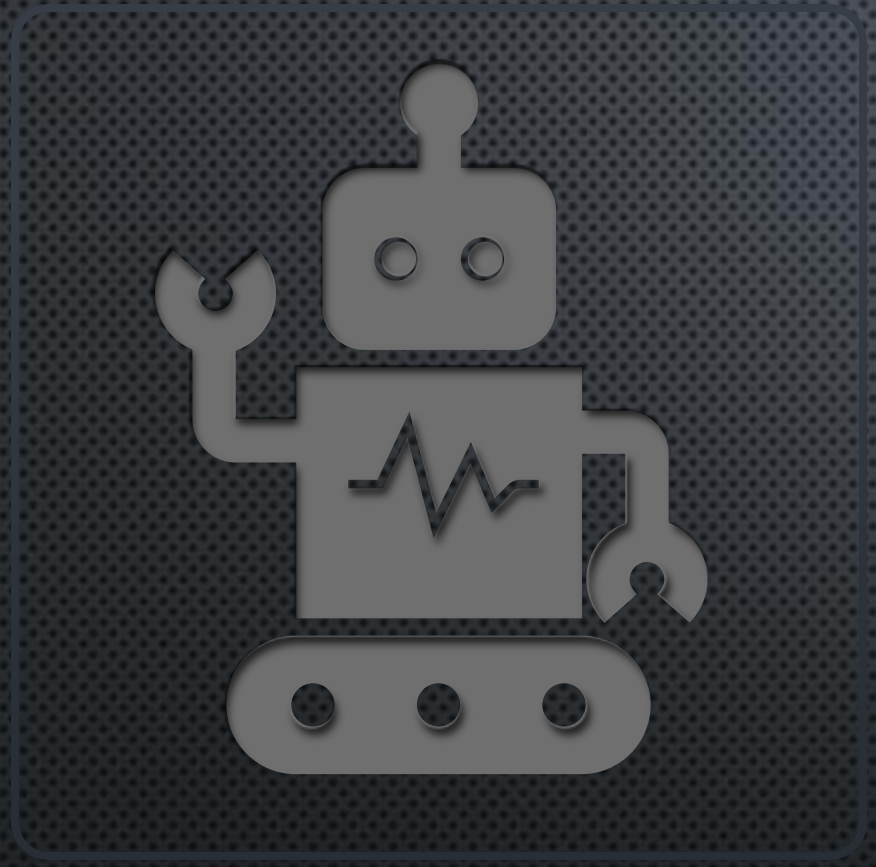


INTRODUCTION TO ADVANCED LARGE LANGUAGE MODELS

A COMPREHENSIVE TUTORIAL ON TECHNIQUES, ARCHITECTURES, AND PRACTICAL APPLICATIONS

PART 1

Giorgio Roffo, PhD

Explore and Connect

LinkedIn: Giorgio Roffo - Linkedin
ResearchGate: Work Done
Google Scholar: My Publications
GitHub: giorgioroffo

Special thanks to Shelbee Eigenbrode, Antje Barth, and Mike Chambers for their work on "Generative AI with Large Language Models" (Coursera, Amazon AWS, 2023), which significantly informed and inspired the material used in these slides.

WE ARE GOING TO TALK ABOUT…

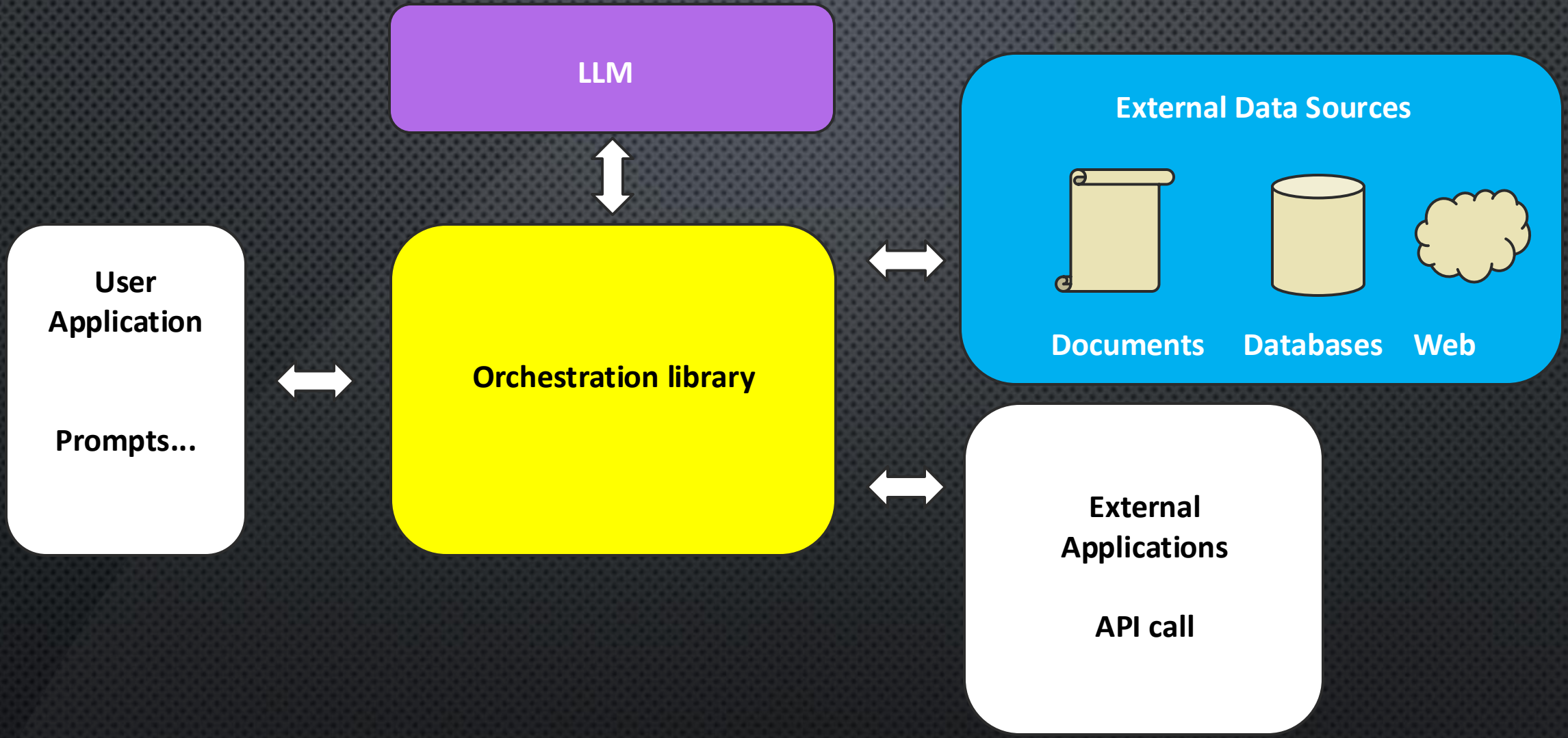
1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
    - Retrieval-Augmented Generation (RAG)
    - Program-Aided Language Models (PAL)
    - Integration of Reasoning and Action (ReAct)
    - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)
2. **Introduction: Transformers**
3. **Selecting LLM Architectures**
4. **LLM Training Resources**: GPU Memory Requirements
    - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
    - GPU RAM for Models:
        - 1B parameters: 24GB (32-bit precision)
        - 175B parameters: 4200GB (32-bit precision)
        - 500B parameters: 12000GB (32-bit precision)
5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-T**uning Strategies: Addressing Specific Tasks and Preventing Overfitting
7. ReST (Google): Reinforced Self-Training for Language Modeling
8. MED-Gemini

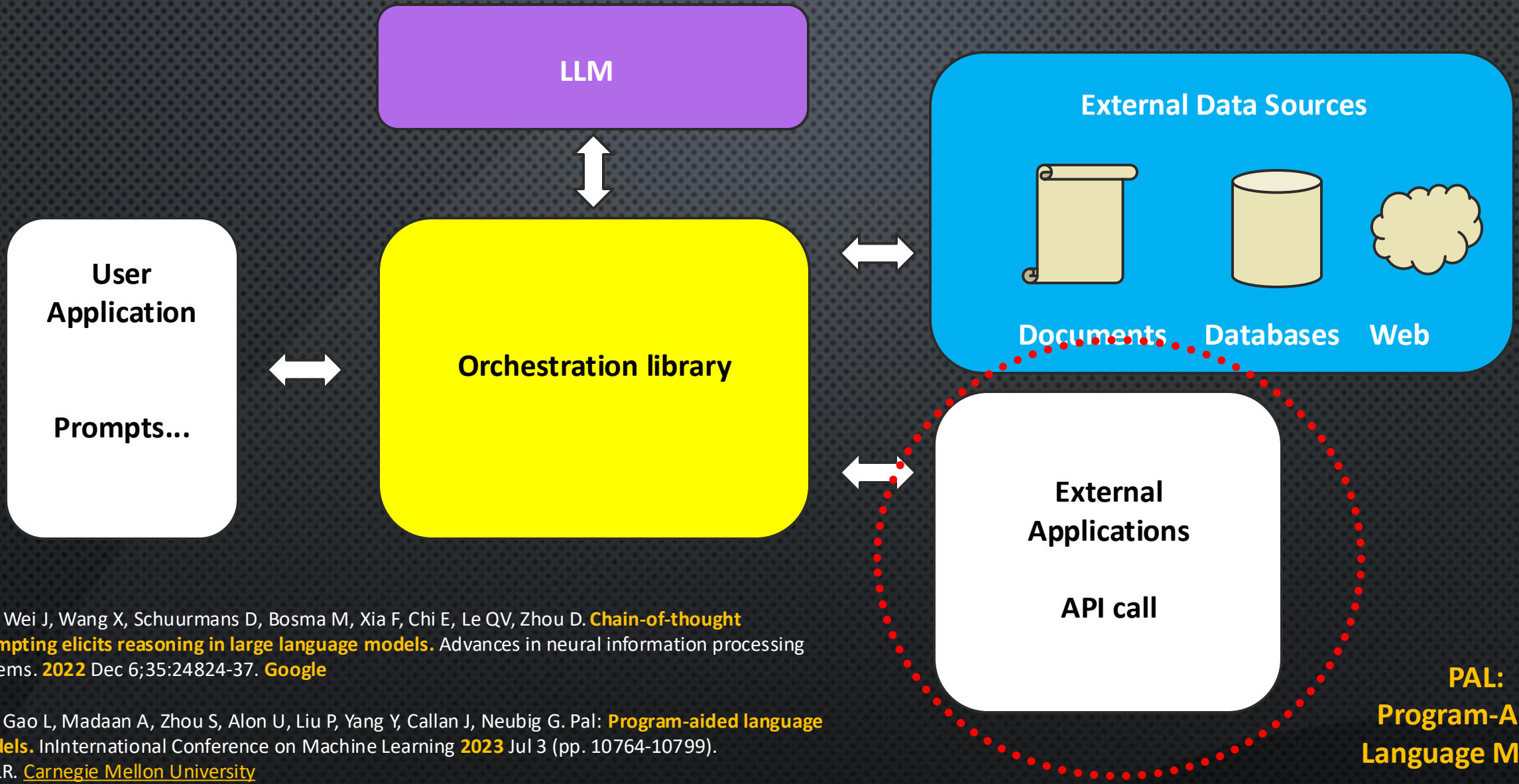
Using the LLM in applications:

