Microsoft Azure

AI Workshop for Developers

Semantic Kernel and Azure AI Foundry!

Chris McKee, Keith Anderson, and Randy Patterson
Cloud Solution Architect

Ride the wave

Connect with Us!

Najib Zarrari

Director, Cloud Solutions Architecture

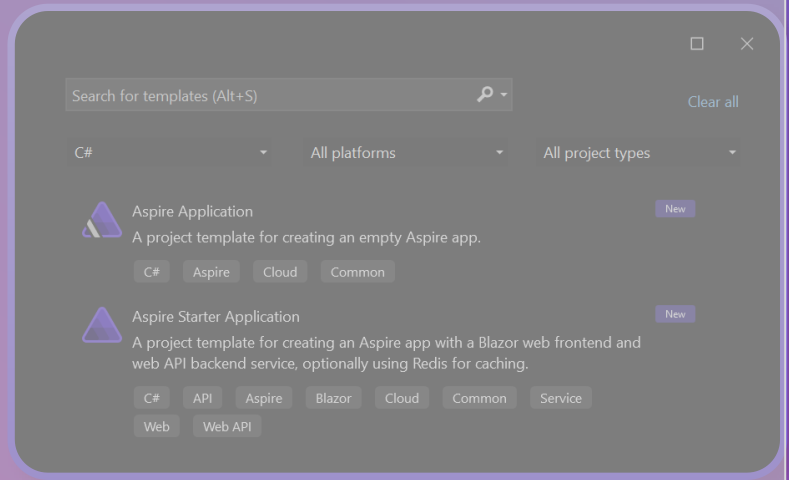
Contents

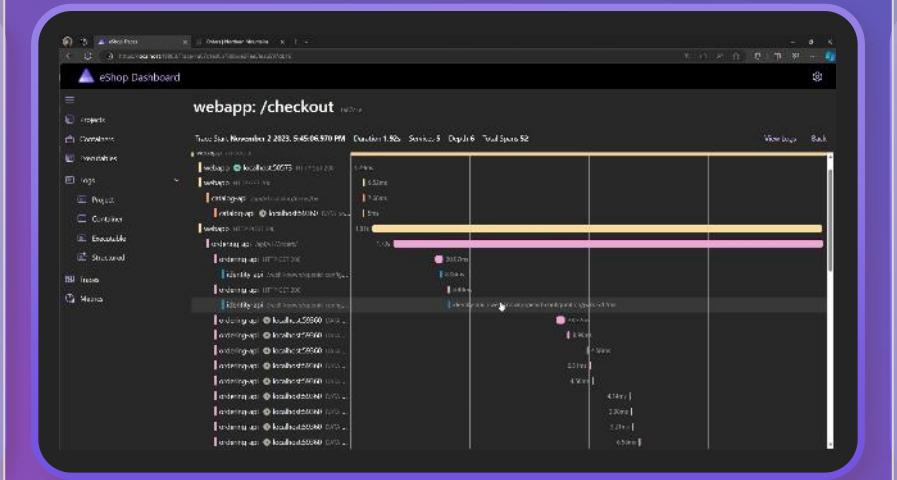
.NET Aspire

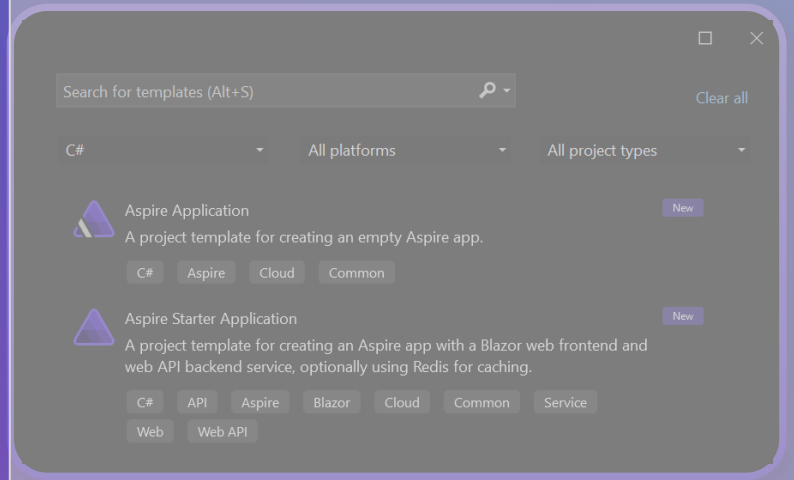
A cloud ready stack for building observable, production ready, distributed applications

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

.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 or use a GitHub flavored markdown extension for Previewing.

