Build your own Copilot with Semantic Kernel & Azure OpenAI

Jakob Ehn
jakob.ehn@activesolution.se
@jakobehn

active
SOLUTION

Microsoft
Solutions Partner
Digital & App Innovation
Azure

Microsoft
Solutions Partner
Data & AI
Azure

MVP
Microsoft®
Most Valuable
Professional

AZURE
STOCKHOLM

YOU GET A COPILOT!
YOU GET A COPILOT!
AND YOU GET A COPILOT!
WE ALL GET A COPILOT!
imgflip.com

Copilot

Your everyday AI companion

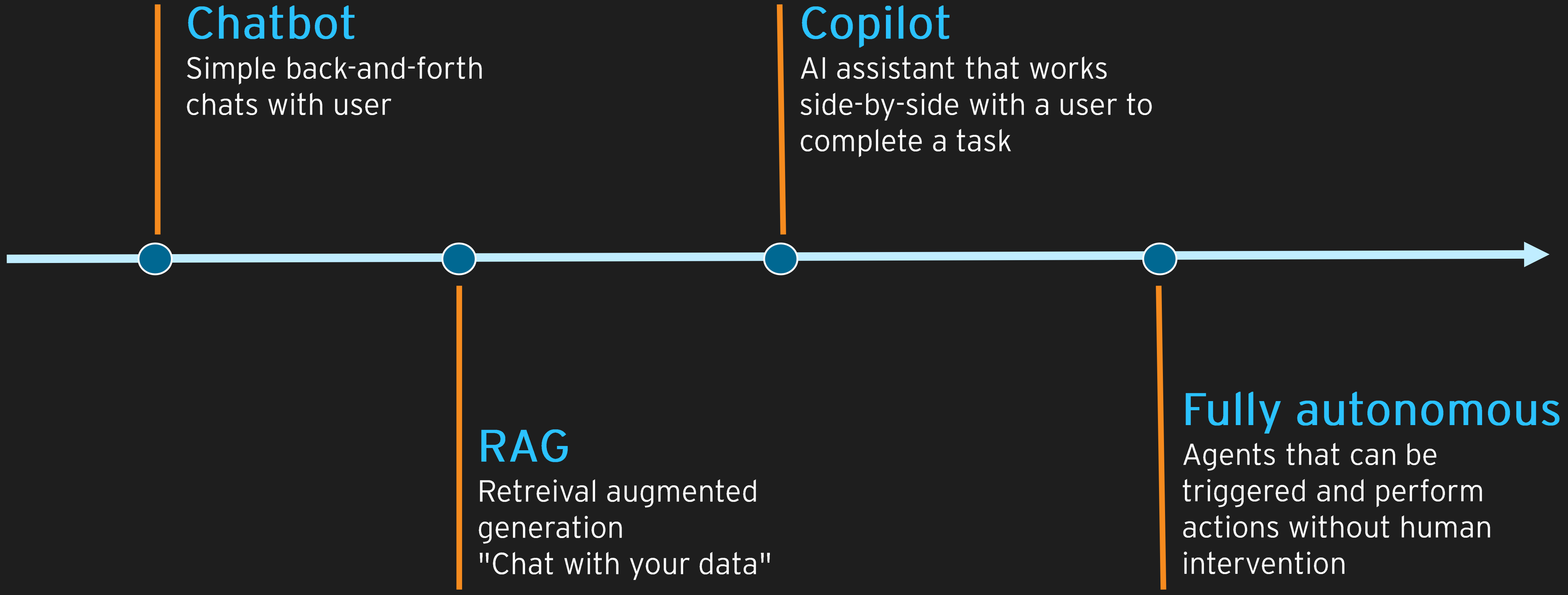
Copilot in Windows

Microsoft Copilot for Microsoft 365

Microsoft Copilot for Azure

Microsoft Security Copilot

Microsoft Dynamics 365 Copilot



Chatbot

Simple back-and-forth chats with user

RAG

Retreival augmented generation
"Chat with your data"

Copilot

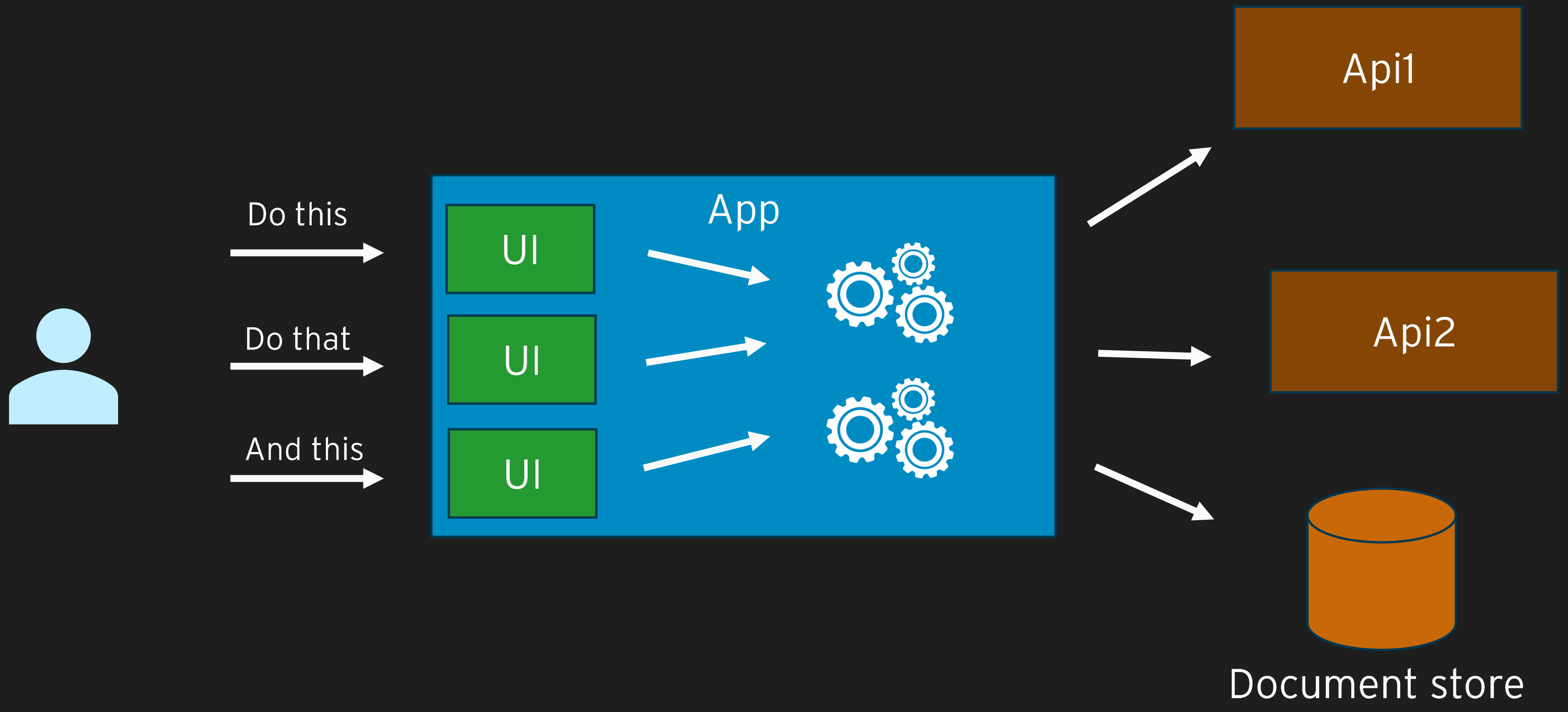
AI assistant that works side-by-side with a user to complete a task

