

1. Introduction: n8n as the “Nervous System” for AI Agents



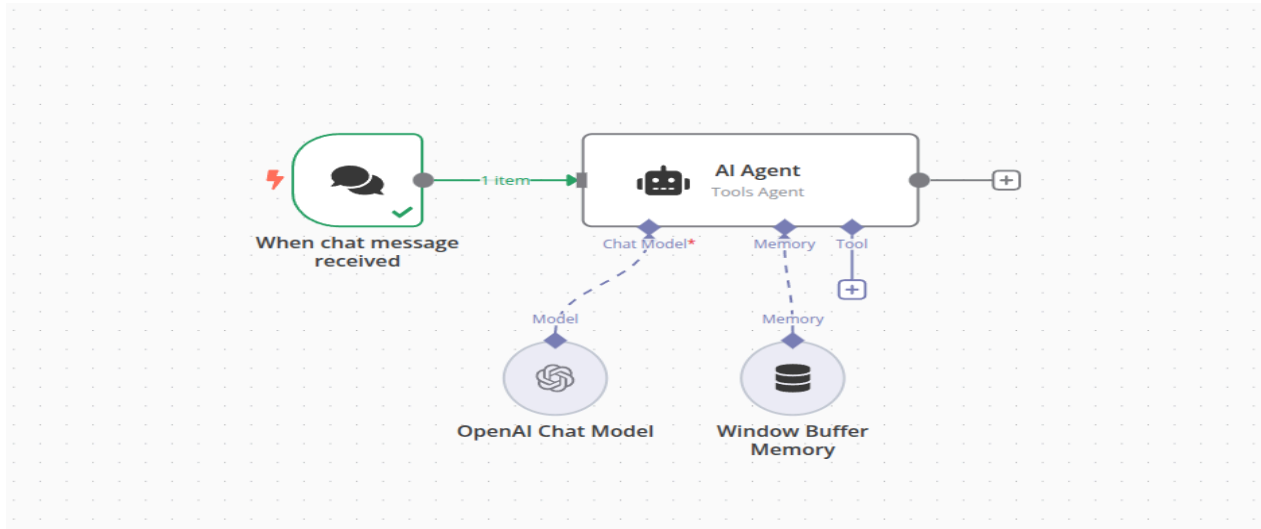
Welcome to the third lecture of our training program. After exploring the theoretical architectures of AI agents and advanced planning mechanisms in depth, it’s time to introduce the tool that will turn these abstract concepts into something practical and tangible: **n8n**.

n8n is a powerful, **open-source automation** tool that allows us to connect different applications and services and build complex **workflows** without writing a lot of code. We can think of n8n as the “nervous system” for our agents: it gives them the ability to interact with the outside world, execute actions, and collect observations in an organized and reliable way.

In this part, we will focus on the theoretical side of n8n—its philosophy, core components, and how it differs from other automation tools. We will explore how n8n can act as a **Tool Factory** for AI agents, enabling them to carry out tasks that go beyond pure language capabilities. This theoretical understanding will lay the foundation for using n8n effectively to build AI agents capable of **hybrid automation**, combining the power of LLMs with the flexibility of rule-based automation.

2. Understanding n8n: The Philosophy of Hybrid Automation

To fully understand n8n, we need to grasp the philosophy behind it: **hybrid automation**, which combines the best of two worlds—LLM intelligence and rule-based automation flexibility.



2.1. What is n8n?

n8n is a **workflow engine** that lets you automate tasks by connecting services and applications. It stands out because it is:

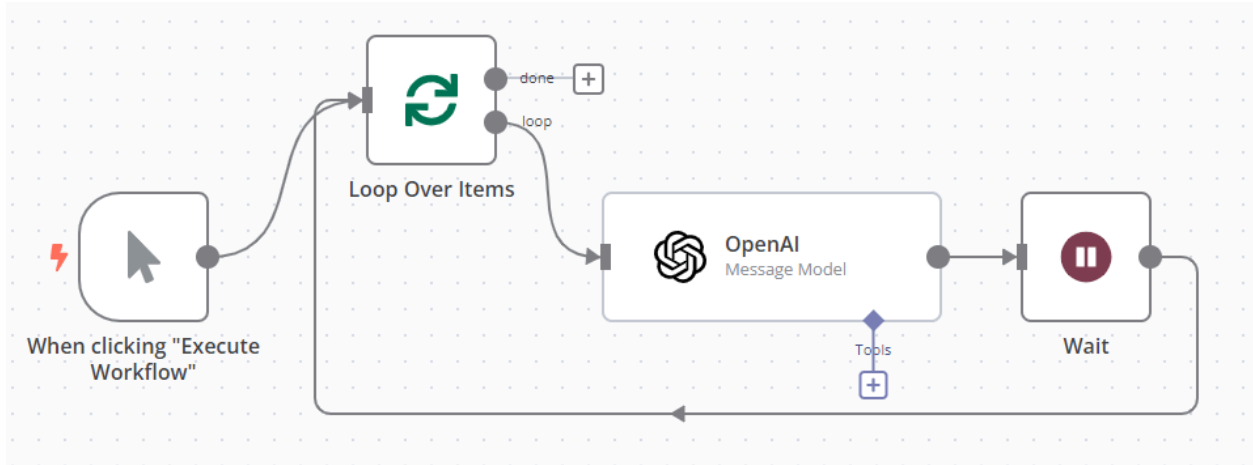
- **Open-source:** Gives you full control over your data and allows customization to fit your needs.
- **Visual:** You build workflows by dragging and dropping **nodes** and connecting them.
- **Extensible:** Supports hundreds of ready-made integrations, and you can also build **custom nodes**.

2.2. n8n in the Context of AI Agents

In AI agent systems, n8n plays a critical role as both an **Orchestrator** and a **Tool Factory**.

2.2.1. n8n as an Orchestrator

- **Task coordination:** n8n can coordinate a sequence of complex actions requested by an AI agent. Instead of the LLM trying to execute every step itself, it can delegate tasks to an n8n workflow.
- **Error handling:** n8n provides strong mechanisms for **error handling** and **retries**, increasing agent reliability.



- **Scheduled execution:** n8n can run workflows on schedules, giving the agent the ability to act proactively.

2.2.2. n8n as a Tool Factory

- **Turning workflows into tools:** Any workflow built in n8n can be converted into a “tool” that the AI agent can call. This means the agent can access any service supported by n8n.
- **Tool descriptions:** n8n supports creating clear and precise descriptions for each tool (workflow), helping the LLM understand what it does and when to use it.
- **Standardized inputs/outputs:** n8n ensures tool inputs and outputs are consistent, making it easier for the LLM to interact with them.

