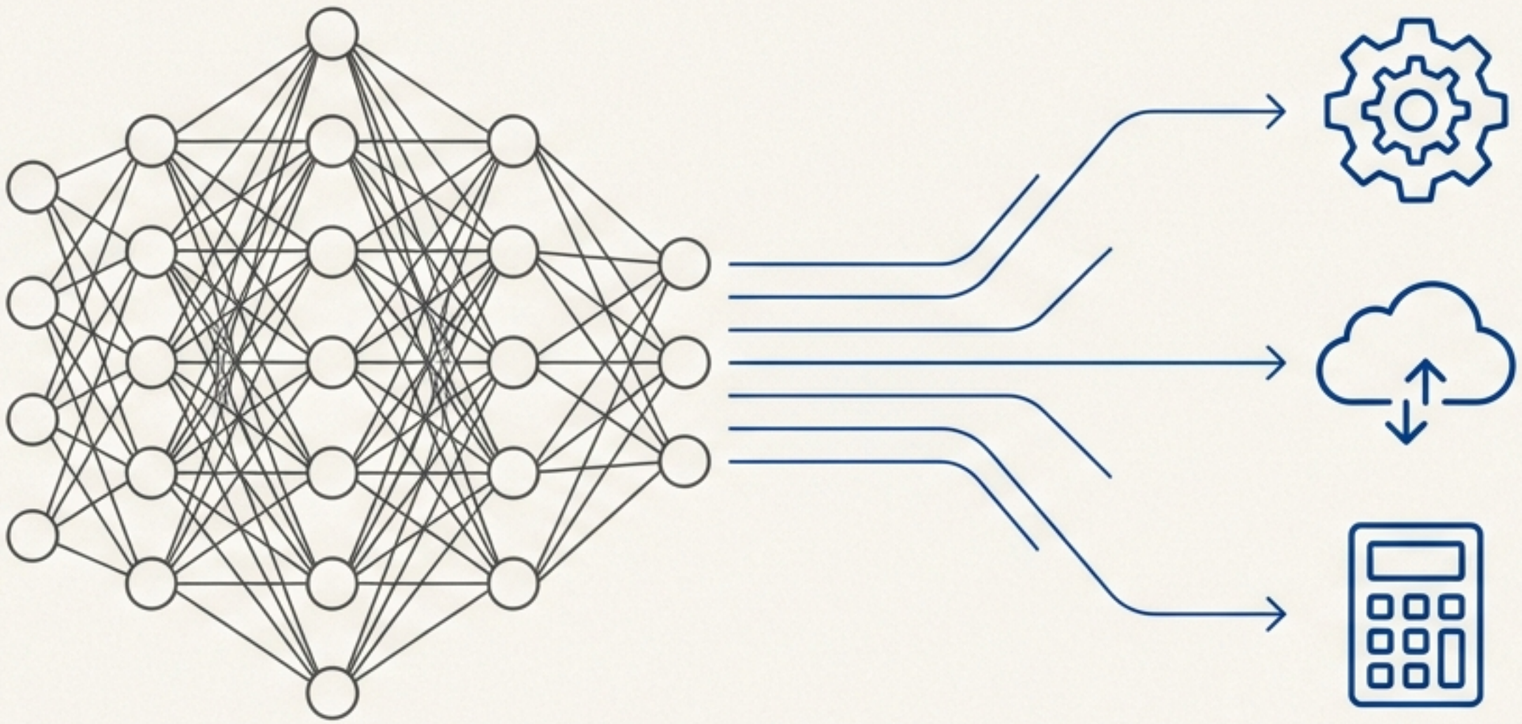


Beyond Words: Giving AI the Tools to Act

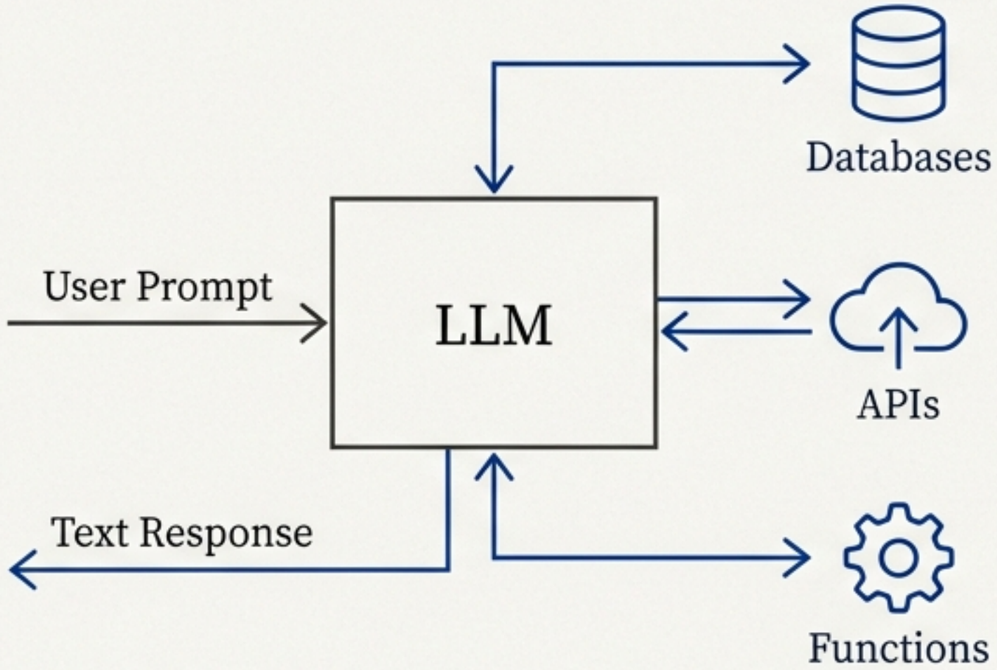
A Practical Guide to AI Tool Use with a Real-World WordPress Plugin

NotebookLM

The fundamental shift from a text generator to an interactive agent.

The Old Way: Text-In, Text-Out

The New Way: Interactive Agent