- **Outdated Information**: A common issue with LLMs arises when asked about recent events, as they may provide incorrect responses due to their training data being outdated.
- **Mathematical Challenges**: LLMs often struggle with mathematical problems because their primary function is to predict the next most likely token from the text, not to perform calculations.
- **Hallucination**: One of the best-known problems of LLMs is their tendency to generate text even when they don't know the answer to a problem.
- **Creativity**: LLMs often demonstrate remarkable creativity, yet their solutions can sometimes be entirely incorrect.

ChatGPT?

- ChatGPT, Gemini, and others employ techniques that address these issues by connecting to external data sources and applications (API, Documents, Web, …).
- **Retrieval Augmented Generation (RAG)**
- **Program-aided Language Models (PAL)**


LLM
User Application
Prompts...
Orchestration library
External Data Sources
Documents
Databases
Web
External Applications
API call

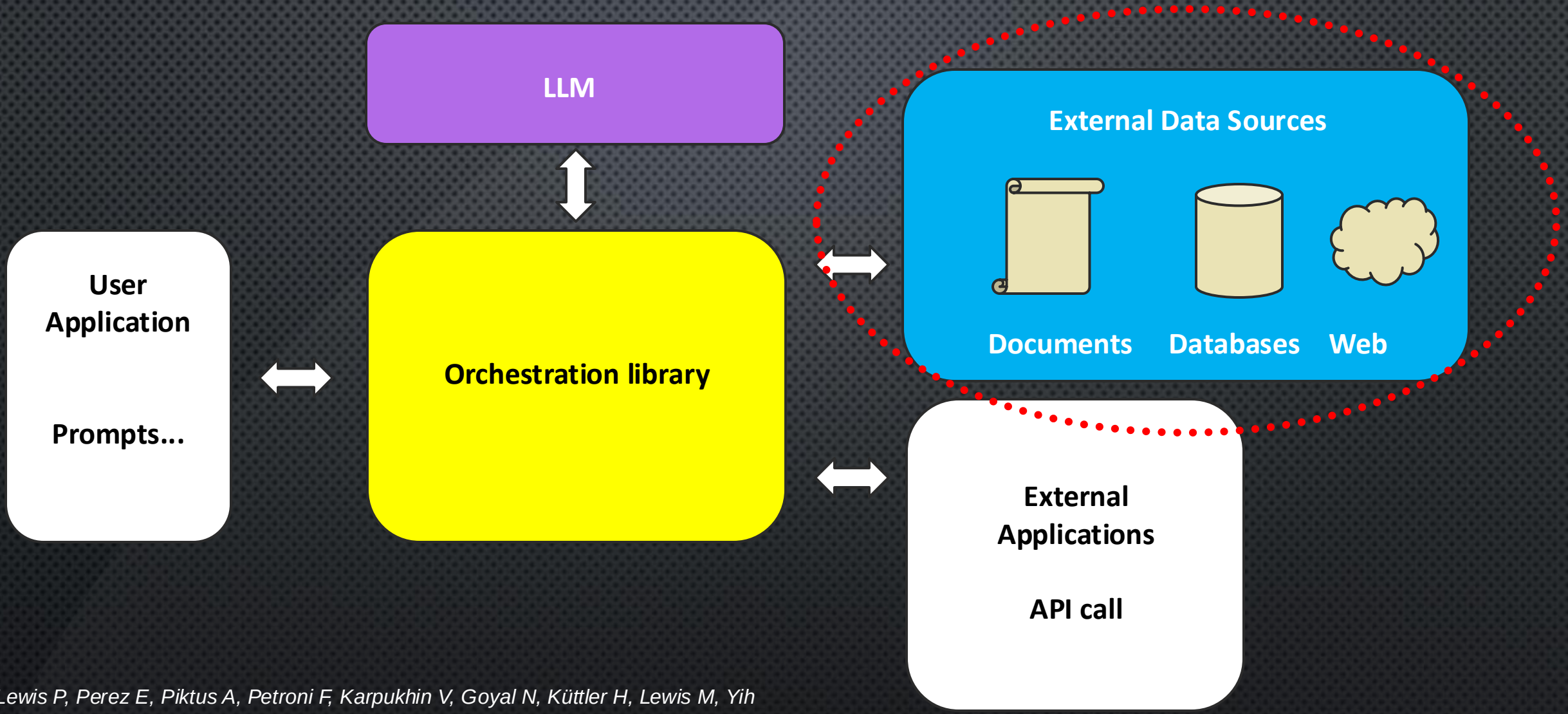


[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. **2022** Dec 6;35:24824-37. **Google**

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. Pal: **Program-aided language models.** InInternational Conference on Machine Learning **2023** Jul 3 (pp. 10764-10799). PMLR. Carnegie Mellon University

PAL: Program-Aided Language Models

RAG: Retrieval-Augmented Generation



[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***

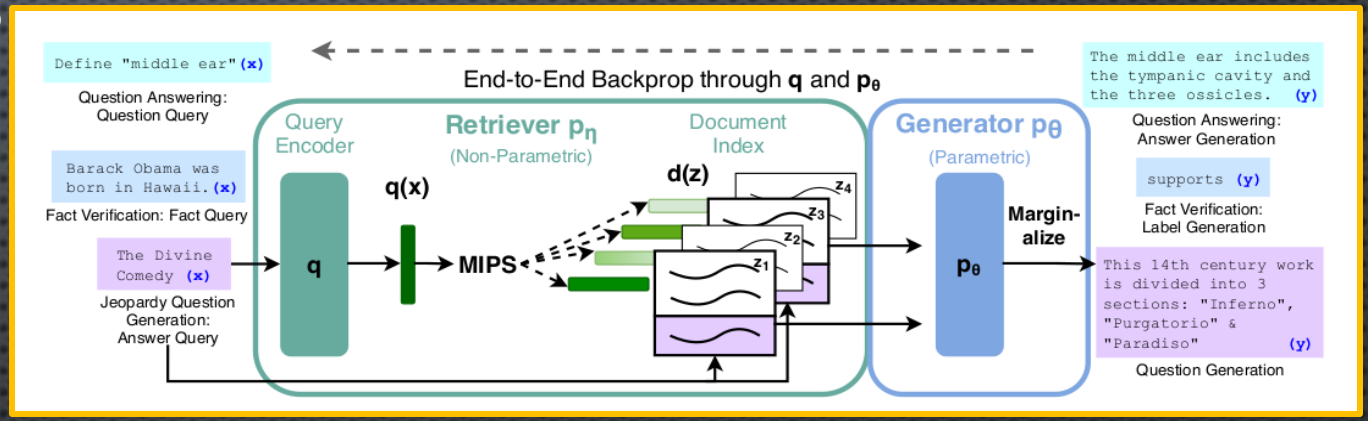
Retrieval Augmented Generation (RAG)

1. **Document Retrieval**:
   - RAG starts by retrieving relevant documents from a dataset using a **dense vector search** to match **the query's context** (documents, wikis, web pages, databases, etc..).

2. **Integration with Language Model**:
   - These **documents are then integrated into a language model**, providing it with external context to inform its responses.

3. **Contextual Generation**:
   - The model uses the **combined** knowledge of the **query and retrieved documents** to generate informed and detailed responses.

4. **End-to-End Training**:
   - **RAG is trained end-to-end**, simultaneously refining both the retrieval mechanism and the response generation to improve relevance and accuracy.



RAG is useful in any case where we want the language model to have access to data that it may not have seen.

This could be new information documents not included in the original training data, or proprietary knowledge stored in the organization's private databases.