💡 = 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

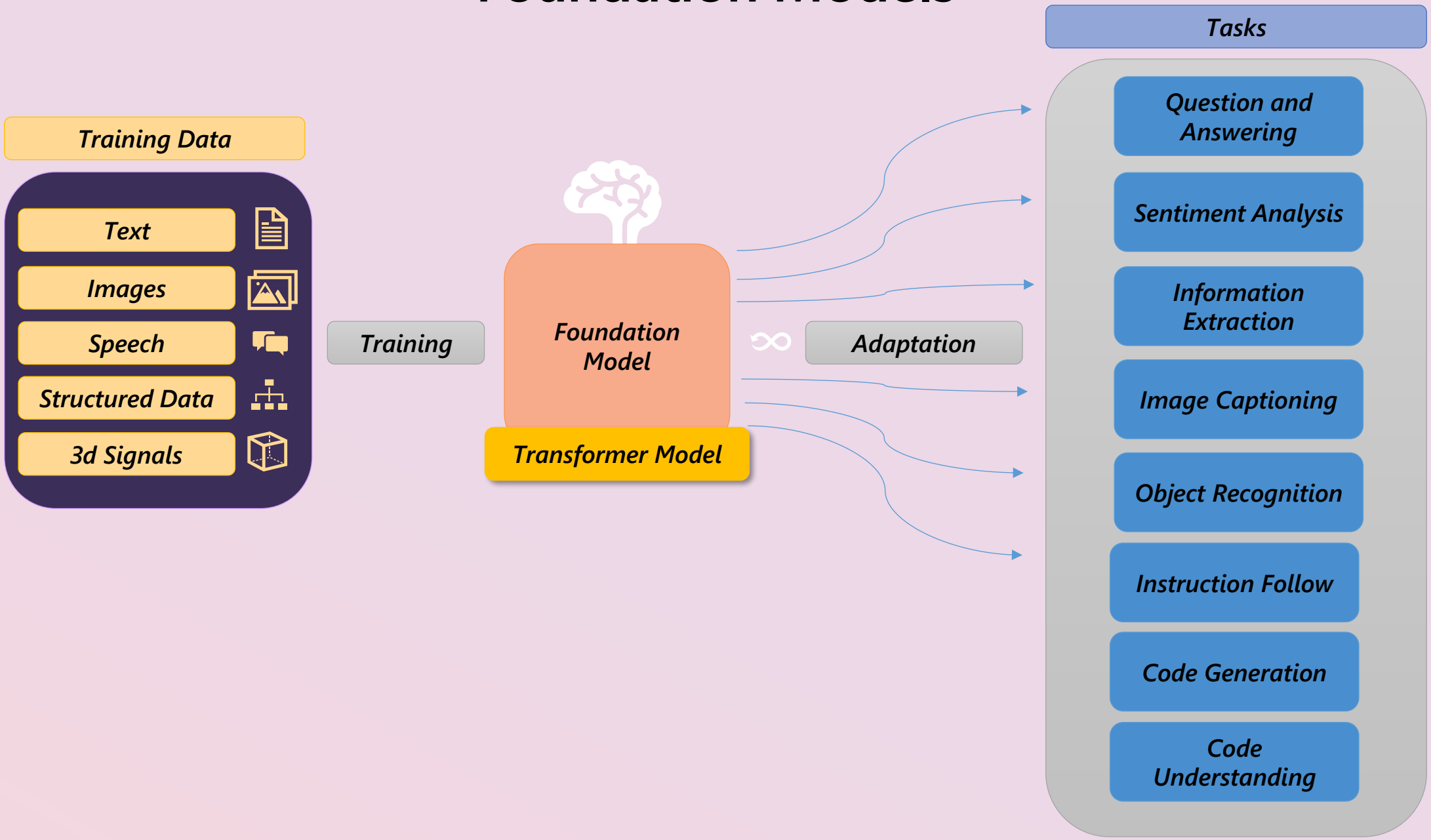
Prerequisites

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

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

Fundamentals of AI

Foundation Models



Comparison of GPT versions

GPT-4o

Fast responses: For general-purpose tasks, GPT-4o offers quick turnaround times, processing straightforward queries in seconds.
Multimodal support: GPT-4o supports multiple forms of input, making it ideal for tasks that require versatility in media.
Cost efficiency: Compared to o1, GPT-4o is far less expensive, both in terms of computation and token usage.

o1

Complex Reasoning: Specialized for tasks requiring deep reasoning, complex problem-solving, and high accuracy, especially in scientific and mathematical contexts.

Customizing Azure AI Models

Your Differentiation

Your Prompts

- *"You're a friendly, informative support agent"*
- *"Only provide answers from the data provided"*
- *"If you can't find the answer, respond with ..."*

Your Data

- *Internal Knowledge Bases*
- *Structured / Unstructured Sources*
- *Operational and Transactional Data*

GPT-3.5 | *GPT-4o* | *o1*

Azure OpenAI Service

1 Token ≈ 0.75 Word

You can think of tokens as pieces of words used for natural language processing.

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

Tokens, Possibilities, and Temperature

Example: I have an orange cat named Butterscotch.

Tokens: I have an orange cat named Butterscotch.

Example: Horses are my favorite

Possibilities:

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

Creative:

IF TEMPERATURE IS 1

Horses are my favorite animal

Horses are my favorite animals

Horses are my favorite !

Horses are my favorite animal

Strict:

IF TEMPERATURE IS 0

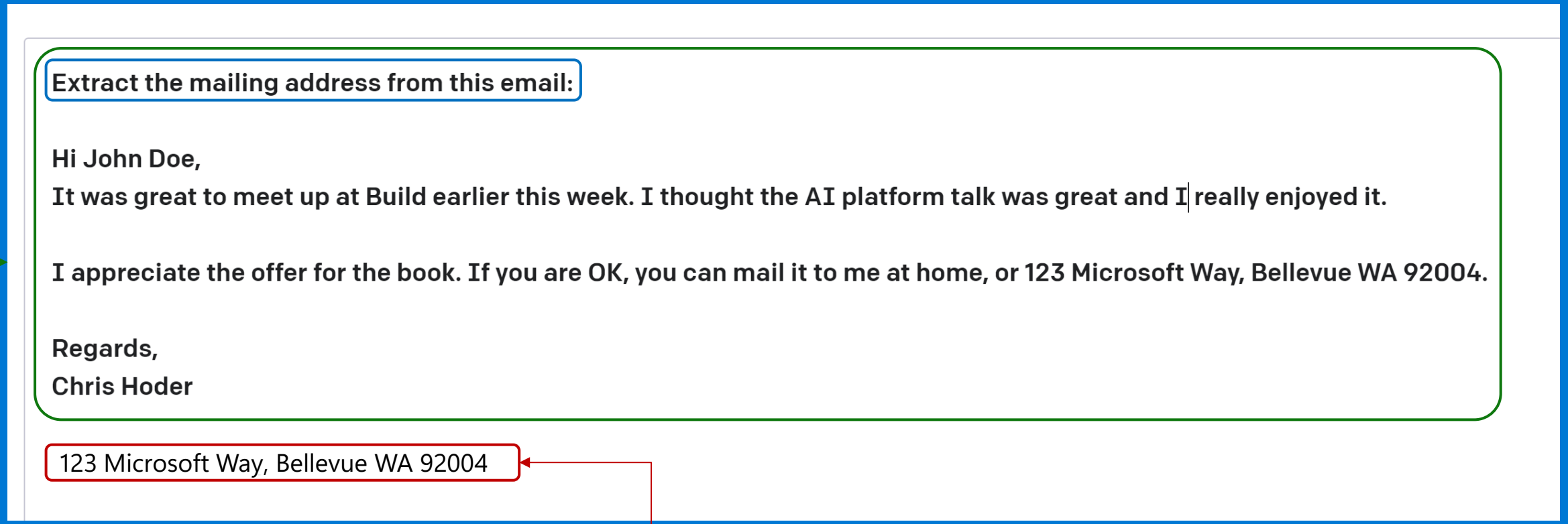
Horses are my favorite animal

Horses are my favorite animal

Horses are my favorite animal

Horses are my favorite animal

Azure OpenAI Service | GPT Prompt Design



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

Completion – Output that GPT 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?

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.

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

Summarization

Please **summarize** the information about the product

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

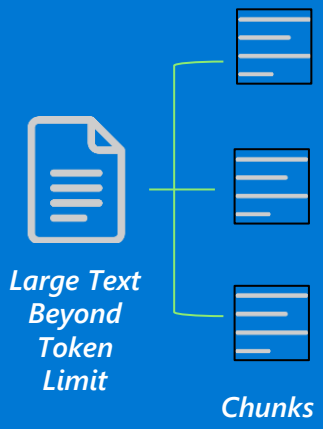
Sentiment Analysis

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.

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.



Please summarize the following text:
'Climate change is a pressing issue that needs to be addressed immediately. The increasing levels of carbon dioxide in the atmosphere are causing the Earth's temperature to rise, which is leading to more frequent and severe natural disasters. Governments around the world need to take immediate action to reduce carbon emissions and invest in renewable energy sources. Failure to do so will have dire consequences for future generations'"

Climate change is a pressing issue that needs to be addressed immediately. Governments around the world need to take immediate action to reduce carbon emissions and invest in renewable energy sources. Failure to do so will have dire consequences for future generations

Please summarize and extract topic: Global warming refers to the gradual increase in the overall temperature of the Earth's atmosphere, primarily caused by the burning of fossil fuels such as coal, oil, and natural gas. This burning releases carbon dioxide and other greenhouse gases into the atmosphere, which trap heat and cause the Earth's temperature to rise.

Climate change is a pressing issue that needs to be addressed immediately. Governments around the world need to take immediate action to reduce carbon emissions and invest in renewable energy sources. Failure to do so will have dire consequences for future generations

The text is discussing global warming, which is the gradual increase in the overall temperature of the Earth's atmosphere caused by the burning of fossil fuels. This burning releases greenhouse gases, such as carbon dioxide, into the atmosphere which trap heat and cause the temperature to rise. The text also highlights the urgency of the issue of climate change and the need for governments to take action to reduce carbon emissions and invest in renewable energy sources. Failure to address this issue will have negative consequences for future generations.

