

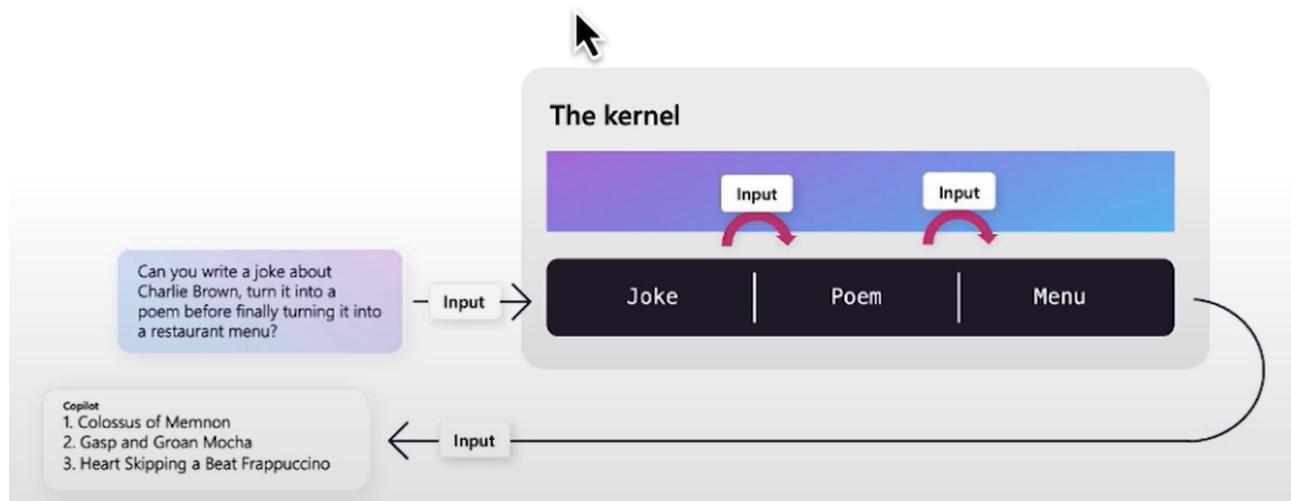
How Business Thinkers Can Start Building AI Plugins With Semantic Kernel

Semantic Kernel is like your AI cooking kitchen



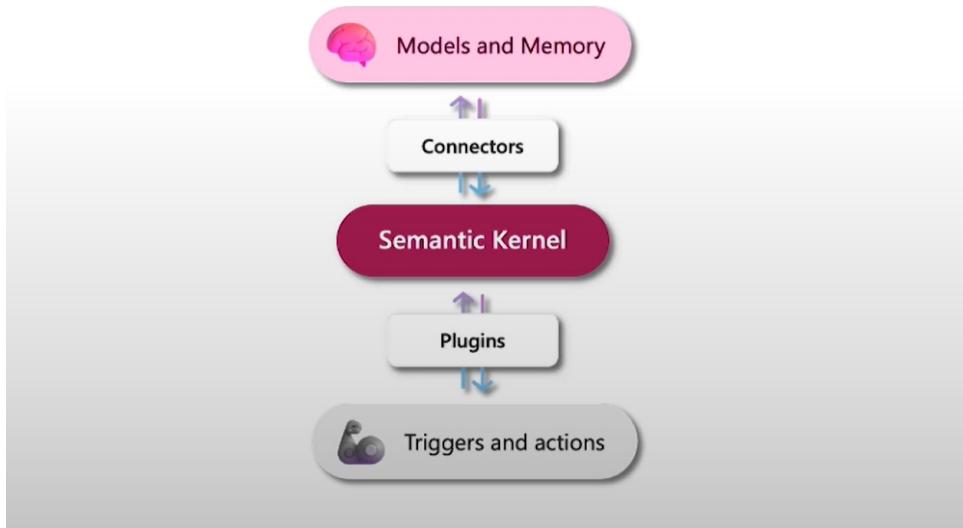
What's a “Kernel”?

Semantic Kernel was inspired by the UNIX kernel – specifically in how it was written to be small, mighty, and fully extensible. We've gotten nice compliments for its technical architecture, and that's going to be more obvious to you as a .NET/C# dev.