[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***

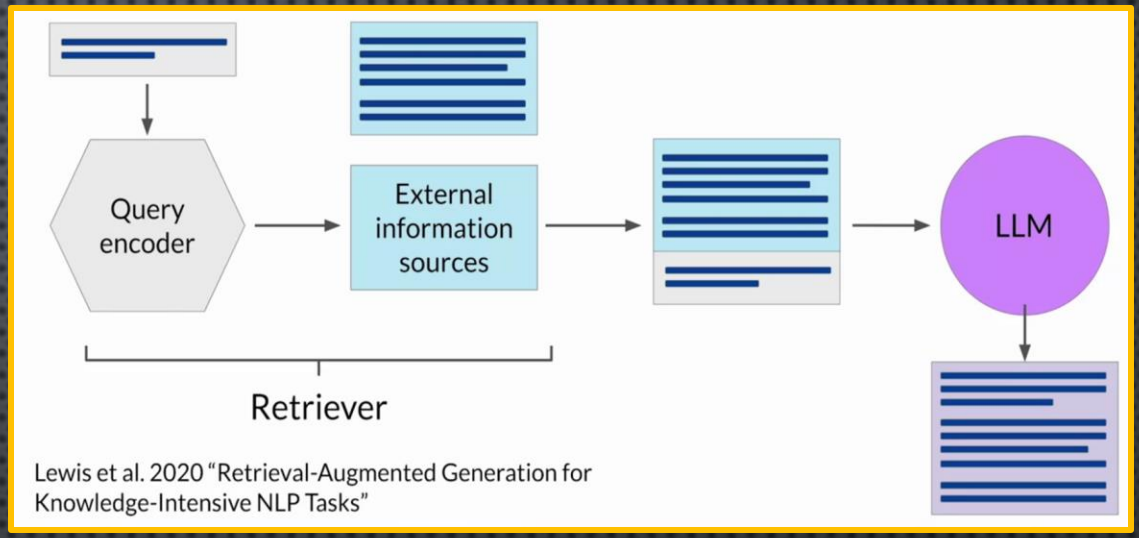
Retrieval Augmented Generation (RAG)

1. **Document Retrieval**:
   - RAG starts by retrieving relevant documents from a dataset using a **dense vector search** to match **the query's context** (documents, wikis, web pages, databases, etc..).

2. **Integration with Language Model**:
   - These **documents are then integrated into a language model**, providing it with external context to inform its responses.

3. **Contextual Generation**:
   - The model uses the **combined** knowledge of the **query and retrieved documents** to generate informed and detailed responses.

4. **End-to-End Training**:
   - **RAG is trained end-to-end**, simultaneously refining both the retrieval mechanism and the response generation to improve relevance and accuracy.



Lewis et al. 2020 "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"

RAG is useful in any case where we want the language model to have access to data that it may not have seen.

This could be new information documents not included in the original training data, or proprietary knowledge stored in the organization's private databases.

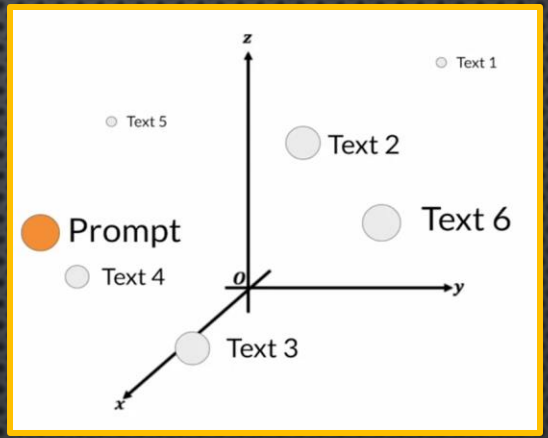
[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***

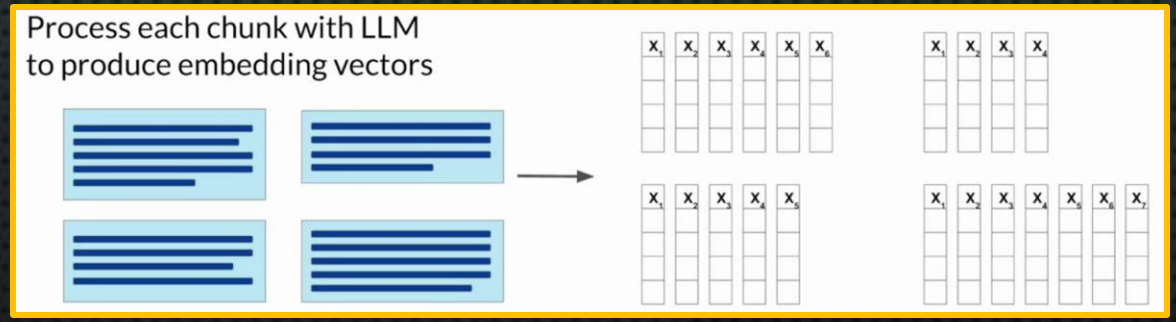
Retrieval Augmented Generation (RAG) - in short

Retrieval-augmented generation (RAG) is a technique for enhancing the accuracy and reliability of generative AI models with facts fetched from external sources.

- **Retrieve Content**: RAG fetches relevant documents, wikis, etc.
- **Chunk Large Documents**: Breaks down extensive documents into manageable chunks to fit the context window.
- **Embedding Generation**: Uses LLMs to create embedding vectors for each resource.
- **Vector Stores**: Saves these embeddings in "Vector stores"
- **Vector Database**: Implements a "Vector store" where each vector is keyed for easy retrieval, often with associated static text like citations.

Vector Database





[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***

Retrieval Augmented Generation (RAG) - Summary

Retrieval-augmented generation (RAG) is a technique for enhancing the accuracy and reliability of generative AI models with facts fetched from external sources.

Components of RAG

1. Parametric Component (Generator):
   - A pre-trained seq2seq model (e.g., BART) that generates responses based on the context provided by the retrieved documents and the query.
   - Responsible for the final text generation.
2. Non-Parametric Component (Retriever):
   - A dense vector index of documents (e.g., Wikipedia articles) that acts as a retrievable external memory.
   - Uses a neural retriever (e.g., DPR—Dense Passage Retriever) to fetch relevant documents based on the input query.

Operation Workflow

1. Query Input:
   - The input query is processed by the retriever to identify relevant context from the document index.
2. Document Retrieval:
   - The retriever computes vector representations of the query and documents, retrieving the most relevant documents using techniques like Maximum Inner Product Search (MIPS).
3. Sequence Generation:
   - The retrieved documents are fed, along with the original query, into the seq2seq generator.
   - The generator then produces the output text by integrating the information from the query and the retrieved documents.

[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***

Retrieval Augmented Generation (RAG) - Summary

Retrieval-augmented generation (RAG) is a technique for enhancing the accuracy and reliability of generative AI models with facts fetched from external sources.

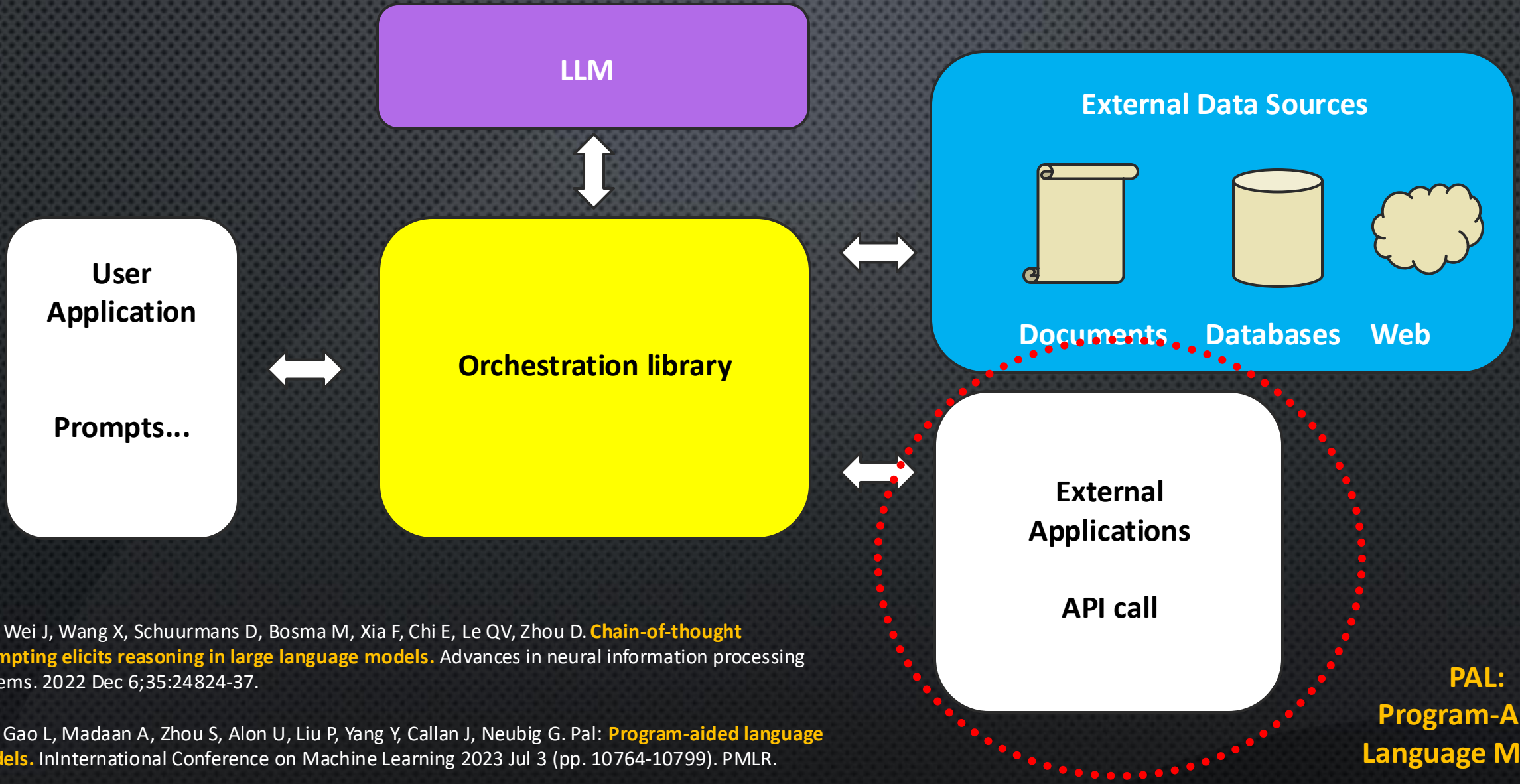
Training and Weights

1. *End-to-End Training:*
   - *Both the retriever and generator are trained simultaneously to optimize the generation of responses.*
   - *The system treats the retrieved document as a latent variable and backpropagates the loss from the output through both the retriever and generator, allowing the model to learn which documents are useful for generating correct responses.*

