Prompt Engineering & Leveraging LLM at work !

sanjayae@bu.edu

Prompt Engineering

User Prompt Input

- Chat / API call.
- Tokenization

Token Embedding

High-dimensional embedding vector using a learned embedding matrix for NN to process

Positional Encoding

Transformer

Transformer Processing

- **Multi-head self-attention**: Looks at all tokens to decide which ones are relevant to each other.
- **Feed-forward neural networks**: Processes the outputs of the attention layer.
- **Layer normalization and residual connections**: Helps stabilize training and performance

A probability distribution

Next-Token Prediction

- **Greedy decoding** (pick the highest probability),
- **Sampling** (random draw from the distribution),
- **Top-k or nucleus sampling** (more diverse results), or
- **Beam search** (used more in translation)

Token to Text Conversion

The generated token sequence is decoded back into human-readable text using the inverse of the tokenizer.

Post-processing (Optional)

Check additional rules or safety filters (e.g., RLHF - Reinforcement Learning from Human Feedback) are applied to adjust or sanitize the output.

User Prompt

“ Write 4-line poem with rhyme about a homeless cat “

Control Tokens

- Length - “4-lines”
- Style – “rhyme”
- Temperature – Controls Accuracy Vs Creativity

Context Tokens

- Homeless

Prompt Parameters

- User can manipulate for better outcomes
- Temporary

Model (LLM)

4+4 = 8

Model Parameters

- Fixed (static) to the model.
- Users cannot manipulate
- Expensive and hard to train.

Prompting Paradigms..

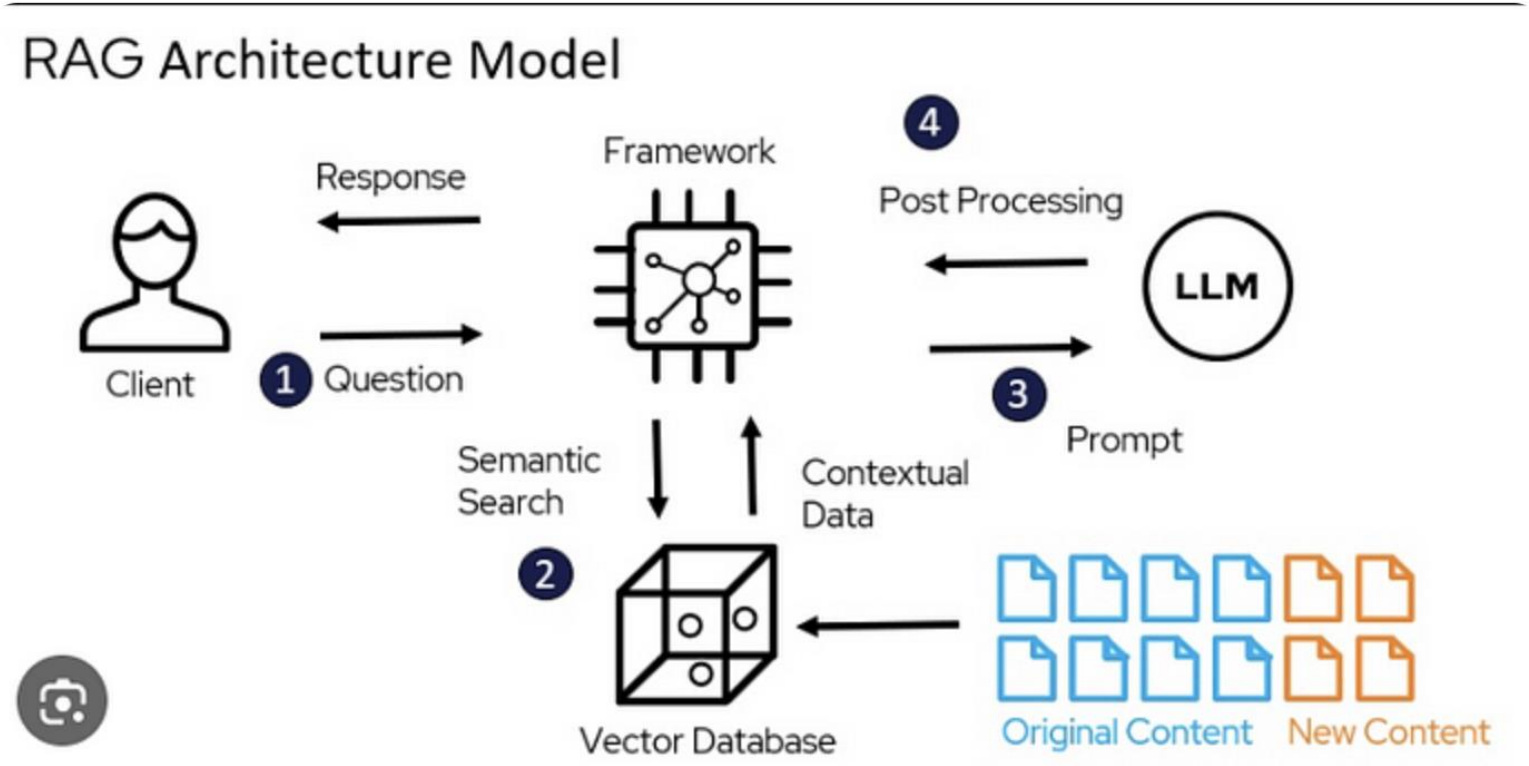
Zero-shot Learning : Ask the model to perform a task without providing any examples.

