



Leibniz-Institut für
Astrophysik Potsdam



HOST
Hochschule Stralsund

Enhancing Data Workflows and Reproducibility with LLM Agents

How effective are LLMs with RAG and Agents in improving data analysis pipelines in terms of effectiveness and accuracy in astrophysics?

- Enrico T. L. Strauß / Bachelor Colloquium / 14.03.2025
- Supervisors: Prof. Dr. rer. nat. André Grüning, Dr. rer. nat. Arman Khalatyan
- <https://github.com/etlstrauss/bachelor-thesis-public>

Table of Contents

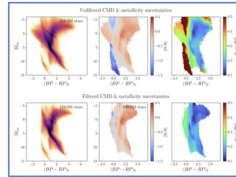
1. Introduction
2. Key Terms and Concepts
3. Methods
4. Results
5. Outlook / Limitations

Goal: The new art of interaction with computer using LLMs

- Chat with data
- Can LLM improving data analysis
- Can we improve reproducibility

Input from paper

Transferring spectroscopic stellar labels to 217 million Gaia DR3 XP stars with SHBoost, by Khalatyan, Anders, et al. (2024), aa51427-24, arXiv:2407.06963

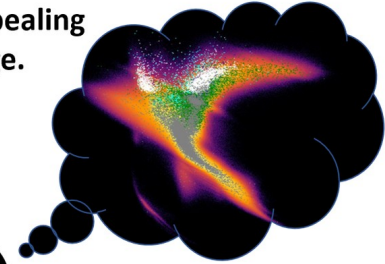


Results:

- Source code
- REANA workflow
- Data
- Plots

For Example

Let's create an appealing press release image.



User input:

Show an example of REANA pipeline for Starhorse datasets from 2024, CMD data is extracted and plotted interactively

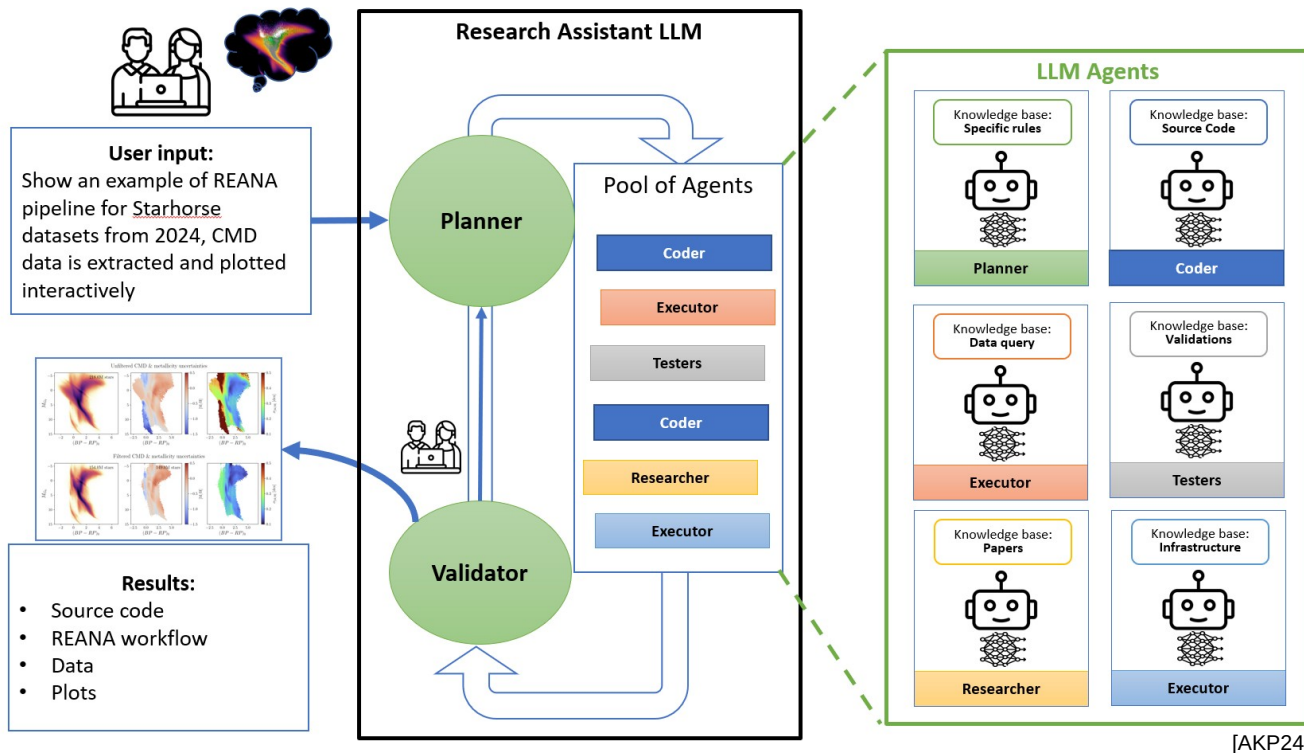


[AKG24]

Goals

- Ask Computer to generate or adjust data analysis pipelines
- Add the new information source to the knowledge base
- Validate results
- Run data analysis using the workflow engines
- Create a git repository and provide it as a final result

To achieve this



Can we use modern LLMs for this tasks

Benefits	Disadvantages
<ul style="list-style-type: none">• Faster• Simple setup and use• Probably more reliable• Less internal responsibility	<ul style="list-style-type: none">• Security concerns• Costs• Reproducibility• Full control of chain of thoughts• Limited customizations

Key Terms and Concepts

- Large Language Models (LLMs) [WALMcs24]
- Quantization [EgaAtEl24, LinAtEl24]
- GGML and GGUF Formats [IGGUF24, HFGG24]
- Retrieval-Augmented Generation (RAG) [GaoEtAl24]
- Agents [IWAA24, FLSA24]



Reproducible research data analysis platform

Flexible

Run many computational workflow engines.



Scalable

Support for remote compute clouds.



Reusable

Containerise once, reuse elsewhere. Cloud-native.



Free

Free Software. MIT licence. Made with ❤ at CERN.



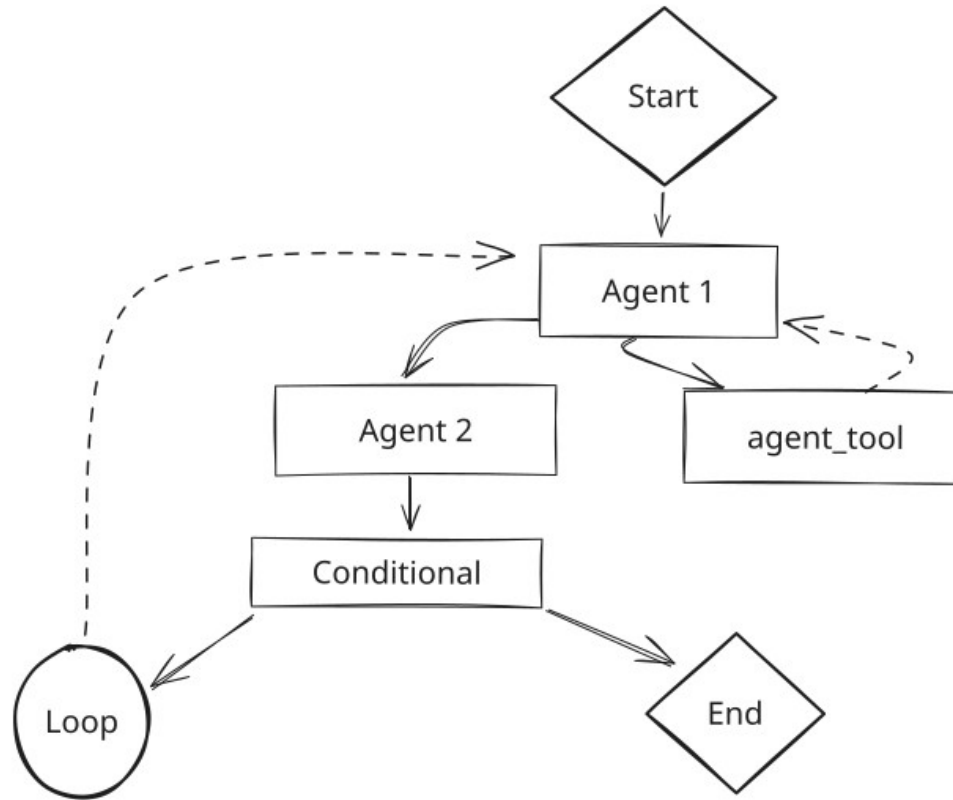
[REA25]

Workflow example: reana.yaml

```
inputs:
  files:
    - helloworld.py
workflow:
  type: serial
  specification:
    steps:
      - environment: 'docker.io/library/python:3.10-
bookworm'
        kubernetes_memory_limit: '100Mi'
        kubernetes_job_timeout: 60 # seconds
        commands:
          - python helloworld.py
```

[P4NR24]

What are Sequential Agents?



