GitHub Copilot - Part 1

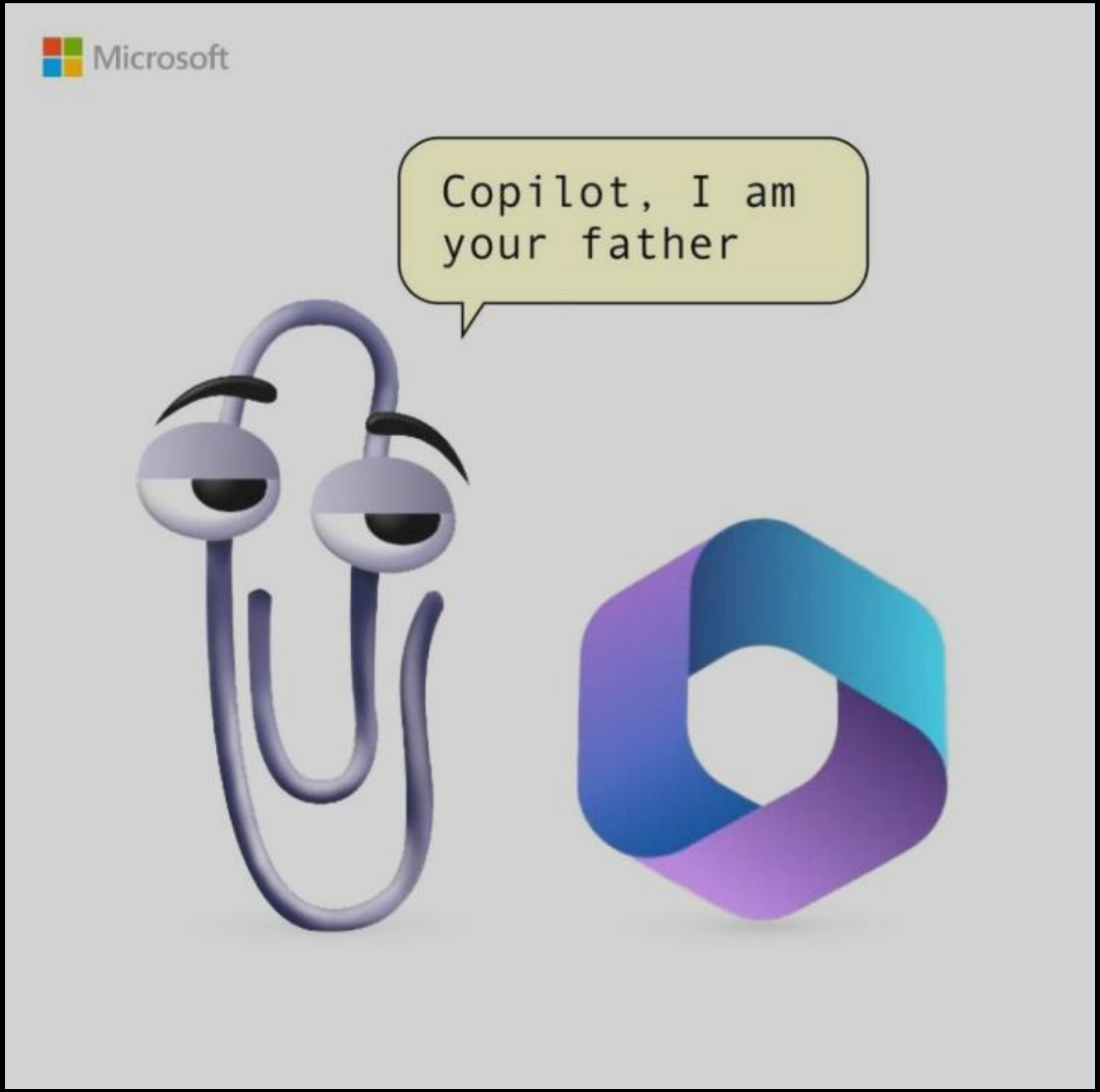
By RaKeTe-Technology

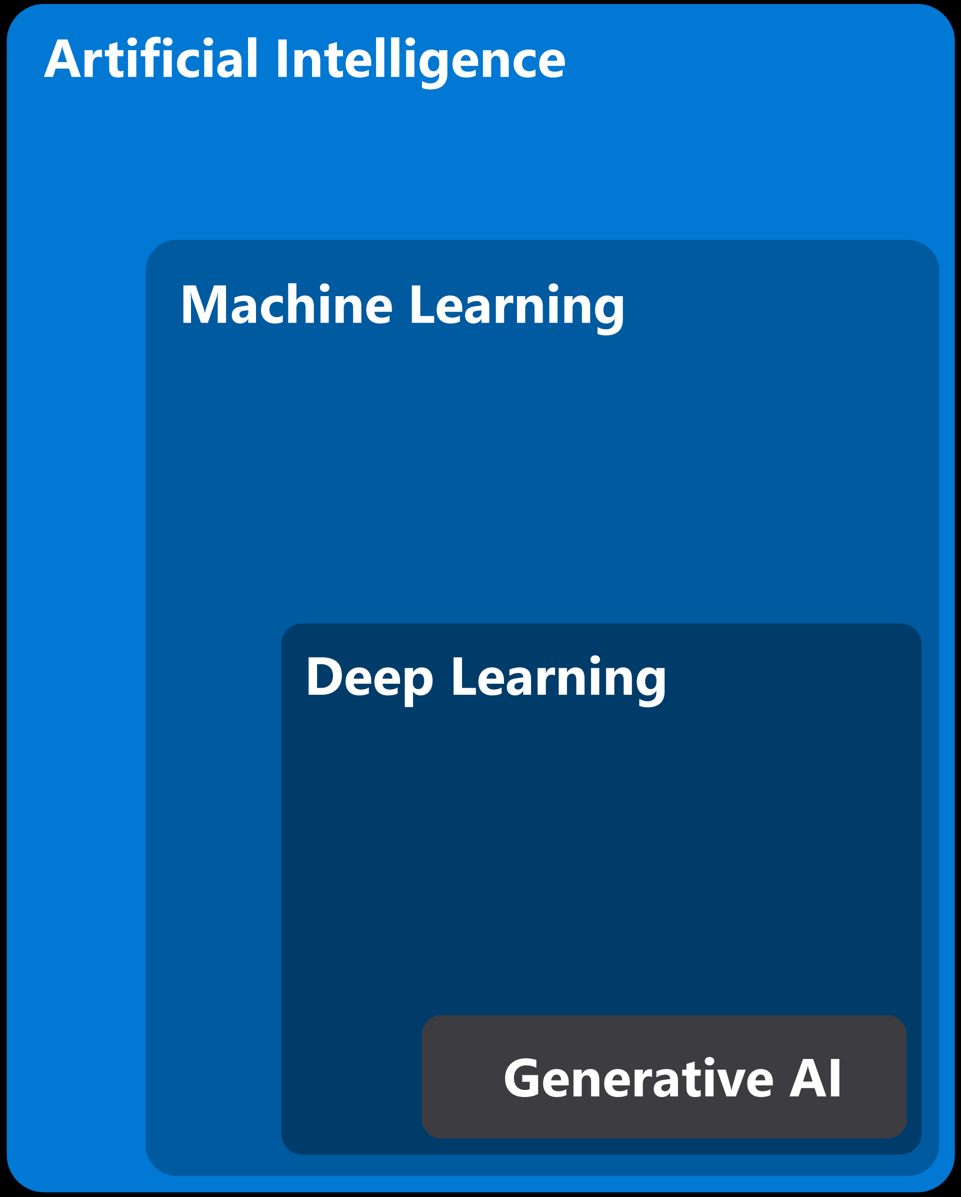
Ralph.Kemperdick@RaKeTeTechnology.com
Cloud Solution Architect, Advanced Analytics AI + Data & Analytics
LinkedIn www.linkedin.com/in/ralphke

Agenda

1. Introduction - Where is it coming from?
2. Key Concepts & Best Practices
3. Technology stack
4. Considerations and Limitations
5. GitHub Copilot Demos
6. Q & A

A little bit of history





Artificial Intelligence

Machine Learning

Deep Learning

Generative AI

1956

Artificial Intelligence

the field of computer science that seeks to create intelligent machines that can replicate or exceed human intelligence

1997

Machine Learning

subset of AI that enables machines to learn from existing data and improve upon that data to make decisions or predictions

2017

Deep Learning

a machine learning technique in which layers of neural networks are used to process data and make decisions

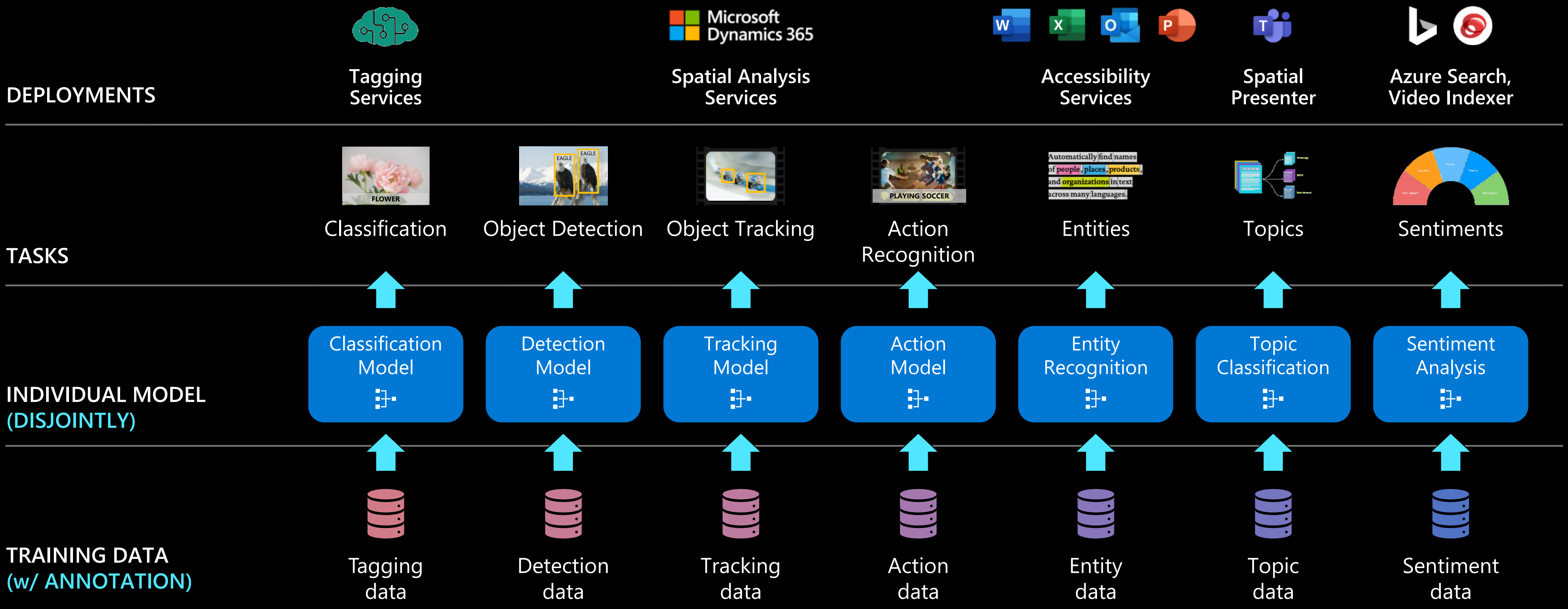
2021

Generative AI

Create new written, visual, and auditory content given prompts or existing data.

