



GitHub Copilot - Part 1

By RaKeTe-Technology

April, 16th, 2025

Ralph.Kemperdick@RaKeTeTechnology.com

Cloud Solution Architect, Advanced Analytics AI + Data & Analytics

LinkedIn www.linkedin.com/in/ralphke



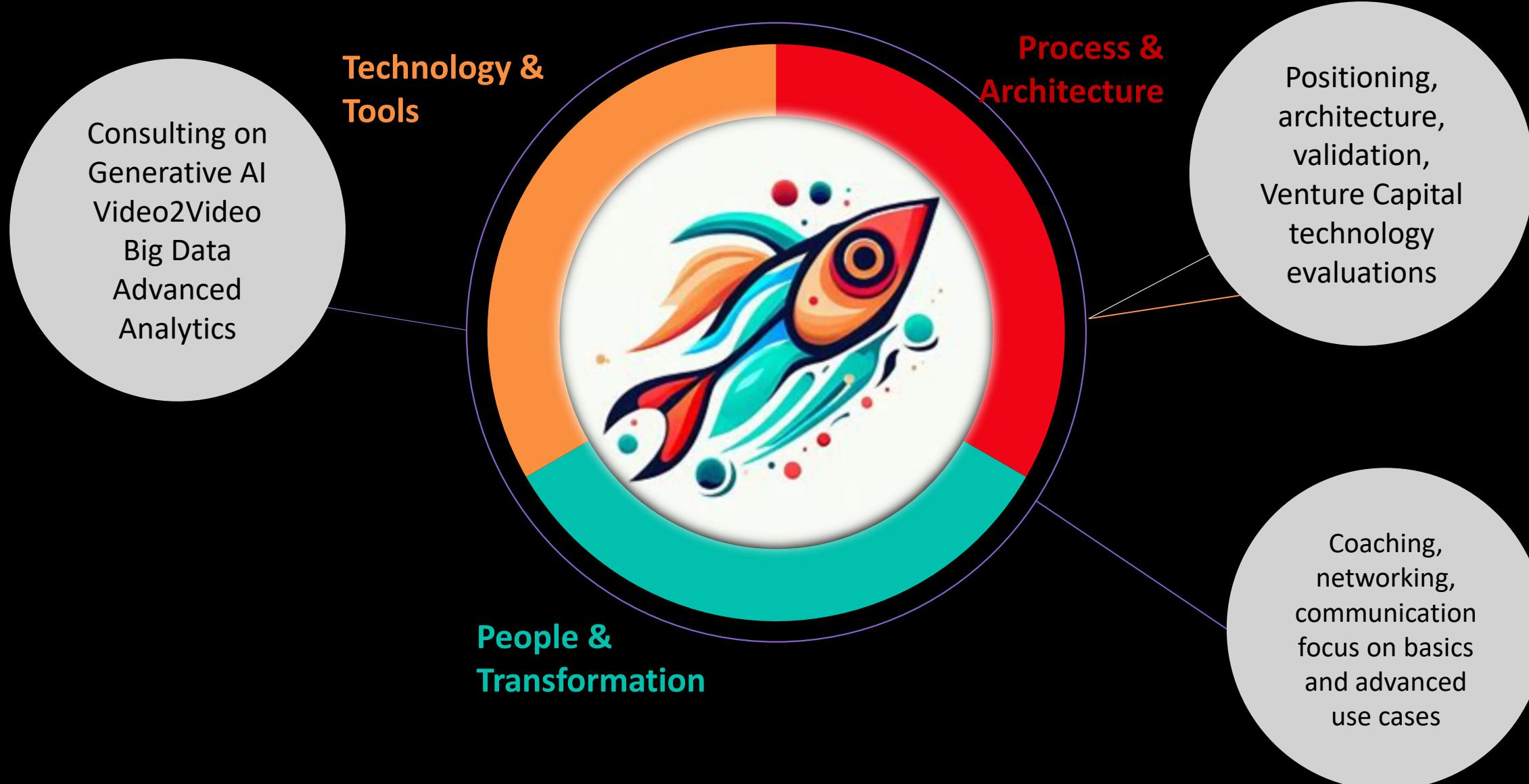
About Ralph Kemperdick

Freelancer: Digital Architect, Big Data & Advanced Analytics

- Coaching customers on Generative AI (OpenAI, ChatGTP), Data Engineering and Business Impact Projects
- Working with diverse teams within MSFT on analytical, competitive and political challenges
- Working with international customers on Data engineering and analytical project for more than 30 years.
- Experienced due to my role in the workers council at Microsoft
- Azure data platform Insider for Big Data and Microsoft Fabric Partner – Running Fabric Academy to educate the teams on the technology
- Board member of the DataMonster.io community
- Bitkom Advisory for Data & AI reviewing/co-authoring whitepapers for European and German politicians and EU related regulations, like the GDPR and the AI-Act.
- Cost and performance optimization of Azure Cloud Services at scale ([CloudXcellence - Control your cloud](#))
- Married, living in Cologne, Germany
- Contact: Ralph.Kemperdick@RaKeTeTechnology.com (RaKeTe-Technology)
- More details: <https://www.linkedin.com/in/ralphke/>
- GitHub: [ralphke \(Ralph Kemperdick\) \(github.com\)](https://github.com/ralphke)



RaKeTe-Technology Offerings



Agenda

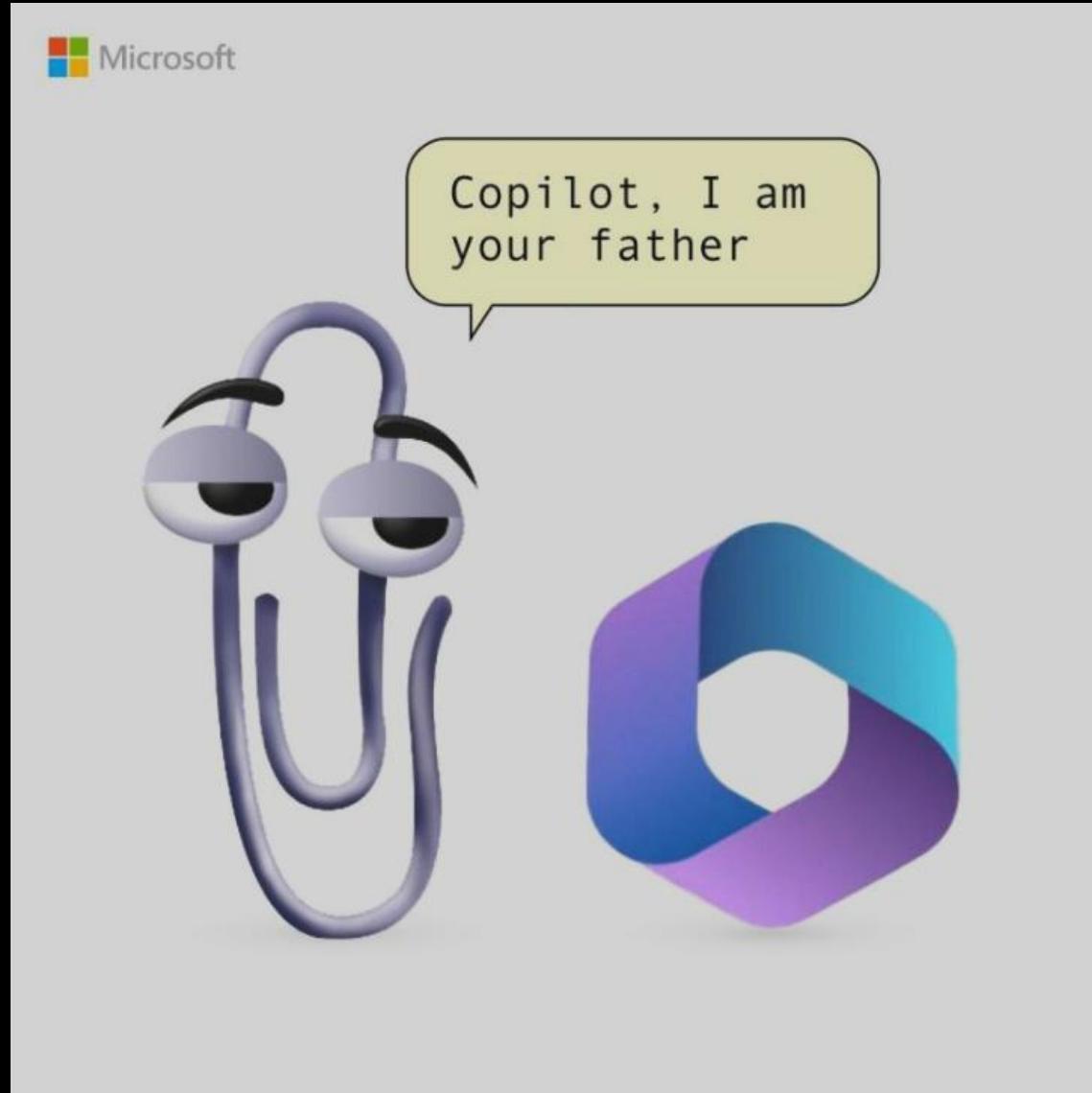
1. Introduction - Where is it coming from?
2. Key Concepts & Best Practices
3. Technology stack
4. Considerations and Limitations
5. GitHub Copilot Demos
6. Q & A



GitHub Copilot HowTo Repro

- In this repro you find several exercises to get familiar with GitHub Copilot
- <https://github.com/ralphke/Copilot-HowTo>

A little bit of history



Artificial Intelligence

Machine Learning

Deep Learning

Generative AI

Agentic AI



Artificial Intelligence

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



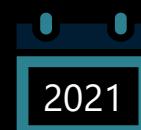
Machine Learning

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



Deep Learning

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



Generative AI

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



Agentic AI

Orchestrate entire workflows via prompts, allowing 24/7 operations and robotic applications.

AI helps across the entire software development lifecycle

1

Code faster with code suggestions

2

Migrate code

3

Faster bug finding and fixing

4

Generate unit test cases

5

Sys admin scripting

6

Code to doc, doc to code



Build



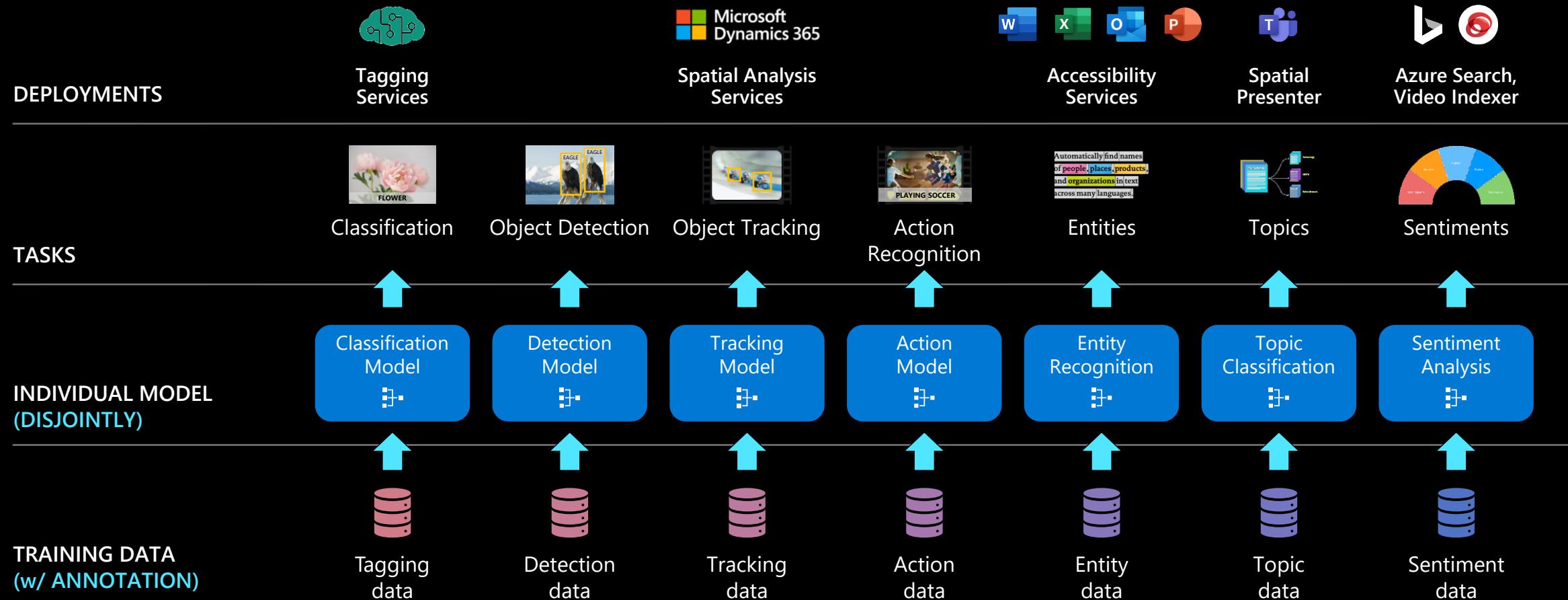
Test



Deploy

Traditional Machine Model Development

High cost and slow deployment—each service is trained disjointly





*Ensure that artificial
general intelligence (AGI)
benefits humanity.*

Groundbreaking research
Latest models
Fast innovation



*Empower every person and
organization on the planet
to achieve more*

Enterprise features
Production SLAs
Data privacy

GPT-4

Codex

DALL·E

ChatGPT



Generative AI

GPT-3/4

Generate and Understand Text

Prompt:

Write a tagline for an ice cream shop.

