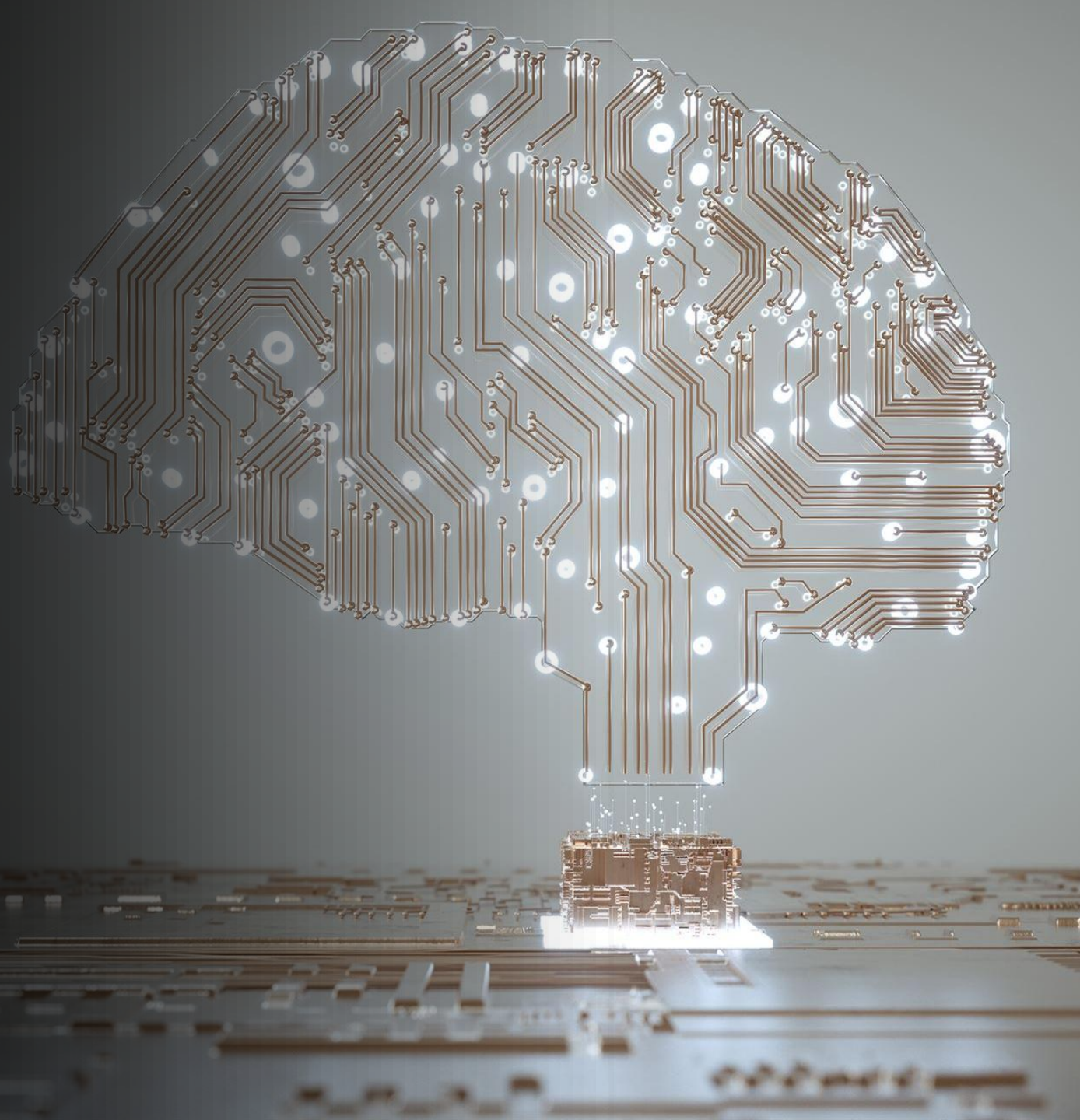
Welcome to the AI Workshop for Developers



Hosted by Microsoft and Kizan

Microsoft

KiZAN


Ride the wave

Connect with Us!

Najib Zarrari

Director, Cloud Solutions Architecture

Eric Rhoads

Practice Lead, Cloud Infra and Apps at KiZAN Technologies

Who We Are

KiZAN is a Microsoft AI Cloud Partner with over 30 years of experience helping organizations achieve their IT business goals. Our singular focus is to offer unrivaled expertise and innovation with exclusively Microsoft technologies.

We have 11 Microsoft specializations and provide best-in-class architecture, design, and deployment to help you unlock the full potential of your applications, data, collaboration, and cloud platforms.

From strategic road mapping and licensing consultation to application development and 24/7 helpdesk support, our commitment to your success extends beyond the initial service. As your trusted technology partner, KiZAN will provide the service and expertise your organization needs as it transforms and expands.

Meet Our Instructors

- Microsoft:
  - Najib Zarrari, Director of Cloud Solution Architect (CSA)
  - Randy Patterson, Sr. CSA
  - Chris McKee, Sr. CSA
  - Eric Evans, Sr. CSA
  - Walid Amro, Sr. CSA
- KiZAN:
  - Eric Rhoads, Practice Lead
  - Zack Way, Principal Consultant
  - Travis Terrell, Senior Azure Consultant

Agenda

- **9:00 am – 10:00 am: Introductions**
  - Introduce Coaches
  - Foundational Slides
  - Break into Teams
- **10:00 am – 12:00 pm: Challenges**
  - Challenge 01: Azure OpenAI Fundamentals
  - Challenge 02: Semantic Kernel Fundamentals
  - Challenge 03: Plugins
- **12:00 pm – 1:00 pm: Lunch, Networking**
- **1:00 pm – 4:30 pm: Challenges**
  - Challenge 04: Plugin with Logic App
  - Challenge 05: RAG
  - Challenge 06: Responsible AI
- **4:30 pm – 5:00 pm: Recap & Survey**
  - Reflect on AI use cases
  - Customer Survey request
  - Next Steps

.NET Aspire

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

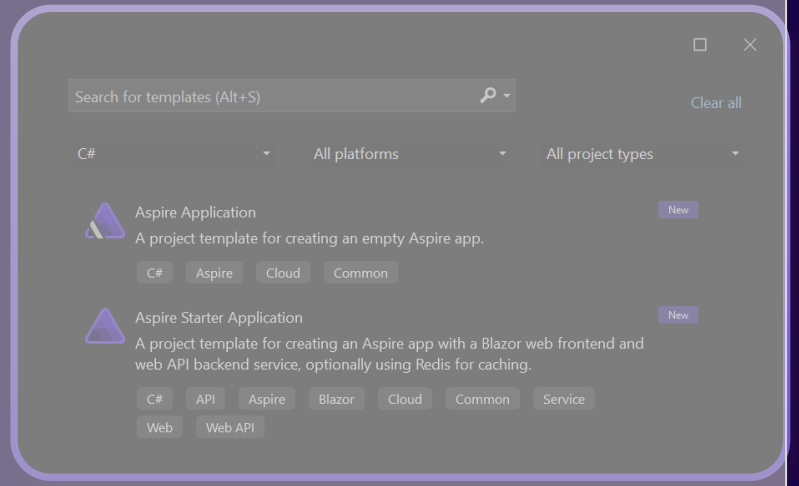
- Smart Defaults
- Developer Dashboard
- Orchestration
- Service Discovery
- Components
- Deployment

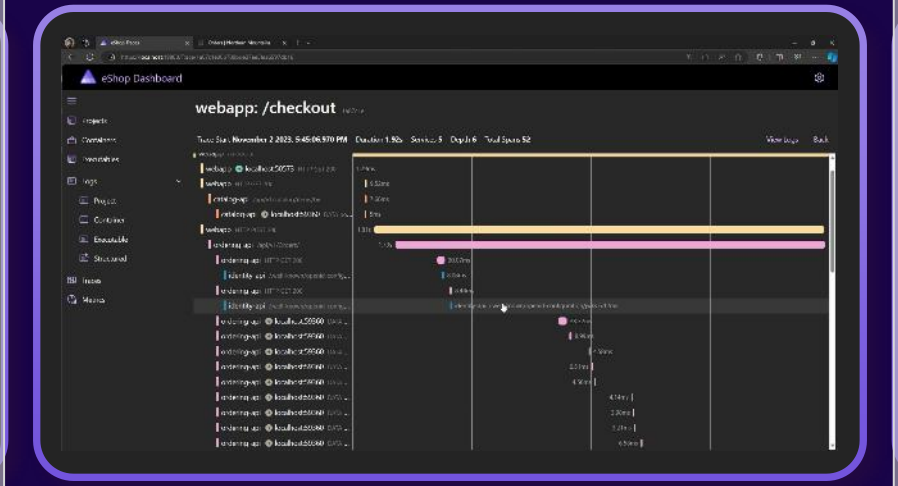
Your average cloud dev deals with:

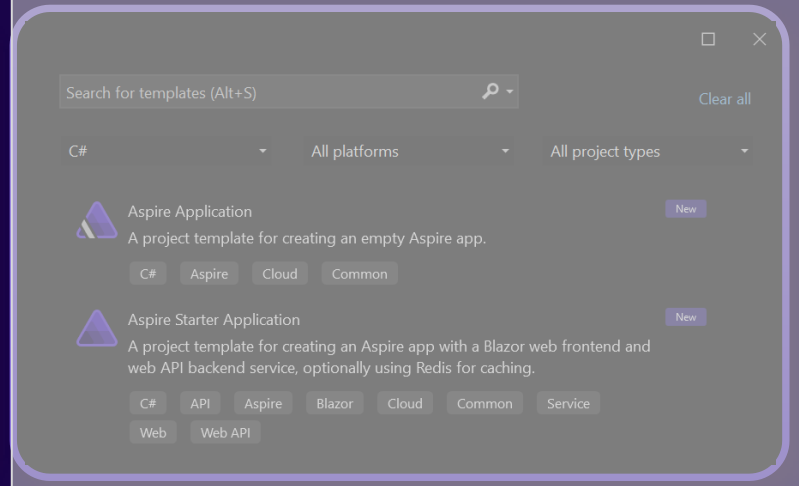
- Caching
- Databases
- Local development tools
- Messaging + queueing
- Containers?!
- Multiple languages
- A zillion deployment targets
- Resiliency + scalability
- Security + networking
- CI/CD
- Storage but... not databases?!
- Telemetry and logs and traces
- Authentication
- AI?!?!?!

and so much more!

.NET Aspire







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!

Challenge #0

Prerequisites

- Prepare your workstation to work with Azure.
- Setup your development environment.

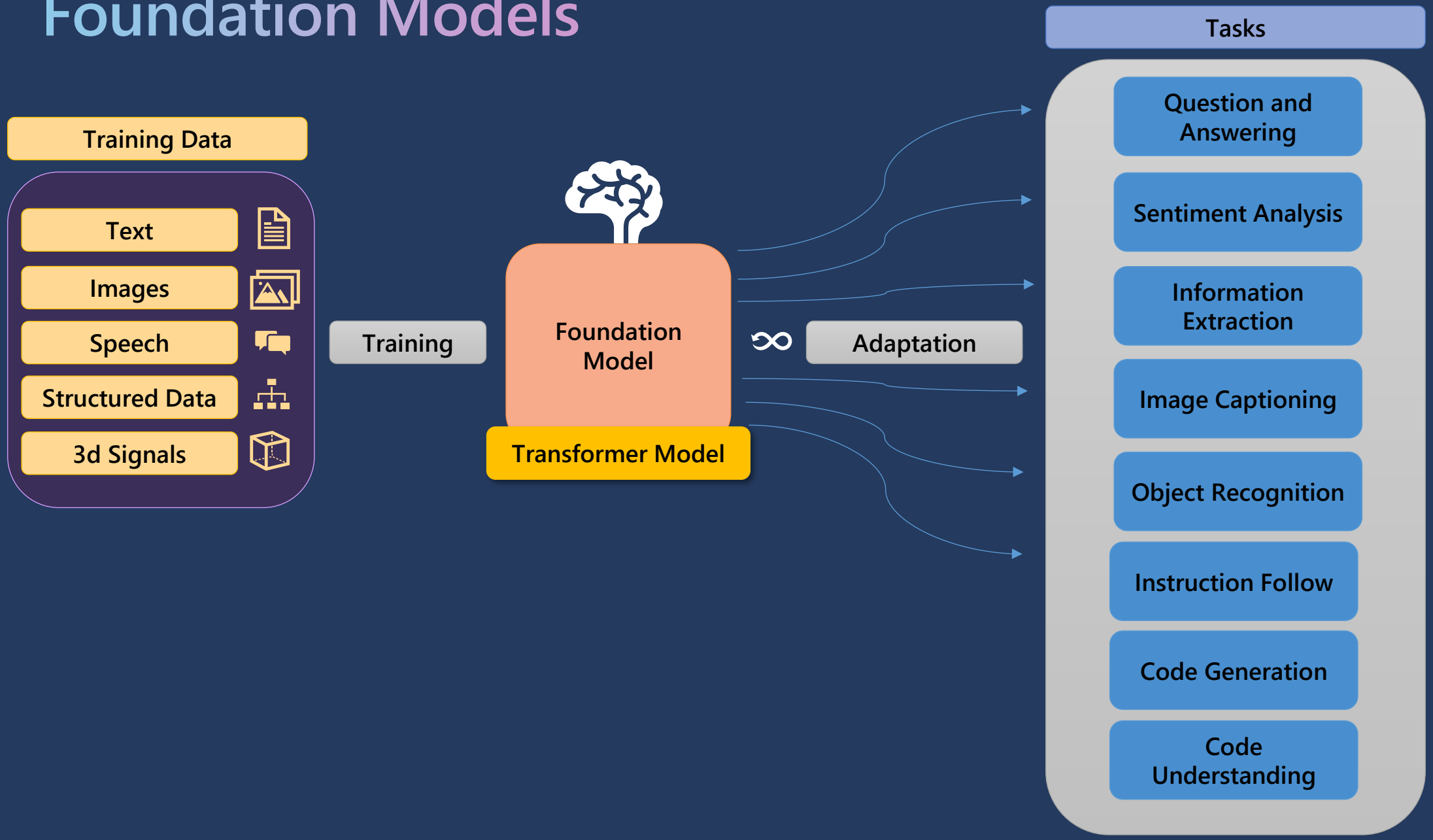
Feel free to start the install in the background.
Visit aka.ms/SKDev

