# **Technical Research Report: Agent-to-Agent Communication via Structured Synthesis Artifacts**

Version: 20.0

Date: 17 June 2025

## **1. Abstract**

This document proposes a critical enhancement to the agentic system's output capabilities, designed to enable a full-cycle, automated research and development workflow. Currently, the system's final product is a human-readable Markdown report. While valuable for user review, this unstructured text is a poor and ambiguous input for downstream specialized agents, such as a "Design Specification Agent" or a "Code Generation Agent". To solve this, we propose that the Synthesis Graph be modified to produce a second, parallel artifact: a structured, machine-readable research\_synthesis.json. This Agent-to-Agent (A2A) artifact will not contain the narrative of the report, but rather the distilled, conceptual essence of the research findings, including key architectural principles, system components, and workflow stages. This structured output will serve as a clean, unambiguous API contract between agents, allowing a downstream Design Agent to reliably ingest the verified concepts from the research phase and translate them into a formal technical specification, thus enabling a true multi-agent R&D pipeline.

## **2. Research: The Limitations of Human-Readable Outputs for A2A Communication**

The current v2 architecture successfully produces a high-quality, human-readable research report. However, this report.md file represents a "last-mile" translation of structured concepts into unstructured prose. For a subsequent specialized agent to use this report, it would need to perform the difficult and error-prone task of re-interpreting the natural language to extract the core architectural concepts. This introduces a significant risk of ambiguity and information loss.

The project has already validated the principle that agents should communicate via structured data internally. The move from Markdown to plan.json and evaluation.json for state management was a critical step in improving system robustness [cite: Technical Research Report v18.docx]. The proposed enhancement applies this same core principle to the system's final, consumable output.

## **3. The A2A Enhancement: Generating** research\_synthesis.json

The proposed architecture modifies the Synthesis graph. In addition to its current task of assembling the Markdown report, it will be given a second responsibility: to populate and output a structured JSON object based on its complete understanding of the topic.

### **3.1. Proposed Schema**

The research\_synthesis.json artifact will be structured to provide a complete, conceptual blueprint of the researched system.

{

"metadata": {

"report\_version": "20.0",

"generation\_date": "2025-06-17",

"primary\_sources": ["TRR-v19.md", "TDS-v4.0.md", "agent\_core\_v2.py"]

},

"high\_level\_summary": "The system evolved from a multi-agent committee to a simplified Teacher-Student model that uses tool-augmented auditing and structured JSON state for robust, verifiable evaluation.",

"key\_architectural\_principles": [

"Plan-Synthesize-Evaluate Loop",

"Human-in-the-Loop (HITL) Supervision",

"Teacher-Student Model for Independent Auditing",

"Tool-Augmented Auditing for Verifiability",

"Structured JSON State Exchange for Robustness",

"Single-Call Evaluator Graph for Simplicity"

],

"system\_components": [

{"name": "Supervisor", "description": "Streamlit-based orchestrator managing UI, state, and graph invocations."},

{"name": "Student Agent", "description": "Responsible for planning and synthesis tasks."},

{"name": "Teacher Agent", "description": "An independent agent responsible for tool-augmented evaluation."},

{"name": "Knowledge Base", "description": "FAISS vector store for semantic retrieval of source documents."}

],

"workflow\_stages": [

"PLANNING",

"SYNTHESIS",

"EVALUATION",

"FEEDBACK\_REFINEMENT"

],

"state\_management": {

"definition": "GraphState TypedDict",

"keys": ["user\_prompt", "documents", "plan", "output", "evaluation\_report", "user\_feedback"]

},

"required\_tools": [

{"name": "citation\_retriever", "description": "Retrieves full text of a source document for claim verification."}

]

}

## **4. Enabling the Full Research & Development Flow**

The research\_synthesis.json file acts as the API contract that enables a true multi-agent R&D pipeline.

1. Research Phase (Current Agent): Consumes the raw knowledge base and produces two artifacts:
   * report.md (for the human user).
   * research\_synthesis.json (for the next agent).
2. Design Phase (Future Agent): A new Design Agent is invoked.
   * Input: It takes research\_synthesis.json as its primary instruction.
   * Process: Its task is simplified and made more reliable. It directly iterates through keys like system\_components and key\_architectural\_principles to generate a formal Technical Design Specification. It does not need to re-read or interpret the natural language report.
   * Output: design\_spec\_v5.0.md.
3. Implementation Phase (Future Agent): A new Code Generation Agent is invoked.
   * Input: It consumes the design\_spec\_v5.0.md as its primary instruction and can use research\_synthesis.json for core definitions (e.g., the state\_management.keys to write the new GraphState TypedDict).
   * Output: supervisor\_v3.py, agent\_core\_v3.py.

This enhancement is the final conceptual step required to transform the system from a powerful research synthesizer into the foundation of a fully automated, end-to-end software development pipeline.