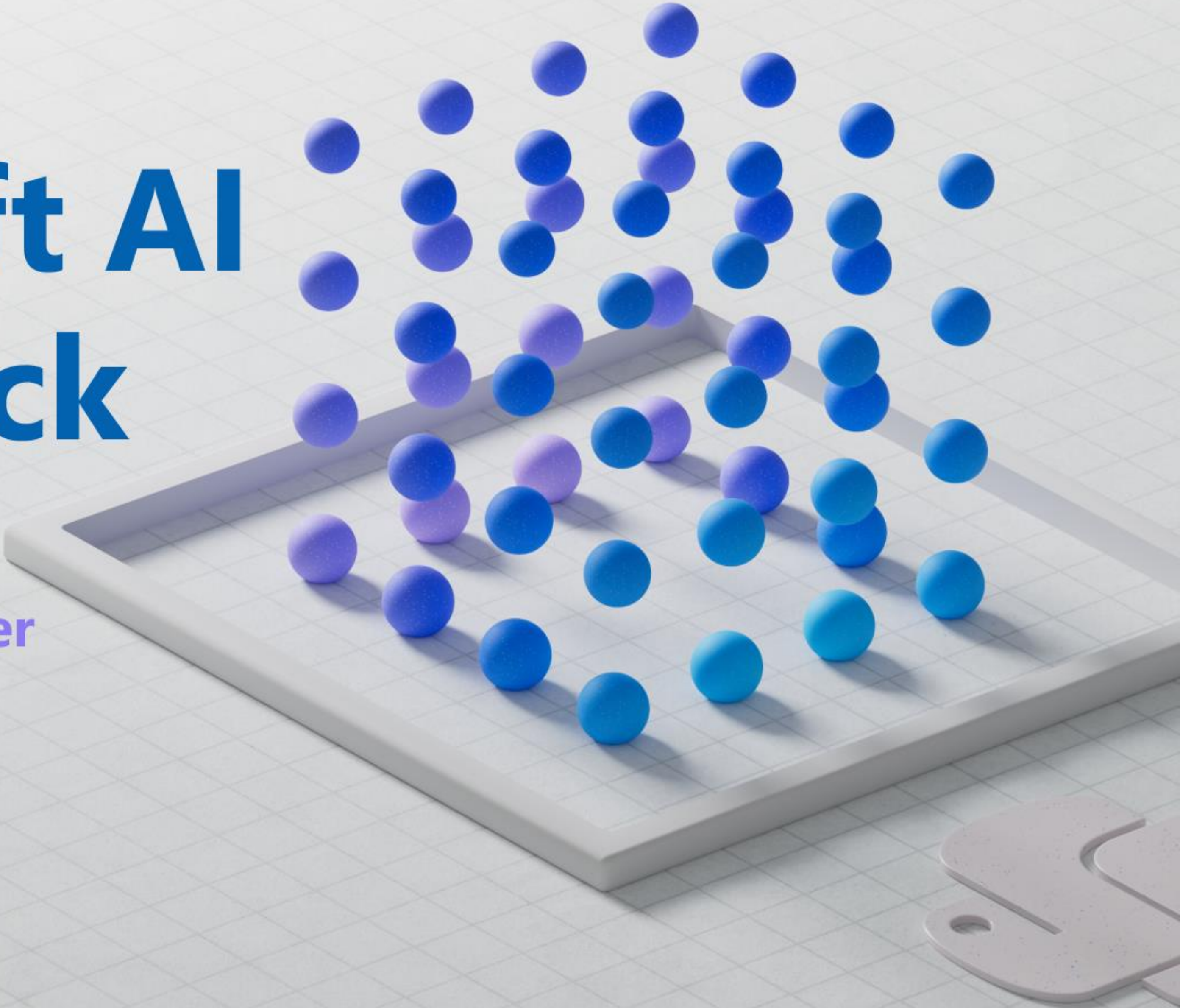
January 29th - February 12th

The Microsoft AI Chat App Hack



Build, innovate, and #HackTogether

aka.ms/hacktogether/chatapp

Microsoft

The AI Chat App Hack

January 29th - February 12th

29th ▶
Building a RAG Chat App in Python

30th ▶
Customizing your RAG Chat App

31st ▶
Azure AI Search Best Practices

1st ▶
GPT-4 with Vision

2nd
HACK
HACK
HACK

3rd
HACK
HACK

4th
HACK
HACK
HACK

5th ▶
AM: RAG Chat Web Components
PM: Access Control in RAG Chat Apps

6th ▶
Evaluating a RAG Chat App

7th ▶
RAG Chat Special Topic

8th ▶
Continuous Deployment of your Chat App

9th
HACK
HACK

10th
HACK

11th
HACK
HACK

12th
SUBMIT YOUR PROJECT

Build, innovate, and #HackTogether

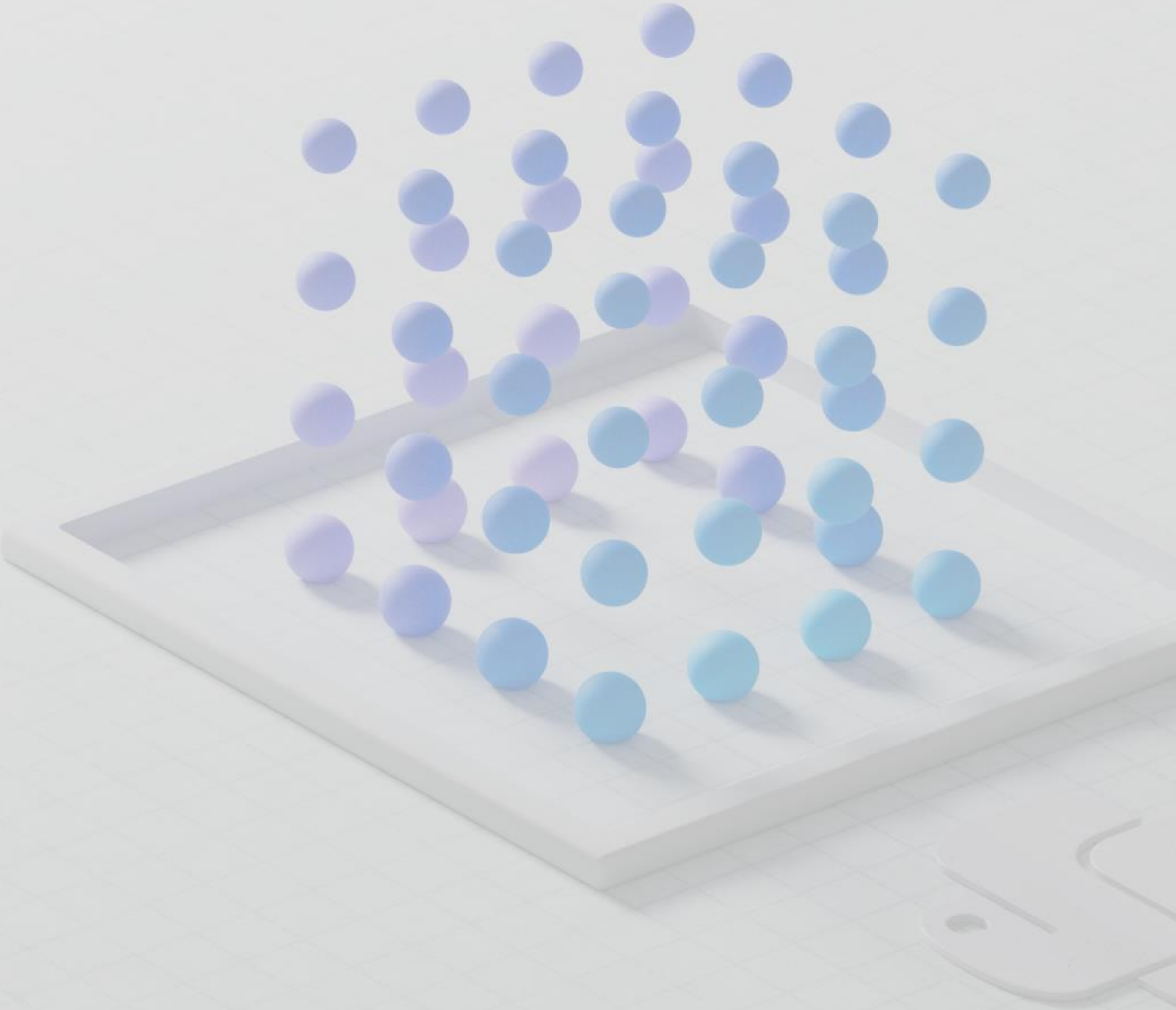