OpenAI Fundamentals

Azure OpenAI Fundamentals

Understanding and utilizing GPT and other generative AI models

Foundation Models



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

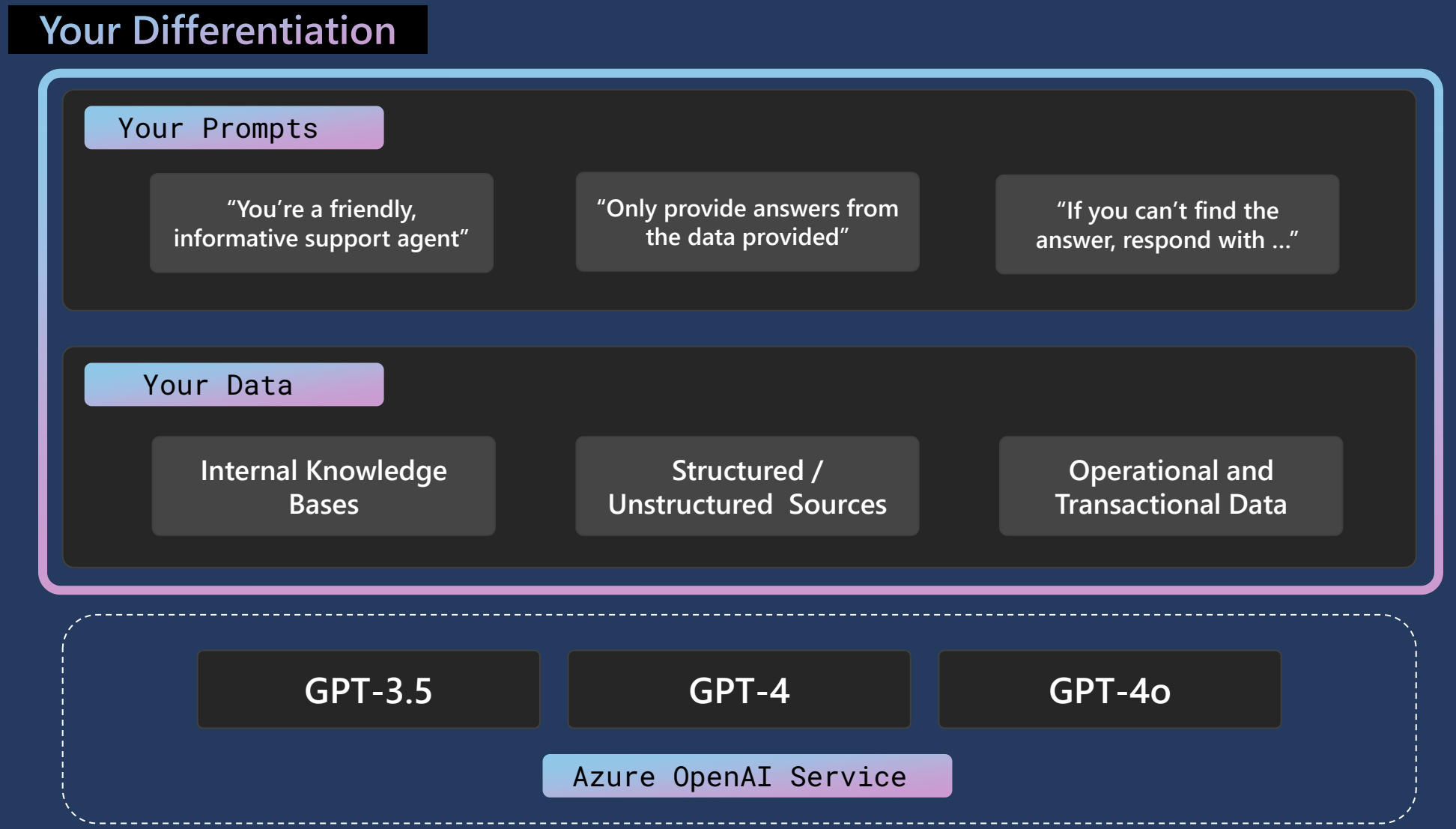
GPT-4

- 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



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:

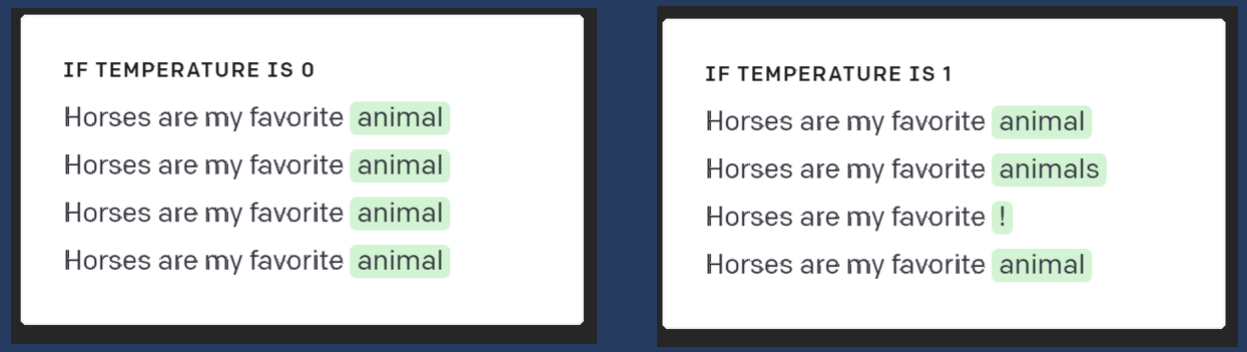
I have an orange cat named Butterscotch.

I have an orange cat named Butterscotch.

Horses are my favorite

animal	49.65%
animals	42.58%
\n	3.49%
!	0.91%

Probabilities:



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)

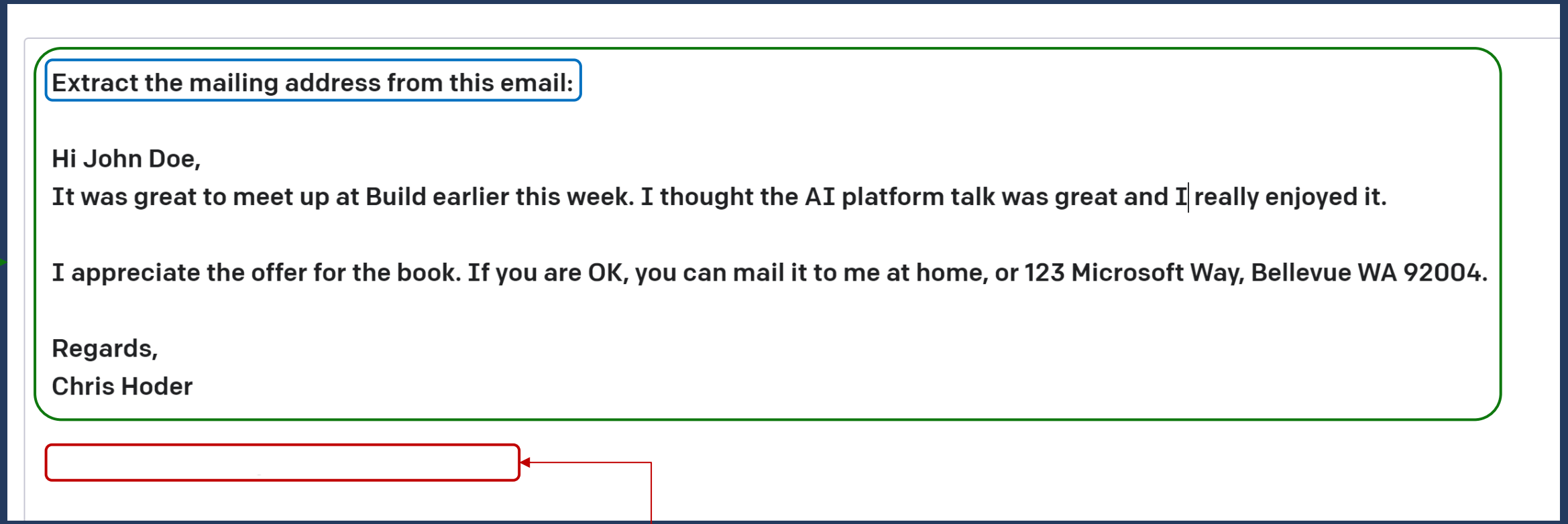
Blaze the Miracle Mare, Pegasus the Winged Warrior, Secretariat the Superhorse

Completion Temperature 1 (often different)

Blaze of Glory, Sterling Silver, Thunderbolt

Prompt Engineering

Azure OpenAI Service | GPT-3 Prompt Design



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.

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 liters x 200 days = 20,000 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.

Other Prompt Engineering Styles

- 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

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 a 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.

Semantic Kernel

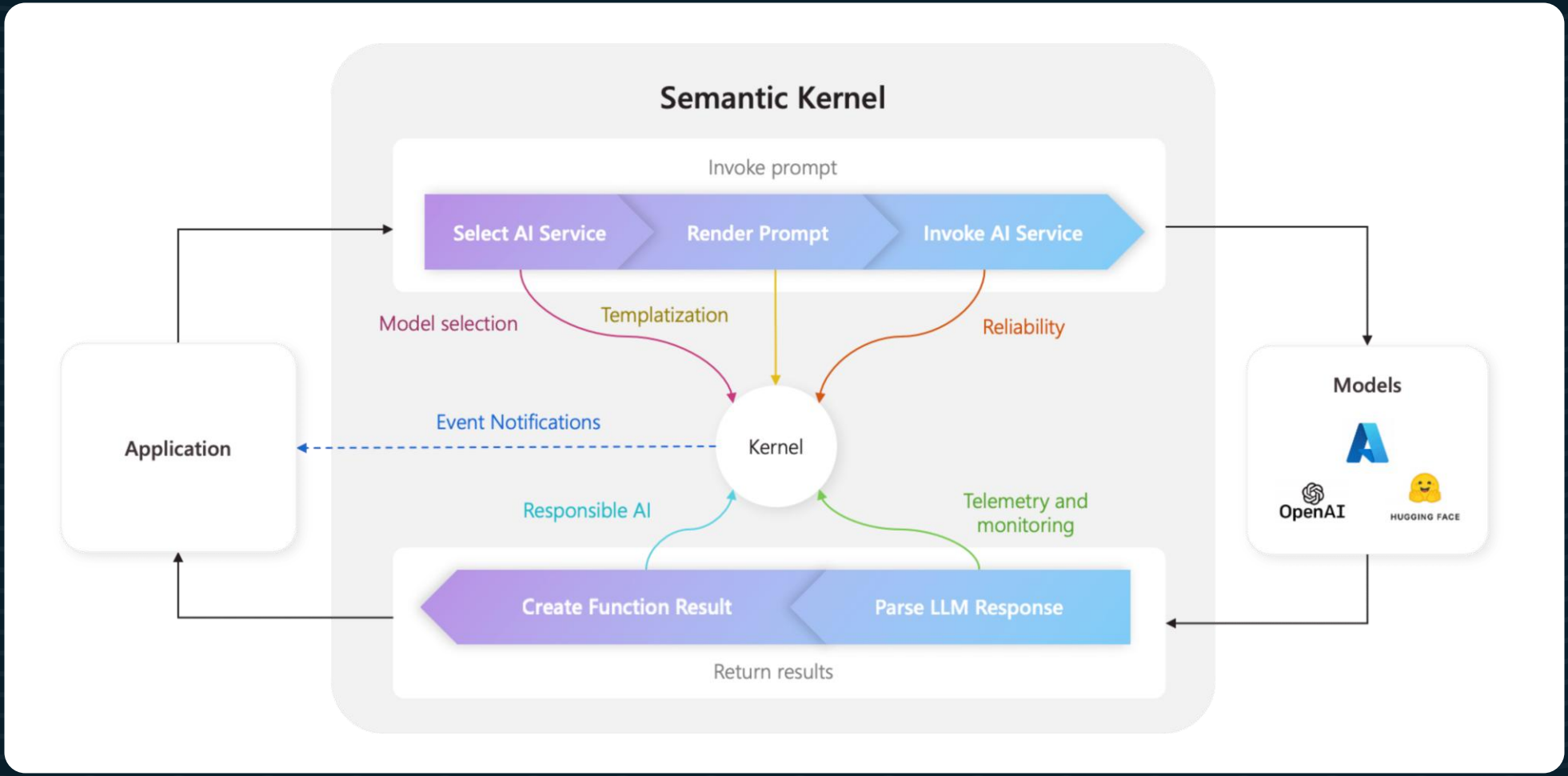
SDK Fundamentals

What is Semantic Kernel?