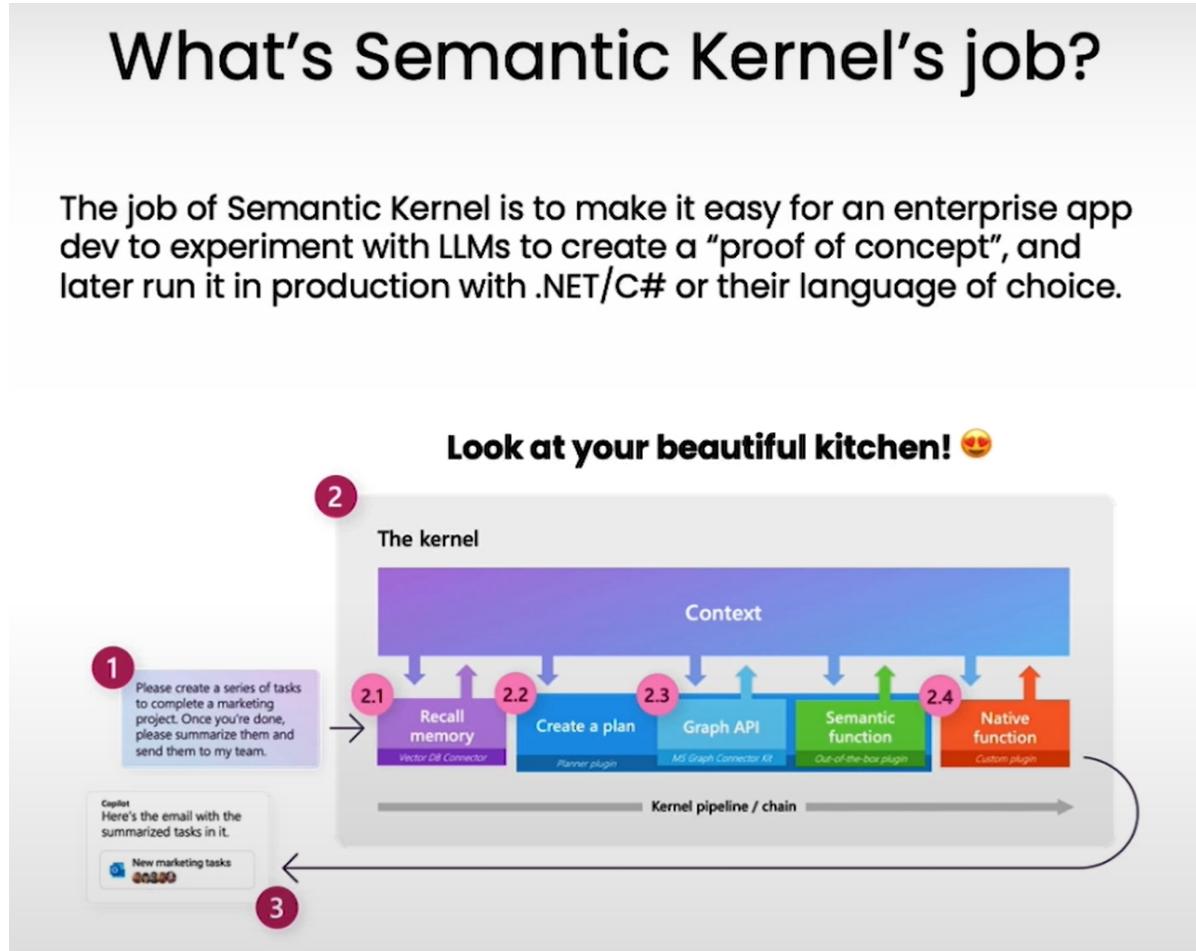
Why is it "Semantic"?

The founding team of Semantic Kernel encountered GPT-4 in the summer of 2022. With a focus on making it easier for app devs to work with LLMs, vector databases, and prompt templates, the word "semantic" helped to signal a shift from traditional native coding.