Fully autonomous

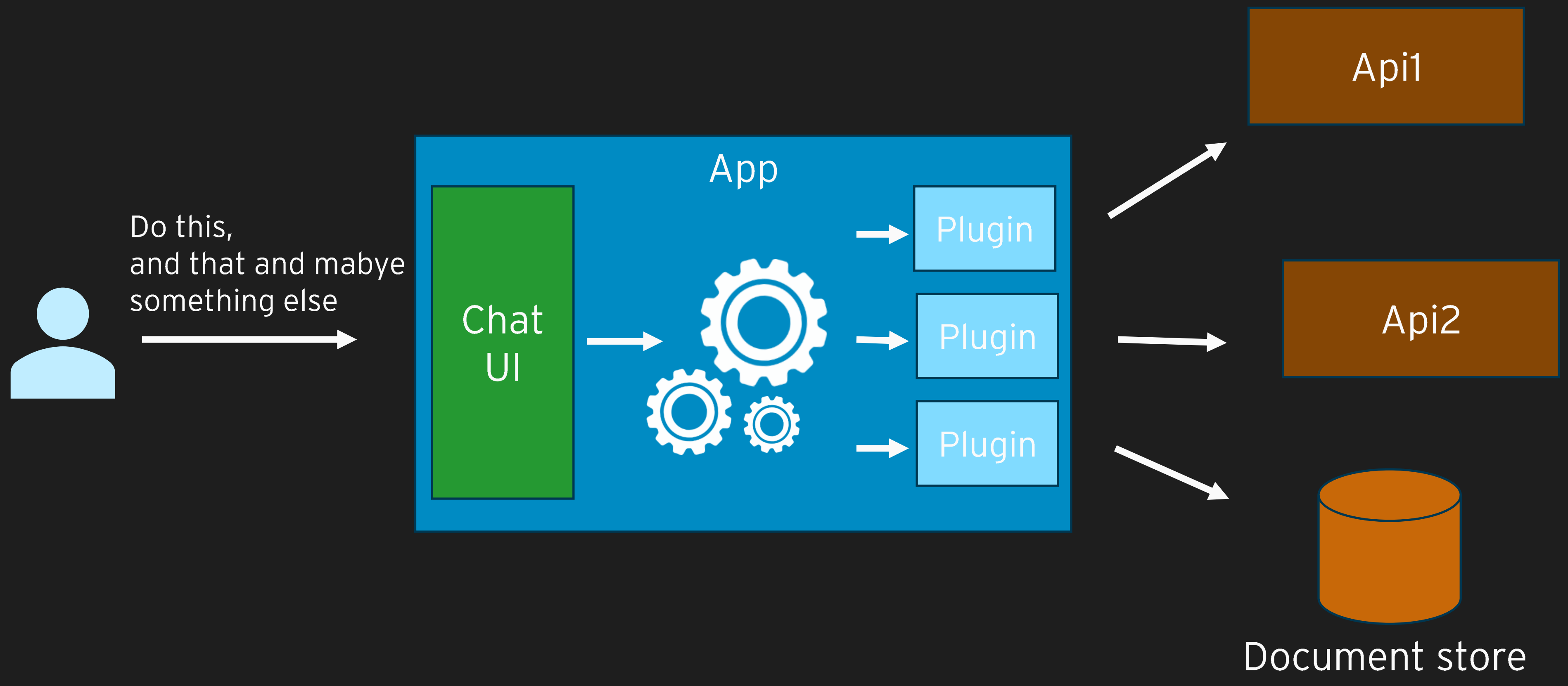
Agents that can be triggered and perform actions without human intervention

What is a Copilot?

Copilot apps



Copilot apps



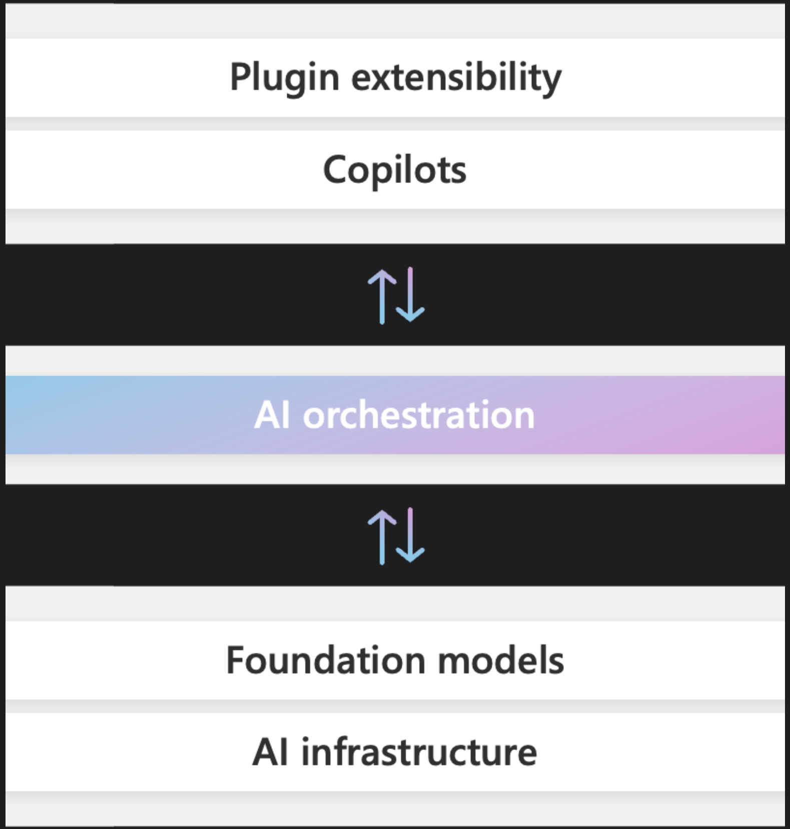
WE ARE AN AI STARTUP
API CALL TO GPT-3
imgflip.com

Dev challenges with LLMs

- Many services and models with various API:s
- Dealing with context, history and hallucinations
- No access to external data
- Hard things are easy, easy things can be hard
- Combine native code with LLMs

Semantic Kernel

- Open-source AI orchestration library (.NET + Python)
- Combine LLMs with convential programming languages
- Extensible through plugins for custom code/prompts
- Core component in the Copilot architecture



