

What is the OpenAI API?

WORKING WITH THE OPENAI API

James Chapman

Curriculum Manager, DataCamp

Coming up...

Course goals

- Use the *OpenAI API* to access AI models
- Use those models to solve various tasks
- Use **Python** code to do this!

Expected knowledge

- Subsetting Python lists and **dictionaries**
- Control flow → `if`, `elif`, `else`
- Loops → `for`, `while`

Not expected to know

- AI or machine learning

OpenAI, ChatGPT, and the OpenAI API

- OpenAI = AI R&D company
- ChatGPT = AI application
- OpenAI API = Interface for accessing OpenAI models



OpenAI

¹ Image Credit: OpenAI

OpenAI, ChatGPT, and the OpenAI API

- OpenAI = AI R&D company
- ChatGPT = AI application
- OpenAI API = Interface for accessing OpenAI models



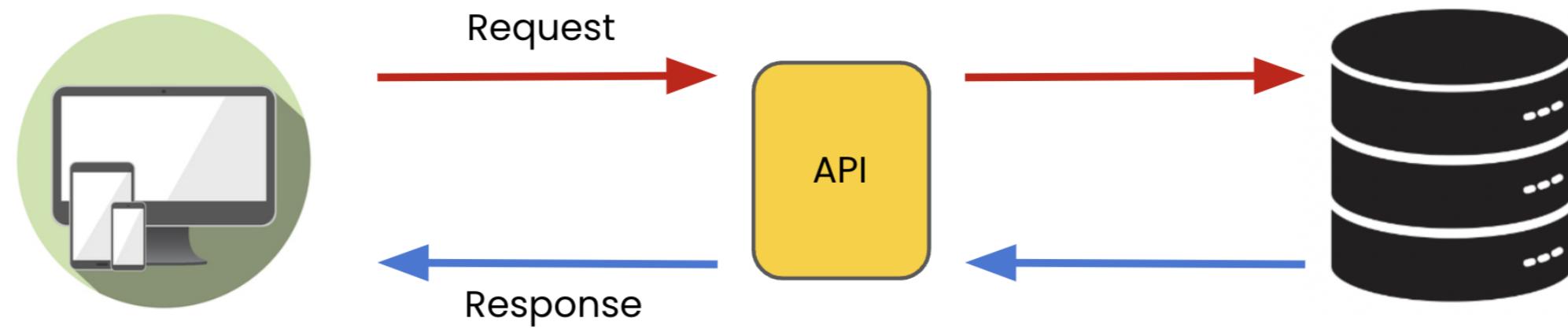
OpenAI, ChatGPT, and the OpenAI API

- **OpenAI** = AI R&D company
- **ChatGPT** = AI application
- **OpenAI API** = Interface for accessing OpenAI models



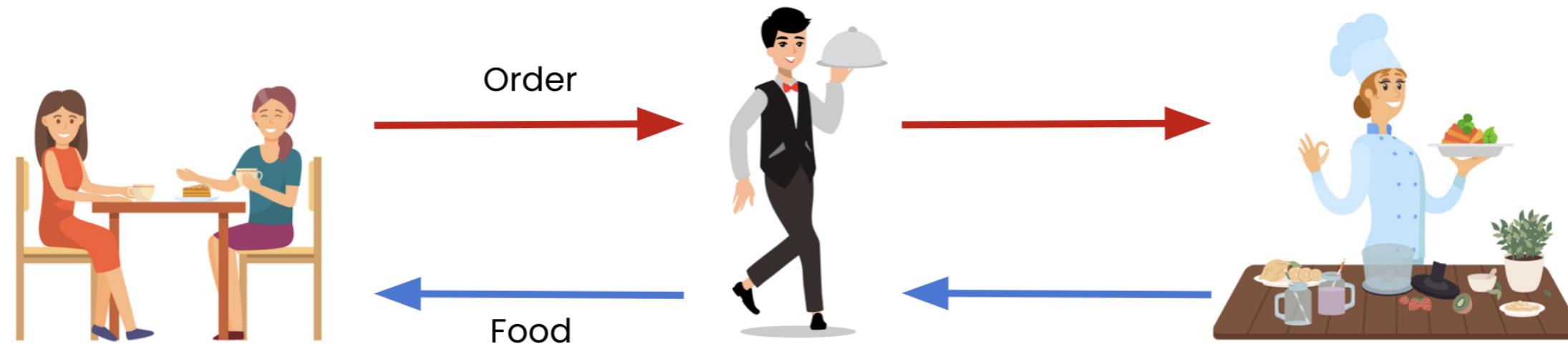
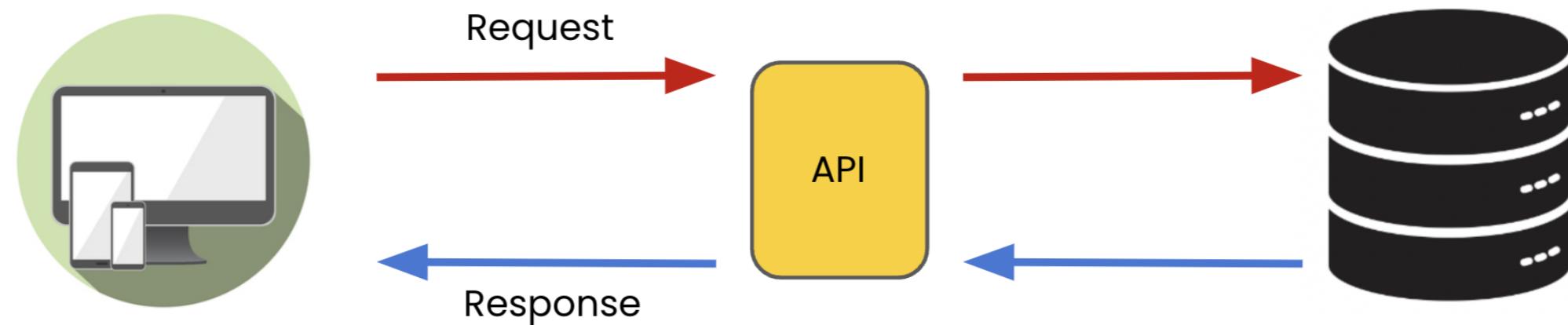
What is an API?