Topic: Climate change and Global warming.

Continue to next prompt

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?

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?

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.

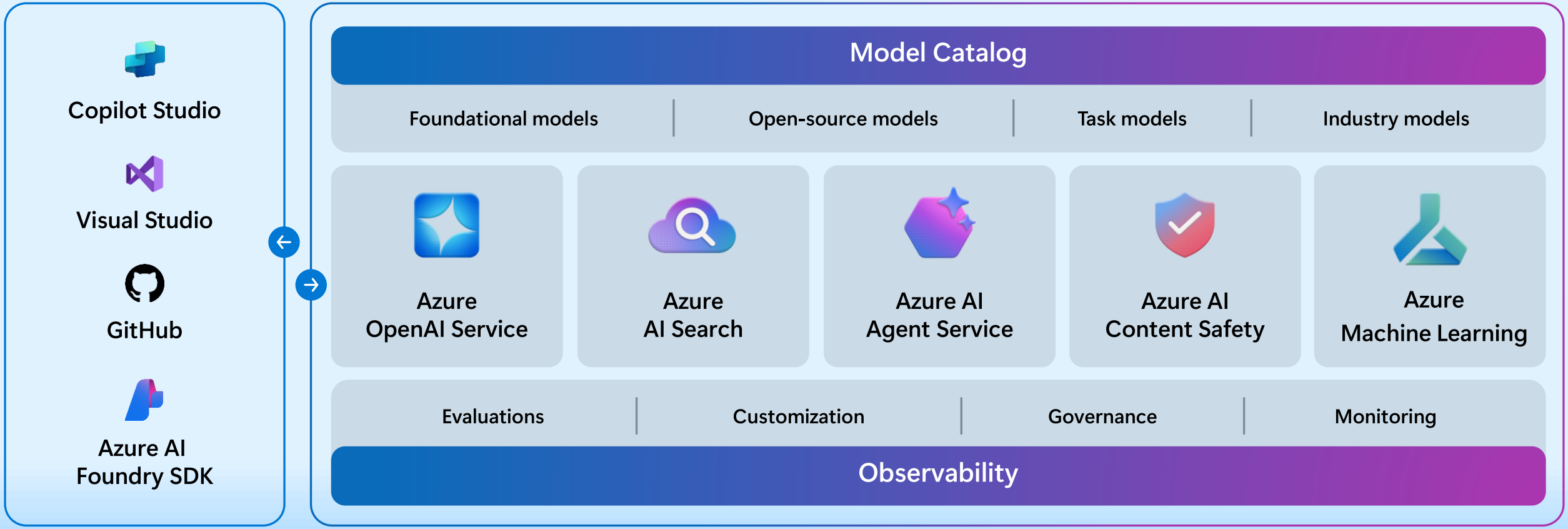
Add Clear Syntax

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. Answer in a as few words as possible.	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 very marital connection 3. "John Smith children" – to check that John has five children.

