internal Agent Communication	Internal API
External Data Sources	External API
Backend Service Access	REST / Streaming API

5.State and Session Management

In the **Collaborative Manufacturing Multi-Agent System**, state and session management are essential to maintain context while multiple agents (Coordinator, Researcher, and Writer) collaborate in a single workflow. Since the sourcing process involves multiple steps, the system must preserve intermediate results and track execution progress.

Session Management

- Every user query generates a unique **session_id** (e.g., MFG-XXXXXX).
- The session links the entire workflow including query, supplier data, logs, and final report.
- Sessions remain active during execution and until the user downloads the report or the system removes it from memory.
- A running session can be manually cancelled using the stop command.

- Sessions automatically expire when the memory limit is reached or the server restarts.

Session Flow:

1. User submits query → new session created
2. Agents process request using same session_id
3. Report generated and stored in session
4. User downloads results or session expires

State Management

The system maintains workflow state using a shared pipeline context so each agent can continue from the previous step without losing information.

Core State Variables

- session_id – unique workflow identifier
- query_text – original manufacturing request
- supplier_data – structured data from Researcher Agent
- report_output – final report from Writer Agent
- execution_status – running / completed / stopped / failed
- logs – step-by-step execution messages

State Transitions

Stage	State
Query Received	Initialized
Researcher Collecting Data	Processing
Data Validated	Ready for Writing
Report Generated	Completed
User Cancels	Stopped

Stage	State
Error Occurs	Failed

Purpose

- Preserves context across multiple agents
- Enables controlled hand-off between agents
- Supports cancellation and recovery
- Allows real-time streaming updates
- Provides traceability for debugging

6.Caching

Caching is used in the **Collaborative Manufacturing Multi-Agent System** to improve performance, reduce repeated processing, and optimize response time during agent workflows. Since supplier sourcing and report generation involve external requests and AI processing, caching helps avoid redundant operations.

Purpose of Caching

- Reduce repeated web scraping for similar queries
- Minimize repeated AI processing (LLM calls)
- Improve response speed for users
- Reduce external API usage and cost

Caching Strategy

Cache Type	Use Case
In-Memory Cache	Stores recent sessions, supplier data, and generated reports
Query Result Cache	Reuses supplier results for similar queries
LLM Output Cache	Stores previously generated summaries or reports

Cache Type	Use Case
Log Cache	Keeps recent workflow logs for real-time display

Cache Workflow

1. User submits query
2. System checks if similar query exists in cache
3. If found → return cached suppliers/report
4. If not found → run full agent pipeline and store result in cache

Cache Expiry

- Old sessions removed automatically when memory limit reached
- Cache cleared on server restart
- Frequently accessed results retained longer

Benefits

- Faster response time
- Reduced system load
- Lower external API calls
- Improved scalability during multiple user requests

7. Non-Functional Requirements (NFRs)

Non-functional requirements define how the **Collaborative Manufacturing Multi-Agent System** should perform and operate, rather than what functions it provides.

1. Performance

- The system should provide real-time progress updates during workflow execution.
- Average response time for report generation should remain within acceptable limits under normal load.
- Streaming updates must appear continuously without blocking the user interface.

2. Scalability

- The architecture must support multiple simultaneous user sessions.
- New agents (e.g., Analyst Agent) can be added without redesigning the system.
- Stateless API design allows horizontal scaling of backend services.

3. Reliability

- Workflow continues even if one data source fails.
- Retry and fallback mechanisms ensure report generation completion.
- System should handle cancellations safely without corrupting session state.

4. Availability

- The system should remain accessible during long-running operations.
- Health check endpoint monitors service availability.
- Failures should not crash the entire application.

5. Security

- Sensitive configuration values stored securely.
- Session isolation prevents unauthorized access.
- Input validation protects against malicious requests.

6. Maintainability

- Modular architecture allows independent updates of agents.
- Clear separation between orchestration logic and data collection modules.
- Logging supports debugging and monitoring.

7. Usability

- Users can submit queries easily through a simple interface.
- Real-time feedback helps users understand workflow progress.
- Reports available in downloadable formats.

8. Efficiency

- Caching reduces repeated processing.
- Memory usage is controlled using session expiration.
- Optimized processing reduces unnecessary external API calls.

8.References

1. **Groq LLM API Documentation**
   https://console.groq.com/docs
2. **Flask Web Framework Documentation**
   https://flask.palletsprojects.com/
3. **Server-Sent Events (SSE) – MDN Web Docs**
   https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events
4. **Python Requests Library Documentation**
   https://docs.python-requests.org/
5. **BeautifulSoup HTML Parsing Documentation**
   https://www.crummy.com/software/BeautifulSoup/bs4/doc/
6. **DuckDuckGo Search (duckduckgo-search library)**
   https://pypi.org/project/duckduckgo-search/
7. **B2B Supplier Directories Used for Data Collection**
   - IndiaMART – https://www.indiamart.com/
   - TradeIndia – https://www.tradeindia.com/
   -
   - ExportersIndia – https://www.exportersindia.com/
   - Alibaba – https://www.alibaba.com/
   - Made-in-China – https://www.made-in-china.com/
   - GlobalSources – https://www.globalsources.com/
   - ThomasNet – https://www.thomasnet.com/
   - Europages – https://www.europages.com/
   - Kompass – https://www.kompass.com/