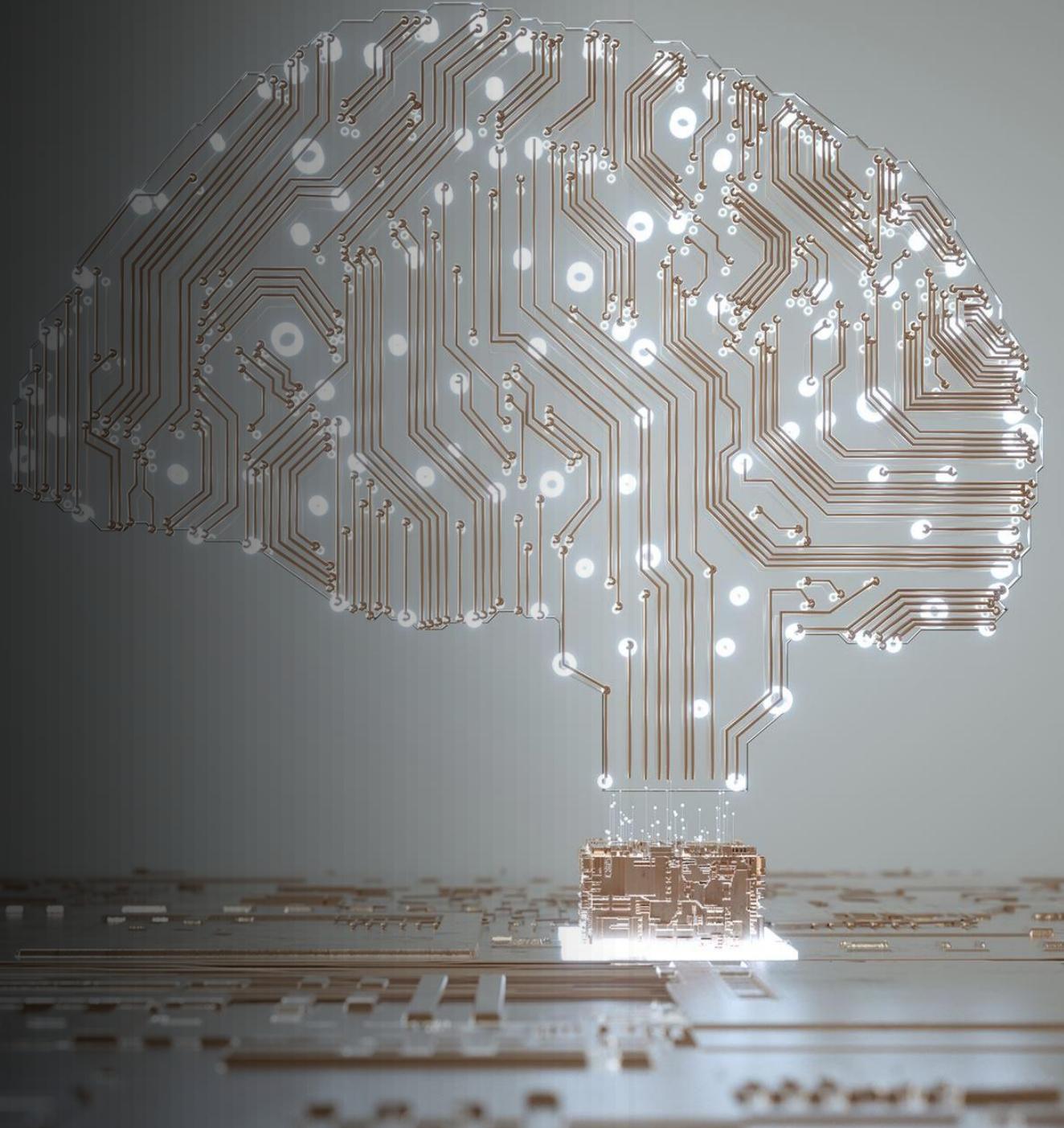


Welcome to the AI Workshop for Developers

Hosted by Microsoft



Ride the wave



Connect with Us!



Najib Zarrari

Director, Cloud Solutions Architecture



Agenda

- **9:00 am – 12:00 pm: Challenges**
 - Introduce Coaches
 - Foundational Slides
 - Challenge 01: Azure OpenAI Fundamentals
 - Challenge 02: Semantic Kernel Fundamentals
 - Challenge 03: Plugins
- **12:00 pm – 1:00 pm: Lunch, Networking**
- **1:00 pm – 4:45 pm: Challenges**
 - Challenge 04: Plugin with OpenAPI
 - Challenge 05: RAG
 - Challenge 07: DallE
 - Challenge 08: Multi-Agent





.NET Aspire

A cloud ready stack for building observable,
production ready, distributed applications

Smart Defaults

Developer Dashboard

Orchestration

Service Discovery

Components

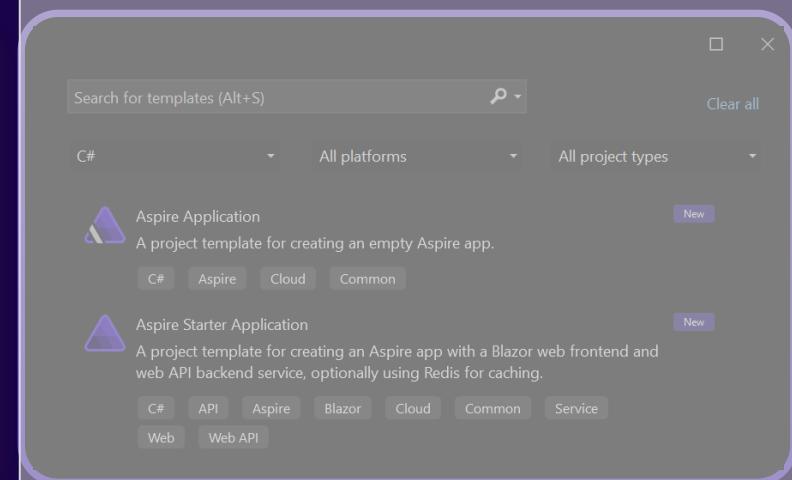
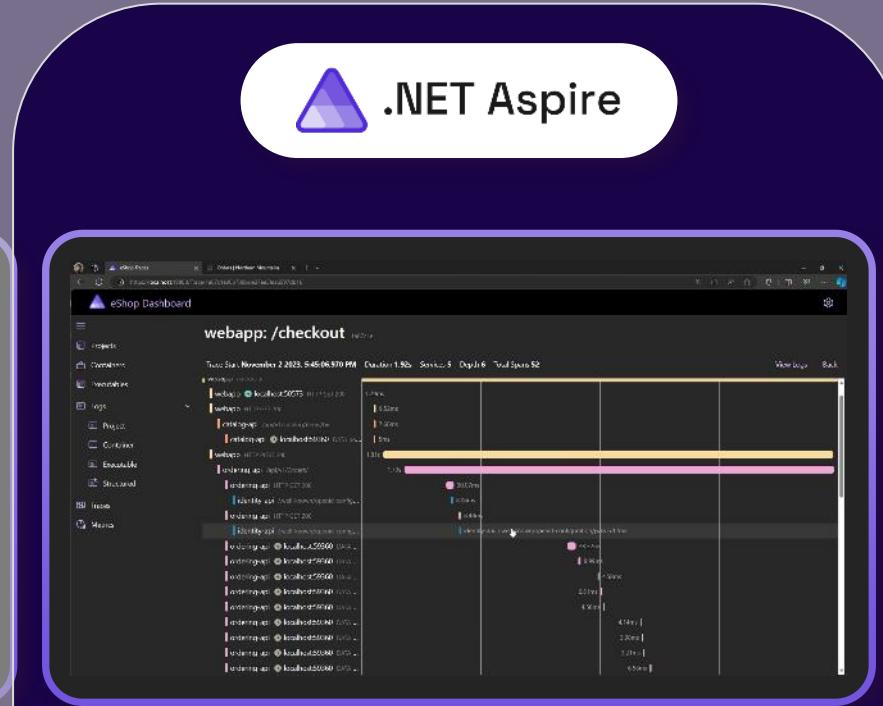
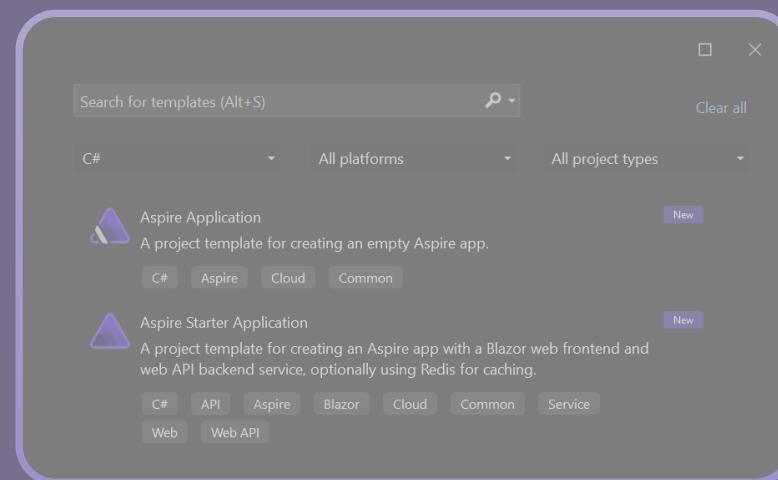
Deployment

Your average cloud dev deals with:

- Caching
- Databases
- Local development tools
- Messaging + queueing
- Containers?!
- Multiple languages
- A zillion deployment targets
- Resiliency + scalability
- Security + networking
- CI/CD
- Storage but... not databases?!
- Telemetry and logs and traces
- Authentication
- AI?!?!?!



and so much more!



