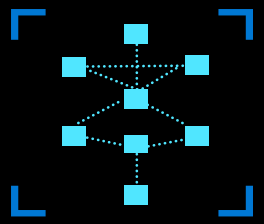
Challenge Deep-Dive

Microsoft

Embeddings

Embeddings



An embedding is a special format of data representation that can be easily utilized by machine learning models and algorithms.

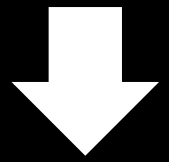
The embedding is an information dense representation of the semantic meaning of a piece of text.

Each embedding is a vector of floating-point numbers, such that the distance between two embeddings in the vector space is correlated with semantic similarity between two inputs in the original format.

For example, if two texts are similar, then their vector representations should also be similar.

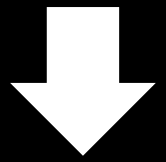
Embeddings make it possible to map content to a “semantic space”

A neutron star is the collapsed core of a massive supergiant star



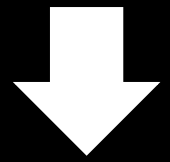
[15 34 24 13 …]

A star shines for most of its active life due to thermonuclear fusion.



[16 22 89 26 …]

The presence of a black hole can be inferred through its interaction with other matter

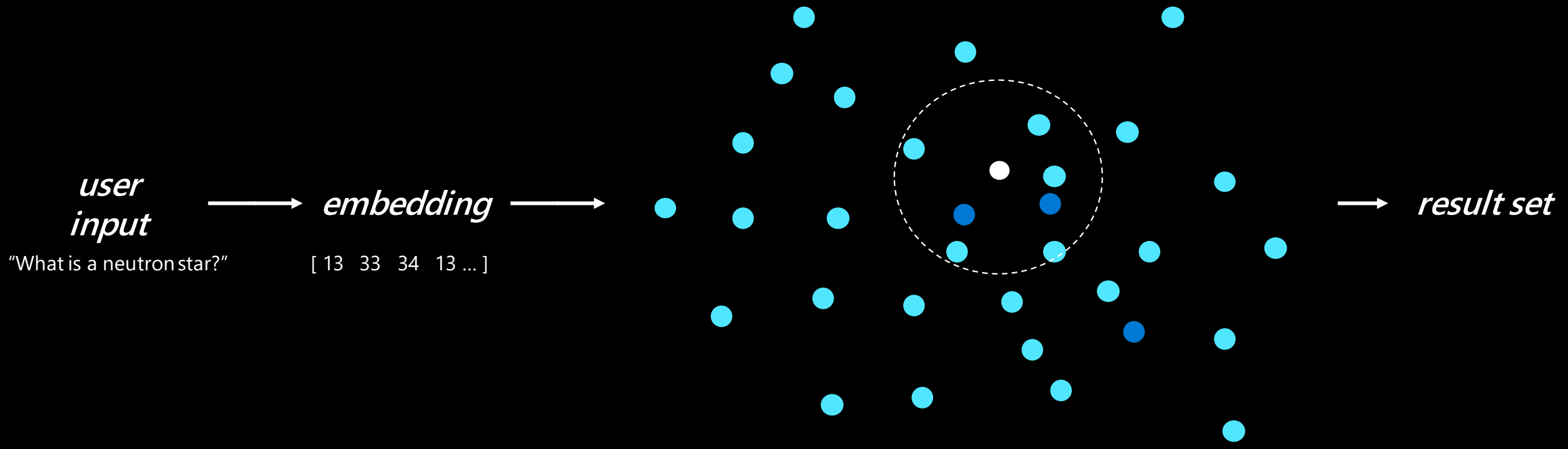


[20 13 31 89 …]