- Application Programming Interface

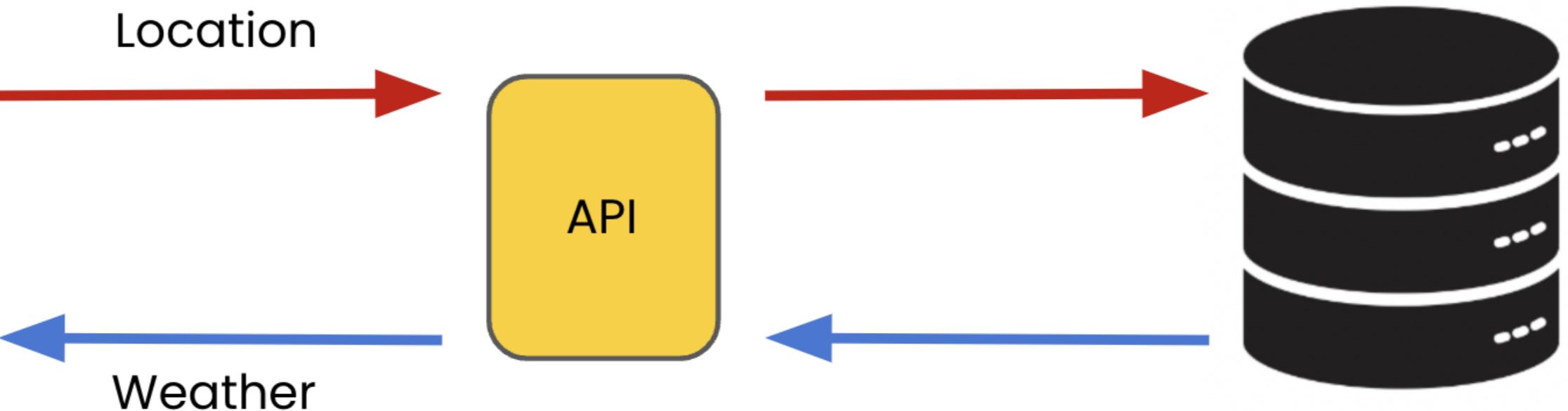


What is an API?