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

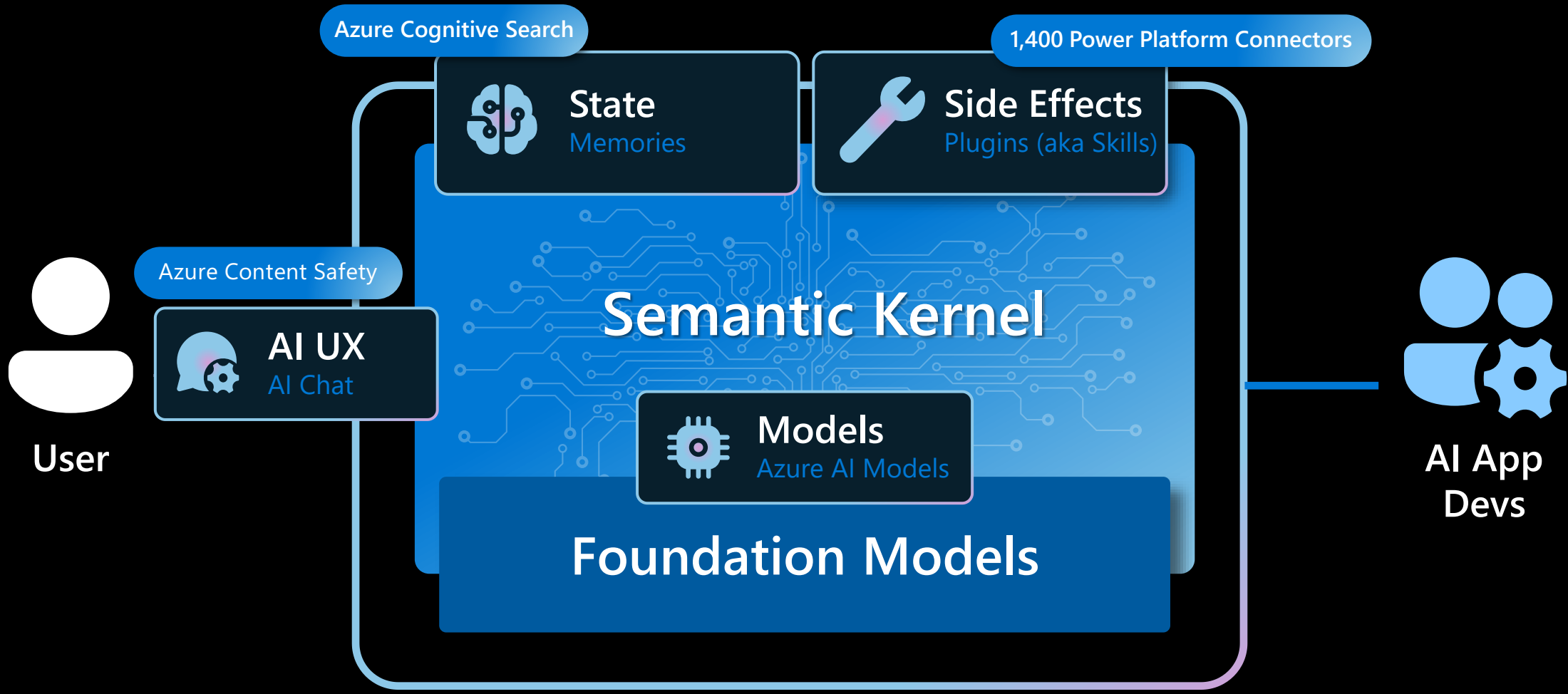
Semantic Kernel

The kernel is at the center of it all

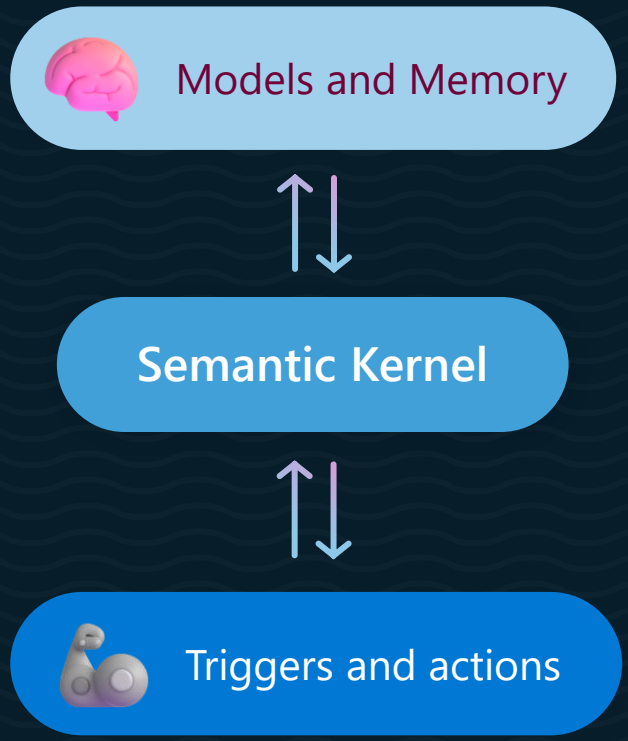


Semantic Kernel

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



Semantic Kernel is extensible



Semantic Kernel is available in...

C#

Python

Java

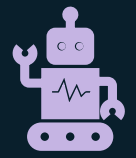
Semantic Kernel's Chat History

Author Roles

User

The end-user interacting with the AI assistant.

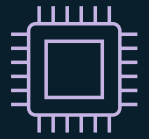
Provides input, questions, or commands.



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

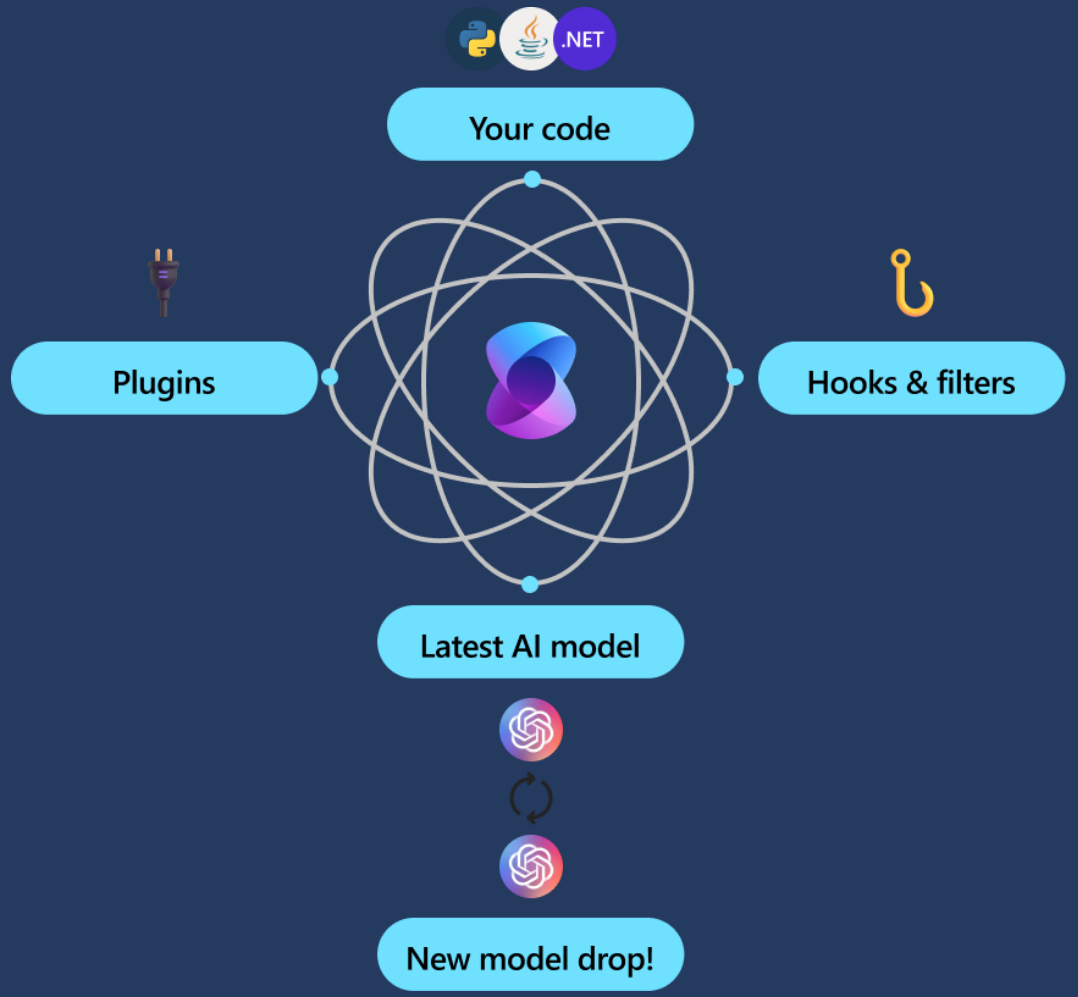
External functions or services invoked by the assistant.

Provides additional information or performs actions to aid responses.

Enhances assistant's capabilities with external data or functions.