Similarity Search with embeddings

DEMO

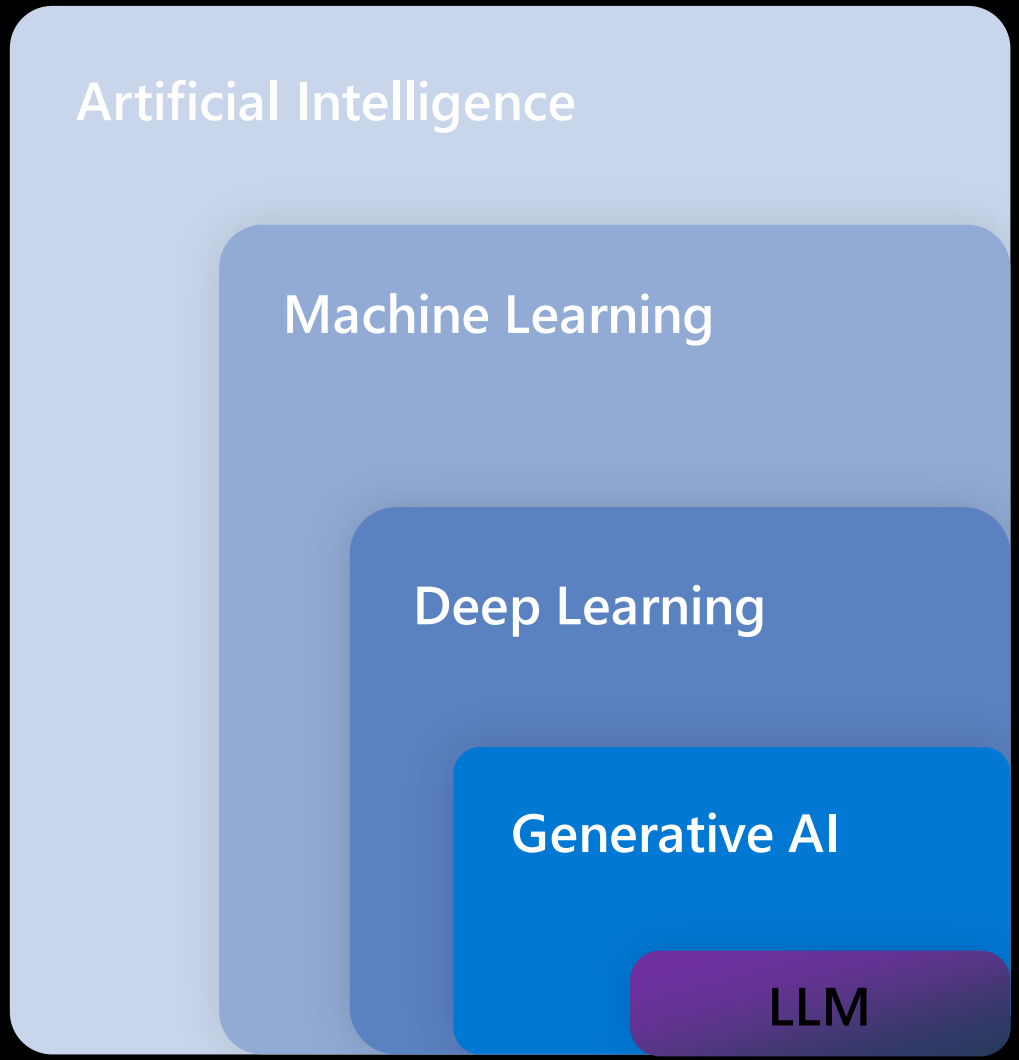
Once you encode your content as embeddings, you can then get an embedding from the user input and use that to find the most semantically similar content.



Microsoft

LLM and Retrieval-Augmented Generation

What can Large Language Model (LLM) do for you?



Provide recommendations

Write a tagline for an ice cream shop.

We serve up smiles with every scoop!

Answering questions

Who won the 1st ESC?