[FLSA24]

Methods

- Human evaluation as sub form of “Benchmarking”
- Store responses of LLMs in a database and grade them
- Compare runtime of different implementations

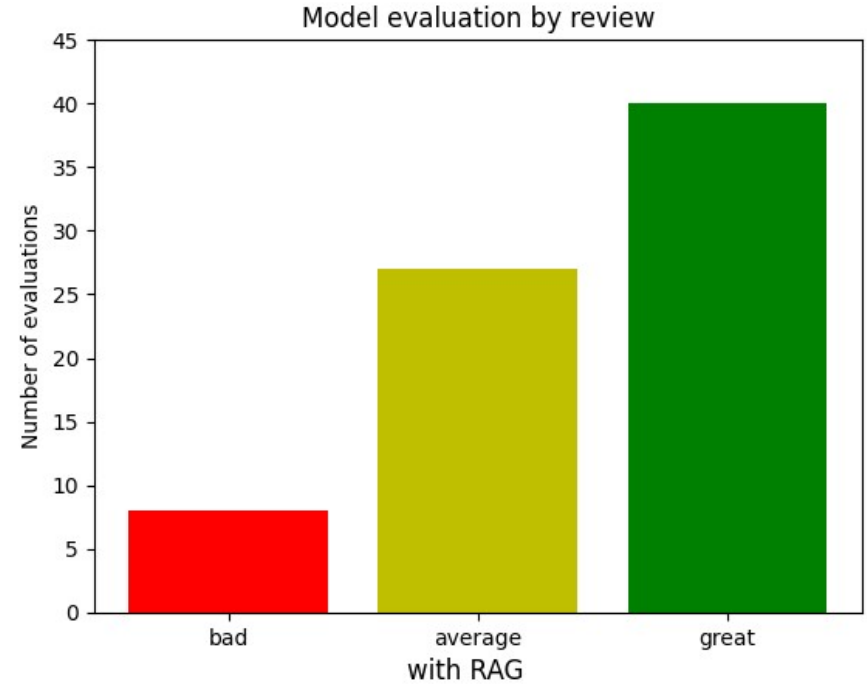
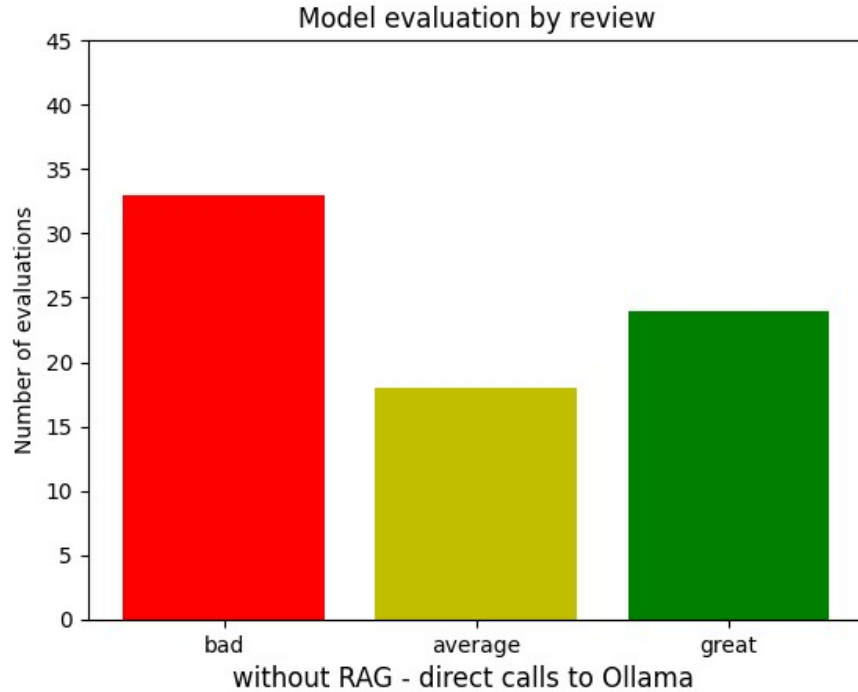
Results

- Selection of tools
- Final prototype
- Evaluation of final prototype

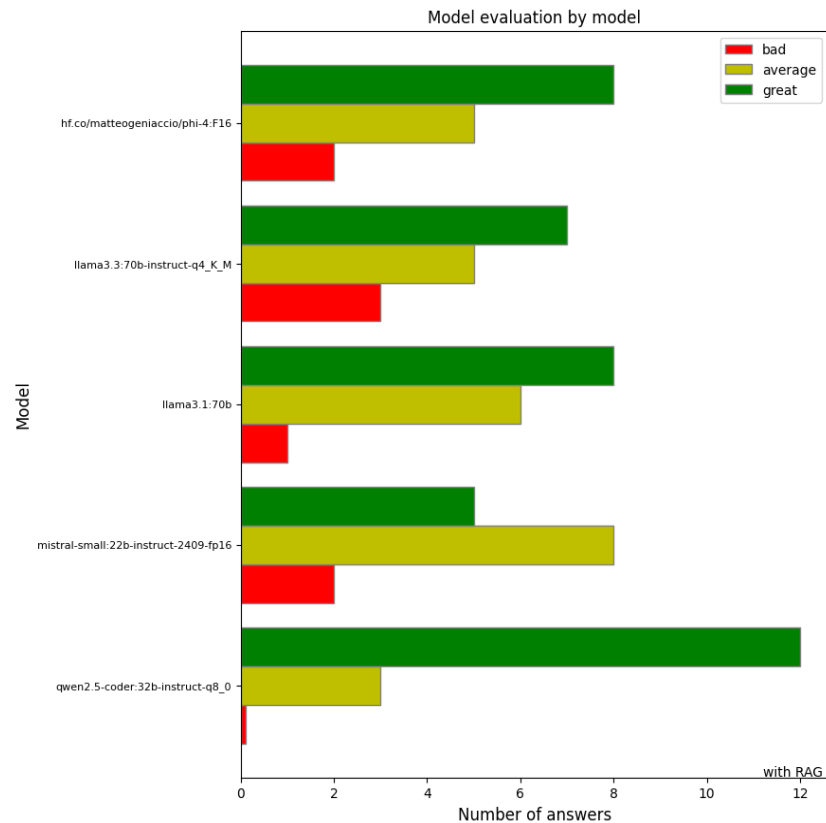
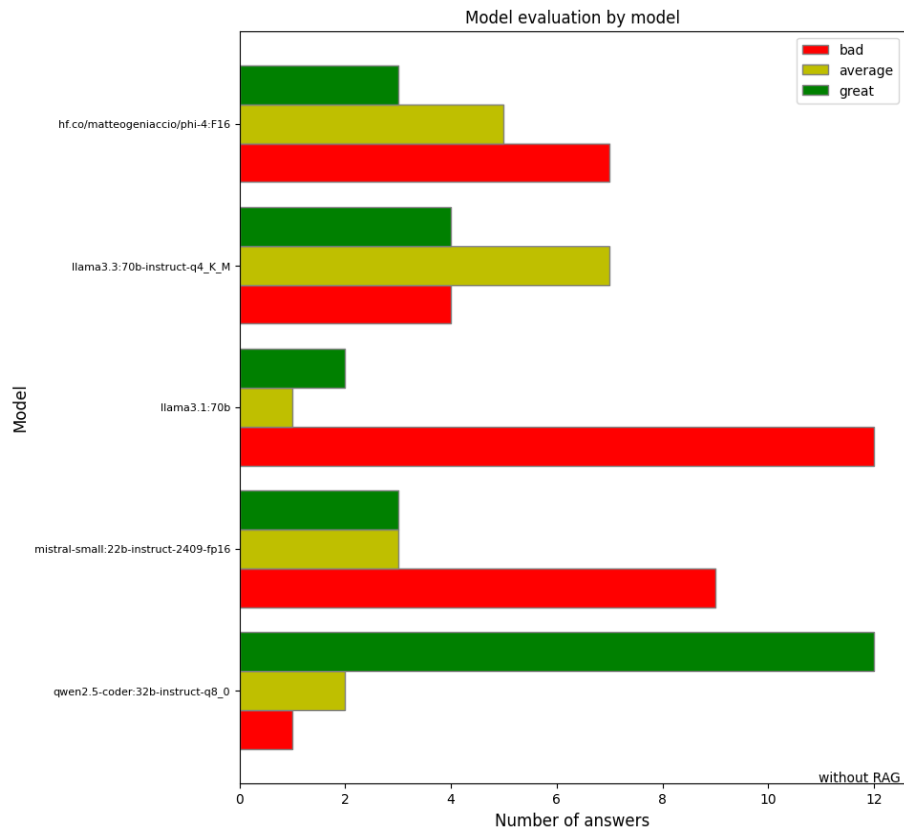
Open Weight Models