Semantic Kernel

Demo

Plugins

Planners

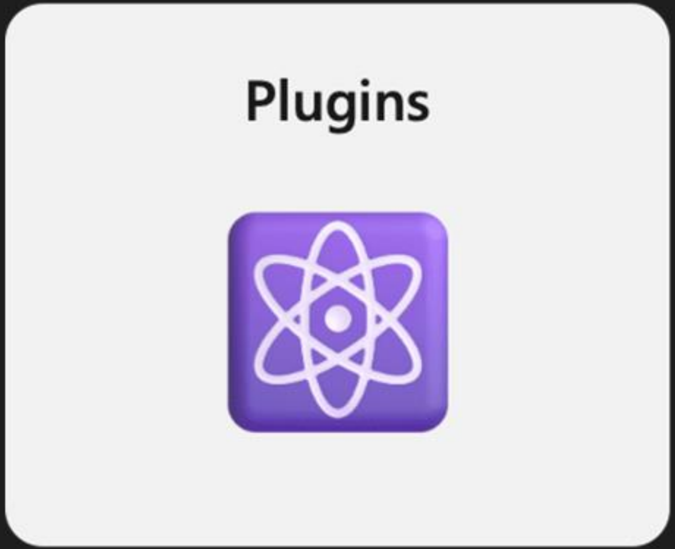
Personas

Plugins

- Fundamental building block
- Can be native or prompt-based
- Do things in the real world

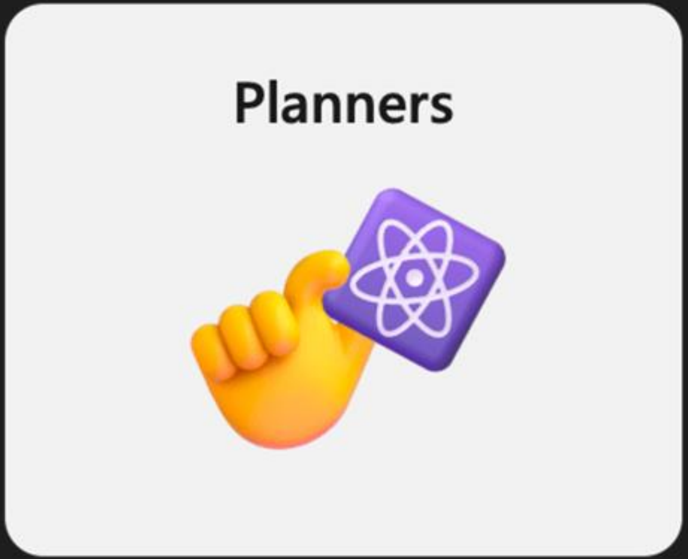
Planners

Personas



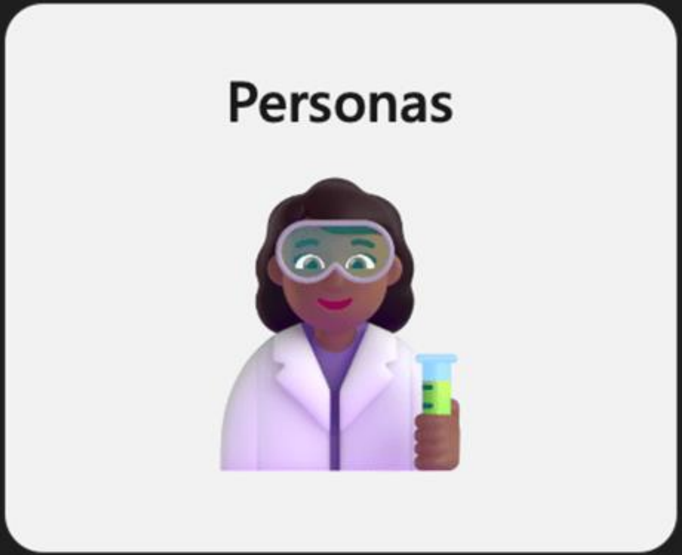
Plugins

- Fundamental building block
- Can be native or prompt-based
- Do things in the real world



Planners

- Given a user prompt, create a plan on how to accomplish the request



Personas

Plugins

- Fundamental building block
- Can be native or prompt-based
- Do things in the real world