Response:

1. Scoops of happiness in every cone.
2. Life is short, eat ice cream first.
3. . .

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 2/3

Generate images from text prompts

Prompt:

create a comic style rocket with stylish colors

Response:



ChatGTP

Chat Bot, Avatar, Contact Center Reply

Prompt:

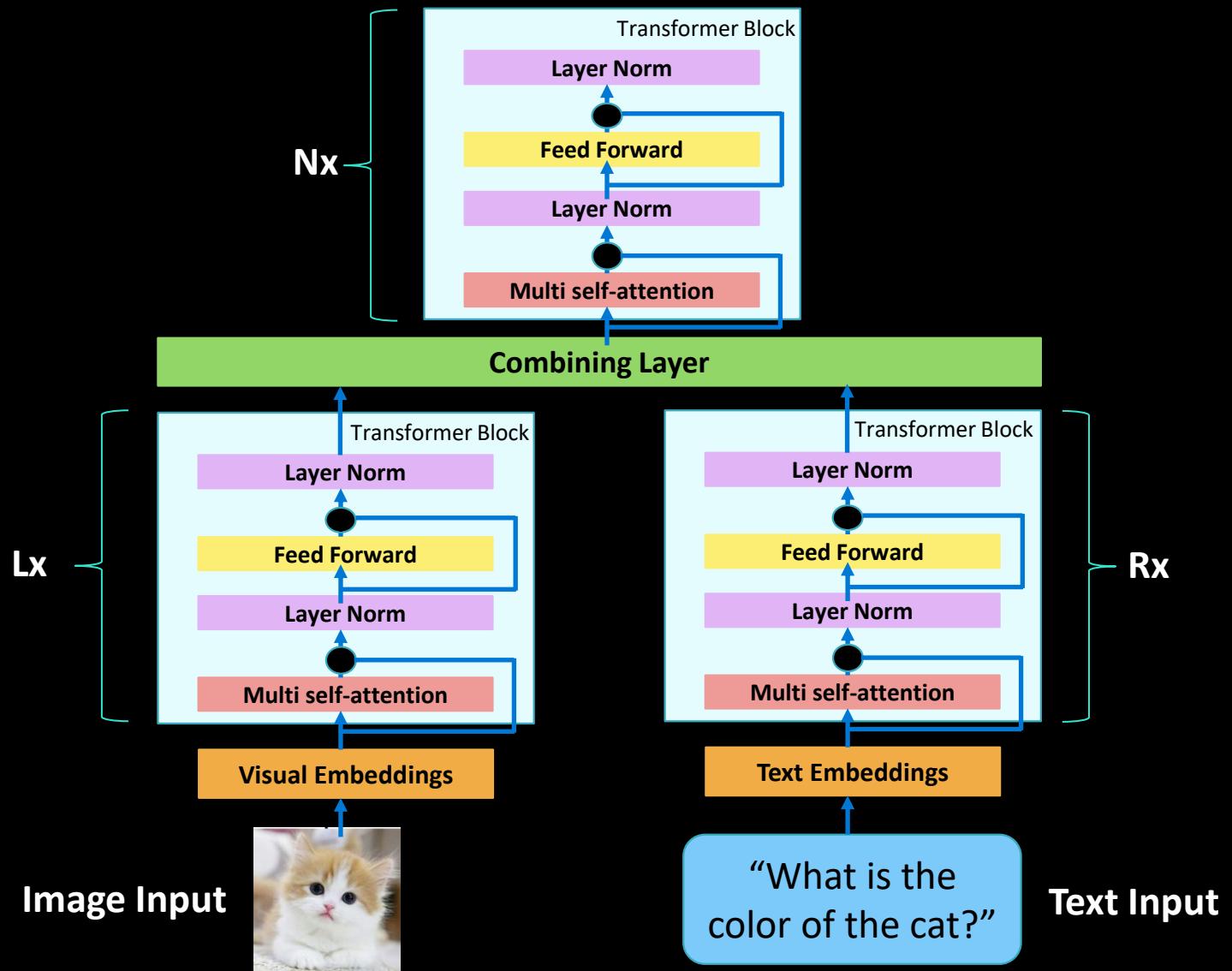
What is the fastest animal on the earth?

The fastest land animal is the cheetah, which can reach speeds of 109.4 to 120.7 km/h

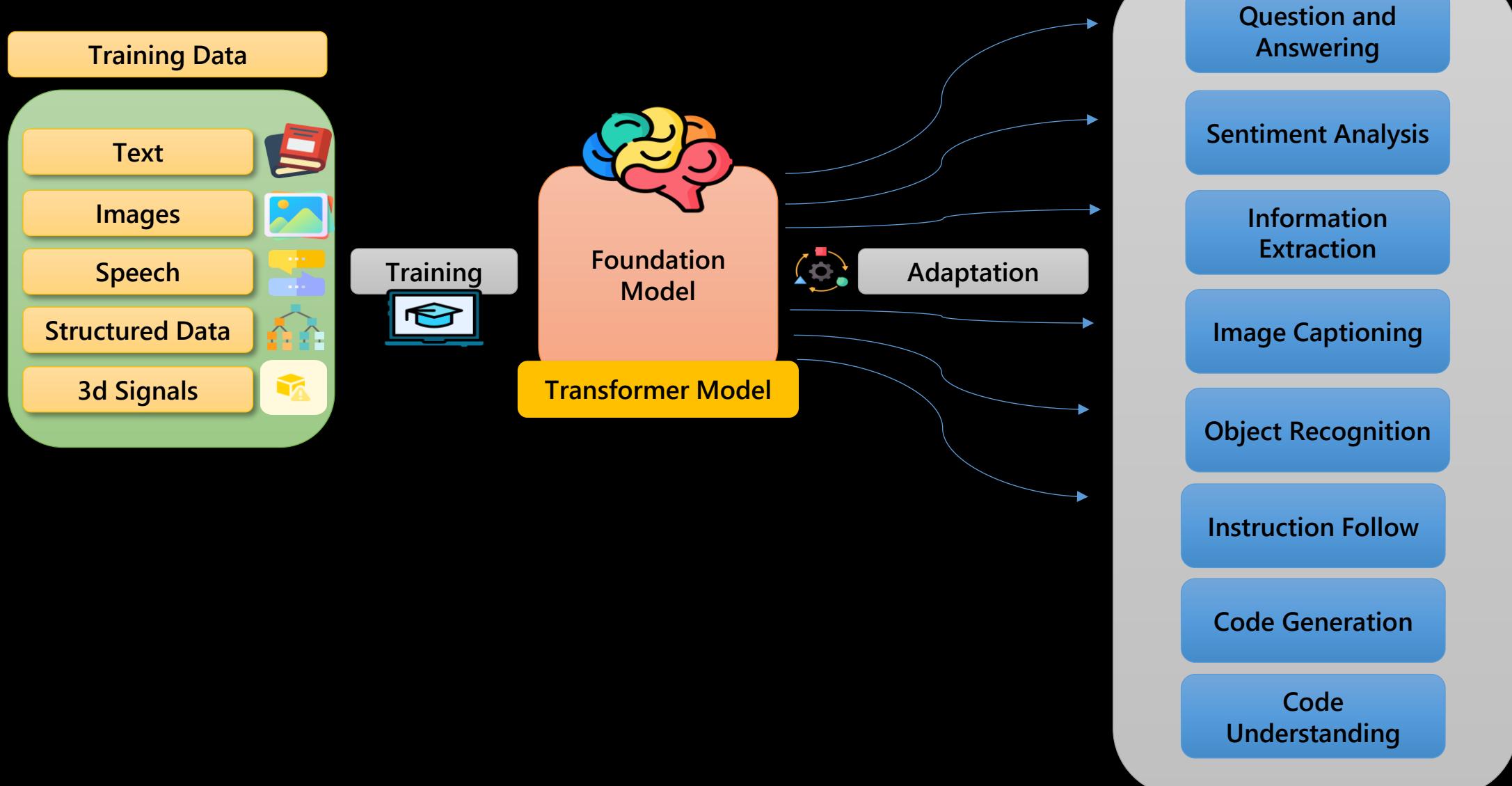
What makes them so fast?

Cheetahs are built for speed. Their slender body, small head, flattened rib cage, and long, thin legs minimize air resistance

Multi-Modal Transformer Architecture



Foundation Models



Next – Small Models (SLMs)

Phi-3 – Family of „small“ language models

Small language model, Best in class performance for size, with frictionless availability

Φ

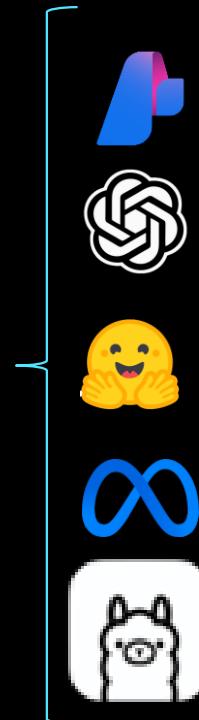
Phi-3-mini-V
(3.8B + 0.3B)

Φ

Phi-3- small
(7B)

Φ

Phi-3-mini
(3.8B)



Azure AI Studio



GPT



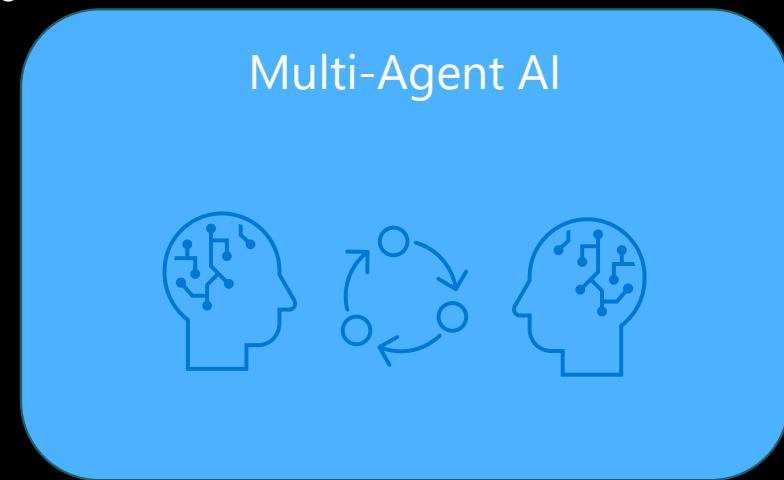
Hugging
Face



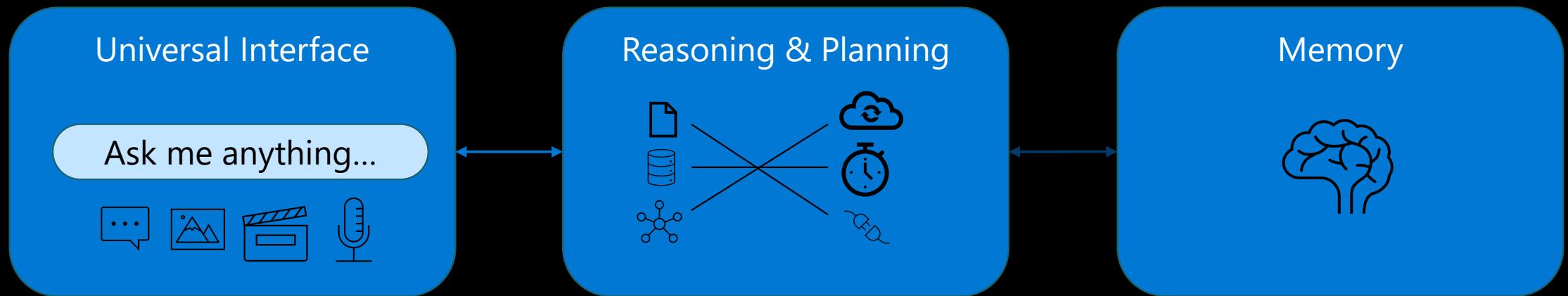
Meta



Ollama



The next step - Copilots + Agents

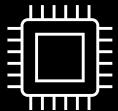


What are agents?