Building a RAG Chat App

with Azure AI Search, OpenAI, and Python 🐍

Goal

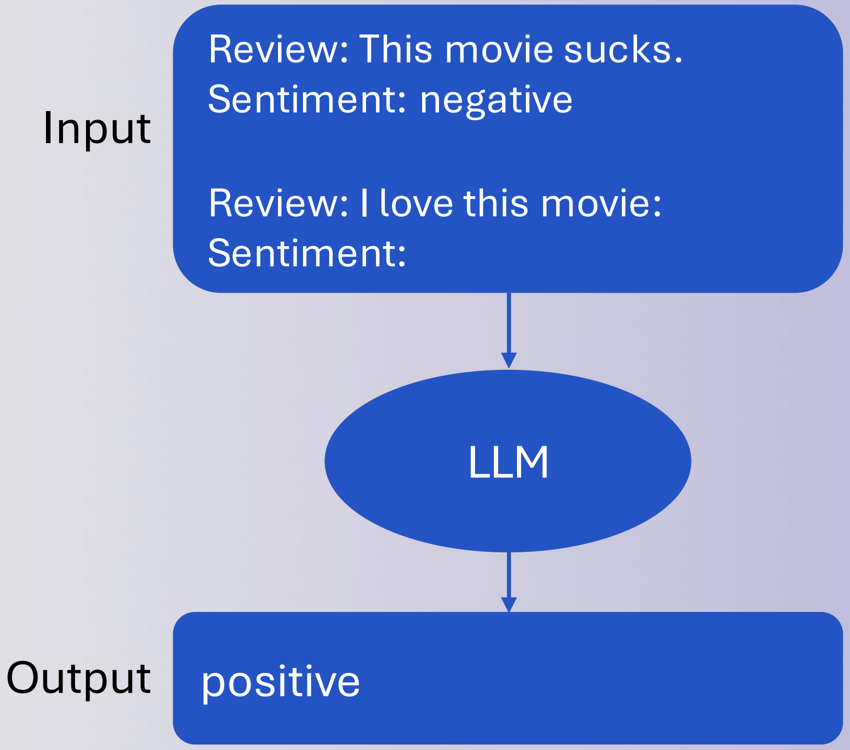
Build an AI Chat App using the **RAG** approach to combine an **LLM** with your own data.

LLMs



LLM: Large Language Model

An LLM is a model that is so large that it achieves general-purpose language understanding and generation.



Input

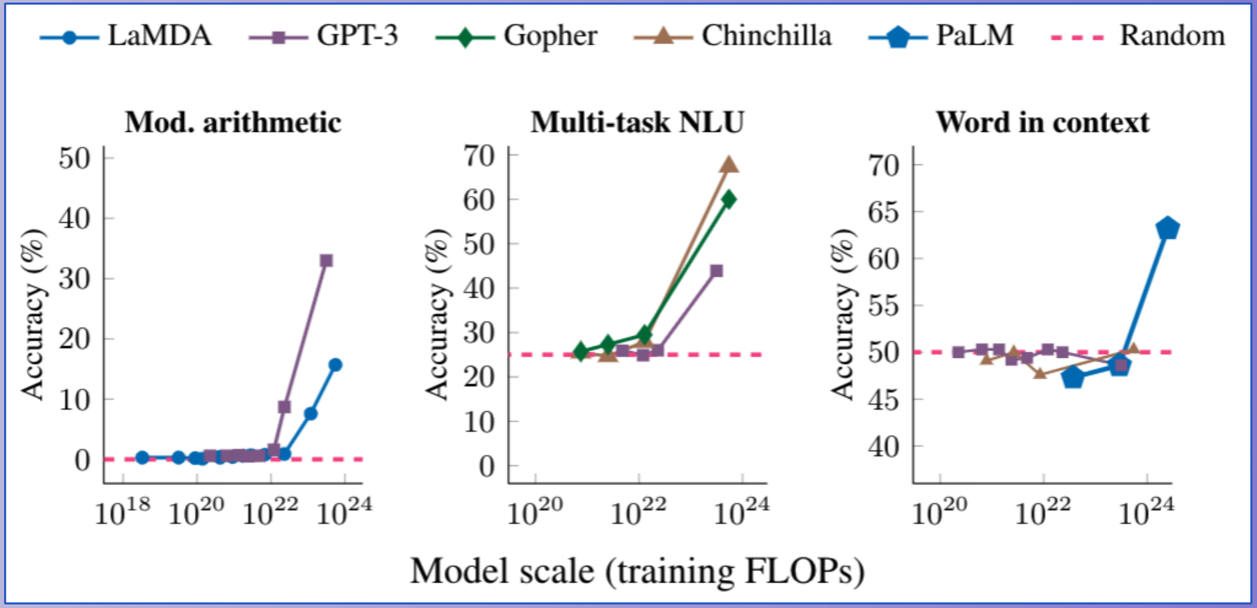
Review: This movie sucks.
Sentiment: negative

Review: I love this movie:
Sentiment:

LLM

Output

positive

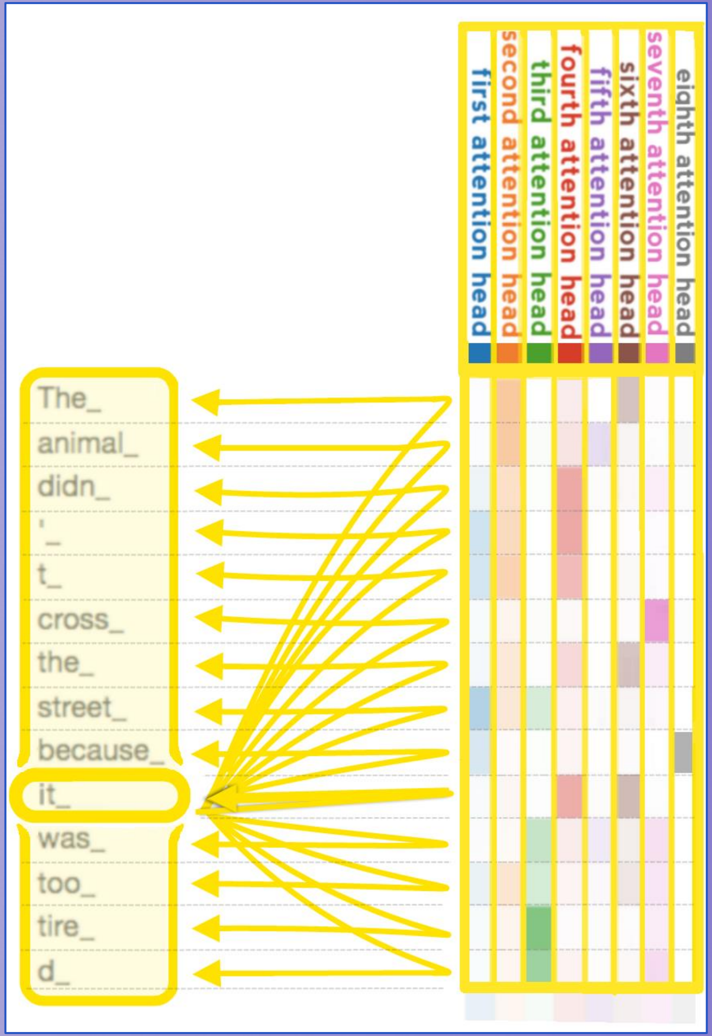


LLMs in use today

Model	# of Parameters	Creator	Uses
GPT 3.5	175 B	OpenAI	ChatGPT, Copilots, APIs
GPT 4	Undisclosed	OpenAI	
PaLM	540 B	Google	Bard
Claude 2	130 B	Anthropic	APIs
LlaMA	70 B	Meta	OSS
Mistral-7B	7 B	Mistral AI	OSS