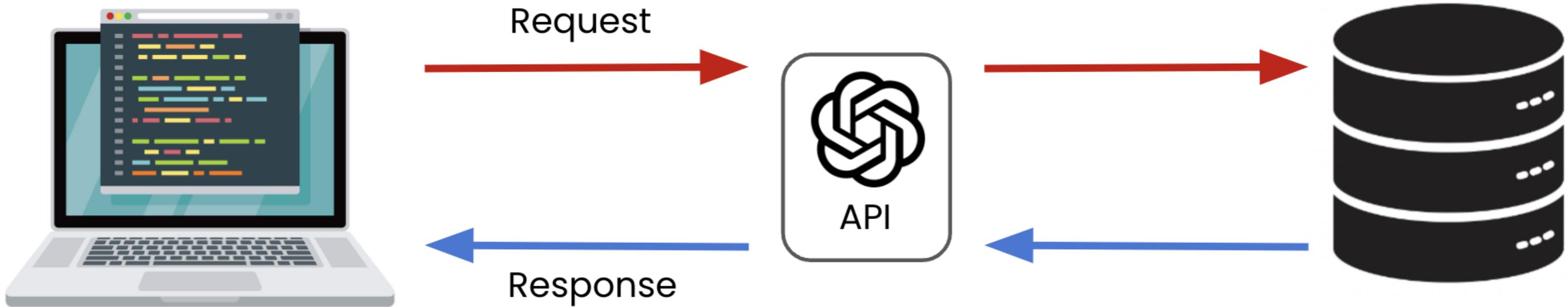
- Application Programming Interface



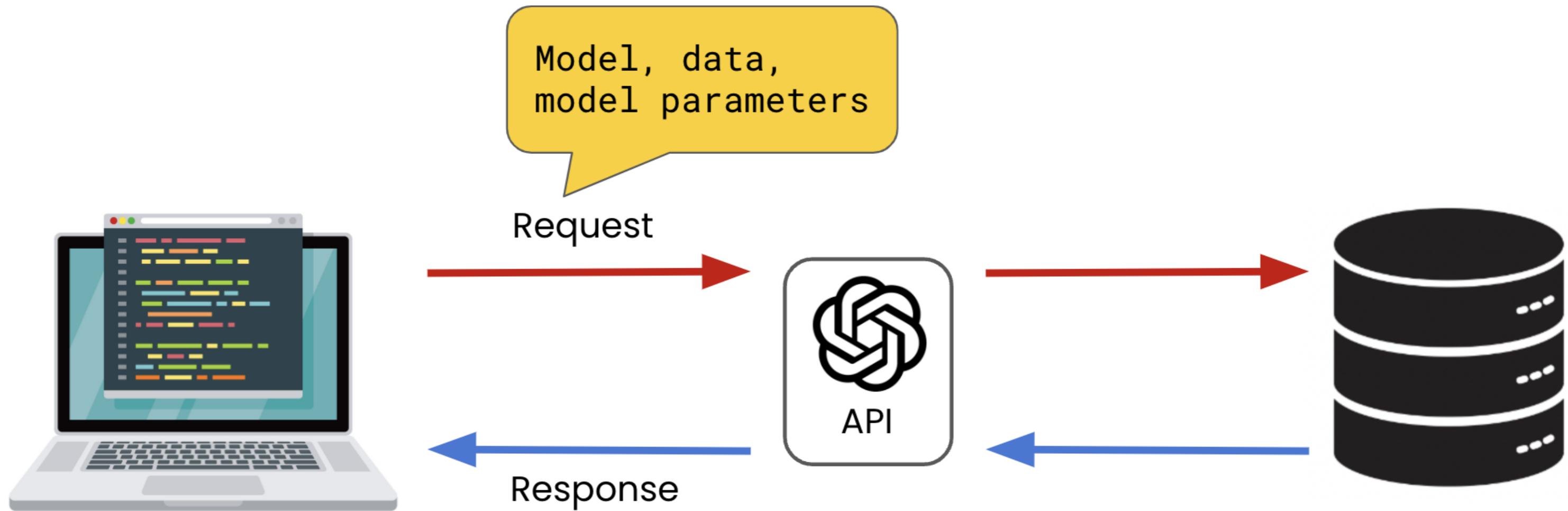
What is an API?



The OpenAI API

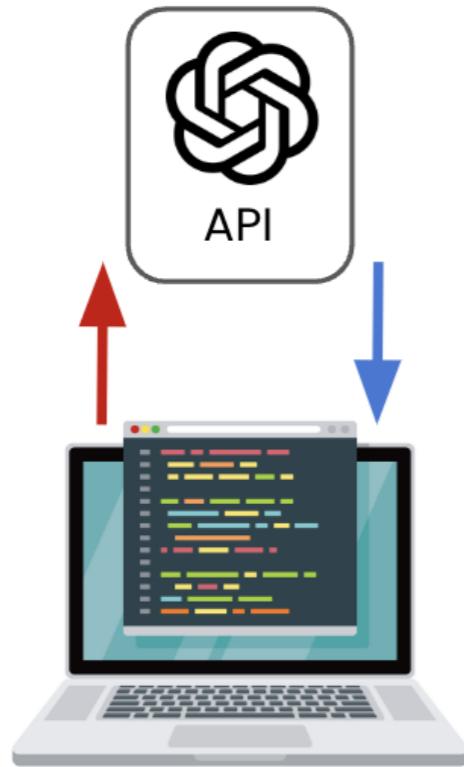


The OpenAI API



API vs. web interface

- Provides *flexibility* for integrating AI into products, experiences, and processes
- Interact via programming language
- Very little setup
- Sufficient for *streamlining* individual's workflows



A screenshot of a web-based AI interface. At the top, a user profile picture and the text 'Write Python code for drawing a scatter plot.' are visible. Below this, a message from the AI says: 'You can use the `matplotlib` library in Python to draw a scatter plot. Here's an example code:' followed by a code block. The code is as follows:

```
python

import matplotlib.pyplot as plt

# Example data
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 3, 5]

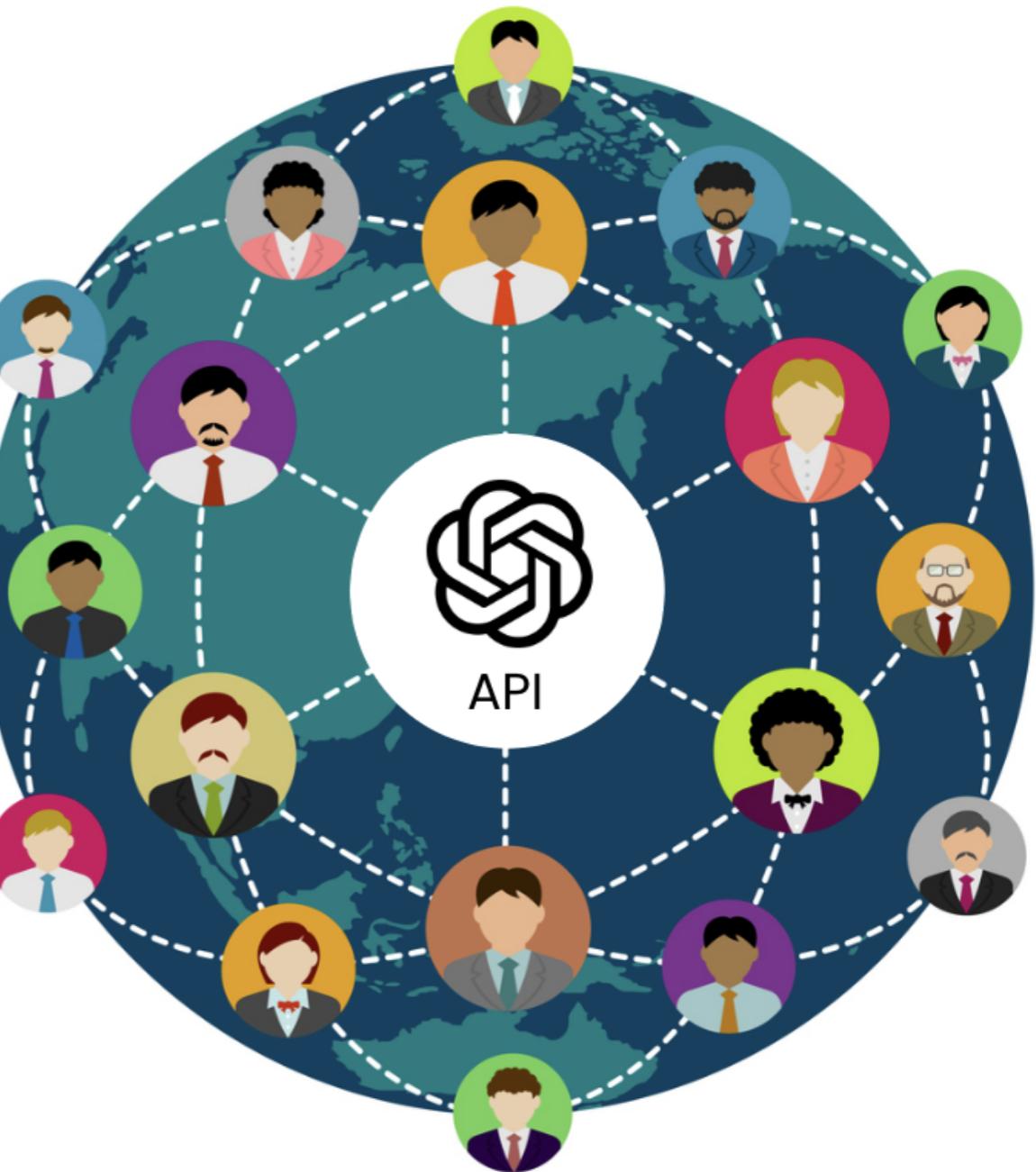
# Draw scatter plot
plt.scatter(x, y)

# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot Example')

# Show plot
plt.show()
```