The Eurovision Song Contest (ESC) was won by Lys Assia, representing Switzerland, in 1956. She won with the song "Refrain".

Answering questions

Who won ESC in 2023?

I'm sorry, but I do not have access to information beyond my last knowledge update in Sep. 2021.

Answering questions

What is included in my Contoso Health Plus plan?

I apologize, but I don't have access to your specific health insurance plan or personal information.

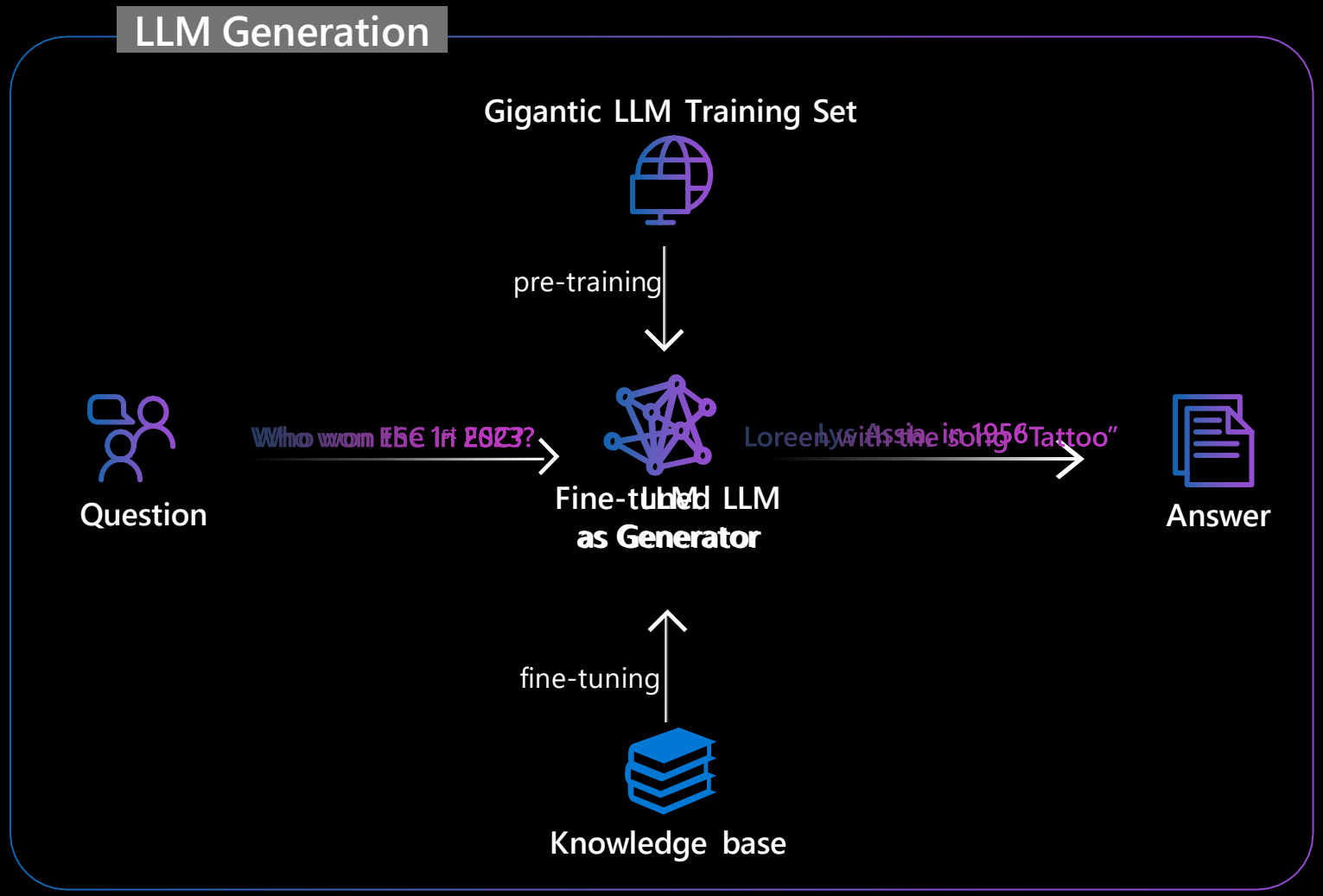
Unlocking LLM opportunities: Addressing challenges

