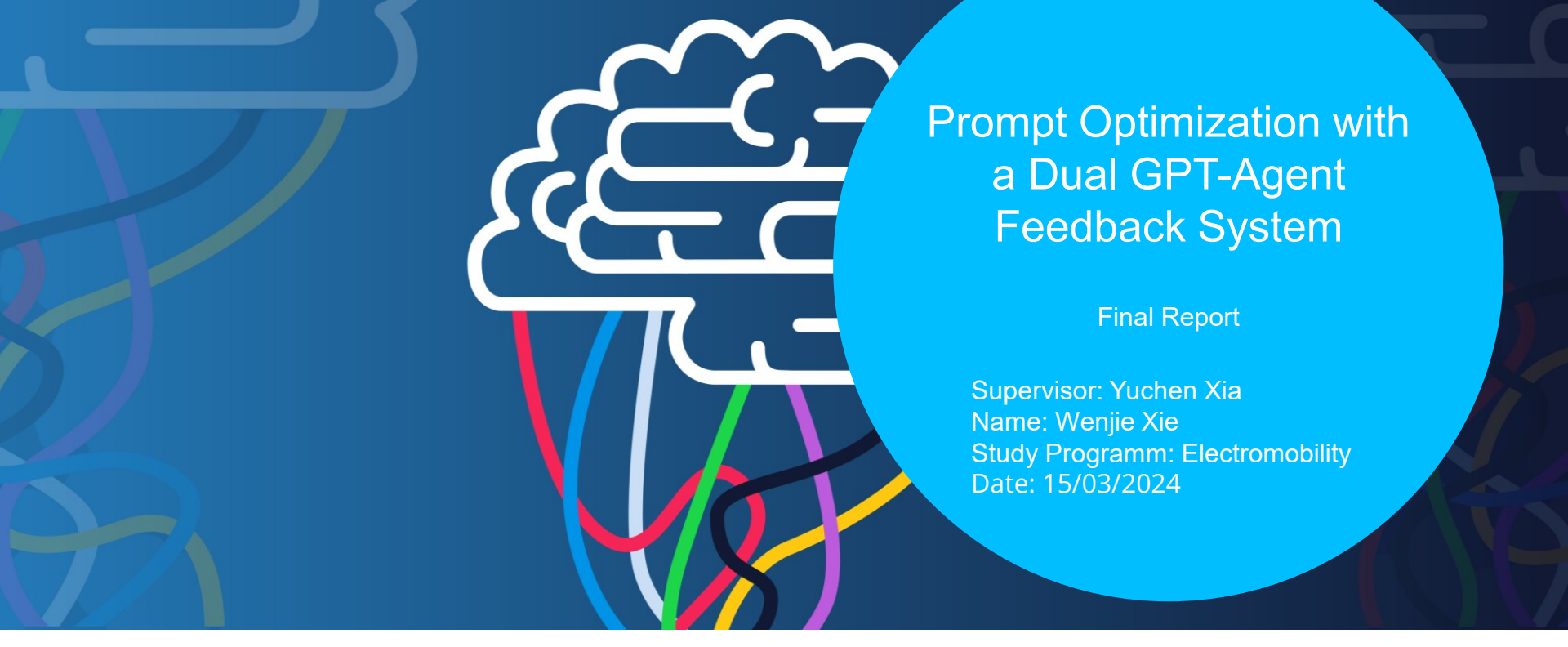
University of Stuttgart
Institute of Industrial Automation
and Software Engineering



Prompt Optimization with a Dual GPT-Agent Feedback System

Final Report

Supervisor: Yuchen Xia
Name: Wenjie Xie
Study Programm: Electromobility
Date: 15/03/2024