Why the OpenAI API?

- Accessibility to AI
- Don't require data or ML experience



Building AI applications

DataCamp Workspace = Cloud-based IDE

The screenshot shows the DataCamp Workspace interface. On the left, a code editor window titled 'mcH' contains Python code for data processing and visualization:

```
1 # import necessary libraries
2 import pandas as pd
3 import plotly.express as px
4
5 # read the csv file
6 df = pd.read_csv('joey_clean.csv')
7
8 # convert start_date_time column to datetime format
9 df['start_date_time'] = pd.to_datetime(df['start_date_time'])
10
11 # create a new dataframe with date and total volume columns
12 date_volume_df = df.groupby('date')[['Pump Total Volume']].sum().reset_index()
13
14 # create a bar chart of total volume over time
15 fig = px.bar(date_volume_df, x='date', y='[Pump] Total Volume', title='Total Volume Over Time')
16 fig.show()
```

On the right, a context menu is open with the following options: Run, Comment, Generate (highlighted in grey), Move up, Move down, Hide code, Hide output, and Delete.

Below the code editor, a chart titled 'Total Volume Over Time' is displayed. The chart is a bar chart with blue bars representing the total volume for each day. The x-axis shows dates from March 5, 2023, to April 2, 2023. The y-axis is labeled '[Pump] Total Volume' and ranges from 0 to 30. The chart shows a general upward trend with some fluctuations.

Date	Total Volume
Mar 5, 2023	~30
Mar 6, 2023	~25
Mar 7, 2023	~26
Mar 8, 2023	~24
Mar 9, 2023	~23
Mar 10, 2023	~26
Mar 11, 2023	~28
Mar 12, 2023	~25
Mar 13, 2023	~28
Mar 14, 2023	~27
Mar 15, 2023	~31
Mar 16, 2023	~31
Mar 17, 2023	~26
Mar 18, 2023	~23
Mar 19, 2023	~29
Mar 20, 2023	~22
Mar 21, 2023	~25
Mar 22, 2023	~24
Mar 23, 2023	~26
Mar 24, 2023	~27
Mar 25, 2023	~28
Mar 26, 2023	~27
Mar 27, 2023	~28
Mar 28, 2023	~27
Mar 29, 2023	~28
Mar 30, 2023	~26
Mar 31, 2023	~27
Apr 1, 2023	~28
Apr 2, 2023	~27
Apr 3, 2023	~26
Apr 4, 2023	~25
Apr 5, 2023	~18

Let's practice!