Language models are no longer confined to their training data. They can now access real-time information and perform actions in external systems, transforming them into powerful agents.

NotebookLM

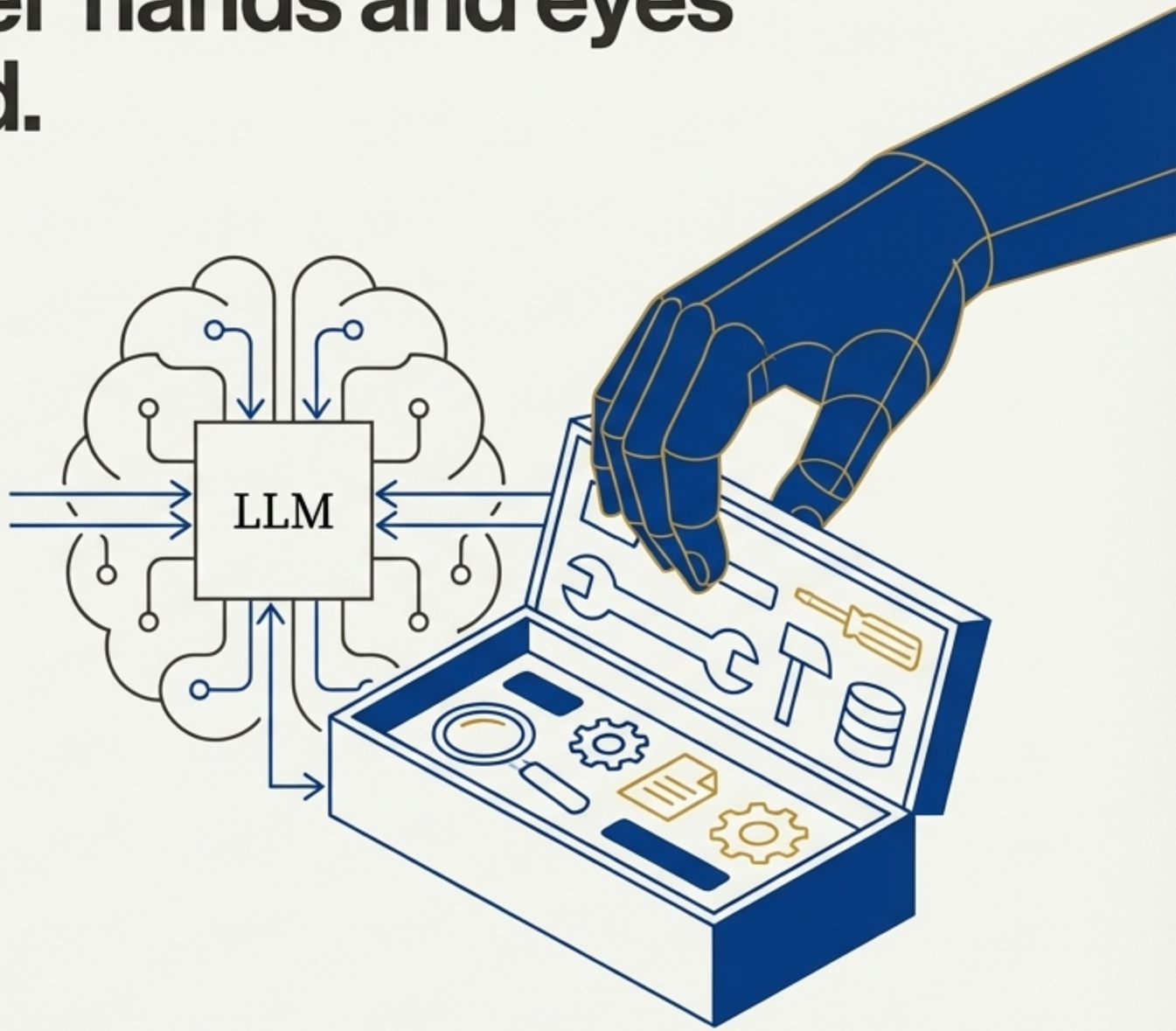
AI Tool Use gives a model ‘hands and eyes’ to interact with the world.