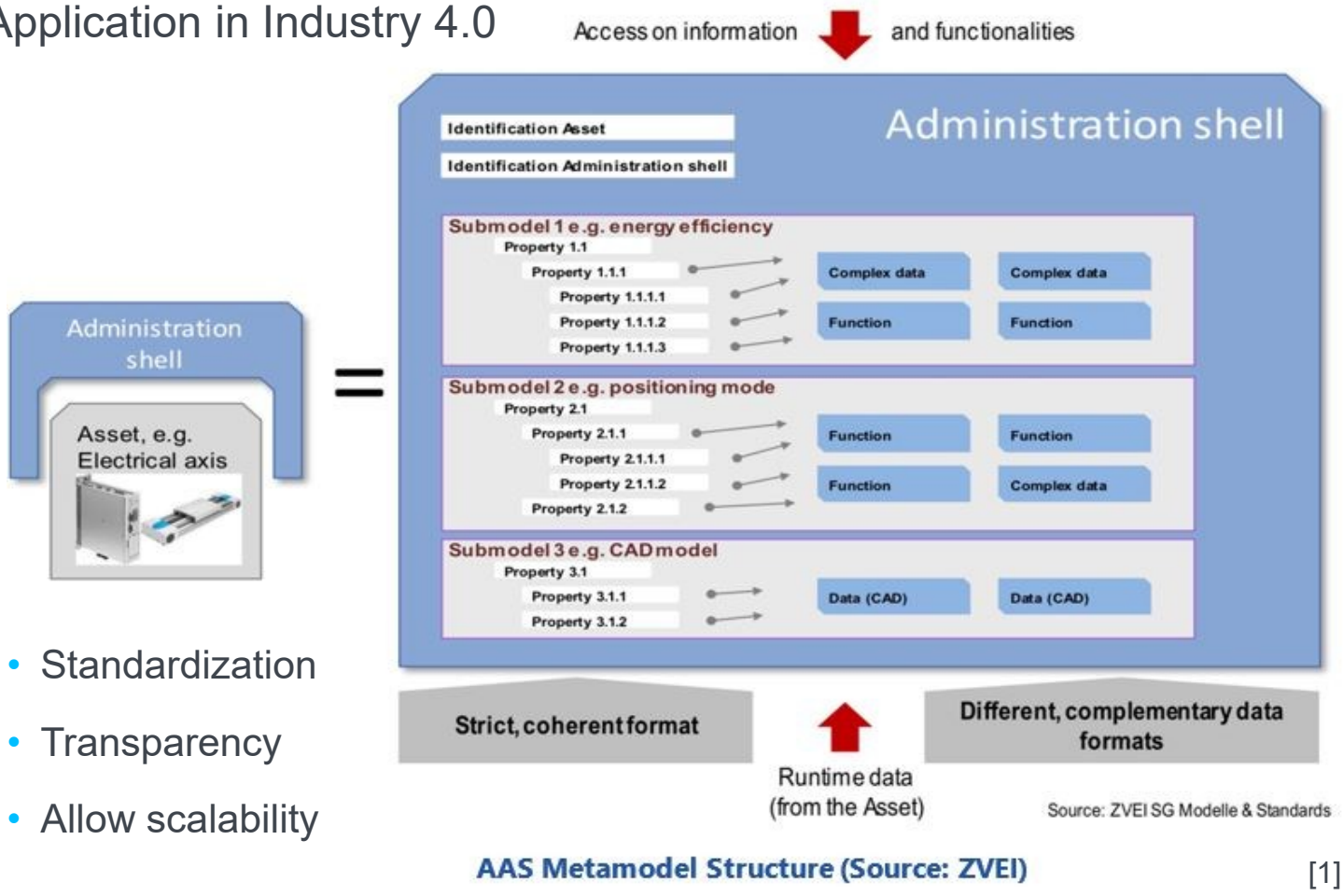
Agenda

- Motivation & Basis
- Conceptual Design
- Implementation
- Evaluation
- Summary and Outlook

1. Motivation & Basis

Motivation & Basis

Trend of AAS-Application in Industry 4.0



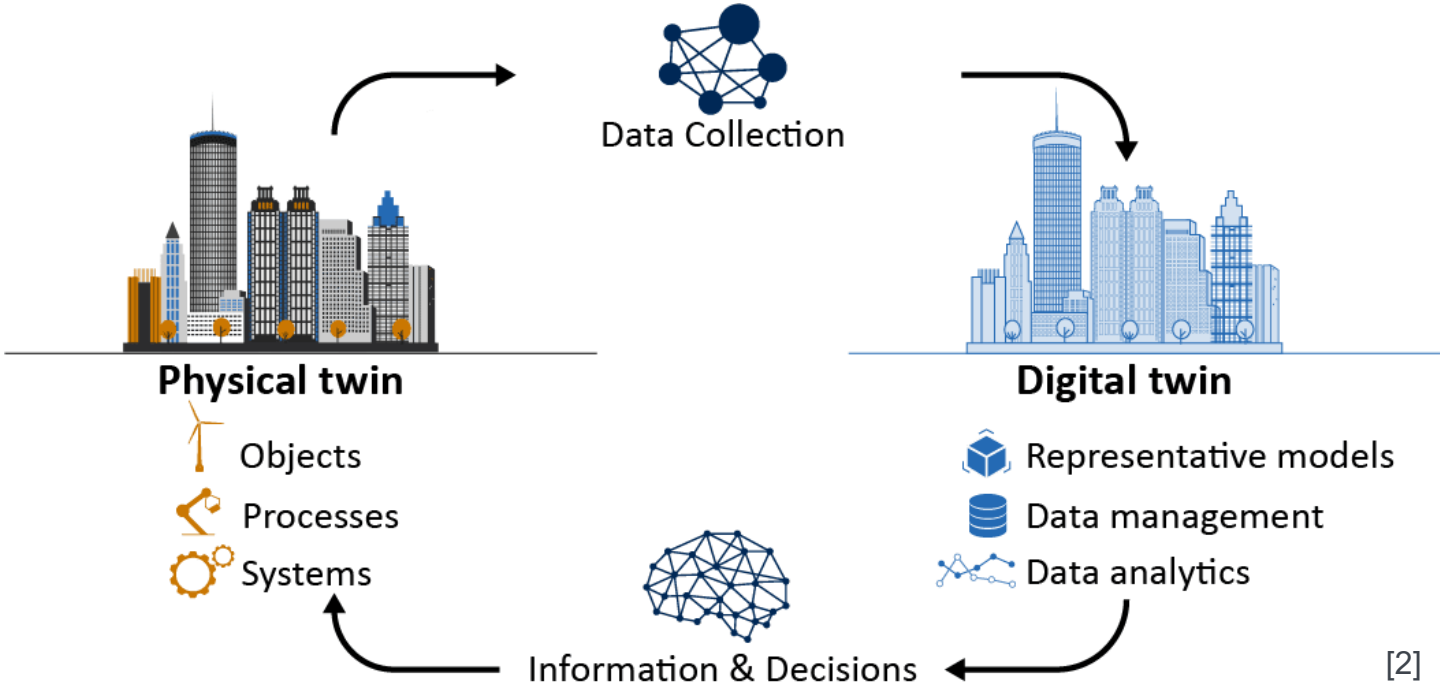
AAS Metamodel Structure (Source: ZVEI) [1]

Advantages

- Standardization
- Transparency
- Allow scalability

Motivation & Basis

Trend of AAS-Application in Industry 4.0



[2]

Disadvantages

- Complexity
- Cost
- Manual Data Management

Motivation & Basis

Asset Administration Shell (AAS)

Cable Housing screw, straight - 3,5-6,5m

Image is for illustration purposes only. Please refer to product description.