WORKING WITH THE OPENAI API

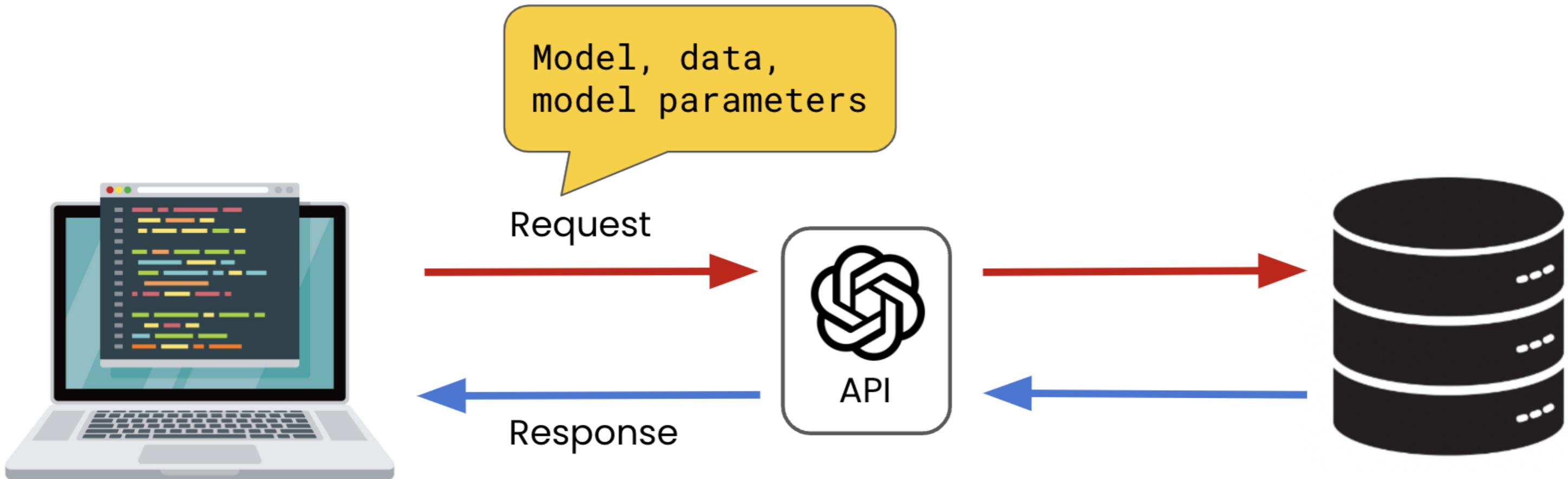
Making requests to the OpenAI API

WORKING WITH THE OPENAI API

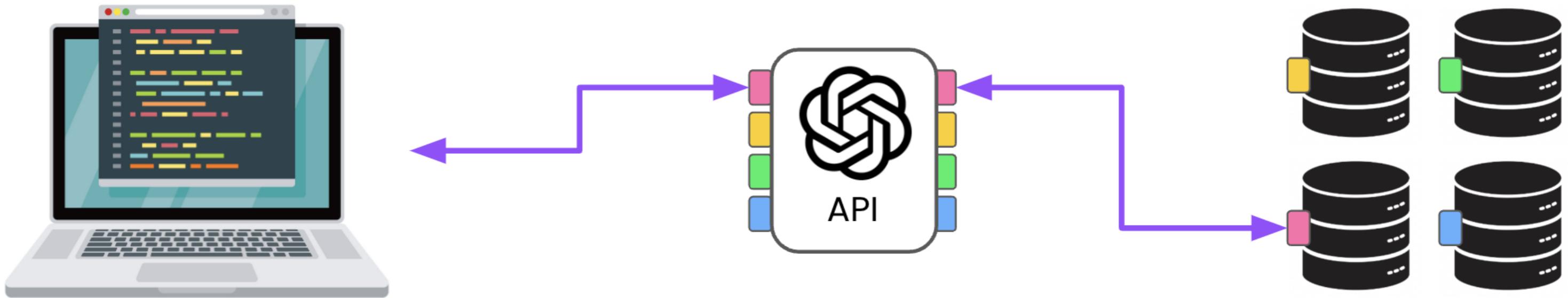
James Chapman

Curriculum Manager, DataCamp

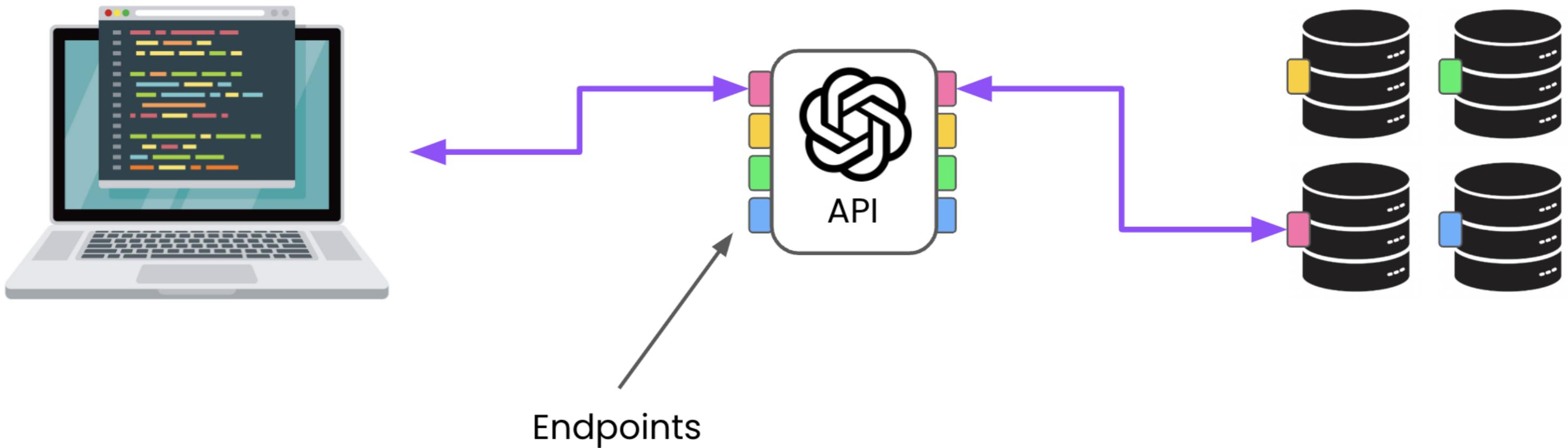
APIs recap...