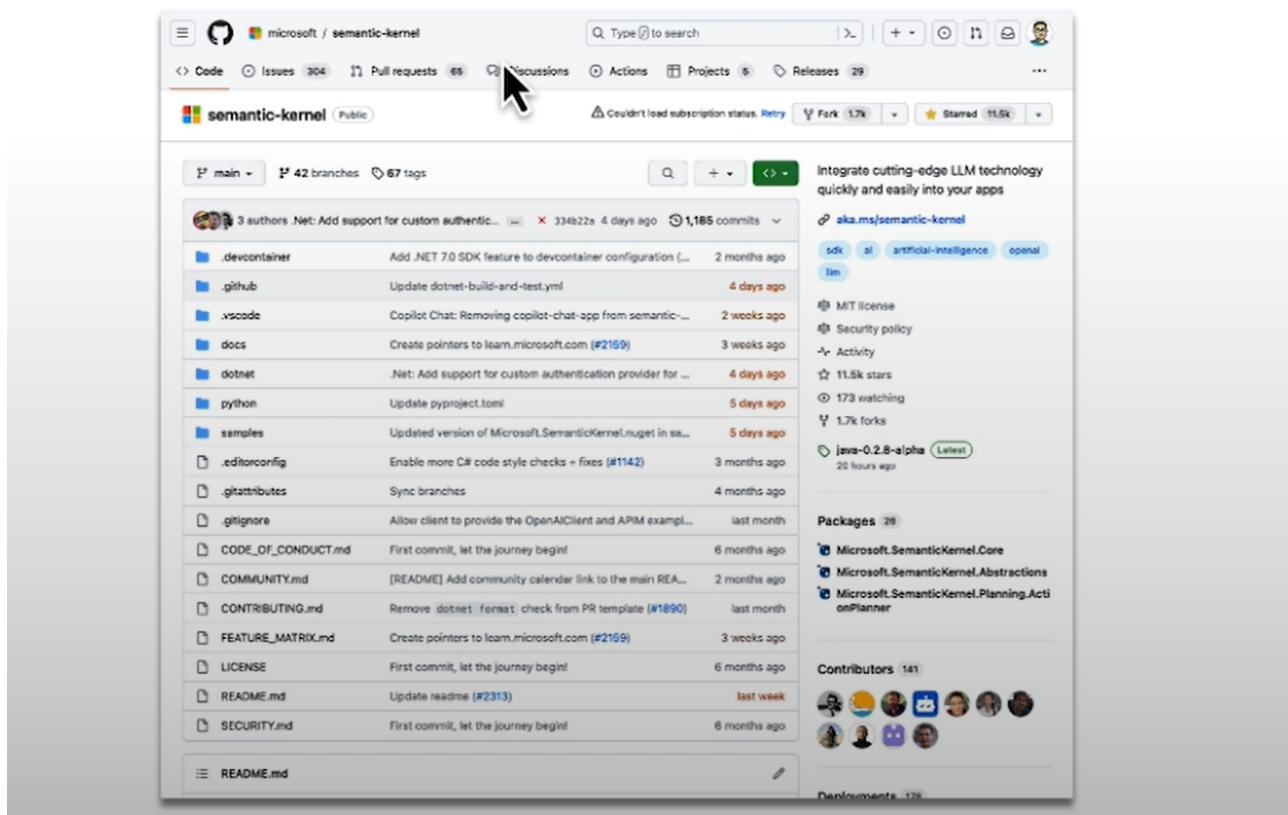
What's Semantic Kernel's job?

The job of Semantic Kernel is to make it easy for an enterprise app dev to experiment with LLMs to create a “proof of concept”, and later run it in production with .NET/C# or their language of choice.



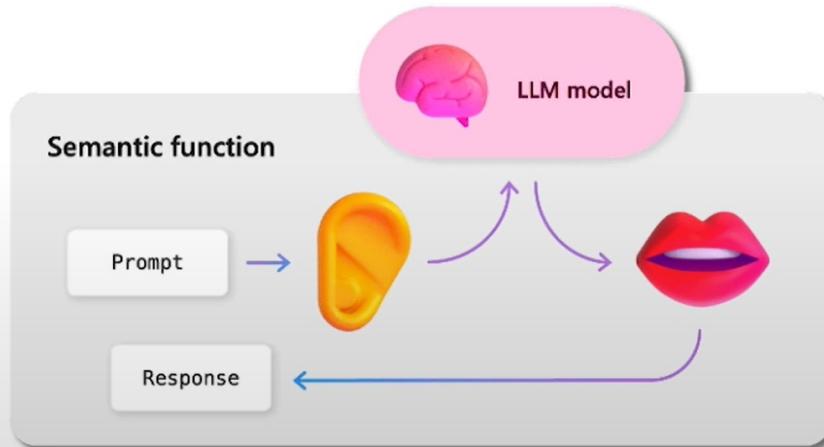
Semantic Kernel is Open-Source

In the open-source world of SDKs for LLMs, Semantic Kernel is the option that is the most enterprise-y and “no frills.” Many of the available code examples demonstrate how to dock into the universe of Azure AI components for things like AI guardrails, prompt tuning, and vector databases. .NET/C# and Python are supported with branches available in TypeScript and Java.