Easy to get started

Open-source
Templates
Integrations

Easy to build

Service discovery
Developer dashboard
Logs, metrics, distributed traces

Easy to deploy

Single command run
App topology in C#
Cloud deployment

How to Navigate the Challenges

Use GitHub website to read the instruction



= Strong Recommendation, Notes, or
Links to Code Samples



= Important, Pay Attention, or Mandatory

Useful links can be found at the bottom under the
Learning Resources section.

Use the links at the top and bottom of
the challenge to help you navigate.

Probably the most important!
Don't skip around. You might miss a step!

Let's talk setup

Challenge #0

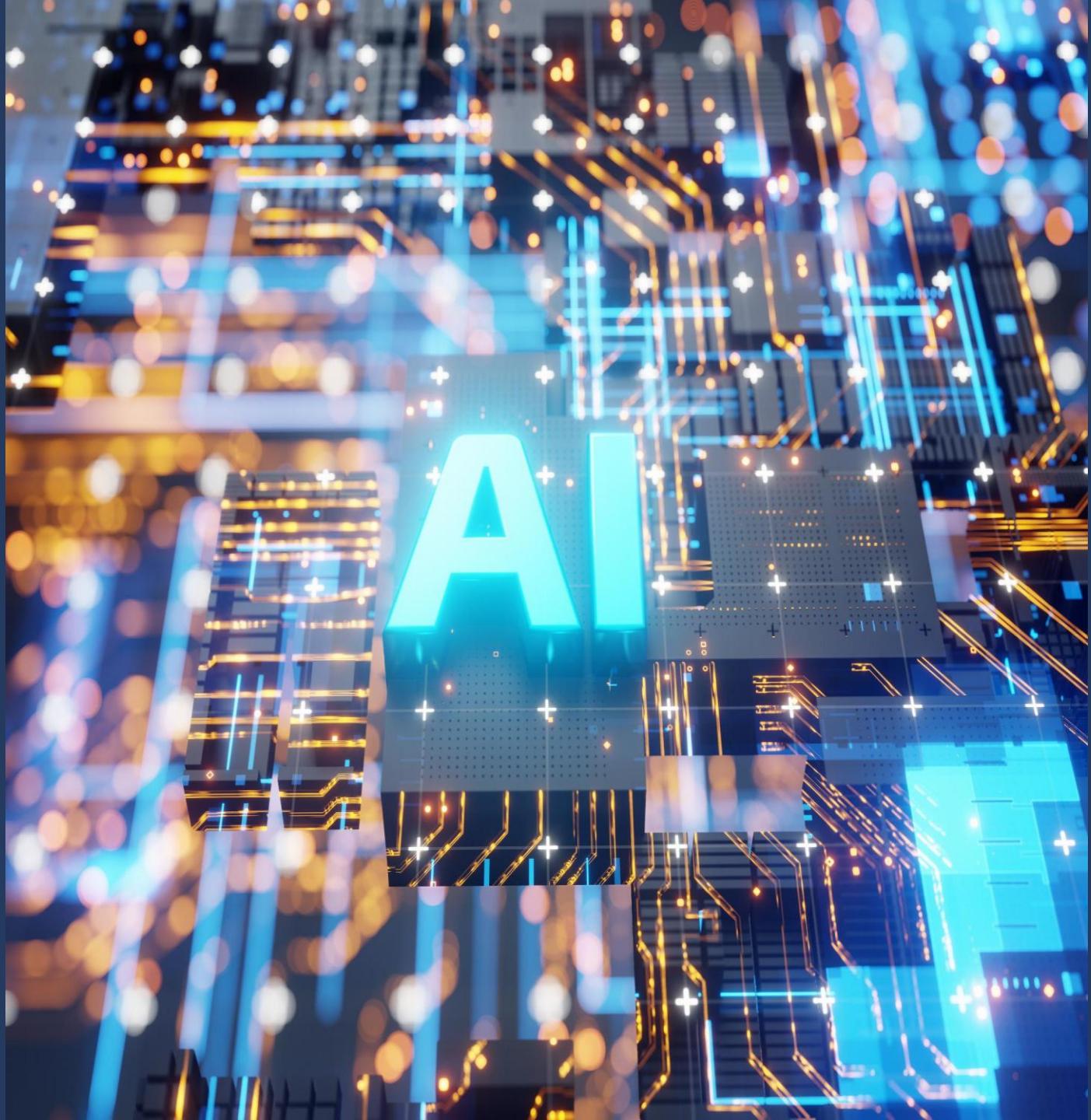
Prerequisites

- Prepare your workstation to work with Azure.
- Setup your development environment local or Dev Box.

Feel free to start the install in the background.
[Visit aka.ms/SKDev](http://aka.ms/SKDev)

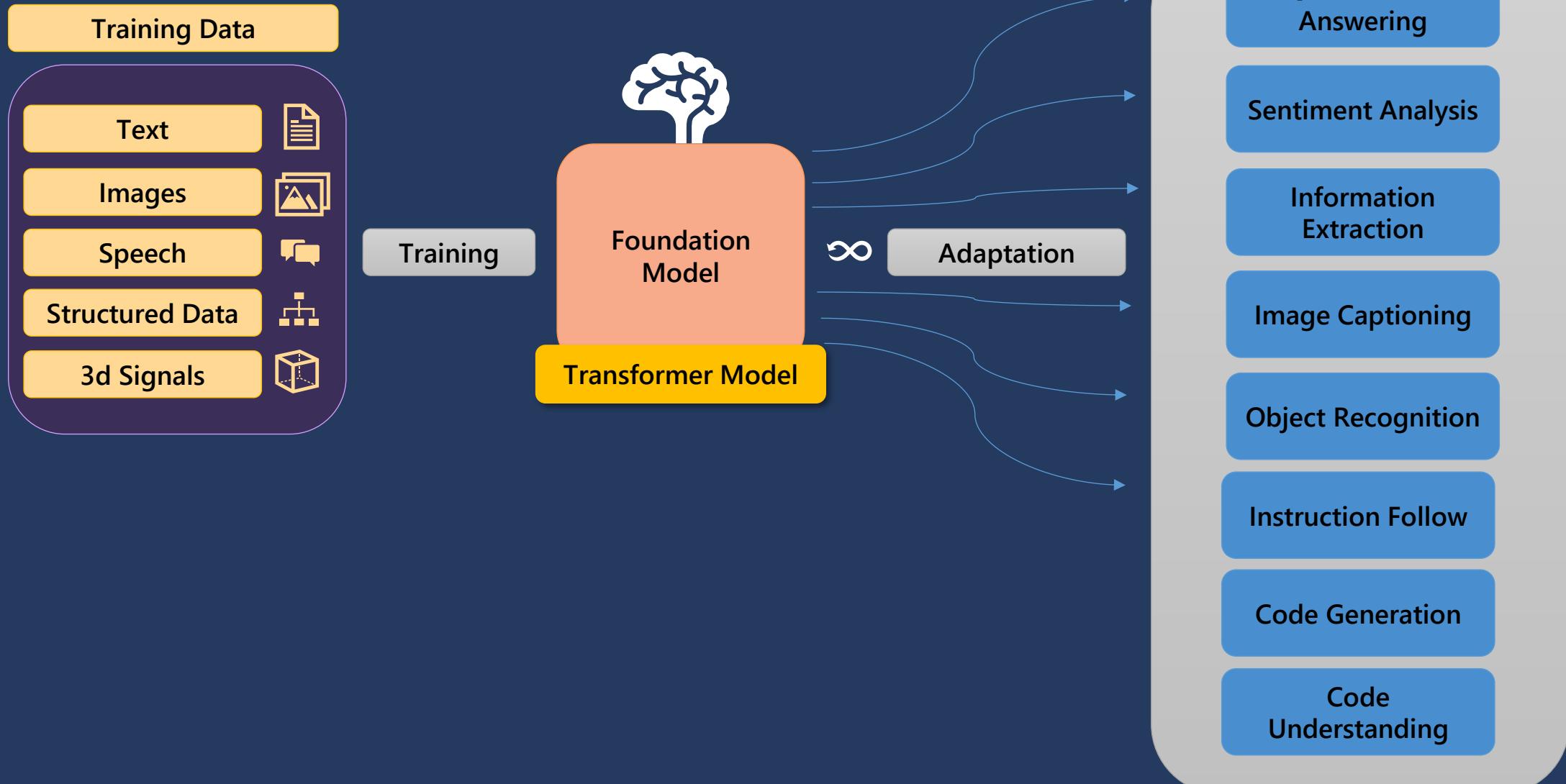
OpenAI Fundamentals

Azure OpenAI Fundamentals



Understanding and utilizing GPT and other generative AI models

Foundation Models



175B

GPT-3 | 2020

17B

Turing-NLG | 2020

340M

BERT-Large | 2018

OpenAI



Microsoft

Comparison of GPT versions

GPT-3.5

- Use-case specific models to optimize inference time and performance
- Suitable for a large range of use cases