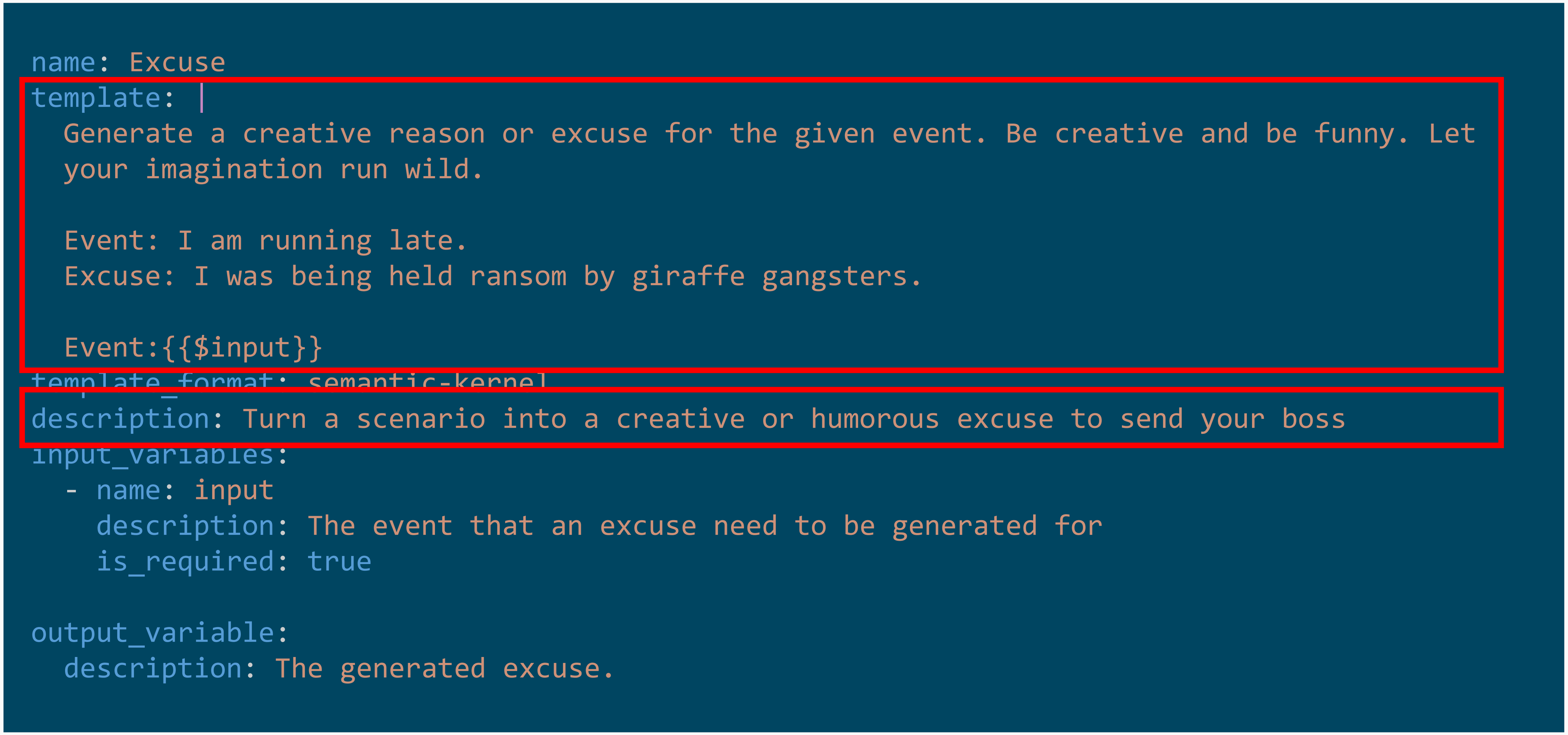
Planners

- Given a user prompt, create a plan on how to accomplish the request

Personas

- Set "personality" and tone of the responses

Prompt-based Plugins (YAML)



```yaml
name: Excuse
template: |
  Generate a creative reason or excuse for the given event. Be creative and be funny. Let
  your imagination run wild.

  Event: I am running late.
  Excuse: I was being held ransom by giraffe gangsters.

  Event:{{$input}}
template_format: semantic-kernel
description: Turn a scenario into a creative or humorous excuse to send your boss
input_variables:
  - name: input
    description: The event that an excuse need to be generated for
    is_required: true

output_variable:
  description: The generated excuse.
```

Native Plugins - C#

```csharp
[KernelFunction, Description("Take the square root of a number")]
public static double Sqrt(
    [Description("The number to take a square root of")] double number)
{
   return Math.Sqrt(number);
}
```

```csharp
//Import plugin
kernel.ImportPluginFromType<MathPlugin>("MathPlugin");
```

Planners

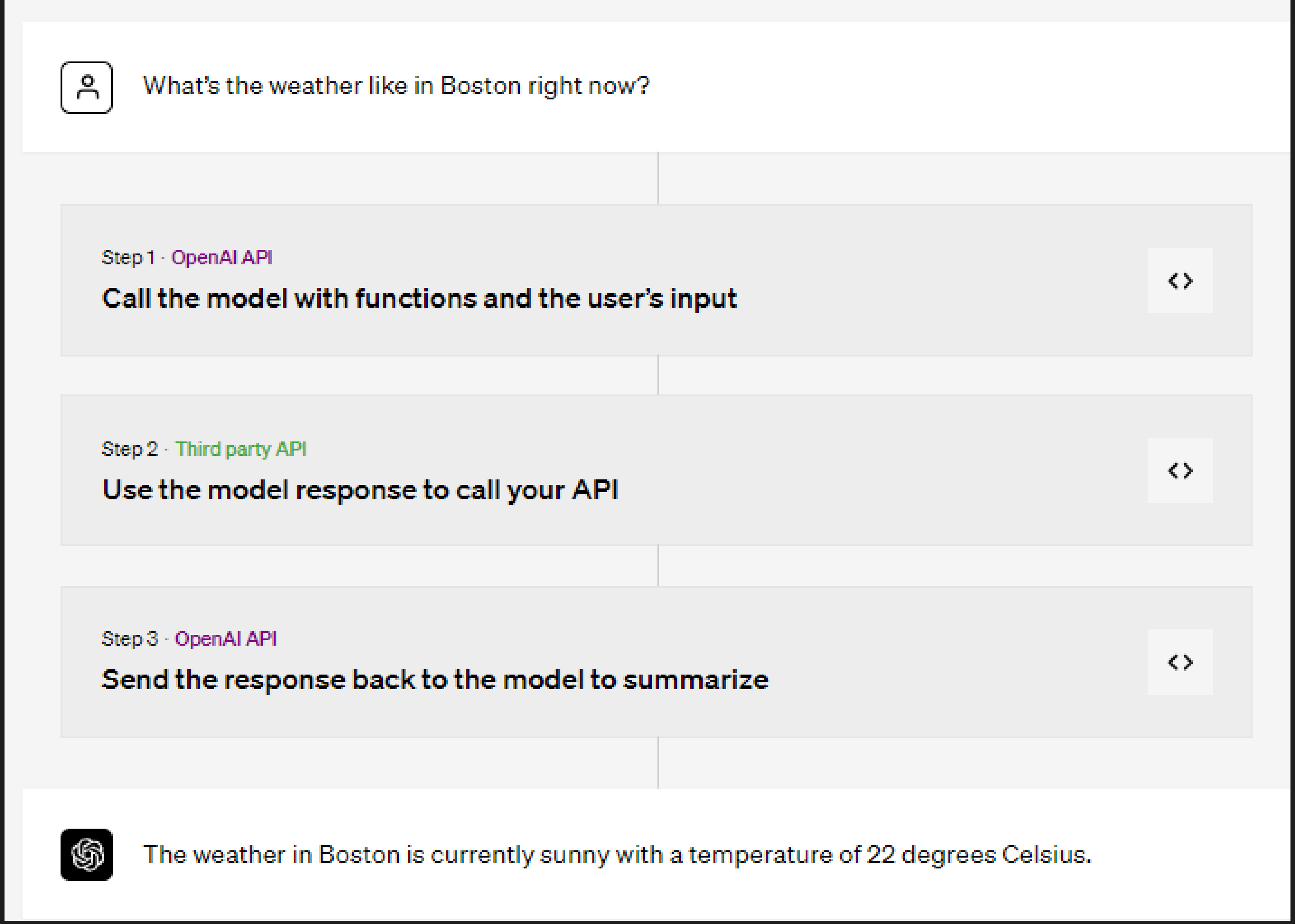
Goal

Given a user prompt, create a plan on how to accomplish the request

Semantic Kernel offers multiple alternatives:

- OpenAI function calling
- HandlebarsPlanner
- FunctionStepwisePlanner

Function Calling in GPT models



What's the weather like in Boston right now?

Step 1 · OpenAI API

Call the model with functions and the user's input

Step 2 · Third party API

Use the model response to call your API

Step 3 · OpenAI API

Send the response back to the model to summarize

The weather in Boston is currently sunny with a temperature of 22 degrees Celsius.