What is AI Tool Use?

(Also known as function calling)

It is an AI capability that allows a model to:

- Detect when a user’s request requires an external action.
- Select the appropriate function (a ‘tool’) from a predefined list.
- Generate the exact parameters needed to call that function.
- Incorporate the function’s result into its final, natural-language response.



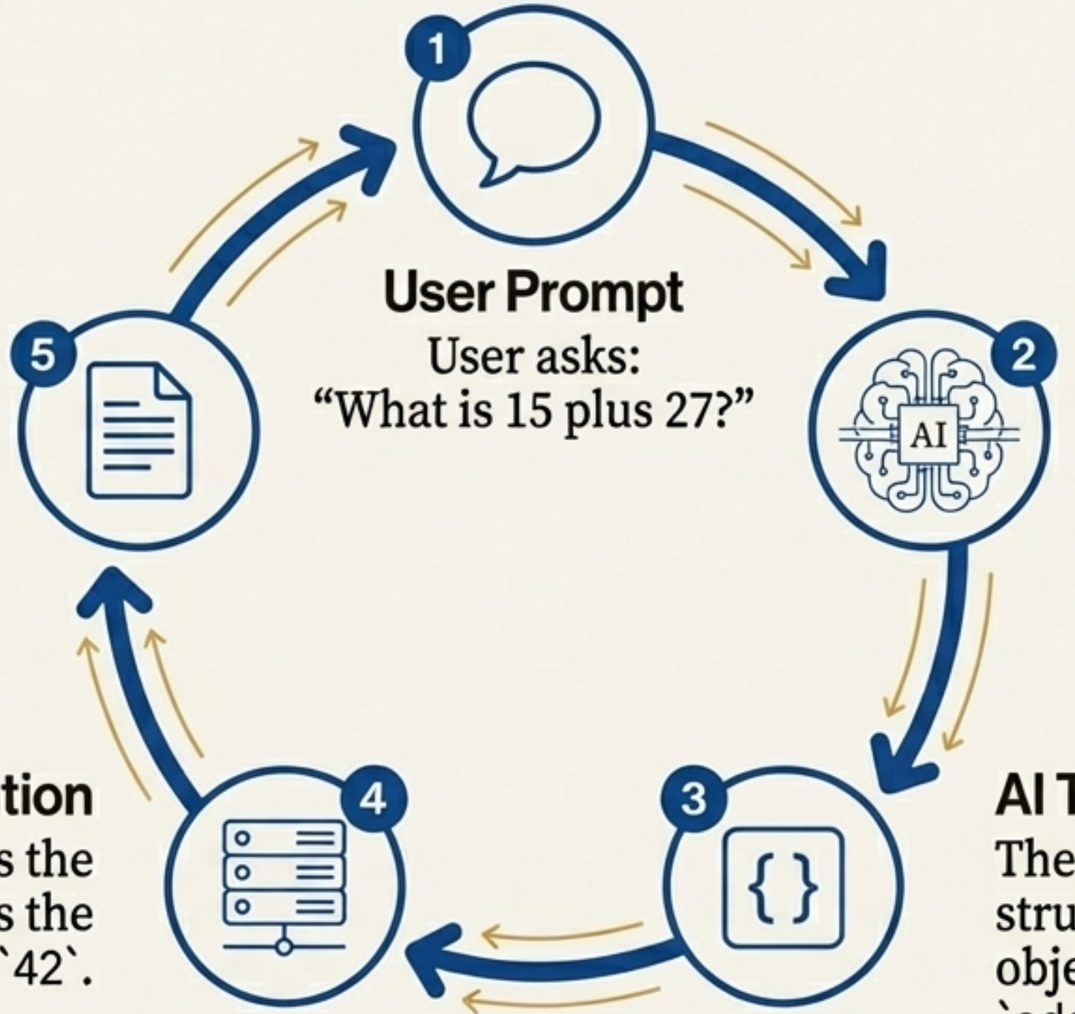
NotebookLM

Our WordPress plugin provides a tangible bridge between the AI and a server's capabilities.