Azure AI Foundry



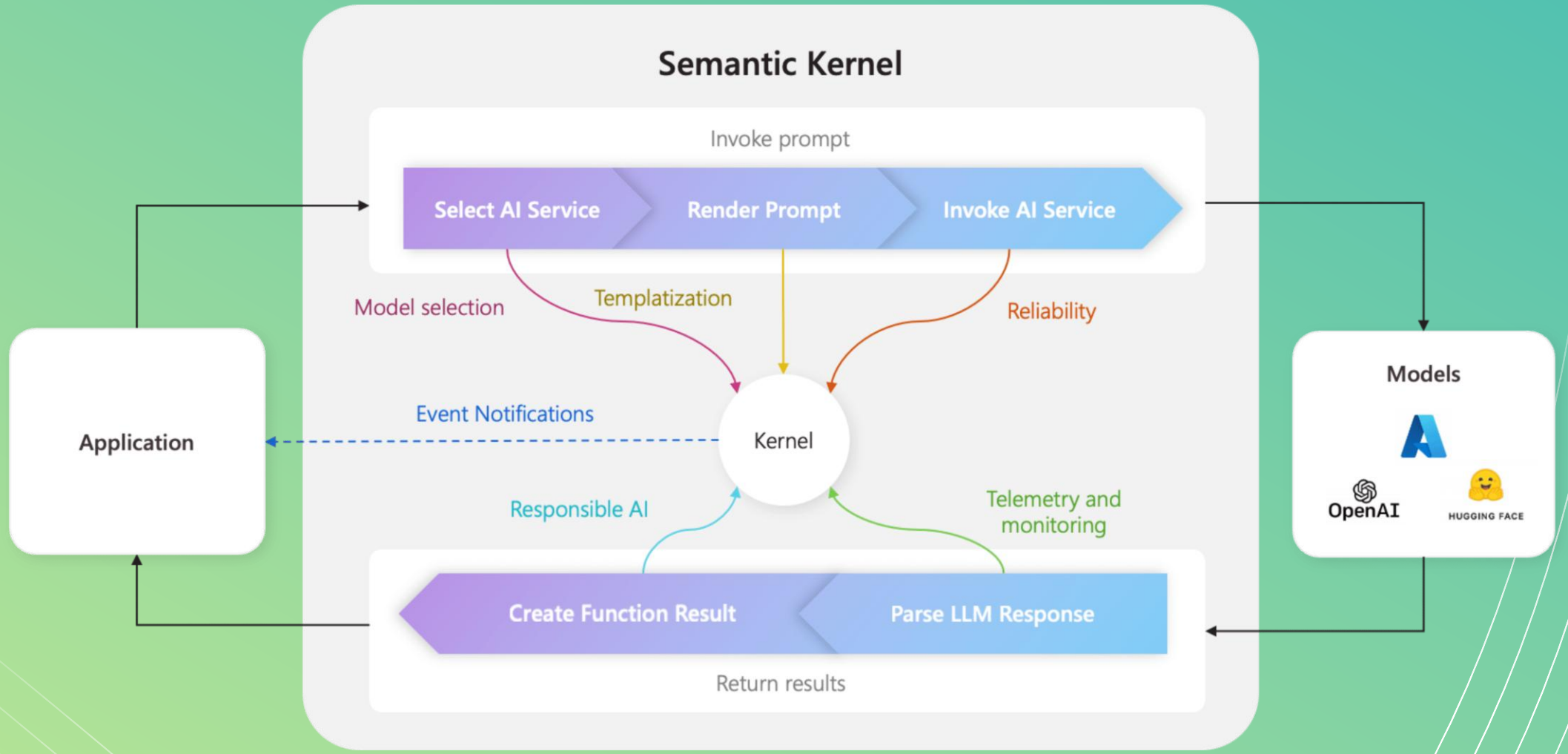
Challenge #1

Azure AI 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

The kernel is at the center of it all



1 Single-agent

Deploy agents with **Azure AI Foundry**

Managed agent micro-services

2 Multi-agent

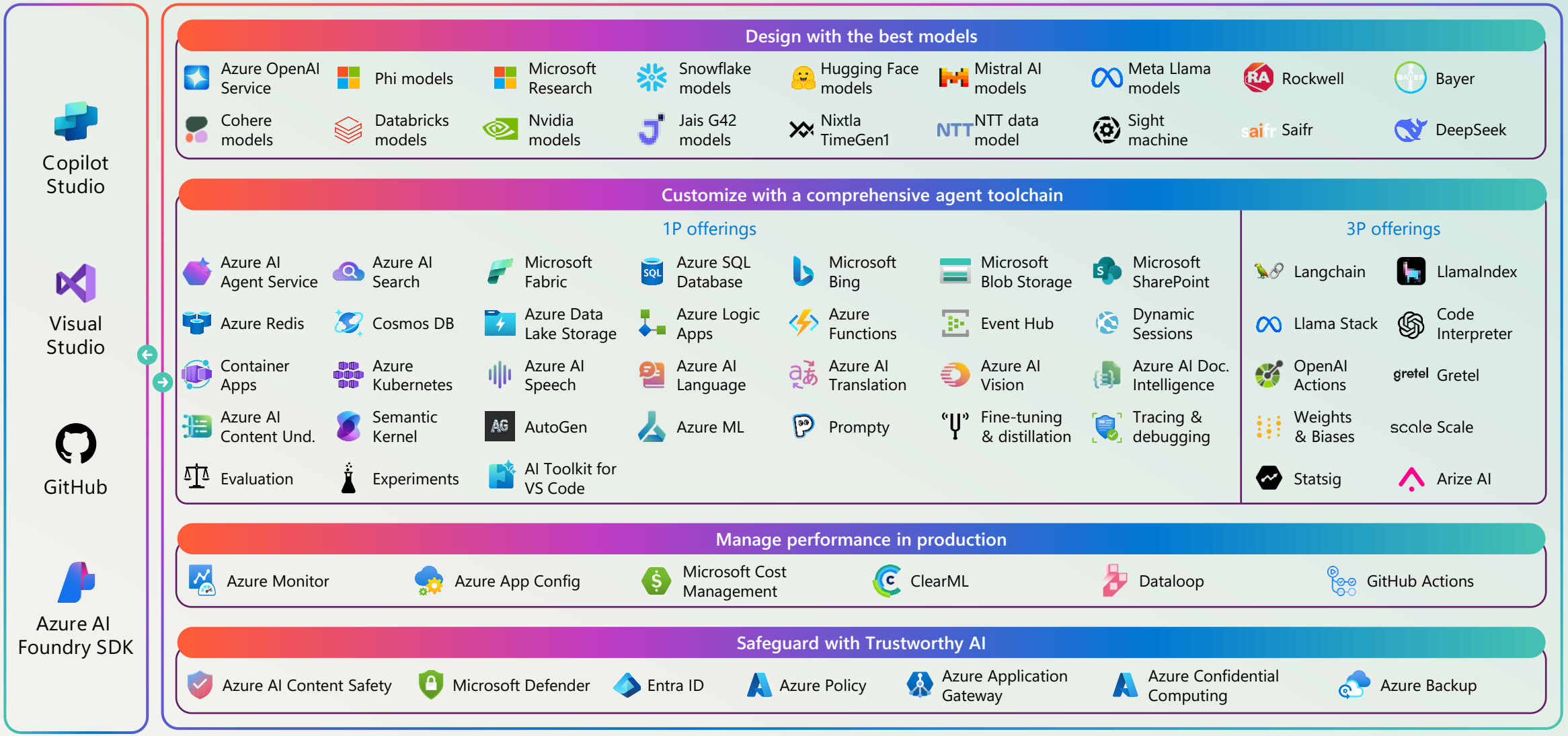
Orchestrate them together with **AutoGen** and **Semantic Kernel**

State-of-the-art research SDK

Production-ready and stable SDK

Ideation → Production

Azure AI Foundry ecosystem



Chat History

Author Roles

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

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

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

Functions or services invoked by the assistant. Provides additional information or performs actions to aid responses.

Challenge #2

Semantic Kernel Fundamentals

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

AI Superpowers Plugins

What is a Plugin?

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

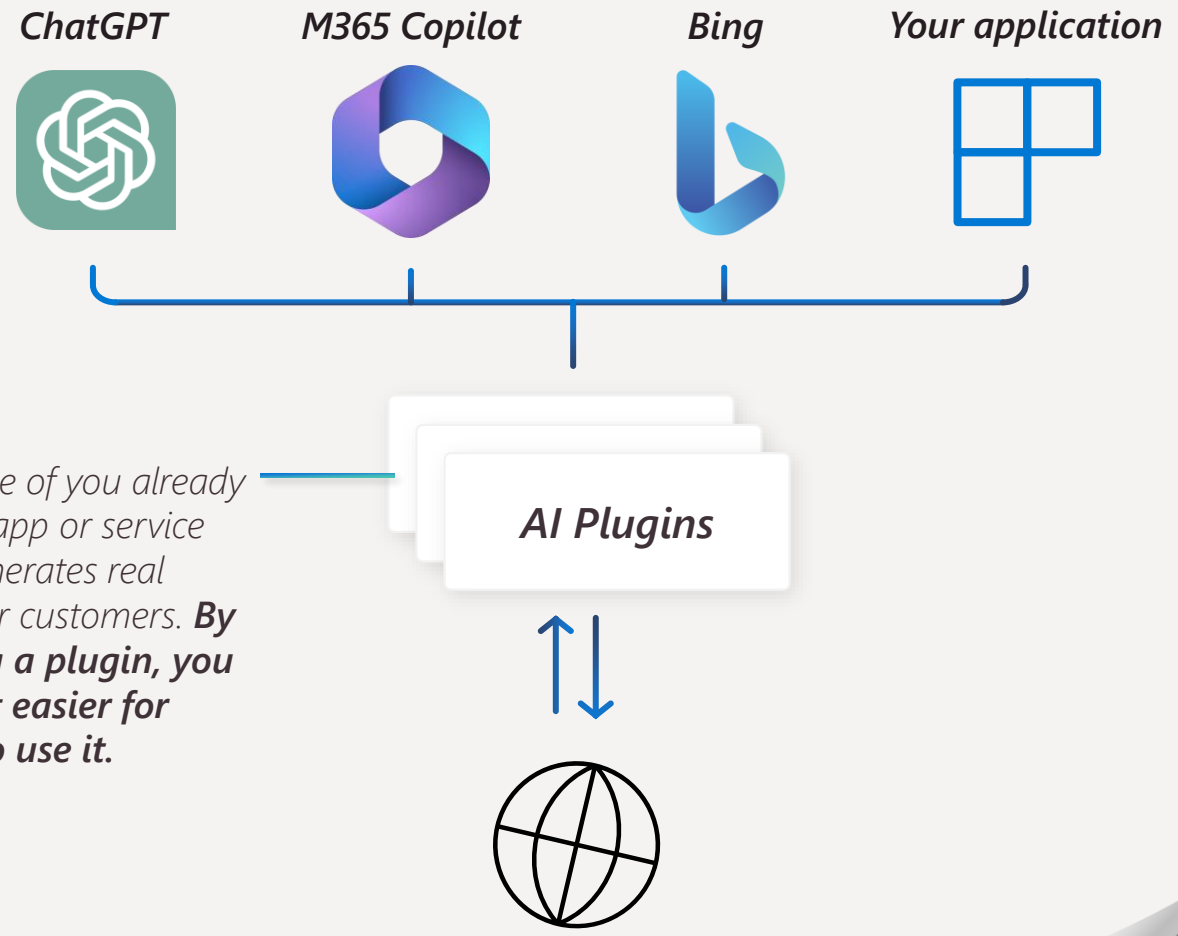
Retrieving data

Sending emails

Completing sales

Making orders

And more!