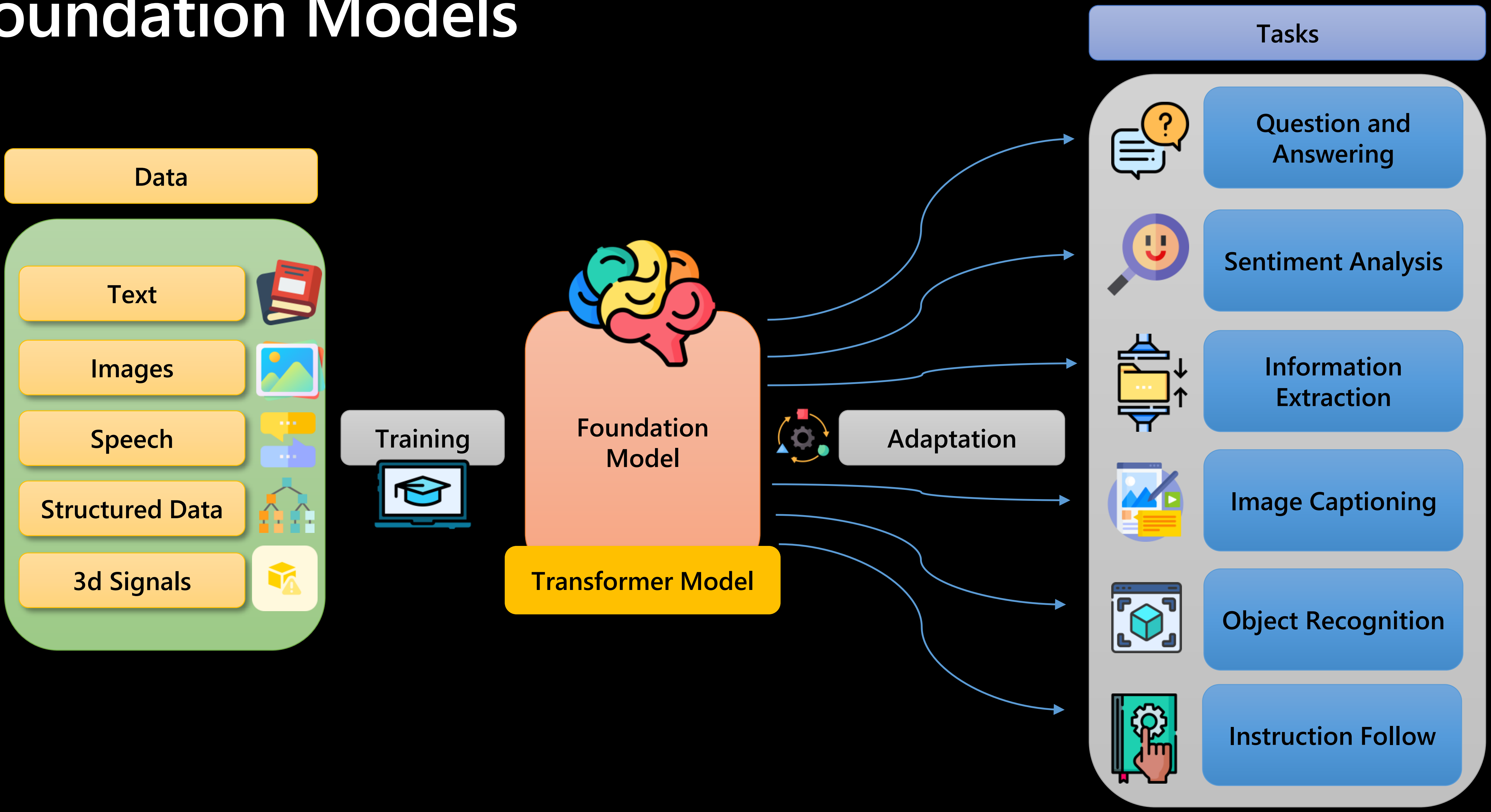
Traditional model development

High cost and slow deployment—each service is trained disjointly


Microsoft Dynamics 365
FLOWER
EAGLE
EAGLE
PLAYING SOCCER
Automatically find names of people, places, products, and organizations in text across many languages.

DEPLOYMENTS	Tagging Services		Spatial Analysis Services		Accessibility Services	Spatial Presenter	Azure Search, Video Indexer
TASKS	Classification	Object Detection	Object Tracking	Action Recognition	Entities	Topics	Sentiments
INDIVIDUAL MODEL (DISJOINTLY)	Classification Model	Detection Model	Tracking Model	Action Model	Entity Recognition	Topic Classification	Sentiment Analysis
TRAINING DATA (w/ ANNOTATION)	Tagging data	Detection data	Tracking data	Action data	Entity data	Topic data	Sentiment data

Foundation Models



Data

Text

Images

Speech

Structured Data

3d Signals

Training

Foundation Model

Transformer Model

Adaptation

Tasks

Question and Answering

Sentiment Analysis

Information Extraction

Image Captioning

Object Recognition

Instruction Follow

OpenAI

Ensure that artificial general intelligence (AGI) benefits humanity.

Groundbreaking research
Latest models
Fast innovation

+

Microsoft

Empower every person and organization on the planet to achieve more

Enterprise features
Production SLAs
Data privacy

GPT-4

Codex

DALL·E

ChatGPT

OpenAI Microsoft

Generative AI

GPT-4

Generate and Understand Text

Prompt:

Write a tagline for an ice cream shop.

Response:

We serve up smiles with every scoop!

Codex

Generate and Understand Code

Prompt:

```
Table customers, columns =
[CustomerId, FirstName,
LastName, Company, Address,
City, State, Country,
PostalCode]

Create a SQL query for all
customers in Texas named Jane
query =
```

Response:

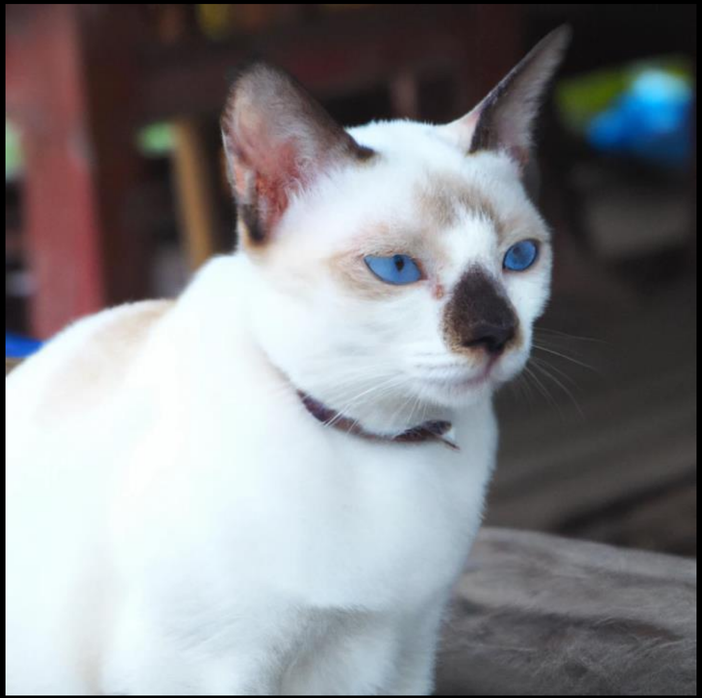
```
SELECT *
FROM customers
WHERE State = 'TX' AND
FirstName = 'Jane'
```

DALL·E

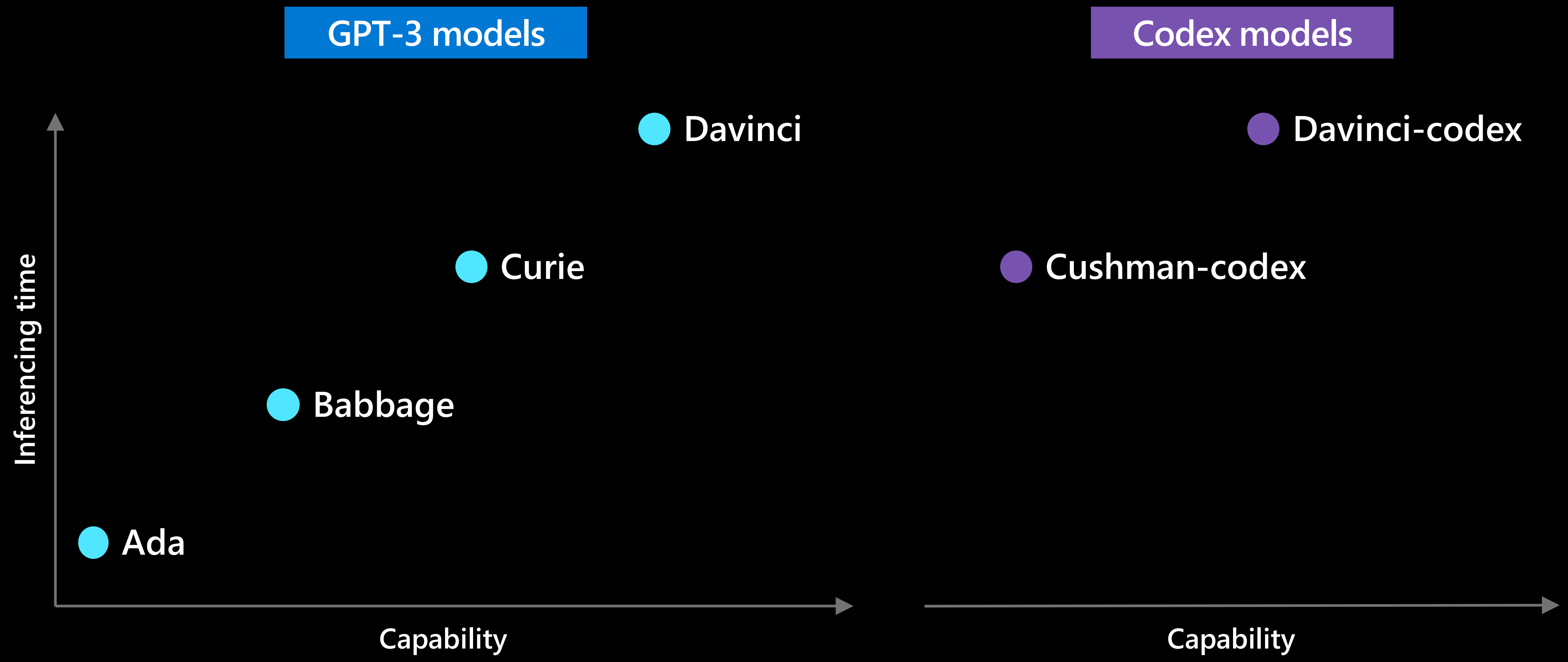
Generate images from text prompts

Prompt: A white Siamese cat

Response:



Azure OpenAI | Family of Models



GPT-3 models

Codex models

Azure OpenAI | Overview of GPT-X

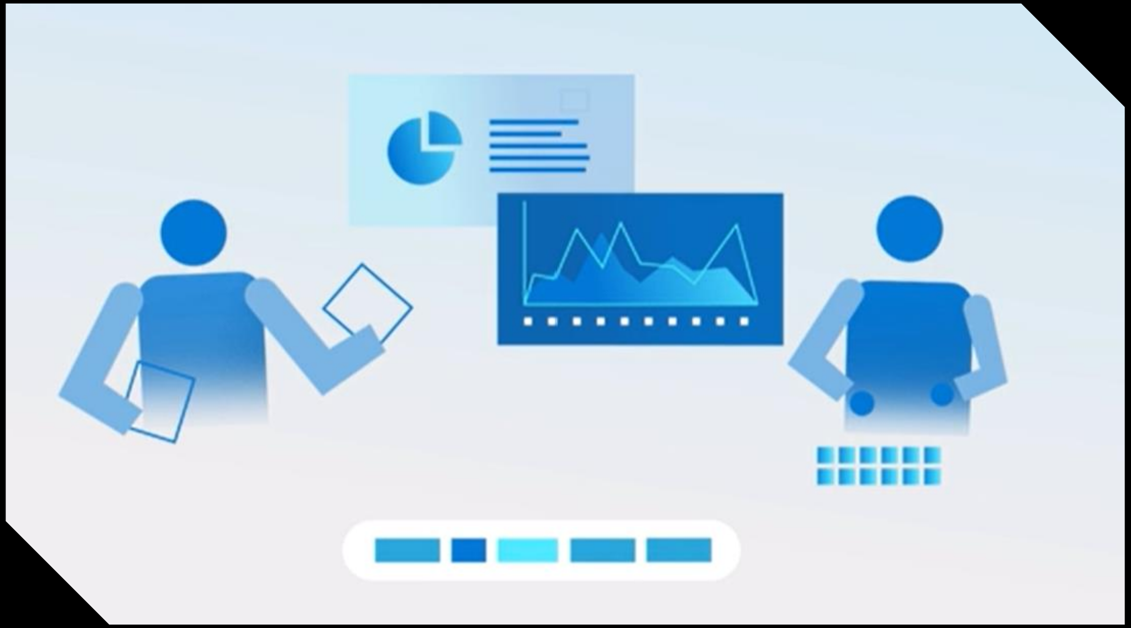
Generative pre-trained transformer (GPT-4)

Autoregressive language model that uses deep learning to produce human-like text

Pre-trained on trillions of words

Predicts the most likely next word based on input text

General text-in/text-out interface



Azure OpenAI | GPT-X Models

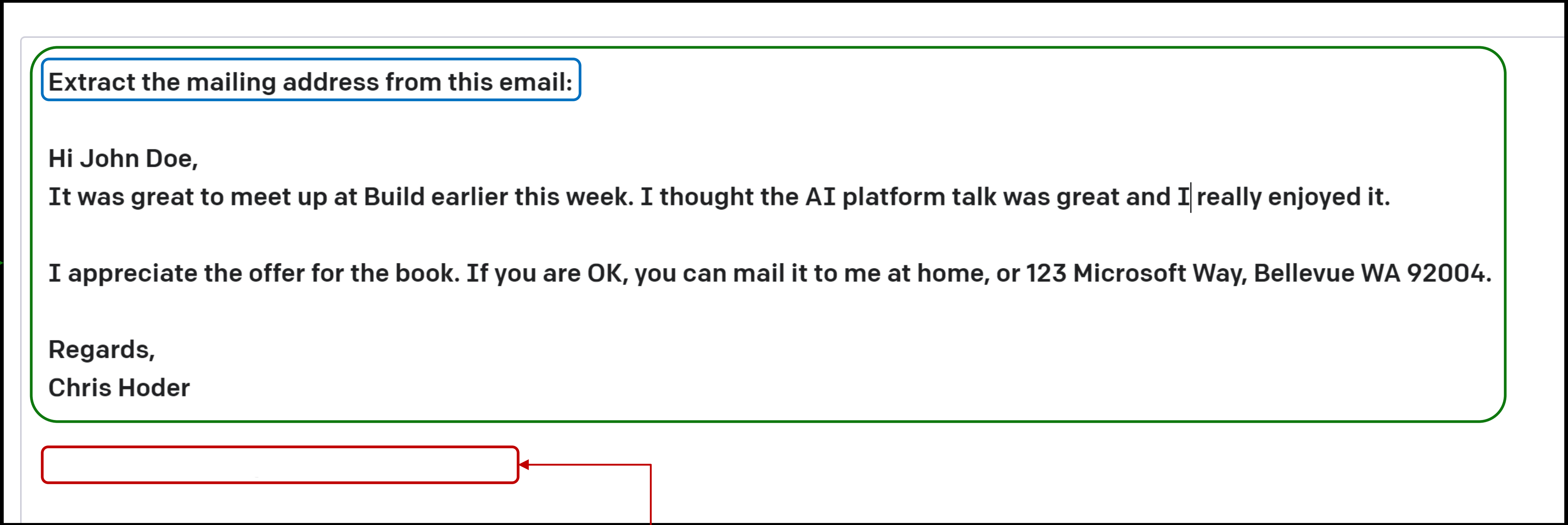
Powerful language models accessible to all skill levels

General purpose primarily text-in ->
code-, text-, image-, sound-, video-out interface—flexibility

Simple UX—validate proof of concepts fast

Built in ML science intuition for everyone, with deeper controls for ML practitioners

Azure OpenAI | GPT-X 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.

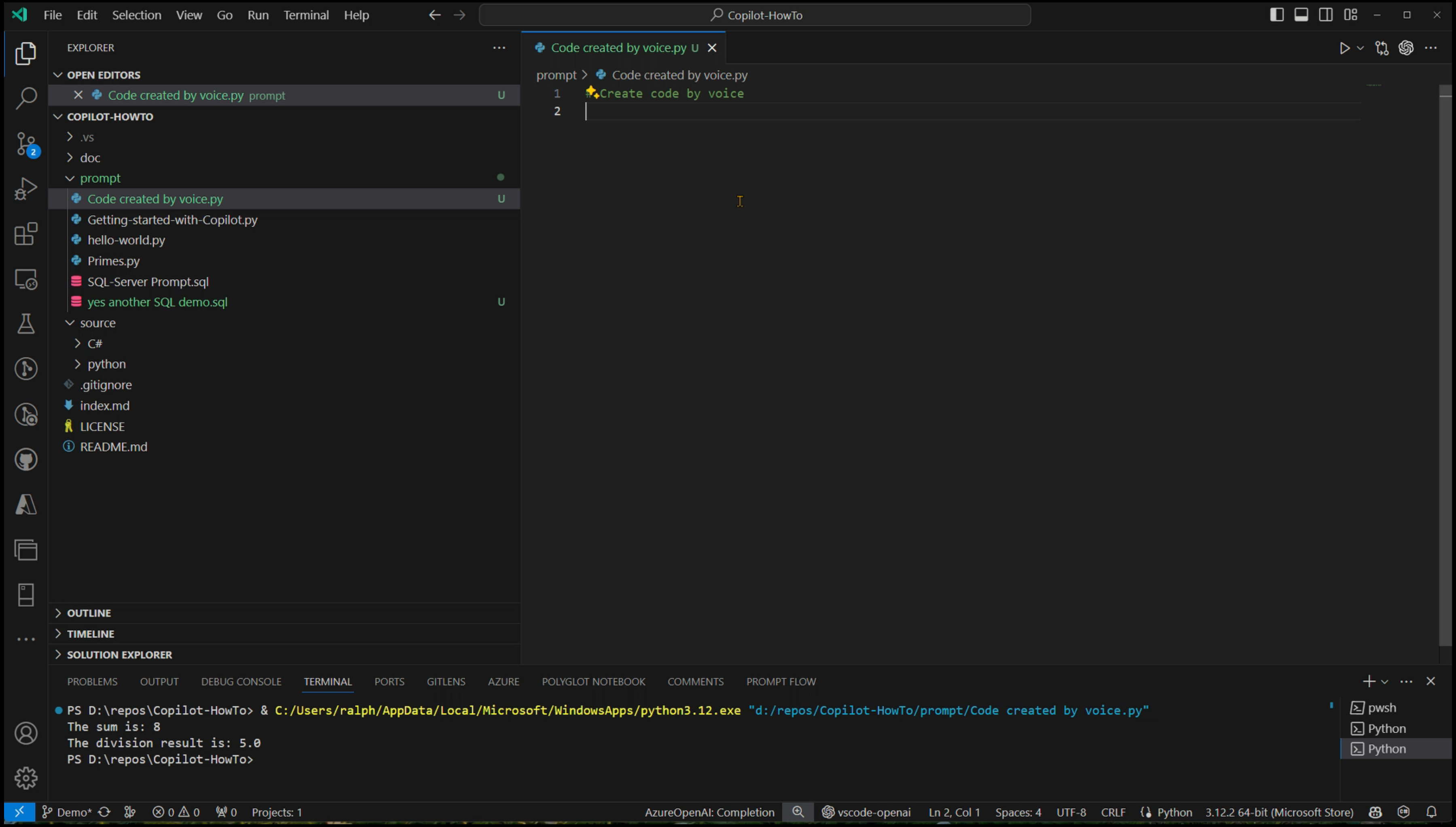
Completion—Output that GPT-X generates based on the prompt.

Demo

GitHub Copilot Demo

- Generating code via voice

Programming with your voice



```
Create code by voice
```

```
PS D:\repos\Copilot-HowTo> & C:/Users/ralph/AppData/Local/Microsoft/WindowsApps/python3.12.exe "d:/repos/Copilot-HowTo/prompt/Code created by voice.py"
The sum is: 8
The division result is: 5.0
PS D:\repos\Copilot-HowTo>
```

Basic Prompting

- Prompt Engineering
  - The concept of Natural Language Processing (NLP) involving integrating descriptions of tasks in input, to prompt the model to output the desired results.
- Single-Shot

One question with a **clear** statement what to do

1. Give me a summary of this website: www.xyz.com

Unclear question:

2. Generate a formular for calculating prime numbers

Missing information about the implementation, like Python code, for example

Azure OpenAI | GPT-X Ideate, Experiment and Fine-Tune

Iterate on ideas with a general-purpose text-in/text-out interface

Prompt (single shot)

What are popular sports played in Munich?:

In **Munich**, several sports are popular among the community.
1.Soccer (Football): Like in many other places, ...

What is the most famous soccer team there?

Sample response

The most famous soccer team in **Munich** is Bayer München ....