GPT: Generative Pre-trained Transformer

GPT models are LLMs based on Transformer architecture from "Attention is all you need" paper



Learn more:

- Andrej Karpathy: 🎥 State of GPT
- Andrej Karpathy: 🎥 Let's build GPT: from scratch, in code

Using OpenAI GPT models: Azure Studio

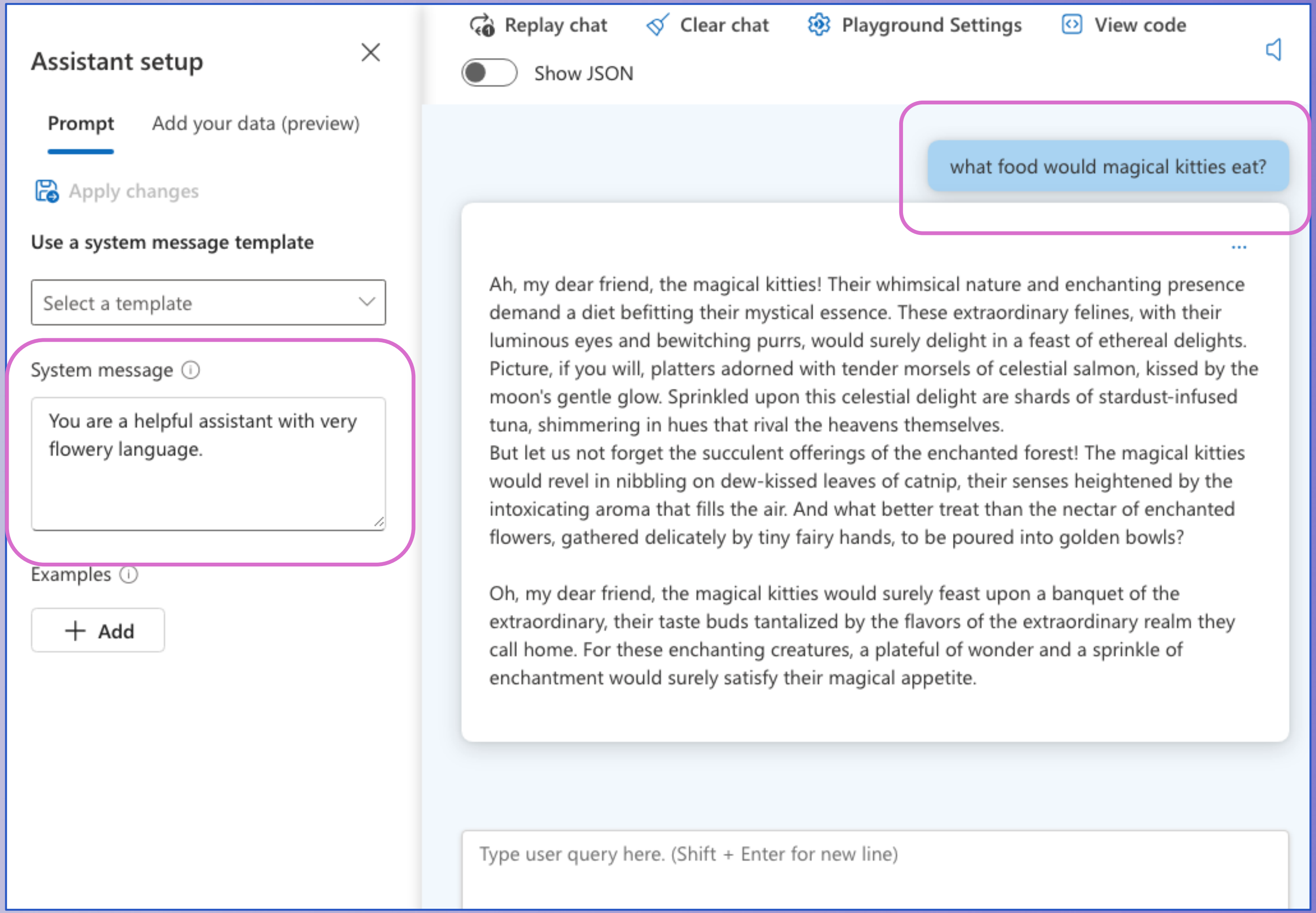
System message

+

User question

=

Chat Completion response



System message

You are a helpful assistant with very flowery language.

what food would magical kitties eat?

Ah, my dear friend, the magical kitties! Their whimsical nature and enchanting presence demand a diet befitting their mystical essence. These extraordinary felines, with their luminous eyes and bewitching purrs, would surely delight in a feast of ethereal delights. Picture, if you will, platters adorned with tender morsels of celestial salmon, kissed by the moon's gentle glow. Sprinkled upon this celestial delight are shards of stardust-infused tuna, shimmering in hues that rival the heavens themselves.

But let us not forget the succulent offerings of the enchanted forest! The magical kitties would revel in nibbling on dew-kissed leaves of catnip, their senses heightened by the intoxicating aroma that fills the air. And what better treat than the nectar of enchanted flowers, gathered delicately by tiny fairy hands, to be poured into golden bowls?

Oh, my dear friend, the magical kitties would surely feast upon a banquet of the extraordinary, their taste buds tantalized by the flavors of the extraordinary realm they call home. For these enchanting creatures, a plateful of wonder and a sprinkle of enchantment would surely satisfy their magical appetite.

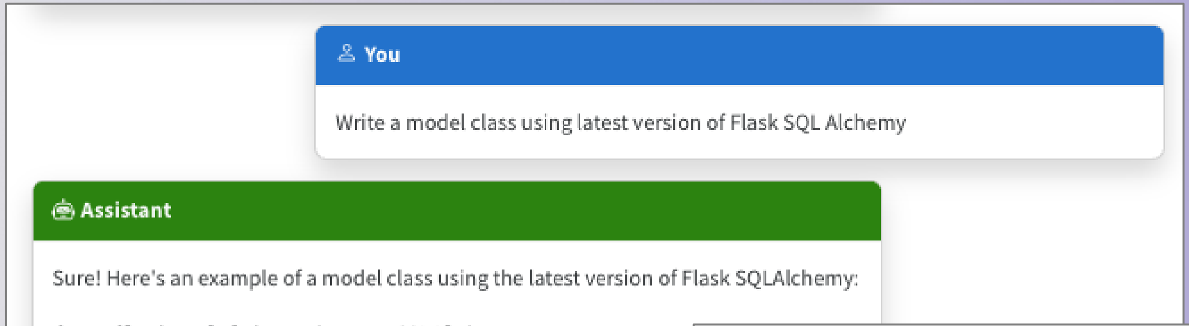
Using OpenAI GPT models: Python 🐍

```
response = openai.ChatCompletion.create(
  stream=True,
  messages = [
    {
     "role": "system",
     "content": "You are a helpful assistant with very flowery language"
    },
    {
     "role": "user",
     "content": "What food would magical kitties eat?”
    }
])

for event in response:
  print(event.choices[0].delta.content)
```

The limitations of LLMs

Outdated public knowledge



You

Write a model class using latest version of Flask SQL Alchemy

Assistant

Sure! Here's an example of a model class using the latest version of Flask SQLAlchemy:

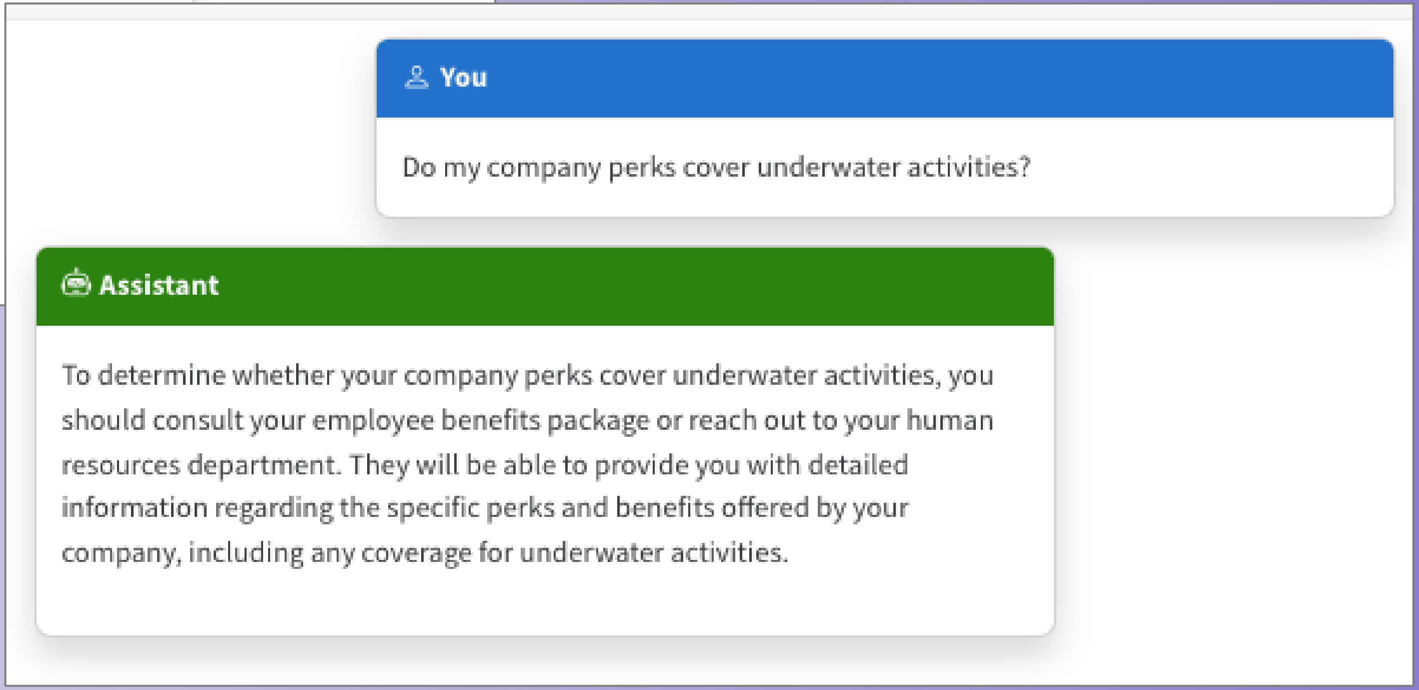
```
from flask_sqlalchemy import SQLAlchemy

db = SQLAlchemy()

class User(db.Model):
    __tablename__ = 'users'

    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(50), unique=True)
    email = db.Column(db.String(120), unique=True)
```

No internal knowledge



You

Do my company perks cover underwater activities?

Assistant

To determine whether your company perks cover underwater activities, you should consult your employee benefits package or reach out to your human resources department. They will be able to provide you with detailed information regarding the specific perks and benefits offered by your company, including any coverage for underwater activities.

Microsoft

Incorporating domain knowledge

Prompt engineering

In-context learning

Fine tuning

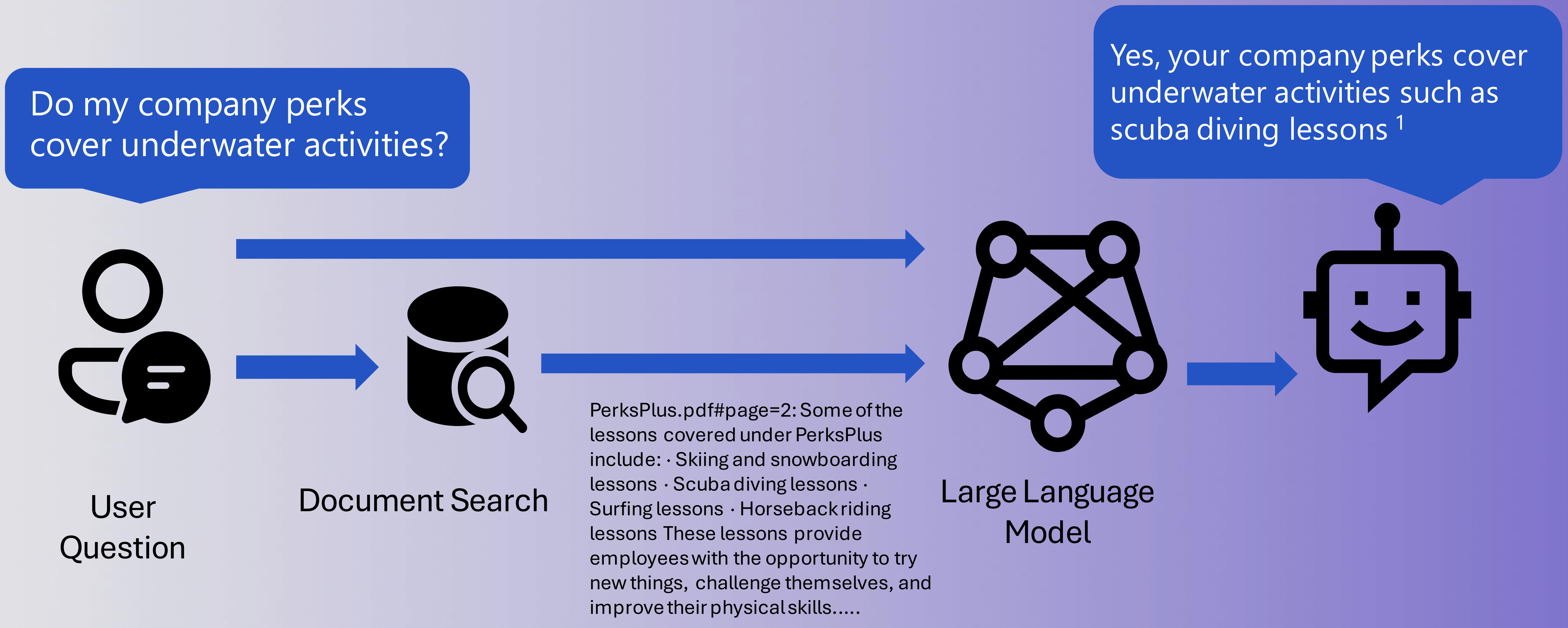
Learn new skills (permanently)

Retrieval Augmented Generation

Learn new facts (temporarily)

Retrieval Augmented Generation

RAG: Retrieval Augmented Generation



Do my company perks cover underwater activities?

User Question

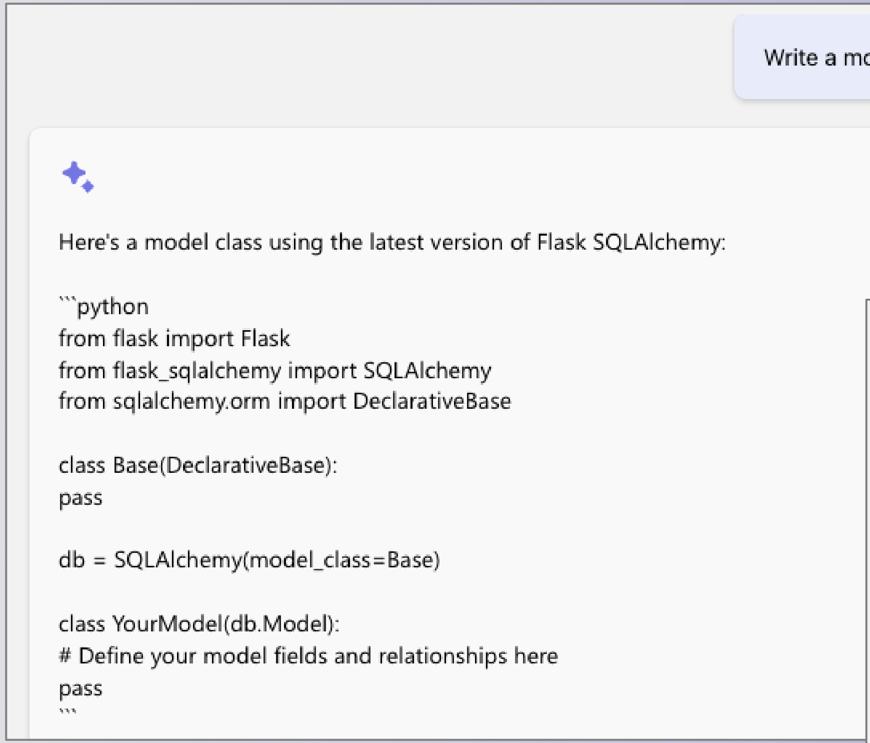
Document Search

PerksPlus.pdf#page=2: Some of the lessons covered under PerksPlus include: · Skiing and snowboarding lessons · Scuba diving lessons · Surfing lessons · Horseback riding lessons These lessons provide employees with the opportunity to try new things, challenge themselves, and improve their physical skills.....