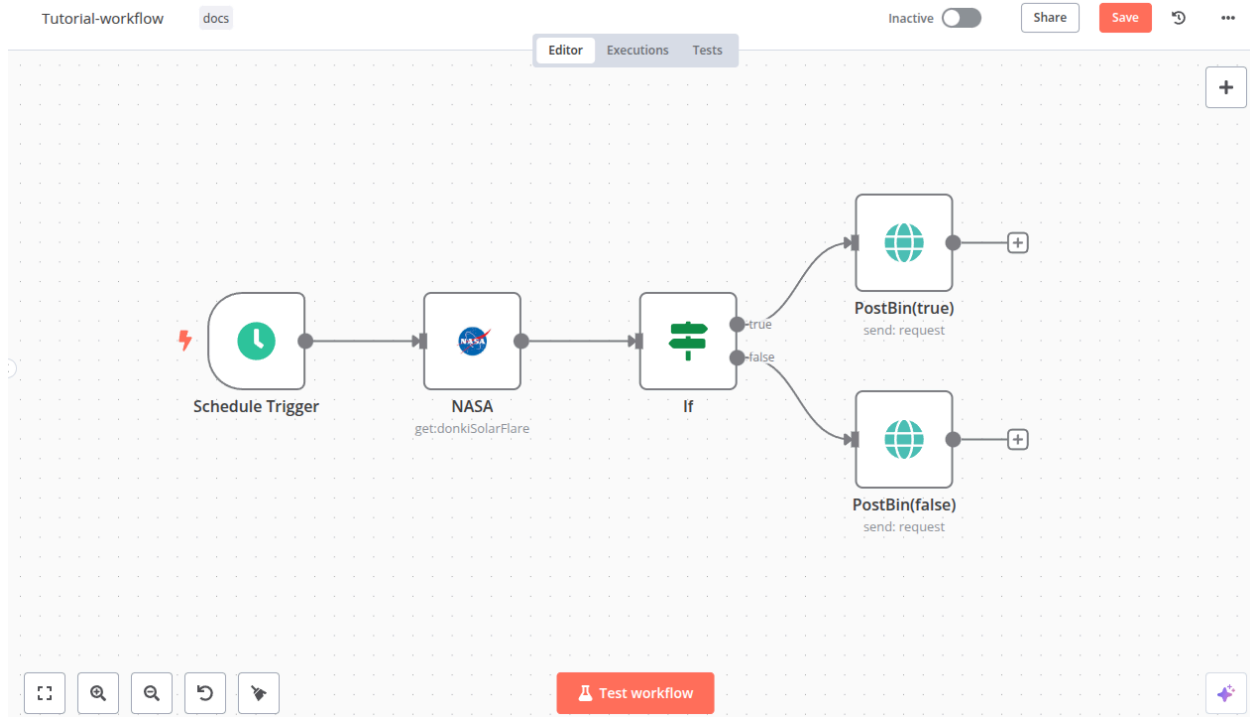
2.3. Comparing n8n with Other Automation Tools

Feature	n8n	Zapier/Make (Integromat)	Custom Code (Python/JS)
Data control	Full (open-source)	Limited (cloud service)	Full (but you manage infrastructure)
Flexibility	Very high (custom nodes, code)	Medium (templates)	Very high (anything possible)
Ease of use	Medium (visual UI + learning curve)	High (simple visual UI)	Low (requires coding skills)
Cost	Free (self-hosted), paid (cloud)	Paid (usage-based)	Code is free, hosting may cost
In agent systems	Ideal as orchestrator & tool factory	Can be used as tools, less flexible	Can be used as tools, more ops effort

Conclusion: n8n provides a unique balance between ease of use (visual interface) and flexibility (open-source, custom nodes), making it an excellent choice for building powerful tools for AI agents.

3. Core Components of Workflows in n8n

To build strong tools for agents using n8n, we need to understand the building blocks of any workflow.



3.1. Nodes

Nodes are the fundamental units of an n8n workflow. Each node performs a specific task.

3.1.1. Types of Nodes

- **Trigger Nodes:** Start the workflow by listening to events (e.g., webhook, schedule, new email).
 - Example: **Webhook** (receive agent requests), **Cron** (run at specific times)
- **App Nodes:** Interact with external services/apps (e.g., Google Sheets, Slack, Salesforce).
 - Example: **Google Sheets** (read/write data), **Slack** (send notifications)
- **Logic Nodes:** Control the flow (e.g., IF, Switch, Merge).
 - Example: **IF** (branch on condition), **Split In Batches** (batch processing)
- **Data Nodes:** Transform or manipulate data (e.g., Set, Code, Split In Batches).
 - Example: **Set** (assign new values), **Code** (run custom JavaScript)
- **Web Nodes:** Communicate with the web (e.g., HTTP Request, Webhook Response).

- o Example: **HTTP Request** (call any API), **Webhook Response** (reply to the agent)

3.1.2. Node Configuration

Every node has parameters that define its behavior. Parameters can be:

- **Static:** Predefined values (e.g., fixed URL).
 - **Dynamic:** Values pulled from previous nodes or passed as agent inputs.
-

3.2. Connections

Connections link nodes and define both data flow and execution order.