Refine with examples with a simple UX

Prompt (few-/multi shot)

Turn game commentary into highlights:

Commentary: What a pickup she has

Main highlights: New York has domina

Examples

###

Commentary:
1. Turner is so important defensively to
2. Griner pulled way out, Hartley with
3. At 1:54 remaining in the quarter, Pho

Inputs

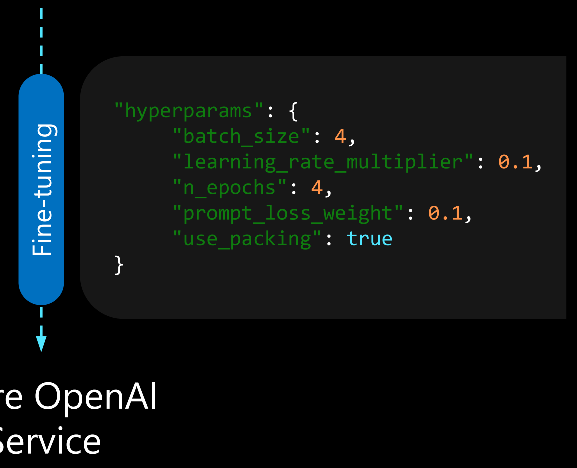
Sample response

Main highlights:
1. New York has had a strong run in the
2. Phoenix leading by 1 point, 24-23
3. New York Liberty's comeback has be

Results

Optimize accuracy and latency to validate proof of concept fast

Prompt and completion examples



```
"hyperparams": {
      "batch_size": 4,
      "learning_rate_multiplier": 0.1,
      "n_epochs": 4,
      "prompt_loss_weight": 0.1,
      "use_packing": true
}
```

Codex Models

Makes APIs more accessible

Accelerates software development

Widens who can code

Codex Models

Derived from base models and trained on both NL and code (billions of Lines of Code)

Support multiple programming languages

- Python, C#, SQL, Java, JavaScript, TypeScript, Go, Perl, PHP, Ruby, Swift, Shell (bash)

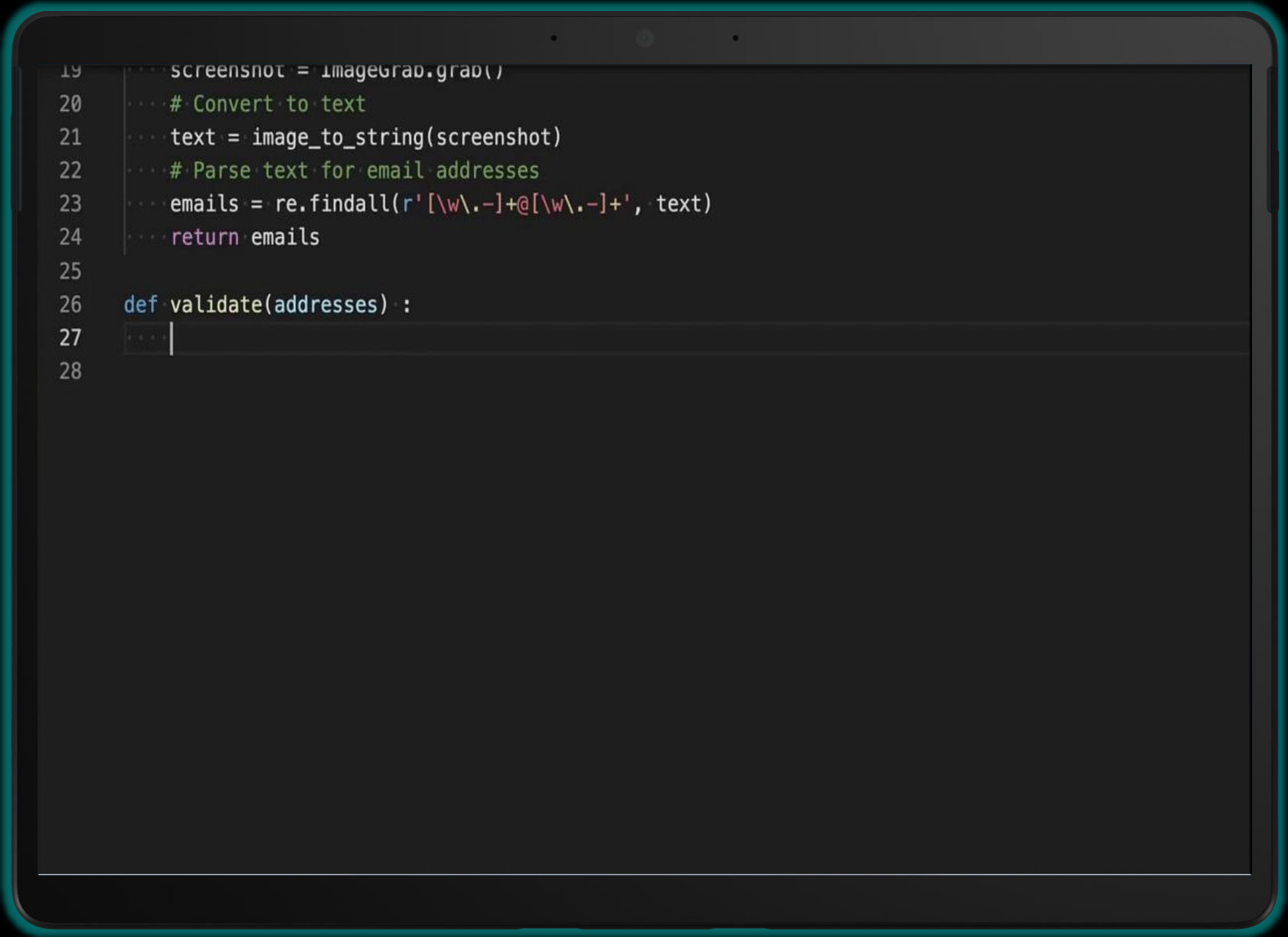
Multiple tasks:

- Comment → Code
- Autocomplete function or next line (in context)
- Knowledge searching (API or Library call)
- Documenting code (comments)
- Refactoring

Use Cases

- Natural Language to Code
- Documenting code (comments)
- Refactoring
- Code to Natural Language
- Natural Language to SQL

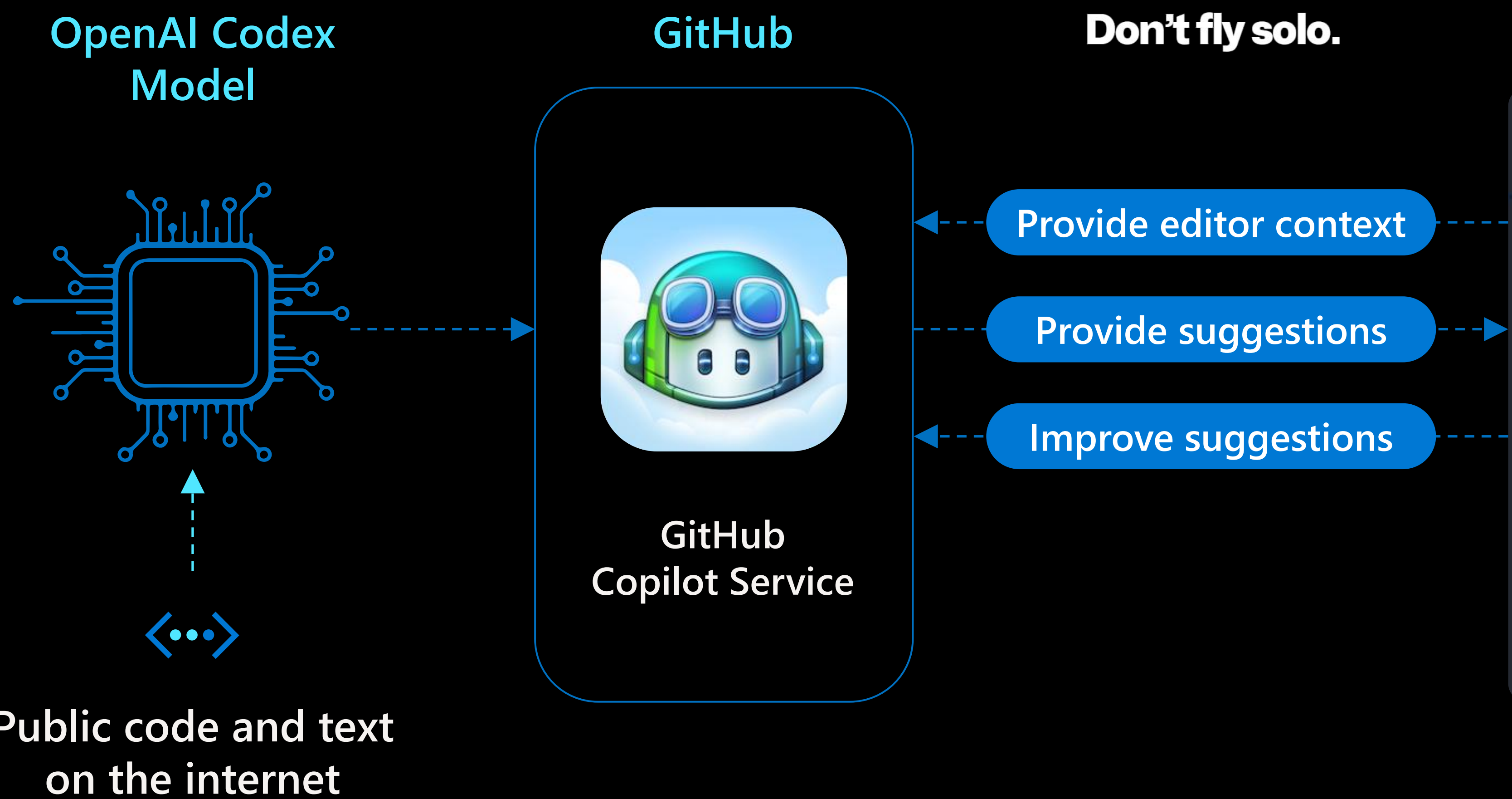
Codex Models

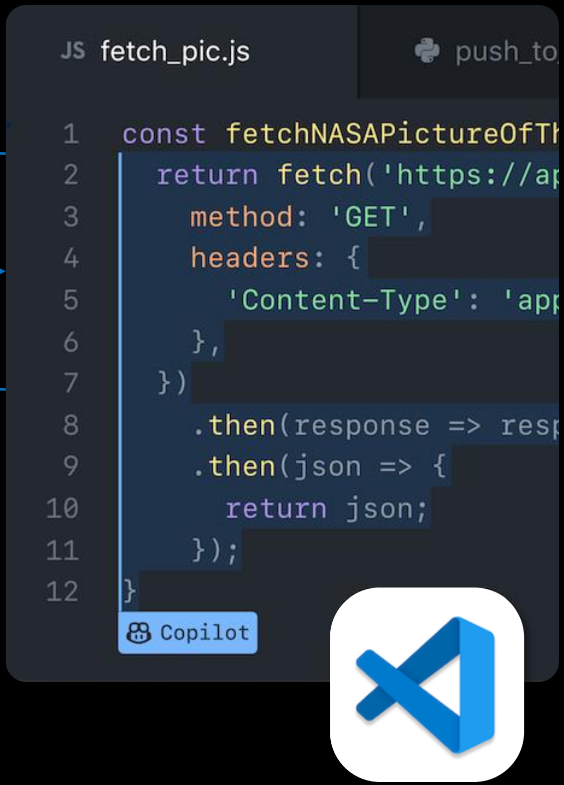


```
    screenshot = ImageGrab.grab()
    # Convert to text
    text = image_to_string(screenshot)
    # Parse text for email addresses
    emails = re.findall(r'[\w\.-]+@[\w\.-]+', text)
    return emails

def validate(addresses) :
    
```

OpenAI + GitHub

Don't fly solo.





fetch_pic.js   push_to

```
const fetchNASAPictureOfTh
  return fetch('https://ap
    method: 'GET',
    headers: {
      'Content-Type': 'app
    },
  })
    .then(response => resp
    .then(json => {
      return json;
    });
}
```

Codex

Alternative: StarCoderEx Model

New open-access LLM model from ServiceNow & Hugging Face

- VS Code Extension “StarCodeEx” can generate code from natural language
- Trained on 15 billion parameters and 1 trillion tokens and 35 billion python tokens
- Based on Megatron-LM, can now process more that the Codes model
- Get involved at BigCode
- Check it out at HuggingFace

Support multiple programming languages

- Python, SQL, Java, JavaScript, PHP, ..