What's a Semantic Function?

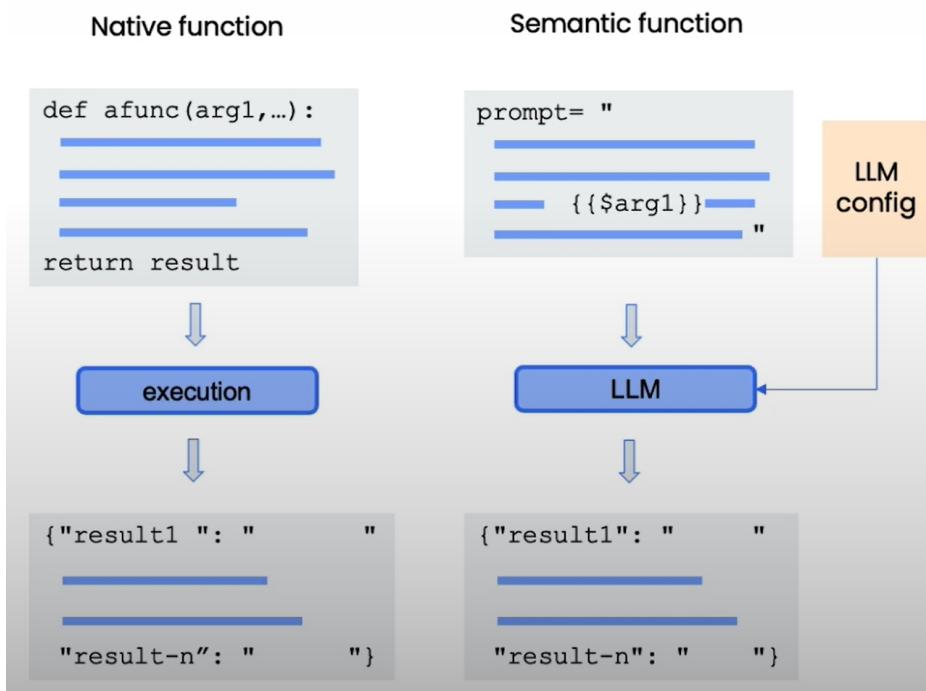
Semantic functions are encapsulations of repeatable LLM prompts that are orchestrated by the kernel. Functions are the building blocks of working productively with LLMs. Non-programmers, like you, are poised to discover how to use them for your own benefit.



💡 Collections of functions are referred to as “Skills” or “Plugins” interchangeably – with the primary intention of seeing functions mature into AI Plugins that can be used in other applications.

Native vs Semantic Function?

Semantic Kernel orchestrates any mixture of native functions, semantic functions, or hybrid native/semantic functions. These functions can be written inline or in a handy, text-only file format.



What's a SWOT?

A SWOT analysis is one of the most basic business strategy tools in the book. It's quite simple to work with — and many would argue that it's oversimplistic. But that's okay because we're just here to learn, and cook, together.

Strengths

Weaknesses

Opportunities

Threats

L3

How Business Thinkers Can Start Building AI Plugins With Semantic Kernel

Organizing the tools you make for later reuse



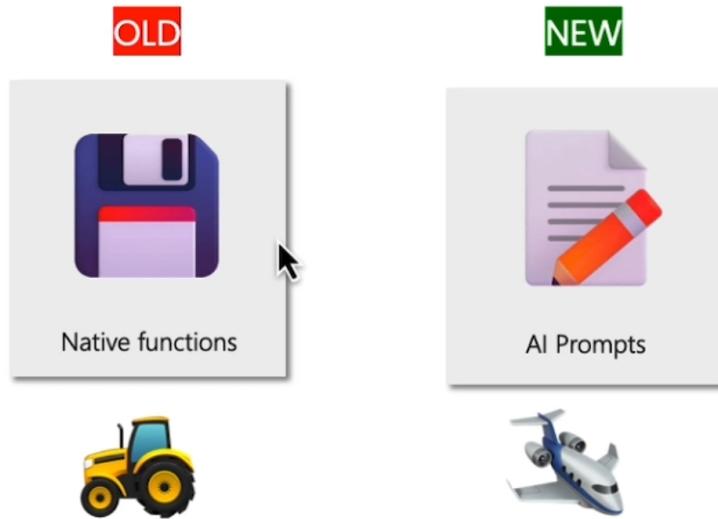
Microsoft



DeepLearning.AI

What's the big switch?