Model names / sizes [L2]		General information
hf.co/matteogeniaccio/phi-4:F16	32 GB	<ul style="list-style-type: none">• 15 practical and theoretical questions related to REANA• Each response evaluated by hand<ul style="list-style-type: none">• Matches the question very well• No standardize format as output• Did not represent the overall performance of this models• Division of the questions into 3 category• Own knowledge base
llama3.3:70b-instruct-q4_K_M	46 GB	
llama3.1:70b	44 GB	
mistral-small:22b-instruct-2409-fp16	48 GB	
qwen2.5-coder:32b-instruct-q8_0	37 GB	

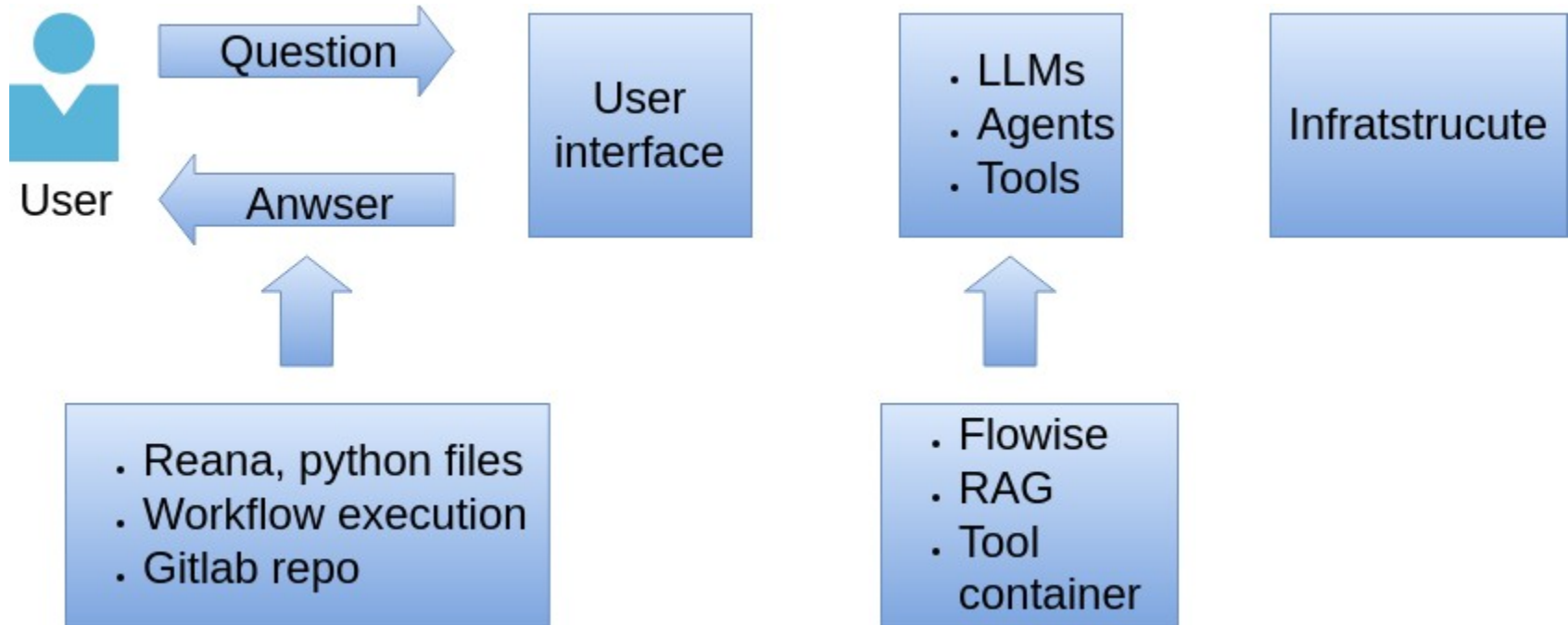
Model Evaluation: no RAG vs RAG (1/2)



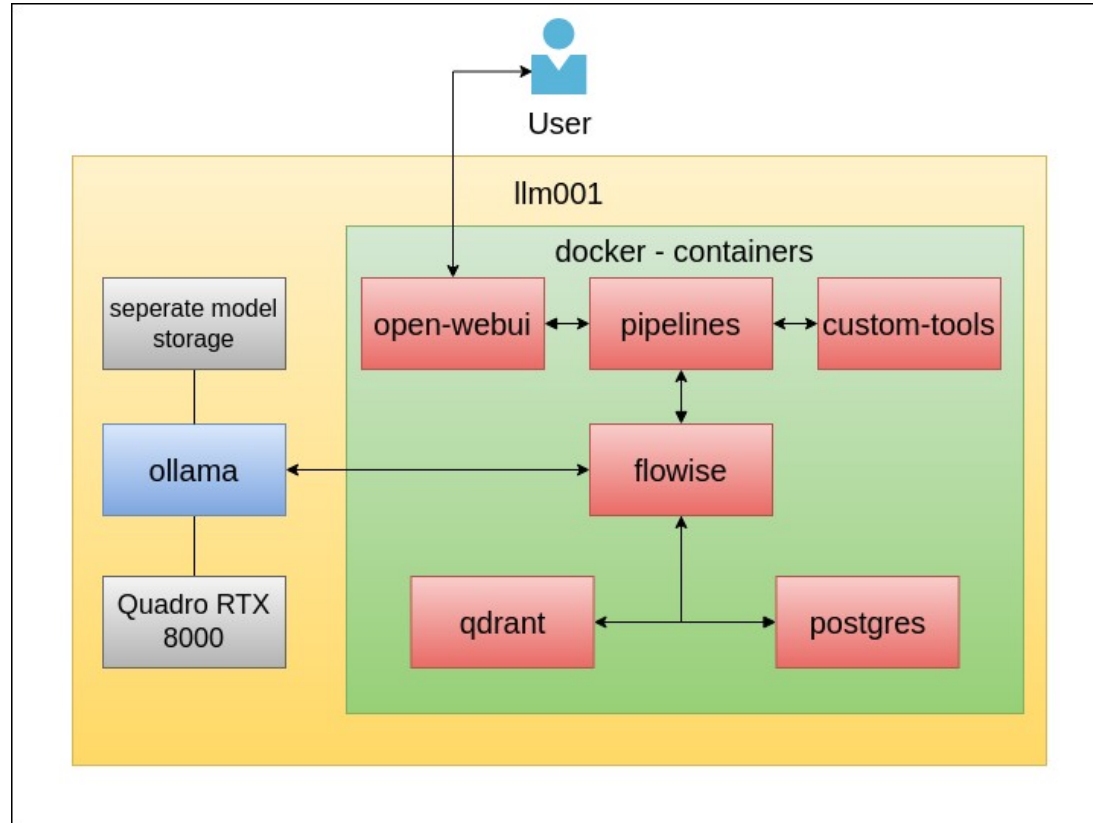
Model Evaluation (2/2)