Struggle with having up-to-date LLM models that live beyond their training lifecycles

Having LLM backed by organization's own knowledge base upon which it wasn't originally trained

Concerns about LLM side effects like hallucination and desire to have verifiable data sources in a cost-efficient way

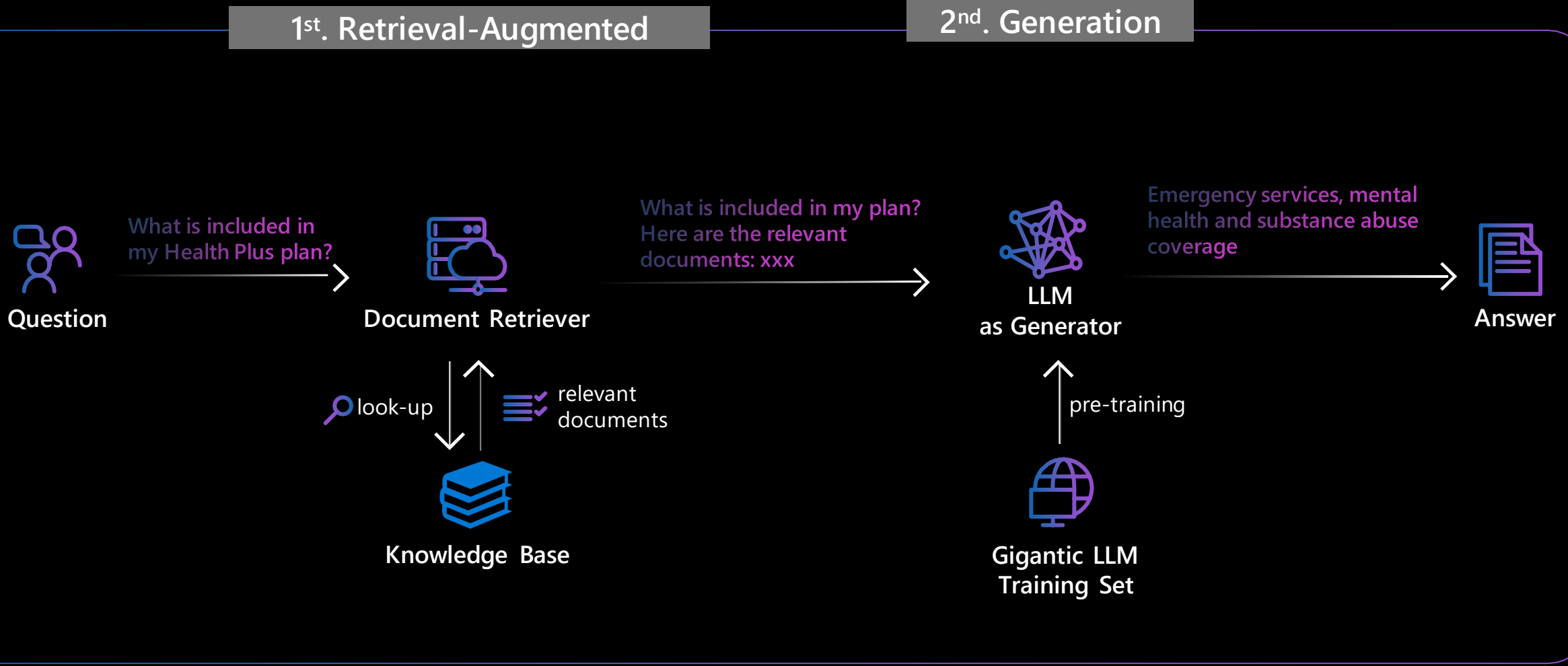
Observations of LLM generation flow



Technical blockers

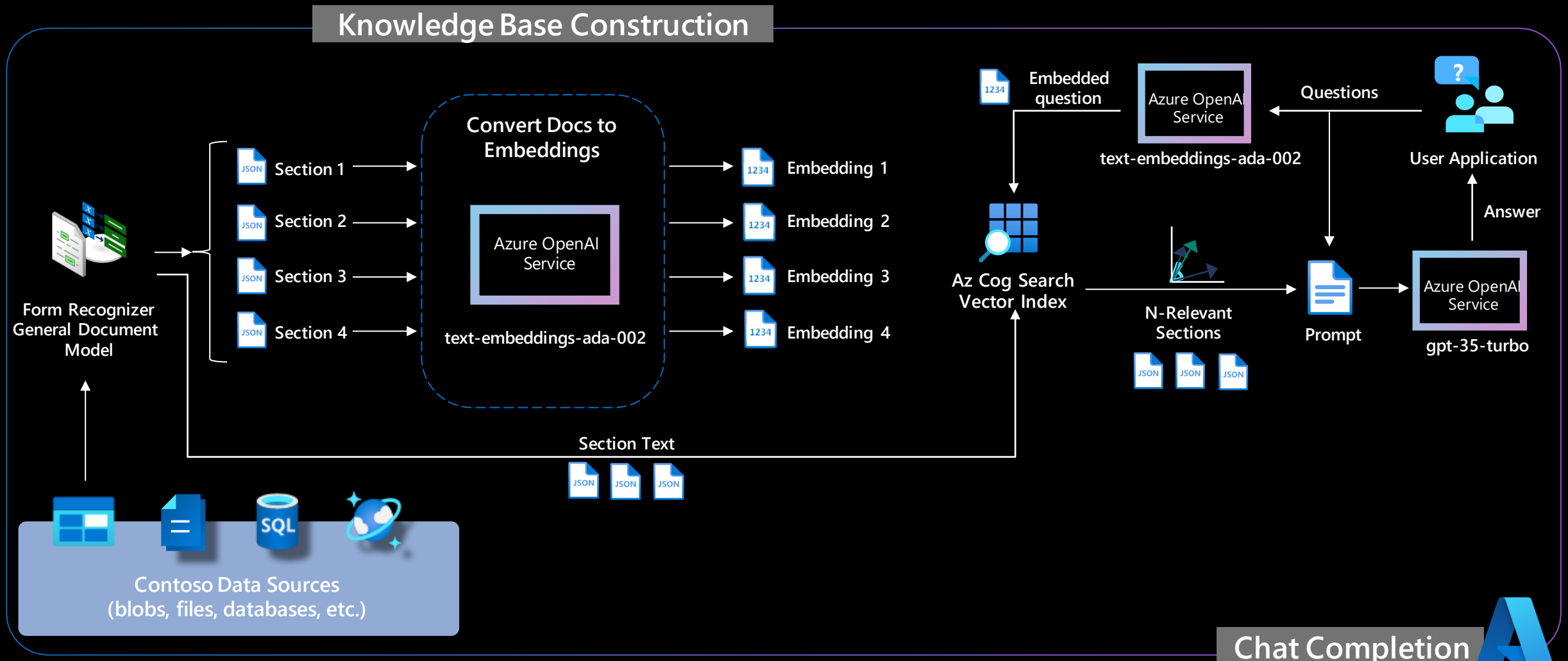
- **No source clarity**
  LLM has no clear distinction between general and specific knowledge
- **No access restriction**
  Hard to leave out certain knowledge at inference time
- **Hosting an LLM is costly**
  Consider data collection, injection and model retraining
- **Fine-tuning repetitions**
  Retraining is required whenever knowledge base changes