AI designed to perform a task

Tasks can vary in level of complexity and capabilities depending on your need

Simple



Generation

Generate summaries, images, audio, and more with an AI model and inputs.

Generally available



Retrieval

Retrieve information from grounding data, reason, summarize, and answer user questions

Generally available



Advanced

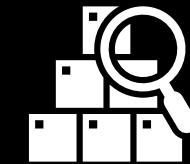
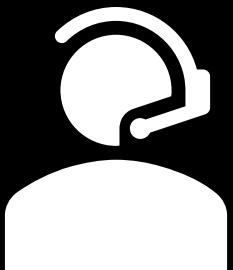
Action

Take actions to automate workflows, and replace repetitive tasks for users

Generally available

A single agent can be used to power a chat bot

Copilot chat bot



Search for developer
relevant information

Or multiple agents can be used together to automate processes

Copilot Agent mode



Request for a task

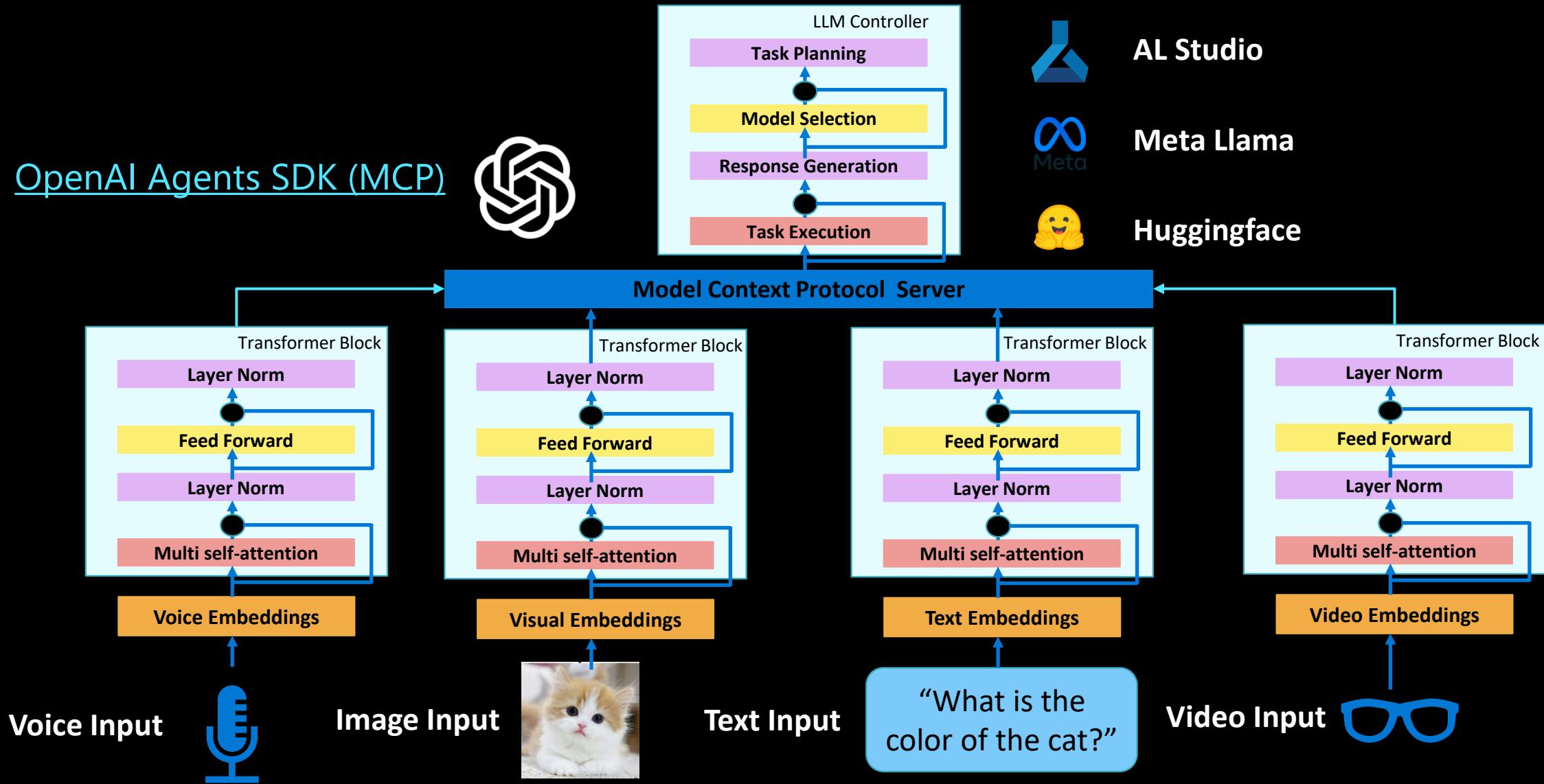
Analyze context and which other agent to ask for help

Create Execution plan for give task and content

create apps from scratch, perform refactoring's, write and run tests

[Model Context Protocol \(MCP\)](#)

Model Context Protocol as Orchestrator



But how can you orchestrate multiple agents?

Azure AI Agent Service

Trust

Customer control over data, networking and security

- BYO – Storage
- BYO – Search Index
- BYO - VNet
- BYO - Thread storage

Choice

Model choice and flexibility with model catalog

Azure OpenAI Service

- Models-as-a-Service
- Llama 3.1-Instruct
 - Mistral Large
 - Cohere-Command

Skills

Rich set of enterprise connectivity

Knowledge



Actions



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



Basic 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.

What is a token and Tokenization?

Transform all text into Vectors

- A very long list of integers
- Differs by algorithm

Typical numbers

- ~10-100k possible tokens
- 1 token ~ = 0.75 words
- Or 100 tokens ~ = 75 words

Typical algorithm

- Byte Pair Encoding

<https://platform.openai.com/tokenizer>

Raw Text

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens and excel at producing the next token in a sequence of tokens.

Tokens

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens and excel at producing the next token in a sequence of tokens.

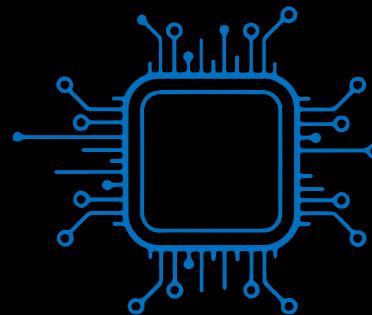
Integers

```
[11505, 20185, 338, 1588, 3303, 4981, 357, 29810, 6412, 284, 355, 402, 11571, 338, 8, 1429, 2420, 1262, 16326, 11, 543, 389, 2219, 16311, 286, 3435, 1043, 287, 257, 900, 286, 2420, 13, 383, 4981, 2193, 284, 1833, 262, 13905, 6958, 1022, 777, 16326, 290, 27336, 379, 9194, 262, 1306, 11241, 287, 257, 8379, 286, 16326, 13]
```

<https://github.com/openai/tiktoken>



OpenAI Codex Model



Public code and text
on the internet

GitHub



GitHub
Copilot Service



Don't fly solo.

Provide editor context

Provide suggestions

Improve suggestions

Private code

JS fetch_pic.js

```
1 const fetchNASAPictureOfTheDay = () => {
2   return fetch('https://api.nasa.gov/mars/rovers/curiosity/images/latest?sol=1000')
3     .then(response => response.json())
4     .then(json => {
5       return json;
6     });
7 }
```

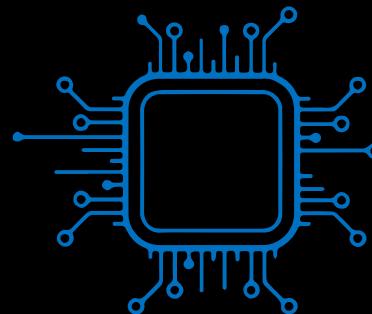
push_to

Copilot



OpenAI + GitHub + [Enterprise]

OpenAI Codex Model



Private code and Context from GitHub Enterprise Repos

GitHub



GitHub Copilot Service

Don't fly solo.



Provide editor context

Provide suggestions

Improve suggestions

Private code

JS fetch_pic.js

```
1 const fetchNASAPictureOfTheDay = () => {
2   return fetch('https://api.nasa.gov/mars/rovers/curiosity/images/latest?sol=1000')
3     .then(response => response.json())
4     .then(json => {
5       return json;
6     });
7 }
```

push_to

Copilot

A screenshot of a code editor window showing a JavaScript file named "fetch_pic.js". The code uses the Fetch API to retrieve a NASA Mars rover image. A GitHub Copilot "Copilot" icon is visible in the bottom right corner of the code editor.

GitHub Platform



- A single integrated enterprise-ready platform
- Industry-best collaborative tools for developers
- Security at every step of the workflow
- Powered by AI



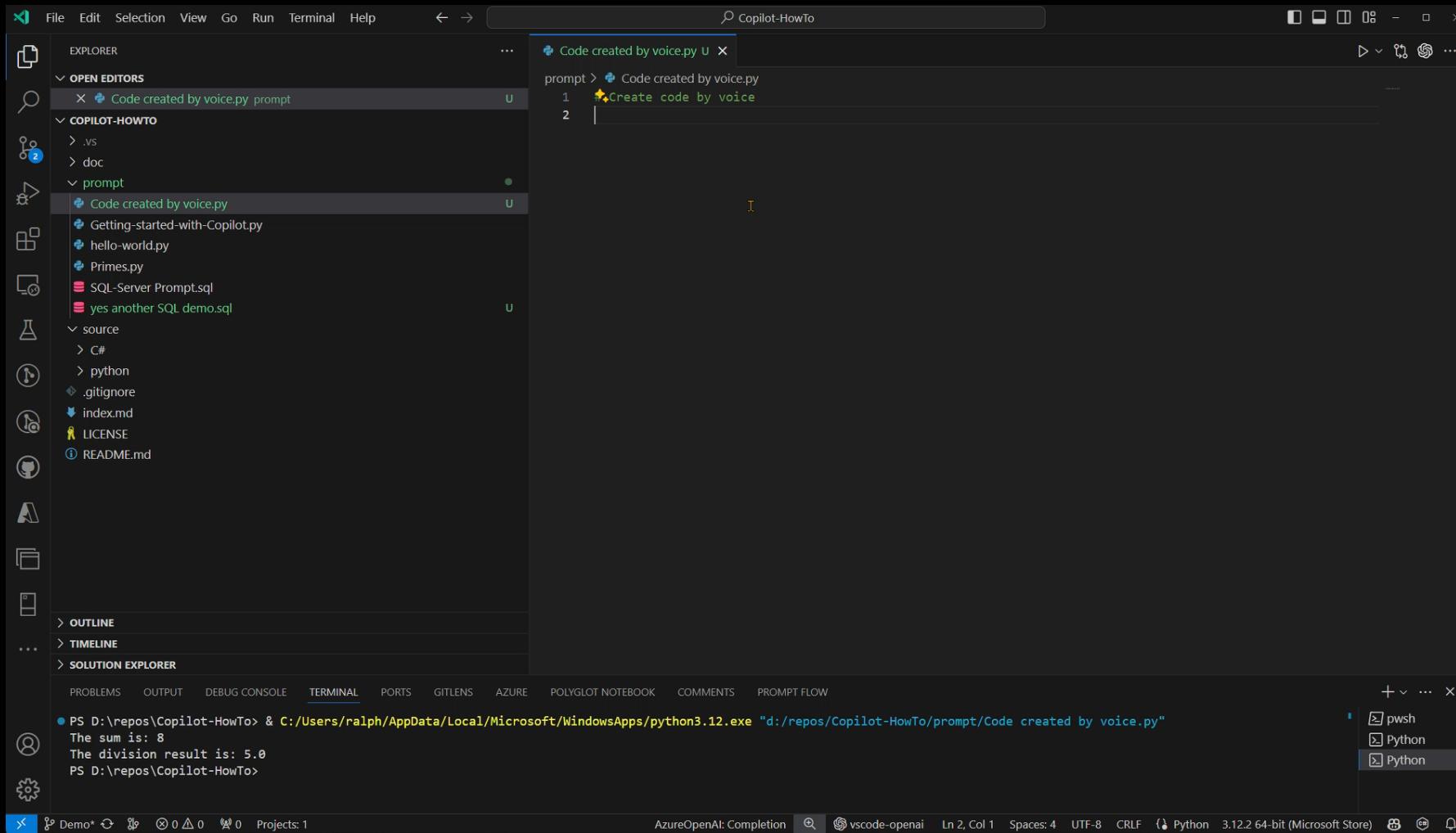
Demo

GitHub Copilot Demo

- Generating code via voice



Programming with your voice



Prompt Engineering



Engineer

DevOps

A.I.