2. *Weights:*
   - *The retriever (neural network for document querying) and the generator (seq2seq model) both have trainable weights.*
   - *The weights are updated during training based on the model's performance in generating accurate and relevant responses.*

3. *Training Objective:*
   - *The main training objective is to minimize the negative log-likelihood of the target sequence given the input and the retrieved documents.*
   - *This approach allows RAG to refine both its retrieval mechanism (to fetch more pertinent documents) and its generation strategy (to produce more accurate outputs).*

[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. ***Retrieval-augmented generation for knowledge-intensive nlp tasks****. Advances in Neural Information Processing Systems. 2020;33:9459-74.* ***Facebook***



[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37.

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. Pal: **Program-aided language models.** InInternational Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR.

PAL: Program-Aided Language Models

Helping LLMs reason and plan with chain-of-thought (CoT)

Prompt

- ✓ **Q:** Roger has 5 apples. He buys 2 more bags of apples. Each bag contains 3 apples. How many apples does he have now?
- ✓ A: The answer is 11

- ✓ Q: The restaurant had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Completion

- ✓ **Q:** Roger has 5 apples. He buys 2 more bags of apples. Each bag contains 3 apples. How many apples does he have now?
- ✓ A: The answer is 11

- ✓ Q: The restaurant had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

- ✓ A: The answer is 25 X

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

Helping LLMs reason and plan with chain-of-thought (CoT)

Prompt

- ✓ **Q:** Roger has 5 apples. He buys 2 more bags of apples. Each bag contains 3 apples. How many apples does he have now?
- ✓ A: The answer is 11

- ✓ Q: The restaurant had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Chain of though Prompting

Start: Roger started with 5 apples
Step 1: add 2 bags of 3 apples each, 3x2=6
Step 2: 5+6 = 11
End: The answer is 11

Reasoning Steps

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

Helping LLMs reason and plan with chain-of-thought

Prompt

- ✓ **Q:** Roger has 5 apples. He buys 2 more bags of apples. Each bag contains 3 apples. How many apples does he have now?
- ✓ A: Roger started with 5 apples. add 2 bags of 3 apples each, 3x2=6. 5+6 = 11. The answer is 11

- ✓ Q: The restaurant had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Completion

- ✓ Q: Roger has 5 apples. He buys 2 more bags of apples. Each bag contains 3 apples. How many apples does he have now?
- ✓ A: Roger started with 5 apples. add 2 bags of 3 apples each, 3x2=6. 5+6 = 11. The answer is 11

…….

……

how many apples do they have?

- ✓ **A: The restaurant had 23 apples. they used 20 to make lunch, 23-20=3. They added 6 more, 3 + 6 = 9. The answer is 9.**

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

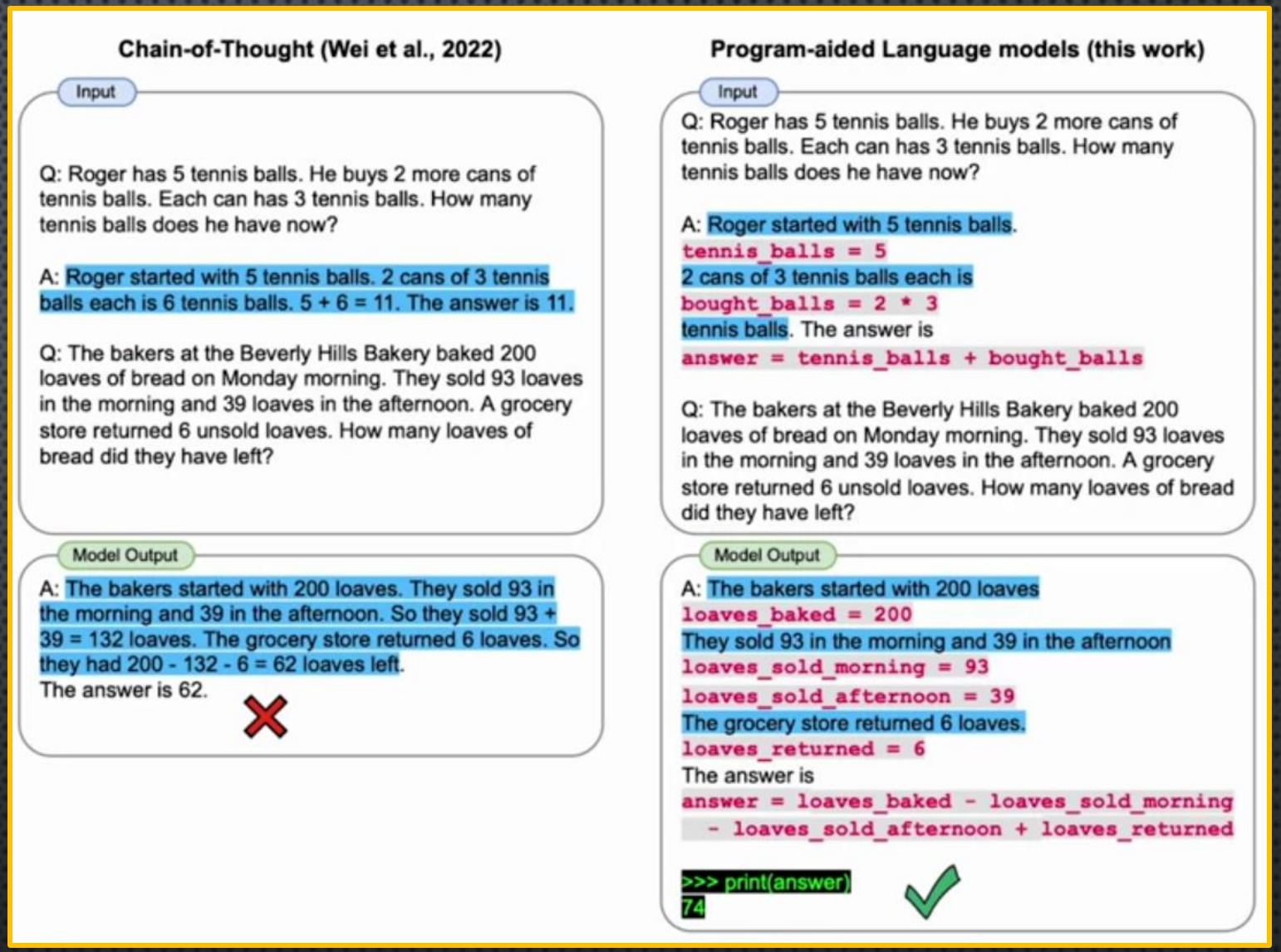
Chain-of-Thought Prompting Process:

1. **Model Selection**:
   - Use pre-trained large language models (e.g., PaLM 540B) without additional fine-tuning.
2. **Prompt Engineering**:
   - Design prompts that include a few input-output pairs demonstrating the task.
   - Each prompt pair consists of an input question, an intermediate reasoning chain (the chain of thought), and the final output.
   - This approach mimics human problem-solving by breaking down the task into intermediate steps leading to the answer.
3. **Model Input**:
   - The model receives the engineered prompts containing the intermediate reasoning steps.
4. **Model Processing**:
   - The language model processes the input while internally mimicking the reasoning steps demonstrated in the prompts.
5. **Output Generation**:
   - The model generates a response that ideally follows the demonstrated reasoning pattern, leading to a final answer.
6. **Evaluation**:
   - Performance of the model is evaluated on various reasoning tasks to assess the effectiveness of the chain-of-thought prompting.

Program-aided language models (PAL)

LLM + Code Interpreter

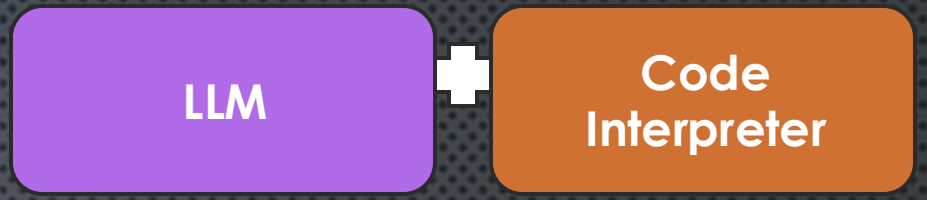
PAL uses Chain of thought prompting to make the LLM to generate completions that are executable python code.