Large Language Model

Yes, your company perks cover underwater activities such as scuba diving lessons[1]

The benefit of RAG



Write a model class using lastest version of Flask SQLAlchemy

Here's a model class using the latest version of Flask SQLAlchemy:

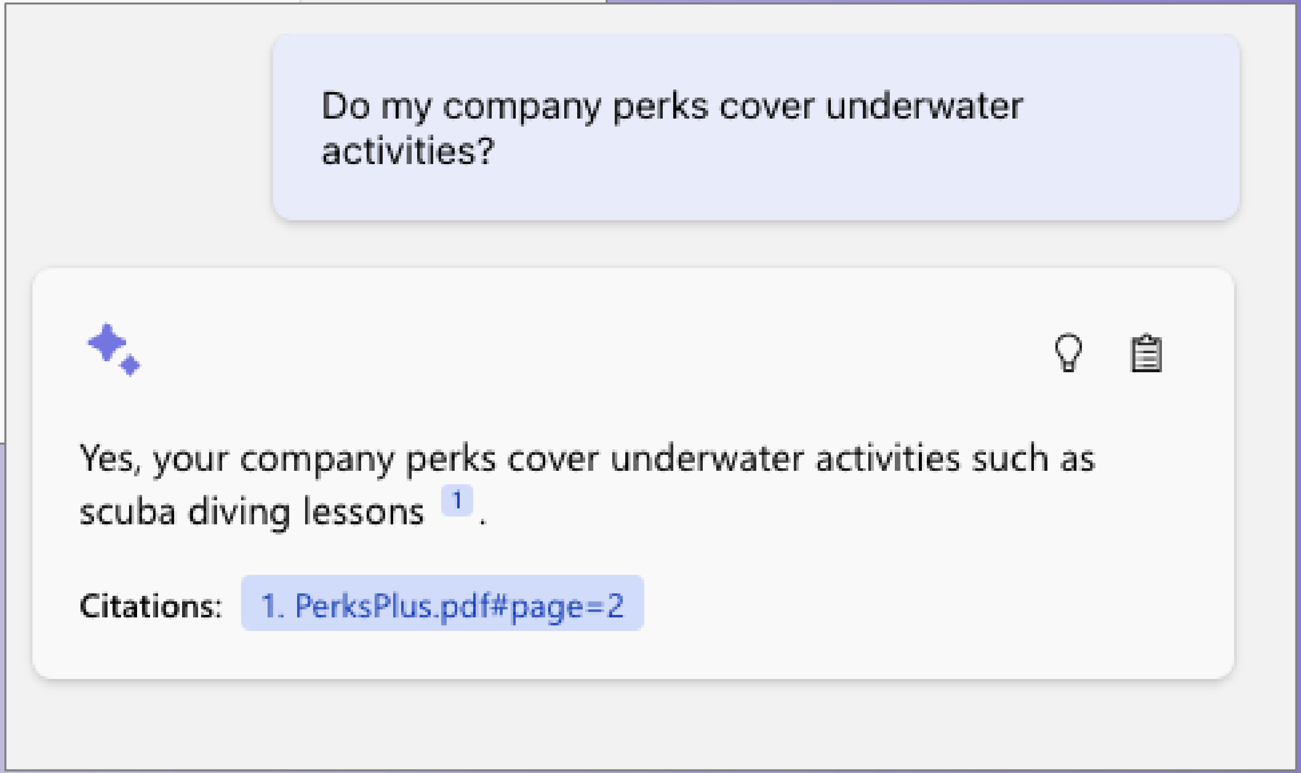
~~~
```python
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import DeclarativeBase

class Base(DeclarativeBase):
pass

db = SQLAlchemy(model_class=Base)

class YourModel(db.Model):
# Define your model fields and relationships here
pass
```
~~~

Up-to-date public knowledge



Do my company perks cover underwater activities?

Yes, your company perks cover underwater activities such as scuba diving lessons [1].

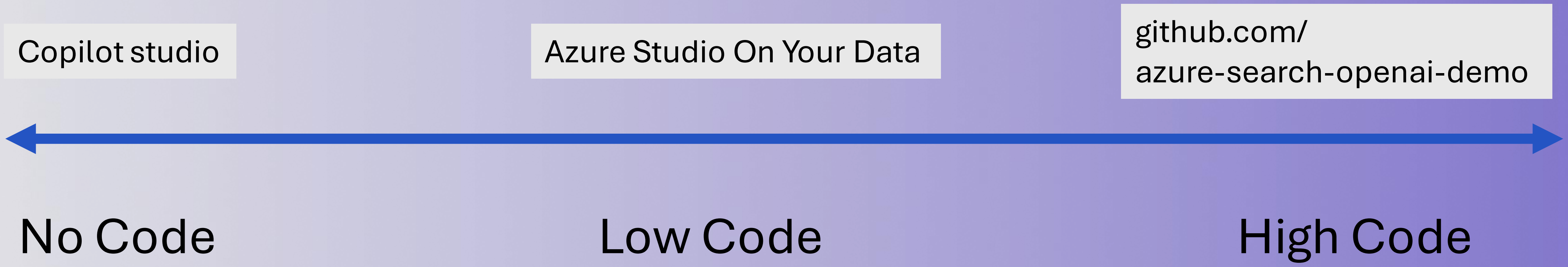
Citations: 1. PerksPlus.pdf#page=2

Access to internal knowledge

RAG components

Component	Examples
Retriever: A knowledge base that can efficiently retrieve sources that match a user query	Azure AI Search, Azure CosmosDB, PostgreSQL, Qdrant, Pinecone
LLM: A model that can answer questions based on the query based on the provided sources, and can include citations	GPT 3.5, GPT 4
Glue: A way to chain the retriever to the LLM (optional)	Langchain, Llamaindex, Semantic Kernel
Features	Chat history, Feedback buttons, Text-to-speech, User login, File upload, Access control, etc.