Identification

Category	Connectors
Series	Circular connectors M17
Type of hood/housing	Cable housing

Version

Size	M17
Shielding	Shielded
Version	Top entry
Locking type	Screw locking

Technical characteristics

Limiting temperature	-40 ... +125 °C
Number of relockings	≥500
Degree of protection acc. to IEC 60529	IP66 / IP67 screwed / locked condition
Clamping range	3.5 ... 6.5 mm

[5]


```
{
  "category": "PARAMETER",
  "idShort": "RatedVoltage",
  "description": [
    {
      "language": "en",
      "text": "Rated voltage refers to the standard or nominal voltage at which
    }
  ],
  "semanticId": {
    "type": "ExternalReference",
    "keys": [
      {
        "type": "GlobalReference",
        "value": "0173-1#02-AAO677#002"
      }
    ]
  },
  "qualifiers": [
    {
      "semanticId": {
        "type": "ExternalReference",
        "keys": [
          {
            "type": "GlobalReference",
            "value": "https://admin-shell.io/SubmodelTemplates/Cardinality/1/0"
          }
        ]
      },
      "type": "Cardinality",
      "valueType": "xs:string",
      "value": "One"
    },
    {
      "semanticId": {
        "type": "ExternalReference",
        "keys": [
          {
            "type": "GlobalReference",
            "value": "https://admin-shell.io/SubmodelTemplates/ExampleValue/1/0"
          }
        ]
      },
```

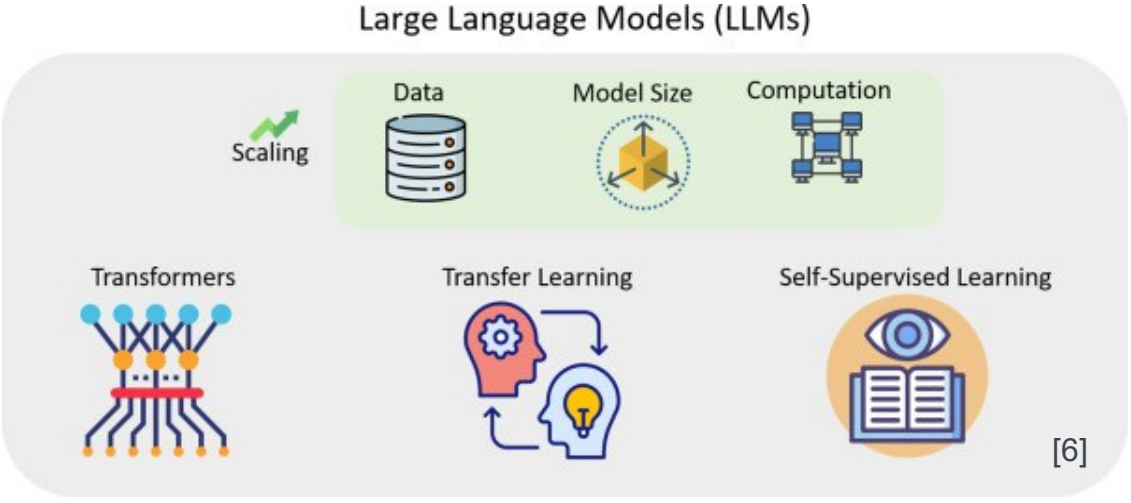


```
{
  "category": "PARAMETER",
  "idShort": "RatedVoltage",
  "description": [
    {
      "language": "en",
      "text": "Rated voltage refers to
    }
  ],
  "embeddedDataSpecifications": [],
  "valueType": "xs:string",
  "value": "",
  "modelType": "Property"
},
```


- Technical Data Sheet
- AAS-Compliant JSON Formats
- Simplified Version

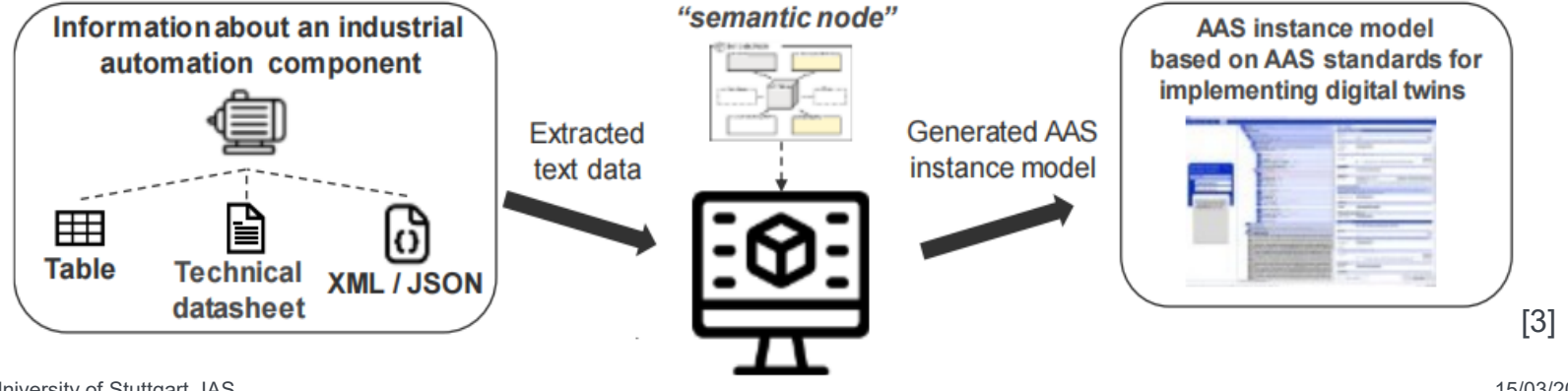
Motivation & Basis

Large Language Model (LLM)



[6]

- Natural Language Processing(NLP)
- Advanced Data Analysis
- Machine Learning Adaptability



[3]