To make this concept concrete, we built a simple WordPress plugin. It demonstrates how to connect OpenAI's language models to custom PHP functions running on a server, turning abstract potential into practical reality.

NotebookLM

The tool-calling process is a five-step loop between the user, the AI, and the server.



User Prompt
User asks:
“What is 15 plus 27?”

AI Analysis
The LLM analyzes the prompt and determines the `add_two_numbers` tool is needed.

AI Tool Call Request
The model’s response is a structured `tool_calls` object requesting `add_two_numbers(15, 27)`.

Server Execution
The plugin executes the PHP function and gets the result: `42`.

AI Response
The result `42` is sent back to the LLM, which generates the final answer: “15 plus 27 equals 42.”

NotebookLM

A tool is defined by its structure and implemented by server-side code.

The Definition (JSON)

```
{
  "type": "function",
  "function": {
    "name": "add_two_numbers",
    "description": "Adds two integers together.",
    "parameters": { ... }
  }
}
```

Crucial for the AI. This is how it knows *when* to use the tool.

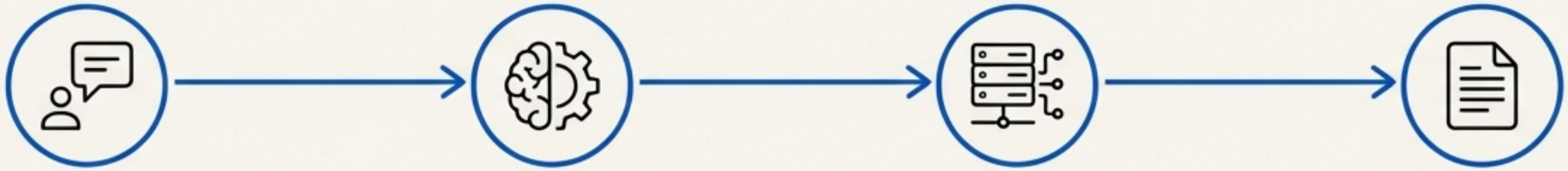
The Implementation (PHP)

```
<?php
function add_two_numbers($a, $b) {
  // Adds two integers and returns the sum.
  return $a + $b;
}
?>
```

The tool definition (sent with the API request) tells the AI *what* the tool can do.
The PHP function on your server is the code that actually *does* it.

NotebookLM

Tools can also connect the AI to real-time data sources.



User Prompt

“What’s the weather like today?”

Tool Selection

Chooses `get_weather()` tool.

Server Execution

Calls weather API. Result: `23°C`.

Final Answer

“Today the temperature is 23°C.”

By equipping the AI with a `get_weather` tool, it can answer questions that go beyond its static training data, providing up-to-the-minute information.

NotebookLM

Security is paramount and must be implemented at every layer.

Layer 1: API Security

- OpenAI API key stored securely in WordPress options.
- Key is never exposed in client-side code.

Layer 2: WordPress Integration

- Uses WordPress nonces for CSRF protection.
- Checks user capabilities (`manage_options`) to restrict access.
- Sanitizes all inputs before processing.

Layer 3: Function-Level Hardening

- The tool functions themselves should have strict parameter validation.
- Limit the scope of what any single tool can do (e.g., read-only access to a database).

NotebookLM

Expanding the AI's toolbox is a straighforward, four-step process.

1. 

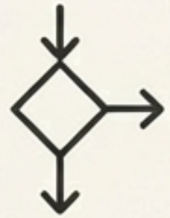
Create the PHP Function

Write the server-side code that performs the action.

2. { }

Add the Tool Definition

Add the JSON object describing the new function to the `$tools` array.

3. 

Handle the Function Call

Add a `case` to the switch statement to execute your new function.

4. 

Update the System Prompt

(Optional) Mention the new tool's capability to guide the AI.

NotebookLM

This pattern unlocks a new class of intelligent applications.

E-commerce

Check inventory or calculate shipping costs in real-time.

Content Management

Perform complex database queries for posts or users.

Business Apps

Fetch customer data from a CRM via an API.

IoT Systems

Query the status of a smart device or send it a command.

Customer Service

Search a knowledge base or summarize a support ticket.

Educational Tools

Solve math problems or provide interactive explanations.

NotebookLM

Six best practices for building robust and reliable AI tools.