Function Calling - Semantic Kernel

```
kernel.ImportPluginFromType<MathPlugin>("MathPlugin");
kernel.ImportPluginFromType<SummarizePlugin>("SummarizePlugin");

var settings = new OpenAIPromptExecutionSettings()
{
   ToolCallBehavior = ToolCallBehavior.AutoInvokeKernelFunctions
};

var result = await kernel.InvokePromptAsync<string>("query", new KernelArguments(settings));
```

Handlebars planner

- Can generate an entire plan using a single LLM call
- Internally uses the Handlebars template language

- You can inspect the entire plan before executing it
- Good plans can be saved and reused, without having to regenerate them
- Uses fewer tokens, since it's only one LLM call

Handlebars planner

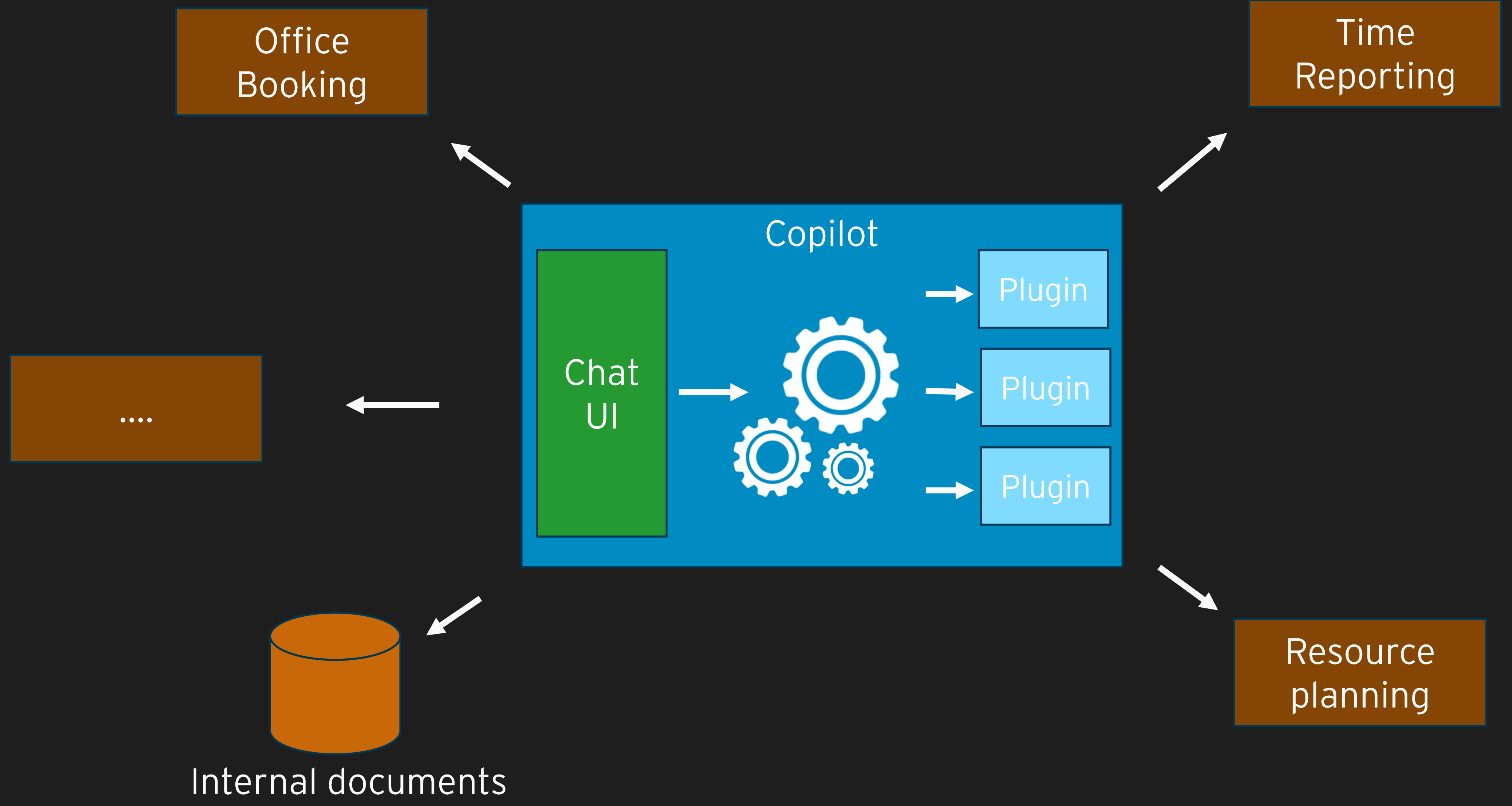
```
var planner = new HandlebarsPlanner();

var ask = @"Get a list of all programs about music that has been aired today on
            radio today. List the programs with name, titel, audiourl and
            websiteurl. Save the list of programs in a file called episodes.txt.
            Then return the list of programs back to the user.";

var plan = await planner.CreatePlanAsync(kernel, ask);

//Print or save the plan
Console.WriteLine(plan.ToString());

var result = await plan.InvokeAsync(kernel);
```