Multiple tasks:

- Comment → Code
- Autocomplete function or next line (in context)
- Documenting code (comments)
- Refactoring

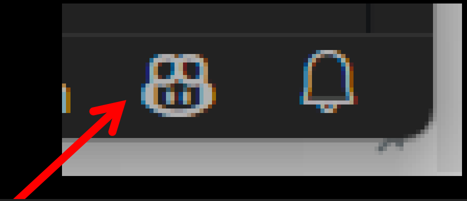
Use Cases

- Natural Language to Code
- Documenting code (comments)
- Refactoring

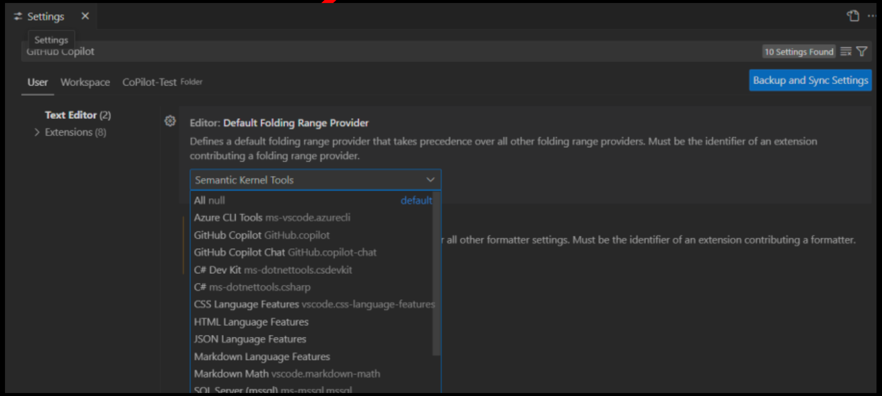
How to configure Copilot

Within Visual Studio Code:

Click on the copilot icon in the lower right:



Editor settings (Range Provider)



Settings

GitHub Copilot

10 Settings Found

User Workspace CoPilot-Test Folder

Backup and Sync Settings

Text Editor (2)

Extensions (8)

Editor: **Default Folding Range Provider**

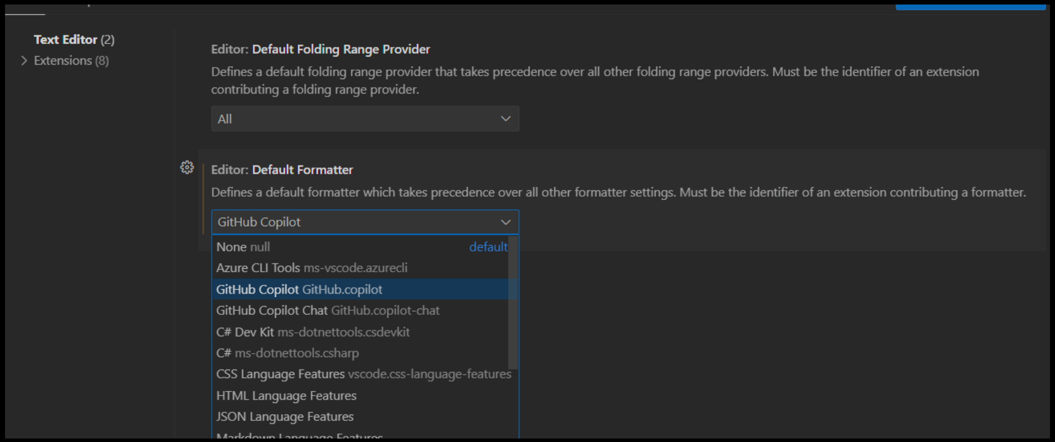
Defines a default folding range provider that takes precedence over all other folding range providers. Must be the identifier of an extension contributing a folding range provider.

Semantic Kernel Tools

All null default
Azure CLI Tools ms-vscode.azurecli
GitHub Copilot GitHub.copilot
GitHub Copilot Chat GitHub.copilot-chat
C# Dev Kit ms-dotnettools.csdevkit
C# ms-dotnettools.csharp
CSS Language Features vscode.css-language-features
HTML Language Features
JSON Language Features
Markdown Language Features
Markdown Math vscode.markdown-math
SQL Server (mssql) ms-mssql.mssql

r all other formatter settings. Must be the identifier of an extension contributing a formatter.

Default Formatter:



Text Editor (2)

Extensions (8)

Editor: **Default Folding Range Provider**

Defines a default folding range provider that takes precedence over all other folding range providers. Must be the identifier of an extension contributing a folding range provider.

All

Editor: **Default Formatter**

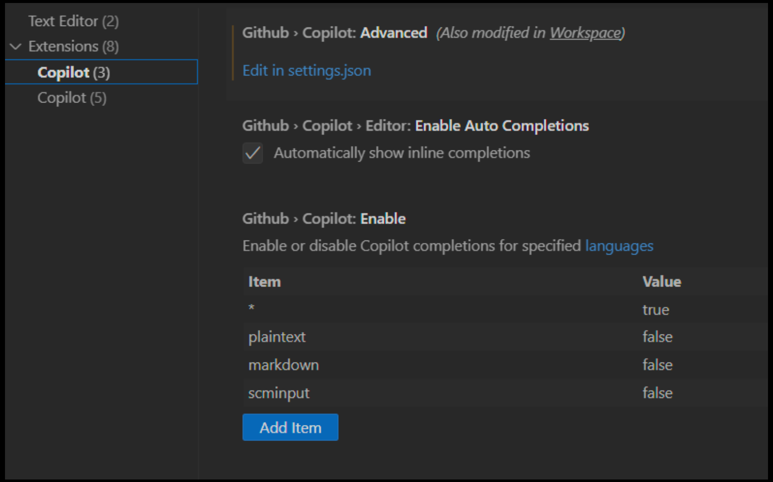
Defines a default formatter which takes precedence over all other formatter settings. Must be the identifier of an extension contributing a formatter.

GitHub Copilot

None null default
Azure CLI Tools ms-vscode.azurecli
GitHub Copilot GitHub.copilot
GitHub Copilot Chat GitHub.copilot-chat
C# Dev Kit ms-dotnettools.csdevkit
C# ms-dotnettools.csharp
CSS Language Features vscode.css-language-features
HTML Language Features
JSON Language Features
Markdown Language Features

How to configure Copilot in VS Code

Copilot Auto Completions:



Github › Copilot: **Advanced** *(Also modified in Workspace)*

Edit in settings.json

Github › Copilot › Editor: **Enable Auto Completions**

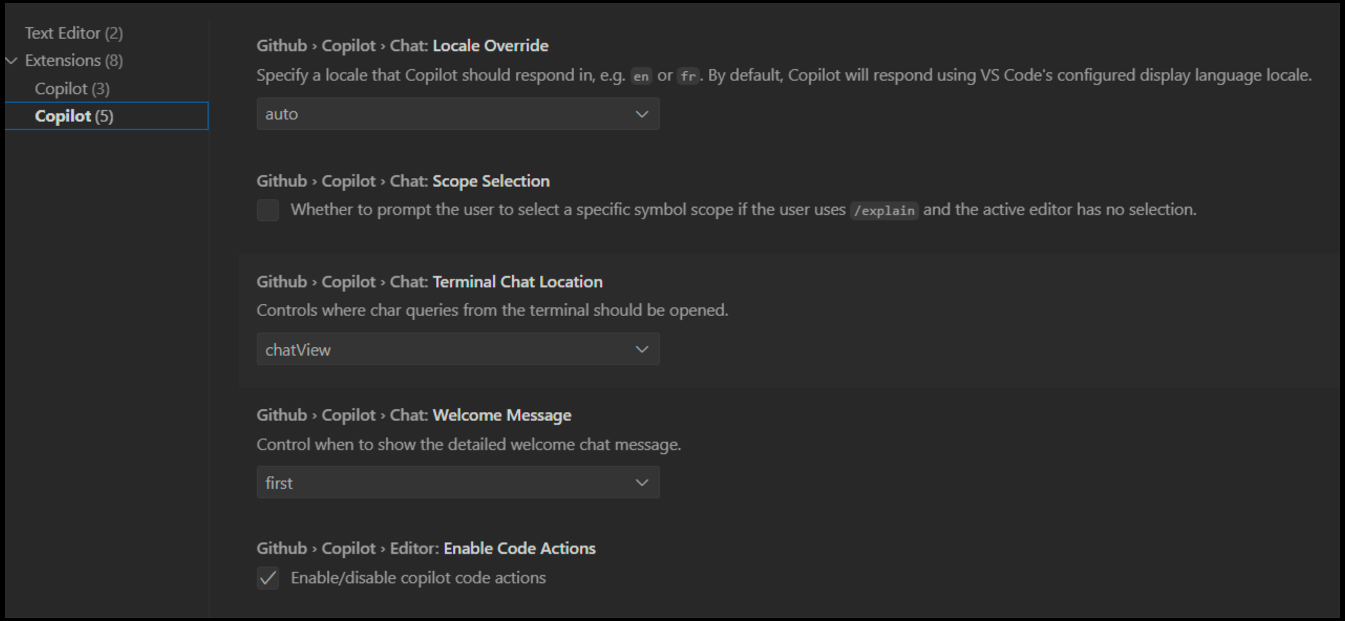
☑ Automatically show inline completions

Github › Copilot: **Enable**

Enable or disable Copilot completions for specified languages

Item	Value
*	true
plaintext	false
markdown	false
scminput	false

Scope settings:



Github › Copilot › Chat: **Locale Override**

Specify a locale that Copilot should respond in, e.g. `en` or `fr`. By default, Copilot will respond using VS Code's configured display language locale.

auto

Github › Copilot › Chat: **Scope Selection**

□ Whether to prompt the user to select a specific symbol scope if the user uses `/explain` and the active editor has no selection.

Github › Copilot › Chat: **Terminal Chat Location**

Controls where char queries from the terminal should be opened.

chatView

Github › Copilot › Chat: **Welcome Message**

Control when to show the detailed welcome chat message.

first

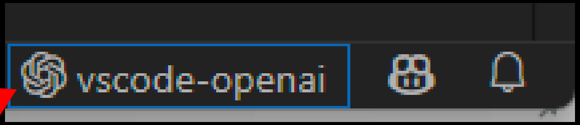
Github › Copilot › Editor: **Enable Code Actions**

☑ Enable/disable copilot code actions

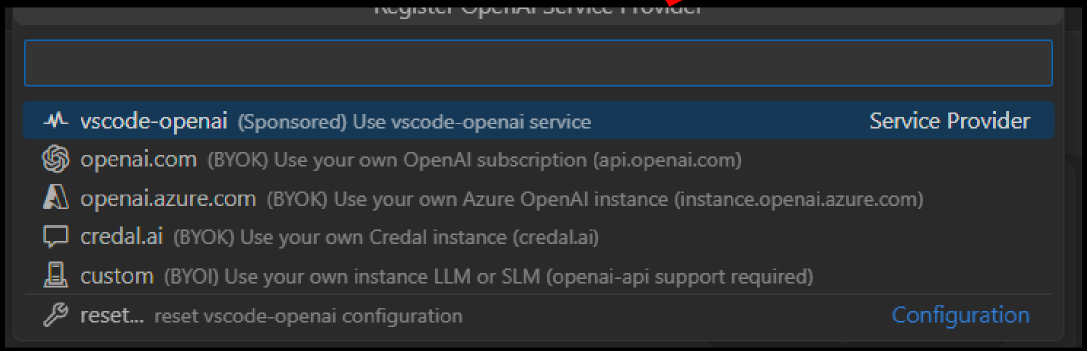
How to configure The Service for Copilot?