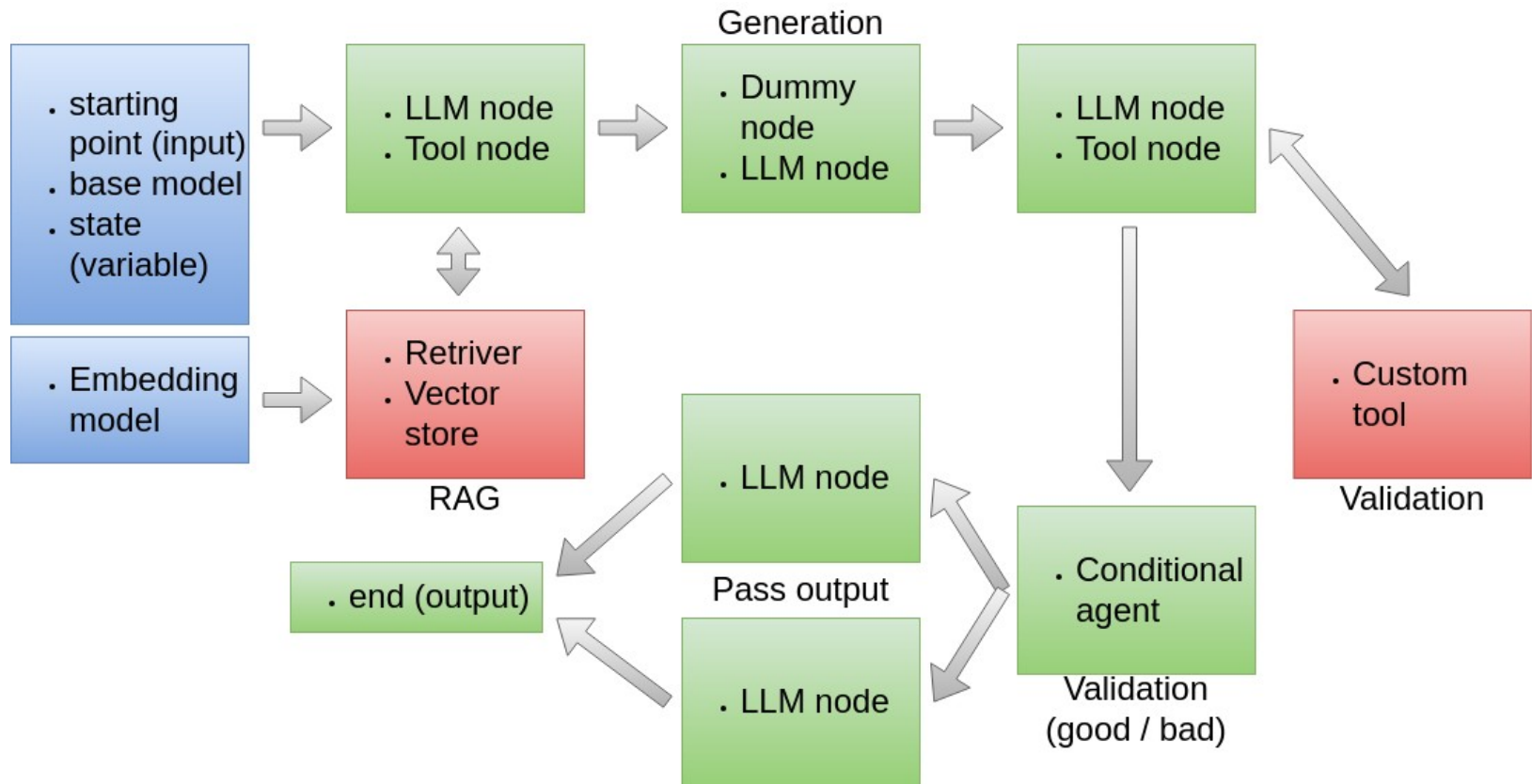
Application Components



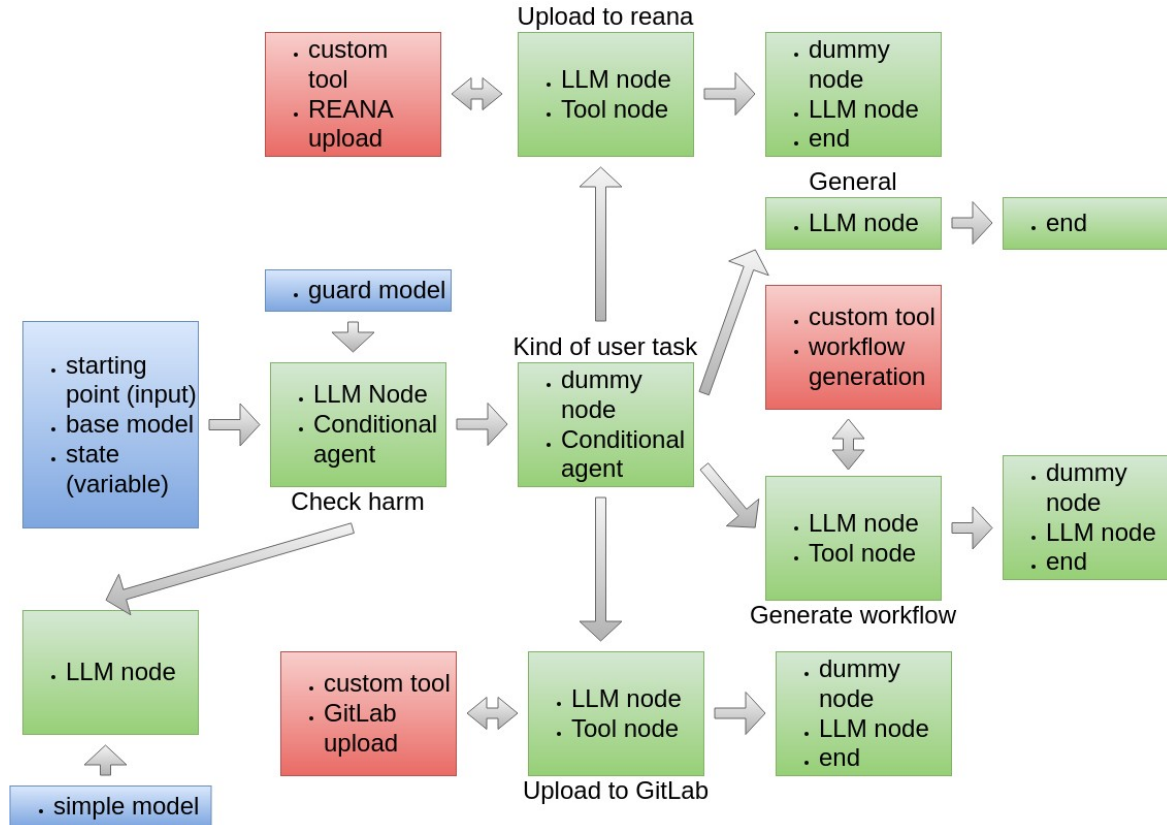
Infrastructure, Containers, Software Stack