Semantic Kernel Planners

Demo

Copilot

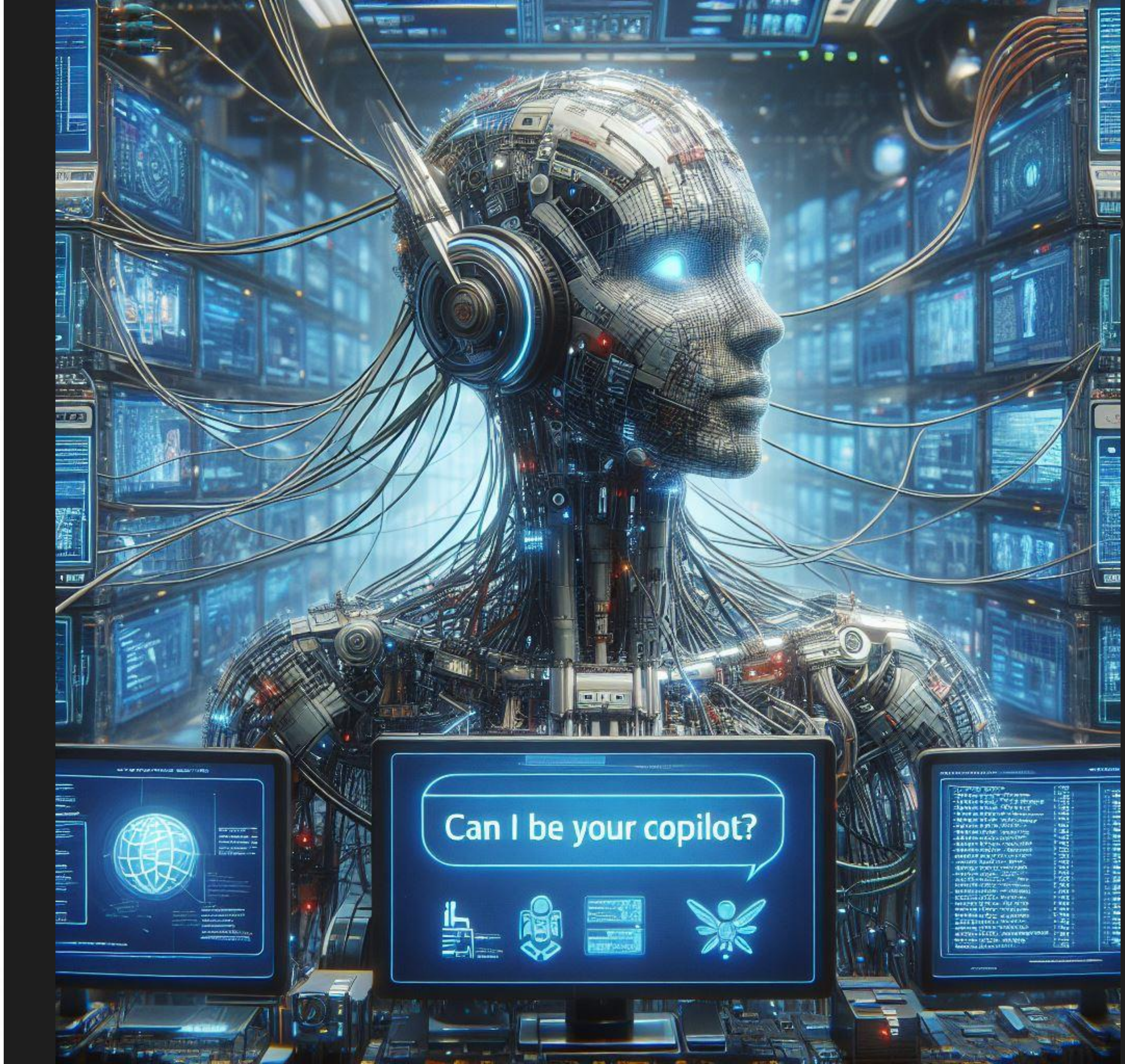
Active Copilot



Copilot with Semantic Kernel

Demo

YOU GET A COPILOT!
YOU GET A COPILOT!
AND YOU GET A COPILOT!
WE ALL GET A COPILOT!
imgflip.com


Can I be your copilot?

Thank You!

Jakob Ehn
jakob.ehn@activesolution.se
@jakobehn

Extra Slides

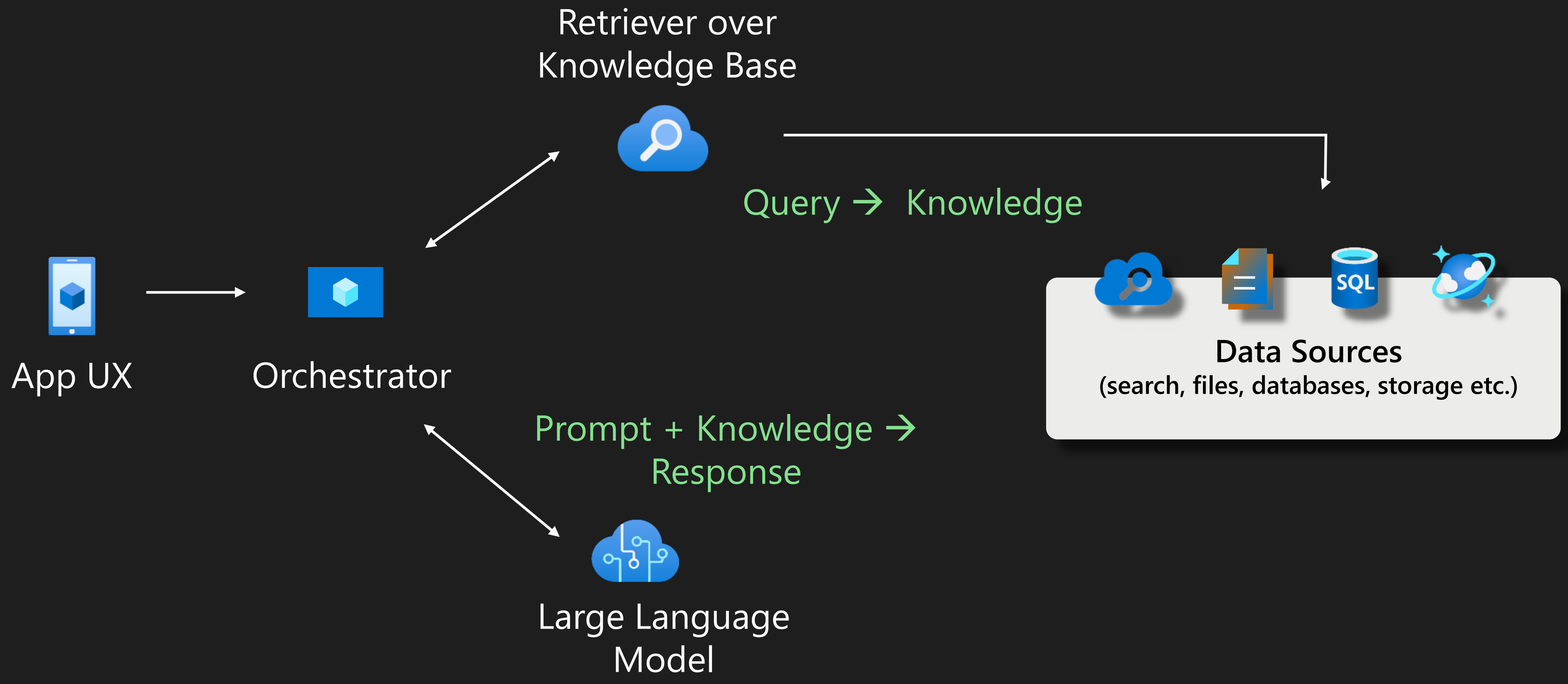
Links

- Semantic Kernel
  https://github.com/microsoft/semantic-kernel

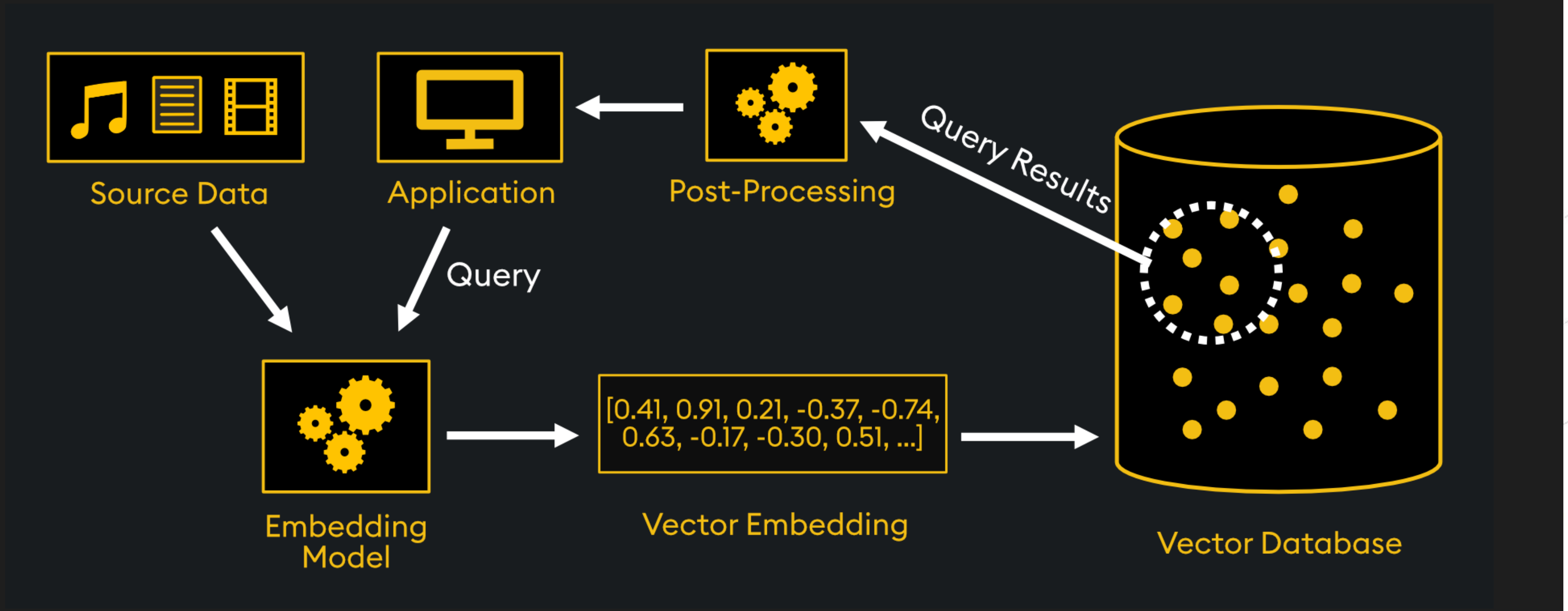
- Kernel Memory
  https://github.com/microsoft/kernel-memory