E.g Prompt: "Translate 'Good morning' to French."**Use Cases**: Language translation, classification, summarization.

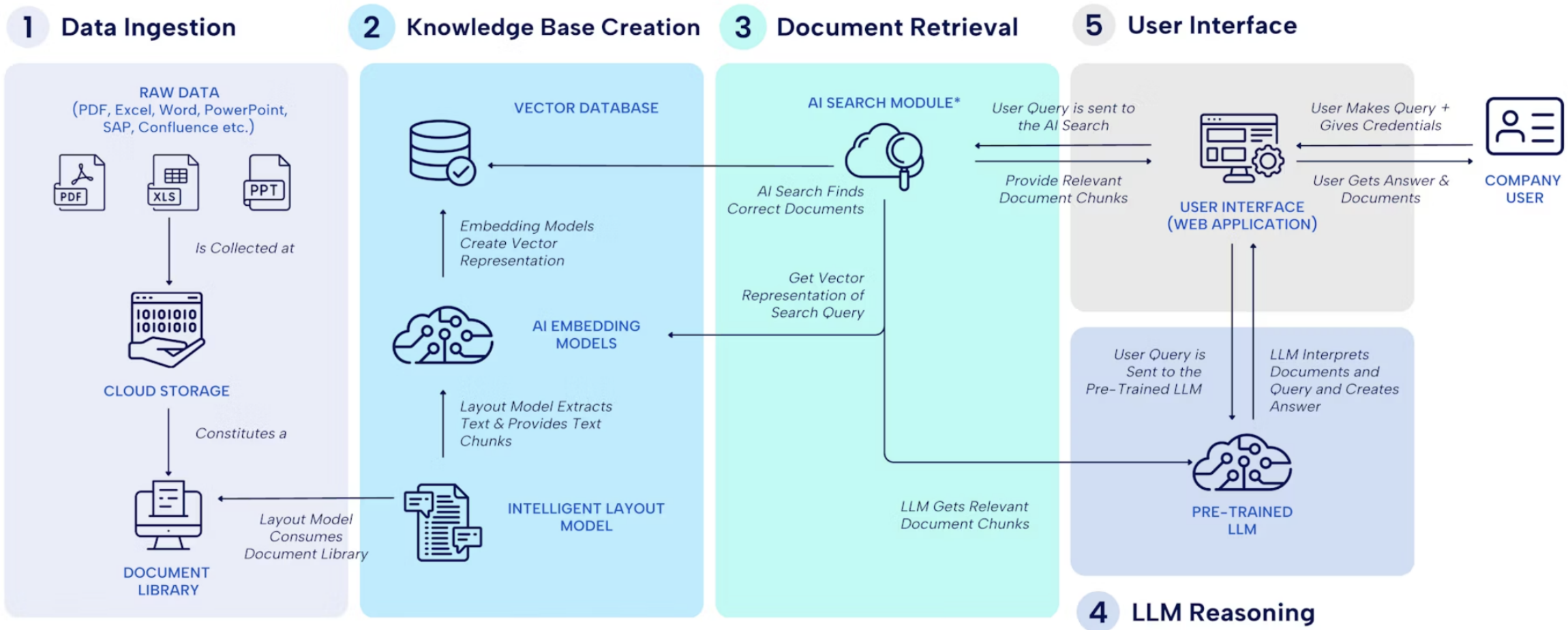
- **One-shot Learning :** Provide one example to guide the model.
- E.g. Prompt: "Translate the following: 'Hello' -> 'Bonjour'. Now translate: 'Good night' ->”

