

# Natural Language Processing

Exercise – Summer term 2026

`klara.gutekunst@uni-kassel.de`

# Organization

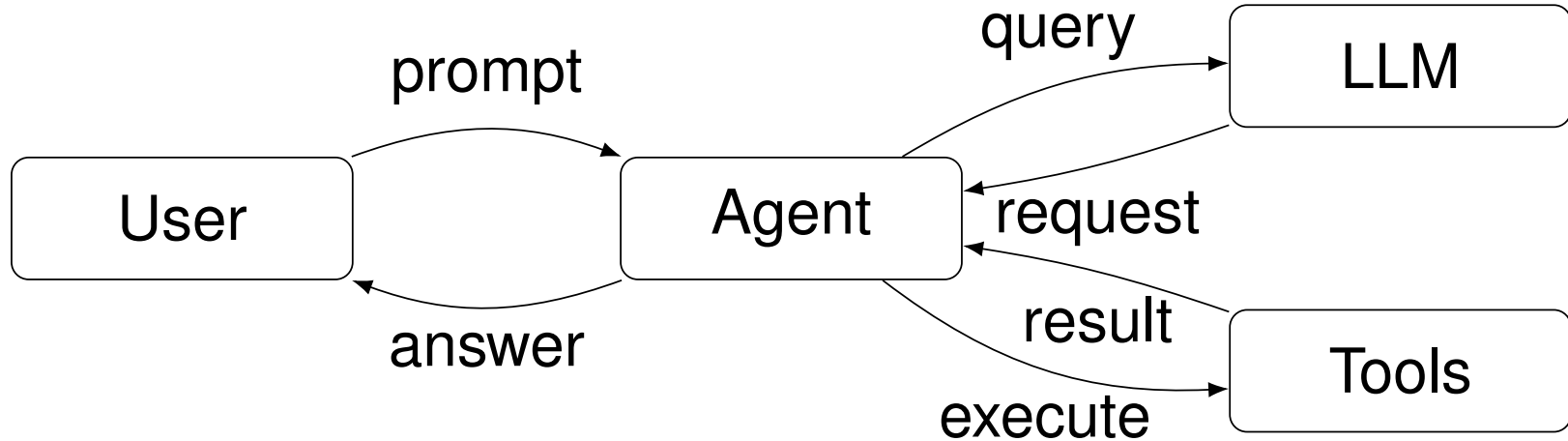
## Studienleistung

### Programming Project: **Forensic Linguist Agent**

- ❑ Develop a modular agent system for forensic authorship analysis.
- ❑ Focus on *idiolectal style*: how a text is written rather than what it is about.
- ❑ A coordinator agent selects and executes reusable linguistic analysis skills.
- ❑ Ensure a transparent and reproducible workflow with inspectable intermediate results and uncertainty.
- ❑ Evaluate the system on authorship datasets and compare agent-based workflows to simpler baselines.

# Agentic Workflow

What is an agent?



**Figure:** Simplified agentic workflow. A user prompt is processed by an agent that queries an LLM. The LLM may request tool calls, which are executed by the agent and returned as results. This loop continues until the agent produces a final answer.

# Studienleistung

## Progress Meeting

Checklist for the progress meeting:

- ❑ Form groups and establish contact with the supervisor?
- ❑ Connect to a backend LLM hosted by Webis? [[Lab Slides Part 2](#)]
- ❑ Develop a concept for the agentic workflow?
- ❑ Implement individual workflow components?
  - Set up a local n8n Docker instance [[Tutorial & GitHub Repo](#)]
  - Backend including API access to n8n and linguistic analysis tools
  - Database integration
  - ...
- ❑ Evaluate preliminary results and define next steps?