- **Data flow:** Data moves from one node to the next; outputs become inputs.
 - **Control flow:** Determines node execution order—sequential, parallel, or conditional.
-

3.3. Data

Data is the core of every workflow. In n8n, data is represented as **JSON objects**.

- **Inputs:** Nodes receive JSON objects from previous nodes.
 - **Outputs:** Nodes produce JSON objects for downstream nodes.
 - **Expressions:** Expressions allow you to access and transform previous node data.
 - o Example: `{{ $json.item.name }}` to access `name` from a prior JSON object.
-

3.4. Credentials

To connect to external services, n8n needs credentials (API keys, usernames, passwords).

- **Secure storage:** Credentials are stored securely and not exposed inside workflows.
 - **Reusability:** You can reuse the same credentials across multiple workflows.
-

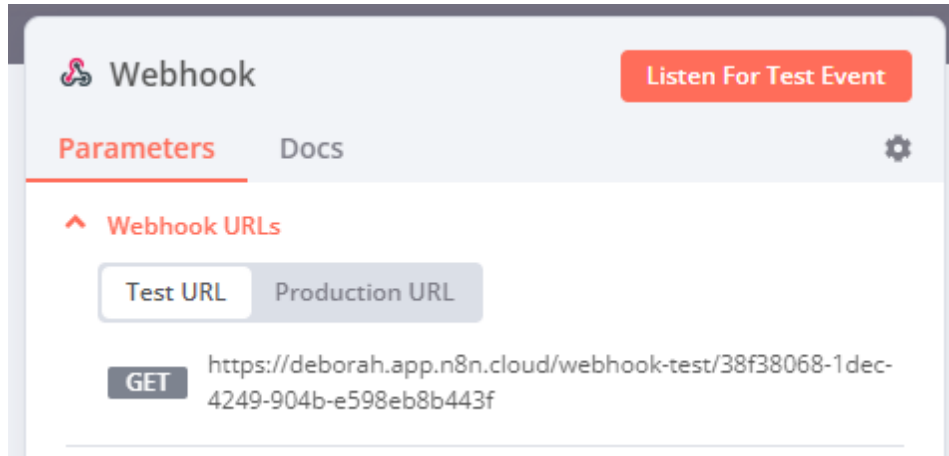
4. n8n as an API for AI Agents

The most important capability of n8n in agent systems is that it can act like a callable **API**.

4.1. Webhooks: The Agent’s Entry Point

A webhook is the primary trigger that allows an AI agent to call an n8n workflow.

- **Definition:** A unique URL that listens for incoming HTTP requests. When the agent sends a request, the workflow starts.



- **Inputs:** The agent can send data (parameters) in the HTTP request body, usually JSON.
- **Response:** The workflow can reply to the agent using a **Webhook Response** node.

4.2. Designing a Workflow as a “Tool” for an Agent

When building an n8n workflow intended to be used as an agent tool, consider:

1. **Clear inputs:** The workflow should expect explicit inputs (e.g., `query` for a search tool; `recipient` and `subject` for email tool).
2. **Standardized outputs:** The workflow should return outputs in a consistent format (usually JSON).
3. **Tool description:** Create a concise, accurate description for the tool—this is what the LLM uses to decide when/how to call it.

Example tool schema:

```
{
  "name": "search_web_n8n",
  "description": "Searches the internet using Google and returns results. Useful for answering questions about current events or general facts.",
  "parameters": {
    "type": "object",
    "properties": {
      "query": { "type": "string", "description": "Search query" }
    },
    "required": ["query"]
  }
}
```

4.3. n8n as an Abstraction Layer

n8n acts as an abstraction layer between the AI agent and external services:

- **Simplified interaction:** The LLM doesn’t need to know each service’s API details; it talks to a unified n8n interface.