Motivation & Basis

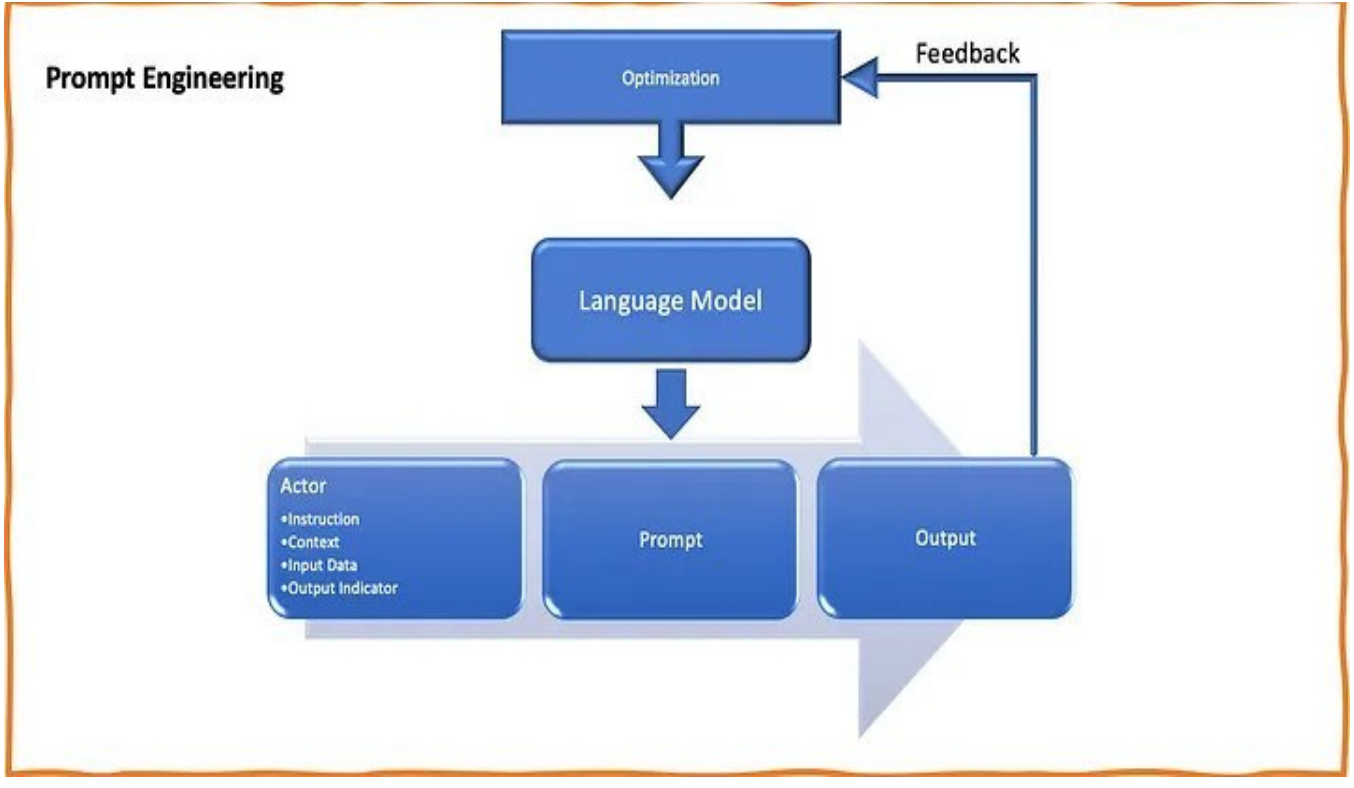
Data-Driven Agent-Based Modeling (DDABM)

[9]

- Decentralization
- Interactivity
- Analytics Integration
- Intelligence
- Adaptation

Motivation & Basis

Prompt Engineering



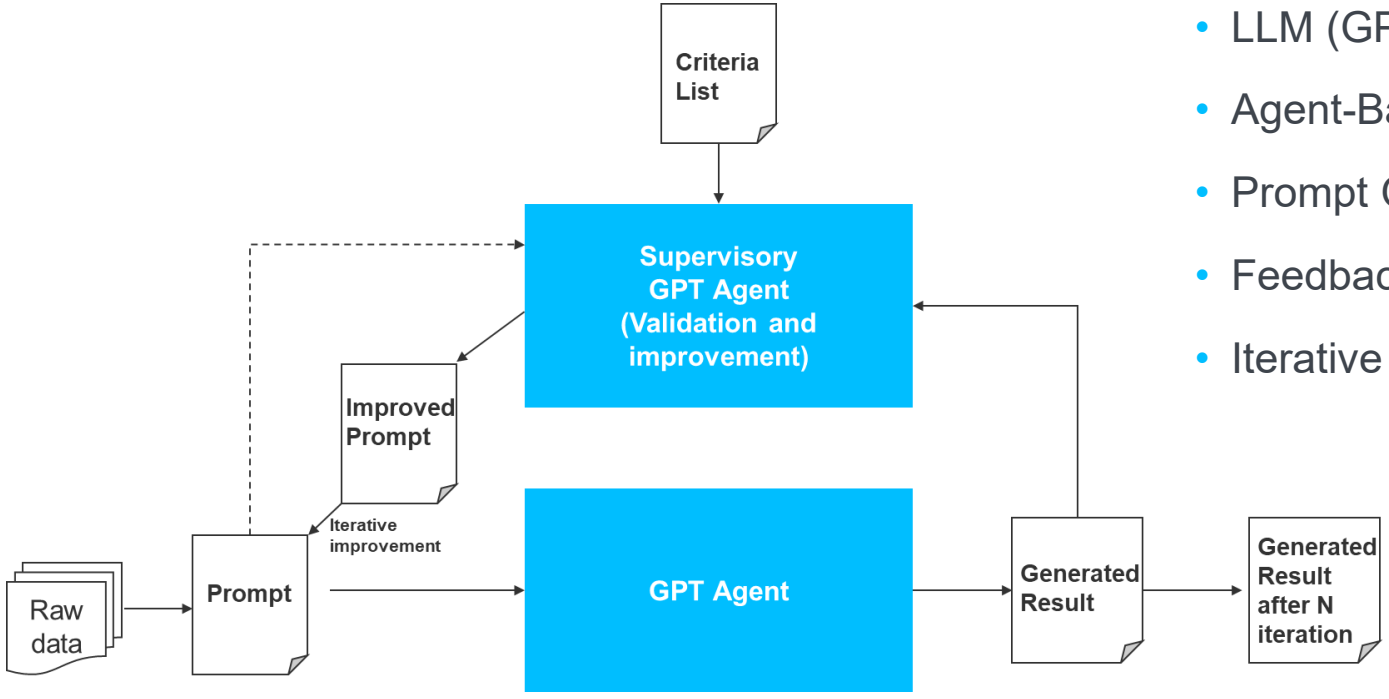
[8]