- **Few-shot Learning:** Provide multiple examples in the prompt.
- E.g. Prompt: "Translate the following: 'Hello' -> 'Bonjour' 'Thank you' -> 'Merci' 'Goodbye' ->"

(RAG) Retrieval-Augmented Generation– Architecture



RAG ARCHITECTURE



* Depending on the cloud provider the AI Search Module may match multiple components in this architecture.

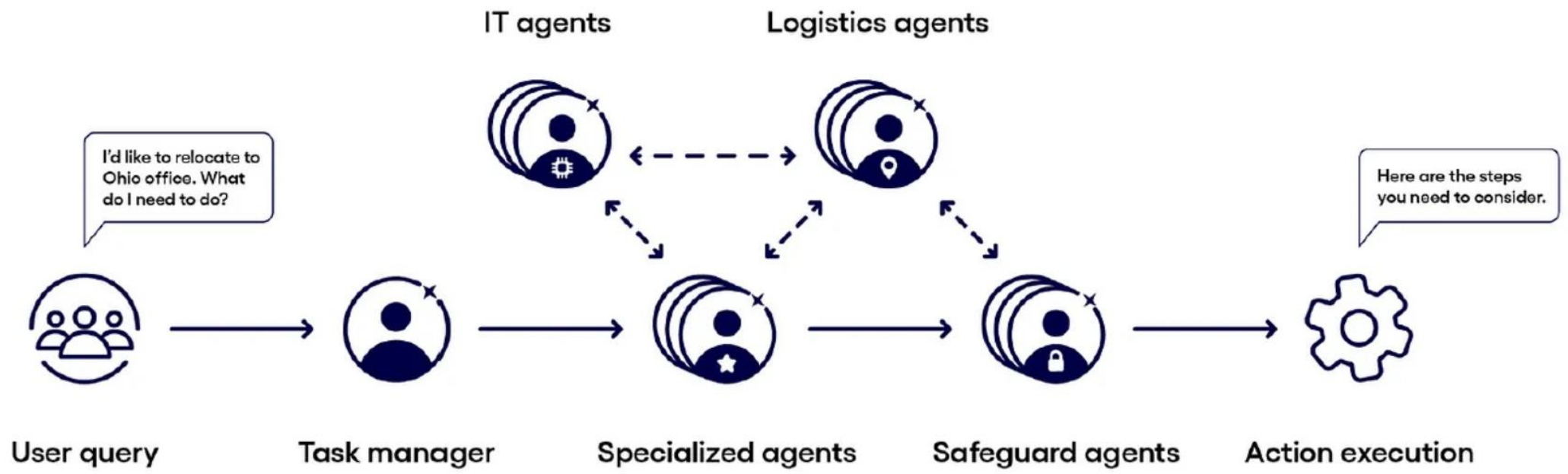
VISIUM

"AI Agents are the future of Apps"