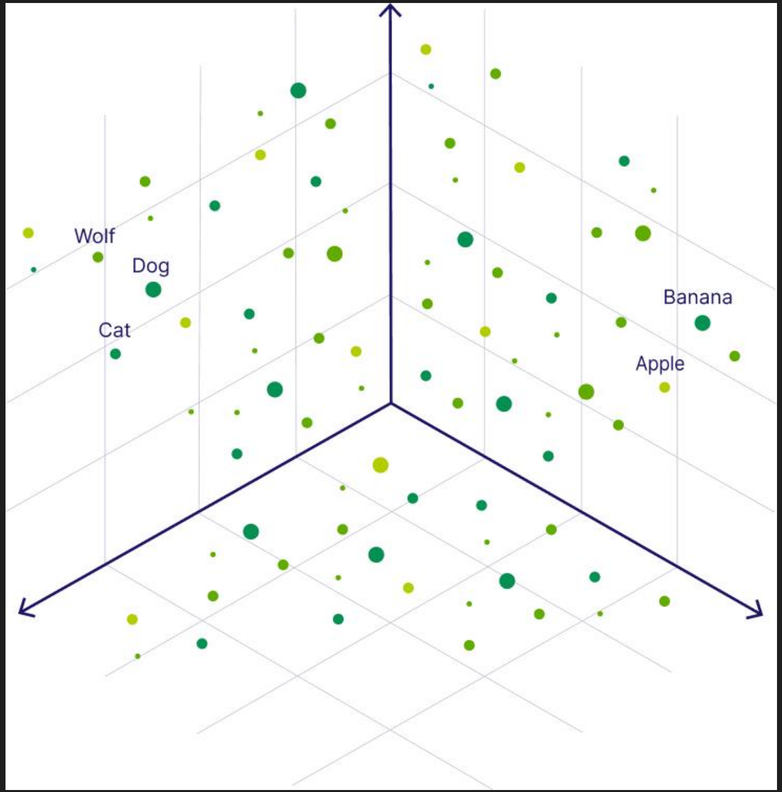
- Azure OpenAI Service (& link to apply)
  https://azure.microsoft.com/en-us/products/ai-services/openai-service

Retrieval Augmented Generation (RAG)



Vector Databases





Stepwise planner

- Sometimes, regular function calling is not enough
- Built on top of function calling
- Uses additional reasoning when generating the plan, to improve reliability

```
You are an expert at generating plans from a given GOAL.
Think step by step and determine a plan to satisfy the specified GOAL using only the FUNCTIONS provided
to you. You can also make use of your own knowledge while forming an answer but you must not use
functions that are not provided.
Once you have come to a final answer, use the UserInteraction_SendFinalAnswer function to communicate
this back to the user.

[FUNCTIONS]
{{$available_functions}}
[END FUNCTIONS]

To create the plan, follow these steps:
0. Each step should be something that is capable of being done by the list of available functions.
1. Steps can use output from one or more previous steps as input, if appropriate.
2. The plan should be as short as possible.
```

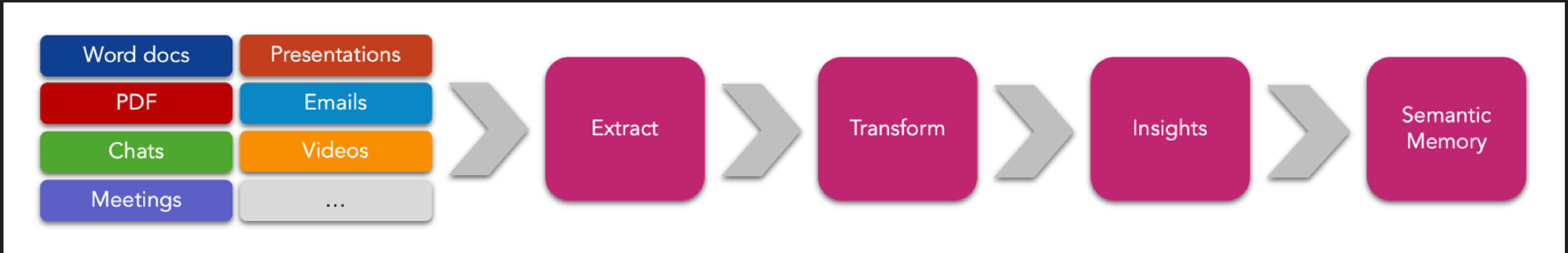
Stepwise planner usage

```
kernel.ImportPluginFromType<MathPlugin>("MathPlugin");

var planner = new FunctionCallingStepwisePlanner();

var ask = @"If I buy 16 apples, then eat half of them and then give another
            three apples to a friend, how many apples do I have left?";

var result = await planner.ExecuteAsync(kernel, ask);
```

- Executed plan available after execution

Kernel Memory

- Store and retreive information in vector databases
- Used by LLMs during prompt generation
- Simplifies RAG



RAG with Kernel Memory

- Convert question into user intent (using text generation model)
- Perform semantic search on the vector database
- Combine the documents retrieved and adds them to the user prompt
- Submit prompt to LLM and it return the response
  - Including the relevant sources/citations