Within Visual Studio Code:

Click on the OpenAI icon in the lower right:



Select Service Provider:


Register OpenAI Service Provider

vscode-openai (Sponsored) Use vscode-openai service — Service Provider

openai.com (BYOK) Use your own OpenAI subscription (api.openai.com)

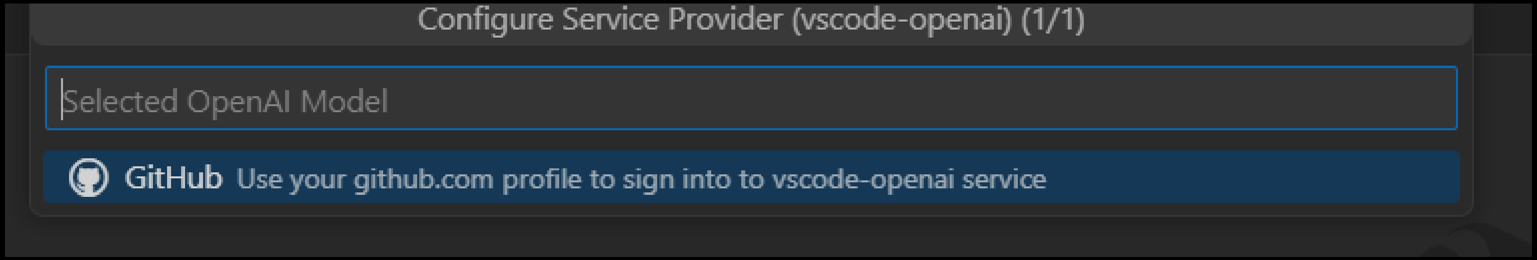
openai.azure.com (BYOK) Use your own Azure OpenAI instance (instance.openai.azure.com)

credal.ai (BYOK) Use your own Credal instance (credal.ai)

custom (BYOI) Use your own instance LLM or SLM (openai-api support required)

reset... reset vscode-openai configuration — Configuration

GitHub account for example:


Configure Service Provider (vscode-openai) (1/1)

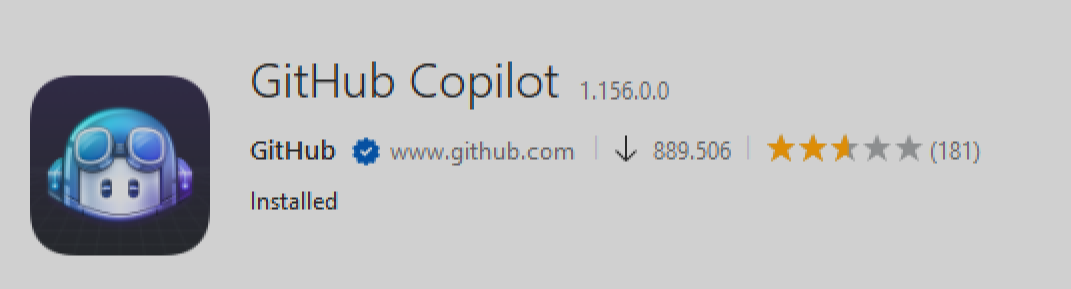
Selected OpenAI Model

GitHub Use your github.com profile to sign into to vscode-openai service

How to configure GitHub Copilot

Within Visual Studio 2022:

Install GitHub Copilot Extension:

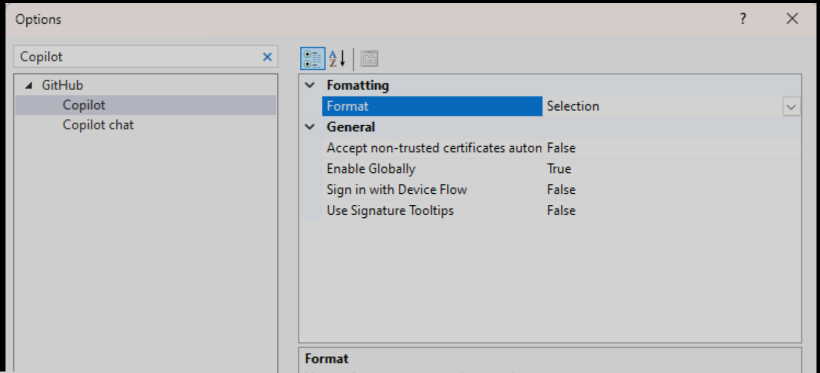


GitHub Copilot 1.156.0.0

GitHub www.github.com | 889.506 | ★★★★★ (181)

Installed

Under Tools -> Options



Options

Copilot

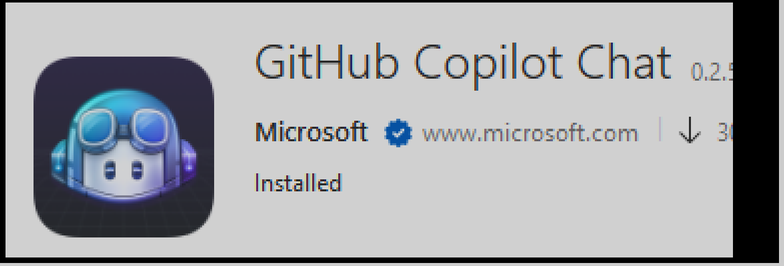
- GitHub
  - Copilot
  - Copilot chat

Fomatting	
Format	Selection
General	
Accept non-trusted certificates auton	False
Enable Globally	True
Sign in with Device Flow	False
Use Signature Tooltips	False

Format

How to format an accepted suggestion

Search for “Copilot”:

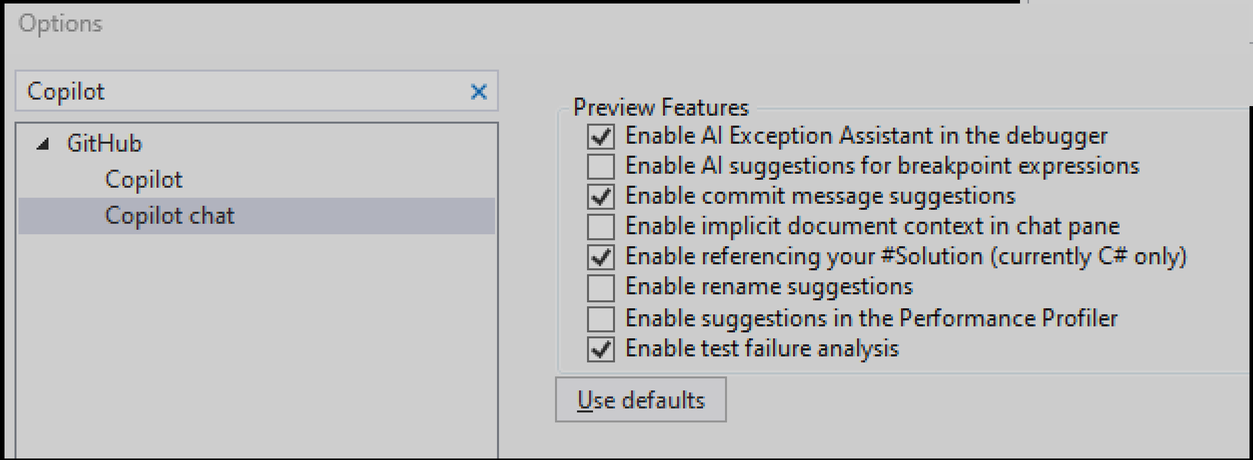


GitHub Copilot Chat 0.2.5

Microsoft www.microsoft.com | 3

Installed

Copilot Chat:



Options

Copilot

- GitHub
  - Copilot
  - Copilot chat

Preview Features

- ☑ Enable AI Exception Assistant in the debugger
- ☐ Enable AI suggestions for breakpoint expressions
- ☑ Enable commit message suggestions
- ☐ Enable implicit document context in chat pane
- ☑ Enable referencing your #Solution (currently C# only)
- ☐ Enable rename suggestions
- ☐ Enable suggestions in the Performance Profiler
- ☑ Enable test failure analysis

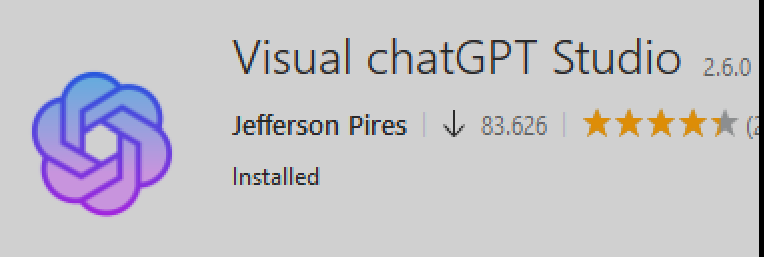
How to configure Visual ChatGPT

Within Visual Studio 2022:

Configure Visual chatGTP Studio:

Under Tools -> Options

Search for “Chat”:

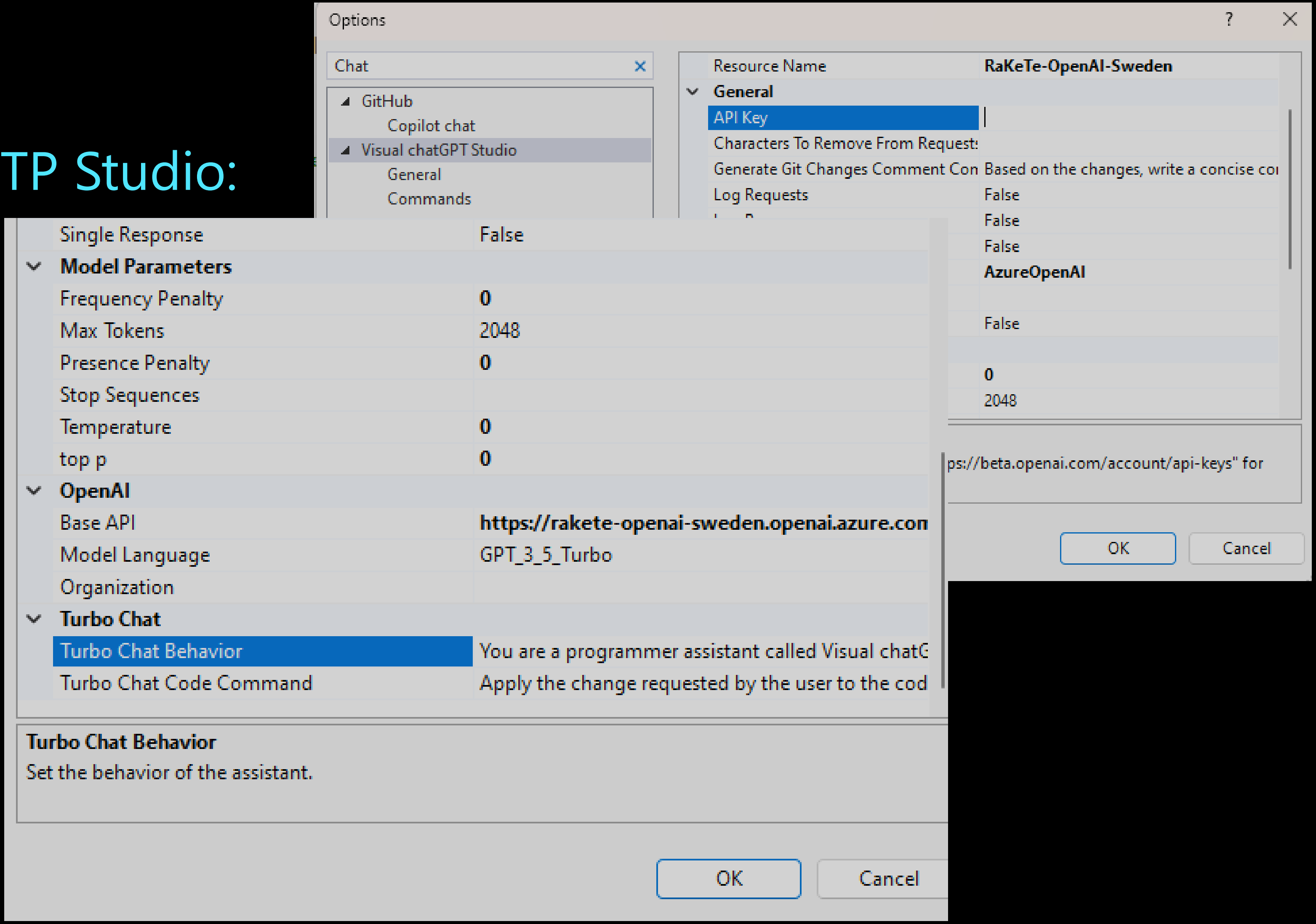


Visual chatGPT Studio 2.6.0

Jefferson Pires | ↓ 83.626 | ★★★★★ (2

Installed

General:



Resource Name	RaKeTe-OpenAI-Sweden
General	
API Key	
Characters To Remove From Request:	
Generate Git Changes Comment Con	Based on the changes, write a concise coı
Log Requests	False
	False
	False
	AzureOpenAI
	False
	0
	2048

ps://beta.openai.com/account/api-keys" for

Single Response	False
Model Parameters	
Frequency Penalty	0
Max Tokens	2048
Presence Penalty	0
Stop Sequences	
Temperature	0
top p	0
OpenAI	
Base API	https://rakete-openai-sweden.openai.azure.con
Model Language	GPT_3_5_Turbo
Organization	
Turbo Chat	
Turbo Chat Behavior	You are a programmer assistant called Visual chatG
Turbo Chat Code Command	Apply the change requested by the user to the cod

Turbo Chat Behavior

Set the behavior of the assistant.

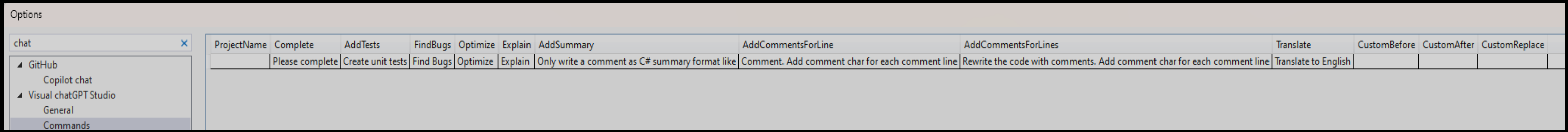
Marketplace: Visual chatGPT Studio - Visual Studio Marketplace

GitHub: jeffdapaz/VisualChatGPTStudio: Add chatGPT functionalities directly on Visual Studio (github.com

How to configure Visual ChatGPT

Commands:

Allow to change the behavior via natural language prompts:



ProjectName	Complete	AddTests	FindBugs	Optimize	Explain	AddSummary	AddCommentsForLine	AddCommentsForLines	Translate	CustomBefore	CustomAfter	CustomReplace
	Please complete	Create unit tests	Find Bugs	Optimize	Explain	Only write a comment as C# summary format like	Comment. Add comment char for each comment line	Rewrite the code with comments. Add comment char for each comment line	Translate to English			

GitHub CoPilot or Visual Chat GPT?

When to use what?

1. Visual Chat GPT exists only as plugin for Visual Studio 2022
    1. Allow fine grained configurations
    2. System Prompts per project
    3. GitHub Copilot in Visual Studio: A Recap of 2023 - Visual Studio Blog (microsoft.com)

Demo

GitHub Copilot Demo

- Visual Studio
- Visual Studio Code

GitHub Copilot in Visual Studio 2022 >= V17.9

- Inline Chat via: Alt-#
- /-Commands:
  - /doc | /explain | /fix | /optimize | /tests
- Context variable
  - #solution #FileName.cs
- Analyze and Fix in Test Explorer

/-Commands in Copilot Inline & Chat

Command	Usage	Chat window	Inline chat
/doc	Add comments for specified or selected code. Examples: - /doc DeleteBasketAsync method in BasketService.cs - select desired code and enter /doc	Yes	Yes
/explain	Get code explanations. Examples: - /explain the AddItemToBasket method in BasketService.cs - select desired code and enter /explain	Yes	Yes
/fix	Propose a fix for problems in the selected code. Examples: - /fix the SetQuantities method in BasketService.cs - select desired code and enter /fix	Yes	Yes
/generate	Generate code to answer specified question. Example: /generate code to add two numbers in Calculator.cs	Yes	Yes
/help	Get help on using Copilot Chat. Example: /help	Yes	Yes
/optimize	Analyze and improve running time of the selected code. Examples: - /optimize the AddItemToBasket method in BasketService.cs - select desired code and enter /optimize	Yes	Yes
/tests	Create unit tests for the selected code. Example: select desired code and enter /tests	Yes	Yes

GitHub Copilot in Visual Studio

- Exception Analysis with Copilot
- Auto Insight with Profiling Tools
- Rename suggestions (Refactoring)
- Commit messages auto generated
- Breakpoint Expressions with IntelliSense
- Deadlock Analysis
- /-Command only in Chat:
  - /askvs
  - /generate

GitHub Copilot in Visual Studio Code

- // q: what dose this function do?
- Other languages work also but might give different results
- # importiere das Modul response
- Inline Chat via: Shift-Ctrl-I
- /-Commands:
  - /doc | /explain | /fix | /optimize | /tests
- Context variables
  - @workspace #FileName.cs
  - @vscode - How to do something inside VS code
  - @terminal – review
- Add commit messages

GitHub Copilot CLI

- Ask Copilot about specific commands and tools
  - `gh copilot explain “winget install”`
- Get suggestions
  - `gh copilot suggest “install git”`
- Setup alias for
  - `gh copilot suggest – ghcs`
  - `gh copilot explain - ghce`
  - `gh copilot alias`
- Context variables
  - @workspace #FileName.cs
  - @vscode - How to do something inside VS code
  - @terminal – review
- Add commit messages

GitHub CoPilot for CLI configuration

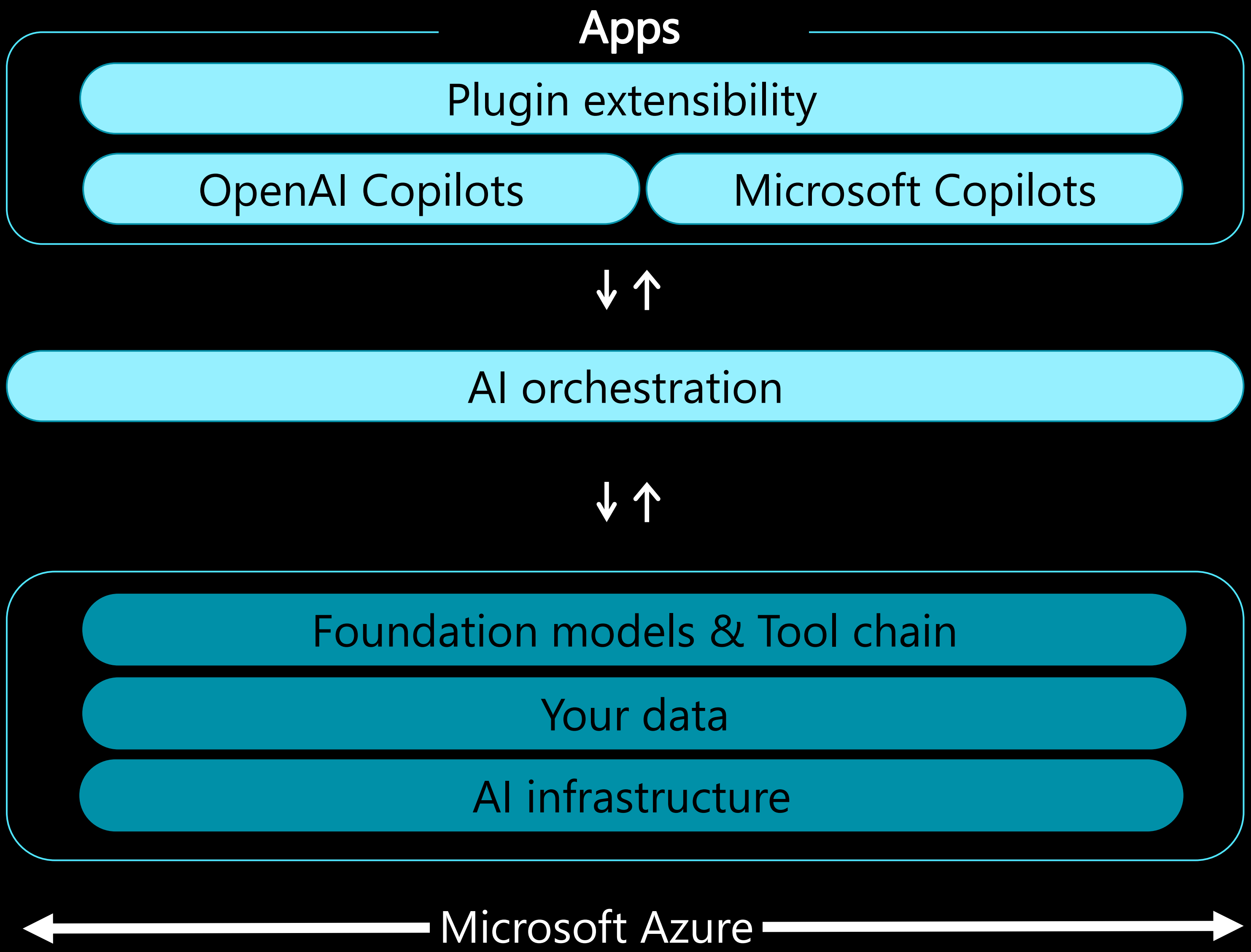
1.
```
&"c:/Program Files/Git/gh.exe" auth login --web -h github.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
! First copy your one-time code: X1Y2-Z3A4
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as Clippy
! You were already logged in to this account
```