Data solution architect

DevOps

A.I.

Prompt Engineering

In-context Learning: Integrate examples into prompts, enable LLMs to handle new tasks without fine-tuning

Zero-shot - Predicting with no sample provided



One-shot - Predicting with one sample provided



Few-shot – Predicting with a few samples provided



Source: [Prompt engineering - Wikipedia](#)

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

| | | |
|---|------------------------------|-------------------------|
| 1 | Translate English to French: | <----- task description |
| 2 | Cheese => | <----- prompt |

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

| | | |
|---|------------------------------|-------------------------|
| 1 | Translate English to French: | <----- task description |
| 2 | Sea otter => loutre de mer | <----- example |
| 3 | Cheese => | <----- prompt |

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed

| | | |
|---|---------------------------------|-------------------------|
| 1 | Translate English to French: | <----- task description |
| 2 | Sea otter => loutre de mer | <----- examples |
| 3 | Peppermint => menthe poivre | <----- |
| 4 | Plush giraffe => girafe peluche | <----- |
| 5 | Cheese => | <----- prompt |

Effective Prompts for optimal Results

| Clarity and Precision | Provide Context | Open Questions | Structured Requests | Provide Examples | Define the Goal | Seek Feedback | Experiment with Variations |
|--|--|---|---|---|--|--|---|
| <ul style="list-style-type: none">Formulate your questions or instructions clearly and precisely. Avoid ambiguous terms. | <ul style="list-style-type: none">Give enough background information so that the context is understood. This helps in obtaining more relevant answers. | <ul style="list-style-type: none">Ask open-ended questions that allow for detailed answers, rather than yes/no questions. | <ul style="list-style-type: none">Use lists or numbered points to clearly outline your requirements. For example: "Name three advantages of..." | <ul style="list-style-type: none">If you are looking for specific information, provide examples to clarify what you mean. | <ul style="list-style-type: none">Explain what goal you are pursuing with your request. This helps to tailor the answer accordingly. | <ul style="list-style-type: none">If you do not receive the desired results, reflect on your prompt and adjust it. | <ul style="list-style-type: none">Try different formulations and approaches to find out which works best. |

Chain-of-Thought Prompting (CoT)

- Chain of Thought
 - Is the process of breaking down a complex topic in multiple, smaller steps and asking the AI to refine its responses
 - It instructs the AI to articulate its reasoning step by step before delivering a response.
 - Explicitly structure the prompts to mirror expert thought processes, breaking down complex tasks for better performance.
 - For example, first prompt the AI to output numerous ideas and then asked it to refine and narrow down those ideas, explaining its reasoning at each step.

Sample Prompts - Problem Definition

- You are an incredibly smart and experienced senior developer assistant asked to gather information to help analyze the following problem: [Insert Problem Statement]
 1. Introduce yourself to the team and let them know that you want to help the team begin their development process.
 2. Ask for any documents which might help with the research. Then ask a series of questions 2-3 about the problem (ask one at a time and wait for a response). Also suggest responses or offer up multiple-choice responses if appropriate; if applicable, provide an all or none of the above option.
 3. The goal is to narrow down your development focus.
 4. Then gather what information you can to try and answer those questions using the documents and what you know. Just do it. Don't just say you'll do it.
 5. You can also suggest other avenues for exploration to help analyze the task.

Link to [Best Practices](#)

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,..  
);
```

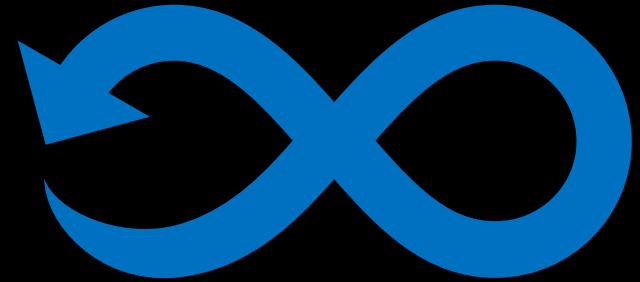
Handle Connections Gracefully

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)
```

Prompt Engineering & DevOps



- Prompt Library ([The Big Prompt Library](#))
- [Prompty](#)
- [Promptflow](#)
- [Langchain](#)
- [Semantic-Kernel](#)
- [What's New in AutoGen? | AutoGen \(microsoft.github.io\)](#)

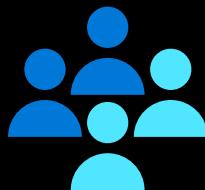
Codex Models



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

```
19     screenshot = ImageGrab.grab()
20     # Convert to text
21     text = image_to_string(screenshot)
22     # Parse text for email addresses
23     emails = re.findall(r'[\w\.-]+@[\\w\.-]+', text)
24     return emails
25
26 def validate(addresses):
27
28
```

Alternative: Continue – for Codestral, Claude, ...

Open-Source Plugin VS Code and JetBrains for different LLMs

- VS Code Extension “Continue” can generate code from natural language
- Supports Mistral Codestral, Claude
- Get involved at [ContinueDev](#)

Support multiple programming languages

- 80+ Programming Languages

Multiple tasks:

- Chat makes it easy to get help from an LLM without needing to leave the IDE
- Autocomplete provides inline code suggestions as you type
- Edit is a convenient way to modify code without leaving your current file
- Actions are shortcuts for common use cases

Use Cases

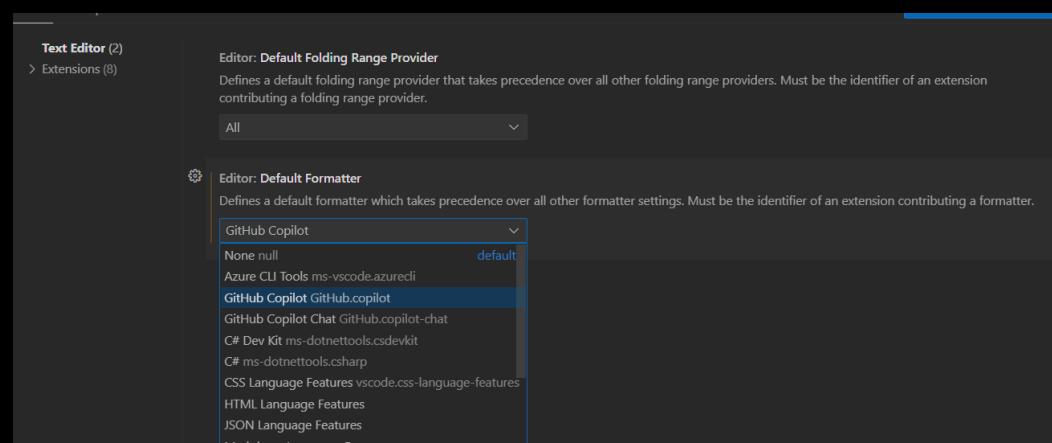
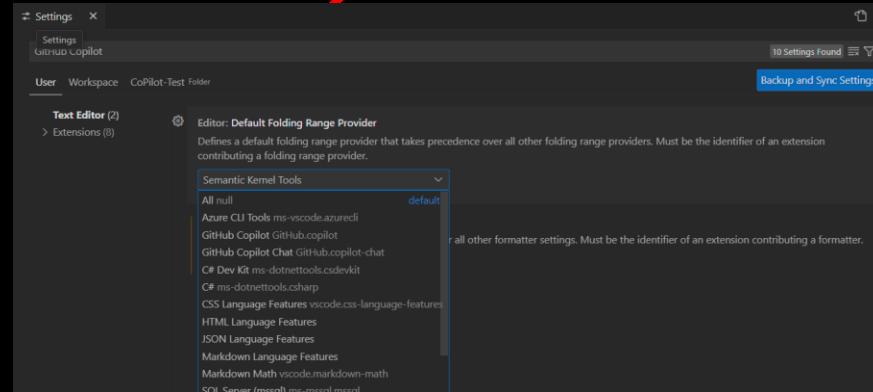
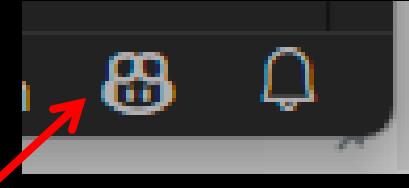
- Natural Language to Code
- Documenting code (comments)
- Refactoring
- Code to Natural Language
- Natural Language to SQL

How to configure Copilot

Within Visual Studio Code:

Click on the copilot icon in the lower right:

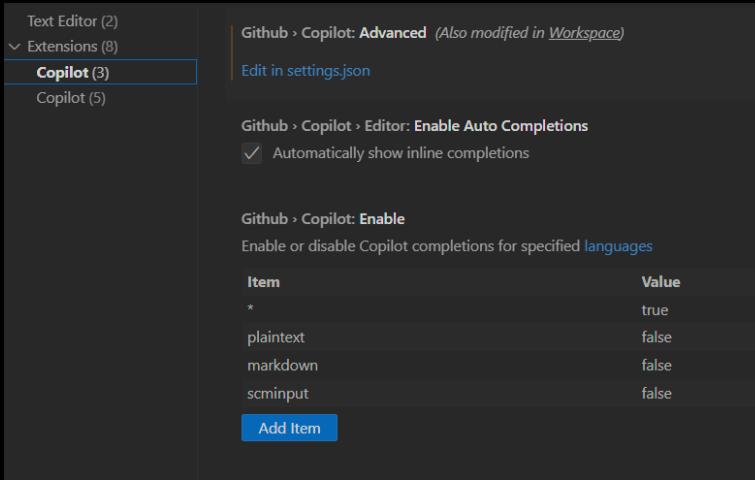
Editor settings (Range Provider)