AI Agents

- An artificial intelligence (AI) agent is a **software system** that can perceive its environment, make decisions, and take actions to **achieve specific goals**. These agents **operate autonomously**, meaning they don't require constant human guidance. They can learn, adapt, and even work with other AI agents to accomplish complex tasks

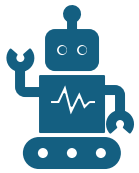
Multi AI Agents systems



https://medium.com/@sanjayae/unlocking-the-power-of-multi-agent-enterprise-ai-systems-ad8afb925aaa

(MCP) Model Context Protocol

MCP is a standard designed to connect AI agents with other agents , external data and tools, enabling them to perform more complex and useful tasks

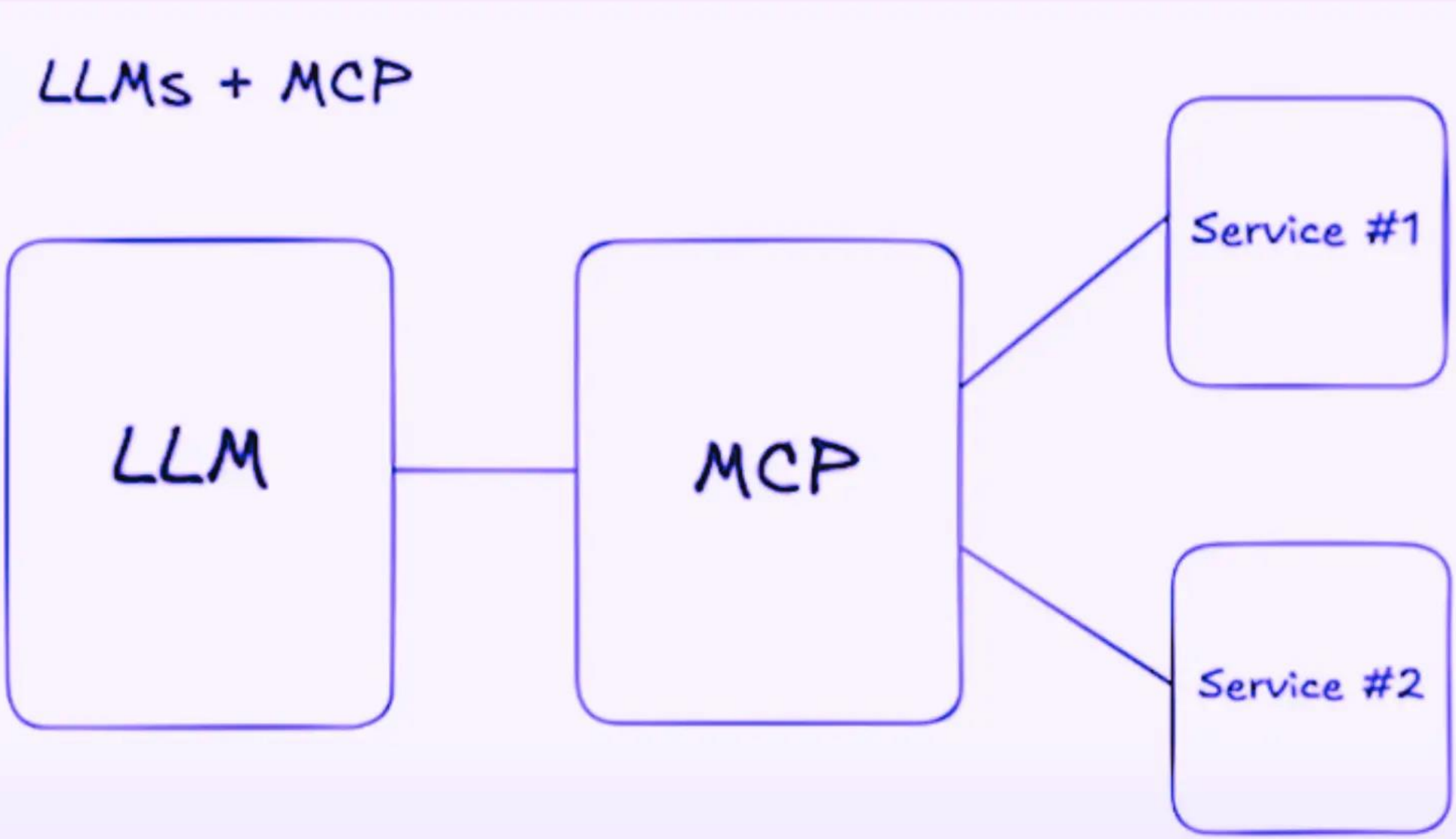


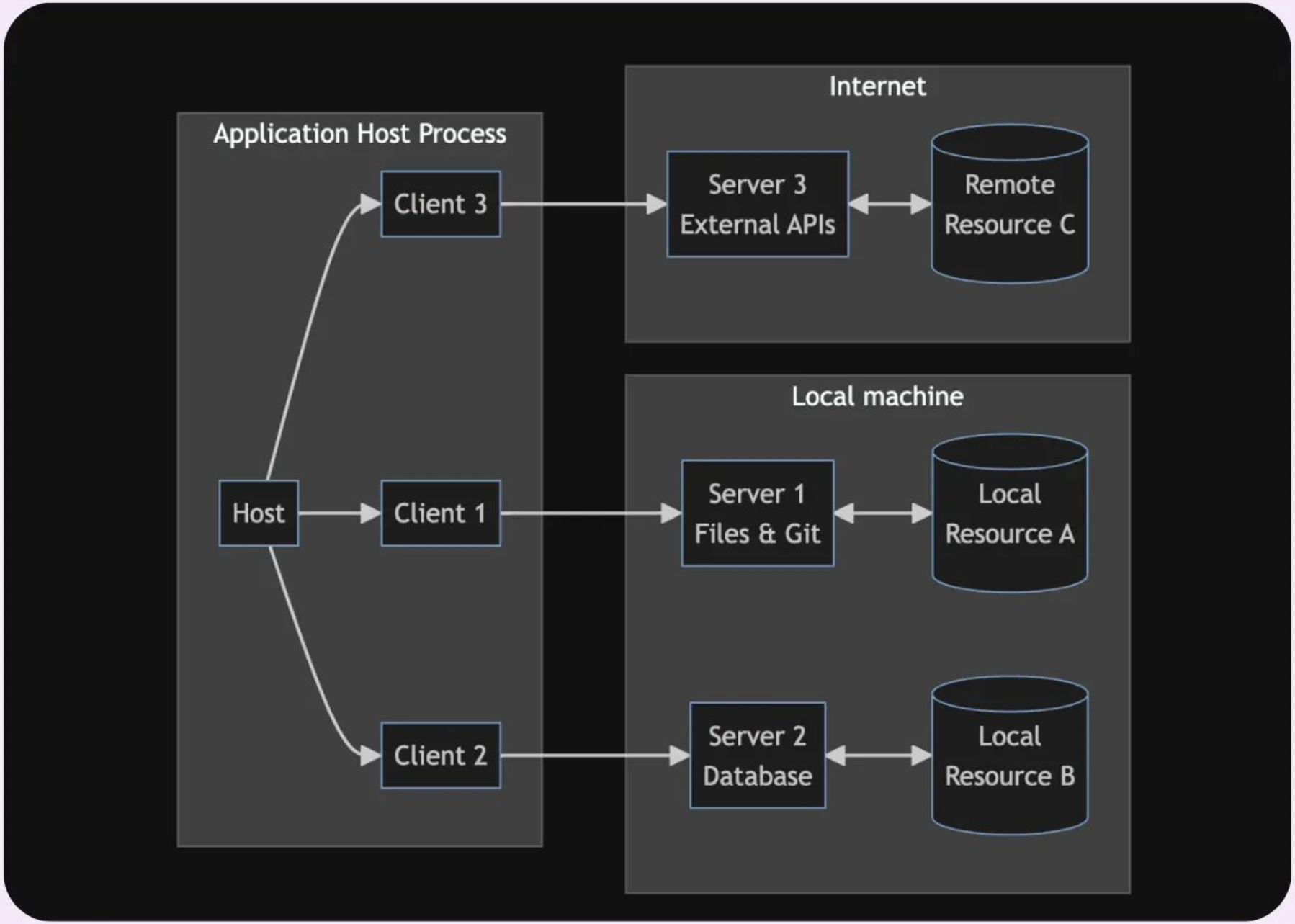
Standardized Interface: consistent way for AI agents to interact with external data sources and tools.