Native functions are conventional coding at-work. And semantic functions are special because they're powered by words that we already use. **They have unusually high leverage**. And they're in your vocabulary instead of the computer's.



Package a semantic function

In the same way that native functions can be stored in their own code files, semantic functions can be packaged for easy reuse, too. You'll need to make a folder with two files: 1/ your semantic function, and 2/ a configuration file in JSON format. Both are text files that you can open and edit in a regular text editor.

skprompt.txt

```
The following are anonymized  
comments from customers:  
---  
{{\$input}}  
---  
Isolate the five most common  
sentiments from customer feedback.  
The five common sentiments are:
```

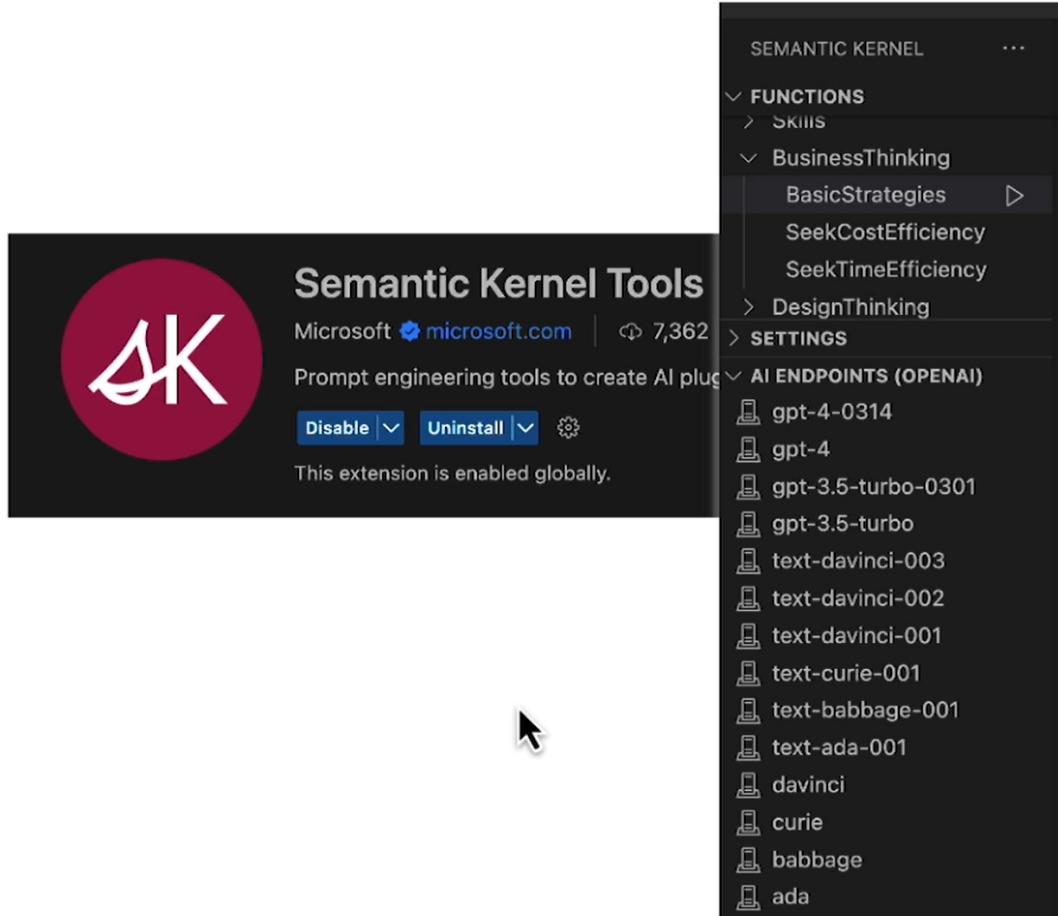


config.json

```
{  
  "schema": 1,  
  "description": "Empathize",  
  "type": "completion",  
  "completion": {  
    "max_tokens": 1000,  
    "temperature": 0.9,  
    "top_p": 0.5,  
    "presence_penalty": 0.0,  
    "frequency_penalty": 0.0  
  },  
  "input": {  
    "parameters": [  
      {  
        "name": "input",  
        "description": "Feedback",  
        "defaultValue": ""  
      }  
    ]  
  }  
}
```

There's a tool for your tools

We can edit semantic functions inline or from within a text editor. But the preferred method is to use the Visual Studio Code extension.