REANA Generation Workflow



Open WebUI Endpoint Workflow



Open WebUI Pipeline

```
API_URL = "http://****:8000/api/v1/prediction/bdddddad9-3b09-44f1-af80-e6788a58d906"
headers = { "Content-Type": "application/json" }
if(len(messages) > 1):
    user_message = user_message + "\n History:\n" + messages[len(messages)-2].get("content")

payload = { "question": user_message, }
try:
    r = requests.post( url=API_URL, json=payload, headers=headers, stream=True, )
```

Custom Tools (1/2)

```
@app.route('/validate', methods=['POST'])
def validate():
    try:
        data = request.get_json()
        reana_file = data.get('reana_file')

        print(len(rcv.validate_reana_yaml(yaml.safe_load(reana_file))))

        if(len(rcv.validate_reana_yaml(yaml.safe_load(reana_file)))<1):
            return jsonify({'validation': True})
        else:
            return jsonify({'validation': False})
    except Exception as e:
        return jsonify({'validation': False})
```

Custom Tools (2/2)

```
const fetch = require('node-fetch');
const url = "http://****:5000/validate";

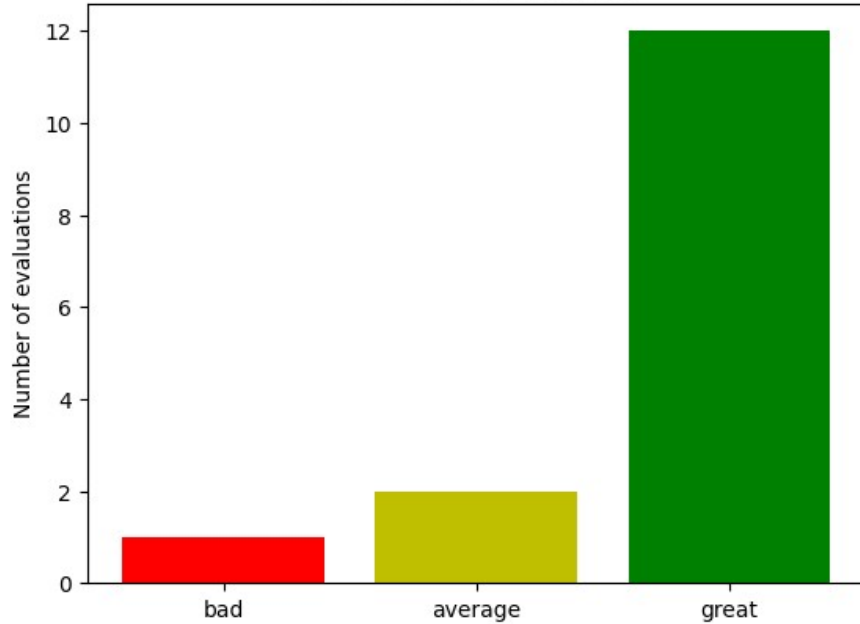
const data = {
  "reana_file": $reana_file,
};

const options = {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify(data),
};

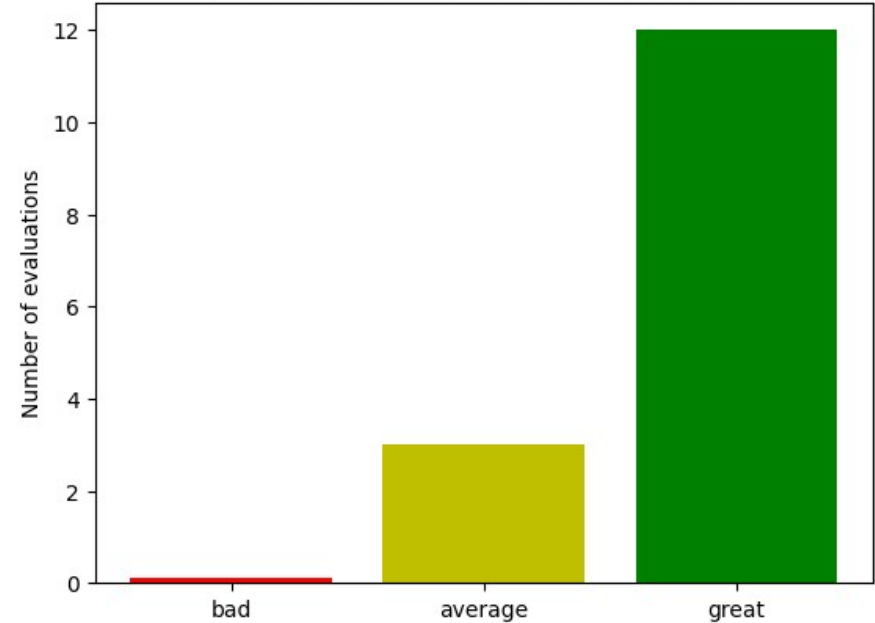
try {
  const response = await fetch(url,
options);
  const text = await response.text();
  return text;
} catch (error) {
  console.error(error);
  return " ";
}
```

Evaluation of final Prototype (1/3)