[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

Program-aided language models (CoT reasoning + PAL)



Prompt with one-shot example

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Answer:

```
# Roger started with 5 tennis balls
tennis_balls = 5
# 2 cans of tennis balls each is
bought_balls = 2 * 3
# tennis balls. The answer is
answer = tennis_balls + bought_balls
```

Q: The bakers at the Beverly Hills Bakery baked 200 loaves of bread on Monday morning. They sold 93 loaves in the morning and 39 loaves in the afternoon. A grocery store returned 6 unsold loaves. How many loaves did they have left?

Completion, CoT reasoning (blue) , and PAL execution (pink)

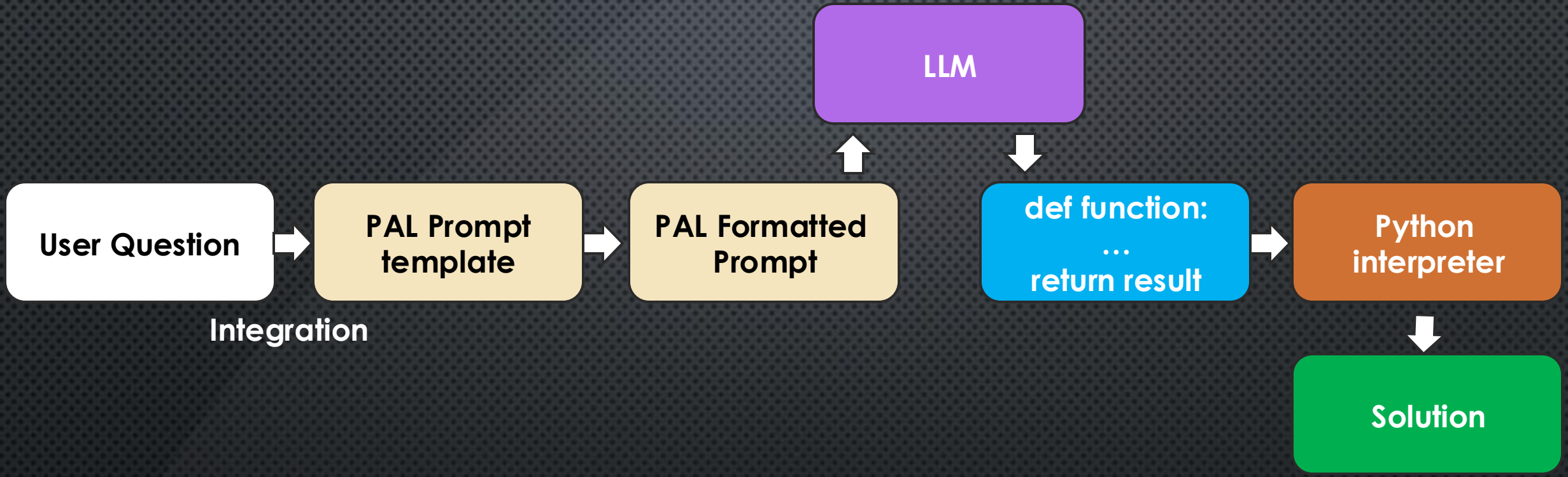
Answer:

```
# The bakers started with 200 loaves
loaves_baked = 200
# They sold 93 in the morning and 39 in the
afternoon
loaves_sold_morning = 93
loaves_sold_afternoon = 39
# The grocery store returned 6 loaves.
loaves_returned = 6
# The answer is
answer = loaves_baked
  - loaves_sold_morning
  - loaves_sold_afternoon
  + loaves_returned
```

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

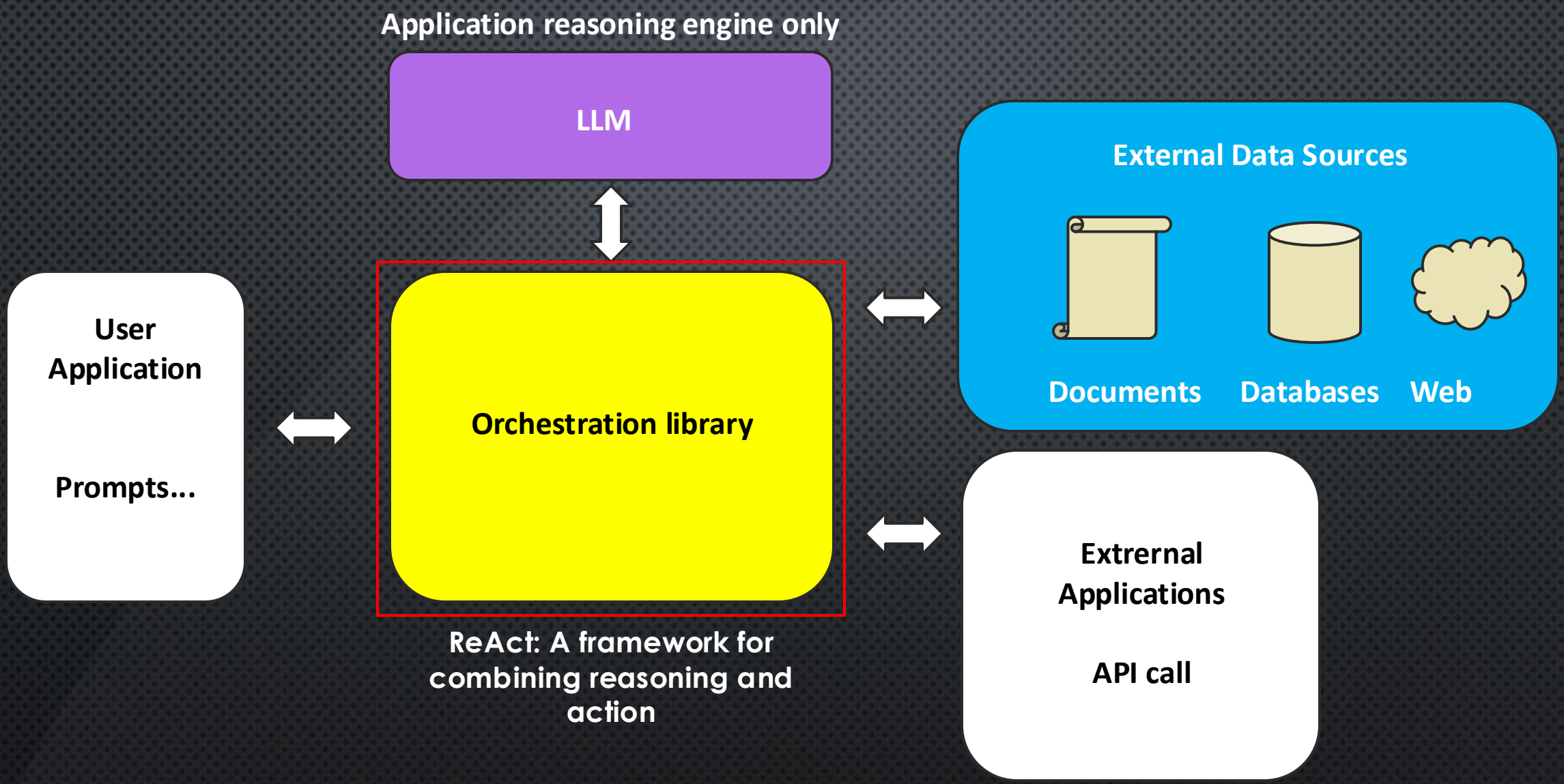
Program-aided language models (CoT reasoning + PAL)



[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37. Google

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. **Pal**: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR. CMU

Application reasoning engine only



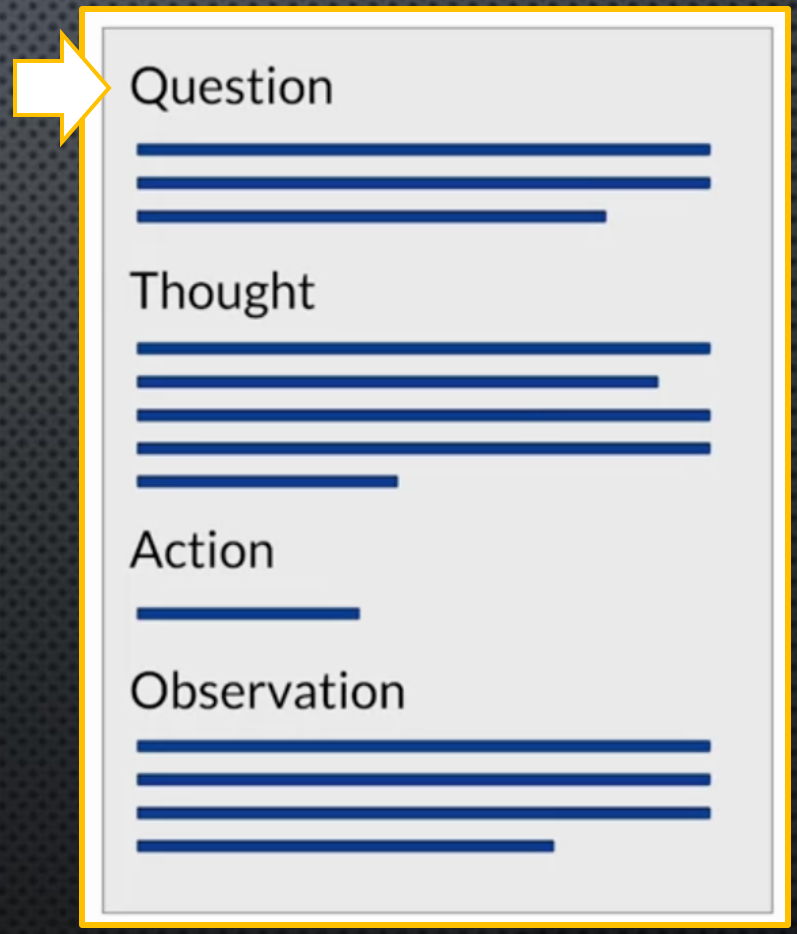
[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.