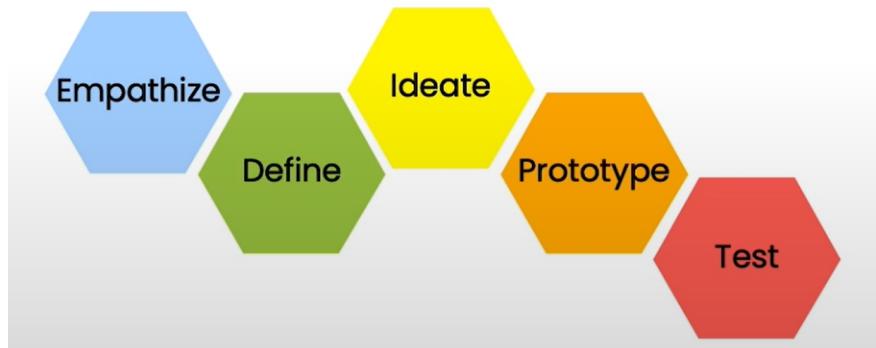


L4

The slide has a light gray background. At the top, the title reads 'How Business Thinkers Can Start Building AI Plugins With Semantic Kernel'. Below the title, the text 'Frozen dinner: The design thinking meal' is displayed. At the bottom left is the Microsoft logo. At the bottom right is the DeepLearning.AI logo, which features a circular emblem with three concentric circles and the text 'DeepLearning.AI'.

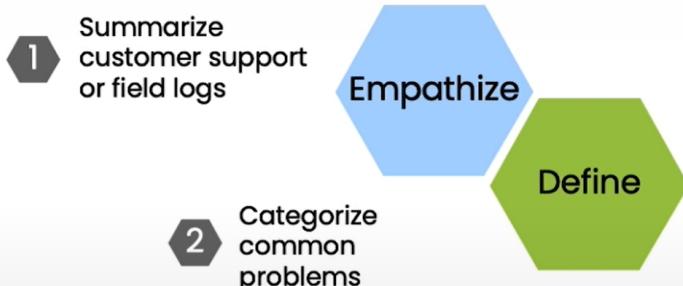
What's Design Thinking?

Design Thinking is a structured process for innovation that places the user or customer at the center of determining how to invent solutions for their needs. It originated at Stanford University and since its introduction in Harvard Business Review, it has become a favored methodology for co-innovating new features or products.



Design Thinking as a plugin?

It's reasonable to think of LLMs as a means to accelerate the design thinking process. We can demonstrate how that works with a couple of semantic functions for the Empathize and Define phases.



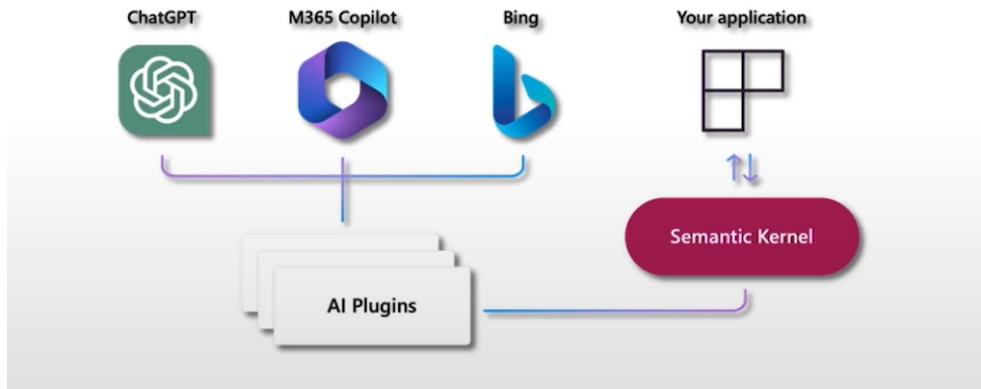
```
plugins-sk/
└── DesignThinking/
    ├── Define/
    │   └── config.json
    │   └── skprompt.txt
    └── Empathize/
        └── config.json
        └── skprompt.txt
```

These two phases are written as semantic functions in the folder

`./plugins-sk/DesignThinking`

Why are plugins so important?