Many ways to build a RAG chat app



Copilot studio

Azure Studio On Your Data

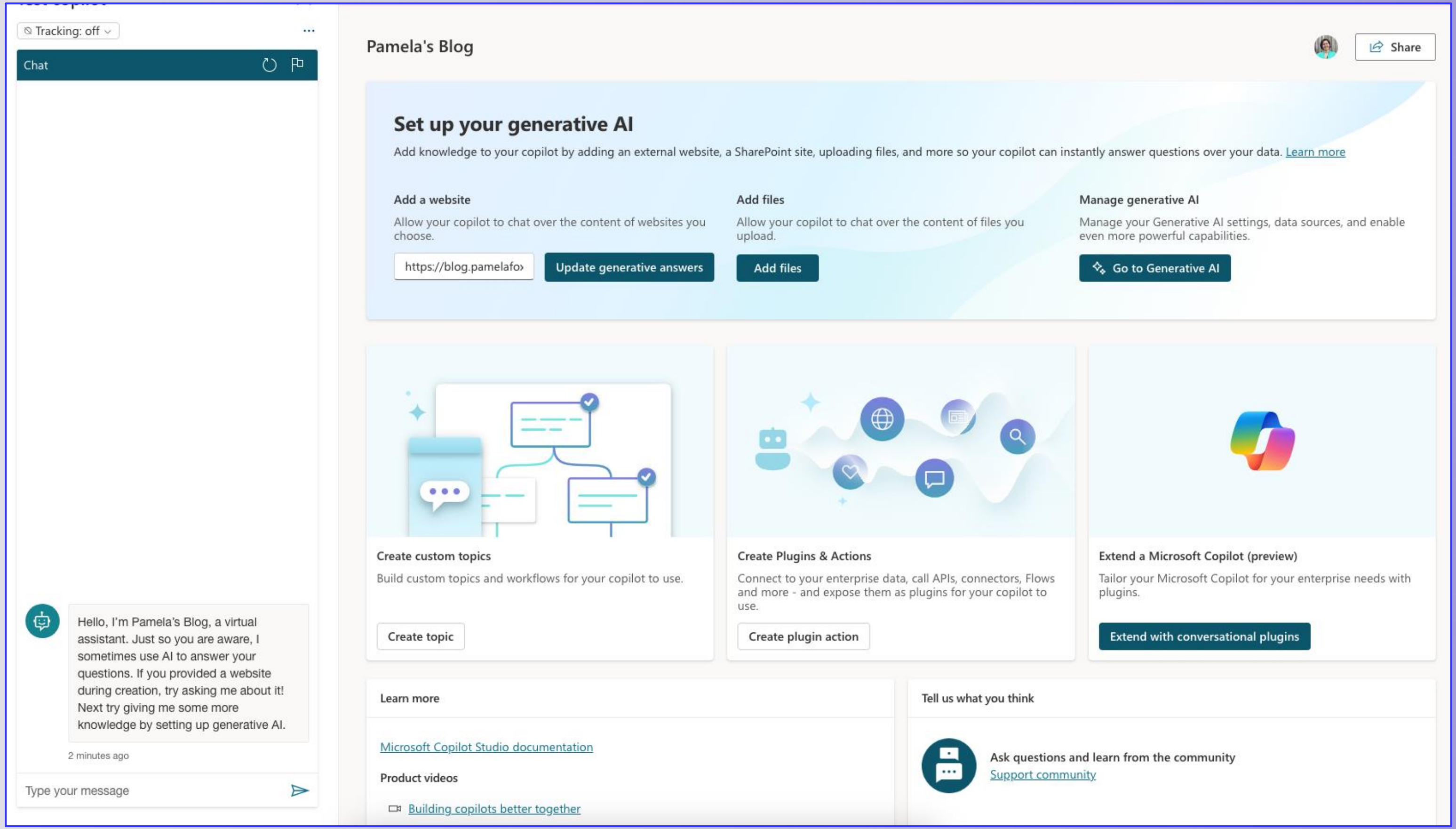
github.com/
azure-search-openai-demo

No Code

Low Code

High Code

Copilot Studio – On Your Data



Hello, I'm Pamela's Blog, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.

2 minutes ago

Set up your generative AI

Add knowledge to your copilot by adding an external website, a SharePoint site, uploading files, and more so your copilot can instantly answer questions over your data. Learn more

Add a website

Allow your copilot to chat over the content of websites you choose.

Add files

Allow your copilot to chat over the content of files you upload.

Manage generative AI

Manage your Generative AI settings, data sources, and enable even more powerful capabilities.

Create custom topics

Build custom topics and workflows for your copilot to use.

Create Plugins & Actions

Connect to your enterprise data, call APIs, connectors, Flows and more - and expose them as plugins for your copilot to use.

Extend a Microsoft Copilot (preview)

Tailor your Microsoft Copilot for your enterprise needs with plugins.

Learn more

Microsoft Copilot Studio documentation

Product videos

Building copilots better together

Tell us what you think

Ask questions and learn from the community

Support community

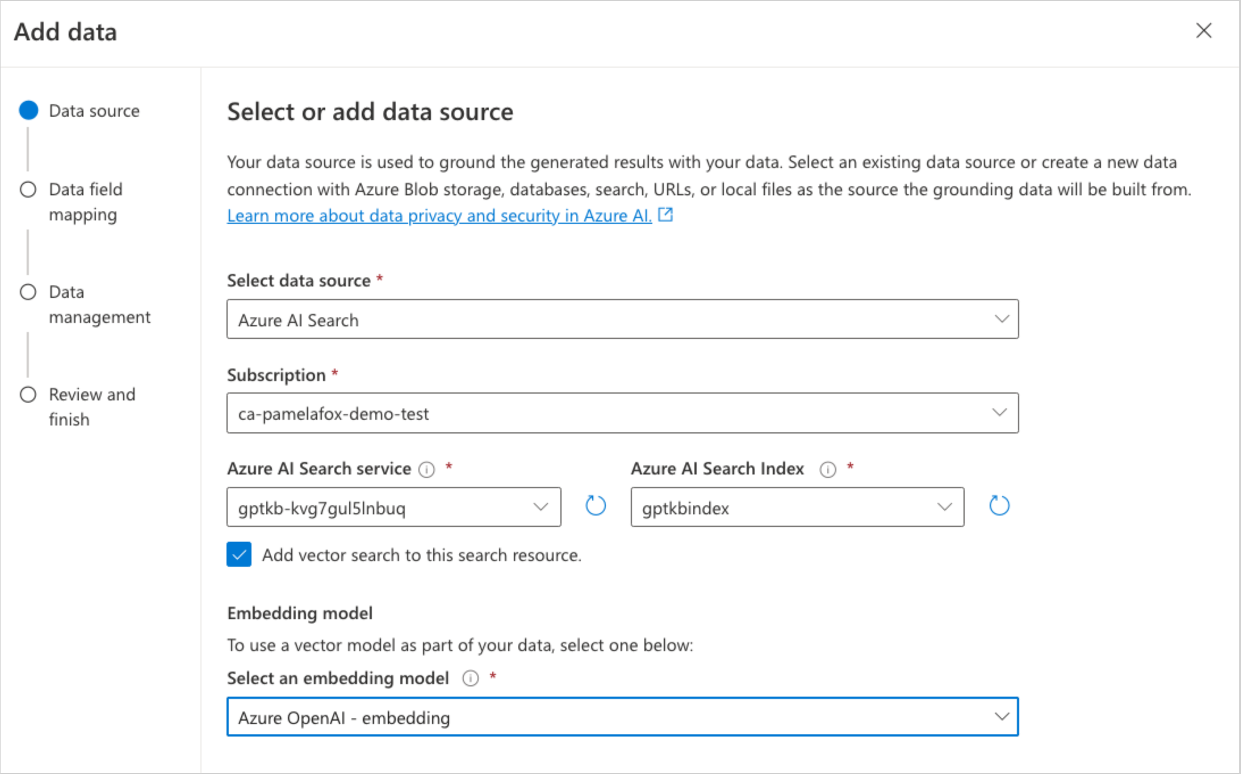
Retriever:

Uploaded files

LLM: GPT 3.5

https://copilotstudio.preview.microsoft.com/

Azure Studio – On Your Data



Select or add data source

Your data source is used to ground the generated results with your data. Select an existing data source or create a new data connection with Azure Blob storage, databases, search, URLs, or local files as the source the grounding data will be built from. [Learn more about data privacy and security in Azure AI.]()

Select data source *: Azure AI Search

Subscription *: ca-pamelafox-demo-test

Azure AI Search service *: gptkb-kvg7gul5lnbuq

Azure AI Search Index *: gptkbindex

☑ Add vector search to this search resource.