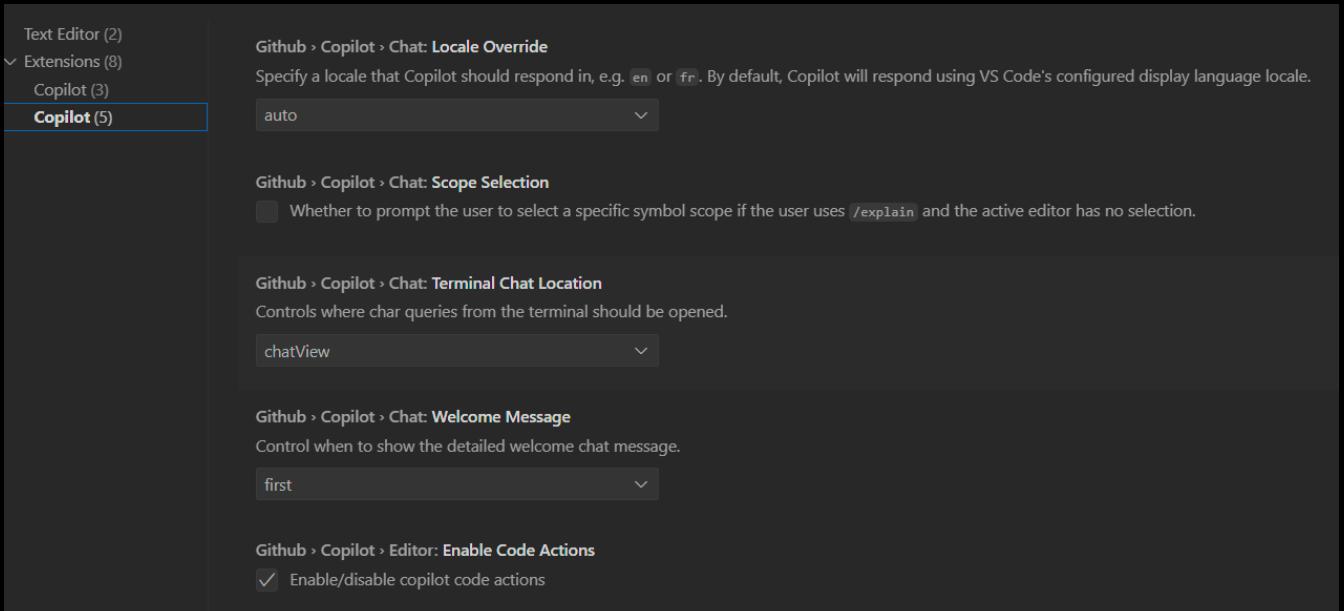
Default Formatter:

How to configure Copilot in VS Code

Copilot Auto Completions:



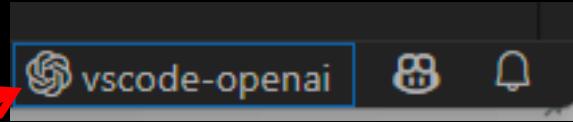
Scope settings:



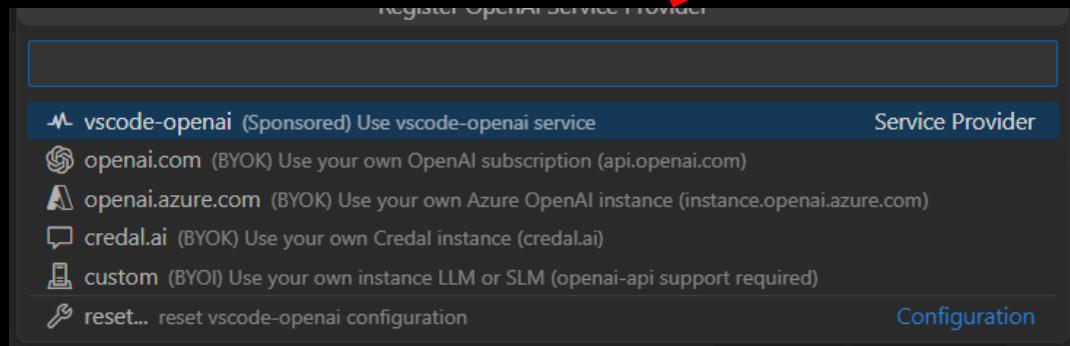
How to configure The Service for Copilot?

Within Visual Studio Code:

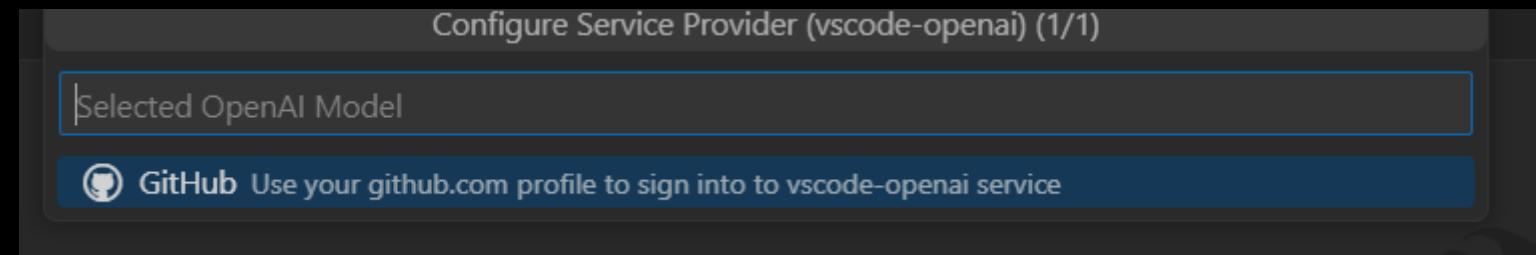
Click on the OpenAI icon in the lower right:



Select Service Provider:



GitHub account for example:



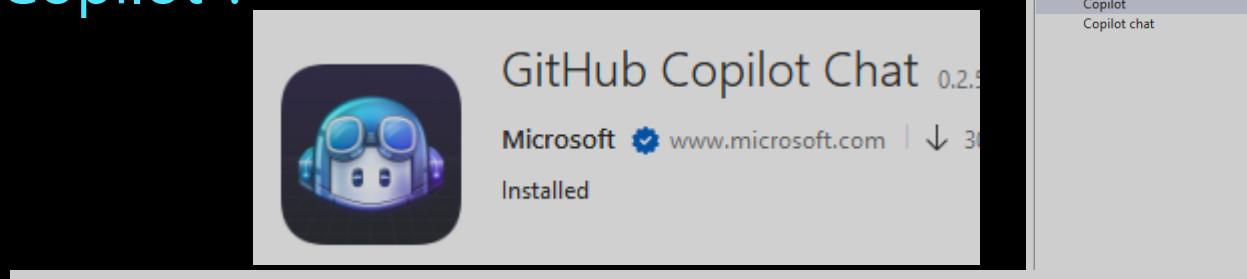
How to configure GitHub Copilot

Within Visual Studio 2022:

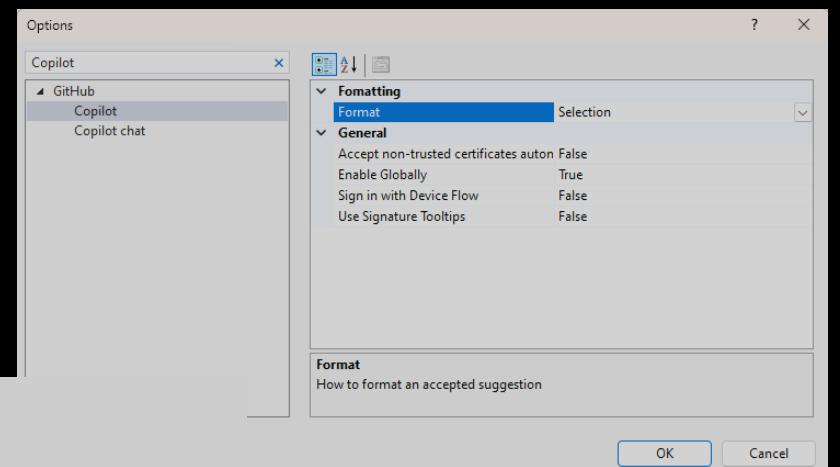
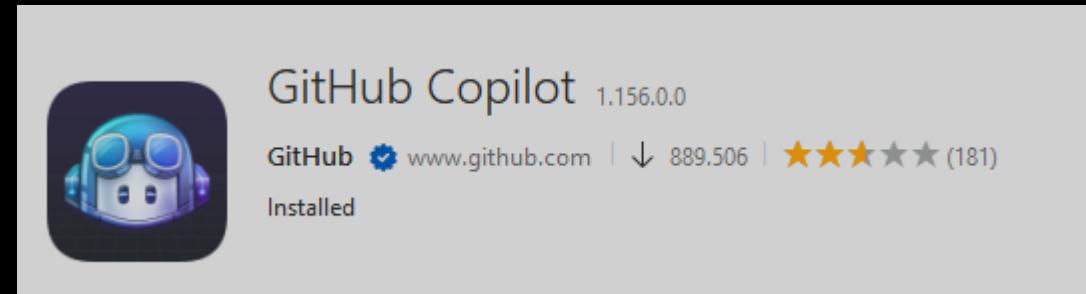
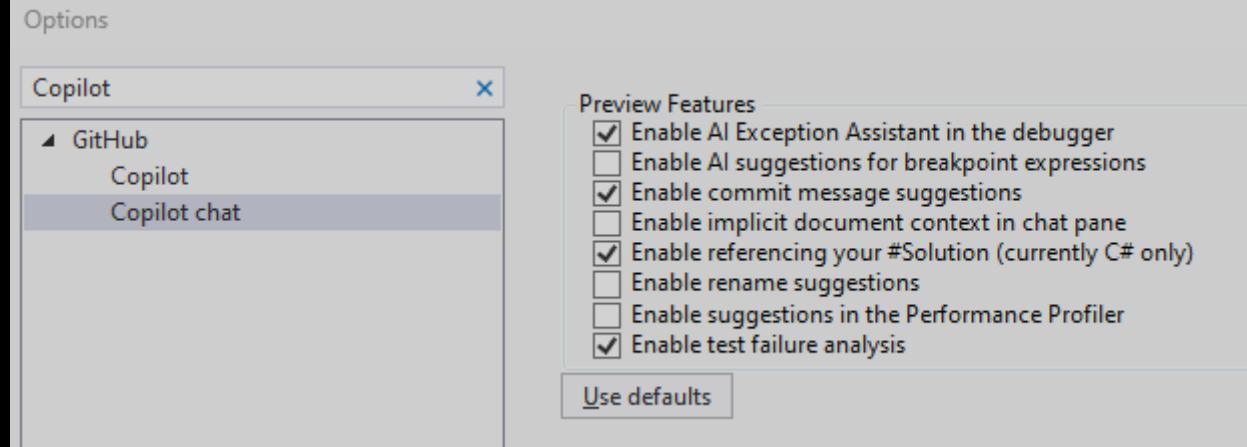
Install GitHub Copilot Now integrated!

Under Tools -> Options

Search for "Copilot":



Copilot Chat:



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

**Learn More at
aka.ms/upgrade/copilot/dotnet**

**Recent announcements
<https://devblogs.microsoft.com/visualstudio/>**

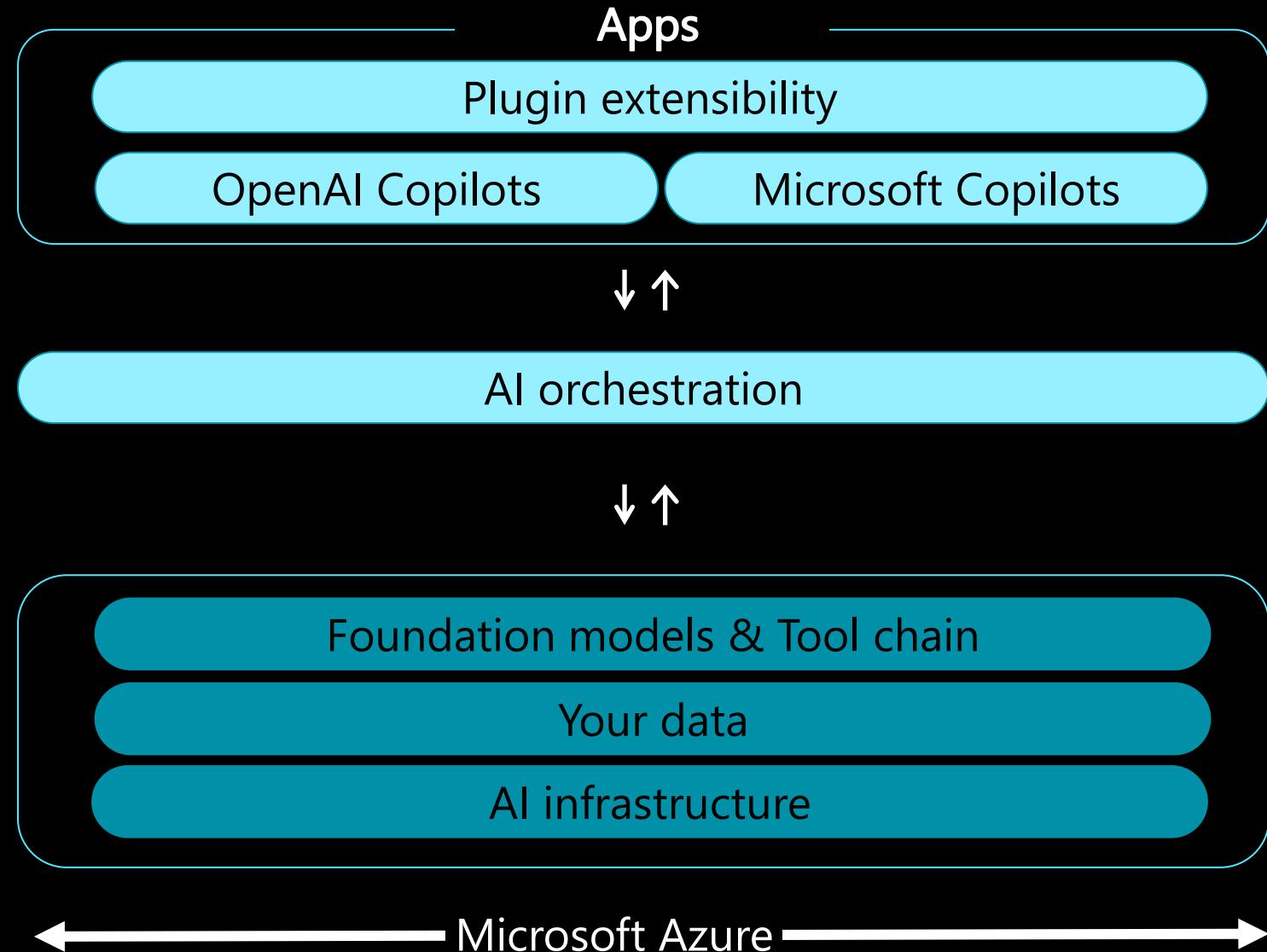
GitHub Copilot CLI

- 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

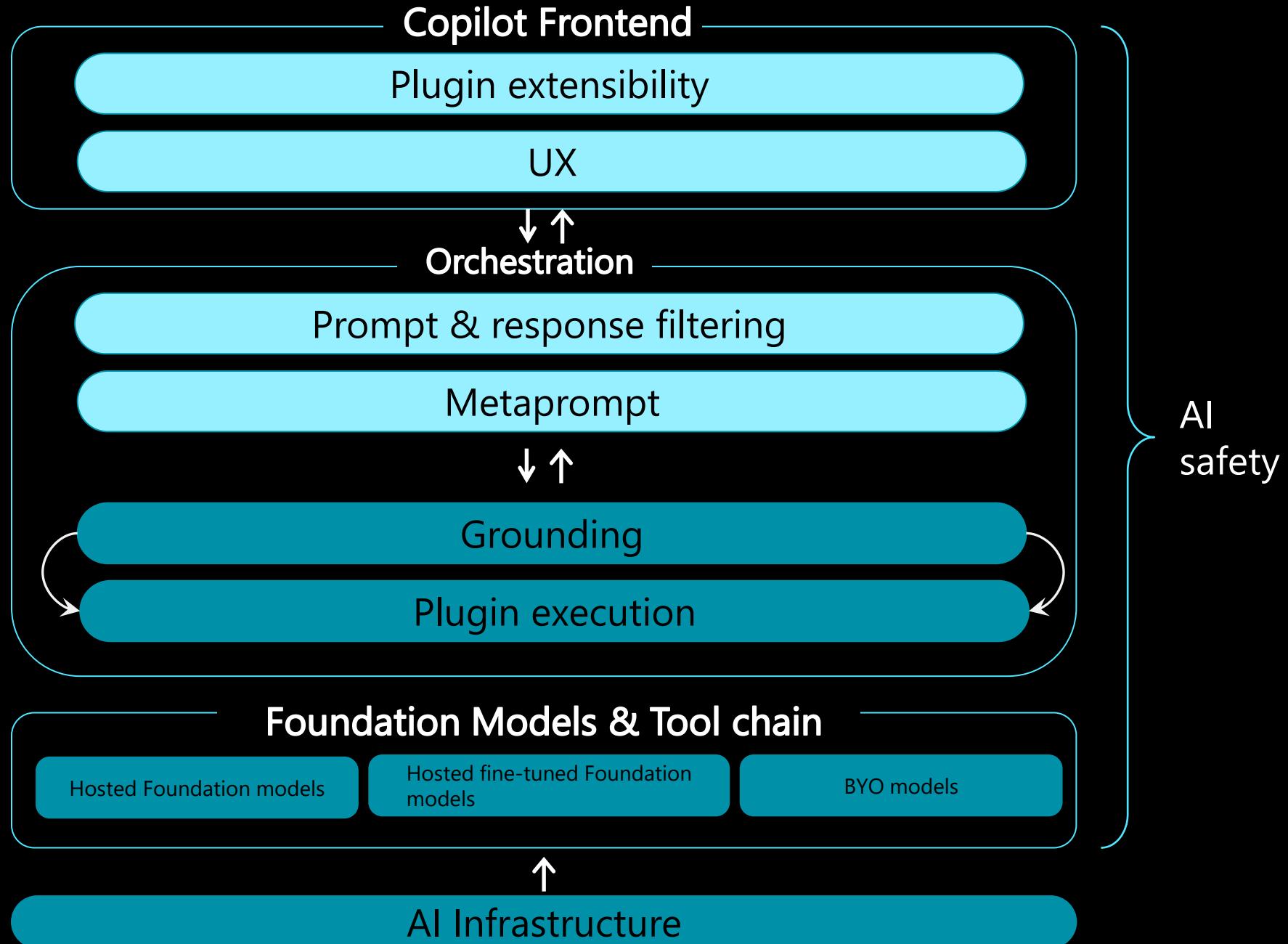
GitHub CoPilot for CLI configuration

```
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
```

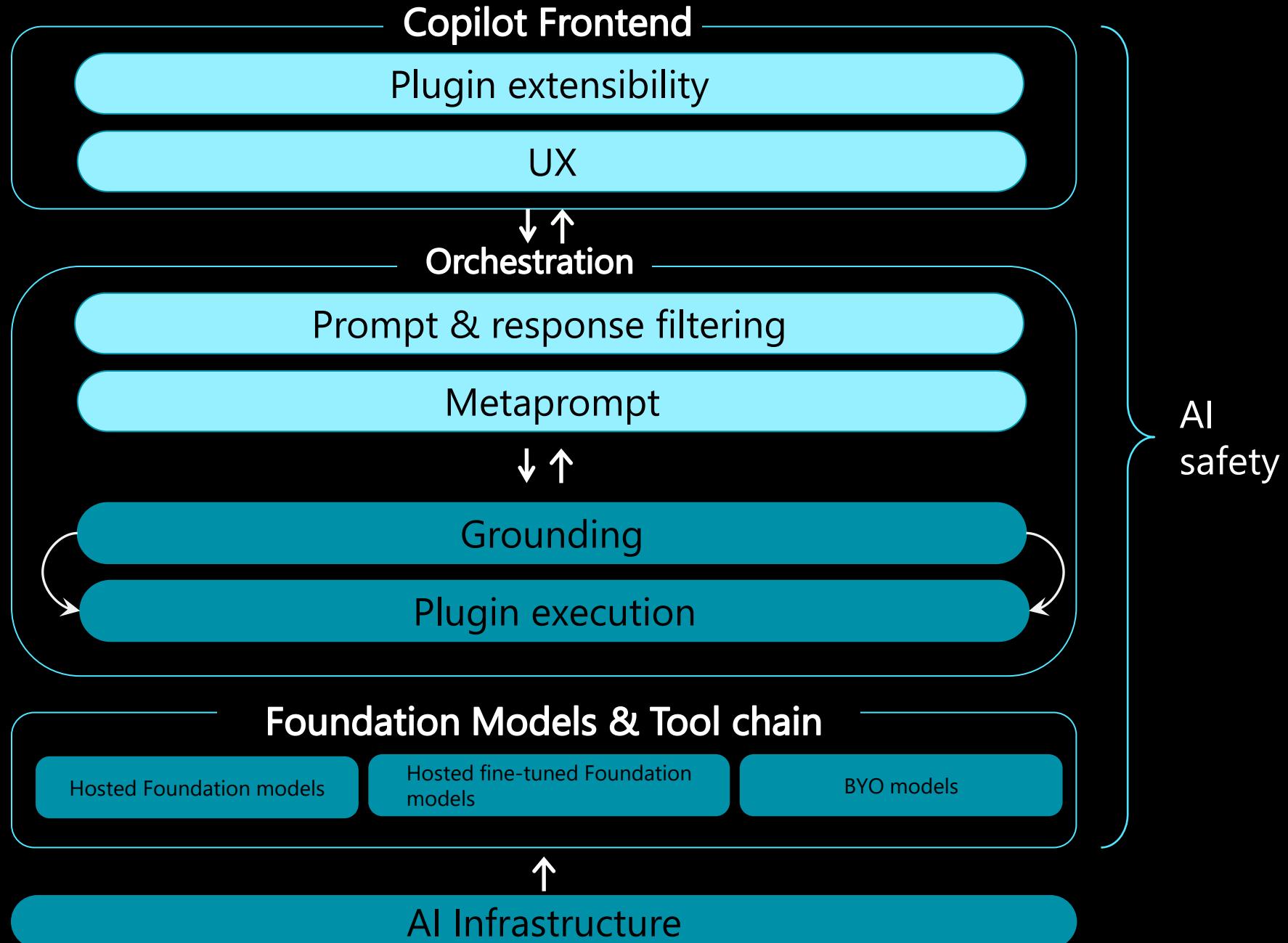
Copilot stack



Copilot stack



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 shadowing¹.](#)



Biased Language and Libraries

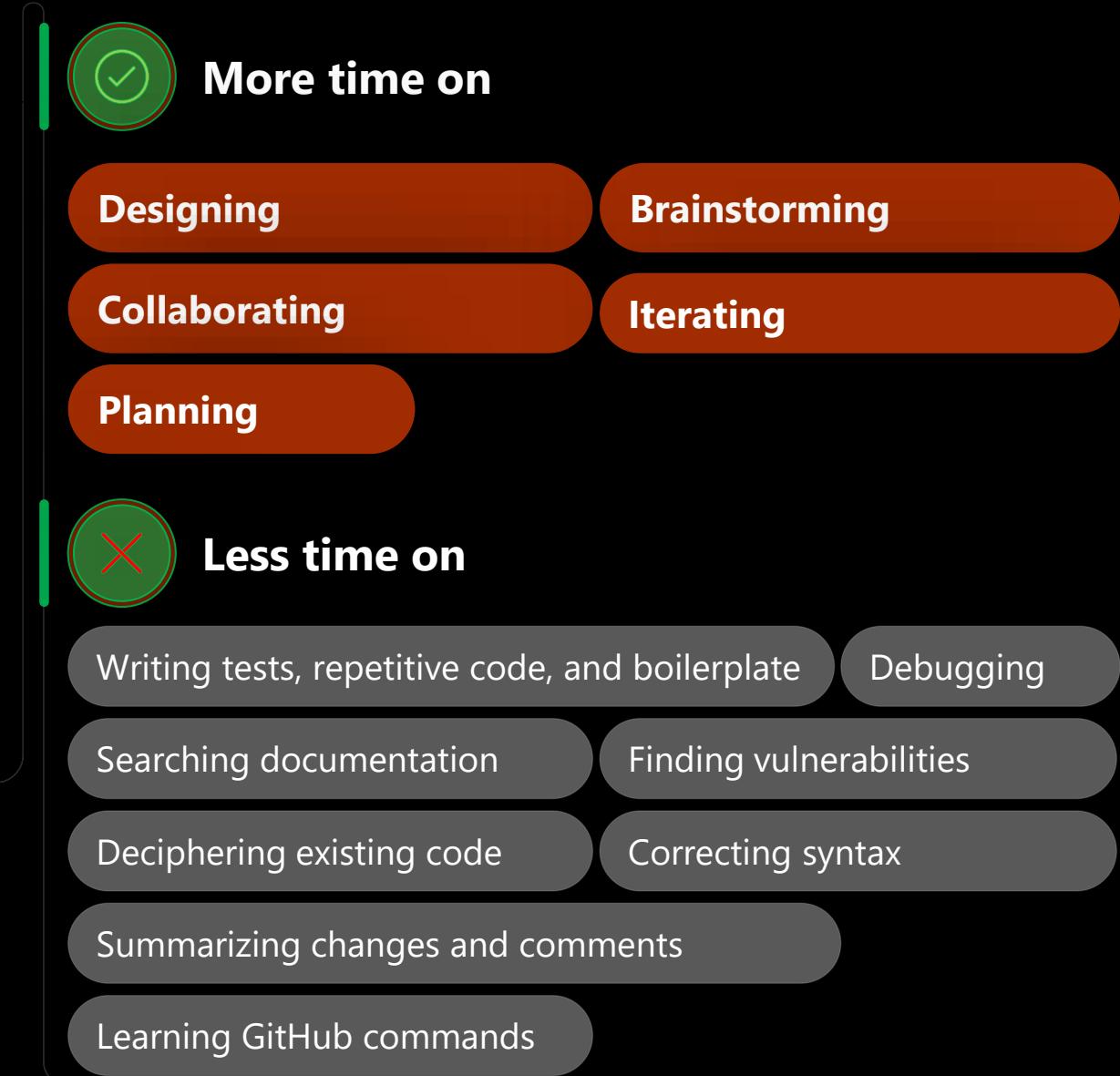
Copilot tends to favor certain libraries or frameworks. [For example, in Python, it heavily leans toward using SQLAlchemy, which may not align with your preferred stack².](#)



