### **Protocol for the THINKING Cell: A Zero-Hallucination Monologue**

#### **Core Mandate: The Principle of Verifiable and Declared Reality**

The THINKING cell is the assistant's internal monologue and a verifiable log of its reasoning. It **must be 100% free of hallucination**. Every entity mentioned—including tools, parameters, and values—must be explicitly traceable to a ground-truth source (**User's Request**, **System Prompt**, or **Tool Output**).

Critically, any action taken in a tool\_code block **must first be explicitly declared and justified** within the immediately preceding THINKING cell.

### **1. Rules of Grounding: Connecting Thought to Reality**

The reasoning process must be firmly anchored in facts from the established context.

* **User Request Adherence:** The thinking must accurately extract all intents, entities, constraints, and values from the user's prompt. It must not invent user needs or ignore user-provided details.
* **System Prompt Adherence:** All reasoning must operate within the rules, permissions, and contextual data (e.g., picker\_ids, supervisor\_id) defined in the system prompt.
* **Tool Output Fidelity:** When referencing the result of a previous tool call, the THINKING cell must use the **exact** data from the tool\_output. Misremembering, altering, or inventing details from past results is a critical hallucination.

### **2. The Bridge to Action: From Thought to Tool Call**

The primary purpose of the THINKING cell is to formulate a precise, executable plan. The connection between the THINKING cell and the subsequent tool\_code cell is non-negotiable and must follow these rules.

* **Rule 1: Explicit Tool Declaration:** The thought process must culminate in a clear and unambiguous statement of the exact tool to be called next.
* **Rule 2: Verifiable Naming (No Hallucinated Entities):**
  + The declared **tool name** must exactly match a name in the available [tools] schema.
  + Any **parameter name** mentioned must be a valid, existing parameter for that specific tool in the schema.
  + **If any parameter name is mentioned, then all parameter names for that tool, along with their values, should also be mentioned.**
  + **Only the parameter values explicitly mentioned in the thinking cell can be used in subsequent tool calls; values not mentioned cannot be used in the tool call below.**
* **Rule 3: Value Provenance (Justify Every Value):** For every parameter being used in the planned tool call, the thinking process must explain where its value came from.
* **Rule 4: The Mandate of Declared Action (No Undeclared Actions):** This is a critical check for hallucination.
  + The tool\_code block is **purely for execution, not for decision-making.**
  + **Only the tool and parameters explicitly declared and justified in the immediately preceding THINKING cell may be used.**
  + Executing a tool call where the tool name, a parameter, or a value was not first mentioned and rationalized in the THINKING cell is a form of hallucination known as an **"Undeclared Action"** and constitutes a critical failure.

### **3. Characteristics of a High-Quality Thought Process**

While adhering to the strict rules above, the monologue's style should be transparent and logical.

* **Step-by-Step Logic:** Break down the problem into a clear sequence of steps.
* **Justification (What, Why, How):** Narrate your internal monologue. Don't just state what you are doing—explain *why* it's the correct step and *how* it helps achieve the user's goal.
* **Self-Correction and Reflection:** If an initial path seems incorrect, acknowledge it and backtrack (e.g., "My first thought was to use tool\_A, but I see it requires a value I don't have. Therefore, the correct approach is to first use tool\_B to acquire that value.").
* **Transparency of Assumptions:** Any inference made (and permitted by the System Prompt) must be explicitly stated.

### **Verification Checklist**

To verify a THINKING cell is valid, confirm the following:

* **[ ] Grounded in Reality:** Is every piece of information traced to the user, system, or a tool output?
* **[ ] Valid Tool and Parameters:** Does the planned tool exist in the schema, and are its parameters valid?
* **[ ] Justified Values:** Is the source of every parameter's value explained?
* \*\*[ ] **Undeclared Action Check:** Does the following tool\_code block **only** contain the tool and parameters that were finalized in the THINKING cell? There should be zero differences between the final plan and the execution.

4. Thinking should exactly contain next