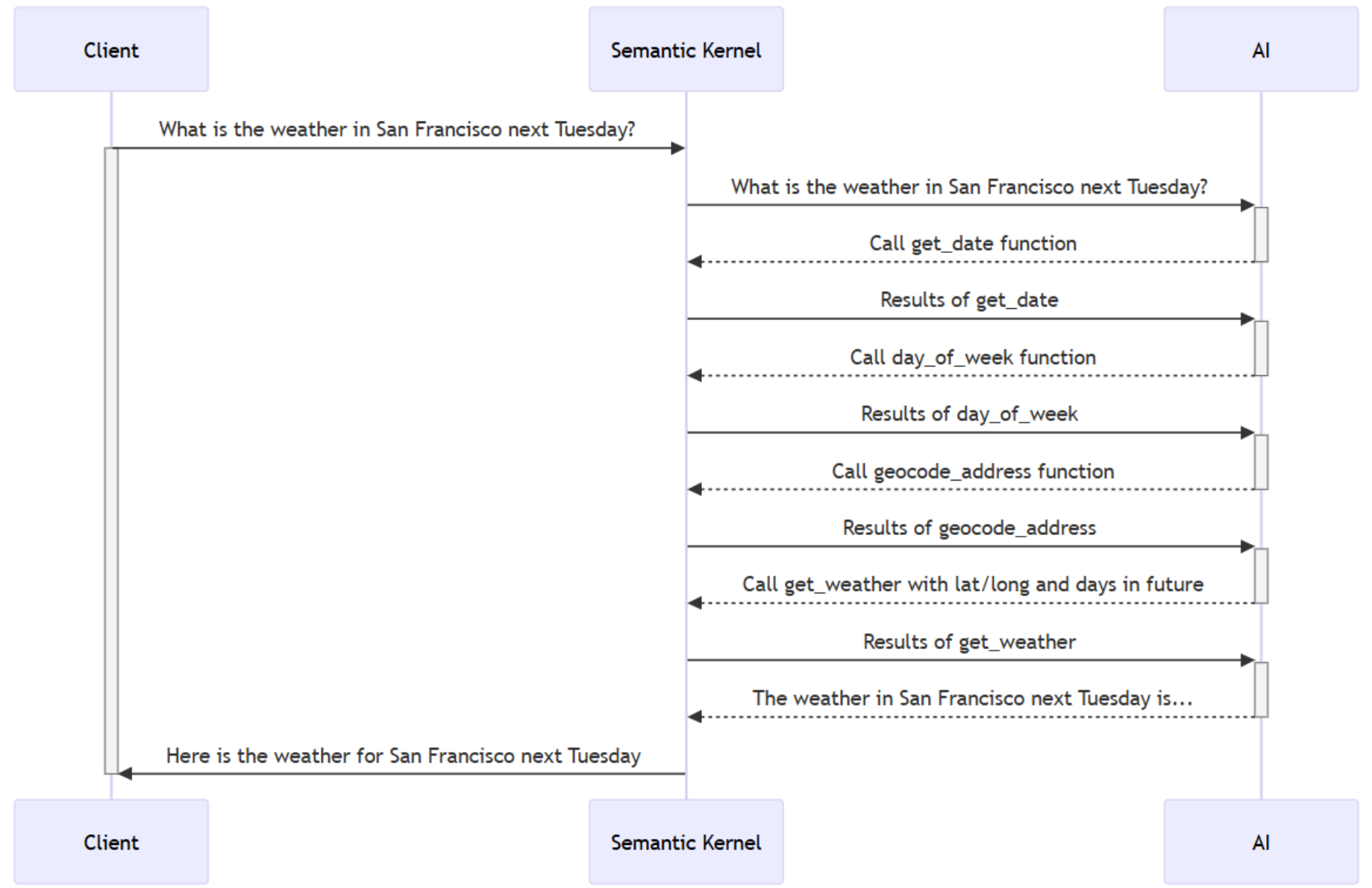


Each one of you already has an app or service that generates real value for customers. ***By making a plugin, you make it easier for users to use it.***

Defining a plugin using a class

```
{
    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")]
    public async Task<List<LightModel>> GetLightsAsync()
    {
        return _lights;
    }

    [KernelFunction("change_state")]
    [Description("Changes the state of the light")]
    public async Task<LightModel?> ChangeStateAsync(LightModel changeState)
    {
        // Find the light to change
        var light = _lights.FirstOrDefault(l => l.Id == changeState.Id);

        // If the light does not exist, return null
        if (light == null)
        {
            return null;
```

Plugin Workflow – Follow the Magic



Challenge #3

Plugins

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

Demo

Aspire Plugin Flow

Import a Plugin

1

```
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
    }
);
```

2

```
// Create the OpenAPI plugin from a local file somewhere at the root of the application
KernelPlugin plugin = await OpenApiKernelPluginFactory.CreateFromOpenApiAsync(
    pluginName: "lights",
    filePath: "path/to/lights.json"
);

// Add the plugin to the kernel
Kernel kernel = new Kernel();
kernel.Plugins.Add(plugin);
```

Challenge #4

Import Plugin with OpenAPI

- Give existing APIs to your semantic kernel via OpenAPI specification.

Retrieval-Augmented Generation

Where does RAG fit in?

LLMs are language calculators

Retrieval/RAG

OpenAI LLMs are trained on a pre-defined dataset

+

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

- Policy documents
- HR benefits
- Weather

=

This enhances AI responses to user inputs

- Improve the accuracy when you look up information
- Improve the model's handling of retrieved data

Yes, You are provided Christmas and the day after Christmas off.

LLM Knowledge

Industry/Company specific knowledge.