Embedding model

To use a vector model as part of your data, select one below:

Select an embedding model *: Azure OpenAI - embedding

Retriever:

- Azure AI Search
- Azure Blob Storage
- Azure CosmosDB for MongoDB vCore
- URL/Web address
- Uploaded files

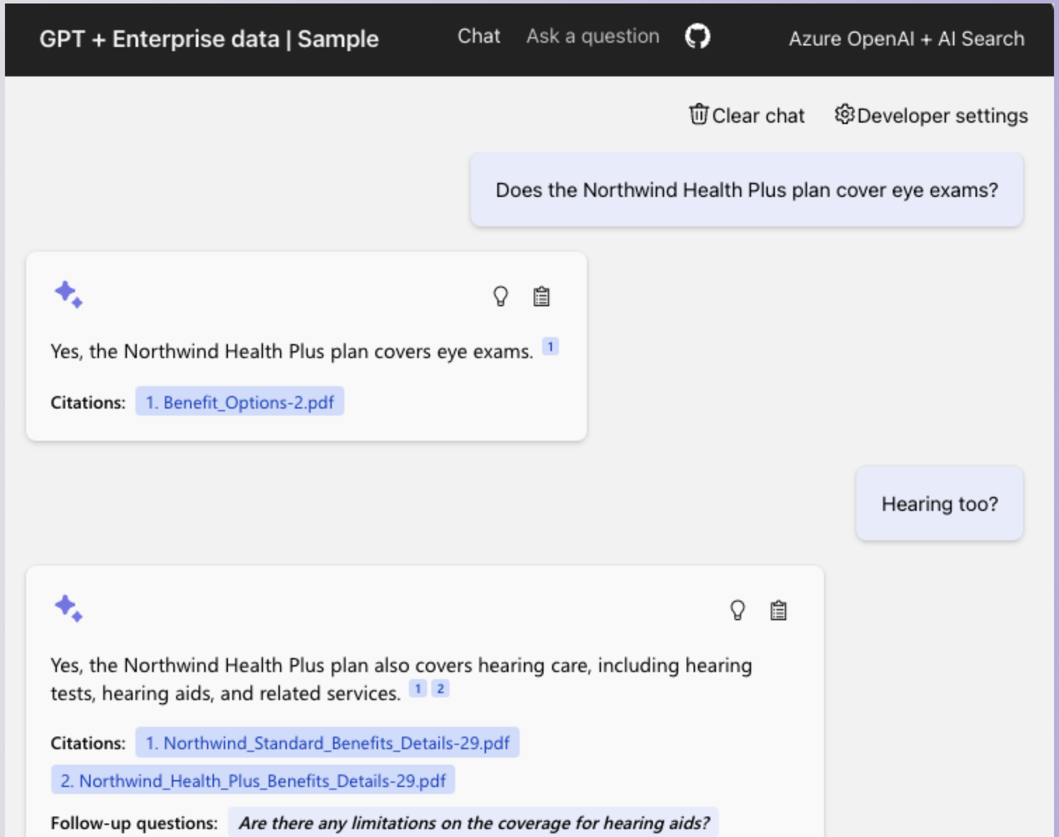
LLM: GPT 3.5/4

Features:

- User authentication
- Chat history persistence

https://learn.microsoft.com/azure/ai-services/openai/concepts/use-your-data

Open source RAG chat app solution



Does the Northwind Health Plus plan cover eye exams?

Yes, the Northwind Health Plus plan covers eye exams. [1]

Citations: 1. Benefit_Options-2.pdf

Hearing too?

Yes, the Northwind Health Plus plan also covers hearing care, including hearing tests, hearing aids, and related services. [1] [2]

Citations: 1. Northwind_Standard_Benefits_Details-29.pdf 2. Northwind_Health_Plus_Benefits_Details-29.pdf

Follow-up questions: *Are there any limitations on the coverage for hearing aids?*

Retriever:
Azure AI Search

LLM: GPT 3.5/4

Features:
Multi-turn chats
User authentication with ACLs
Chat with image documents

https://github.com/Azure-Samples/azure-search-openai-demo/ aka.ms/ragchat

Deep dive: RAG chat app solution

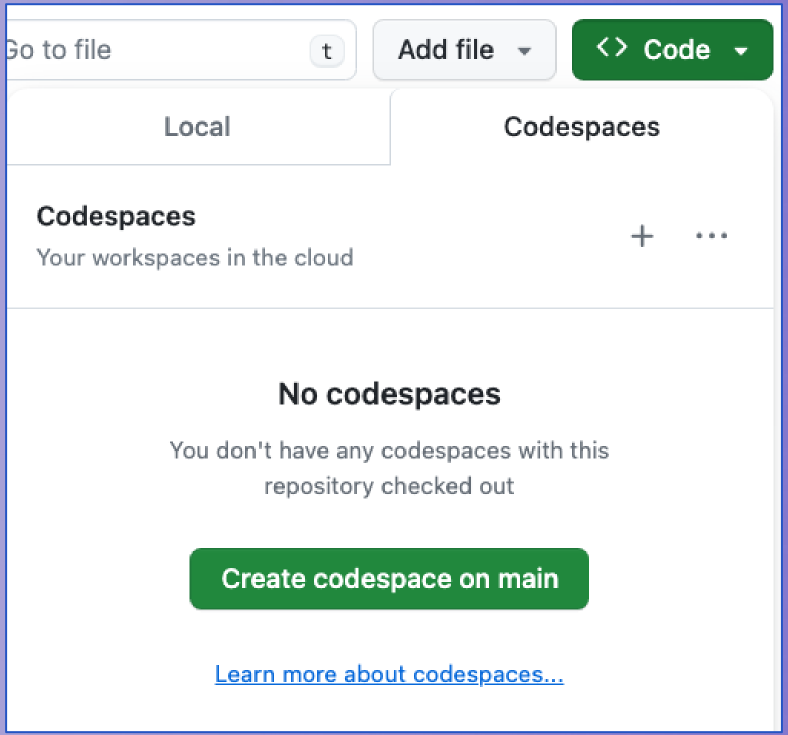
Prerequisites

- Azure account and subscription
  - A free account can be used, but will have limitations.
- Access to Azure OpenAI or an openai.com account
  - Request access to Azure OpenAI today! https://aka.ms/oaiapply

https://github.com/Azure-Samples/azure-search-openai-demo/#azure-account-requirements

Opening the project: 3 options

- GitHub Codespaces →
- VS Code with Dev Containers extension
- Your Local Environment
  - Python 3.9+
  - Node 14+
  - Azure Developer CLI



Codespaces

Your workspaces in the cloud

No codespaces

You don't have any codespaces with this repository checked out

https://github.com/Azure-Samples/azure-search-openai-demo/?tab=readme-ov-file#project-setup

Deploying with the Azure Developer CLI

Login to your Azure account:

```
azd auth login --use-device-code
```

Create a new azd environment: (to track deployment parameters)