GPT-4

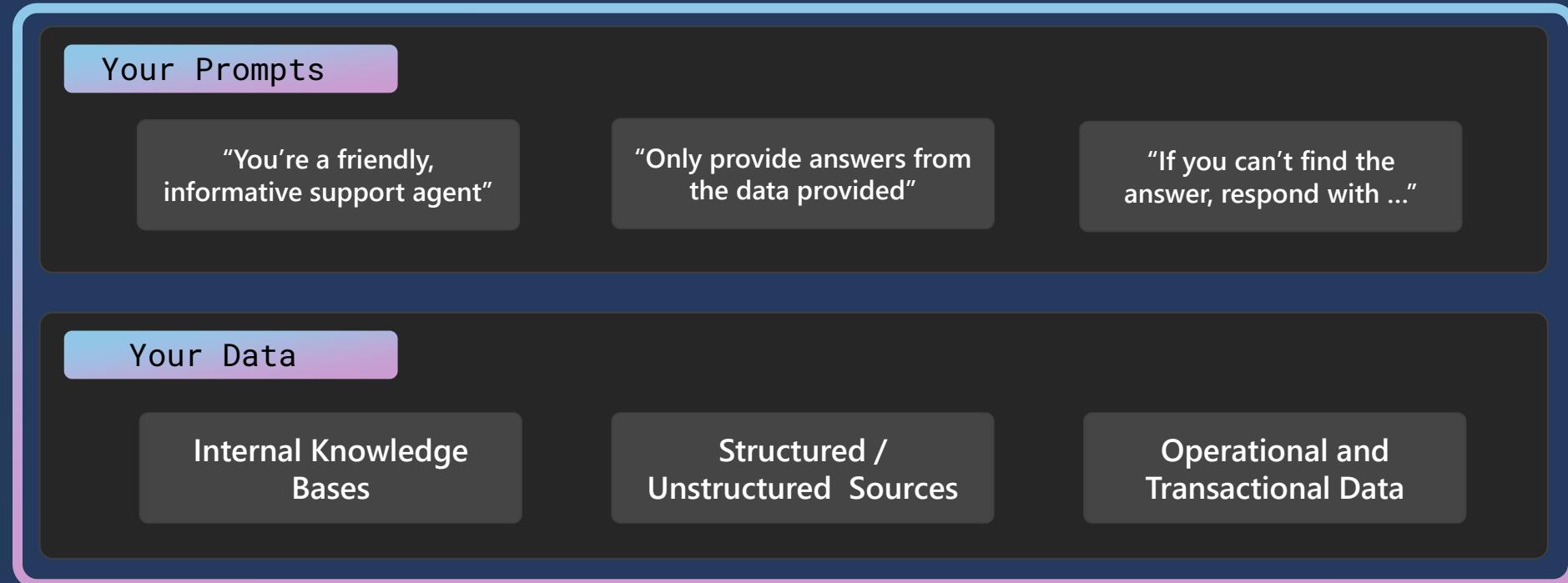
- Improved problem solving and reasoning capabilities
- Iterative refinement:
 - Paste in code errors & GPT-4 will fix for you
 - Iterate on stories
- Increased token limit - works well for long content

GPT-4o

- Should be first choice for most use cases
- Most economical GPT model in Azure OpenAI Service
- Integrates text and images in a single model, enabling it to handle multiple data types simultaneously

Customizing Azure OpenAI

Your Differentiation



Tokens

You can think of tokens as pieces of words used for natural language processing. For English text, 1 token is approximately 4 characters or 0.75 words.

As a point of reference, the collected works of Shakespeare are about 900,000 words or 1.2M tokens.

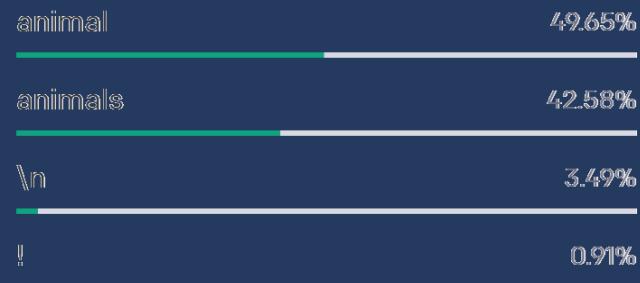
Understanding tokens and possibilities

Tokens:

I have an orange cat named Butterscotch.

I have an orange cat named **Butterscotch**.

Horses are my favorite



Probabilities:

IF TEMPERATURE IS 0
Horses are my favorite **animal**
Horses are my favorite **animal**
Horses are my favorite **animal**
Horses are my favorite **animal**

IF TEMPERATURE IS 1
Horses are my favorite **animal**
Horses are my favorite **animals**
Horses are my favorite **!**
Horses are my favorite **animal**

Prompt Instruction

Suggest three names for an animal that is a superhero.

Animal: Cat

Names: Captain Sharpclaw, Agent Fluffball, The Incredible Feline

Animal: Dog

Names: Ruff the Protector, Wonder Canine, Sir Barks-a-Lot

Animal: Horse

Names:

Completion Temperature 0 (always the same)

Mighty Equine, The Great Galloper, Thunderhoof

Completion Temperature 1 (often different)

Blaze the Miracle Mare, Pegasus the Winged Warrior, Secretariat the Superhorse

Completion Temperature 1 (often different)

Blaze of Glory, Sterling Silver, Thunderbolt

Prompt Engineering



| Azure OpenAI Service | GPT-3 Prompt Design

Extract the mailing address from this email:

Hi John Doe,

It was great to meet up at Build earlier this week. I thought the AI platform talk was great and I really enjoyed it.

I appreciate the offer for the book. If you are OK, you can mail it to me at home, or 123 Microsoft Way, Bellevue WA 92004.

Regards,

Chris Hoder

Prompt – Text input that provides some context to the engine on what is expecting.

Completion – Output that GPT-3 generates based on the prompt.

What is Prompt Engineering?

- Prompt engineering is a concept in Natural Language Processing (NLP) that involves embedding descriptions of tasks in input to prompt the model to output the desired results.
- Prompt typically includes problem descriptions, instructions on how to solve the problem, and examples of correct problem and solution pairs.

Prompt Engineering Strategies

- Chain of Thought
- Few-shot Prompting
- Zero-shot Prompting
- Step-by-Step Prompting
- Self-Consistency
- Contextual Priming
- Retrieval-Augmented Generation (RAG)
- Iterative Refinement
- Repetition Reduction
- Role-based Prompting
- Instruction Tuning
- Error Analysis and Correction

Model Reasoning—Chain-of-Thought

Before

What is the annual water demand of a single-family household containing four people who are at home an average of 200 days per year and use an average of 100 liters of water per day?

The annual water demand of a single-family household containing four people who are at home an average of 200 days per year and use an average of 100 liters of water per day is 8,000,000 liters per year.

After

What is the annual water demand of a single-family household containing four people who are at home an average of 200 days per year and use an average of 100 liters of water per day?

Let's think step by step and explain the calculation step by step.

Step 1: Calculate the total number of days in a year:
There are 365 days in a year.

→ **Step 2:** Calculate the total number of days the family is at home:
The family is at home an average of 200 days per year.

→ **Step 3:** Calculate the total water demand of the family:
The family uses an average of 100 liters of water per day. Therefore, the total water demand of the family is $100 \text{ liters} \times 200 \text{ days} = 20,000 \text{ liters per year}$.

→ **Step 4:** Calculate the annual water demand of the single-family household:
The annual water demand of the single-family household is 20,000 liters per year.

Prompt Chaining

Entity Extraction

Please **extract entities** from the following news article: 'The new iPhone model is set to be released next month. It has been highly anticipated by Apple fans and is expected to feature a larger screen and improved camera'

```
{'text': 'iPhone', 'type': 'PRODUCT'}, {'text': 'Apple', 'type': 'ORGANIZATION'}
```

Summarization

Please **summarize** the information about the product

```
[{'text': 'iPhone', 'type': 'PRODUCT'},  
 {'text': 'Apple', 'type': 'ORGANIZATION'}]
```

The iPhone is a line of smartphones designed and marketed by Apple Inc. These devices combine a computer, camera, and internet connectivity in a single device, and are known for their sleek design and user-friendly interface.