GitHub Copilot HowTo Repro

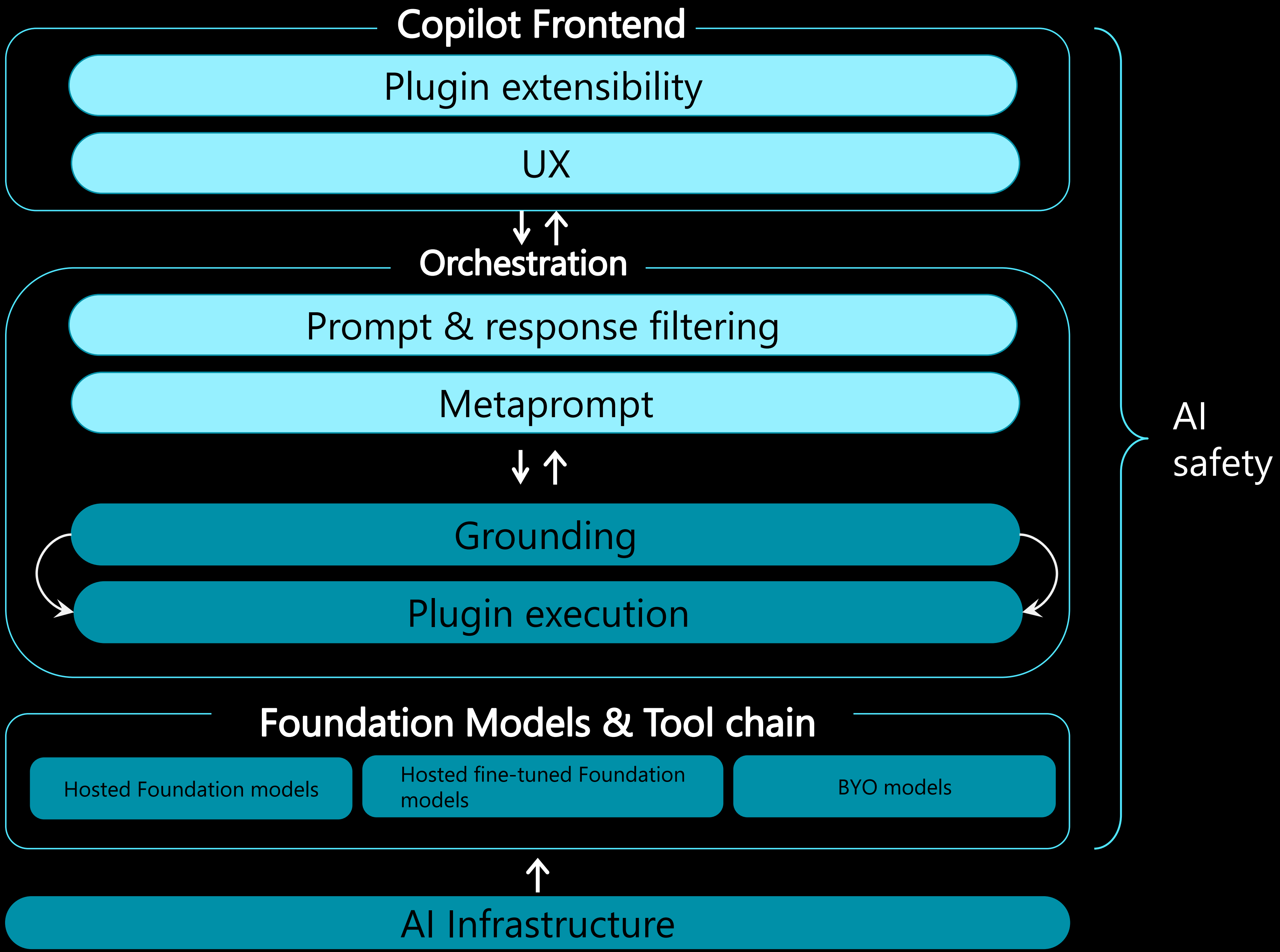
In this repro you find several exercises to get familiar with GitHub Copilot

https://github.com/ralphke/Copilot-HowTo

Copilot stack



Copilot stack



Copilot Frontend

Plugin extensibility

UX

Orchestration

Prompt & response filtering

Metaprompt

Grounding

Plugin execution

Foundation Models & Tool chain

Hosted Foundation models

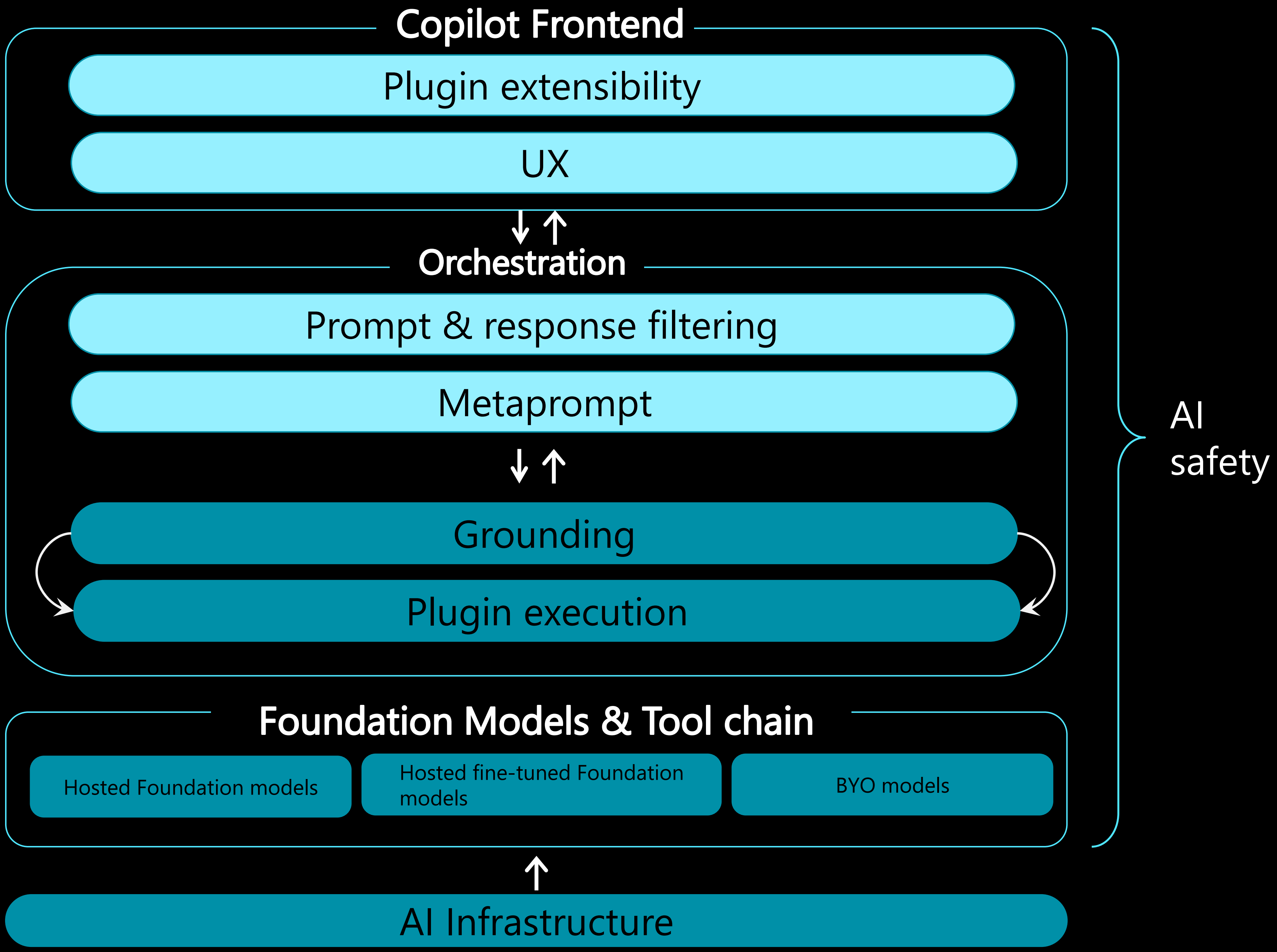
Hosted fine-tuned Foundation models

BYO models

AI Infrastructure

AI safety

Copilot stack



Limitations

Quality of Generated Code

While Copilot is impressive, it doesn’t always produce code that adheres to best practices. For instance, in JavaScript, it might generate var and == instead of const and ===, which can lead to subtle bugs and shadowing[1].

Biased Language and Libraries

Copilot tends to favor certain libraries or frameworks. For example, in Python, it heavily leans toward using SQLAlchemy, which may not align with your preferred stack[2].

Single Context

You can only provide one piece of context when asking Copilot a question. This limitation can impact its ability to generate accurate code when dealing with complex scenarios[2].

Best practices

Single, Specific, Short

- Single responsibility
- Specific prompt
- Short response

Patterns

- Regex, CRON, PowerShell

Trust but verify

Context is important

Iterate, iterate, iterate

Best Practices

1. **Contextual Comments**
2. **Brain Dump Approach**
3. **Handle Connections Gracefully (Python + SQL Server)**

GitHub Copilot is a powerful tool, but it’s **essential** to **review** and refine the generated code to ensure it aligns with your specific requirements and best practices

Brain Dump Approach

- **Expand Context**: If Copilot struggles with complex queries, provide additional context. Right-click on relevant tables to get their creation scripts (e.g., Sales.OrderLines and Sales.Orders).
- **Example**

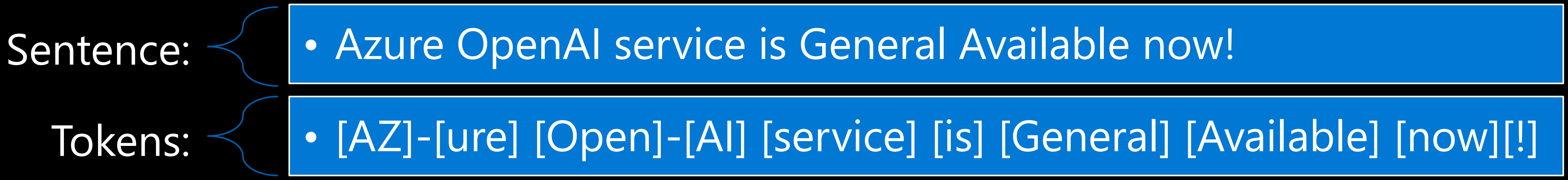
```
-- Sales.OrderLines table
CREATE TABLE Sales.OrderLines (
OrderLineID INT PRIMARY KEY,…

 -- Sales.Orders table
 CREATE TABLE Sales.Orders (
        OrderID INT PRIMARY KEY,..
        );
```

Handle Connections Gracefully

Here: Python + SQL Server

- **Close Connections**: Only establish connections when necessary, and close them promptly after use. Leverage connection pooling for efficiency.
- **Exception Handling**: Implement proper exception handling to gracefully handle connection errors.
- **Example (Python)**:
-
  ```
  import pyodbc
  def execute_sql(query):
      try:
          conn = pyodbc.connect("your_connection_string_here")
          cursor = conn.cursor()
          cursor.execute(query)
          result = cursor.fetchall()
          return result
      except pyodbc.Error as e:
          print(f"Error executing query: {e}")
      finally:
          conn.close()
  # Example usage
  query = "SELECT * FROM Sales.Orders WHERE OrderDate >= ‘2024-04-03’”
  results = execute_sql(query)
  ```

What is a token?

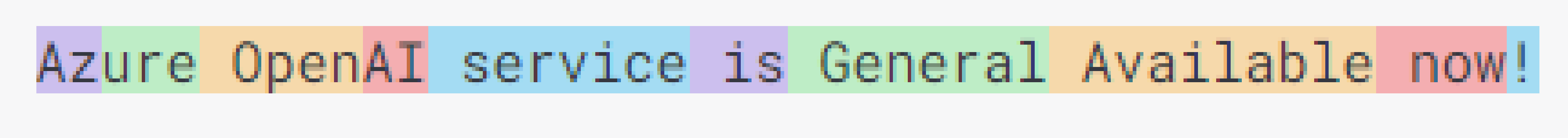


Sentence:

- Azure OpenAI service is General Available now!

Tokens:

- [AZ]-[ure] [Open]-[AI] [service] [is] [General] [Available] [now][!]



Azure OpenAI service is General Available now!

1 token is approximately 4 characters or 0.75 words.

Do you know:
The collected works of Shakespeare are about 900,000 words or 1.2M tokens.

Next Steps

Learning

- Emphasizing continuous learning and upskilling in a fast-changing environment is key to able to push boundaries

Discovering

- Encourage experimenting to uncover what really works and strive for meaningful impact

Deploying

- Applying models to real-life and create value.
  Monitor and finetune, learn and repeat the process.

Resources

- Learn more about Visual Studio Copilot
- Getting started with GitHub Copilot - GitHub Enterprise Cloud Docs
- Microsoft Learn AI Skills Challenge
- Welcome | Learn how to interact with OpenAI models (microsoft.github.io)

Thank you!