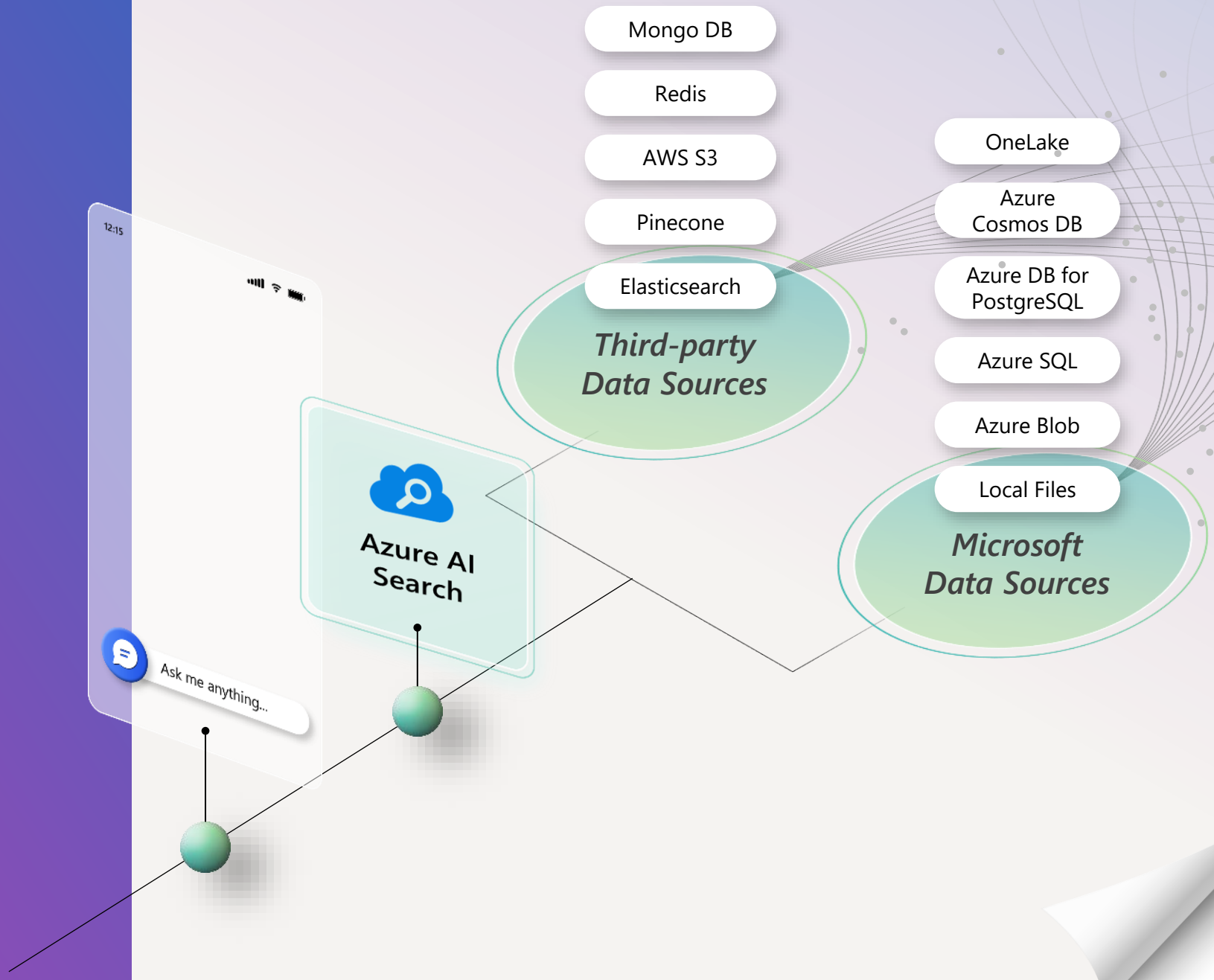
Tailored LLM

Output

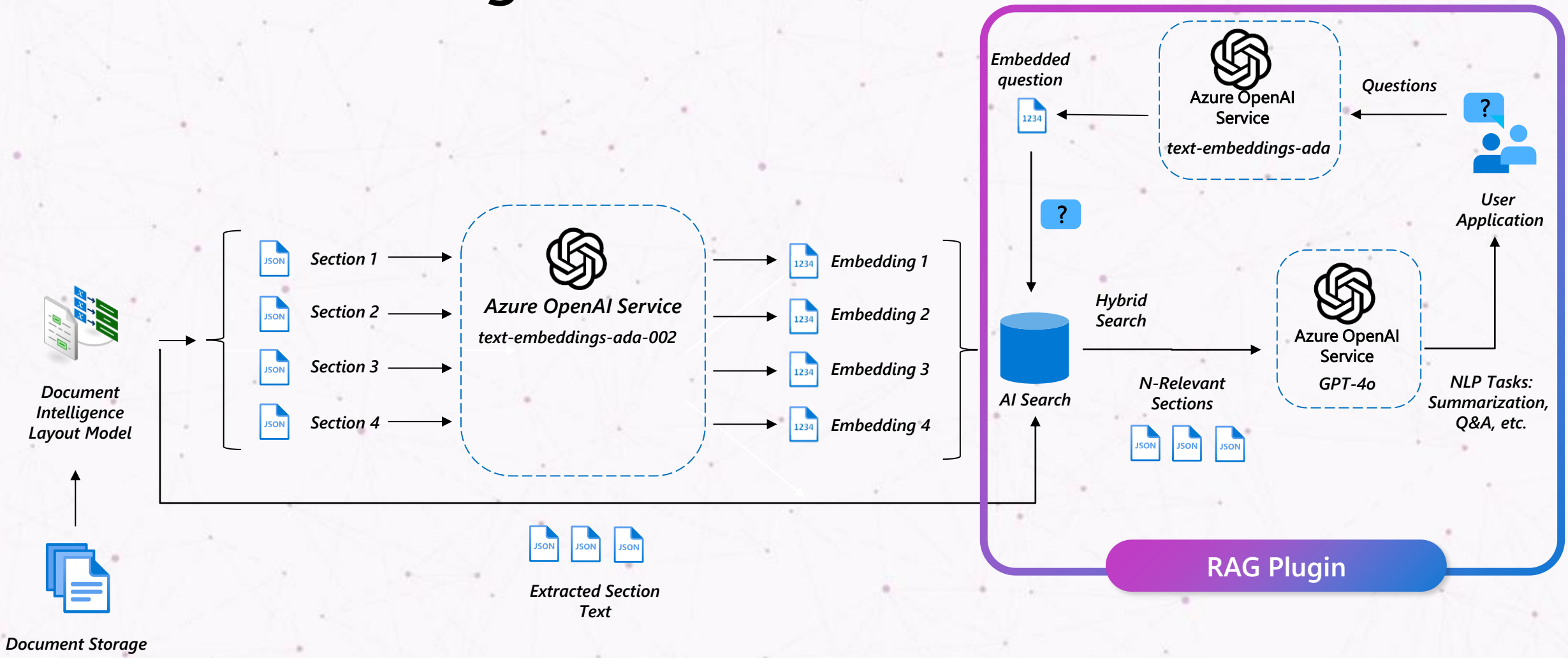
Wherever your data resides, you can bring it to life through Azure AI Search

The Azure AI Foundry difference:

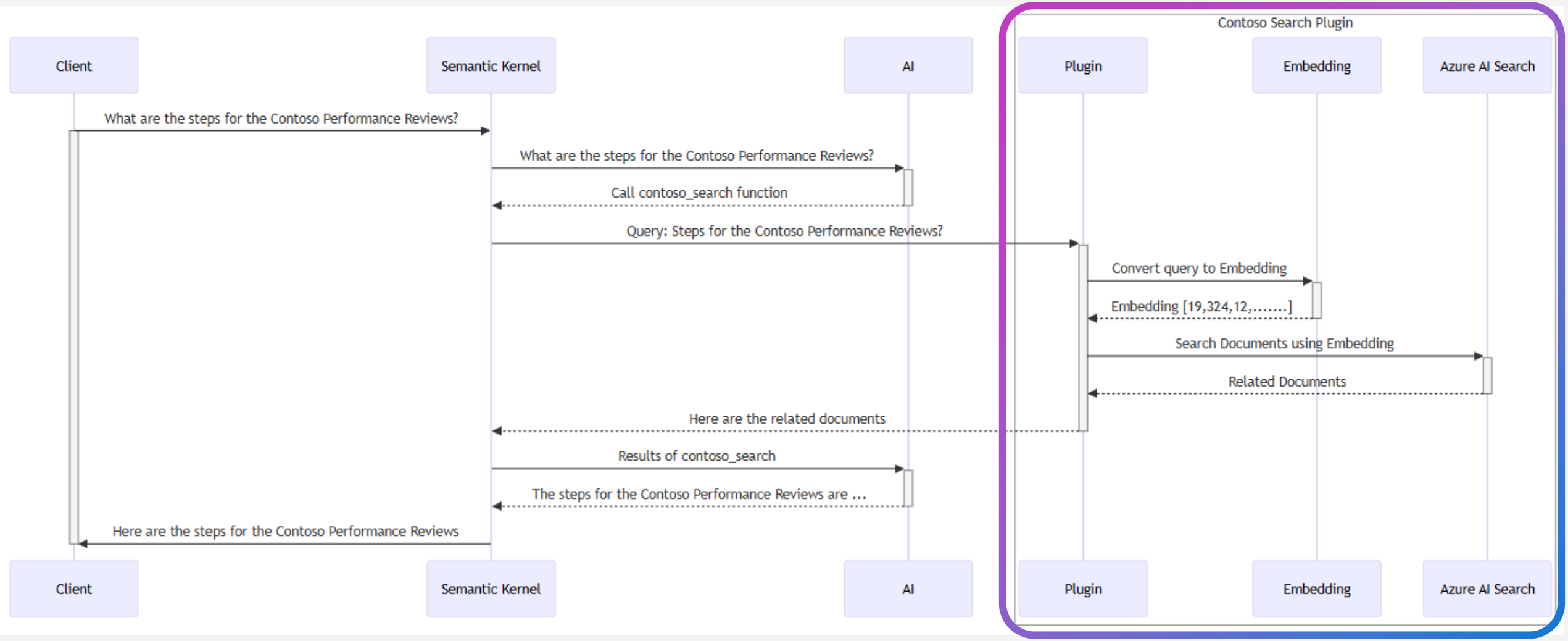
- State-of-the-art ranking
- Automatic indexing
- All data welcome



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

Challenge #5

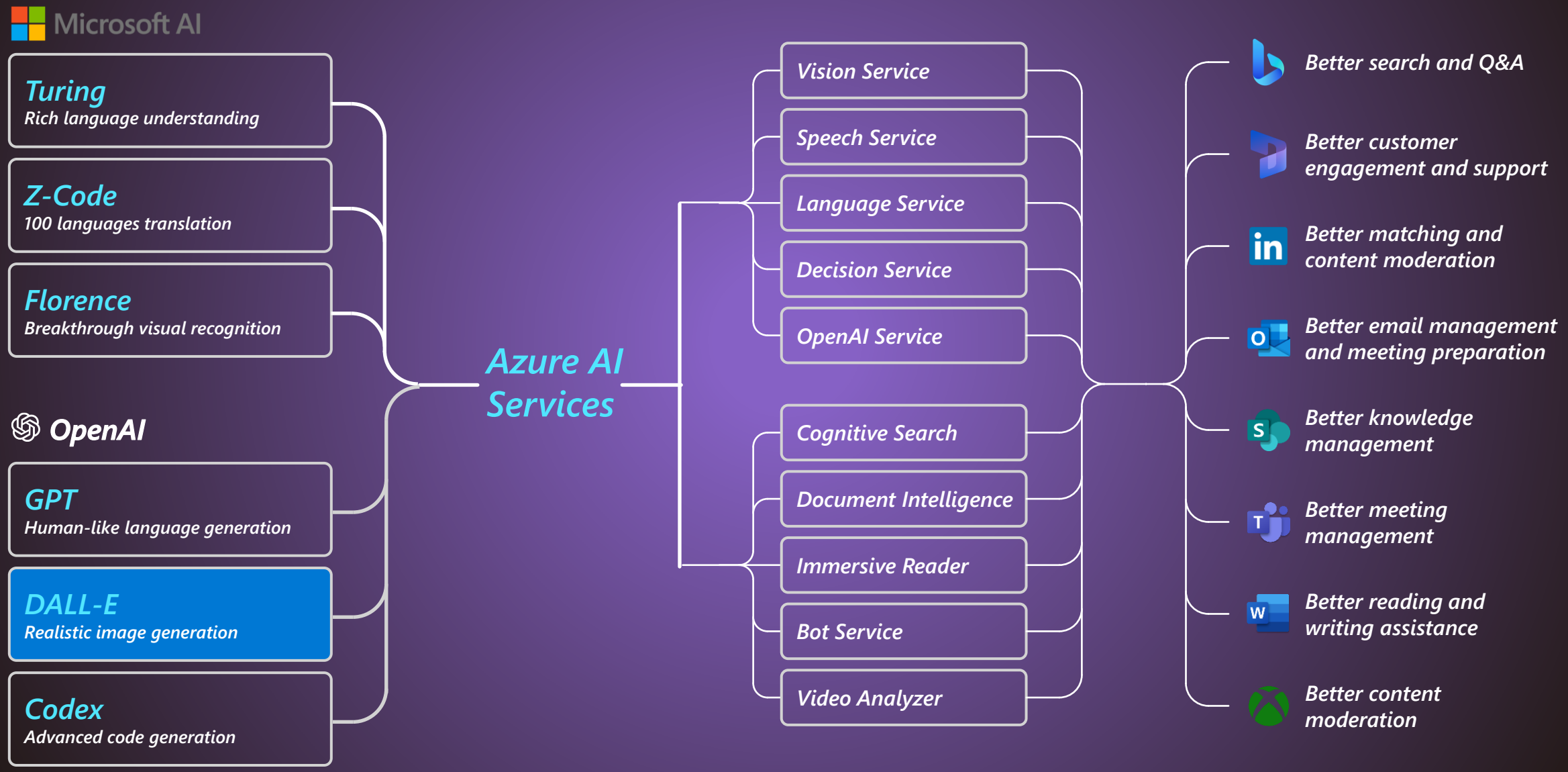
Retrieval-Augmented Generation (RAG)

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

DALL-E Plugin

Image Generation

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

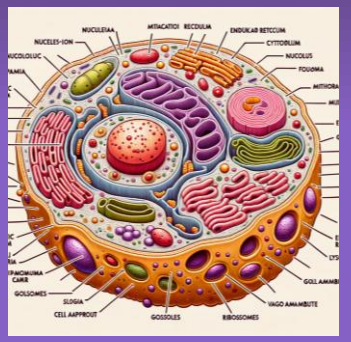
LOGO & BRANDING: QUICK CONCEPT GENERATION.