Plugins

Semantic Kernel SDK Fundamentals

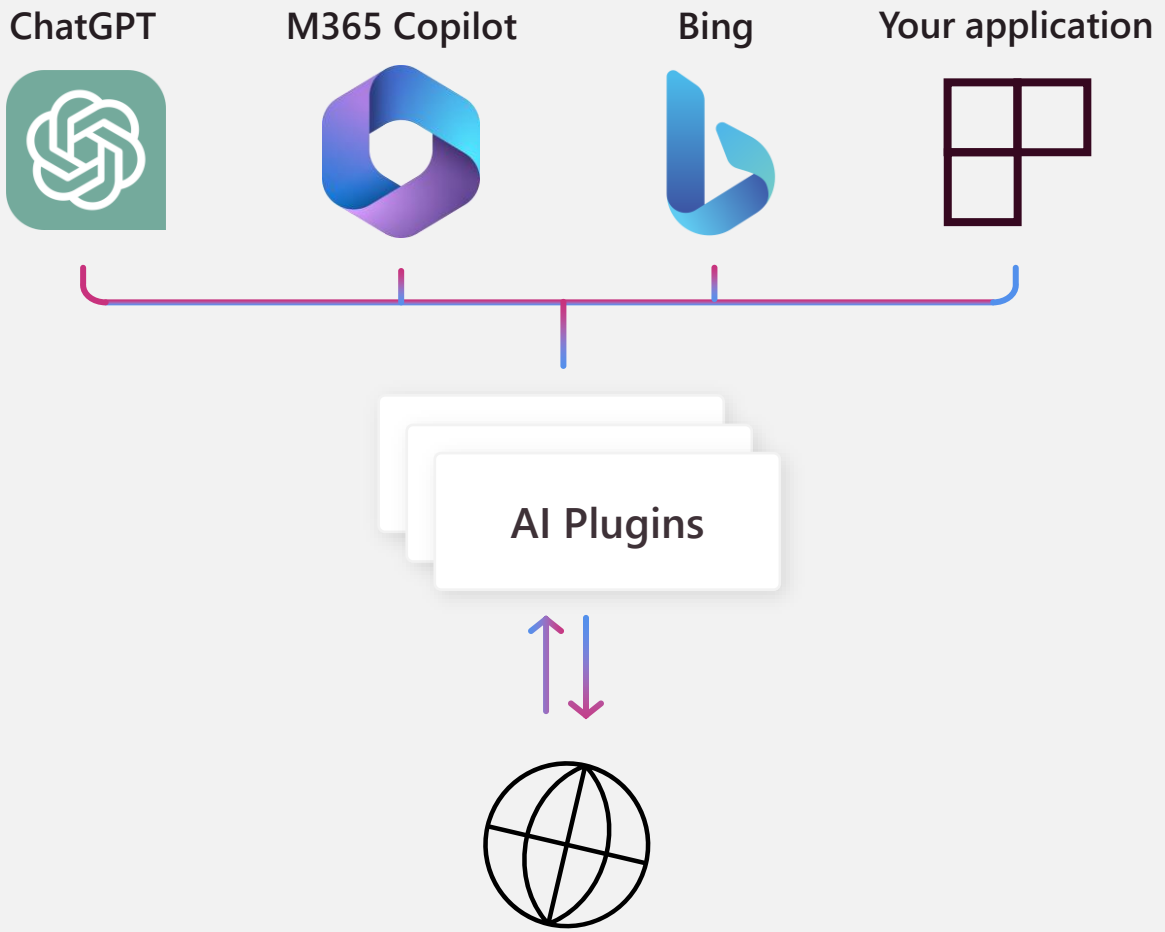


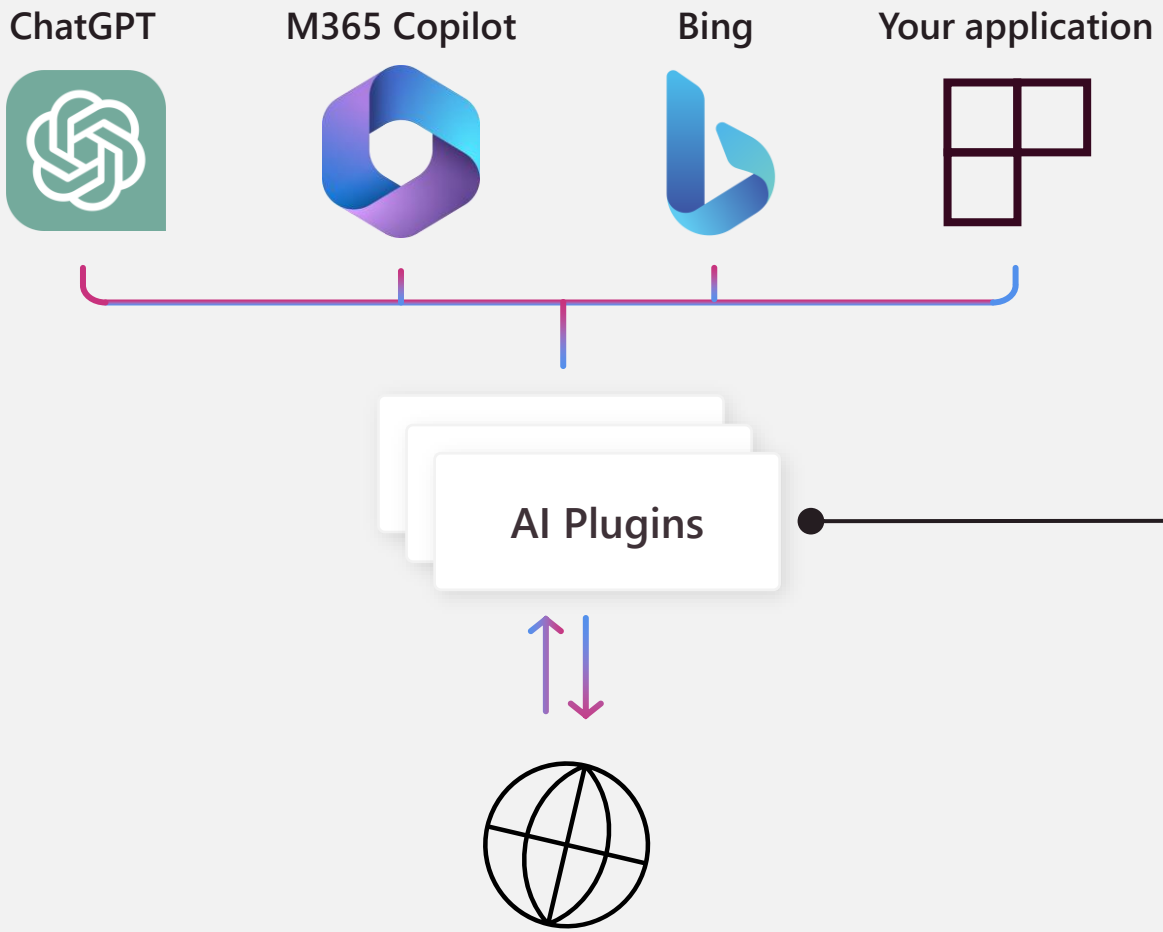
Connect AI to the real world

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!





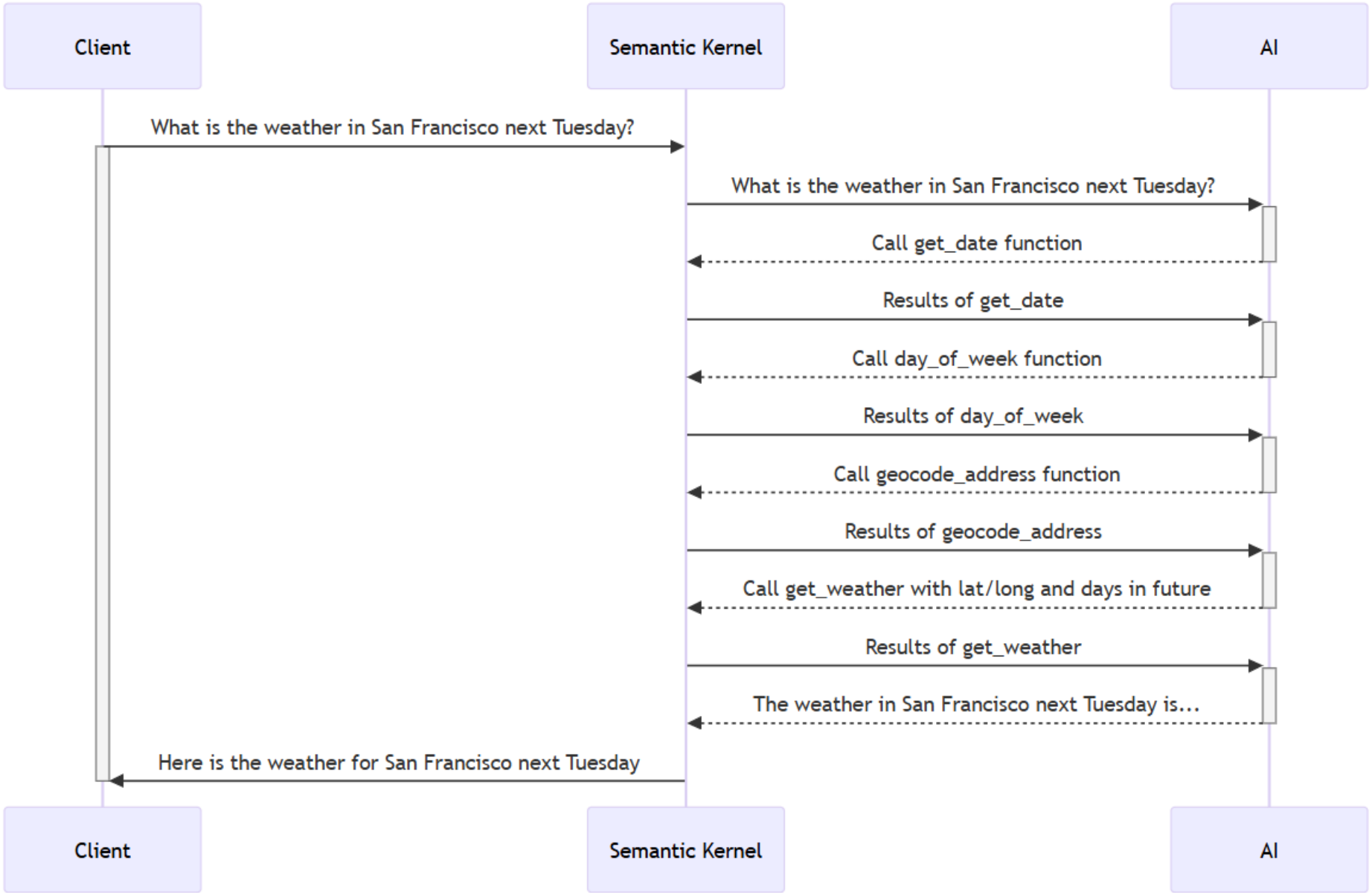
Each one of you already has an app our service that generates real value for customers. **By making a plugin, you make it easier for users to use it.**

Defining a plugin using a class

C#

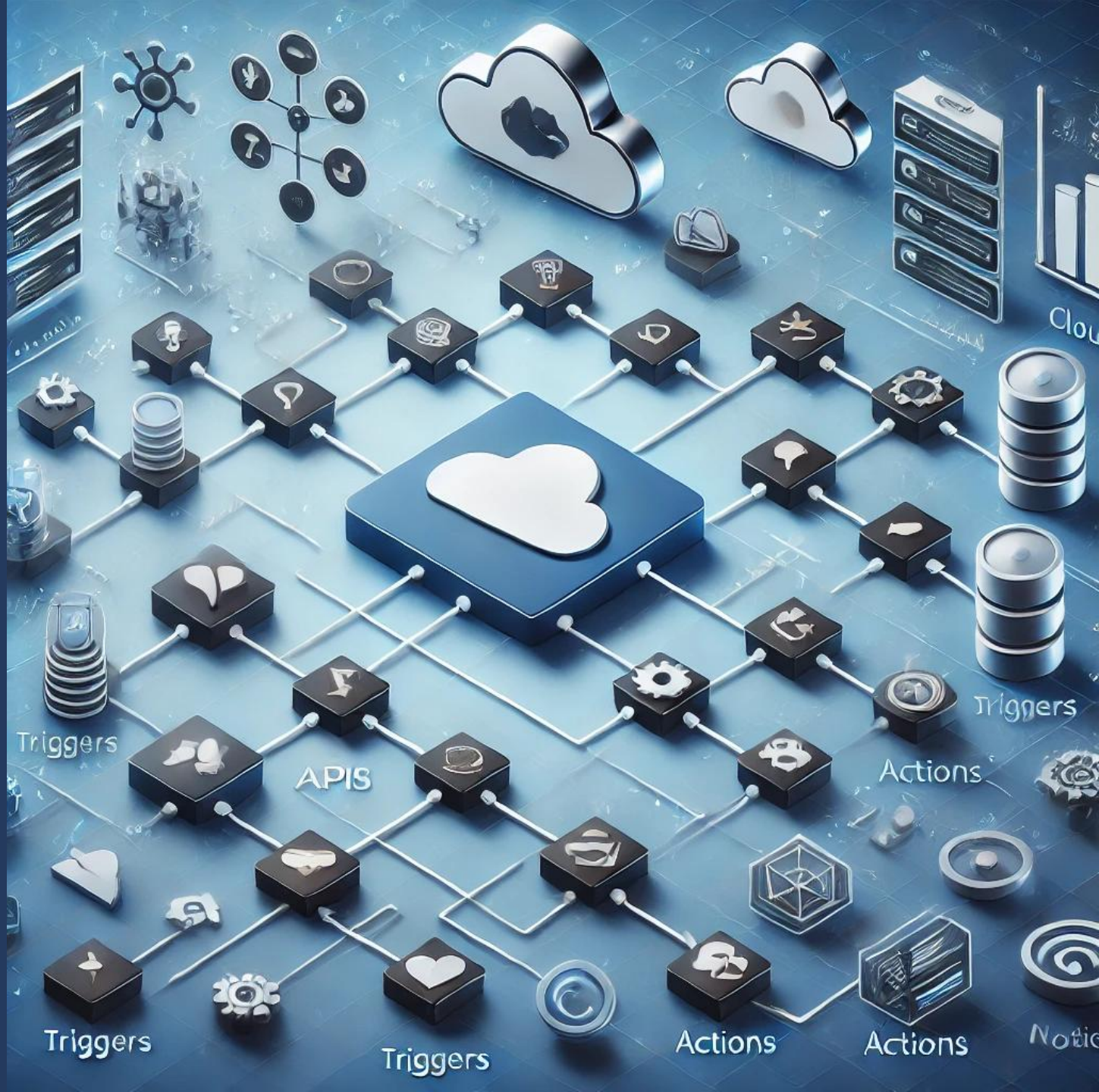
```
public class LightsPlugin
{
    private readonly List<LightModel> _lights;

    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")]

    }
}
```