Understanding the working principle of RAG



Azure semantic answering architecture example

Azure Implementation



Azure OpenAI Demo

LLM and Retrieval-Augmented Generation

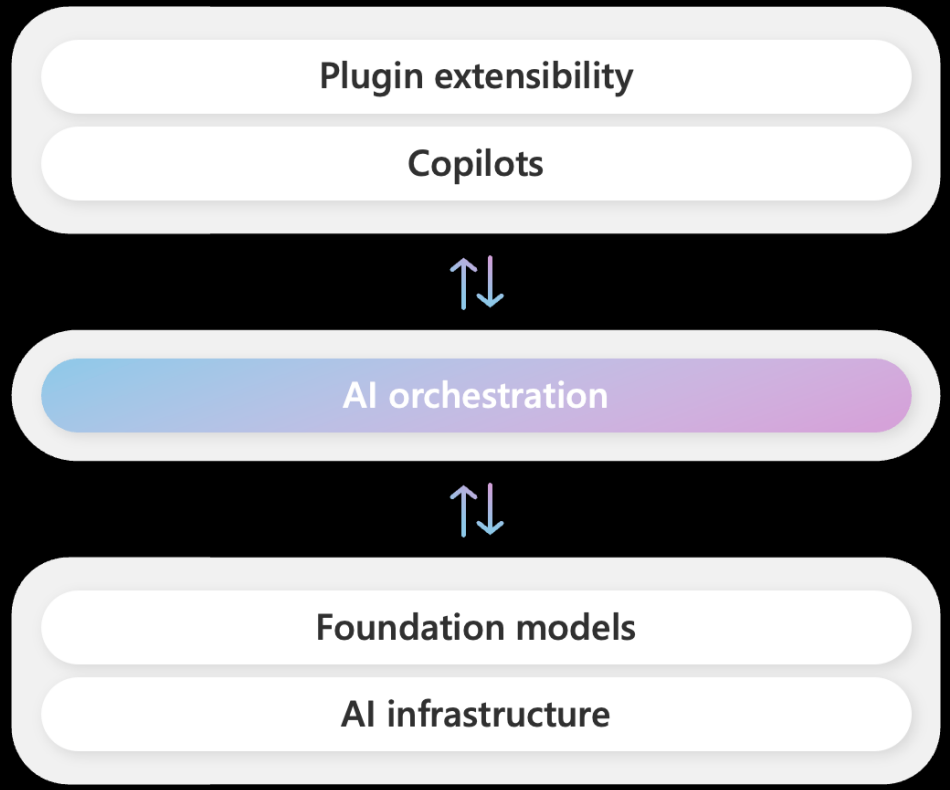
DEMO

Microsoft

AI Orchestration and Semantic Kernel

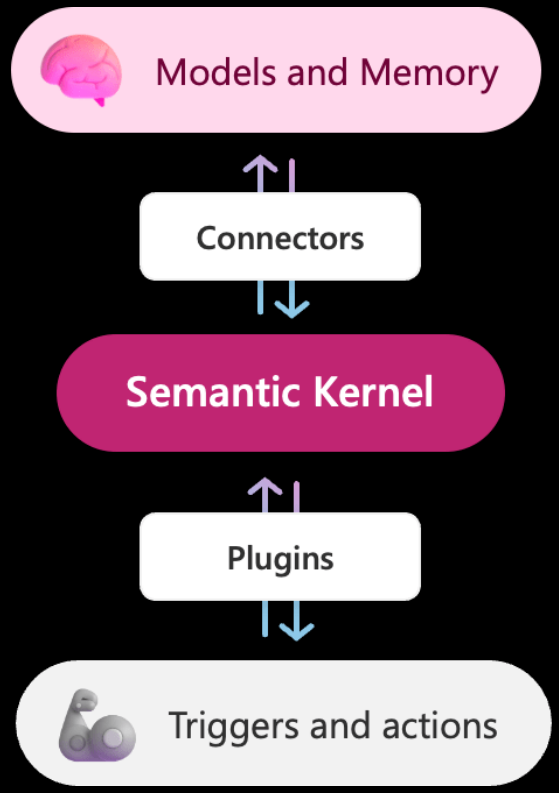
What is Semantic Kernel?

- Semantic Kernel is an open-source SDK that lets you easily combine AI services like OpenAI, Azure OpenAI with conventional programming languages like C# and Python.
- It is at the center of the Copilot stack, allowing developers to flexibly integrate AI services into their existing apps using the same orchestration patterns that power Microsoft 365 Copilot and Bing.