API endpoints



API endpoints



API endpoints

- Endpoints → API access point designed for specific interactions



API authentication

- Authentication → Controls on access to API endpoint services (often unique key)



API usage costs

- For OpenAI API:
 - Larger inputs + outputs = Greater cost



¹ <https://openai.com/pricing>

API documentation



Overview Documentation API reference Examples Playground

API REFERENCE

Introduction

Authentication
Making requests

Models
Completions

Chat
Edits

Images
Embeddings

Audio
Files

Fine-tunes
Moderations
Engines
Parameter details

Introduction

You can interact with the API through HTTP requests from any language, via our official Python bindings, our official Node.js library, or a [community-maintained library](#).

To install the official Python bindings, run the following command:

```
pip install openai
```



To install the official Node.js library, run the following command in your Node.js project directory:

```
npm install openai
```



Authentication

The OpenAI API uses API keys for authentication. Visit your [API Keys](#) page to retrieve the API key you'll use in your requests.

Remember that your API key is a secret! Do not share it with others or expose it in any client-side code (browsers, apps). Production requests must be routed through your own backend server where your API key can be securely loaded from an environment variable or key management service.

¹ <https://platform.openai.com/docs/api-reference>

Creating an OpenAI API key

Create an OpenAI account

1) Go to the API signup page →

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

2) Create your account (you'll need to provide
your *email address* and your *phone number*)

OpenAI often provides free trial credit

Create your account

Please note that phone verification is required for
signup. Your number will only be used to verify
your identity for security purposes.

Email address

Continue

Already have an account? [Log in](#)

OR



Continue with Microsoft Account



Continue with Google

Creating an OpenAI API key

- 3) Go to the API keys page → <https://platform.openai.com/account/api-keys>
- 4) Create a new secret key and **copy** it (if you lose it, delete the key and create a new one)

NAME	KEY	CREATED	LAST USED ⓘ	
Secret key	sk-...Vpp5	5 Feb 2023	25 Apr 2023	 
+ Create new secret key				

Note: DataCamp doesn't store API keys used in the exercises

Making a request

```
import openai

openai.api_key = "ENTER YOUR KEY HERE"

response = openai.Completion.create(
    model="gpt-3.5-turbo-instruct",
    prompt="What is the OpenAI API?"
)

print(response)
```

The response

- JSON (JavaScript Object Notation) response

```
{  
  "choices": [  
    {  
      "finish_reason": "stop",  
      "index": 0,  
      "logprobs": null,  
      "text": "\n\nOpenAI is an artificial intelligence research laboratory that focuses on developing machine learning algorithms with the goal of advancing artificial intelligence technology. The OpenAI API is a set of tools, services, and data that can be used to create, train, and deploy models. With the OpenAI API, developers have access to advanced AI algorithms, datasets, pre-trained models, and tools to help them create applications powered by artificial intelligence."  
    }  
  ],  
  "created": 1682437639,  
  "id": "cmpl-79FAt7HqtdlQ1DDQIyvrkKLqFRThX",  
  "model": "gpt-3.5-turbo-instruct",  
  "object": "text_completion",  
  "usage": {  
    "completion_tokens": 86,  
    "prompt_tokens": 7,  
    "total_tokens": 93  
  }  
}
```

Interpreting the response

```
print(response["choices"])
```

```
[<OpenAIObject at 0x7f3146559c70> JSON: {  
    "finish_reason": "stop",  
    "index": 0,  
    "logprobs": null,  
    "text": "\n\nOpenAI is an artificial intelligence research laboratory that focuses  
on developing machine learning algorithms with the goal of advancing artificial  
intelligence technology. The OpenAI API is a set of tools, services, and data that  
can be used to create, train, and deploy models. With the OpenAI API, developers  
have access to advanced AI algorithms, datasets, pre-trained models, and tools to  
help them create applications powered by artificial intelligence."  
}]
```

Interpreting the response

```
print(response["choices"][0])
```

```
{  
    "finish_reason": "stop",  
    "index": 0,  
    "logprobs": null,  
    "text": "\n\nOpenAI is an artificial intelligence research laboratory that focuses  
on developing machine learning algorithms with the goal of advancing artificial  
intelligence technology. The OpenAI API is a set of tools, services, and data that  
can be used to create, train, and deploy models. With the OpenAI API, developers  
have access to advanced AI algorithms, datasets, pre-trained models, and tools to  
help them create applications powered by artificial intelligence."  
}
```

Interpreting the response

```
print(response["choices"][0]["text"])
```

OpenAI is an artificial intelligence research laboratory that focuses on developing machine learning algorithms with the goal of advancing artificial intelligence technology. The OpenAI API is a set of tools, services, and data that can be used to create, train, and deploy models. With the OpenAI API, developers have access to advanced AI algorithms, datasets, pre-trained models, and tools to help them create applications powered by artificial intelligence.

Let's practice!