Sentiment Analysis

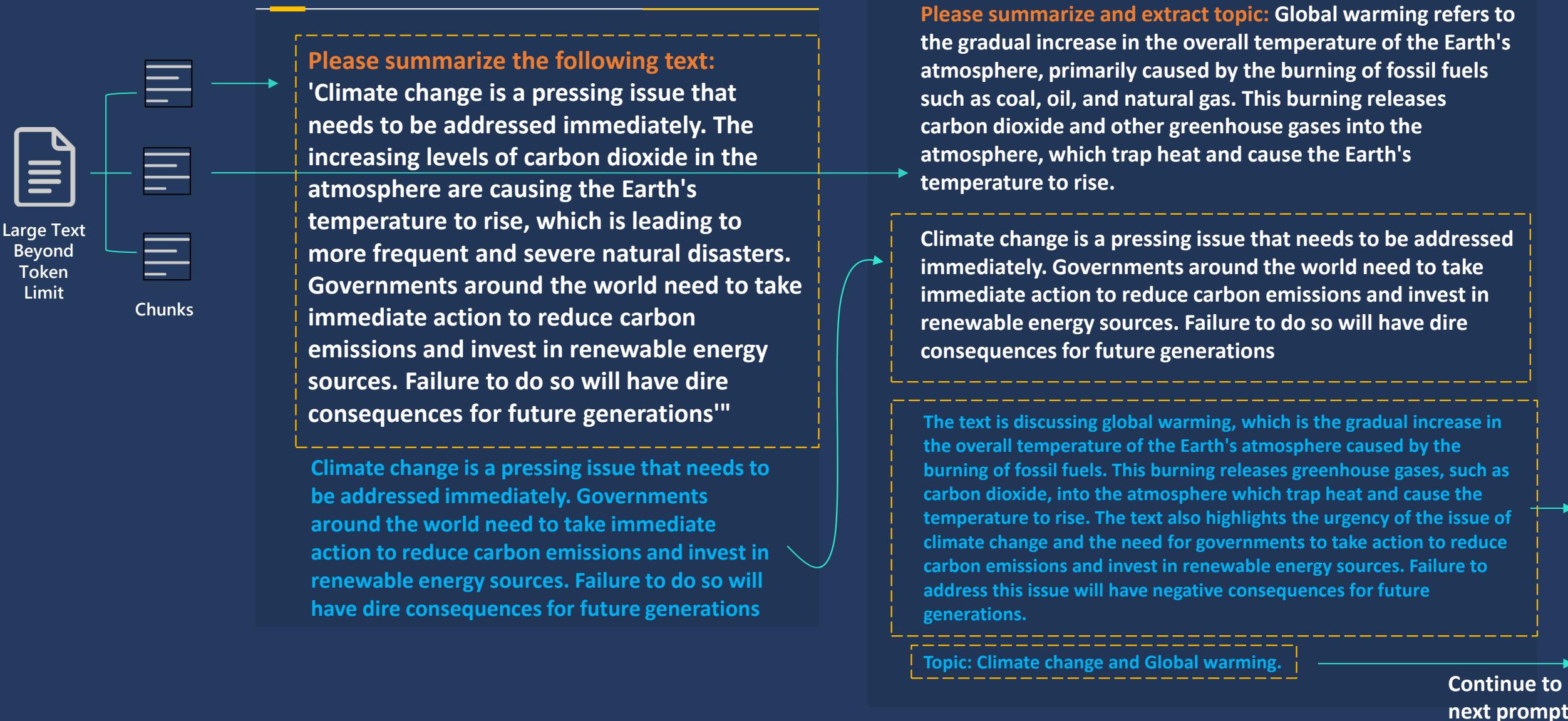
Please provide a **sentiment** for the following text:

The iPhone is a line of smartphones designed and marketed by Apple Inc. These devices combine a computer, camera, and internet connectivity in a single device, and are known for their sleek design and user-friendly interface.

Positive

Prompt Chunking (token limit)

Then all summaries are passed to GPT again to summarize, i.e. second round of summarization. The sum of the summaries has to fit within the token limit.



Few-Shot Reasoning (Human version)

1st prompt

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: The answer is 11.

The cafeteria has 23 apples. If they used 20 to make lunch and bought 6 more, how many do they have?

The answer is 27



2nd prompt—provide reasoning

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 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5+6 = 11$. The answer is 11.

The cafeteria has 23 apples. If they used 20 to make lunch and bought 6 more, how many do they have?

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23-20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.



Repeat Instruction at the End

Models can be susceptible to recency bias, which in this context means that information at the end of the prompt might have more significant influence over the output than information at the beginning of the prompt.

Therefore, it is worth experimenting with repeating the instructions at the end of the prompt and evaluating the impact on the generated response.

Prime the Output

“Priming the output” means including a few words or phrases at the end of the prompt to obtain a model response that follows the desired form.

For example, using a cue such as “Here’s a bulleted list of key points.” can help make sure that the output is formatted as a list of bullet points.

System Message	User Message	Assistant
You are an AI assistant that helps people find information. Answer in as few words as possible.	John Smith is married to Lucy Smith. They have five kids, and he works as a software engineer at Microsoft. What search queries should I do to fact check this? ## One possible search query is:	“John Smith married Lucy Smith five kids software engineer Microsoft”

In the above example, the text “**One possible search query is:**” primes the model to produce a single output. Without this cue the model would likely produce several search queries as an output.

Add Clear Syntax

Using clear syntax for your prompt – including punctuation, headings, and section markers helps communicate intent and often makes outputs easier to parse.

In the example shown here, **separators** (---) have been added between different sources of information. This allows the use of “---” as a stopping condition for generation. In addition section headings or special variables are presented in uppercase to differentiate them.

System Message	User Message	Assistant
You are an AI assistant that helps people find information.	You will read a paragraph, and then issue queries to a search engine in order to fact-check it. Also explain the queries. --- PARAGRAPH John Smith is married to Lucy Smith. They have five kids, and he works as a software engineer at Microsoft. What search queries should I do to fact-check this? --- QUERIES	1. “John Smith Microsoft” – to verify employment 2. “John Smith Lucy Smith” – to verify marital connection 3. “John Smith children” – to check that John has five children.
Answer in as few words as possible.		

Challenge #1

Azure OpenAI Fundamentals

- Deploy an Azure OpenAI Model
- Prompt Engineering
- What's possible through prompt engineering
- Best practices when using OpenAI text and chat models

Semantic Kernel

SDK Fundamentals

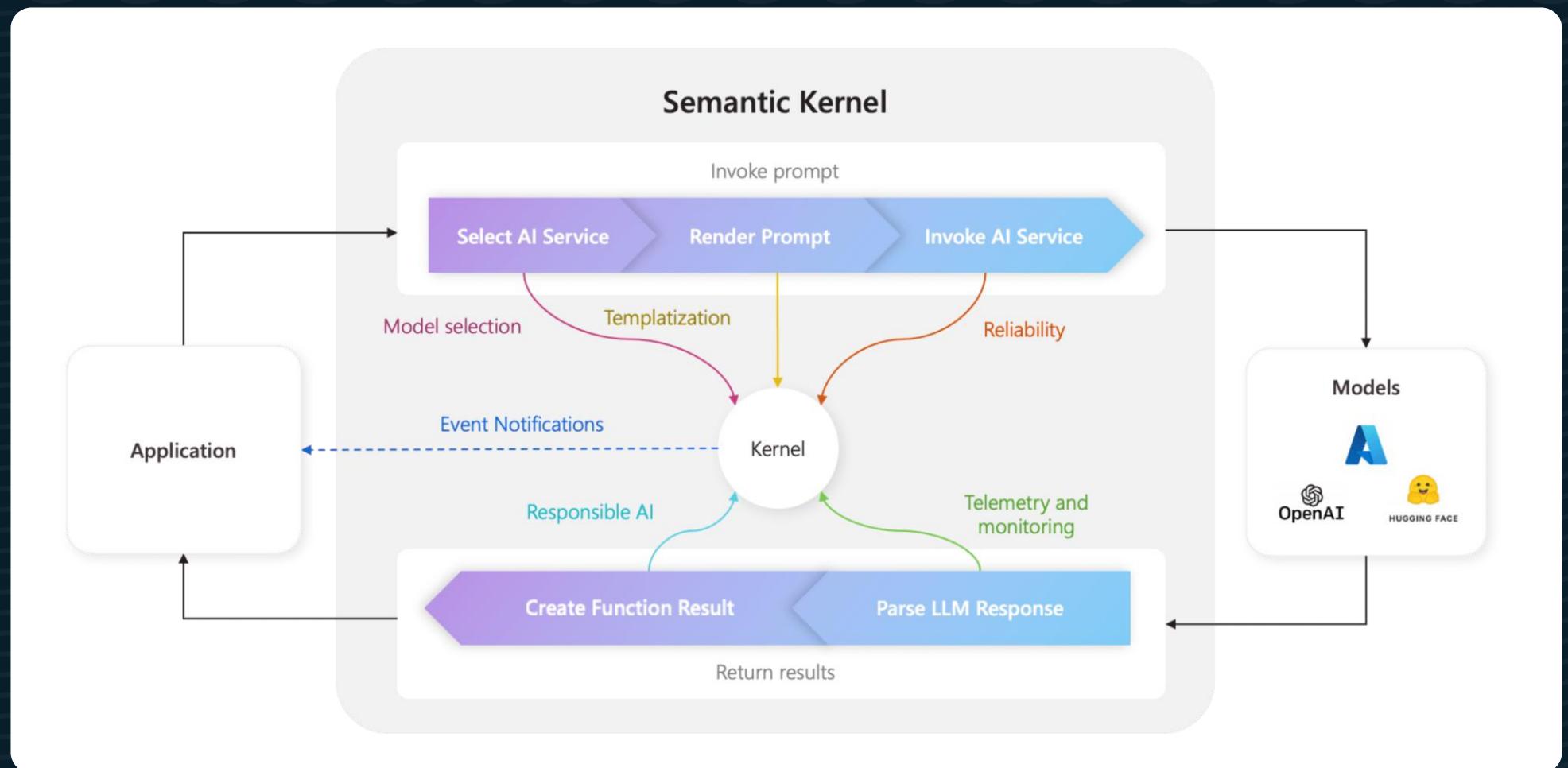


What is Semantic Kernel?