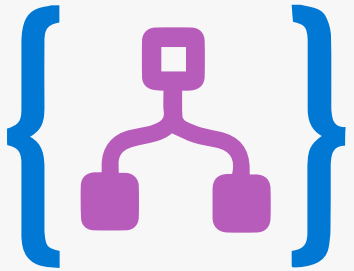
Plugin Workflow



Logic App Plugin

Create a workflow plugin using Logic Apps





Azure Logic Apps

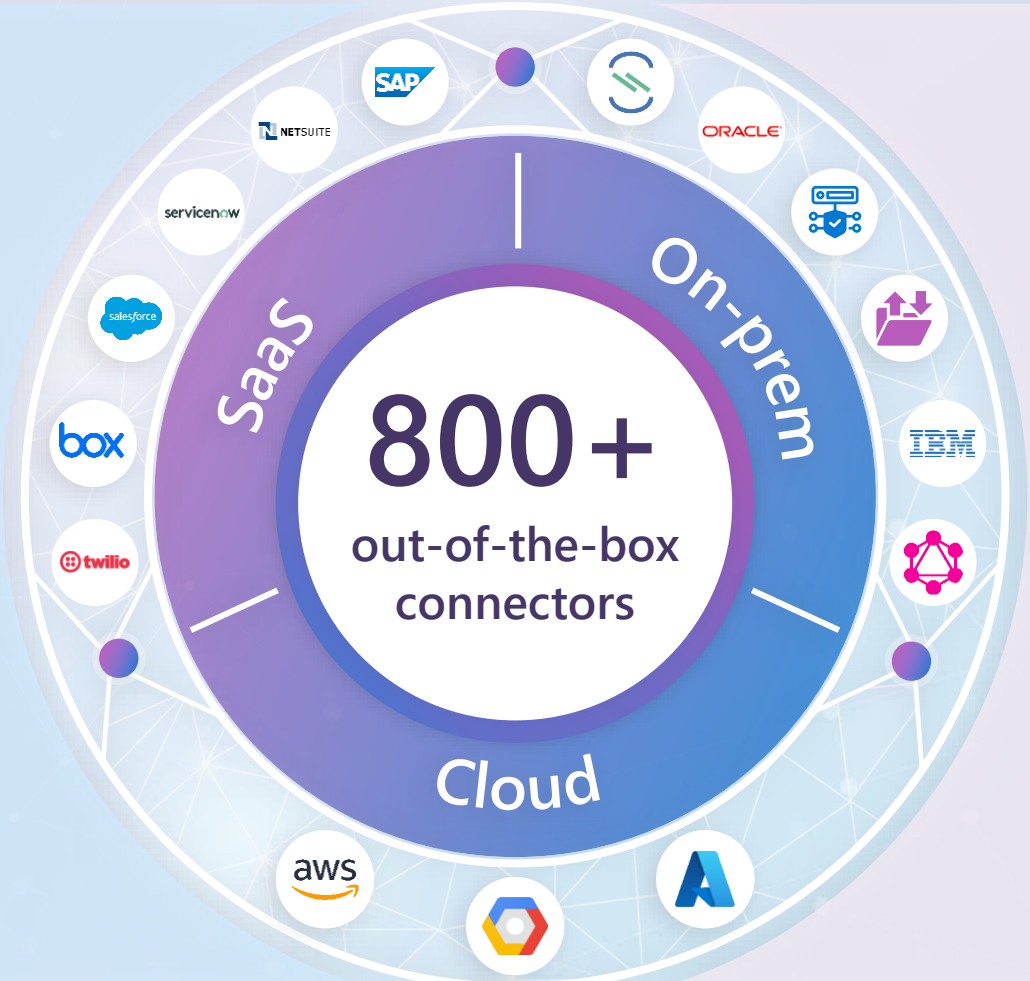
Cloud-based workflow tool for automating workflows and integrating business processes across hybrid environments in a visual manner

Connectivity for new and legacy apps using Connectors

Visual Workflow Designer

Data Mapper Drag and Drop Pre-built Functions

Workflow Assistant



Custom Connectors Capabilities

Custom Code Extensibility

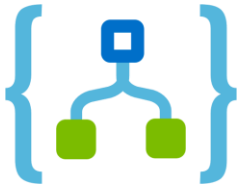
Pro-developer Experience

Azure Logic Apps Programming Model

Events - Triggers

React to timers, HTTP,
or events from your favorite
Azure services, with more
on the way

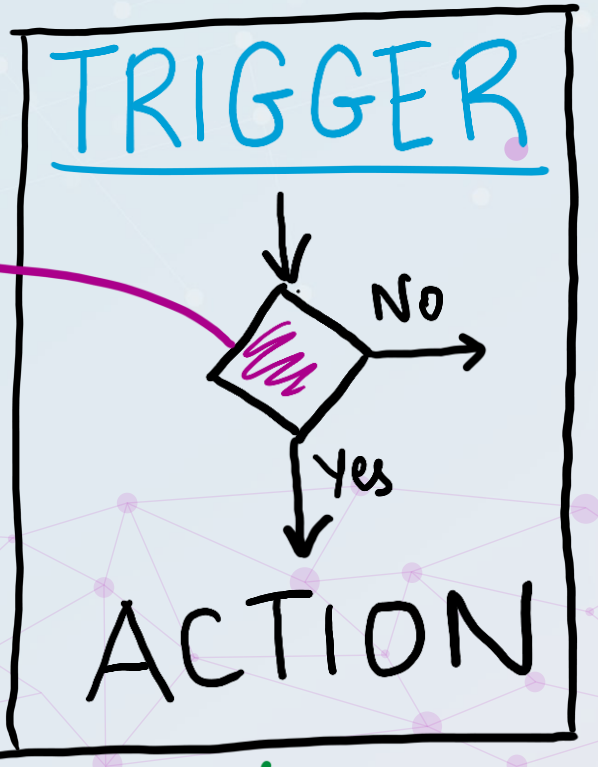
Action



If, switch, for, do-until.
Join, filter, select, create table
Execute JavaScript code
XML/JSON transform, validate
Connect to 800+ services

Outputs

Send results
to an ever-growing
collection of services



CONTROL

ALSO AN ACTION

- BLOB
- OFFICE 365
- REQUEST
- DATE & TIME

and a lot of connectors.

THIS SERIES OF STEPS IS CALLED A WORKFLOW

Semantic Kernel Plugin

Import Plugins from OpenAPI Specification

C#

```
await kernel.ImportPluginFromOpenApiAsync(
    pluginName: "lights",
    uri: new Uri("https://example.com/v1/swagger.json"),
    executionParameters: new OpenApiFunctionExecutionParameters()
    {
        // Determines whether payload parameter names are augmented with namespaces.
        // Namespaces prevent naming conflicts by adding the parent parameter name
        // as a prefix, separated by dots
        EnablePayloadNamespacing = true
    }
);
```

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

Challenge #2

Semantic Kernel Fundamentals

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

Challenge #3

Plugins

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

Challenge #4

Logic Apps

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

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

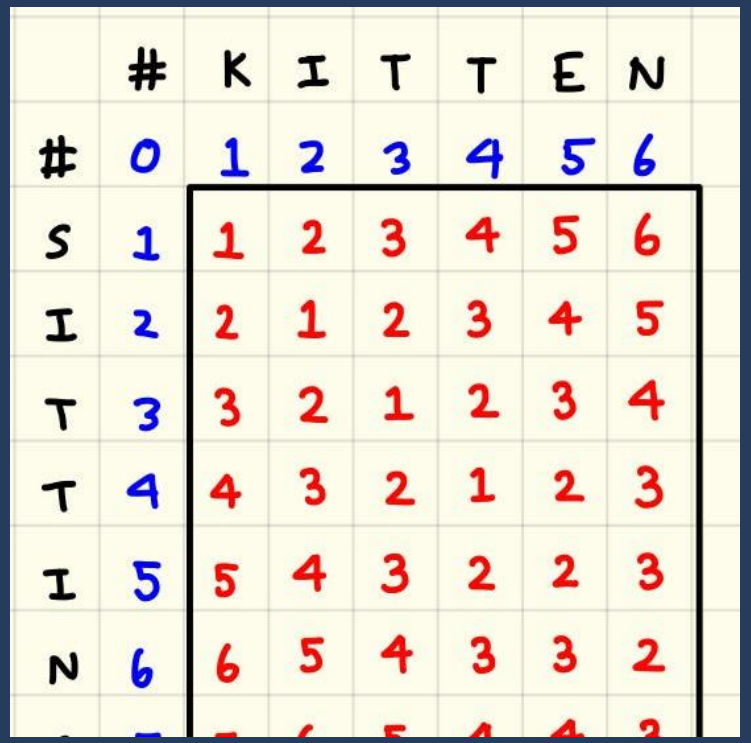
Document Chunking and External Sources

EMAIL

WEATHER

KNOWLEDGE BASE

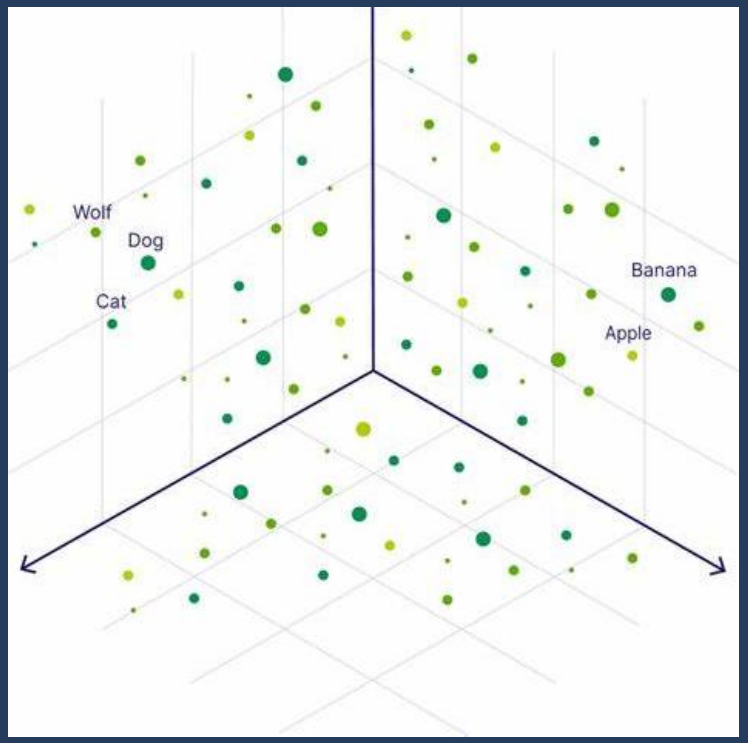
Embeddings and Vector Search



	#	K	I	T	T	E	N
#	0	1	2	3	4	5	6
S	1	1	2	3	4	5	6
I	2	2	1	2	3	4	5
T	3	3	2	1	2	3	4
T	4	4	3	2	1	2	3
I	5	5	4	3	2	2	3
N	6	6	5	4	3	3	2