In the Semantic Kernel world, functions are the first building block to get the kernel working for you. With semantic and native functions, it's easy to create AI Plugins to be used anywhere.



L5

How Business Thinkers Can Start Building AI Plugins With Semantic Kernel

Don't forget to save the generated drippings, or "the gravy"



How Business Thinkers Can Start Building AI Plugins With Semantic Kernel

A kitchen that responds to your “I’m hungry” is more than feasible



Two Buckets: Time and Money

The small business owner is said to be carrying two buckets that are always leaking one of two things: time and money. They can't afford to think deeply how to fix their business; they just wish to make a little more money or save a little more time each day.





LLMs can fill in any ____.

As LLMs get more powerful, they improve at completing a sentence, paragraph, page, etc in a useful manner. Thus it makes sense that one could say, "I wish I was \$10 richer. I will need to..."



"I wish I was \$10 richer. I will need to _____.

But this is the ideal set up for a 🤖 hallucination to occur. There is a lot of missing context for the COMPLETION to be usable. We need to do three things to steer the prompt in the right direction:

1. RAG this prompt by reaching into the pile of business we have stored in memory, and find advice that is 💡 similar to "making more money." Attach this context to the prompt.
2. Much of the advice we've generated involves writing appropriate marketing copy or sending out surveys to gather data. Prepare a few plugin functions for doing those tasks.



Remember the descriptions?

Every function has a written description for a human to understand what it does. This is especially useful if you ever needed to find a function that fit your needs. But with the SIMILARITY engine, we can use those descriptions to search our toolbox for the right tool.

"I need to translate text"



Writer plugin

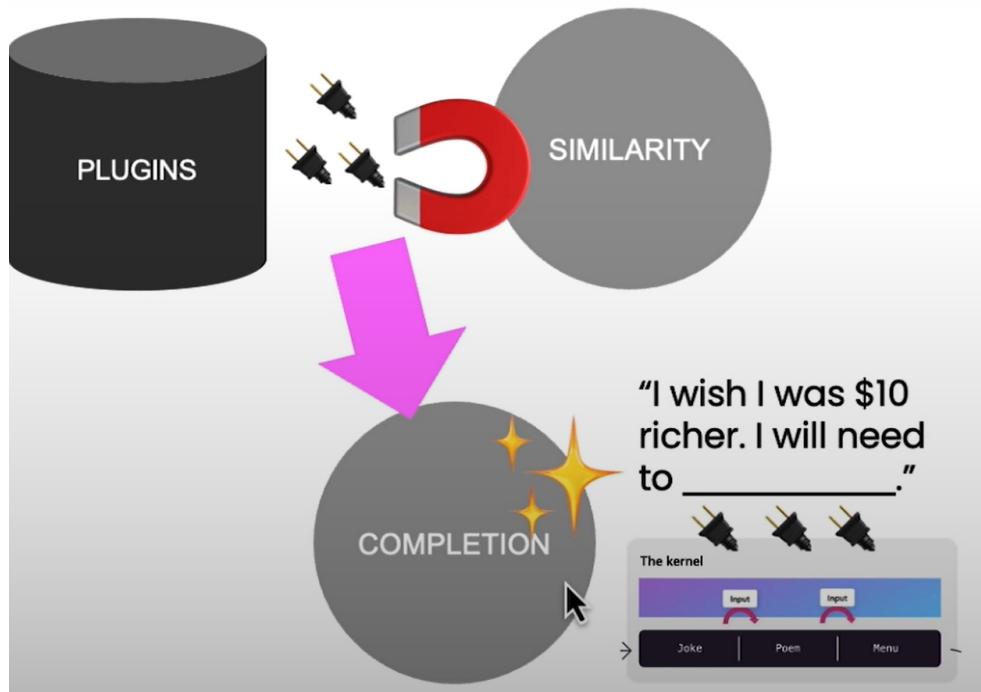
Function	Description for model
Brainstorm	Given a goal or topic description generate a list of ideas.
EmailGen	Write an email from the given bullet points.
ShortPoem	Turn a scenario into a short and entertaining poem.
StoryGen	Generate a list of synopsis for a novel or novella with sub-chapters.
Translate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

Which means the third step should be:

3. In the prompt include all the 💡 relevant Plugin functions' descriptions so that the ⚡ COMPLETION will know which tools are available and how to achieve the overall ask.