[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.

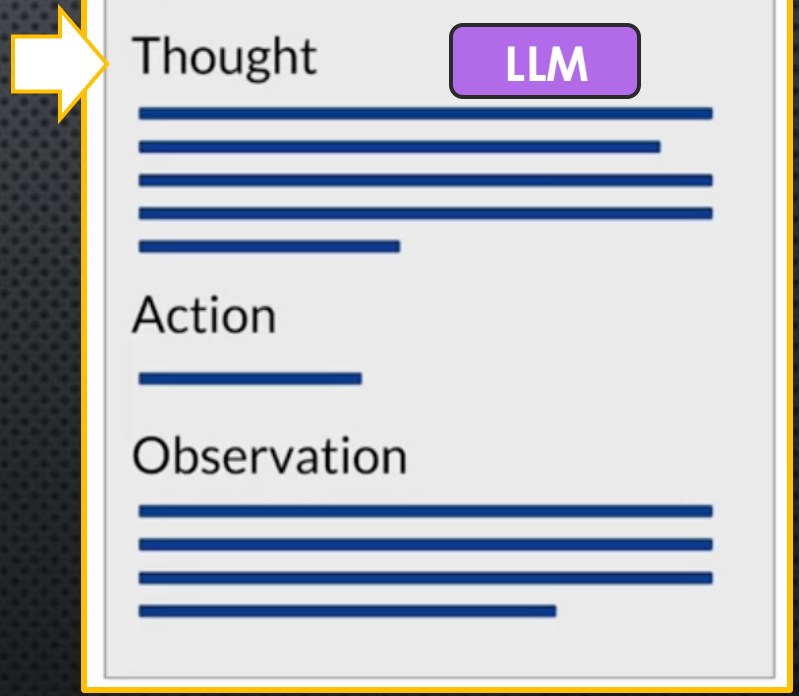
The question requires multiple steps answer.



[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

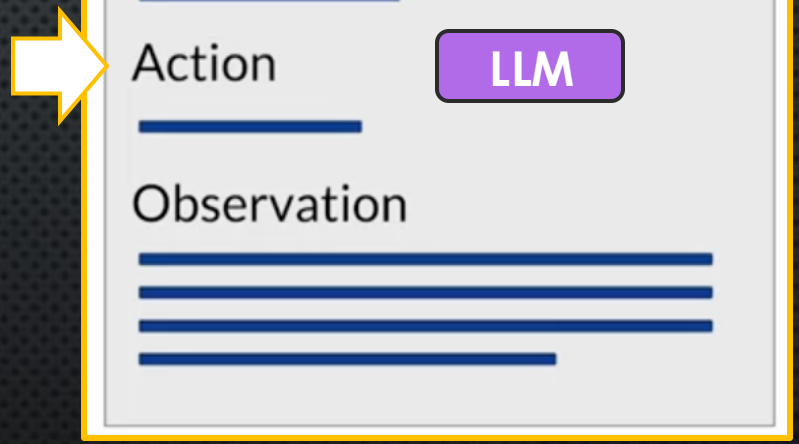
- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.

The thought is a reasoning step that demonstrates to the model how to tackle the problem and identify an action to take.



[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.



There must be a pre-determined list of actions.

In [20], the actions are:

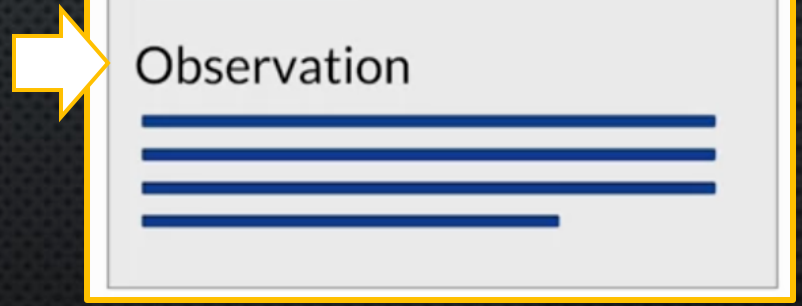
- **Search[entity]**
- **Lookup[string]**
- **Finish[answer]**

From wikipedia database.

[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.

The result of the action taken above is added here.
1 action at a time



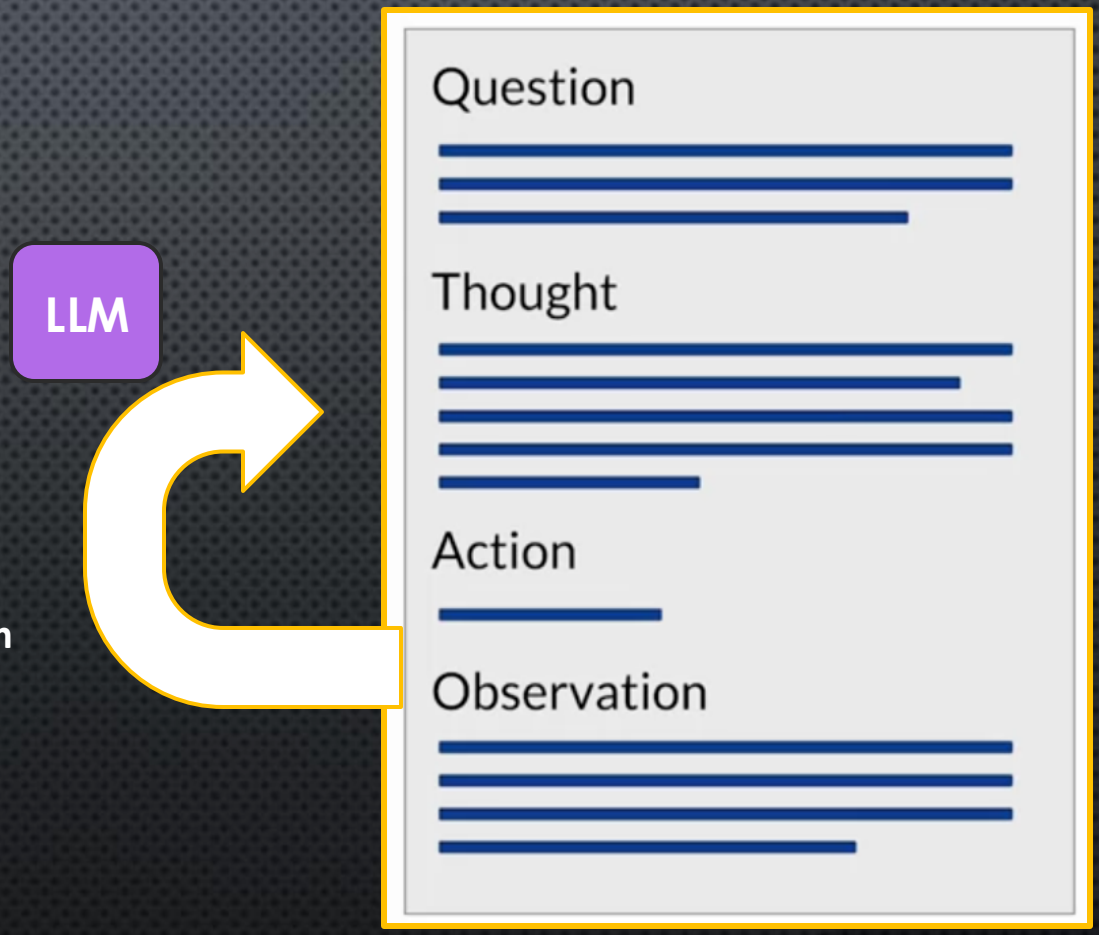
[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

- **ReAct: A framework for combining reasoning and action**
- **Note: the LLM will be called for inference multiple times, to determine the actions to take.**
- **ReAct** uses **structured examples** to show a large language model how to **reason through a problem** and **decide on actions to take** that move it closer to a solution.

The process is repeated multiple time, to retrieve all the information to solve the task by taking actions.

To generate the thought the LLM is instructed to choose from the available actions [A1,A2,A3].

The new thought i+1 comes from the current context i with new observations.