- Adaption
- Feedback
- Iteration
- Optimization

2. Conceptual Design

Conceptual Design

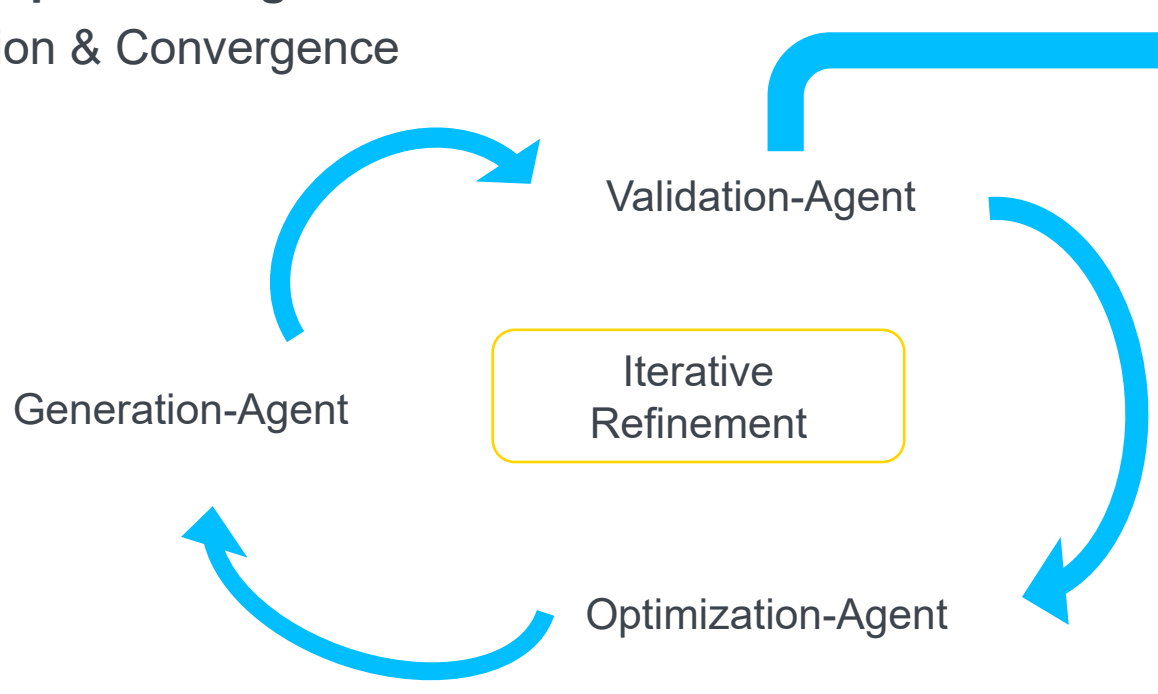
System Overview



- LLM (GPT)
- Agent-Based Modeling
- Prompt Optimization
- Feedback Loop
- Iterative Refinement