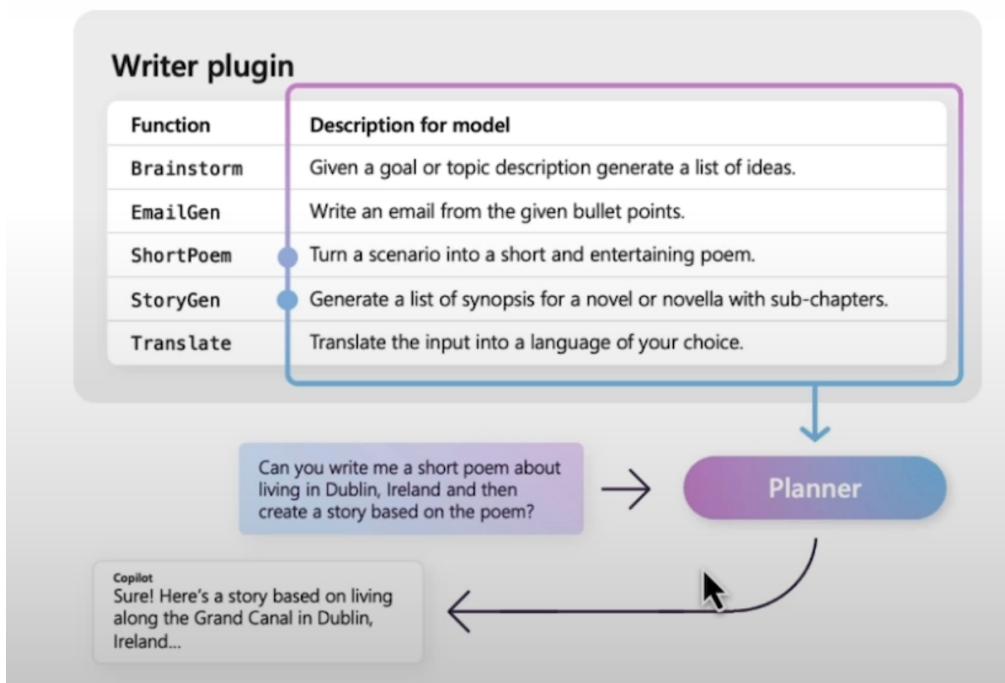
There's more than chat to this AI

Once you have enough plugins that are neatly labeled and packaged, it becomes possible to find the plugins that match a certain need or intent. And when laid out in a **plan of action**, your wish may come true. Because all the tools needed are put to use.



SK ❤️ Plugins & Planners

AI Plugins are specifically designed to be accessed by AI Planners so that a high-level goal can be the starting point. And given the right set of Plugins, a logical Plan is generated to achieve the goal.



Plugins & Planners & Personas

L7

Plugins & Planners & Personas

Open-Source Chat Copilot

The screenshot shows the Chat Copilot application interface. At the top, there's a dark header bar with the title "Chat Copilot" on the left, a "Plugins" button with a gear icon, and a user profile icon on the right. Below the header is a navigation bar with tabs: "Chat" (which is active, indicated by a red underline), "Documents", "Plans", and "Personas".

The main area is a chat window. On the left, there's a sidebar with a "+" button and a "Today" section containing three messages:

- A yellow circular icon followed by the text "What is the tallest b...".
- A red circular icon followed by the text "Where is the world?".
- A red circular icon followed by the text "How is my day today?".

The main chat area shows a conversation between the user ("JM") and the AI ("Copilot"). The AI's first message is timestamped at 8:07 AM: "Hello, thank you for democratizing AI's productivity benefits with open source! How can I help you today?". The user responds at 8:08 AM: "What is the tallest building that you know about?". The AI's second message is timestamped at 8:08 AM: "The tallest building in my knowledge is the Burj Khalifa in".

At the bottom of the chat window, there's a progress bar indicating "Generating bot response" with three dots, and a large empty text input field.