[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

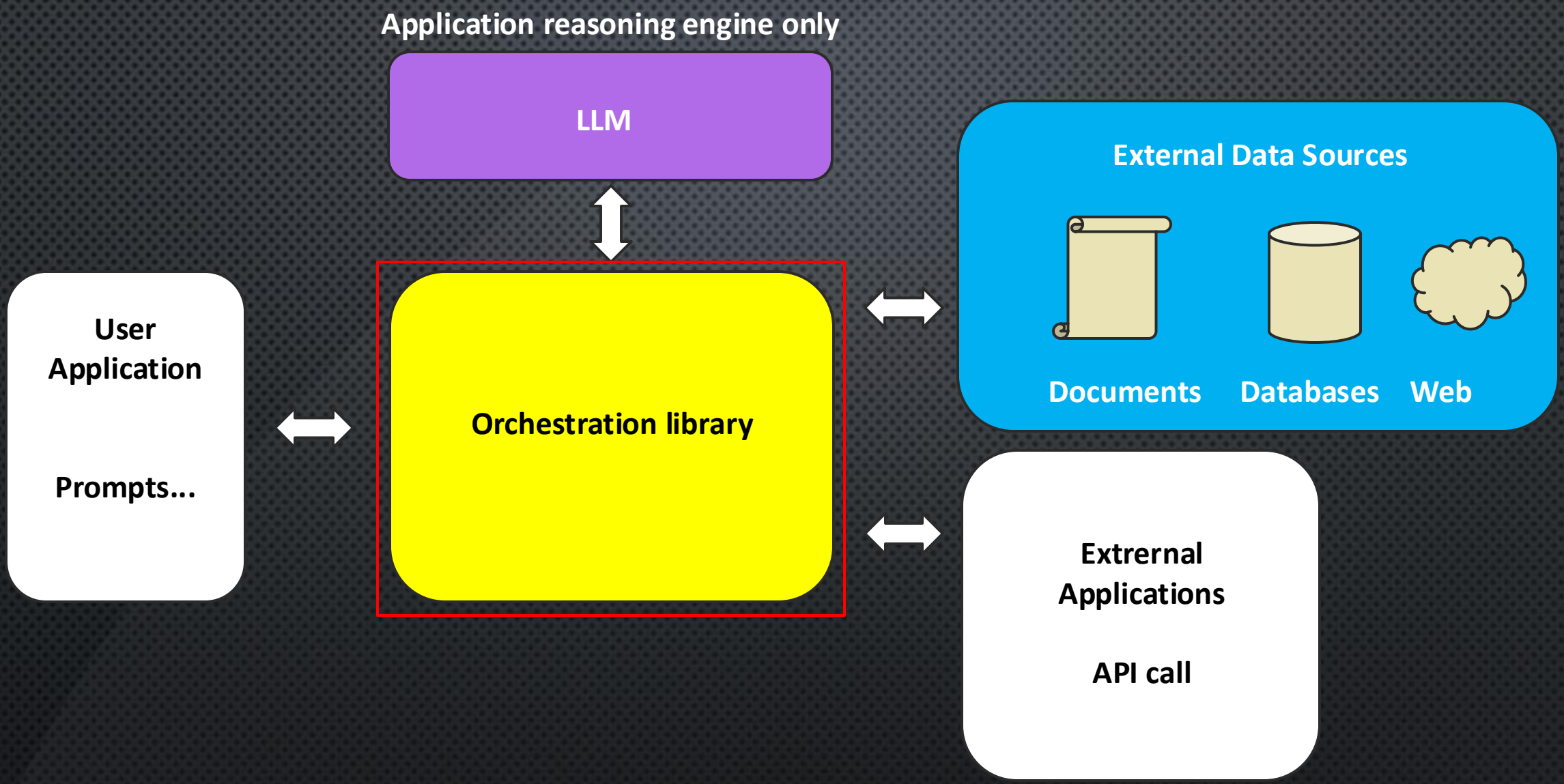
- ReAct: A framework for combining reasoning and action
- Note: the LLM will be called for inference multiple times, to determine the actions to take.
- ReAct uses structured examples to show a large language model how to reason through a problem and decide on actions to take that move it closer to a solution.

From the paper [20] by Google:

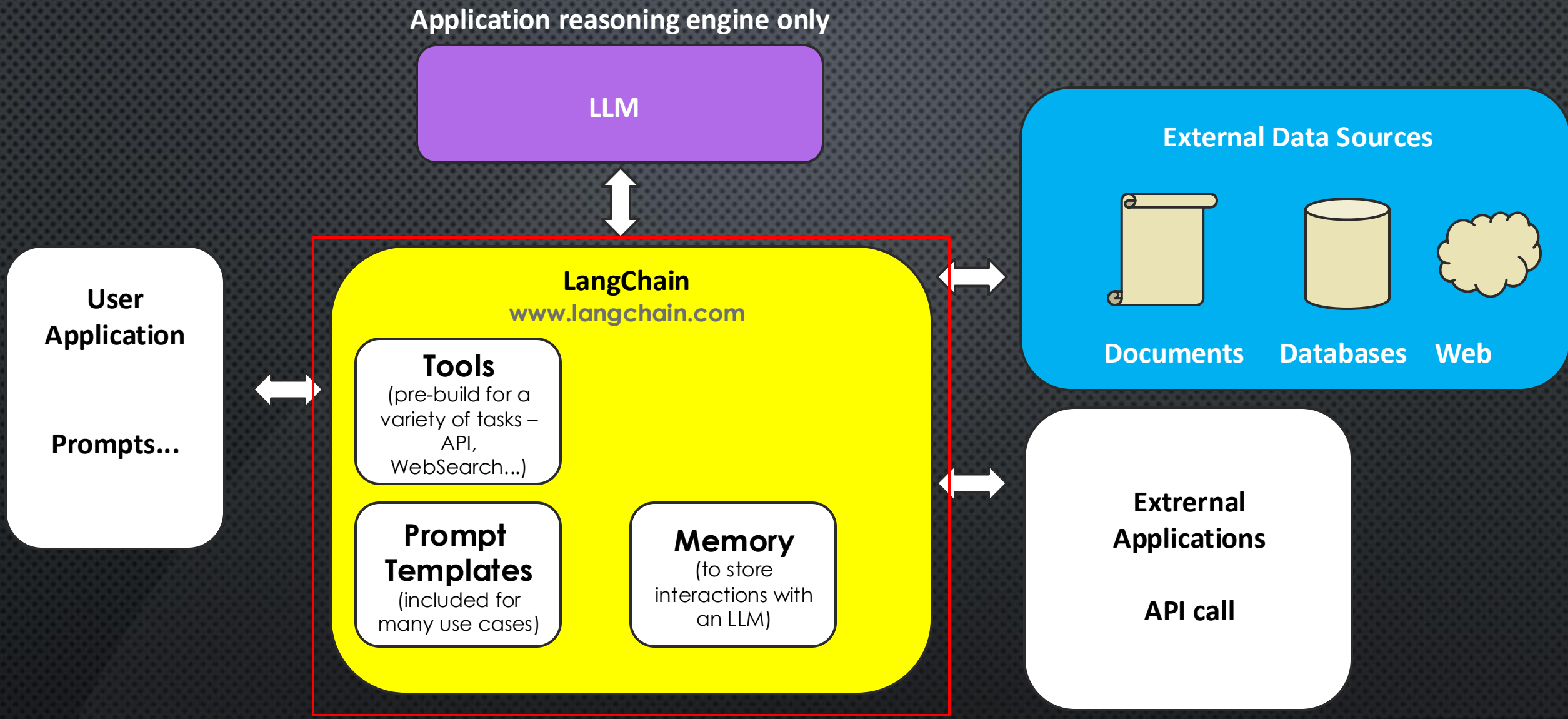
- ReAct outperforms imitation and reinforcement learning methods in interactive decision making, even with minimal context examples.
- ReAct overcomes issues like hallucination and error propagation, provides interpretability and trustworthiness. (*we know from where the result come from*).
- The generation of these thoughts and actions is a result of the model's interaction with its environment and the specific task prompts provided, which guide the model on how to alternate between thinking and acting to solve tasks.
- The ReAct framework utilizes a "frozen" large language model, specifically referencing PaLM-540B for its experiments.
- The model is prompted with few-shot in-context examples to generate both domain-specific actions and free-form language thoughts for task solving.
- This indicates that the foundational LLM model used in ReAct is not fine-tuned.

[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**

Application reasoning engine only



[21] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. In International Conference on Learning Representations (ICLR) **2023** Jan. **Google**



[22] Pandya K, Holia M. **Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations**. arXiv e-prints. **2023** Oct:arXiv-2310.

Application reasoning engine only

LLM

User Application

Prompts...

LangChain
www.langchain.com

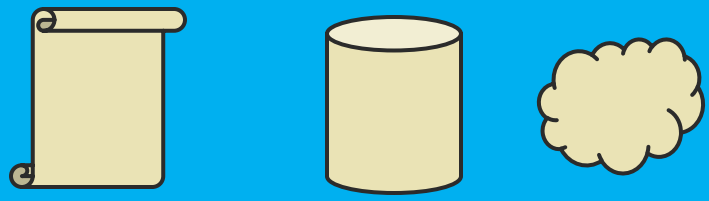
Tools
(pre-build for a variety of tasks – API, WebSearch...)

Agents
(what chain to use to complete the task)

Prompt Templates
(included for many use cases)

Memory
(to store interactions with an LLM)

External Data Sources



Documents Databases Web

Extrernal Applications

API call

[22] Pandya K, Holia M. **Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations**. arXiv e-prints. **2023** Oct:arXiv-2310.

There are different chains (pipeline of solutions) for each task.

Application reasoning engine only

LLM

User Application

Prompts...

LangChain

www.langchain.com (Python)

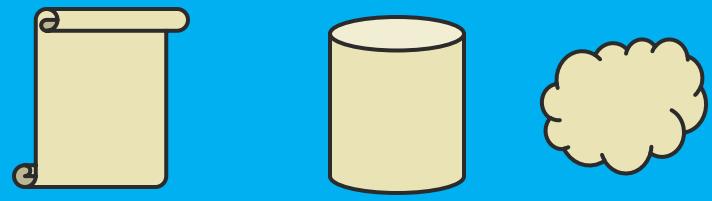
Tools
(pre-build for a variety of tasks – API, WebSearch...)

Agents
(PAL, ReAct, ..)