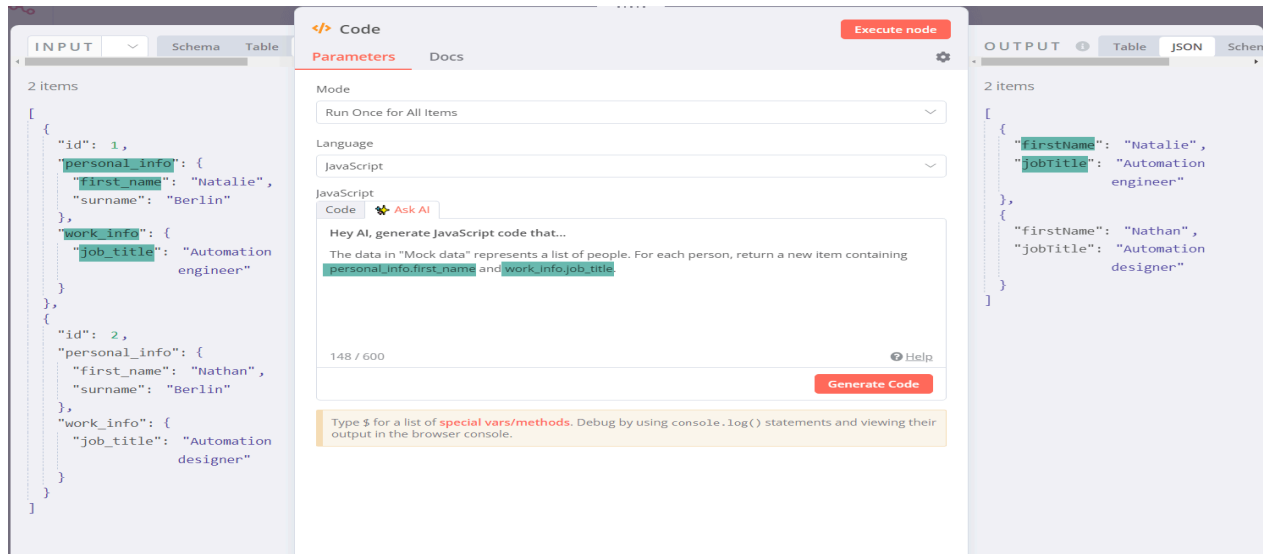
- **Complexity management:** n8n handles authentication, data formatting, and error handling, reducing the burden on the LLM.
- **Flexibility:** If a service API changes, you update the n8n workflow without needing to change the agent's prompt.

5. Advanced n8n Concepts for AI Agents

Beyond the basics, these advanced features significantly increase an agent's capabilities.

5.1. Code Node

The Code node lets you run custom JavaScript inside a workflow, giving unlimited flexibility for transformations and custom logic.



- **Uses:** Data transformation, complex calculations, using external JS libraries.
- **In agents:** The agent can request running specific logic, expanding what it can do.

5.2. HTTP Request Node

This node can send any HTTP request to any URL, meaning n8n can connect to virtually any web API.

- **Uses:** Calling external APIs, sending data to custom services, connecting to systems without native n8n integrations.
- **In agents:** Enables access to countless services and data sources across the web.

5.3. Sub-Workflows

You can call one workflow from another, enabling modular and reusable automation.

- **Uses:** Break down large workflows into reusable components.
 - **In agents:** Lets the agent call a sub-workflow as a tool, improving multi-step execution.
-

6. Summary

In this section, we explored n8n from a theoretical perspective and understood how it acts as a “nervous system” and a “tool factory” for AI agents. We learned that n8n is not just an automation tool—it is a bridge that connects LLM language ability to the real world through action execution.

Key points to remember:

- n8n is an open-source workflow engine that combines flexibility with usability.
- It serves as both an orchestrator and tool factory for AI agents.
- Workflows are built from nodes, connections, data, and credentials.
- Webhooks are the primary entry point for agents to call workflows.
- Tool workflows need clear inputs, standardized outputs, and accurate descriptions.
- n8n functions as an abstraction layer that simplifies integration with external services.
- Code, HTTP Request, and sub-workflows are advanced features that strengthen agent capabilities.