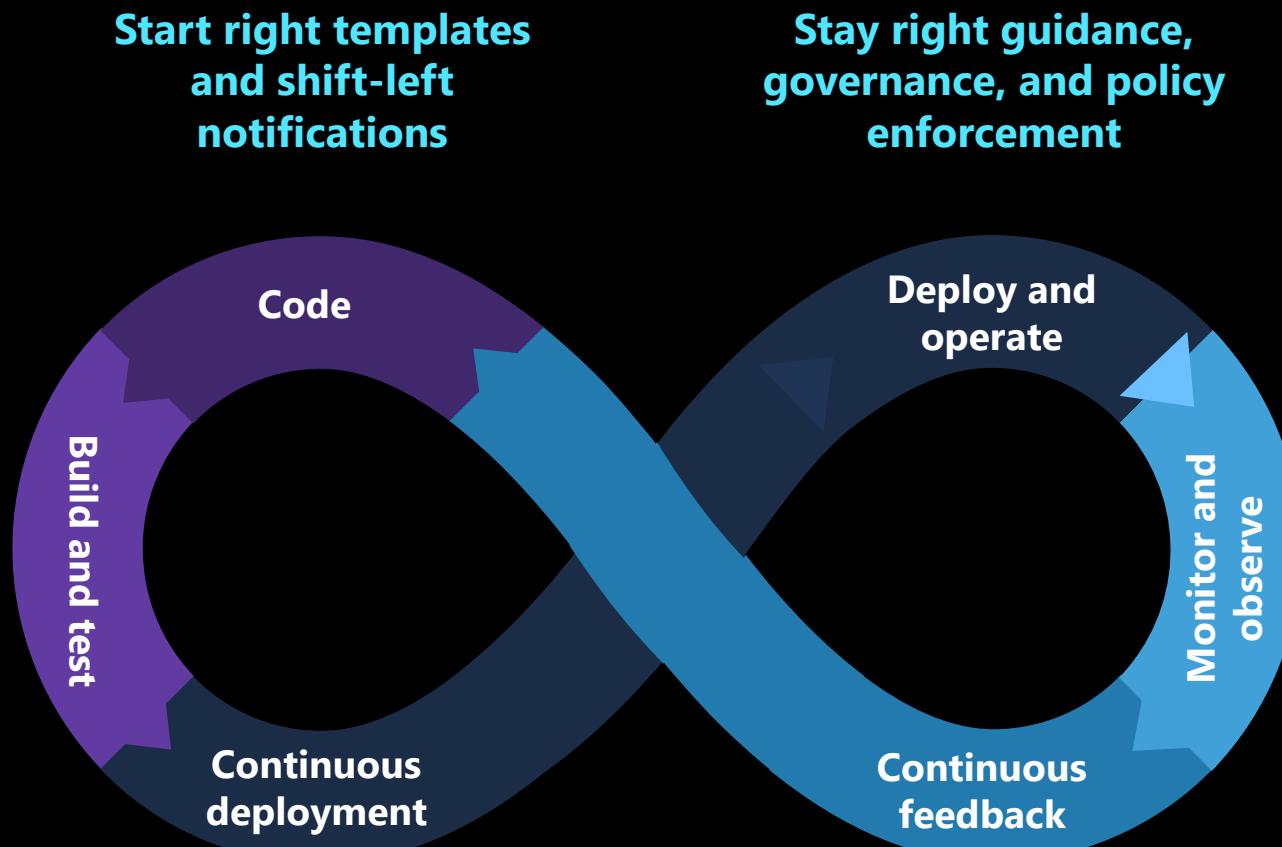
Single Context

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 scenarios².](#)

GitHub Copilot lets developers focus on what matters most



Platform engineering enables DevOps at scale



Next gen developers



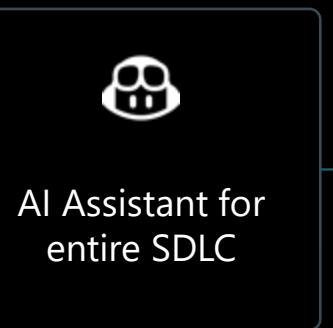
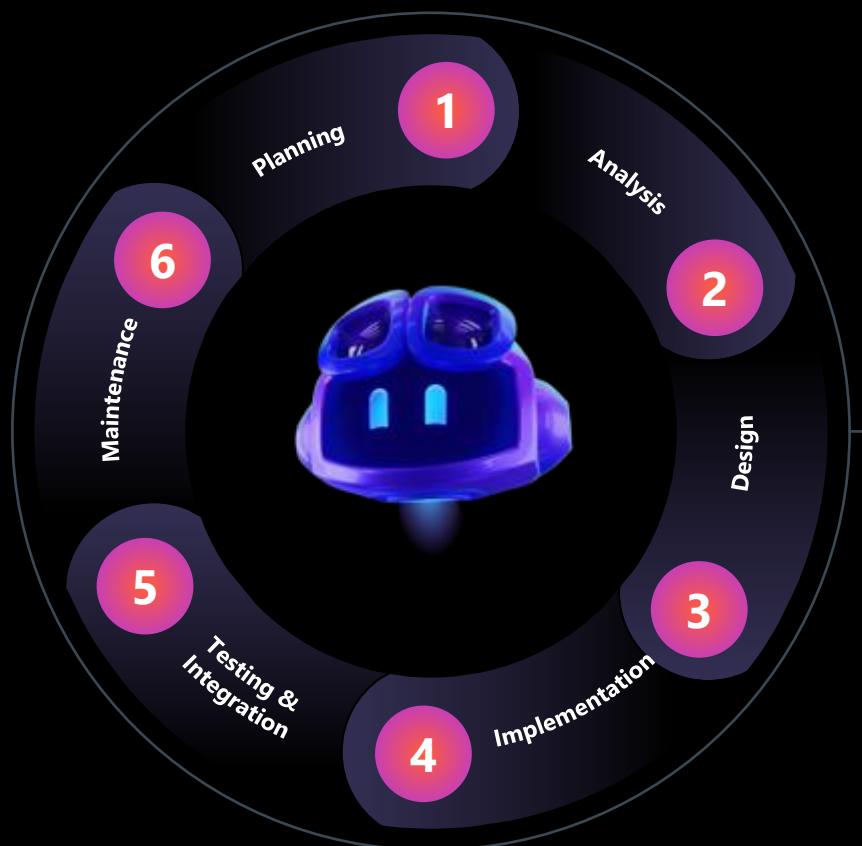
Build using AI
AI-assisted onboarding,
development, and deployment



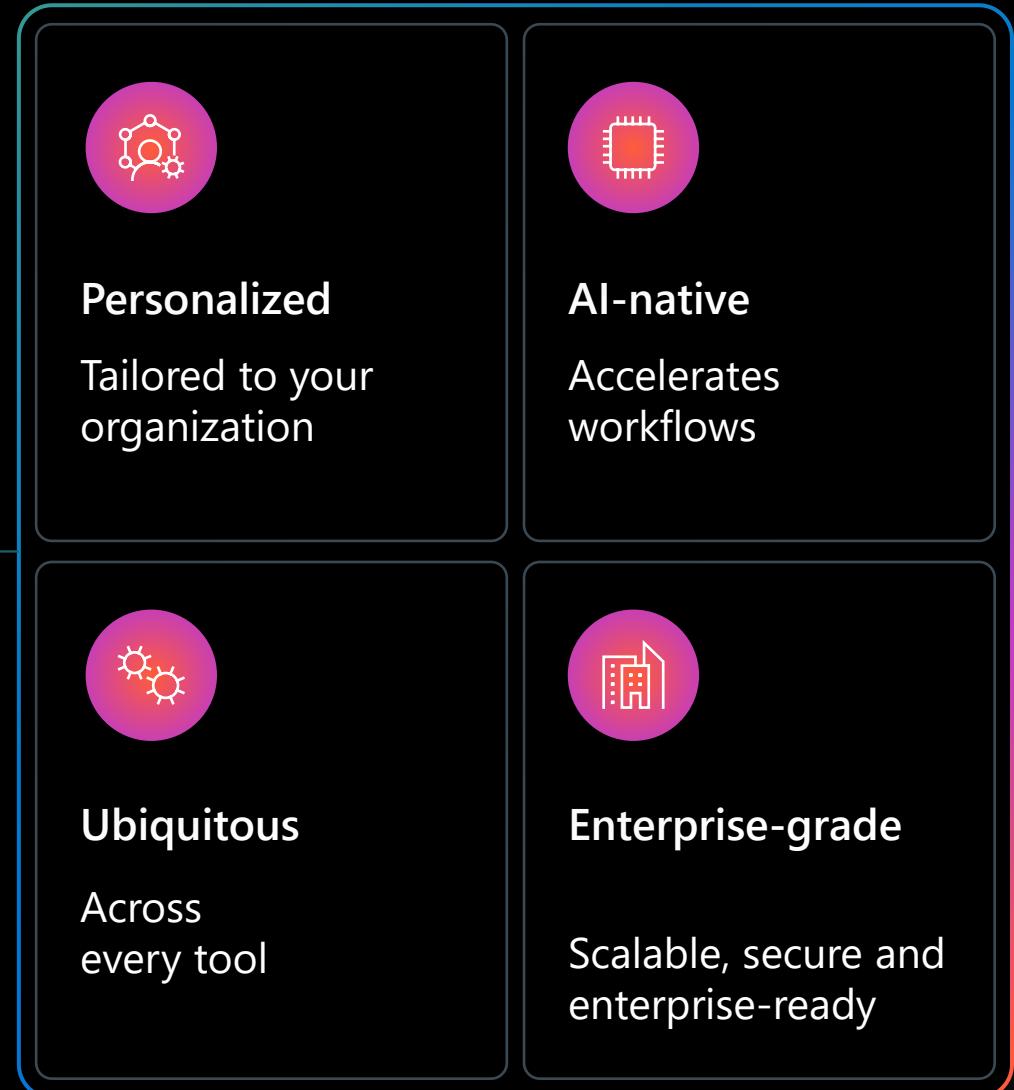
Build AI into apps
Create increasingly
intelligent applications



Leading the AI Revolution in Developer Experience



AI Assistant for
entire SDLC



Public preview

GitHub Copilot workflows in VS Code



Multi-model choice

Powerful model choices
to pick from



Edit mode

Iterate fast and seamlessly
with multi-file editing



VS Code

extensions for Copilot

Speed up more tasks with slash
commands and add new ones with
Copilot-powered VS Code extensions