Security and Trust: MCP offers a secure and reliable way to access sensitive data and perform actions, ensuring a trustworthy environment for AI agents.

Flexibility and Scalability: MCP supports a wide range of data sources and tools, allowing AI agents to be built for diverse use cases and scale.

Ease of Integration: MCP simplifies the process of integrating AI agents with external systems, reducing the need for custom coding


LLMs + MCP
LLM
MCP
Service #1
Service #2


Internet
Application Host Process
Client 3
Server 3
External APIs
Remote
Resource C
Local machine
Host
Client 1
Server 1
Files & Git
Local
Resource A
Client 2
Server 2
Database
Local
Resource B

Appendix

Prompt Engineering Tips and Best practices..

Prompt Engineering tips and best practices.. (1)

- 1. No need to be polite with LLM.

2. Integrate the intended audience in the prompt, e.g., the audience is an expert in the field.

3. Break down complex tasks into a sequence of simpler prompts in an interactive conversation.

4. Use positive commands like "do" and avoid negative words like "don't".

5. When you need clarity, utilize the following prompts:

- Explain [insert specific topic] in simple terms.
- Explain to me like I'm 11 years old.
- Explain to me as if I'm a beginner in [field].

6. Add "I'm going to tip $xxx for a better solution!"

- 7. Use one or more line breaks to separate instructions, examples, questions, context, and input data.

Prompt Engineering tips and best practices.. (2)

8. When formatting your prompt, start with '### Instruction ###', followed by either '### Example ###' or '### Question ###' if relevant.

9. Incorporate the following phrases: "Your task is" and "You MUST".

10. Incorporate the following phrases: "You will be penalized if".

11. Use the phrase "Answer a question given in a natural, human-like manner" in your prompts.

12. Use leading words like writing "think step by step" (chain-of-thought prompting).

13. Add to your prompt the following phrase "Ensure that your answer is unbiased and does not rely on stereotypes".

14. Let the model ask you questions to get enough details to give the right answer (e.g., "Please ask me questions to get the info you need...").

15. To ask about a specific topic:

"Teach me [any theorem/topic/rule] and include a test at the end. Don't give me the answers; just tell me if I got them right."

Prompt Engineering tips and best practices.. (3)

16. Assign a role to the large language models.

17. Use Delimiters.

18. Repeat a specific word or phrase multiple times within a prompt.

19. Combine Chain-of-thought (providing step-by-step) with few-shot prompts (giving examples).

20. End your prompt with the start of the desired response to guide the model to the right answer.

21. "Write a detailed [essay/text/paragraph] on [topic], including all necessary information."

22. To correct specific text without changing its style:

"Revise each paragraph to improve grammar and vocabulary while keeping the original writing style."

Prompt Engineering tips and best practices.. (4)

23. For complex coding prompts across files:

"Generate a [programming language] script that can create or modify files as needed to include the generated code. [your question]"

24. For starting or continuing a text with specific words:

"Here is the beginning [song lyrics/story/paragraph/essay...]: [Insert lyrics/words/sentence]. Finish it and keep the flow consistent."

25. Clearly state the requirements for the model, including keywords, rules, hints, or instructions.

26. To write text similar to a sample, include this instruction: "Use the same language style as the provide