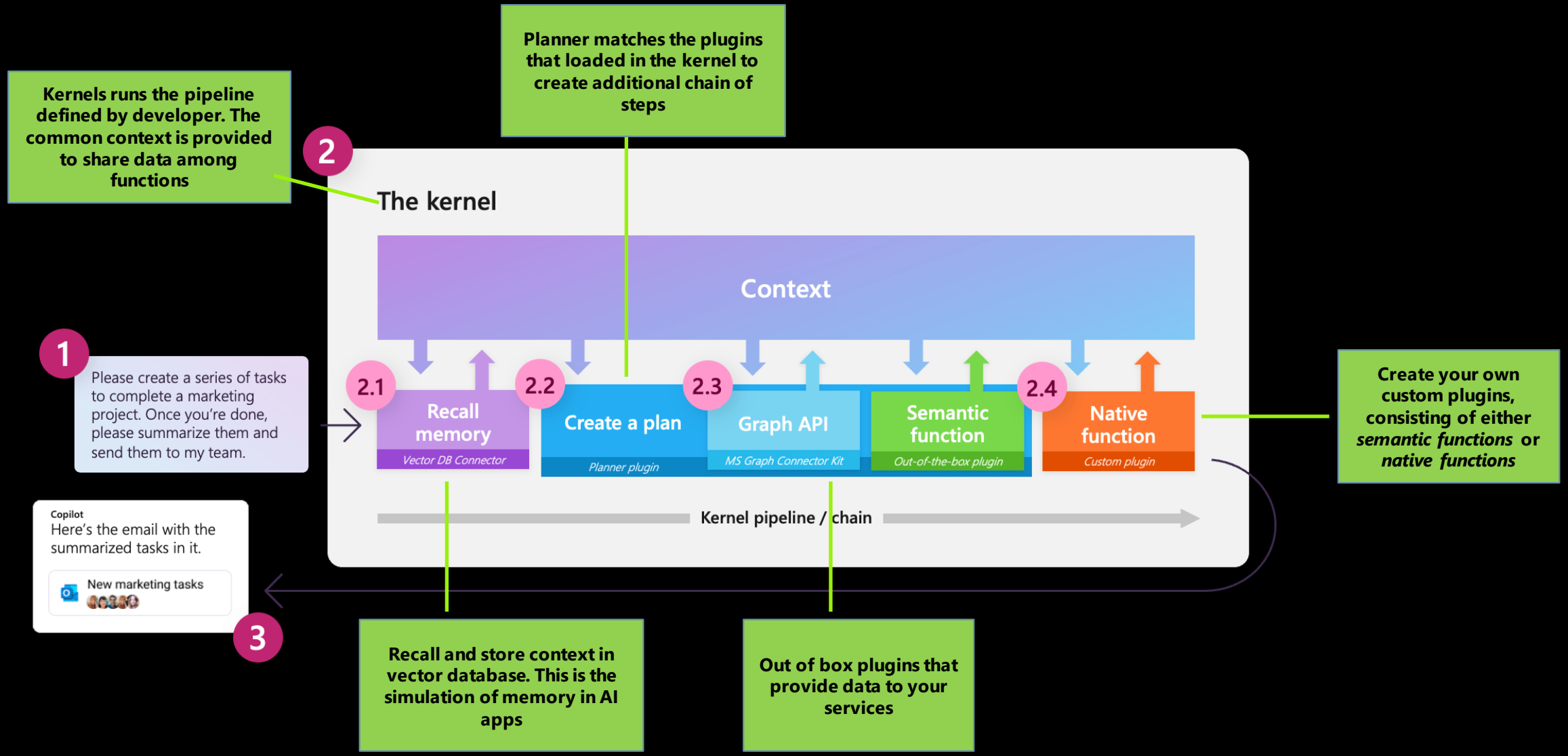


Semantic Kernel makes AI development extensible

- On one hand, semantic Kernel provides connectors for adding memories e.g. embeddings and models e.g. GPT-4
- On the other hand, semantic kernel enables to add skills to applications with AI plugins that respond to triggers and perform actions
- For example, you can use semantic kernel to orchestrate plugins built for ChatGPT and Bing on top of Azure OpenAI