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

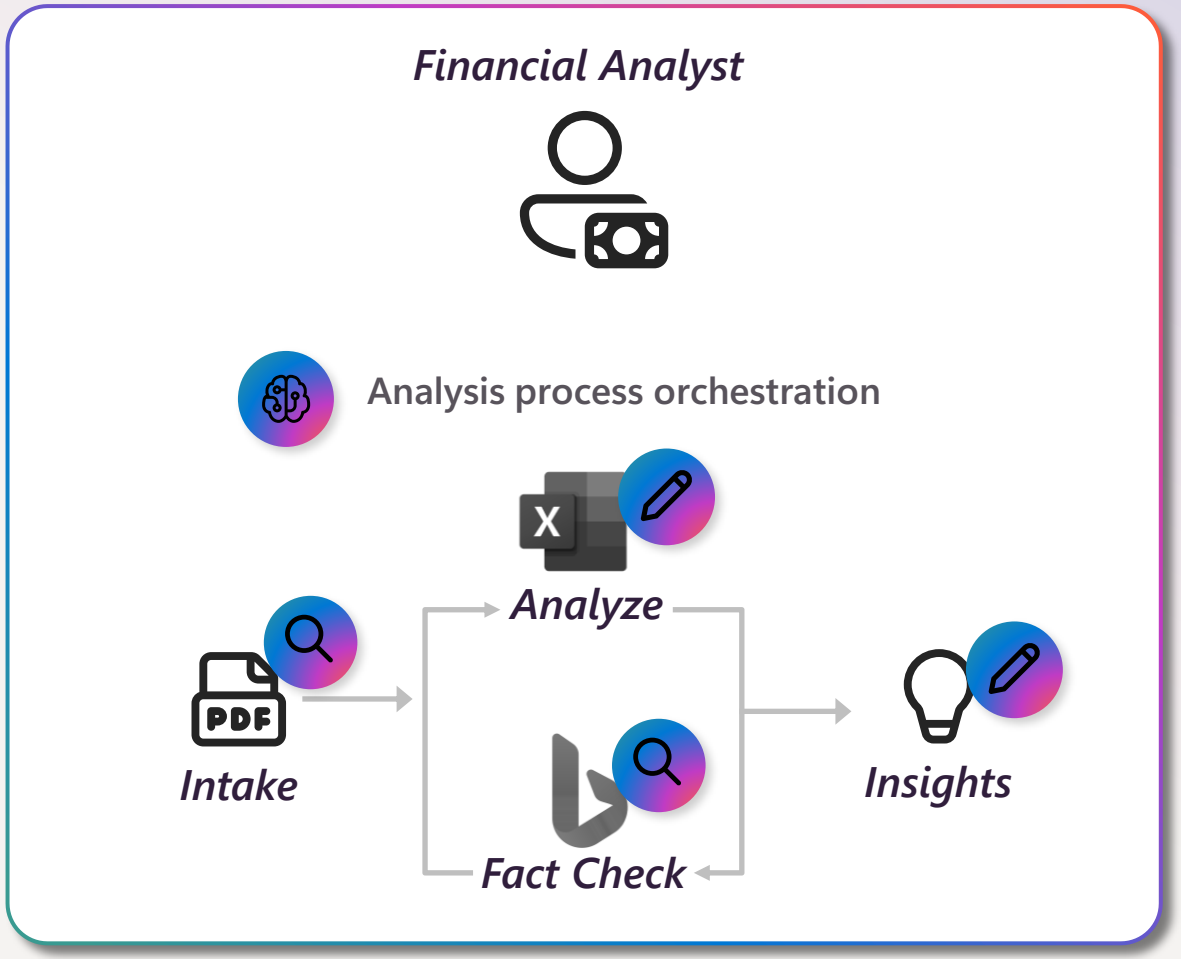
Image Generation using DALL-E

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

Multi-agent

What does an agent do?

- **Reason** over a provided business process
- **Retrieve** context to complete the process
- **Act** on behalf of the user



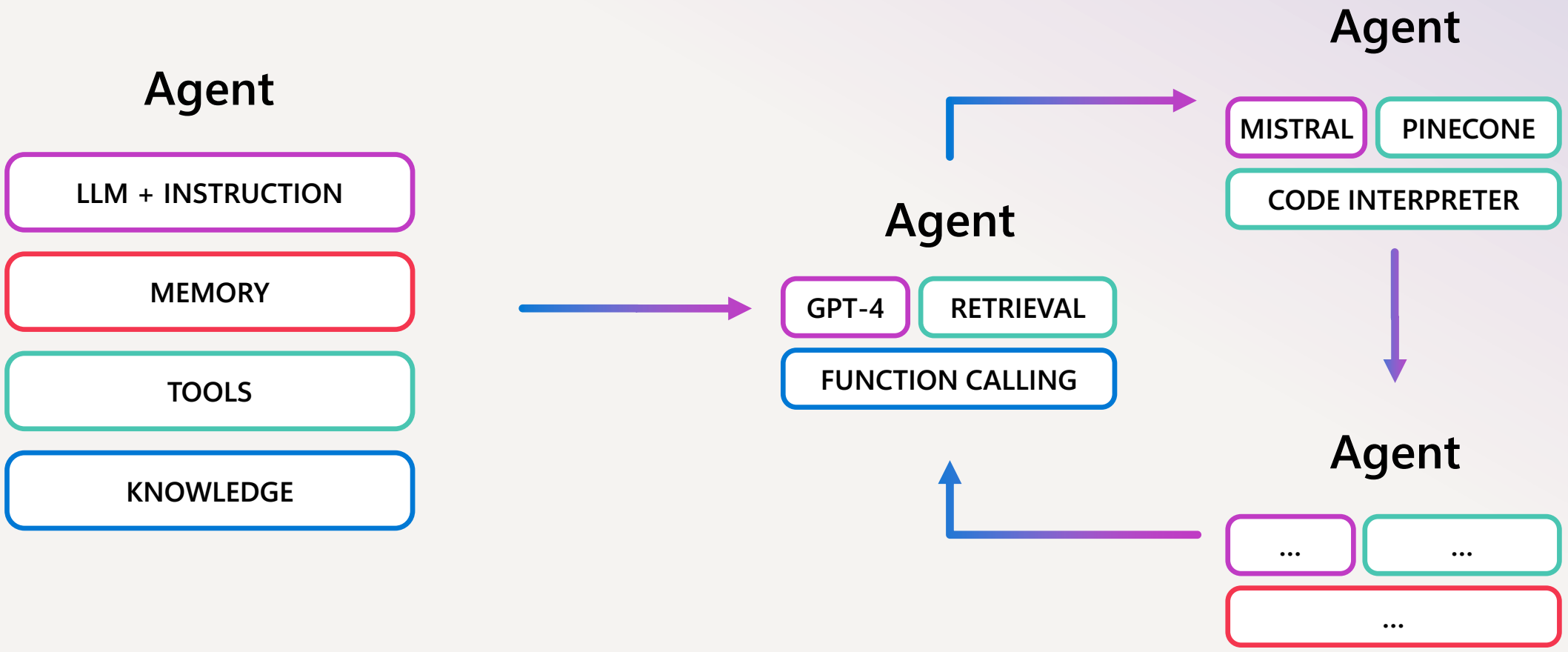
Chat vs. Agent frameworks

Chat	Agents
Lightweight and powerful	Stateful (inbuilt conversation state management)
Inherently stateless	Access persistent threads
	Access files in several formats. API handles chunking, embeddings, storage and creation, and implementing vector search
	Automatic management of the model's context window
	Access multiple tools in parallel (up to 128 tools per assistant) including code interpreter
	Build your own function calling

Customizing your AI Applications

	RAG	Function/API	Agent
Scenario Example	One well-defined task (Q&A) Known outcomes (answers) Looks for answers in product documentation	Multiple tasks Controlled outcomes (calls) Query an API to validate account number	Complex task Open outcomes Chain multiple apis to solve a problem
Models	Question answering (understanding/summarization)	Intent detection and planning	Multiple
Orchestration	Systematic workflow	Flexible Resilient (Code is the trigger)	Dynamic Self-healing (Agent is the trigger)

A single and multi-agent world



Challenge #8

Multi-Agent Systems

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

Microsoft Azure

Thank you