WORKING WITH THE OPENAI API

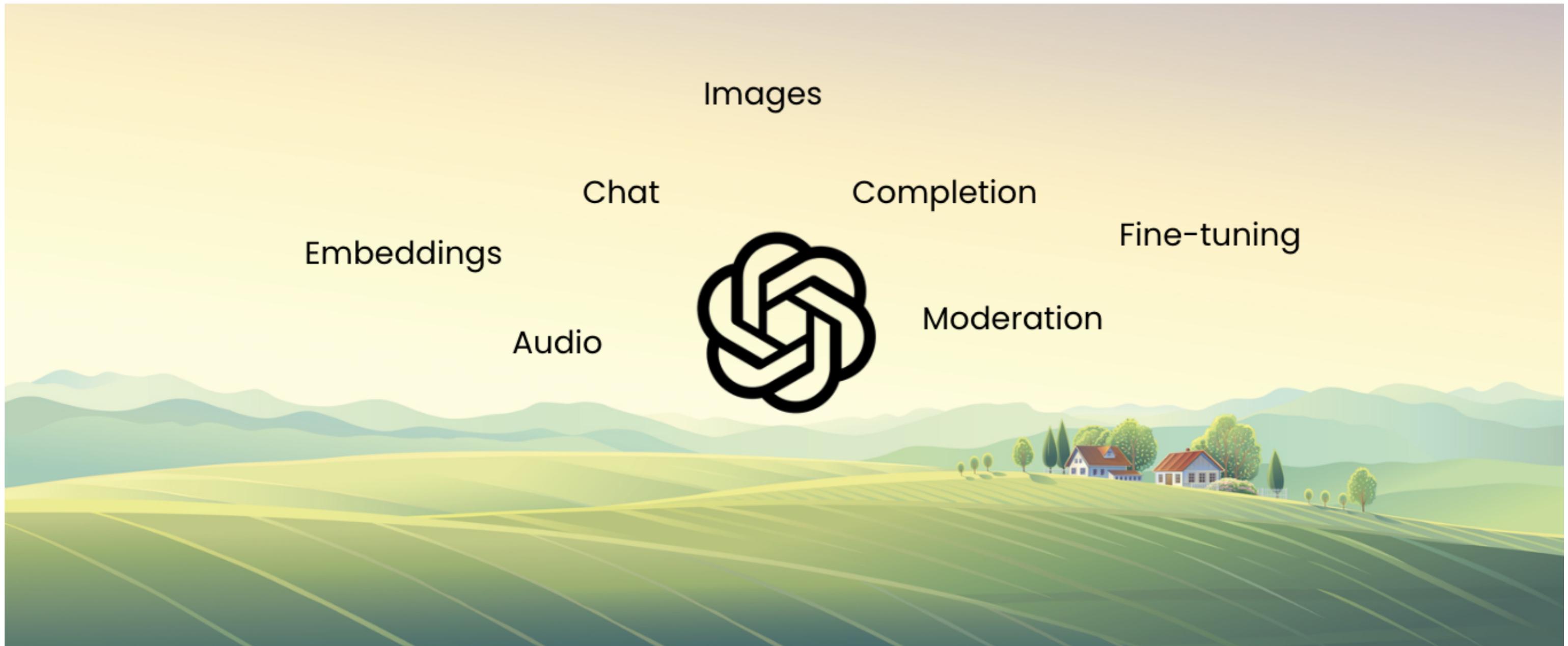
The OpenAI developer ecosystem

WORKING WITH THE OPENAI API

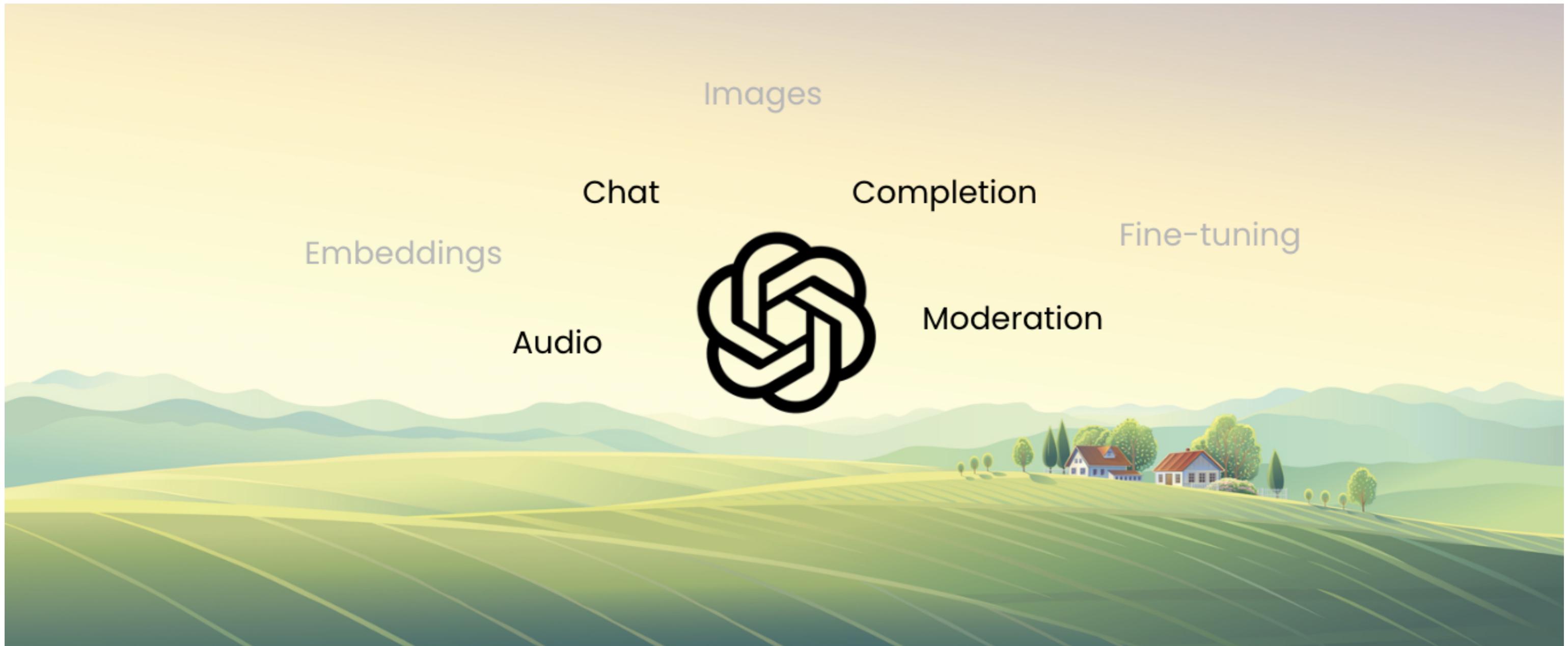
James Chapman