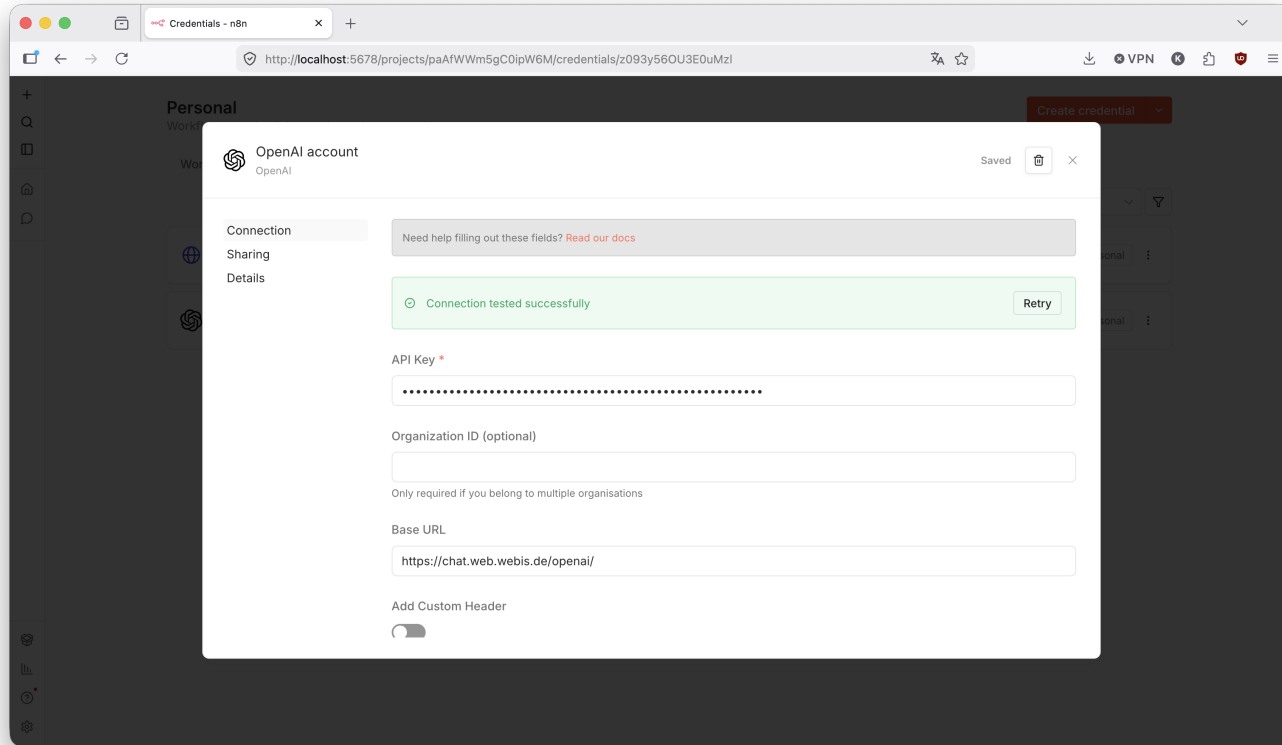
# Local n8n Docker Instance

## Tutorial: n8n Self-Hosted AI Starter Kit

1. Clone the [n8n Starter Kit repository](#) and follow the installation instructions.
2. After running `docker compose up`, access the n8n editor at <http://localhost:5678>.
3. Create a new workflow and explore different trigger nodes (“When chat message received” vs. POST “Webhook”).
  - For the webhook trigger, add a “Respond to Webhook” node.
4. Add an “OpenAI Chat Model” node (e.g., connect to Webis’ `qwen3-30b-a3b`), including authentication via the n8n API key.
5. Add a “Simple Memory” node.

# n8n Workflow

## Configure the LLM Backend



Configure the Webis-hosted LLM using the OpenAI-compatible endpoint [n8n documentation].

# n8n Workflow Example

## Chat Message Trigger

The screenshot displays the n8n workflow editor interface. The workflow is titled "custom-chat-template" and is located at the URL `http://localhost:5678/workflow/0QsBejAcwjboxuRo`. The workflow consists of the following nodes:

- When chat message received**: The trigger node, which sends 1 item to the AI Agent.
- AI Agent**: The central node, which receives 1 item from the trigger and sends 1 item to the next node.
- OpenAI Chat Model (Web)**: A node that receives 1 item from the AI Agent and sends 1 item to the Simple Memory node.
- Simple Memory**: A node that receives 1 item from the OpenAI Chat Model and sends 2 items to the AI Agent.

The workflow is currently in the "Editor" view. Below the workflow diagram, there is a chat interface with the following messages:

- Input: "The number is 7. What day is tomorrow?"
- Output: "Tomorrow is Wednesday, May 12, 2026. The number '7' does not affect the current date (2026-05-11 is a Tuesday), so tomorrow is **Wednesday** regardless of the number."
- Input: "What is the number?"
- Output: "The number is 7."

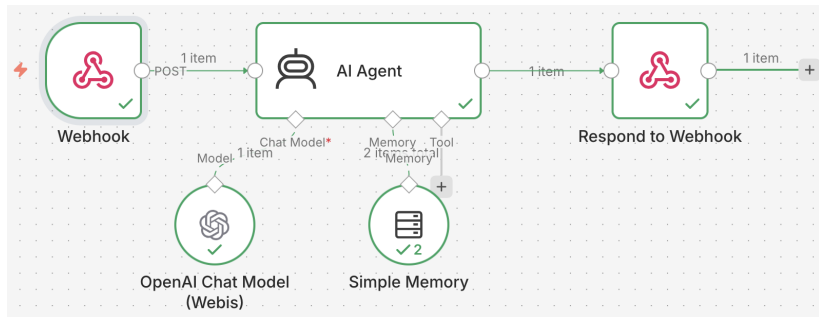
The chat interface includes a session ID "cbddf...", a "Clear execution" button, and a "Logs" panel on the right side. The logs show the following entries:

- Success in 10... ~108 To...
- When chat message r...
- AI Agent
- Simple Memory
- OpenAI Chat Mo...
- Simple Memory

Chat trigger and memory workflow example [n8n documentation].

# fragile

## n8n Workflow Example Webhook Trigger



(a) Webhook trigger [n8n documentation].

```
const response = await fetch(
  "http://localhost:5678/webhook-test/tmp",
  {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "test-custom-chat-key": "1234"
    },
    body: JSON.stringify({
      input: "What day is it today?",
      sessionId: "klara-test"
    })
  }
);
const data = await response.json();

console.log(data);
```

(b) Script to trigger webhook  
(run via `node script.js`).

# Studienleistung

## Discussion

- ❑ How do you plan to interact with the agentic workflow?
  - n8n chat interface, API requests from a custom UI, command-line interface, ...
- ❑ How will intermediate results be stored and inspected?
  - n8n execution history, databases, custom logging solutions, ...
- ❑ Which input format(s) will you support?
  - Plain text strings, .txt files, ...
- ❑ ...

# n8n Documentation

## Additional Resources

- ❑ [Agents vs. Chains](#) in n8n
- ❑ [Using Memory](#) in n8n
- ❑ [Working with Tools](#) in n8n
- ❑ [Integrating External APIs](#) in n8n