Semantic Kernel is a lightweight open-source SDK
that lets you orchestrate native code with LLMs.

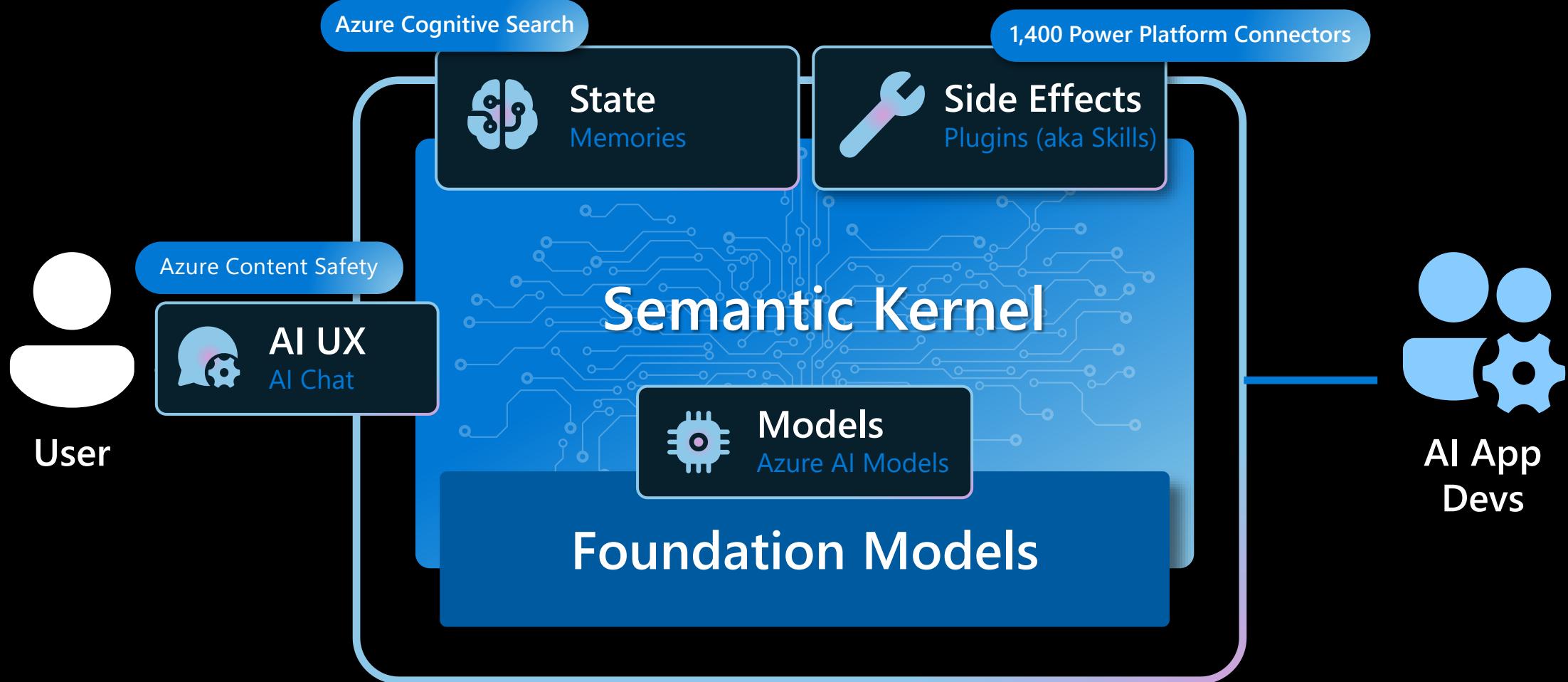
Semantic Kernel

The kernel is at the center of it all

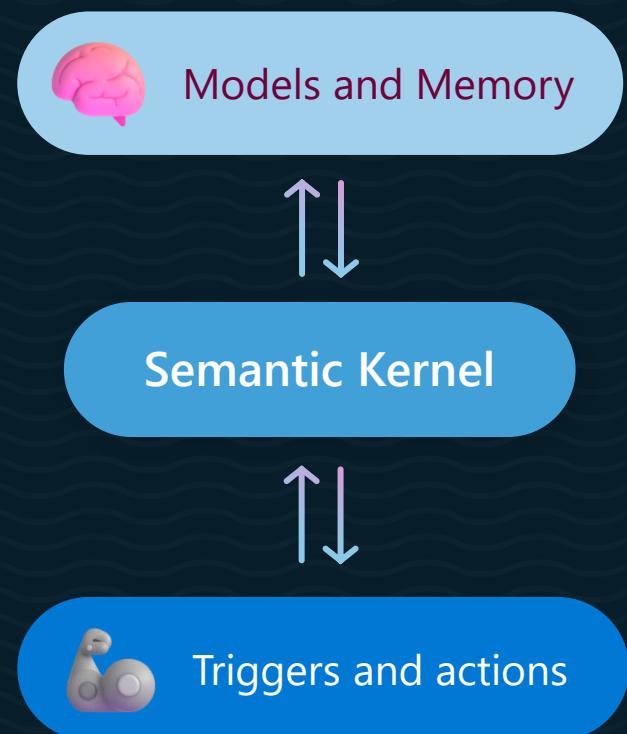


Semantic Kernel

Integrate AI into your existing apps using C#, Python & Java



Semantic Kernel is extensible



Semantic Kernel is available in...

C#

Python

Java

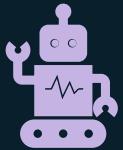
Semantic Kernel's Chat History

Author Roles



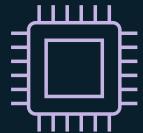
User

The end-user interacting with the AI assistant.
Provides input, questions, or commands.



Assistant

The AI assistant responding to the user.
Generates replies or actions based on user input.



System

The underlying system instructions.
Sets initial context, behavior guidelines, or constraints for the assistant.
Influences assistant's behavior throughout the interaction.



Tool

External functions or services invoked by the assistant.
Provides additional information or performs actions to aid responses.
Enhances assistant's capabilities with external data or functions.

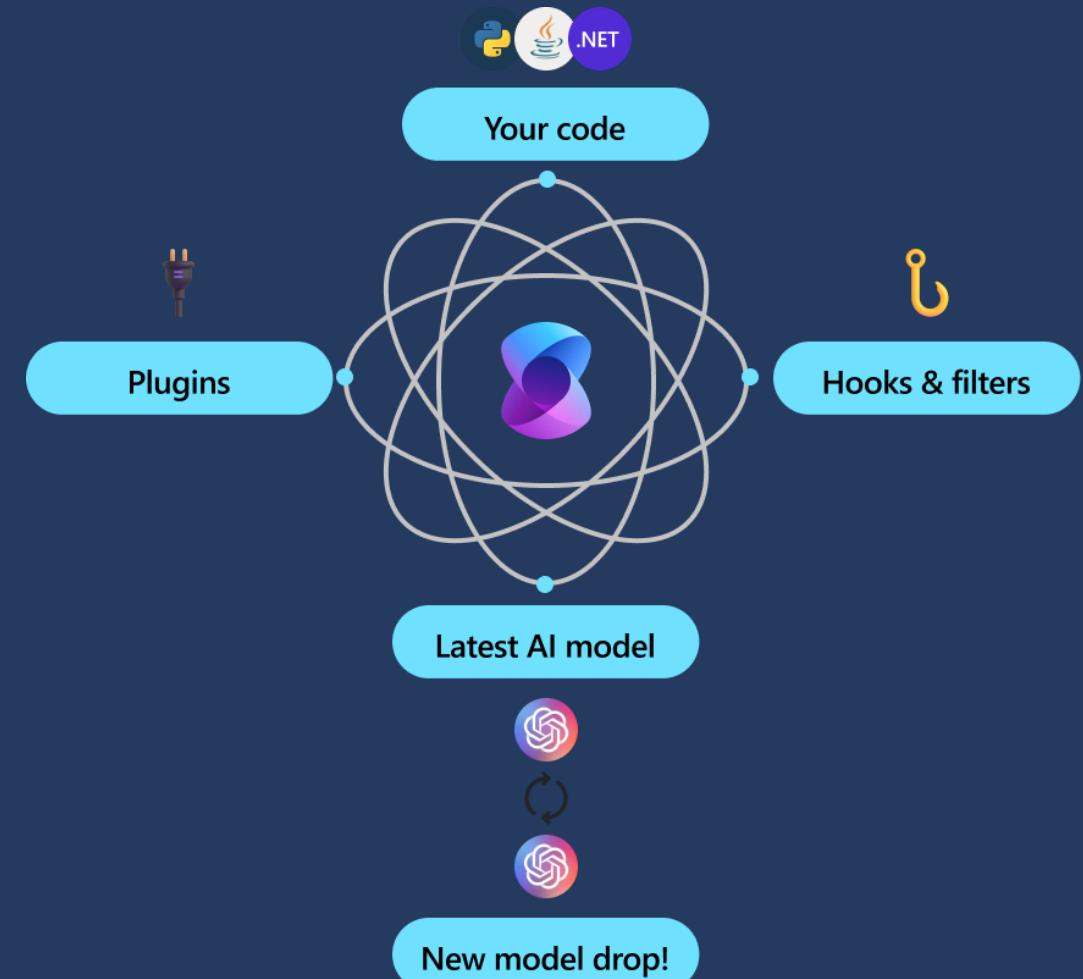
Challenge #2

Semantic Kernel Fundamentals

- Semantic Kernel Fundamentals
- Connect your OpenAI model using Semantic Kernel
- Test Your Application