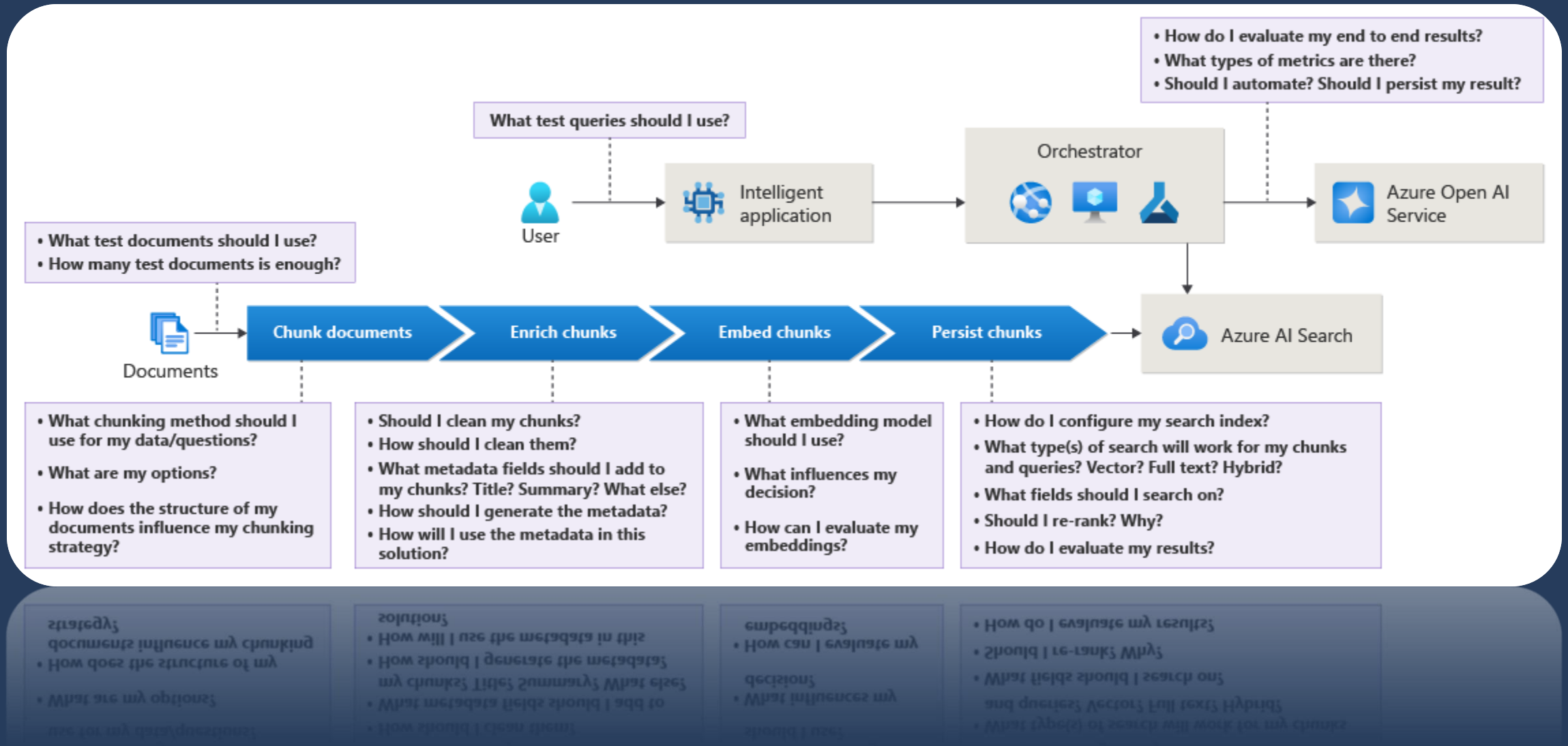
Levenshtein Distance

Search Algorithms
Past vs Present

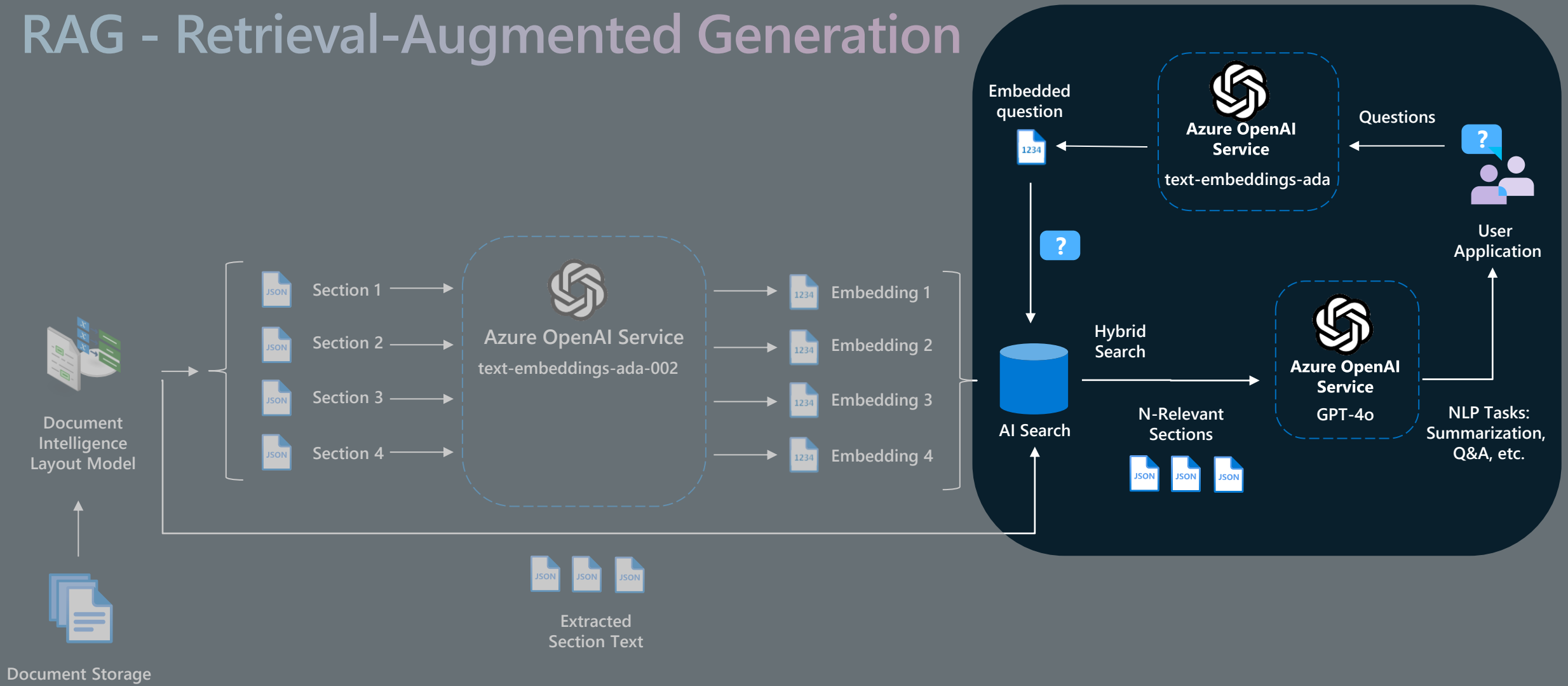


Semantic Search using
Euclidean Distance

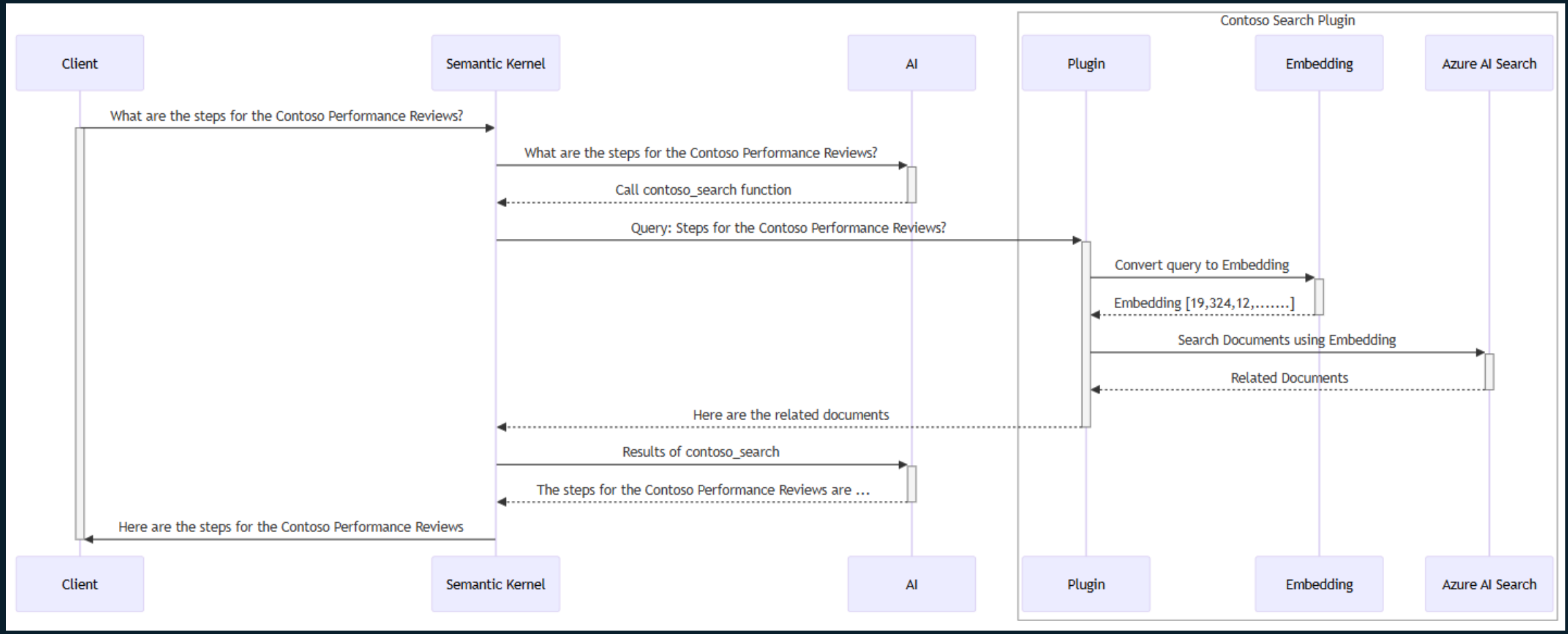
Questions to Consider for RAG Pattern



RAG - Retrieval-Augmented Generation

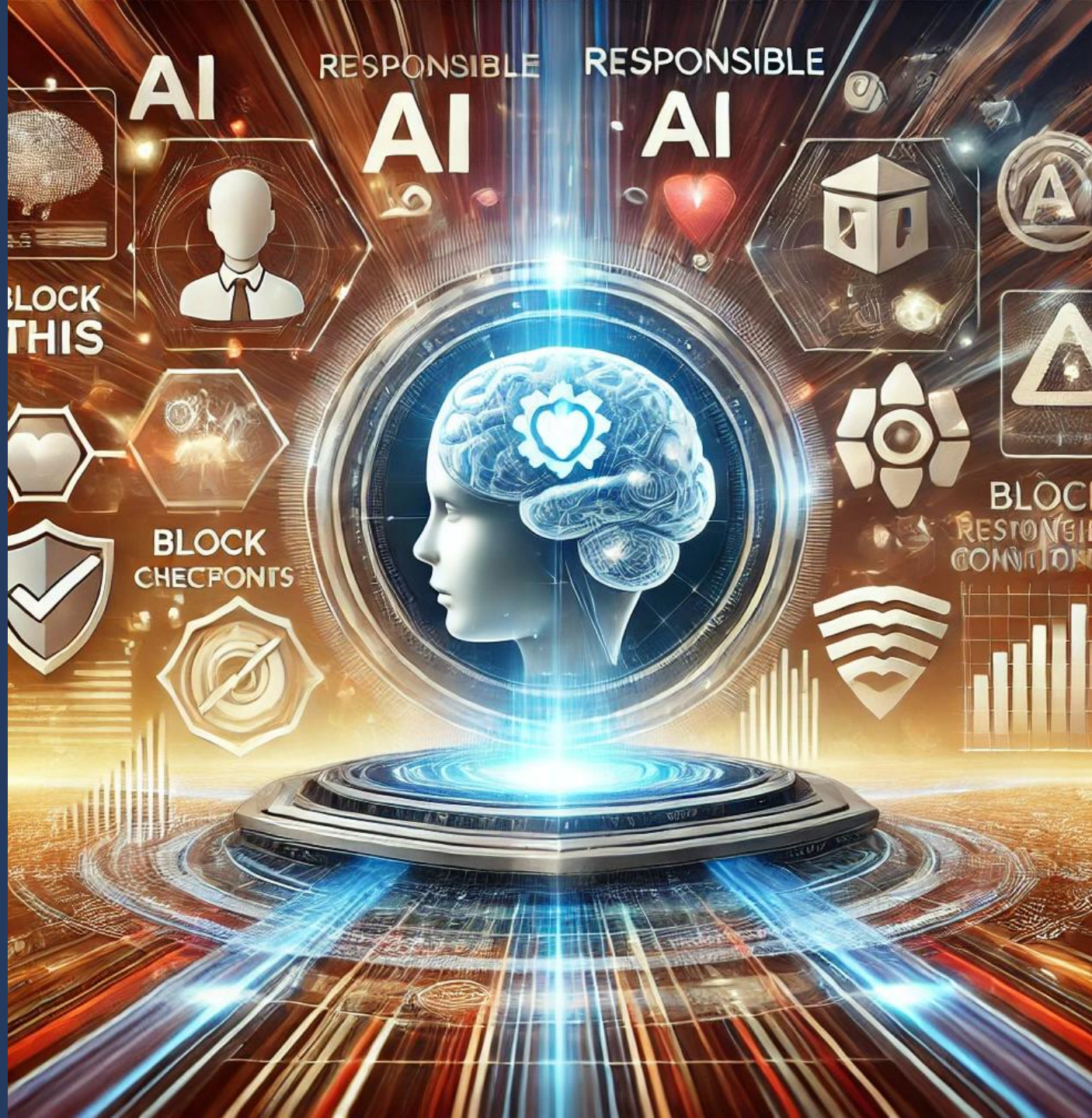


RAG – Plugin


Contoso Search Plugin
Client
Semantic Kernel
AI
Plugin
Embedding
Azure AI Search
What are the steps for the Contoso Performance Reviews?
What are the steps for the Contoso Performance Reviews?
Call contoso_search function
Query: Steps for the Contoso Performance Reviews?
Convert query to Embedding
Embedding [19,324,12,.....]
Search Documents using Embedding
Related Documents
Here are the related documents
Results of contoso_search
The steps for the Contoso Performance Reviews are ...
Here are the steps for the Contoso Performance Reviews
Client
Semantic Kernel
AI
Plugin
Embedding
Azure AI Search

Responsible AI

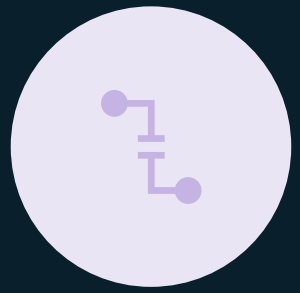
Content Filters in Azure OpenAI



Four Pillars of Responsible AI

Identify	Measure	Mitigate	Operate
Prioritize potential harms the AI system could cause	Measure frequency and severity of harms	Tools and strategies to address the risks	Operational readiness

