**Core Principle: The tool schema is the single, absolute source of truth. Any deviation in tool definition, call, or output is a critical failure.**

**1. Tool Design Mandates**

* **Minimum Capability:** A minimum of 10 distinct tools must be defined.
* **User Context:** Tools that perform user-specific actions (e.g., booking, getting order status) **must** include a user\_id or equivalent identifying parameter.
* **Realism & Logical Consistency (New Point):**
  + **Parameter Relevance:** For each tool, critically assess whether its defined parameters are logically necessary and sufficient for its stated function. For example:
    - A booking tool **must** require user\_id (or similar user context), item\_id, quantity, and booking\_date.
    - An availability checking tool **should** require item\_id and potentially date\_range or quantity\_desired.
  + **Name & Description Alignment:** The tool's function\_name and description must accurately and clearly reflect its purpose and the action it performs. Avoid ambiguous or misleading names.
  + **Return Value Relevance:** The returns schema of a tool must logically correspond to the information expected as a result of its execution. For instance, a booking tool's return should confirm the booking status, ID, and perhaps remaining availability, while an availability check should return available quantities or dates.
  + **Realistic Constraints:** Consider real-world constraints and model them in the schema (e.g., maximum number of tickets for an event, valid date ranges).

**2. Tool Call (tool\_use) Mandates**

* **Exact Naming:** The function\_name used in a tool call must be an exact, case-sensitive match to a name in the tool schema.
* **Parameter Integrity:**
  + **Defined Only:** Use only parameters that are explicitly defined for that tool in the schema. Do not invent parameters.
  + **Required Parameters:** Provide all parameters marked as required.
  + **Enum Compliance:** If a parameter uses an enum, the provided value must be an exact, case-sensitive match from the defined list.
  + **Type & Pattern:** The value for every parameter must strictly conform to its defined data type (e.g., integer, string) and pattern (regex).

**3. Tool Output (tool\_output) Mandates**

* **Schema Compliance:** The JSON structure of a tool's output must perfectly match its defined returns schema, including all field names, data types, and nesting.
* **No Extraneous Data:** The output must not contain any fields or properties that are not explicitly defined in the returns schema.
* **Accurate Consumption:** The agent's subsequent reasoning must correctly and accurately use the data provided in the tool output.

1. These are all not problems don’t take exact things only take context:  
    1. The actual tool output for fetch\_vehicle\_details is a JSON array containing an object, which contradicts the returns schema defined in cell 3 for this tool, which specifies a single JSON object. - wrapped in a JSON array [], which is a type mismatch.