# **Technical Design Specification: A2A Synthesis Artifact Generation**

Version: 5.0

Date: 17 June 2025

Status: Proposed (Based on Technical Research Report v20.0)

## **1. Architectural Overview**

This document provides the technical implementation plan for the Agent-to-Agent (A2A) communication enhancement detailed in TRR v20.0. The primary goal is to modify the existing v2 agentic system so that, in addition to its human-readable Markdown report, it produces a second, machine-readable research\_synthesis.json artifact.

This enhancement will be achieved by augmenting the existing Synthesis graph. After the graph assembles the final Markdown report, it will perform a final cognitive step: invoking an LLM with a new, dedicated prompt to distill the concepts from the completed report into the required structured JSON format. The Supervisor's artifact-saving logic will also be updated to handle this new JSON output.

This modification lays the foundational technical groundwork for a full multi-agent R&D pipeline, where the structured JSON output of this Research Agent can be reliably consumed by future, specialized agents (e.g., a Design Specification Agent or a Code Generation Agent).

## **2. State Management Modifications**

To accommodate the new A2A artifact, the GraphState TypedDict in agent\_core\_v2.py will be updated with one new key.

agent\_core\_v2.py -> GraphState

class GraphState(TypedDict):

# ... (all existing keys: user\_prompt, output, etc.)

# --- NEW KEY for v3 implementation ---

a2a\_output: Optional[Dict] # Will hold the structured research\_synthesis.json content

## **3. Computational Graph Modifications**

The core logic change will occur within the build\_synthesis\_graph function in agent\_core\_v2.py. Specifically, the assemble\_draft node will be expanded.

agent\_core\_v2.py -> assemble\_draft node

The responsibilities of this node will be updated as follows:

1. Assemble Markdown Draft (Existing Logic): The node will first assemble the final human-readable report from the working\_memory and add indexed citations, storing the result in state['output']. This logic remains unchanged.
2. Generate A2A Artifact (New Logic): After the Markdown report is finalized, the node will immediately perform a *new, final LLM call*.
   * It will use a new prompt, create\_a2a\_synthesis\_prompt, which will be added to prompts.yaml.
   * This prompt will be formatted with the full text of the just-assembled Markdown report (state['output']).
   * The LLM will be instructed to read the report and generate a single JSON object that conforms to the schema defined in TRR v20.0.
3. Update State: The node will parse the JSON from the LLM's response and store the resulting dictionary in the new state['a2a\_output'] key.

This ensures that the structured JSON is a direct, distilled synthesis of the final, user-visible report.

## **4. New Prompt Specification**

To guide the new A2A generation step, a new prompt must be added to prompts.yaml.

prompts.yaml -> Add new prompt

# ... (existing prompts)

# --- NEW PROMPT FOR A2A SYNTHESIS ---

create\_a2a\_synthesis\_prompt: >

You are a metadata and systems analysis expert. Your task is to read the following final research report and distill its core conceptual and architectural information into a structured JSON object.

\*\*CRITICAL:\*\* Your entire output MUST be a single, valid JSON object that conforms to the schema below. Do not add any other text, comments, or markdown formatting.

JSON Schema:

{{

"metadata": {{

"report\_version": "<The version of the system being described, e.g., 19.0>",

"generation\_date": "<The current date in YYYY-MM-DD format>",

"primary\_sources": ["<List of key source documents mentioned in the report>"]

}},

"high\_level\_summary": "<A one-sentence summary of the system's final architecture>",

"key\_architectural\_principles": [

"<A list of the core design principles, e.g., 'Plan-Synthesize-Evaluate Loop'>"

],

"system\_components": [

{{"name": "<Component Name>", "description": "<A one-sentence description>"}}

],

"workflow\_stages": [

"<A list of the main workflow phases, e.g., 'PLANNING'>"

],

"state\_management": {{

"definition": "<The name of the state object, e.g., GraphState TypedDict>",

"keys": ["<A list of the most important keys in the state object>"]

}},

"required\_tools": [

{{"name": "<Tool Name>", "description": "<A one-sentence description>"}}

]

}}

### FINAL RESEARCH REPORT TO ANALYZE:

{final\_report\_text}

## **5. Supervisor Modifications**

The supervisor\_v2.py script requires a minor modification to its save\_artifact function to handle the new potential output.

supervisor\_v2.py -> save\_artifact function

The function will be updated to check for the presence of the a2a\_output in the final state.

* Existing Logic: The function will continue to save the content of state['output'] to a .md file as it does now.
* New Logic: After saving the Markdown file, the function will add a new if block:

if state.get('a2a\_output'):

# ... logic to save the a2a\_output dictionary as a .json file ...

# e.g., with open(".../research\_synthesis.json", "w") as f:

# json.dump(state['a2a\_output'], f, indent=2)

This ensures that whenever the agent successfully produces the A2A artifact, it is automatically saved alongside its human-readable counterpart, ready for use by other agents in the R&D pipeline.