IDENTIFY



1. Identify harms that are relevant for your specific model, application, and deployment scenario.

2. Prioritize harms based on elements of risk such as frequency and severity.

3. Conduct red team testing and stress testing starting with the highest priority harms.

4. Share this information with relevant stakeholders using your compliance processes.

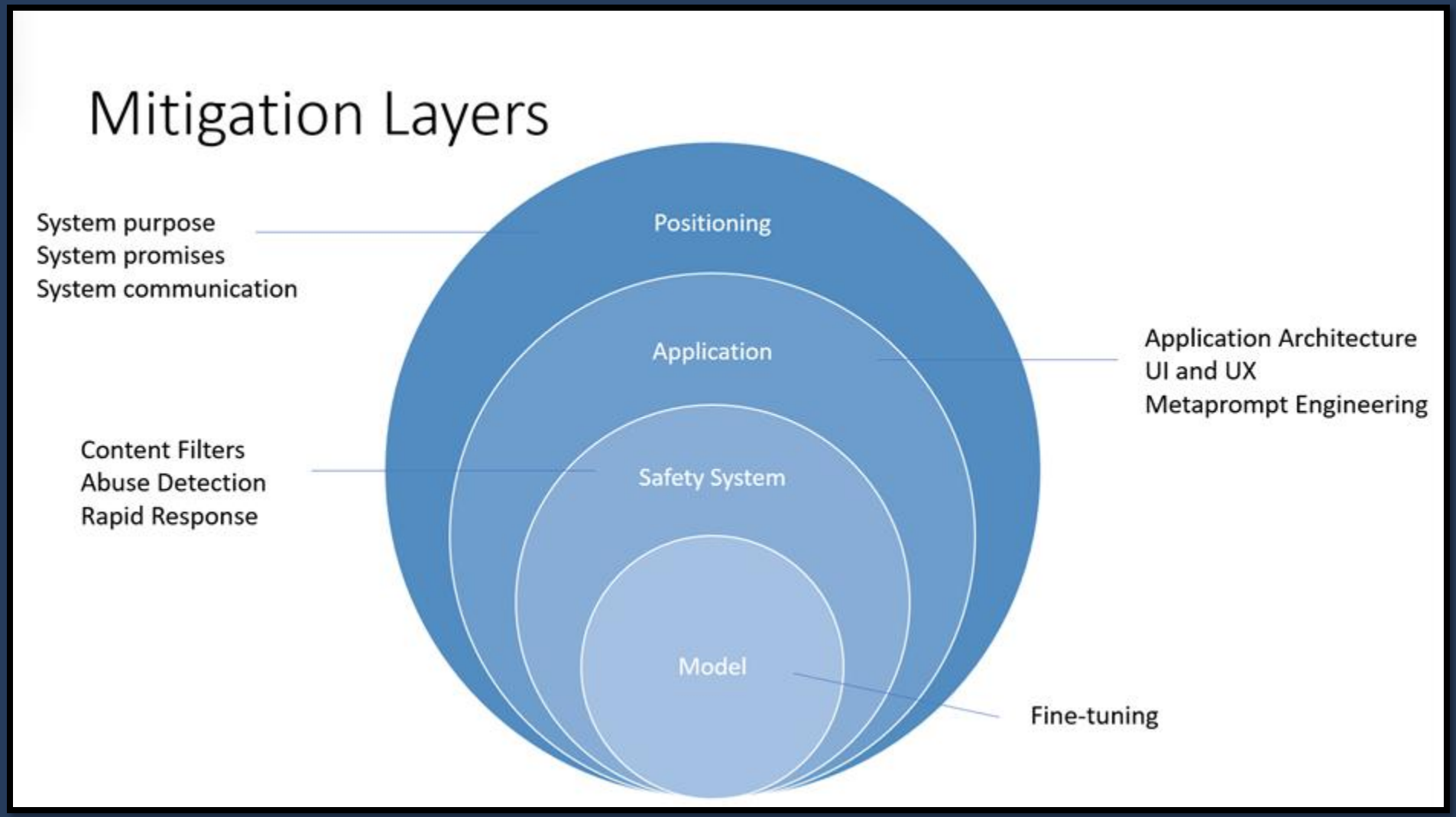
MEASURE

1. CREATE INPUTS THAT ARE LIKELY TO PRODUCE EACH PRIORITIZED HARM.

2. GENERATE SYSTEM OUTPUTS AND DOCUMENT.

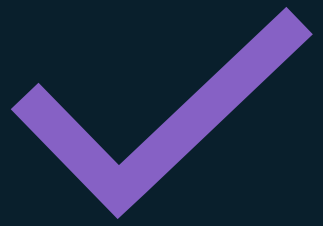
3. EVALUATE SYSTEM OUTPUTS AND REPORT RESULTS TO RELEVANT STAKEHOLDERS.

MITIGATE


Mitigation Layers
System purpose
System promises
System communication
Positioning
Application
Application Architecture
UI and UX
Metaprompt Engineering
Content Filters
Abuse Detection
Rapid Response
Safety System
Model
Fine-tuning

OPERATE

1. Determine compliance review requirements (legal, privacy, security, accessibility, etc.)



2. Develop and implement:

- Phased delivery
- Incident response plan
- Rollback plan
- Prepare for immediate action with unanticipated harms
- Develop mechanism to block people misusing the system
- Build effective user feedback channels
- Telemetry data

Content Filters

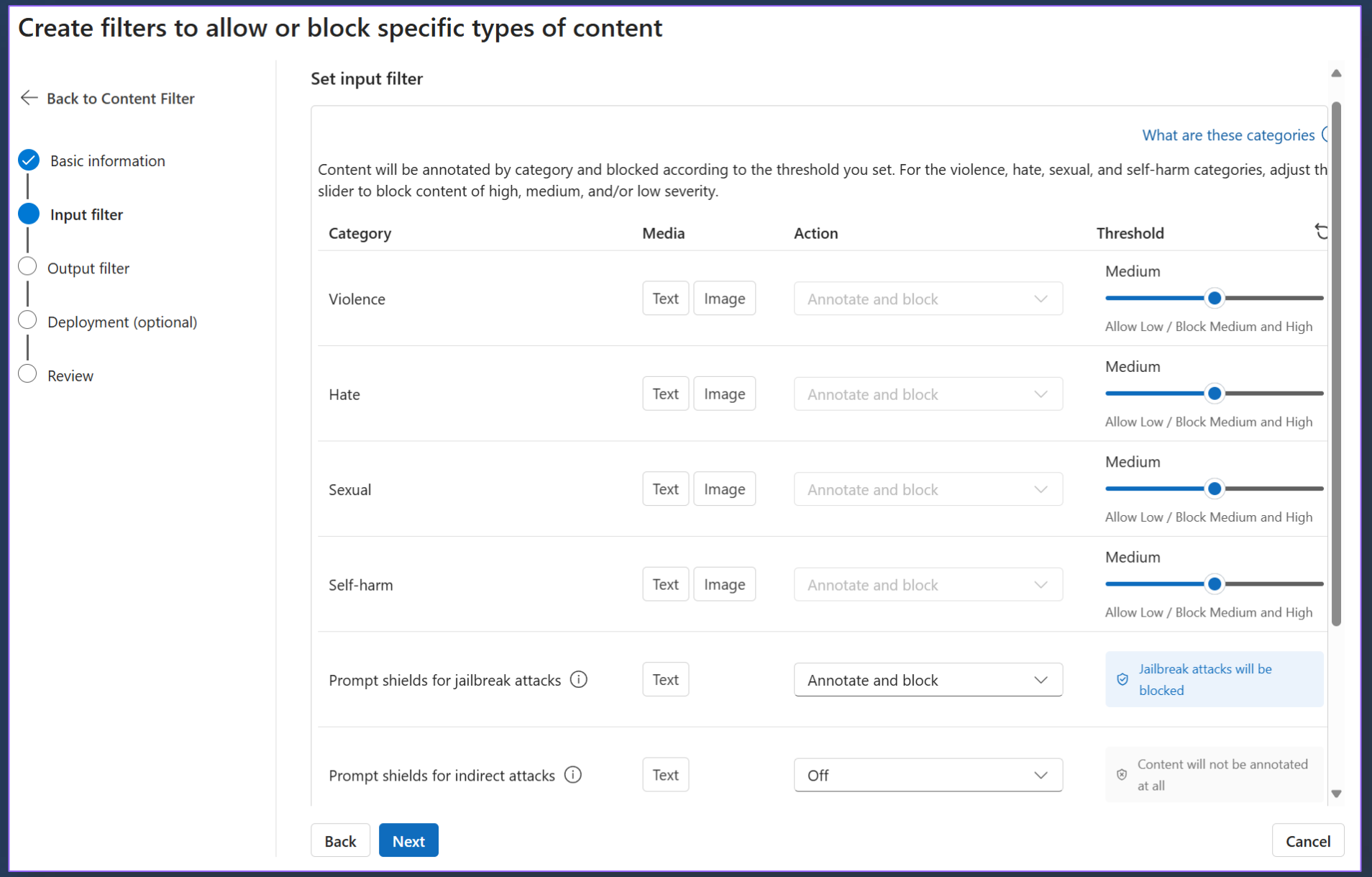
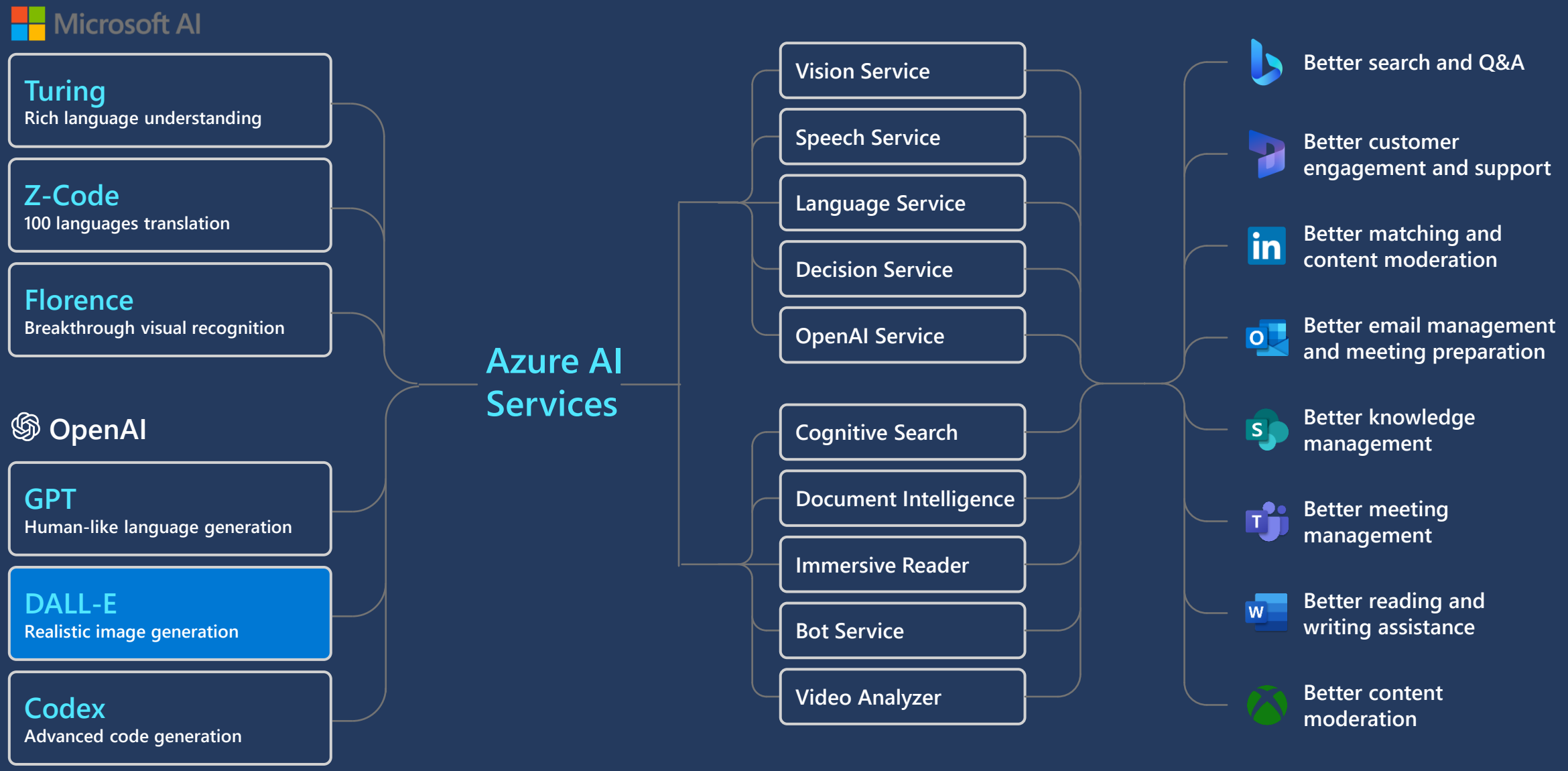


Image Generation

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

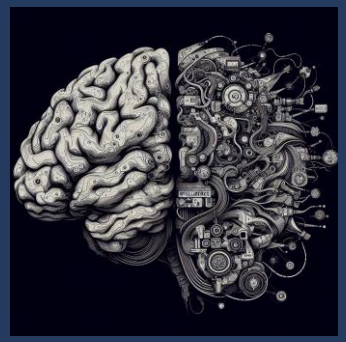
Teddy bear working on new AI research on the moon in 1980

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

Use Cases for DALL·E 3

LOGO & BRANDING: QUICK CONCEPT GENERATION.

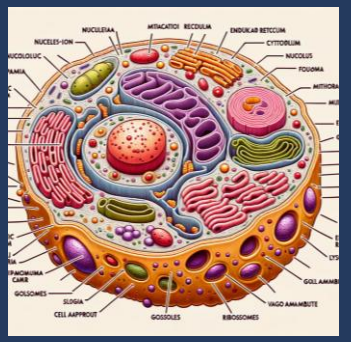
CREATIVE INSPIRATION: OVERCOME DESIGN BLOCKS.