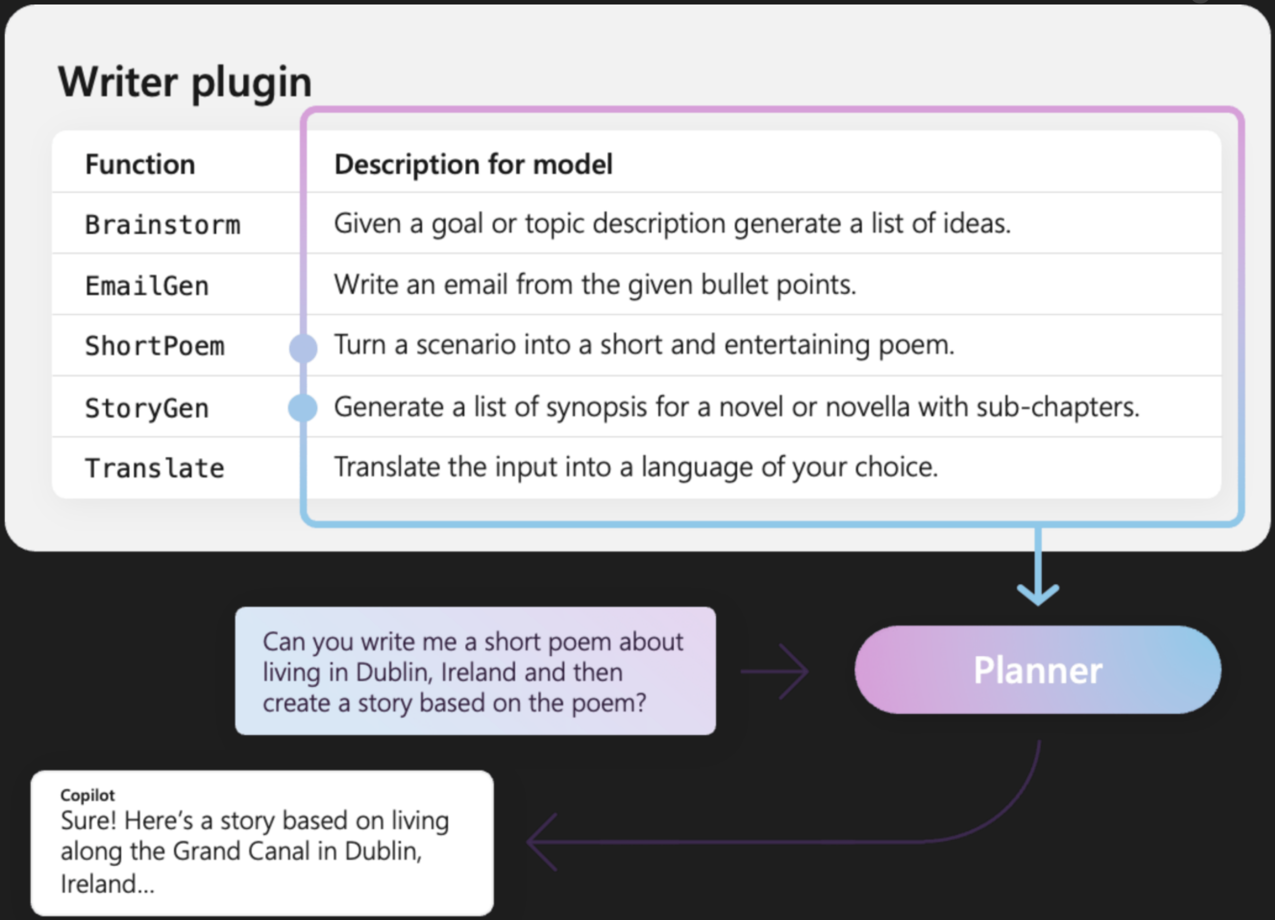
```
//Find (semantically) matching content
var result = await kernelMemory.SearchAsync(query)

//Answer a question based on matching content
var answer = await kernelMemory.AskAsync(question)
```

Semantic Kernel Memory

Demo

Plugins

- Fundamental building block
- Can be native or prompt-based
- Retreive data from external data sources
- Completing tasks in the real world
- Can be invoked explicitly or automatically



Writer plugin

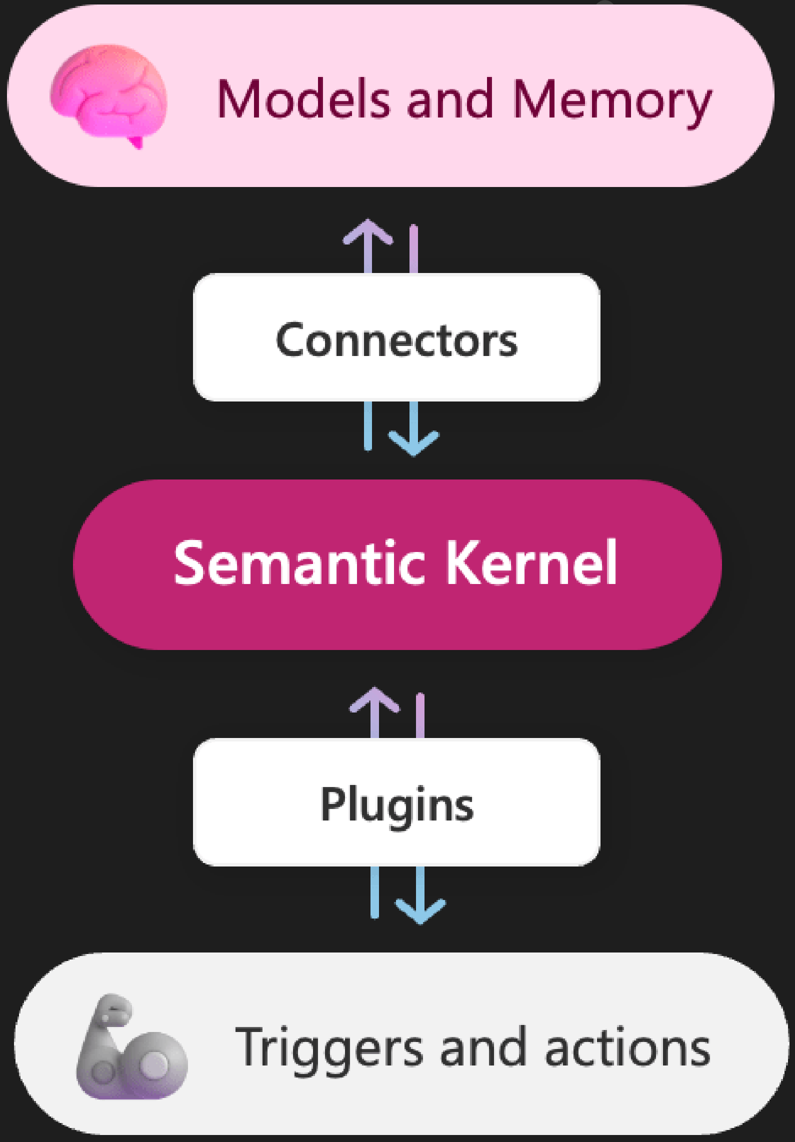
Function	Description for model
`Brainstorm`	Given a goal or topic description generate a list of ideas.
`EmailGen`	Write an email from the given bullet points.
`ShortPoem`	Turn a scenario into a short and entertaining poem.
`StoryGen`	Generate a list of synopsis for a novel or novella with sub-chapters.
`Translate`	Translate the input into a language of your choice.

Can you write me a short poem about living in Dublin, Ireland and then create a story based on the poem?

Copilot
Sure! Here's a story based on living along the Grand Canal in Dublin, Ireland...

Semantic Kernel

- Integrate any AI service/models
- Plugins for custom code/prompts
- Automatic orchestration with planners
- Store/Read contextual information in memory


Models and Memory
Connectors
Semantic Kernel
Plugins
Triggers and actions