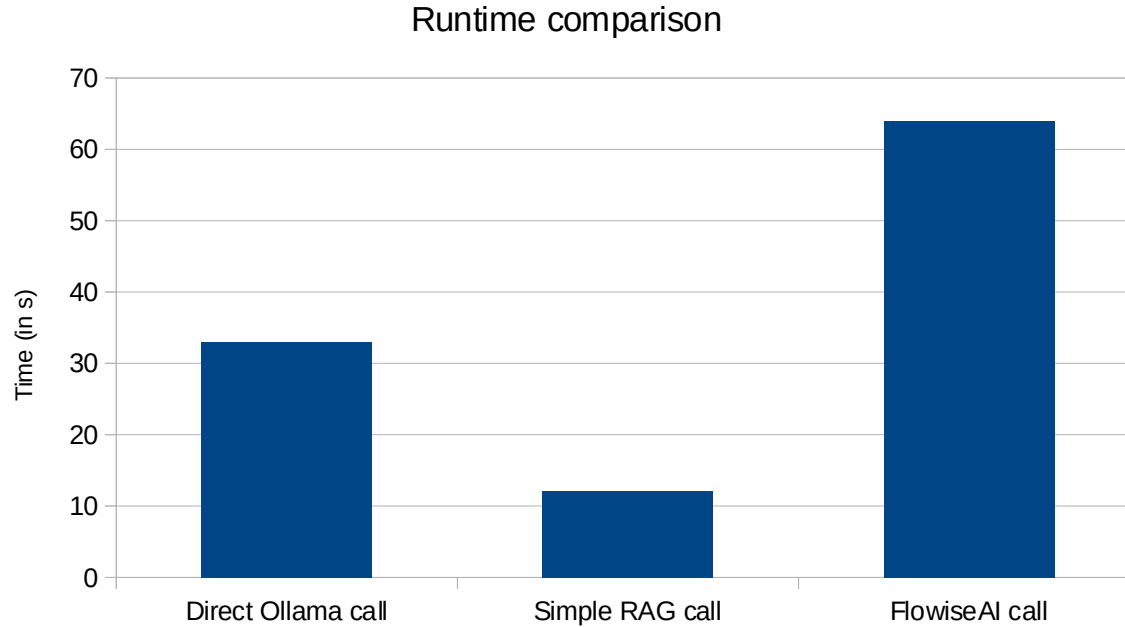
Direct call to Ollama



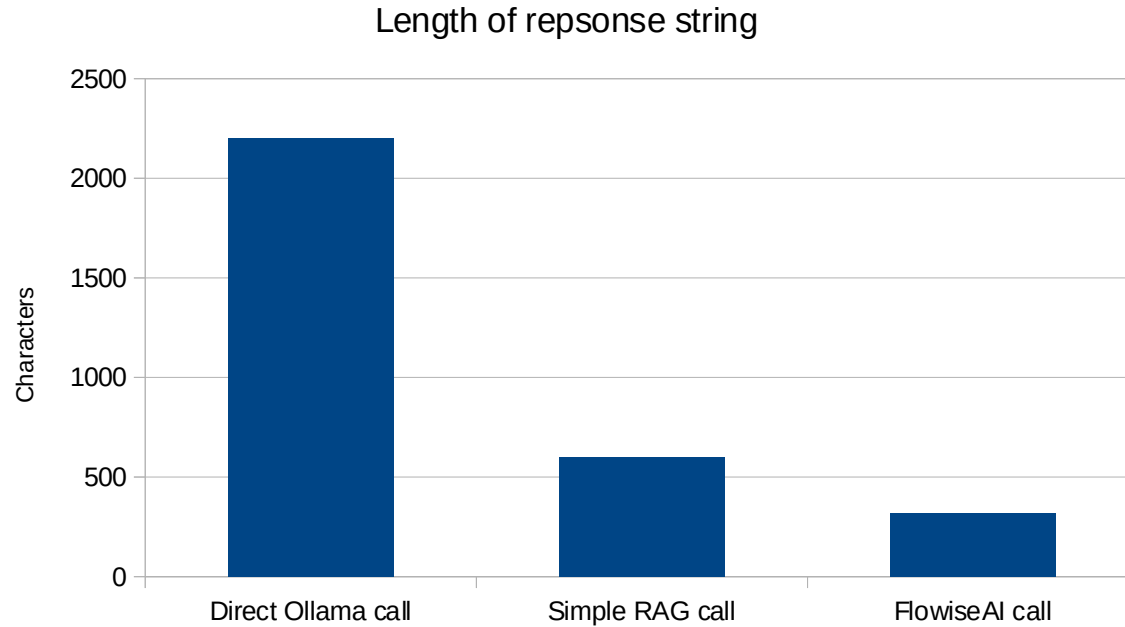
FlowiseAI call Open WebUI endpoint



Evaluation of final Prototype (2/3)



Evaluation of final Prototype (3/3)



Outlook and Achievements

Outlook	Achievements
<ul style="list-style-type: none">• Adaptive learning• Integrate existing infrastructure• Distributed AI architectures• Security mechanisms	<ul style="list-style-type: none">• Proof of concept<ul style="list-style-type: none">• Usable prototype• Storing all data locally• Partial control of chain of thoughts

Sources

[LvZF124] L. van Zyl, Flowise AI (2024) Tutorial - YouTube. [Online]. Available: <https://www.youtube.com/playlist?list=PL4HikwTaYE0H7wBxhvQqxYcKOKZ4O3zXh> (visited on 01/06/2025).

[OH24] Home — Open WebUI. [Online]. Available: <https://docs.openwebui.com/> (visited on 11/04/2024).

[AKG24] Dr. Arman Khalatyan, The problem of creating REANA workflows.

[AKP24] Dr. Arman Khalatyan, Solution approach for generating REANA workflows.

[REA25] <https://reana.io>

[WALMcs24] What are large language models (LLMs)? — Microsoft Azure. [Online]. Available:

<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-large-language-models-llms> (visited on 01/01/2025).

[EgaAtEl24] K. Egashira, M. Vero, R. Staab, J. He, and M. Vechev, Exploiting LLM Quantization, arXiv:2405.18137[cs], Nov. 2024. doi: 10.48550/arXiv.2405.18137. [Online]. Available: <http://arxiv.org/abs/2405.18137> (visited on 01/02/2025).

[LinAtEl24] J. Lin et al., “AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration,” in,

[IGGUF24] GGUF versus GGML — IBM. [Online]. Available: <https://www.ibm.com/think/topics/gguf-versus-ggml> (visited on 01/02/2025).

[HFGG24] Introduction to ggml. [Online]. Available: <https://huggingface.co/blog/introduction-to-ggml> (visited on 01/28/2025).

[GaoEtAl24] Y. Gao et al., Retrieval-Augmented Generation for Large Language Models: A Survey, arXiv:2312.10997[cs], Mar. 2024. [Online]. Available: <http://arxiv.org/abs/2312.10997> (visited on 11/18/2024).

[IWAA24] What Are AI Agents? — IBM. [Online]. Available: <https://www.ibm.com/think/topics/ai-agents> (visited on 01/02/2025).

[FLSA24] Sequential Agents — FlowiseAI. [Online]. Available: <https://docs.flowiseai.com/using-flowise/agentflows/sequential-agents> (visited on 01/02/2025).

Usage Example

Reana-Final ▾ +

...  

Please create an hello world reana workflow

OI Reana-Final

```
yaml Save Copy
1 # Hello World Workflow
2 inputs:
3   files:
4     - helloworld.py
5 workflow:
6   type: serial
7   specification:
8     steps:
9     - environment: 'docker.io/library/python:3.10-bookworm'
10       kubernetes_memory_limit: '100Mi'
11       kubernetes_job_timeout: 60 # seconds
12   commands:
13     - python helloworld.py

python Run Save Copy
1 # helloworld.py
2 print("Hello, World!")
```

Please upload it to reana!

OI Reana-Final

It looks like there was a small typo in the JSON response. The correct key should be "workflow_id" instead of "worflow_id". Assuming this is corrected, your REANA workflow is available at the provided workflow ID: `11m-gen-e1b97266-934a-40fb-9c92-c01ea9616f8b`.



+ Send a Message



?

Usage Example: REANA workflow monitoring interface, green-success

✓ **llm-gen-e1b97266-934a-40fb-9c92-c01ea9616f8b** #1

Finished 7 minutes ago

finished in 9 seconds
step 1/1

⋮

⚙️ Engine logs

>_ Job logs

📁 Workspace

📄 Specification

Step

reanadev-run-job-bf1f8191-e2f9-447... ▾

finished in 3 seconds

☁️ Kubernetes

🐳 docker.io/library/python:3.10-book...