Announcement

Making Azure easier than ever

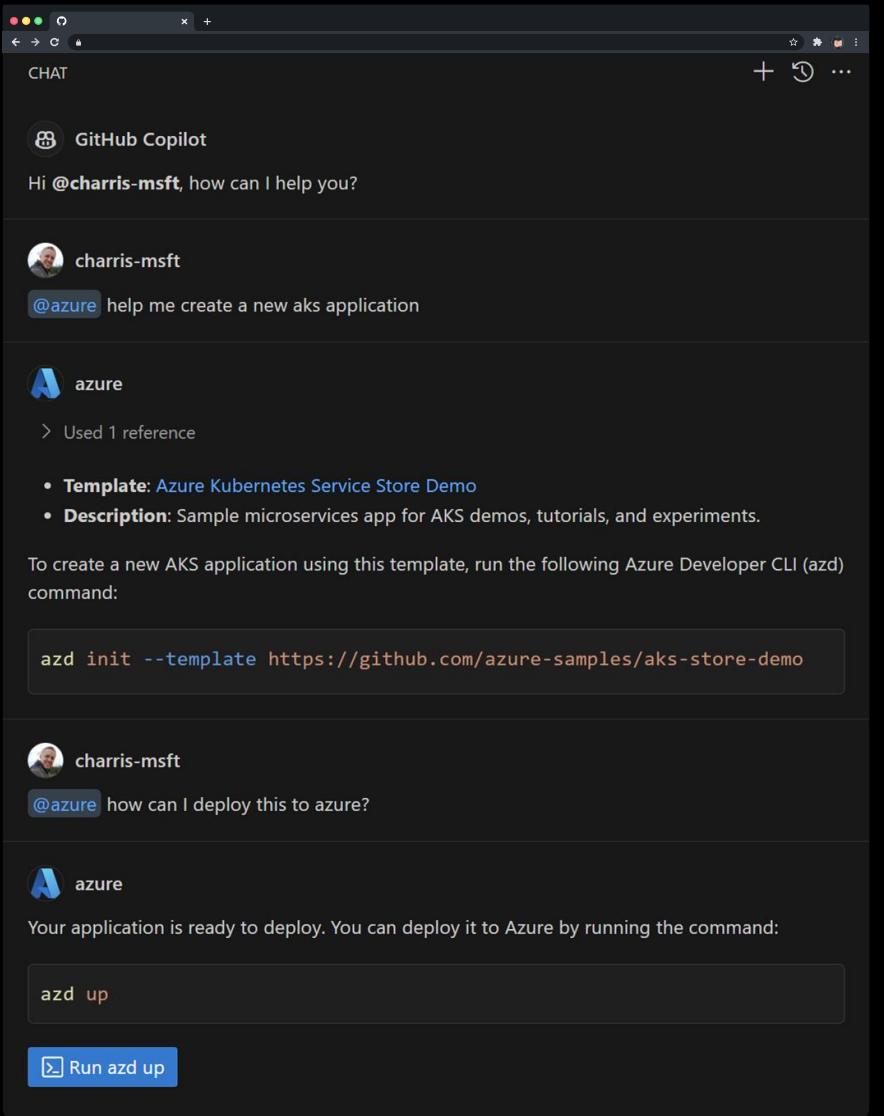
GitHub Copilot for Azure public preview

Learn about the services Azure provides and how to use them.

Deploy app templates to kickstart development.

Diagnose and resolve app issues quickly.

Retrieve information about your services and costs.

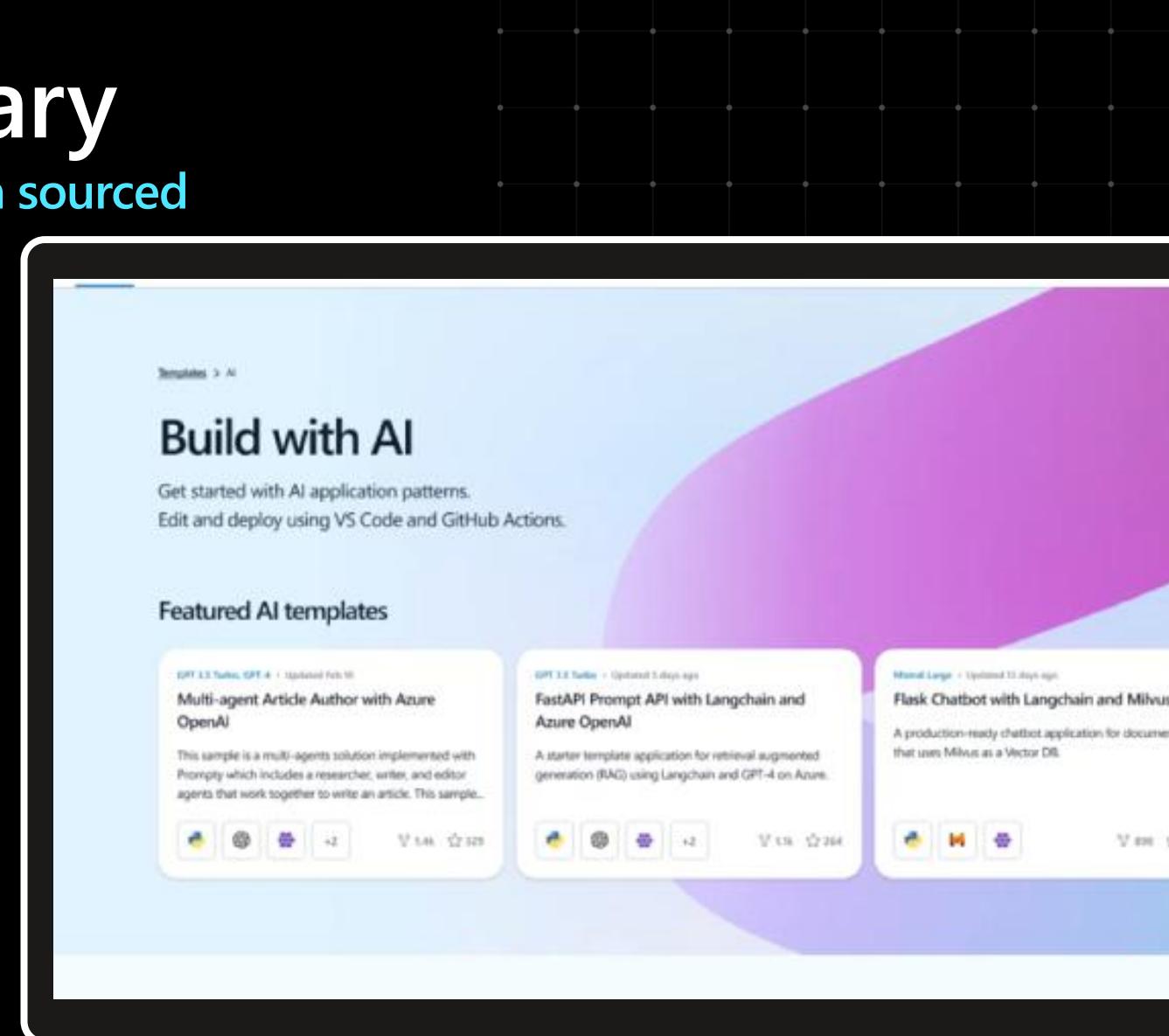


<https://aka.ms/GetGitHubCopilotForAzure>

AI App template library

Azure AI application template gallery open sourced

- Curated repos for common AI app patterns
- Codified best-practices for evaluating, deploying, and monitoring production-quality apps
- Application repo includes both application code and infrastructure as code files
- Including customizable infrastructure as code assets and CI/CD pipelines.
- Deployment for each sample application to Azure using VS Code and GitHub Actions



<http://aka.ms/ai-apps>

GitHub Enterprise Cloud

Multi-tenant, enterprise
SaaS deployment option
for GitHub, running on
Microsoft Azure



Self-provision from Azure Portal



Enterprise Managed Users (EMU)
configuration option



Powered by Azure for advanced security, business
continuity, and disaster recovery features



GitHub Enterprise Cloud with data residency
available in the US and EU**

**More regions coming soon

New GitHub Advanced Security capabilities

Help customers reduce security debt



GitHub Copilot Autofix



Security Campaigns



Copilot Autofix for Code
Scanning Partner Tools

Summary of predictions

There will be a change in the way Developers and DevOps will work

Productivity will increase by an order of magnitude

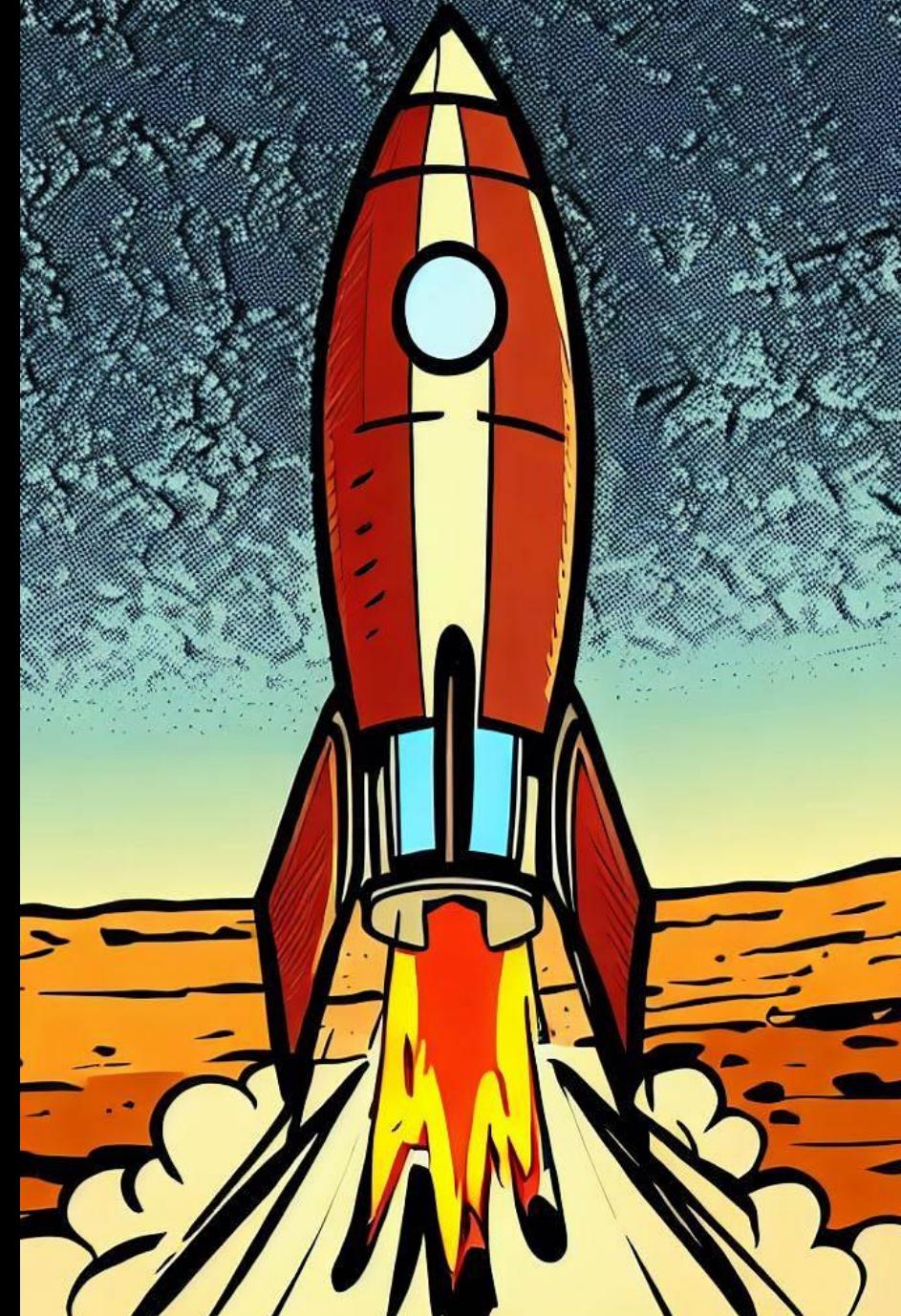
Learning new stuff will become much easier

Tackling complex challenges will get easier but at the edge still need expertise

Attacks will come in new shapes and unexpected forms

Creativity will boost

Learning the basics will be relevant for a long time to come



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

Todays Presentation

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

Prompt Engineering

- [prompt engineering: Onlinekurse, Trainings und Tutorials auf LinkedIn Learning](#)
- [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!