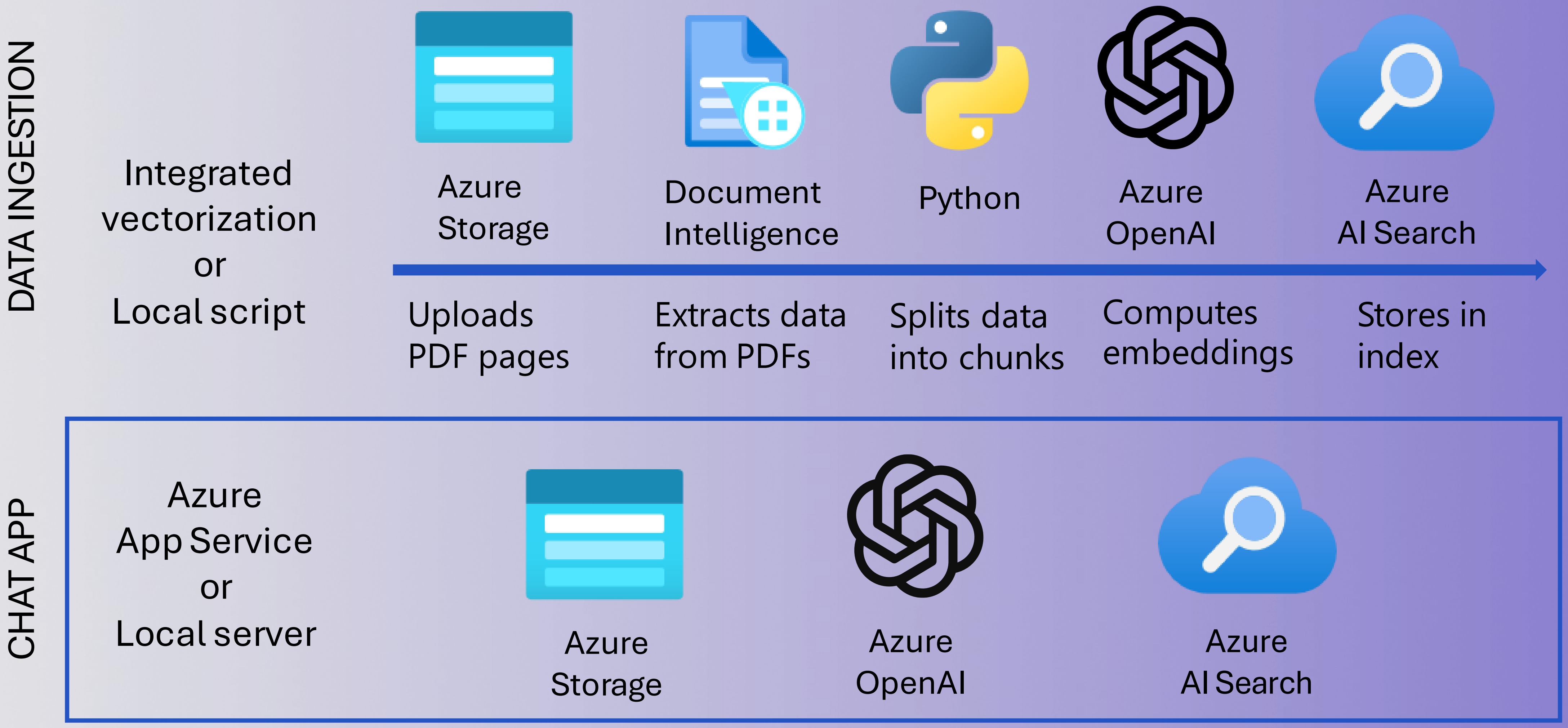
```
azd env new
```

Provision resources and deploy app:

```
azd up
```

azd up is a combination of azd provision and azd deploy

Application architecture on Azure



DATA INGESTION

Integrated vectorization
or
Local script

- Azure Storage: Uploads PDF pages
- Document Intelligence: Extracts data from PDFs
- Python: Splits data into chunks
- Azure OpenAI: Computes embeddings
- Azure AI Search: Stores in index

CHAT APP

Azure
App Service
or
Local server

- Azure Storage
- Azure OpenAI
- Azure AI Search

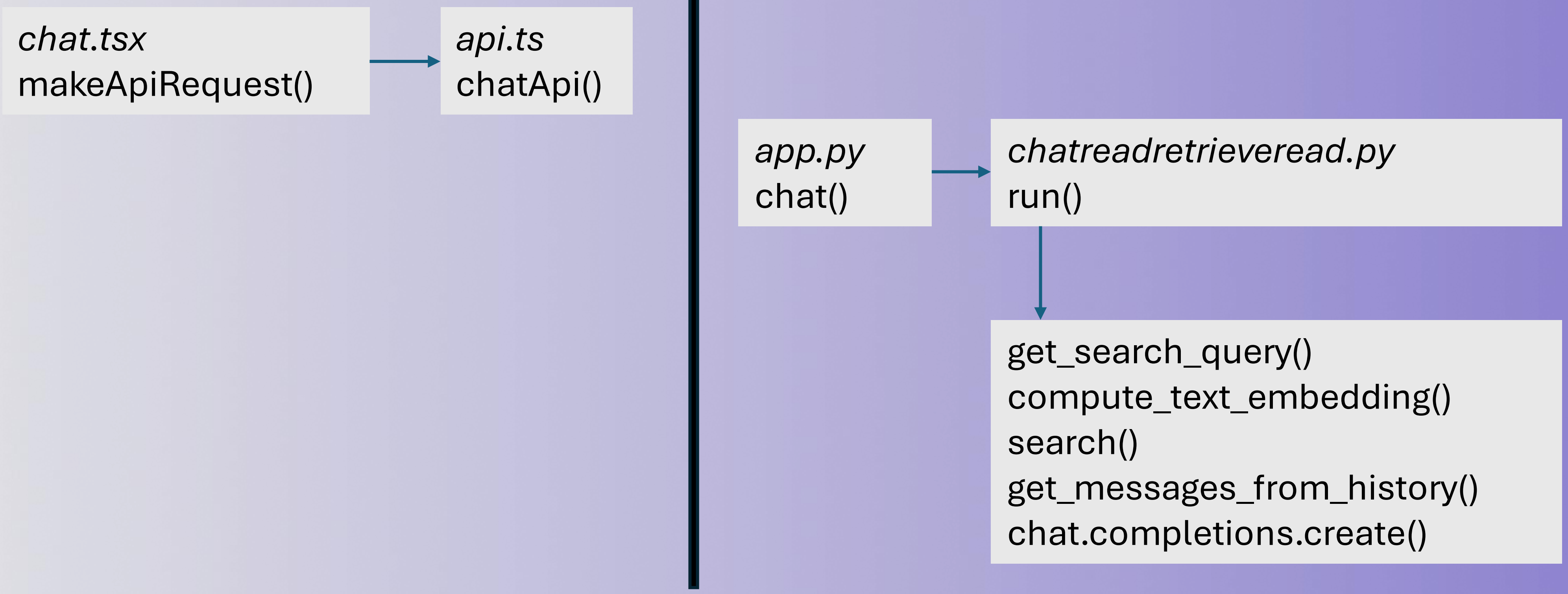
Code walkthrough

Typescript frontend

(React, FluentUI)

Python backend

(Quart, Uvicorn)



chat.tsx
makeApiRequest()

api.ts
chatApi()

app.py
chat()

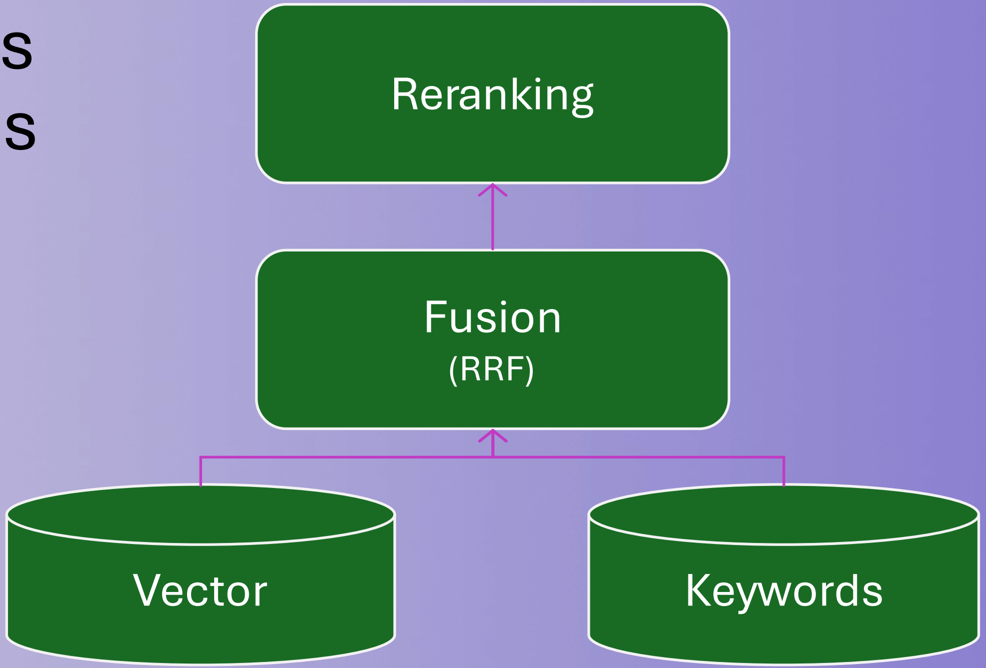
chatreadretrieveread.py
run()

get_search_query()
compute_text_embedding()
search()
get_messages_from_history()
chat.completions.create()

Search approach

For optimal retrieval, search() uses hybrid retrieval (text + vectors) plus the semantic ranker option.

https://aka.ms/ragrelevance



📢 Learn more at this week's session: Azure AI Search Best Practices

Next steps

- Register for the hackathon → aka.ms/hacktogether/chatapp
- Introduce yourself in our discussion forum
- Deploy the repo with the sample data
  - See steps on low cost deployment → aka.ms/ragchat/free
- Post in forum if you have any questions or issues deploying.
- Join tomorrow’s session: Customizing your RAG Chat App!