Plugins

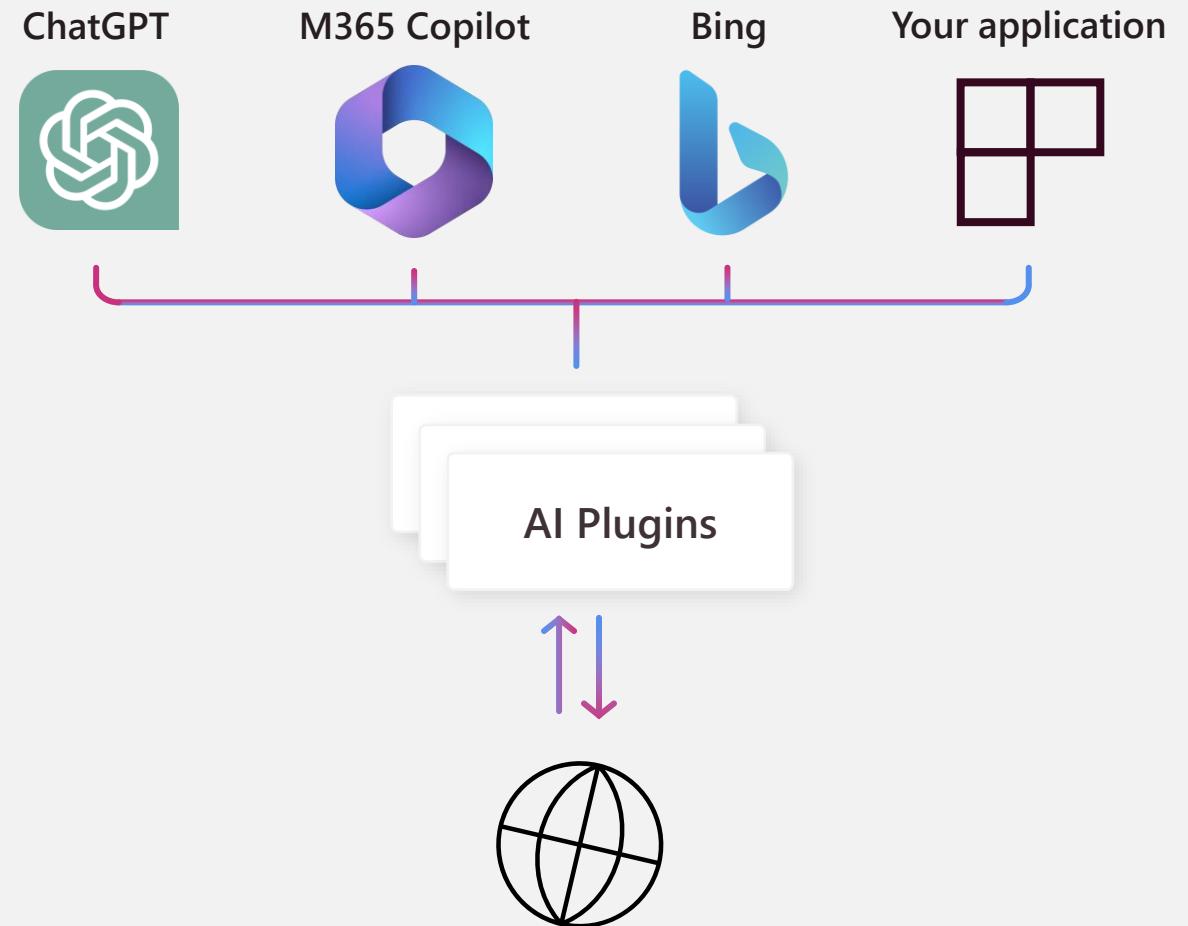
Semantic Kernel
SDK Fundamentals

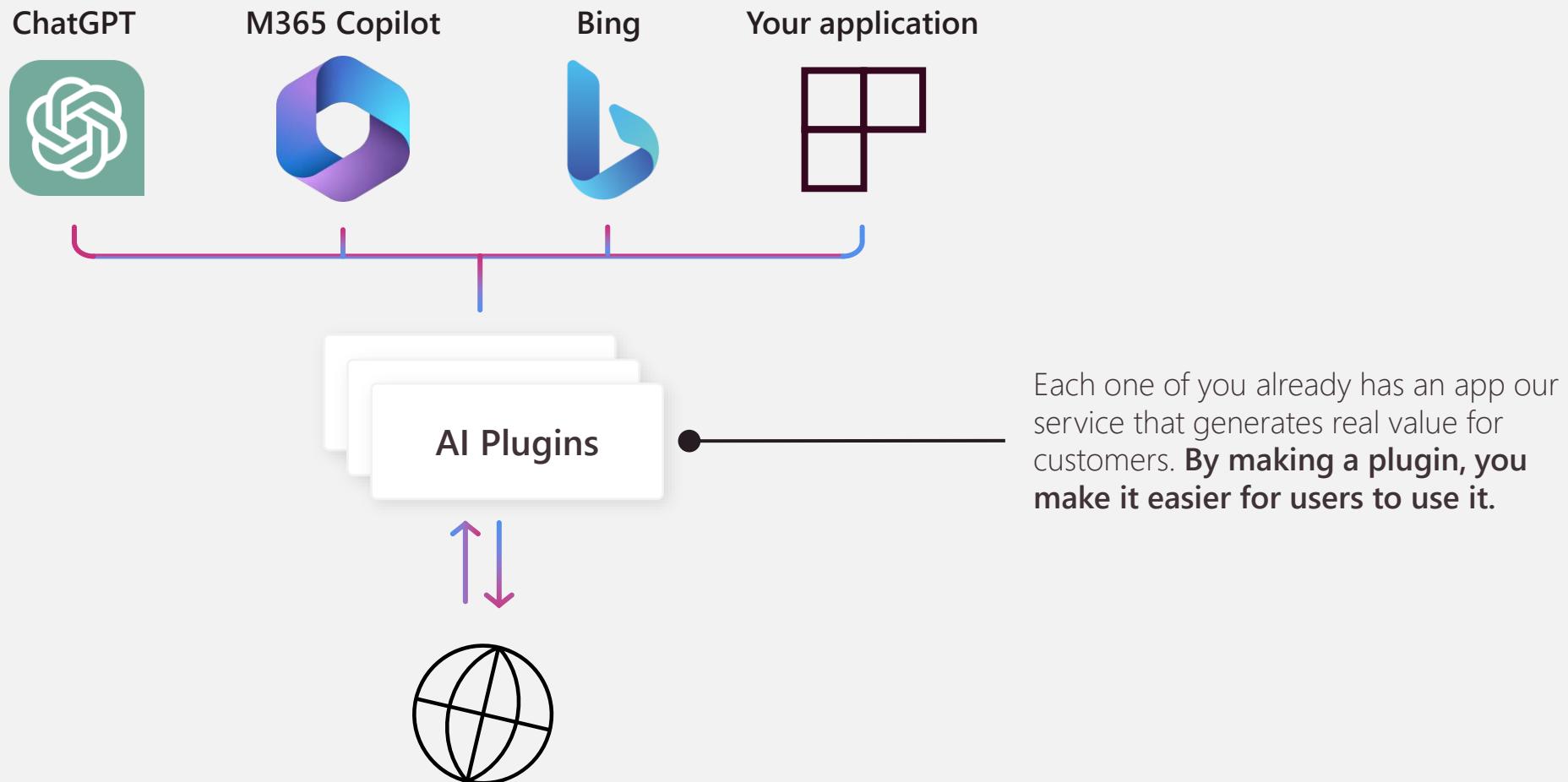


What *is* a plugin?

Chatbots are *nice*, but they aren't *useful* to your users until they can interact with the real world by...

- 1 Retrieving data
- 2 Sending emails
- 3 Completing sales
- 4 Making orders
- 5 And more!