Conceptual Design

Iteration & Convergence



- Converged
- Terminated

Converged: generated real Output = initialized ideal Output

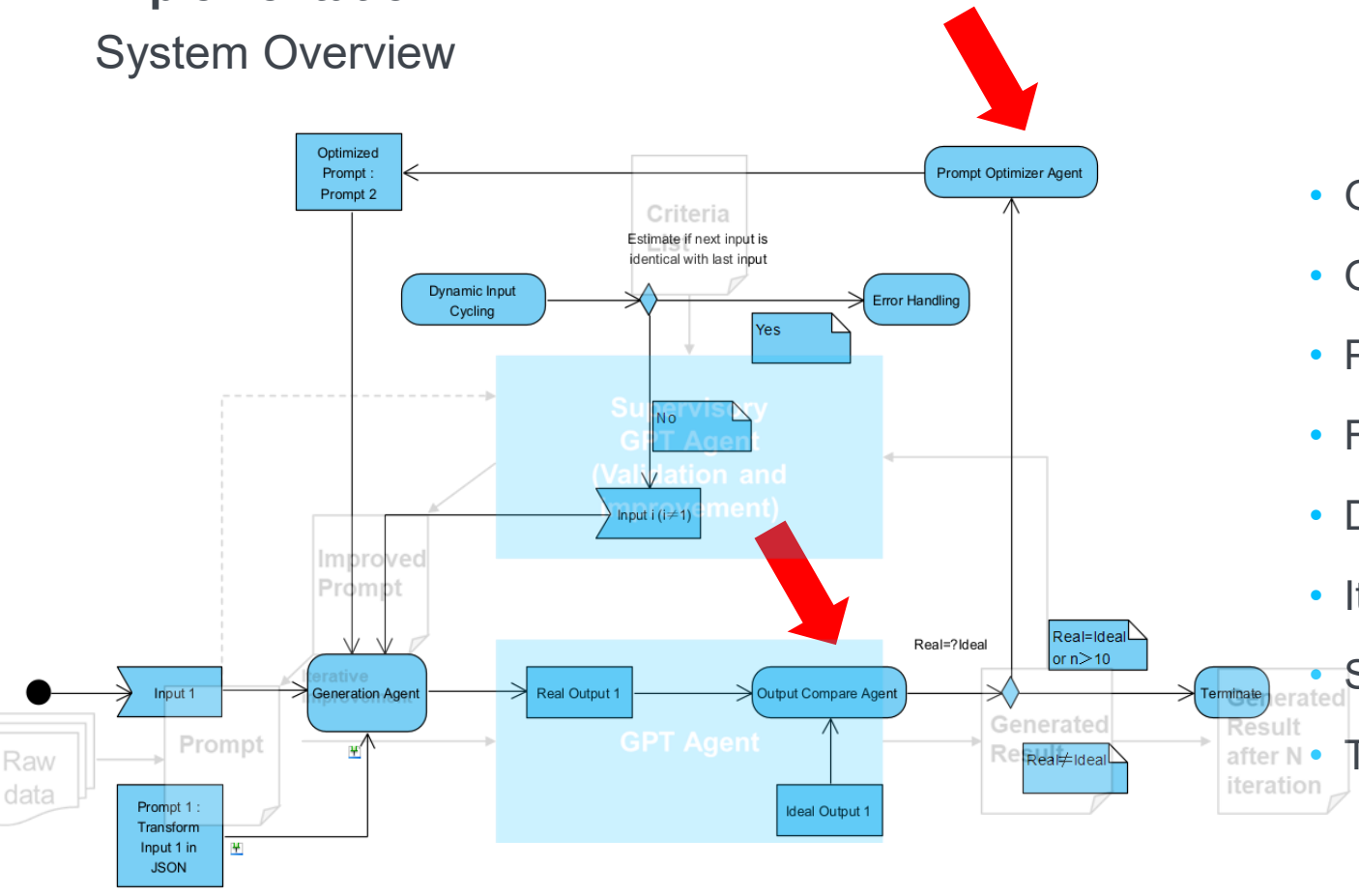
Threshold: 10 Iterations

- Within 10: Efficient!
- After 10: Terminated

3. Implementation

Implementation

System Overview



- Generation Agent
- Output Comparison
- Prompt Optimization
- Feedback Loop
- Dynamic Input
- Iterative Refinement
- Self-Regulation
- Termination Logic

Implementation

Initialization

```
input_data = [{"Power-Supply": "7-12 Vdc"}, {"RMS noise": "0.2 Pa"}, {"Flash Memory": "8 Mbit"}]

ideal_output = [
[
    {
        "category": "CONSTANT",
        "idShort": "powerSupply",
        "description": [
            {
                "language": "en",
                "text": "Power supply in encompasses the voltage range required for operation."
            }
        ],
        "embeddedDataSpecifications": [],
        "valueType": "xs:string",
        "value": "7-12 Vdc",
        "modelType": "Property"
    }
],
```

- Initialization of Input
- Initialization of corresponding ideal Output

```
# Initializing the prompt and generating the initial output
self.current_prompt = (f"Transform the following technical data into {len(input_data)} "
                       f"JSON Object.\nInput Data: {json.dumps(input_data)}\n")
```

- Initialization of initial Prompt

Implementation

Running and Termination

```
++++++++++++
Current POJO:
POJO Object with input {'RMS noise': '0.2 Pa'}
Current iteration 1
Current Prompt: Convert the technical data "RMS noise: 0.2 Pa" into a JSON object with the following structure:
- Include a category field with the value "CONSTANT"
++++++++++++
Current POJO:
POJO Object with input {'RMS noise': '0.2 Pa'}
Current iteration 3
Current Prompt: Generate a JSON object with the following structure based on the technical data "RMS noise: 0.2 Pa":
- Set the category to "CONSTANT".
- Use "rmsNoise" as the idShort.
- Provide a descriptive explanation in English for RMS noise, covering its definition and significance.
- Keep the embeddedDataSpecifications array empty.
- Define the valueType as "xs:string".
- Include the specific RMS noise value as "0.2 Pa".
- Assign the modelType as "Property".
Yes.
Two similar Json objects!
Process completed.

Process finished with exit code 0
```

- System runs with the first Input and initialized Prompt
- System terminates when real Output matches corresponding initialized ideal Output

Implementation

Generation and Comparison


```
"Iteration": 3,
"Prompt": "Absolutely! Here is a refined prompt in natural language to generate the ideal JSON object:\n\n\"Please craft a JSON object that provides technical details about 'Flash Memory'
"Real Output": [
  {
    "technicalData": {
      "category": "CONSTANT",
      "idShort": "flashMemory",
      "description": "Flash memory refers to the storage capacity, typically 8 megabits (Mbit).",
      "embeddedDataSpecifications": [],
      "valueType": "xs:string",
      "modelType": "Property",
      "Flash Memory": "8 Mbit"
    }
  }
],
"Discrepancies": "Here is a list of the differences between the real output provided and the ideal output as you have described:\n\n1. In the real output, the key `'Flash Memory': '8 Mbit'
"Is Terminated": false
```


- Generation of Real Output


```
{
    "category": "CONSTANT",
    "idShort": "flashMemory",
    "description": [
        {
            "language": "en",
            "text": "Flash memory refers to the storage capacity."
        }
    ],
    "embeddedDataSpecifications": [],
    "valueType": "xs:string",
    "value": "8 Mbit",
    "modelType": "Property"
}
```