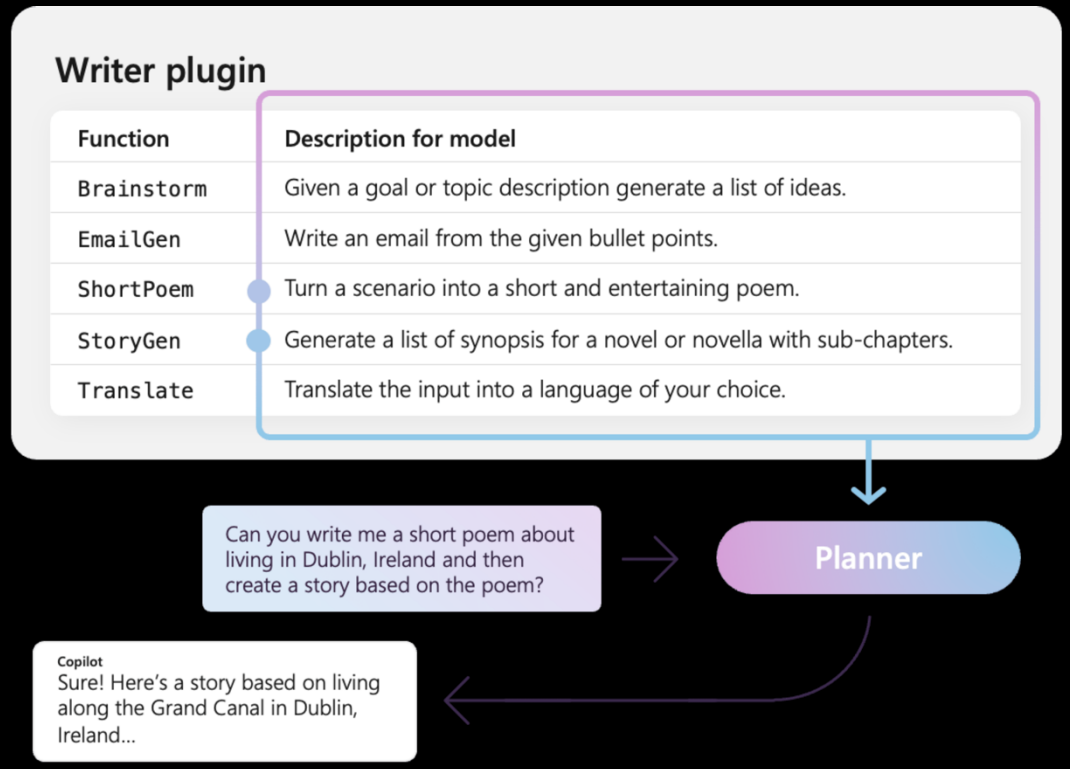


Seeing AI Orchestration with Semantic Kernel



Understanding Plugins

- Plugins are the fundamental building blocks of semantic kernel and can interoperate with plugins in ChatGPT, Bing and MS 365. This means any plugins you build can be exported so they are usable in ChatGPT, Bing or MS 365
- A plugin is a group of functions that can be invoked either manually (chaining functions) or automatically with a planner
- Everything from the function like input, output, and side effects should be well documented



Create functions for plugins

My Plugin

Create semantic function

Create native function (Python, C#)

Declaratively define semantic function with settings

config.json

```
{
  "schema": 1,
  "description": "Summarize given text or any text document",
  "models": [
    {
      "max_tokens": 512,
      "temperature": 0.0,
      "top_p": 0.0,
      "presence_penalty": 0.0,
      "frequency_penalty": 0.0
    }
  ],
  "input": {
    "parameters": [
      {
        "name": "input",
        "description": "Text to summarize",
        "defaultValue": ""
      }
    ]
  }
}
```

skprompt.txt

```
[SUMMARIZATION RULES]
DONT WASTE WORDS
USE SHORT, CLEAR, COMPLETE SENTENCES.
DO NOT USE BULLET POINTS OR DASHES.
USE ACTIVE VOICE.
MAXIMIZE DETAIL, MEANING
FOCUS ON THE CONTENT

[BANNED PHRASES]
This article
This document
This page
This material
[END LIST]

Summarize:
Hello how are you?
+++++
Hello

Summarize this
{{$input}}
+++++
```

Augmenting LLMs with native functions

Planners are able to use annotations to understand how the function behaves

```python
@sk_function(
    description="Adds value to a value",
    name="Add",
    input_description="The value to add",
)
@sk_function_context_parameter(
    name="Amount",
    description="Amount to add",
)
def add(self, initial_value_text: str, context: SKContext) -> str:
    """
    Returns the Addition result of initial and amount values provided.

    :param initial_value_text: Initial value as string to add the specified amount
    :param context: Contains the context to get the numbers from
    :return: The resulting sum as a string
    """
    return MathPlugin.add_or_subtract(initial_value_text, context, add=True)
```

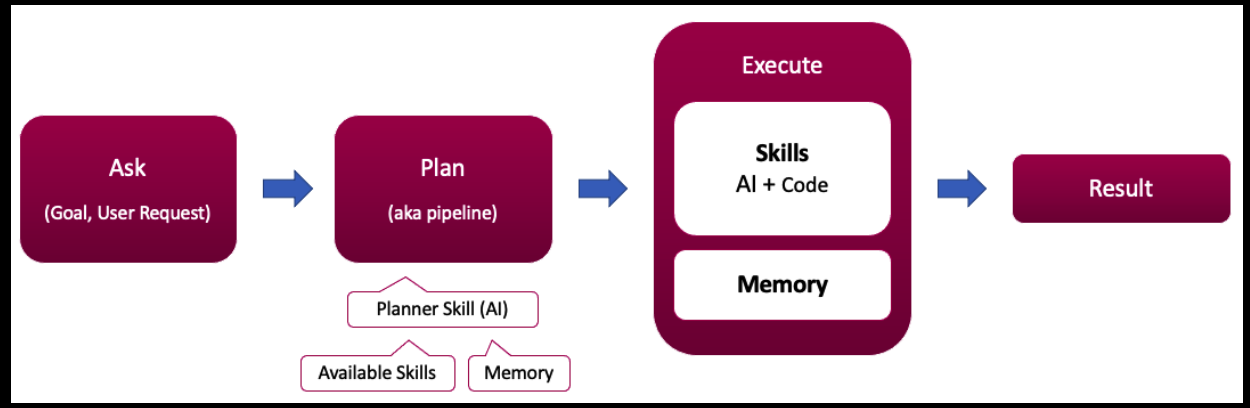
Automatically orchestrate AI with planners

- Planner is a function that takes a user's ask and returns a plan on how to accomplish the request
- Planner allows a more scalable solution as the developers don't have to predict all possible requests

Planner	Description	C#	Python	Java
BasicPlanner	A simplified version of SequentialPlanner that strings together a set of functions.	❌	✅	❌
ActionPlanner	Creates a plan with a single step.	✅	✅	❌
SequentialPlanner	Creates a plan with a series of steps that are interconnected with custom generated input and output variables.	✅	✅	❌
StepwisePlanner	Incrementally performs steps and observes any results before performing the next step.	✅	✅	❌