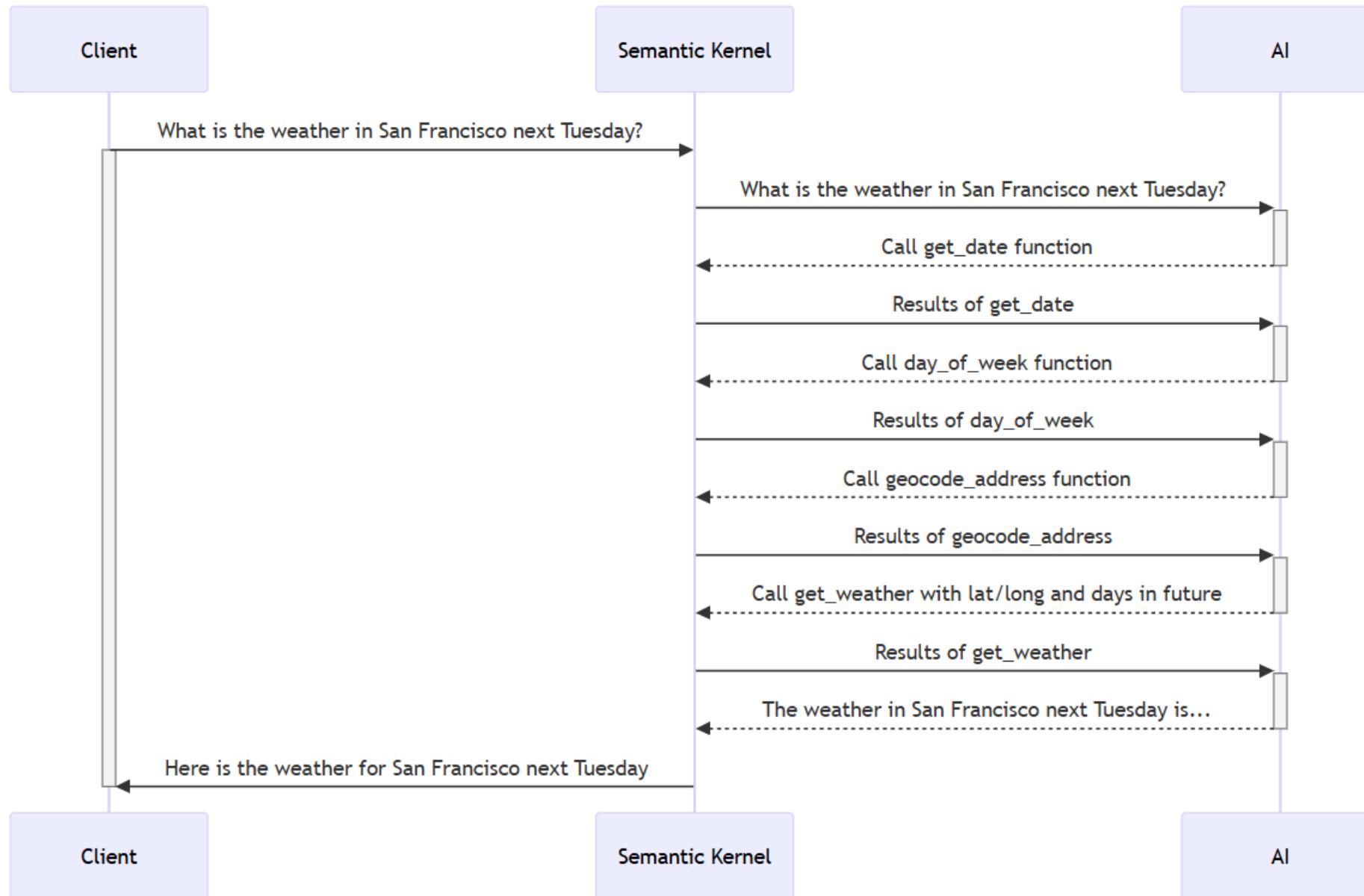




Defining a plugin using a class

```
C#  
  
public class LightsPlugin  
{  
    private readonly List<LightModel> _lights;  
  
    public LightsPlugin(LoggerFactory loggerFactory, List<LightModel> lights)  
    {  
        _lights = lights;  
    }  
  
    [KernelFunction("get_lights")]  
    [Description("Gets a list of lights and their current state")]  
    [return: Description("An array of lights")]  
    public async Task<List<LightModel>> GetLightsAsync()  
    {  
        return _lights;  
    }  
  
    [KernelFunction("change_state")]  
    public void ChangeState(LightModel light, LightState state)  
    {  
        _lights[_lights.IndexOf(light)].State = state;  
    }  
}
```

Plugin Workflow



Challenge #3

Plugins

- Functions and Plugins Fundamentals
- Creating Semantic Kernel Plugins
- Enable auto function calling
- What is a Planner

Demo

Aspire Plugin Flow

Challenge #4

Logic Apps

- Setup Azure DevOps
- Create a Logic App
- Import Logic App into Semantic Kernel

Break into groups

- Group Size 3 – 5
- Get a group number from an Instructor
 - This will let you know what code to redeem.
- Go to this link: <https://bit.ly/4hvdn7j>
 - Code is ACTIVATE45603XX
 - The XX will be your team number assigned by an Instructor

Retrieval-Augmented Generation

Give knowledge to the AI
using your data



What is RAG



OpenAI LLMs are trained on a pre-defined dataset



Retrieval Augmented Generation (RAG) is how to provide additional knowledge to the LLMs



This enhances AI responses to user inputs

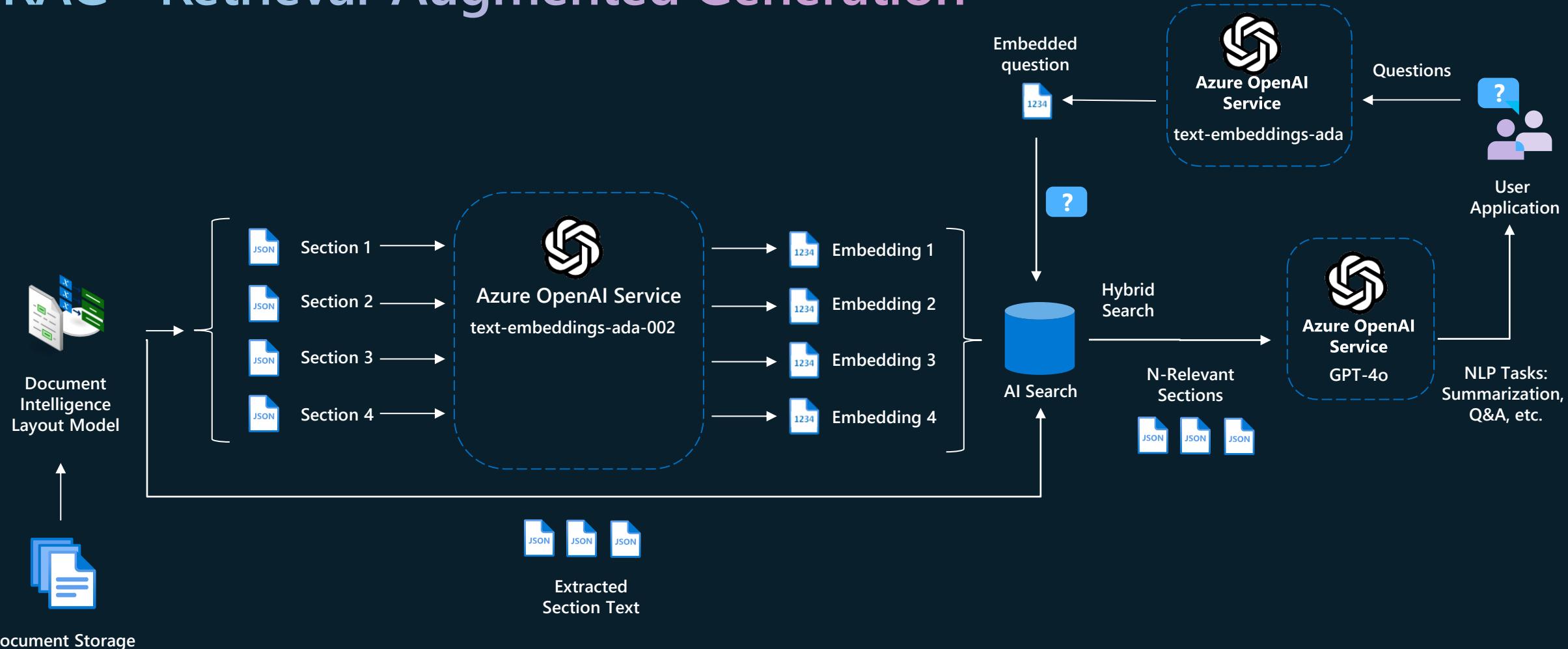
When to Use RAG

RAG fills the gaps in AI's knowledge.