- Comparison with Ideal Output

- Discrepancy Analysis

- Real Output generated and compared in each iteration

Implementation

Iterative Refinement and Optimization


```
{
    "Iteration": 3,
    "Prompt": "To generate the appropriate JSON object representing Flash Memory with a storage capacity of 8 megabits (Mbit), please provide a detailed and clear description of Flash memory in English,
    "Real Output": [],
    "Discrepancies": "Here is a list of the differences between the real output and the ideal output based on the provided prompt:\n\n1. Real Output: The real output is \"[]\" which indicates an empty JS
    "Is Terminated": false
},
{
    "Iteration": 4,
    "Prompt": "To create a JSON object representing technical data for a power supply with specific details, including:\n- **Category**: Set to CONSTANT\n- **ID Short**: Identified as powerSupply\n- **De
    "Real Output": [
        {
            "category": "CONSTANT",
            "idShort": "powerSupply",
            "description": [
                {
                    "language": "en",
                    "text": "Power supply in encompasses the voltage range required for operation."
                }
            ],
            "embeddedDataSpecifications": [],
            "valueType": "xs:string",
            "value": "7-12 Vdc",
            "modelType": "Property"
        }
    ],
    "Discrepancies": "Here are the differences between the real output and the ideal output:\n\n1. In the real output, the JSON object is wrapped inside a list `[ ]`, while the ideal output does not have
    "Is Terminated": true
}
```


- Discrepancy Detection
- Prompt Adjustment
- Continuous Improvement
- Data-driven Enhancements

Implementation

Dynamic Input Cycling

- Q: How to prevent overfitting?

  (Overfitting in this system design means that LLM simply echo back memorized responses but doesn't truly understands and processes each input.)

- A: Varied Inputs —— Non-repetitive Data

```
pojo = PoJoList[iteration % len(PoJoList)]
```

- Iteration 1: Select pojo1 (1 % 3 = 1)
- Iteration 2: Select pojo2 (2 % 3 = 2)
- Iteration 3: Select pojo3 (3 % 3 = 0)
- Iteration 4: Select pojo1 again (4 % 3 = 1)

- System Robustness
- Adaptive Learning

4. Evaluation

Evaluation

Experiment Overview

Nr.	How many iterations?	Converged?	Within 10 iterations?
1	7	√	√
2	33	√	×
3	4	√	√
4	13	?	?
5	5	?	?
6	2	√	√
7	8	√	√
8	5	?	?
9	6	√	√
10	5	?	?
11	5	√	√
12	6	√	√
13	8	√	√
14	13	?	?
15	15	√	×
16	27	√	×
17	2	√	√
18	3	√	√
19	2	?	?
20	21	?	?