\$ python helloworld.py

```
job: :  
  Hello, World!
```

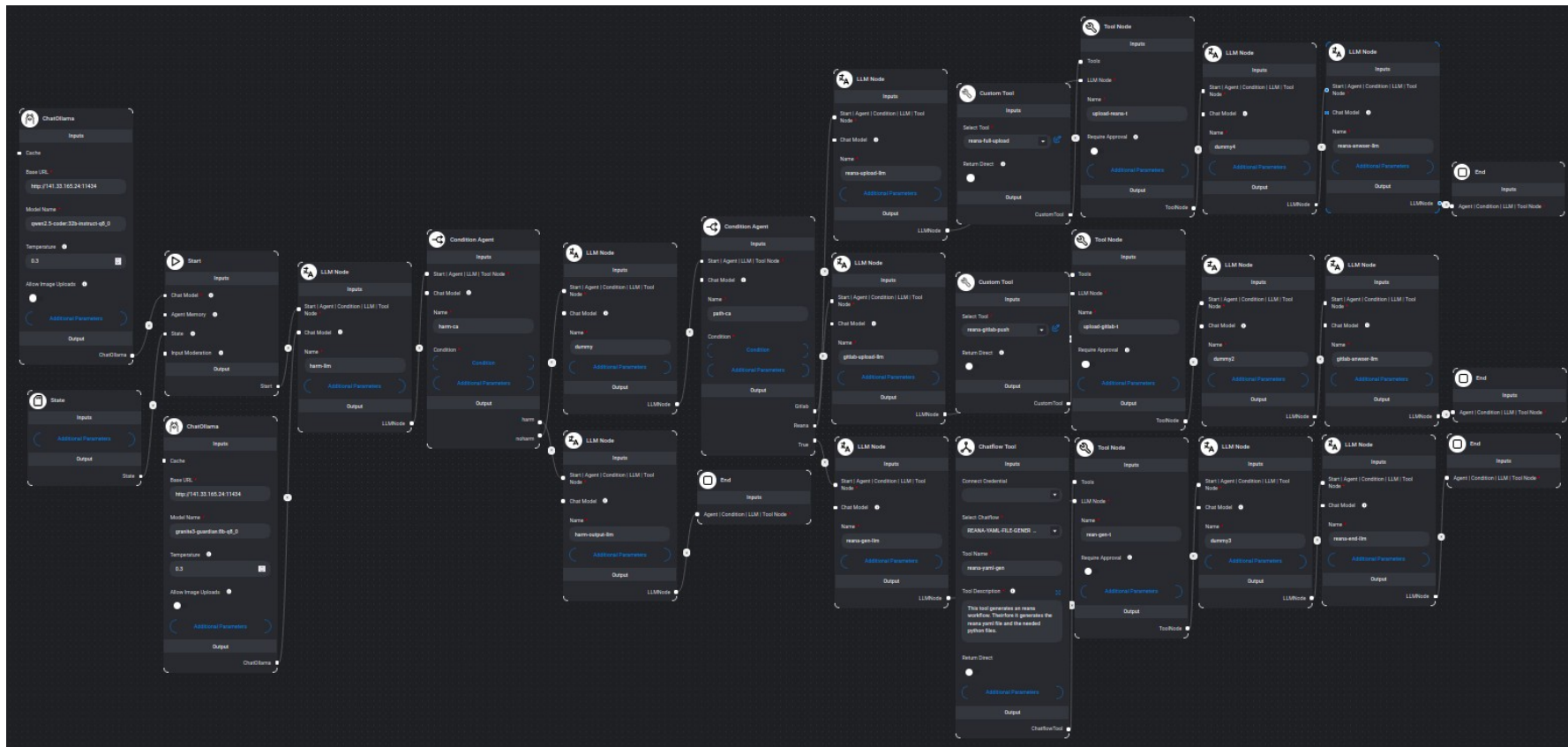
Completed

The screenshot displays a Reana workflow editor with a dark theme. The workflow is composed of several interconnected nodes:

- ChatOllama**: The starting point, configured with a base URL and model name.
- Ollama Embeddings**: A node that takes input and generates embeddings.
- Qdrant**: A vector database node for storing and retrieving documents. It includes fields for document, embeddings, and Qdrant server URL.
- Retriever Tool**: A tool node that interacts with the Qdrant database to retrieve relevant documents.
- LLM Node**: Multiple Large Language Model nodes used for processing and generating text. They are configured with names like 'reana-dummy', 'reana-node2', and 'reana-pass-llm'.
- Custom Tool**: A node for custom logic, including a 'Select Tool' dropdown.
- Condition Agent**: A node that evaluates conditions to determine the next step in the workflow.
- End**: The final output node of the workflow.

The workflow is connected by a series of lines representing the flow of data and control between these components.

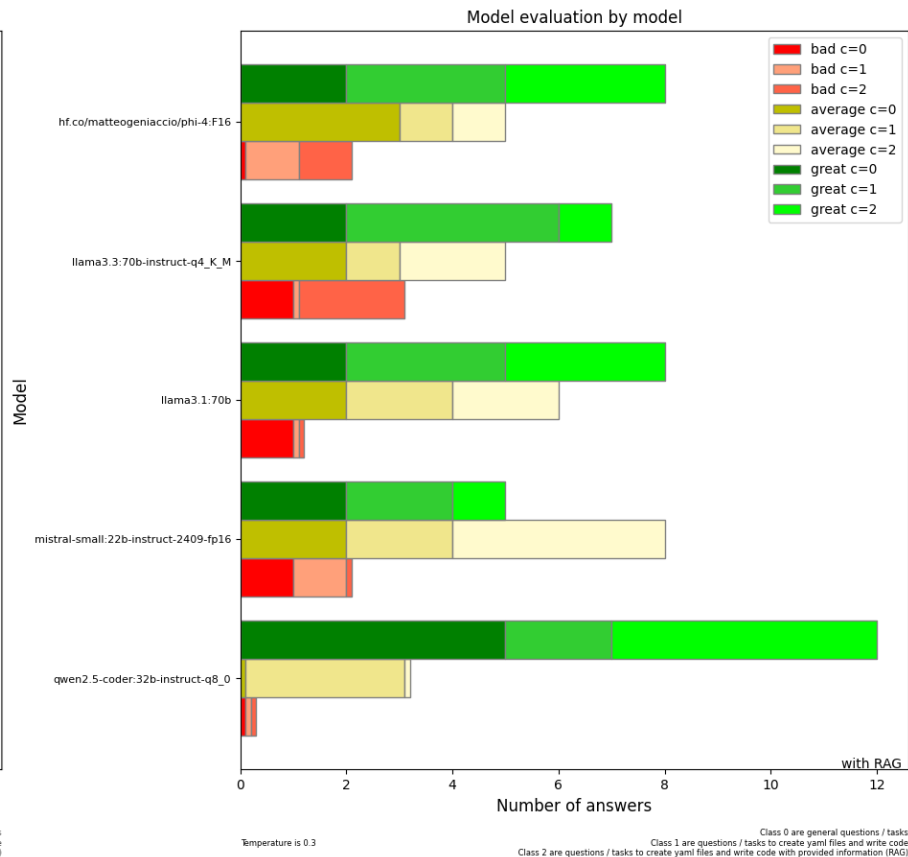
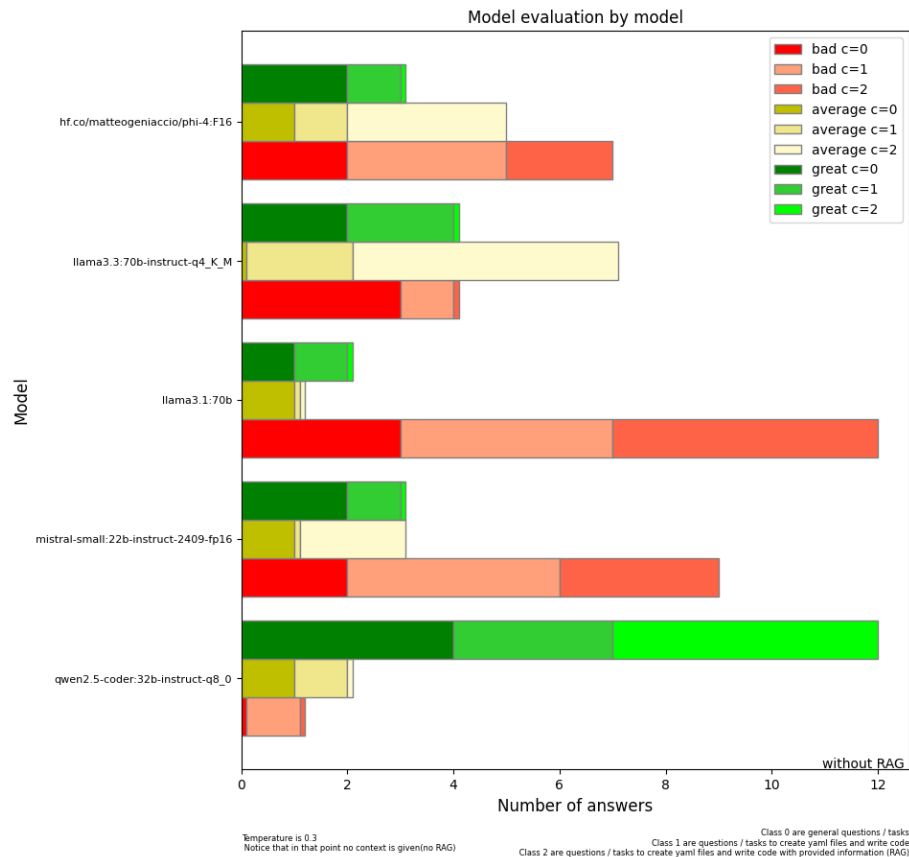
Open WebUI Endpoint Workflow