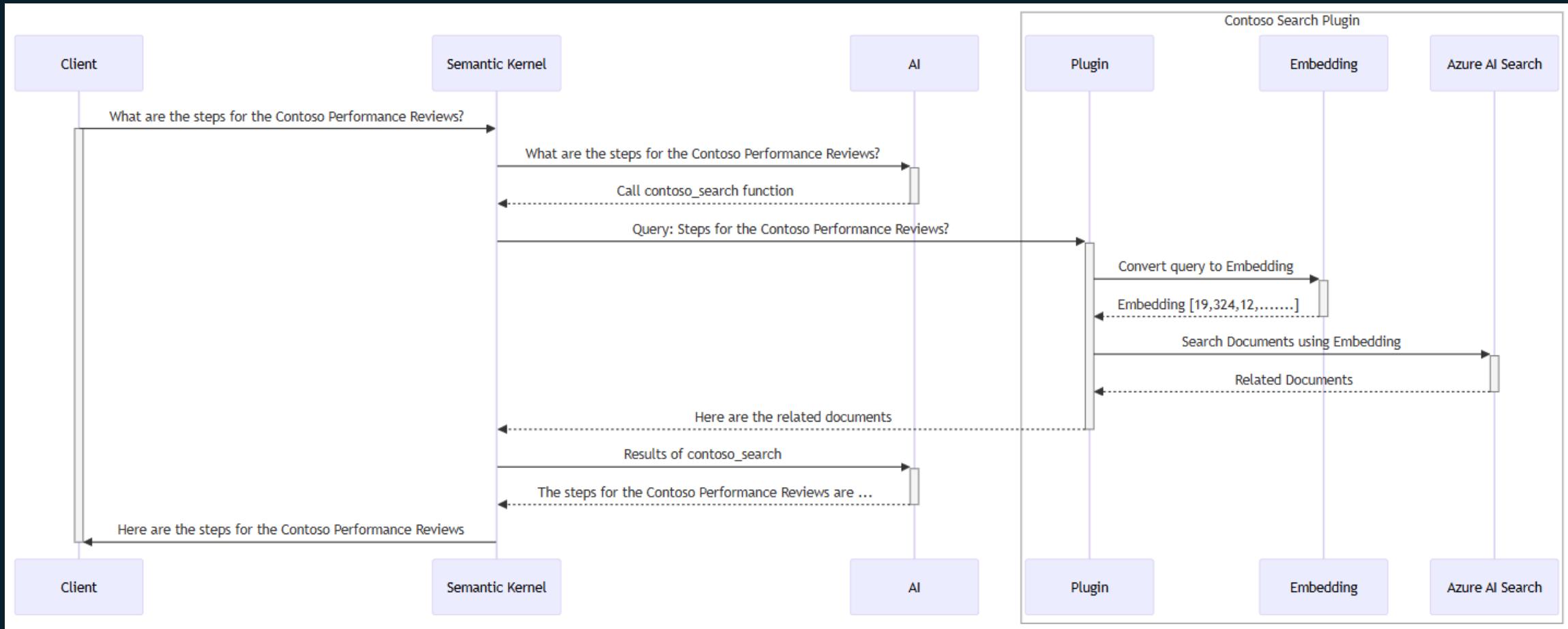
Use this any time we expect the user to ask domain knowledge questions or time-sensitive information



RAG - Retrieval-Augmented Generation



RAG – Plugin



Challenge #5

Retrieval-Augmented Generation (RAG)

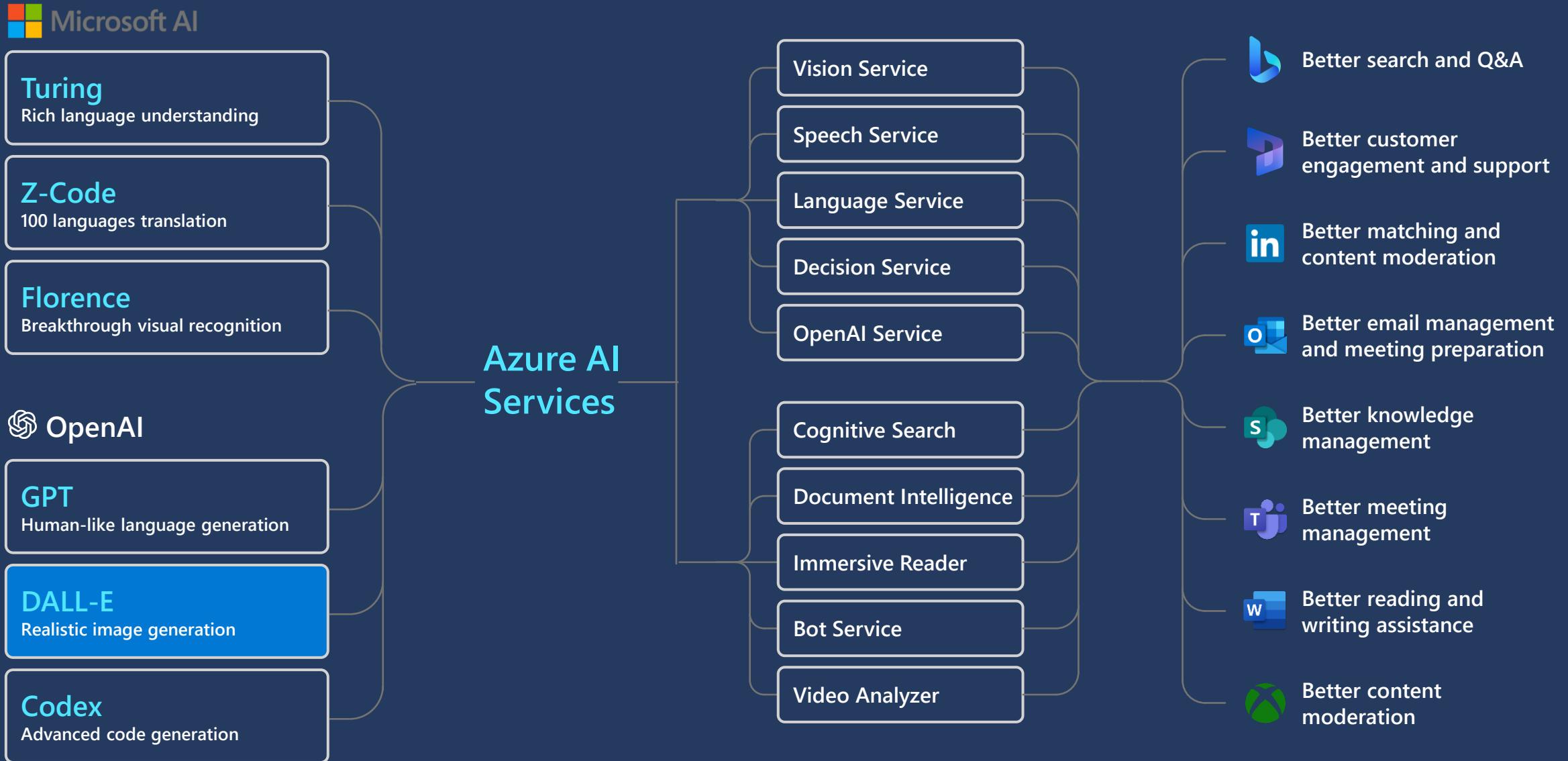
- Document Chunking & Embedding
- Enhance AI responses by searching external sources

Image Generation

DALL-E Plugin



Large models at scale



DALL-E 3

Azure OpenAI Service

DALL-E 3 is an image generation model that allows you to generate images from text prompts



OpenAI

Dall-E 3

An astronaut riding a horse in a photorealistic style



Teddy bear working on new AI research on the moon in 1980



A bowl of soup that looks like a monster knitted out of wool



Use Cases for DALL·E 3



LOGO & BRANDING:
QUICK CONCEPT
GENERATION.



**CREATIVE
INSPIRATION:**
OVERCOME DESIGN
BLOCKS.



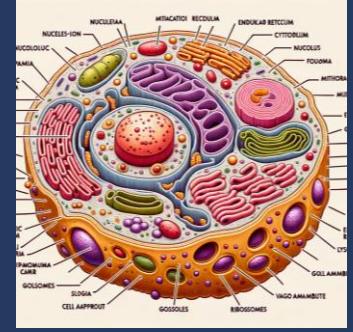
**CONTENT
ILLUSTRATIONS:**
UNIQUE IMAGES FOR
BLOGS/ARTICLES.



AD CAMPAIGNS:
VISUALIZE MARKETING
CONCEPTS.



**PRODUCT
VISUALIZATION:** GAUGE
INTEREST & FEEDBACK.



EDUCATION: CUSTOM
IMAGERY FOR COURSES.



FASHION DESIGN:
VISUALIZE CLOTHING
PATTERNS.



GAMING: CHARACTER &
ENVIRONMENT
CONCEPTS.

Image Plugin

Tips and Tricks

You can add a `[Description()]` Attribute to a parameter
This can help the LLM generate a better prompt for images

```
[KernelFunction("generate_image_from_text")]
[Description("returns an image url from a text description")]
public async Task<string> GetImageURLAsync([Description("Descriptive prompt
optimized for DALL-E")] string imageDescription)
```



Challenge #7

Image Generation using DALL-E

- Working with Text to Image Models
- Creating an Image Generating Plugin

Multi-Agent Systems

Multi-agent conversation using Semantic Kernel



Understanding Multi-Agent LLM Solutions

- Multi-agent systems use multiple AI agents to work together and solve complex tasks.
- Each agent in the system has specialized roles and responsibilities.
- Agents communicate and coordinate to achieve a common goal.
- Applications include automated customer service, collaborative problem-solving, and more.
- Ensuring seamless interaction between agents improves overall system efficiency.

Example of a Multi-Agent System

Demo Challenge 8!

Challenge #8

Multi-Agent Systems

- Create a multi-agent conversation using Semantic Kernel
- Implement a multi-agent conversation using Azure OpenAI

We Value Your Feedback!



<https://forms.office.com/r/4hLJfZq3LA>



AI

DEVELOPER WORKSHOP SEMANTIC KERNEL . SEMANTIC KERNEL

No More Slides!