1. Write Crystal-Clear Descriptions

The AI relies entirely on your description to know when and how to use a tool.

2. Validate Every Parameter

Never trust input. Sanitize and validate all arguments passed to your functions.

3. Implement Graceful Error Handling

Plan for failures. Return clear error messages to the AI so it can inform the user.

4. Enforce the Principle of Least Privilege

A tool should only have the permissions it absolutely needs to do its job.

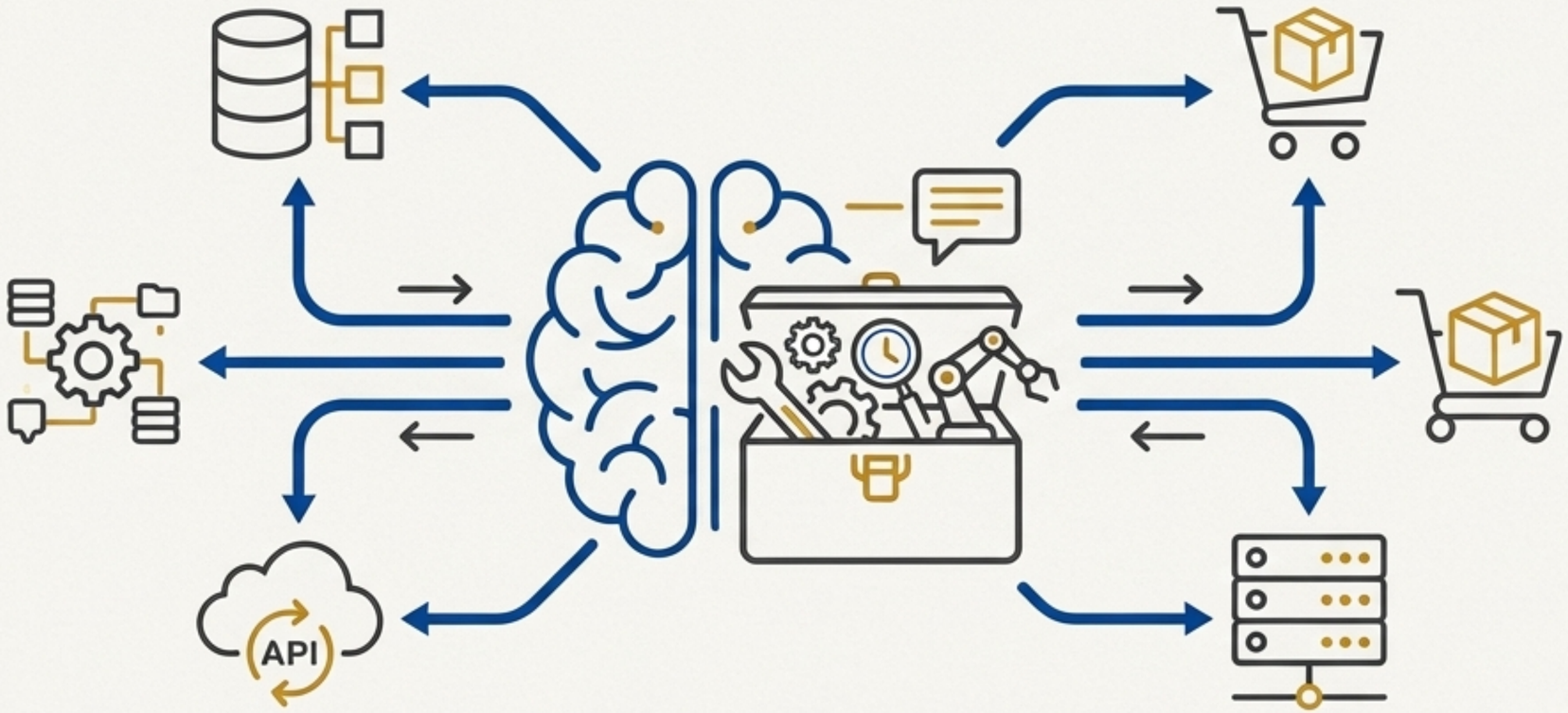
5. Document Everything

Thoroughly comment your tool functions and their expected inputs/outputs.

6. Test Extensively

Test your tools with a wide variety of inputs, including edge cases and invalid data.

NotebookLM



Tool Use transforms the AI from a simple knowledge base into an active collaborator.

The ability to connect language models to external functions is not just an incremental feature; it is a fundamental evolution. It enables the creation of dynamic, responsive, and genuinely useful AI agents that can participate in workflows, access live data, and take action in the digital world. The concepts in this plugin are the building blocks for that future.

NotebookLM