Prompt Templates
(included for many use cases)

Memory
(to store interactions with an LLM)

External Data Sources



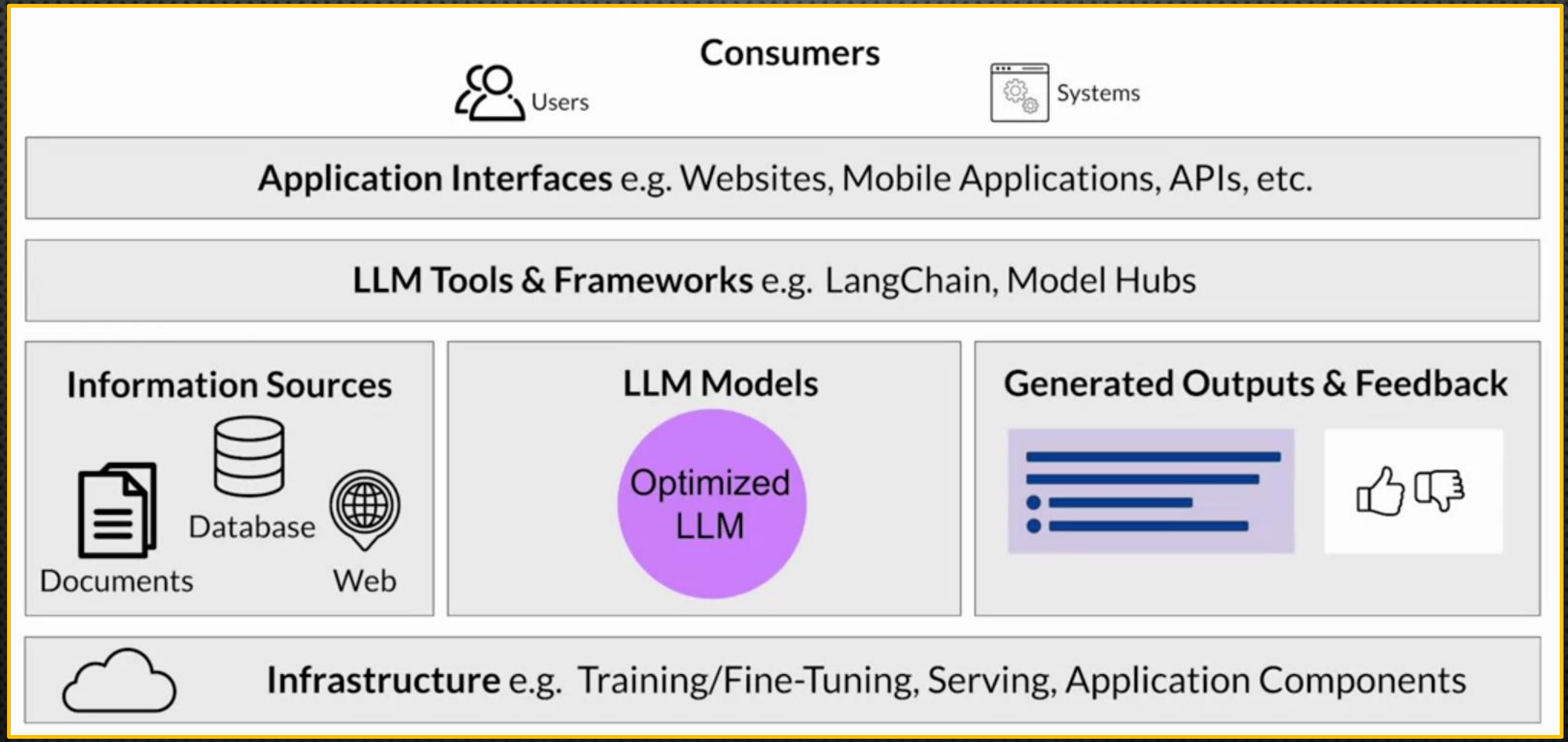
Documents Databases Web

External Applications

API call

[22] Pandya K, Holia M. **Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations**. arXiv e-prints. **2023** Oct:arXiv-2310.

The model (LLM) is only one part of the story



References

Special thanks to Shelbee Eigenbrode, Antje Barth, and Mike Chambers for their work on "Generative AI with Large Language Models" (Coursera, Amazon AWS, 2023), which significantly informed and inspired the material used in these slides.

[9] Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G. **Bloomberggpt: A large language model for finance**. arXiv preprint arXiv:2303.17564. 2023 Mar 30.

*[2] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.*

*[11] Lialin V, Deshpande V, Rumshisky A. **Scaling down to scale up: A guide to parameter-efficient fine-tuning**. arXiv preprint arXiv:2303.15647. 2023 Mar 28.*

*[12] Hu EJ, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. **LoRA: Low-Rank Adaptation of Large Language Models**. In International Conference on Learning Representations 2021 Oct 6.*

[14] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

[16] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. **Proximal policy optimization algorithms**. arXiv preprint arXiv:1707.06347. 2017 Jul 20.

*[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. **Retrieval-augmented generation for knowledge-intensive nlp tasks**. Advances in Neural Information Processing Systems. 2020;33:9459-74.*

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37.

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. Pal: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR.

[20] Topsakal O, Akinci TC. **Creating large language model applications utilizing langchain: A primer on developing llm apps fast.** In International Conference on Applied Engineering and Natural Sciences 2023 Jul (Vol. 1, No. 1, pp. 1050-1056). https://www.langchain.com/

[21] Yao S, Zhao J, Yu D, Shafran I, Narasimhan KR, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. InNeurIPS 2022 Foundation Models for Decision Making Workshop 2022 Nov 18.

[22] Pandya K, Holia M. **Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations**. arXiv e-prints. **2023** Oct:arXiv-2310.

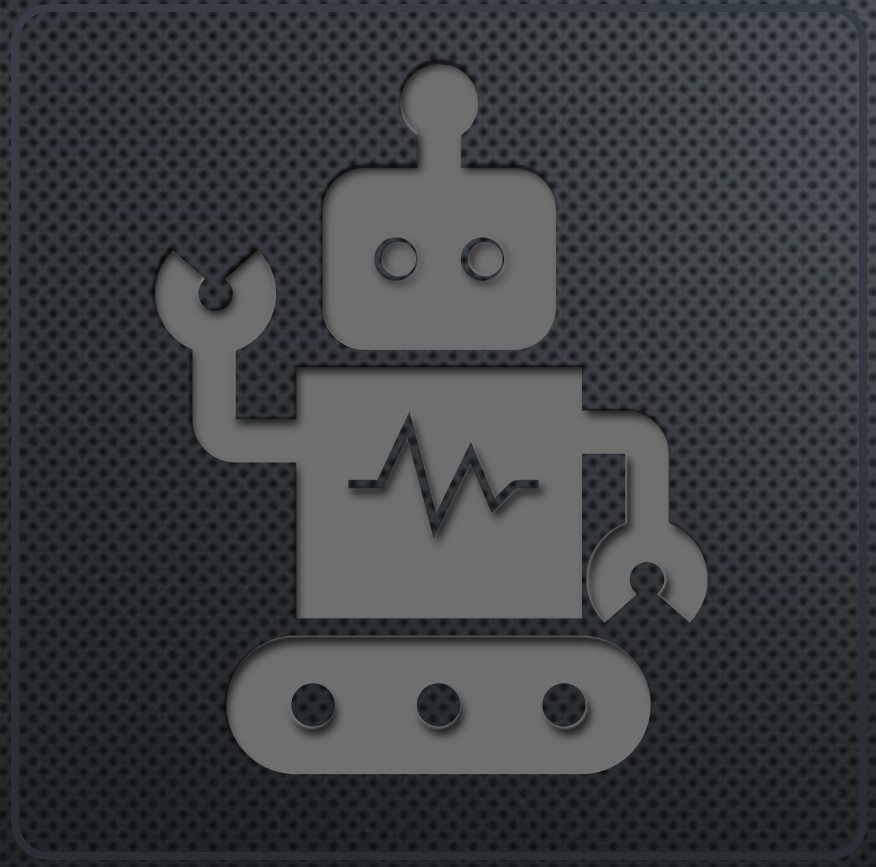
1. **Transformer Architecture**
   - Attention is All You Need
     This paper introduced the Transformer architecture, with the core "self-attention" mechanism. This article was the foundation for LLMs.
   - BLOOM: BigScience 176B Model
     BLOOM is a open-source LLM with 176B parameters trained in an open and transparent way. In this paper, the authors present a detailed discussion of the dataset and process used to train the model + high-level overview of the model here
   - Scaling Laws for Neural Language Models
     Empirical study by researchers at OpenAI exploring the scaling laws for large language models.

2. **Model architectures and pre-training objectives**
   - What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?
     The paper examines modeling choices in large pre-trained language models and identifies the optimal approach for zero-shot generalization.
   - HuggingFace Tasks and Model Hub - Collection of resources to tackle varying machine learning tasks using the HuggingFace library.
   - LLaMA: Open and Efficient Foundation Language Models
     Article from Meta AI proposing Efficient LLMs (their model with 13B parameters outperform GPT3 with 175B parameters on most benchmarks)

3. **Scaling laws and compute-optimal models**
   - Language Models are Few-Shot Learners
     This paper investigates the potential of few-shot learning in Large Language Models.
   - Training Compute-Optimal Large Language Models
     Study from DeepMind to evaluate the optimal model size and number of tokens for training LLMs. Also known as "Chinchilla Paper".
   - BloombergGPT: A Large Language Model for Finance
     LLM trained specifically for the finance domain, a good example that tried to follow chinchilla laws.

Thank you for your time and attention

Any Questions?



INTRODUCTION TO ADVANCED LARGE LANGUAGE MODELS

A Comprehensive Tutorial on Techniques, Architectures, and Practical Applications

Giorgio Roffo, PhD

Explore and Connect With My Professional Work:

LinkedIn: Giorgio Roffo - Linkedin
ResearchGate: Work Done
Google Scholar: My Publications
GitHub: giorgioroffo