Glossary

- Semantic kernel is the orchestrator fulfils a user's ask
- Ask is a user request
- Plugins are domain specific function collection made available to the SK
- Function is a computation comprised of AI or native code that's available in a plugin
- Native function traditional expression in a language (Python, C#)
- Semantic function is developed using prompt engineering defined by a template file
- Memory is a collection of semantic knowledge based on facts, events, documents and indexed with embeddings

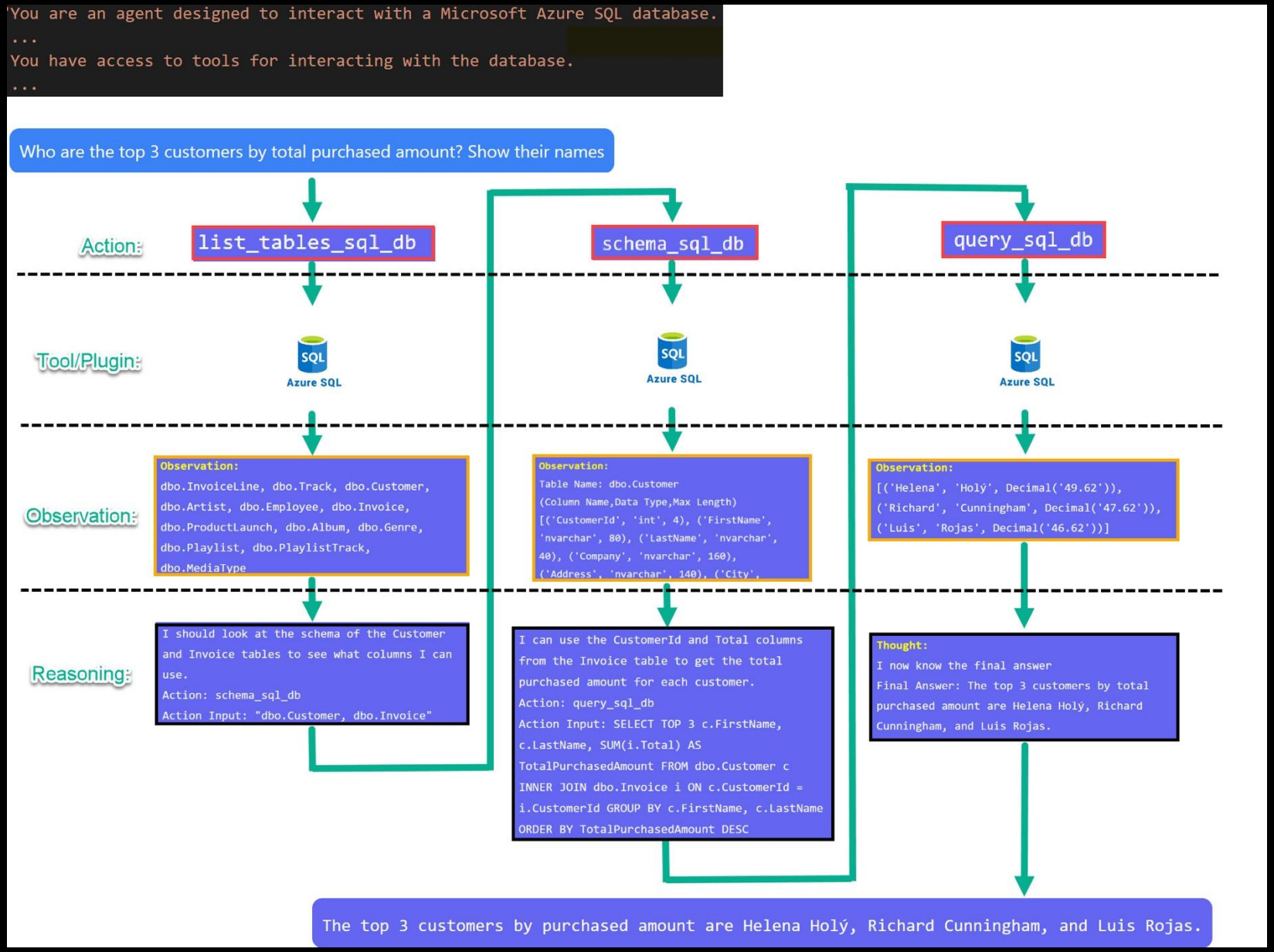


Azure OpenAI Demo

- SqlGPT using Langchain
- Semantic kernel in a nutshell
- Build chat copilot with customized plugins

SqlGPT using langchain

- How many tables are there?
- How many customer transactions are not finalized yet?
- Show me the top 3 customer names who have the highest transaction amount



Bring your own plugins to ChatGPT

For a custom plugin e.g. MathPlugin, there are 3 steps to turn this into a ChatGPT plugin:

1. Create HTTP endpoints for each native function
2. Create an OpenAPI specification and plugin manifest file that describes the plugin
3. Test the plugin in either Semantic Kernel or ChatGPT