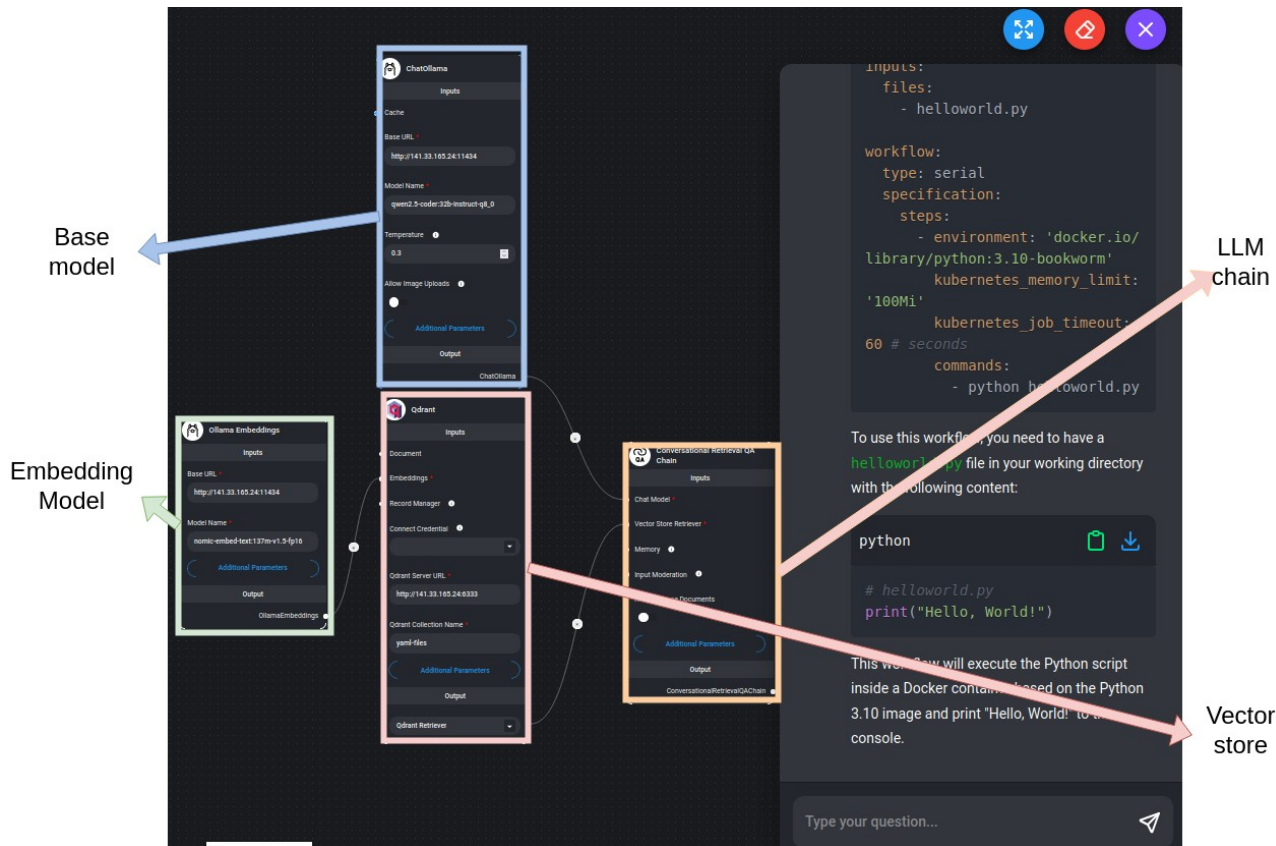
Evaluation of final Prototype (2/2)

	Time (in s)	Number of letters in response (round)
Direct Ollama call	33.04	2200
Simple RAG call	12.09	600
FlowiseAI call	64.03	320

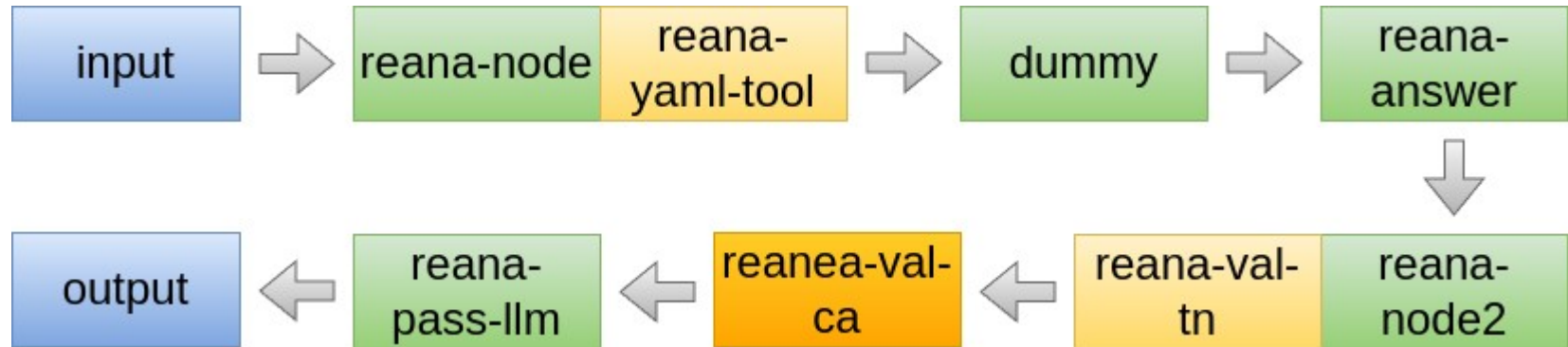
Model Evaluation



Model evaluation: FlowiseAI Workflow



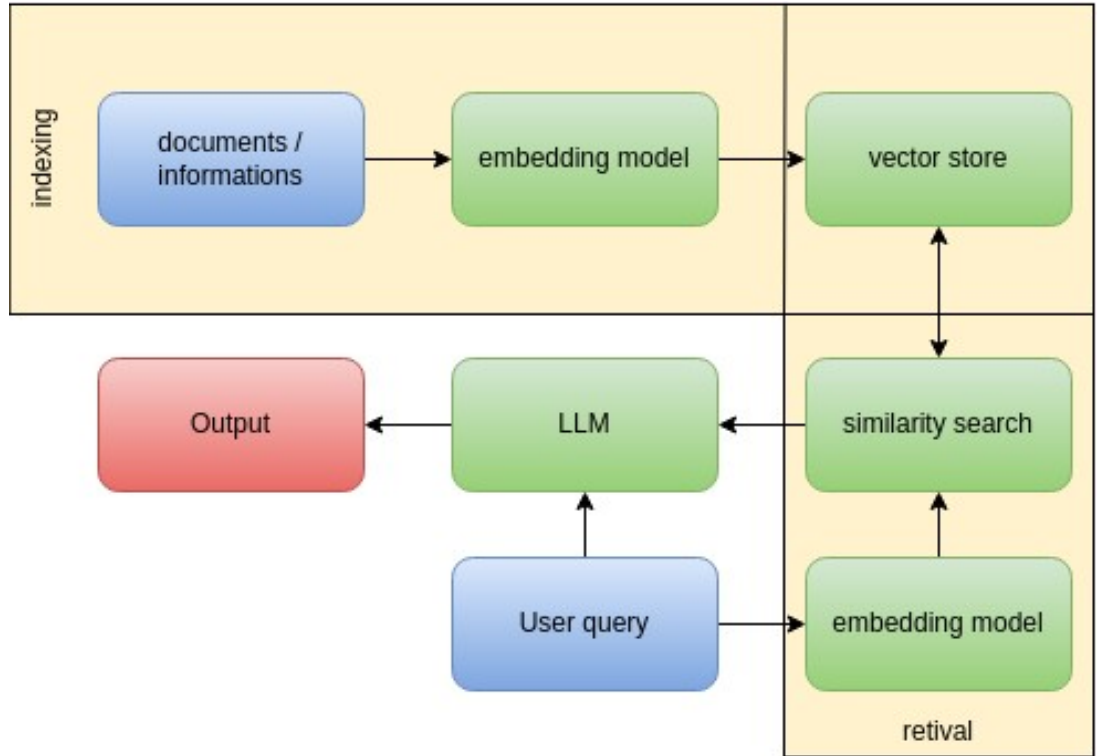
FlowiseAI Workflow: Example process



What is **R**etrival **A**ugmented **G**eneration?

Adding missing information into the LLM:

- fine tune(extrem expensive)
- **RAG**
 - Simple way to add missing information
 - Using small models
 - Knowledge base is manageable



[GaoEtAl24]