CONTENT ILLUSTRATIONS: UNIQUE IMAGES FOR BLOGS/ARTICLES.

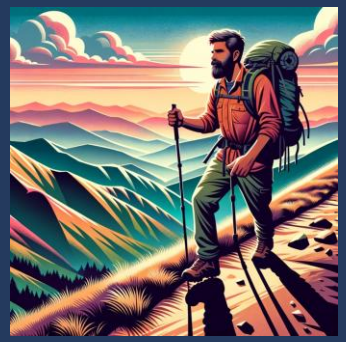
AD CAMPAIGNS: VISUALIZE MARKETING CONCEPTS.

PRODUCT VISUALIZATION: GAUGE INTEREST & FEEDBACK.



EDUCATION: CUSTOM IMAGERY FOR COURSES.

FASHION DESIGN: VISUALIZE CLOTHING PATTERNS.



GAMING: CHARACTER & ENVIRONMENT CONCEPTS.

Image Plugin

Tips and Tricks

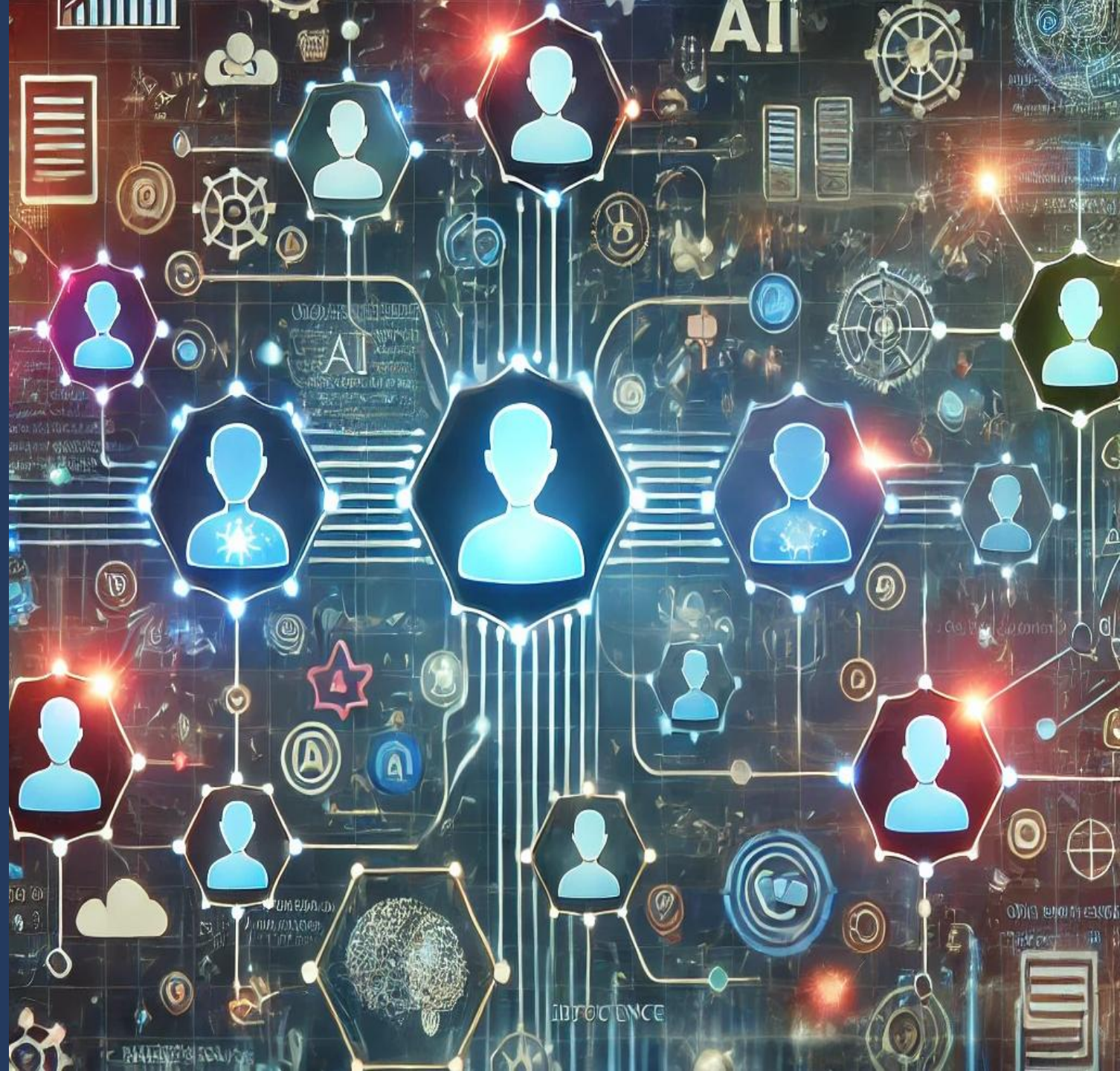
You can add a `[Description()]` Attribute to a parameter

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)
```

Multi-Agent Systems

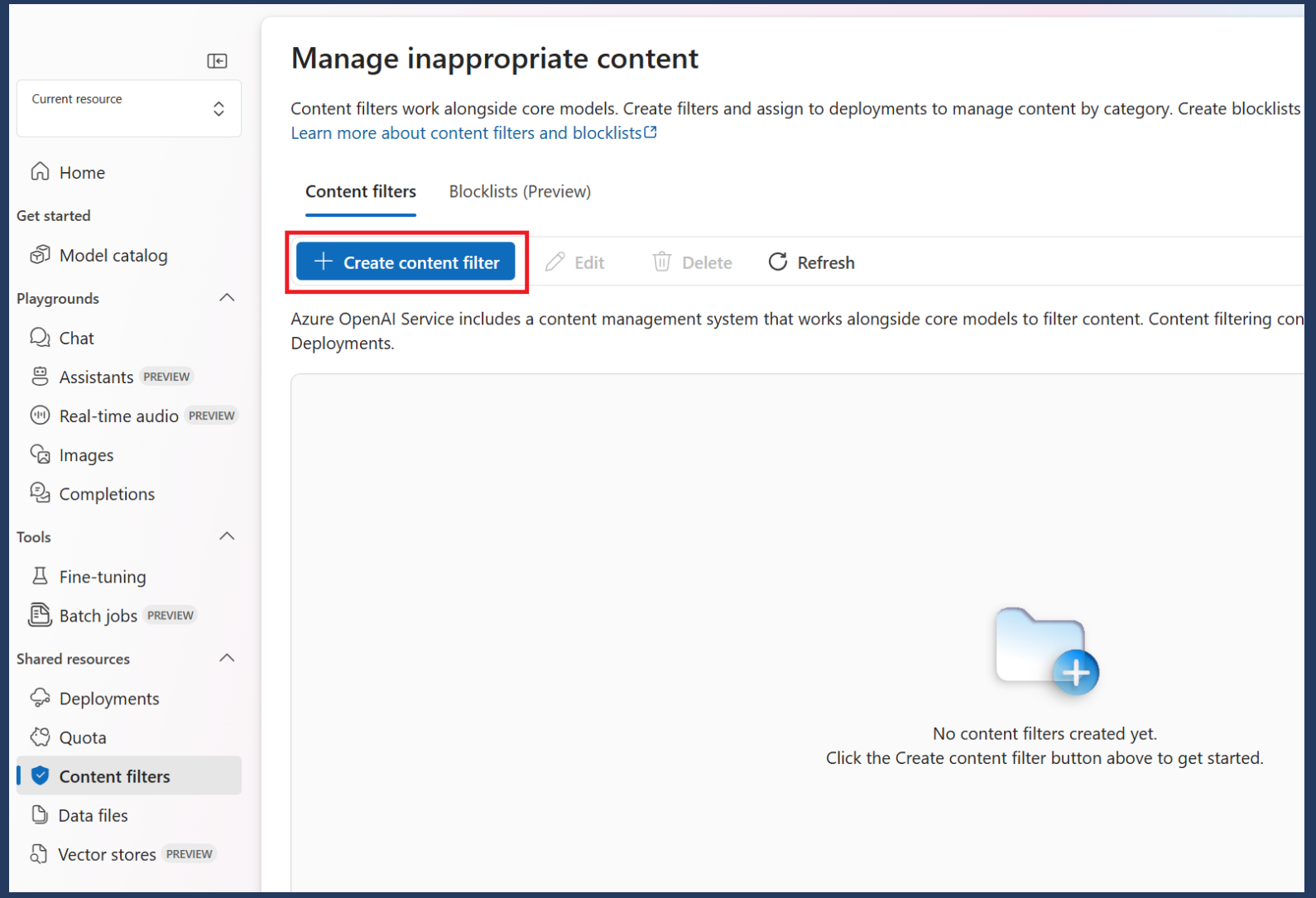
Multi-agent conversation using Semantic Kernel



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



Challenge #5

Retrieval-Augmented Generation (RAG)

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

Challenge #6

Responsible AI: Exploring Content Filters in Azure OpenAI

- Configuring content filters in Azure OpenAI Studio.
- Testing content filters.
- Creating custom filters to block specific words or phrases.

Challenge #7

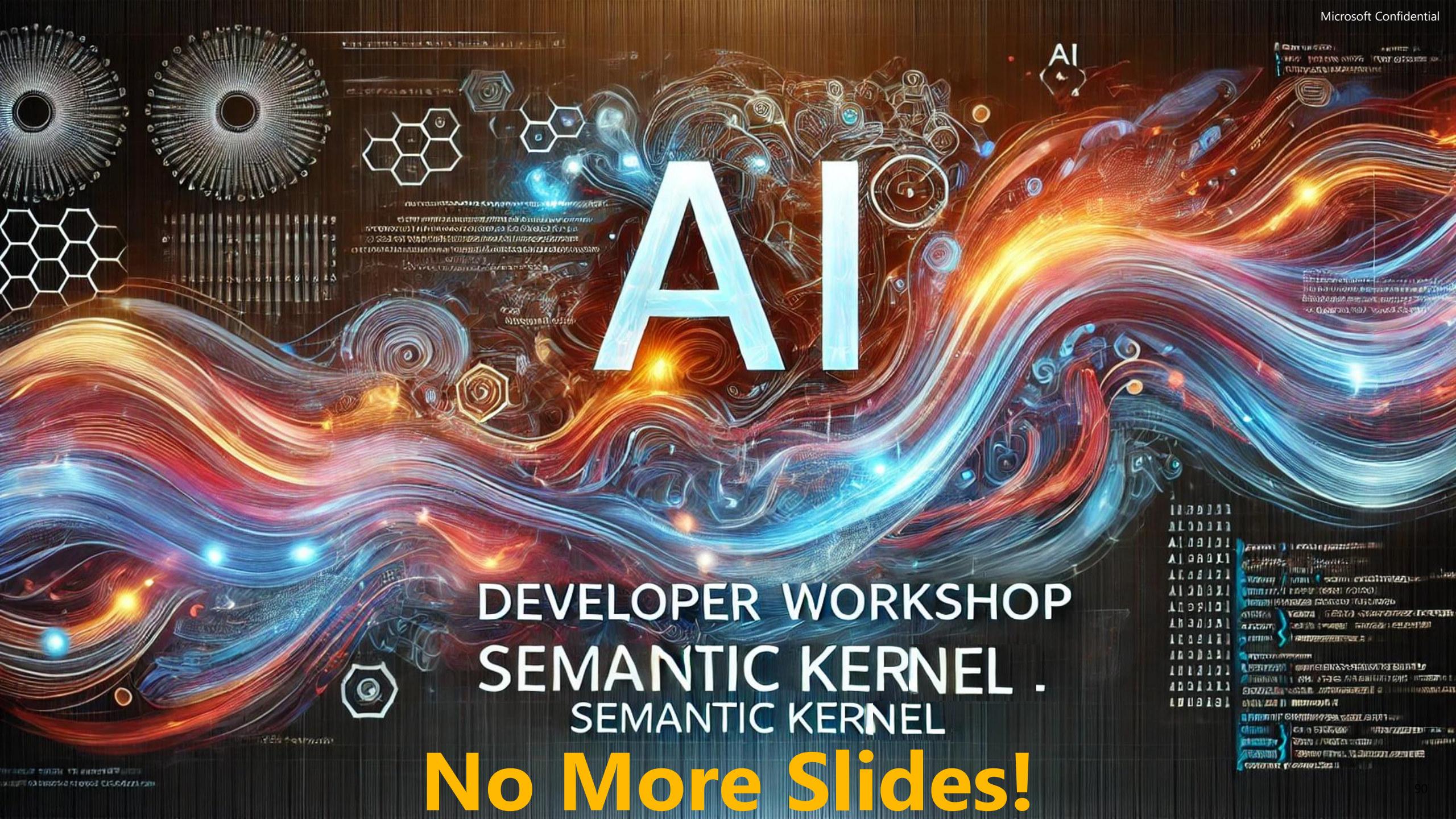
Image Generation using DALL-E

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

Challenge #8

Multi-Agent Systems

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


AI
AI
DEVELOPER WORKSHOP
SEMANTIC KERNEL .
SEMANTIC KERNEL
No More Slides!