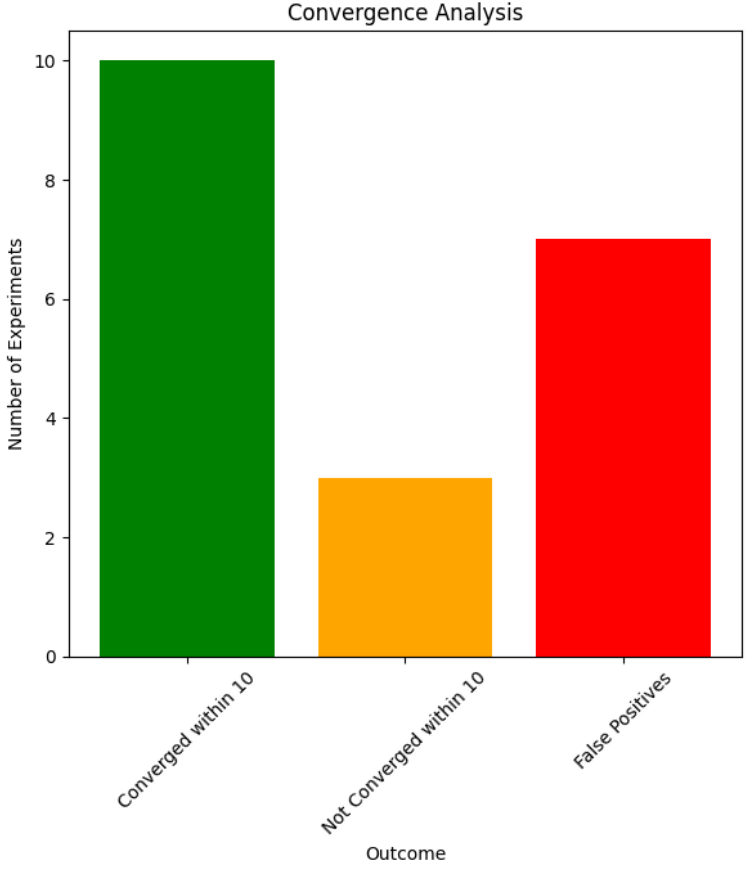
√: Converged

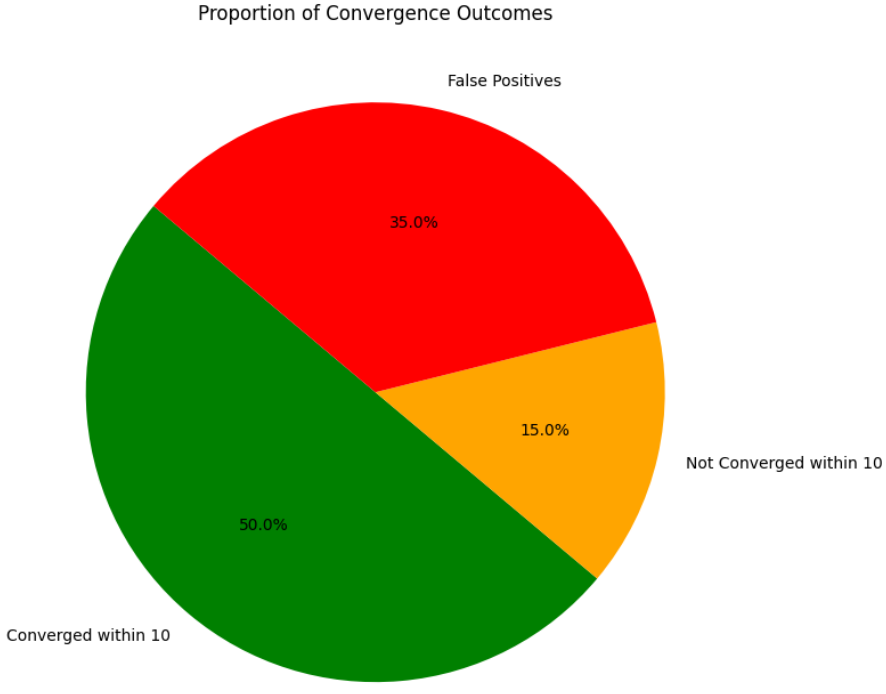
×: Not Con. within 10 iterations

?: False Positive

Evaluation

Results Visualization





Evaluation

Optimization Analysis

IN-PROGRESS PROMPT

```
"**Prompt:** Create a JSON object representing the RMS noise value of 0.2 Pa. Include details such as category, short identifier, description in English, value type as a string, actual value (0.2 Pa), and model type as a property.\n\nThis refined prompt should help in generating a JSON object structure that aligns more closely with the intended format and content."
```

- Detailed Instruction
- Contextual Clarity
- Semantic Precision
- Descriptive Quality

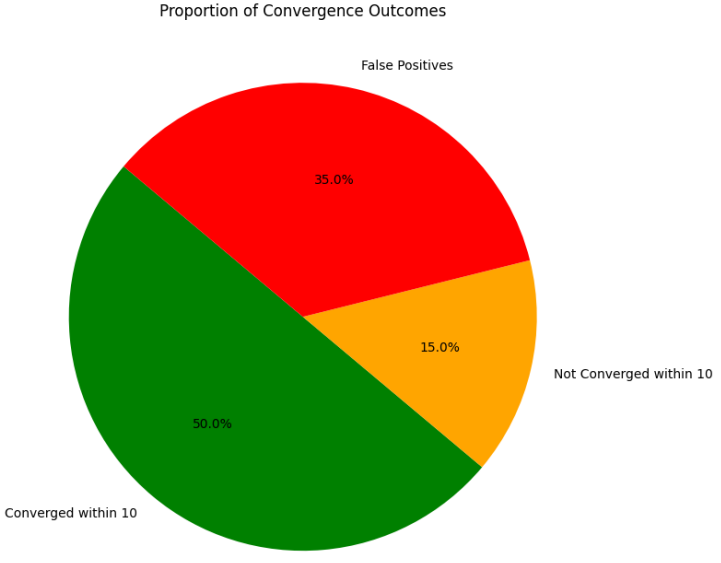
OPTIMIZED PROMPT

```
"Generate a JSON object that describes a power supply. Set the category as CONSTANT and use 'powerSupply' as the ID Short. Provide a description for the power supply. Define the Value Type as xs:string and set the value to '7-12 Vdc'. Lastly, reflect the Model Type as a Property in the JSON object.\" \n\nThe ideal JSON object should have the following structure:\n```json\n{\n  \"category\": \"CONSTANT\",\n  \"idShort\": \"powerSupply\",\n  \"description\": [{\"language\": \"en\", \"text\": \"Explain the power supply as providing the necessary voltage range for operation for operation.\"}],\n  \"embeddedDataSpecifications\": [],\n  \"valueType\": \"xs:string\",\n  \"value\": \"7-12 Vdc\",\n  \"modelType\": \"Property\"\n}\n``` \n\nThis format ensures clarity and completeness in defining the technical data for the power supply."
```

Evaluation

Results Analysis

- Average Iterations to Convergence: The system required an average of **9.7** iterations.
- Range of Iterations: The iterations to convergence varies from a minimum of **2** to a maximum of **33** iterations.
- Converged within 10 Iterations: 50%
- Converged after 10 Iterations: 15%
- False Positives (?): 35%



Strengths:

- Rapid Convergence
- Prompt Efficiency

Improvement:

- Extended Iterations
- Validation Accuracy

5. Summary and Outlook

Summary and Outlook

Summary:

- GPT-Driven Data Transformation: Automated conversion of technical data into AAS-compliant JSON formats, demonstrating LLM's robust ability in NLP.
- Prompt Optimization Methodology: Developed a system for iterative prompt refinement to enhance accuracy and compliance of generated outputs.
- LLM Learning and Generalization: Proved LLM's ability to adapt and generalize across different technical data, achieving convergence within predefined iterations.

Outlook:

- To enhance test scenarios for broader system capabilities.
- Update and utilize newer GPT versions for greater efficiency.
- Broaden the range of test specs to assess system adaptability.

University of Stuttgart
Institut of Industrial Automation
and Software Engineering

Thank you!

Wenjie Xie

e-mail st180778@stud.uni-stuttgart.de
phone +49 (0) 711 685-
fax +49 (0) 711 685-

University of Stuttgart
Institut of Industrial Automation and Software Engineering
Pfaffenwaldring 47, 70550 Stuttgart, Germany

Quelle

[1] https://www.arcweb.com/blog/concepts-applications-i40-asset-administration-shell

[2] https://www.gao.gov/products/gao-23-106453

[3] https://doi.org/10.48550/arXiv.2403.17209

[4] https://dev.to/emanuelferreira/understanding-context-augmentation-behind-llama-index-for-beginners-jdl

[5] https://b2b.harting.com/ebusiness/zh/Cable-Housing-screw-straight-35-65m/21171000100?newPDP=true&eshopNews=true

[6] https://www.sciencedirect.com/science/article/pii/S2949719123000456

[7] https://xebia.com/blog/archetype-llm-batch-use-case/

[8] https://medium.com/@ankur.goel_34001/prompt-engineering-backbone-of-generative-ai-1c164d672dd2

[9] https://developer.nvidia.com/blog/introduction-to-llm-agents/