Curriculum Manager, DataCamp

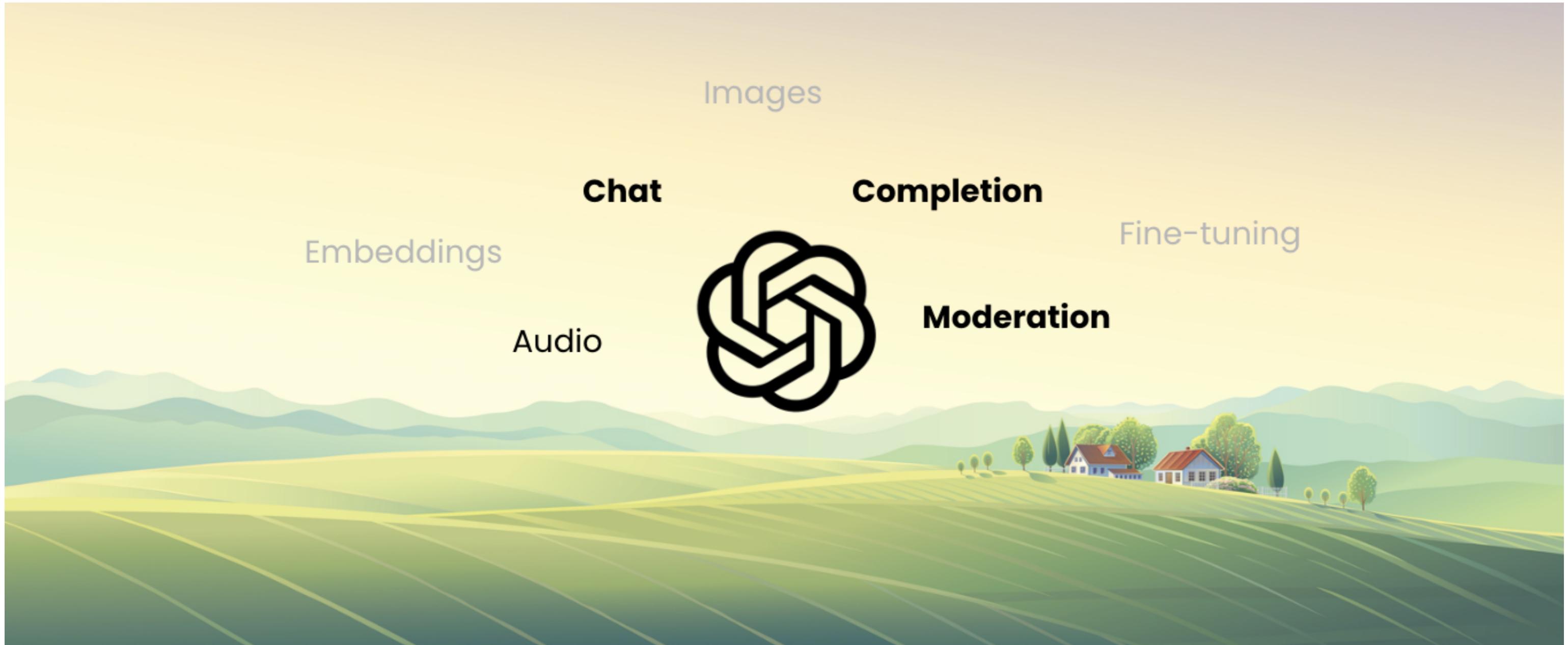
The OpenAI landscape



The OpenAI landscape

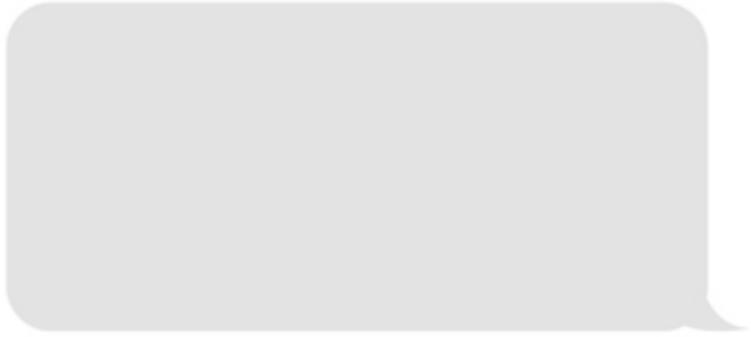


The OpenAI landscape



Completions

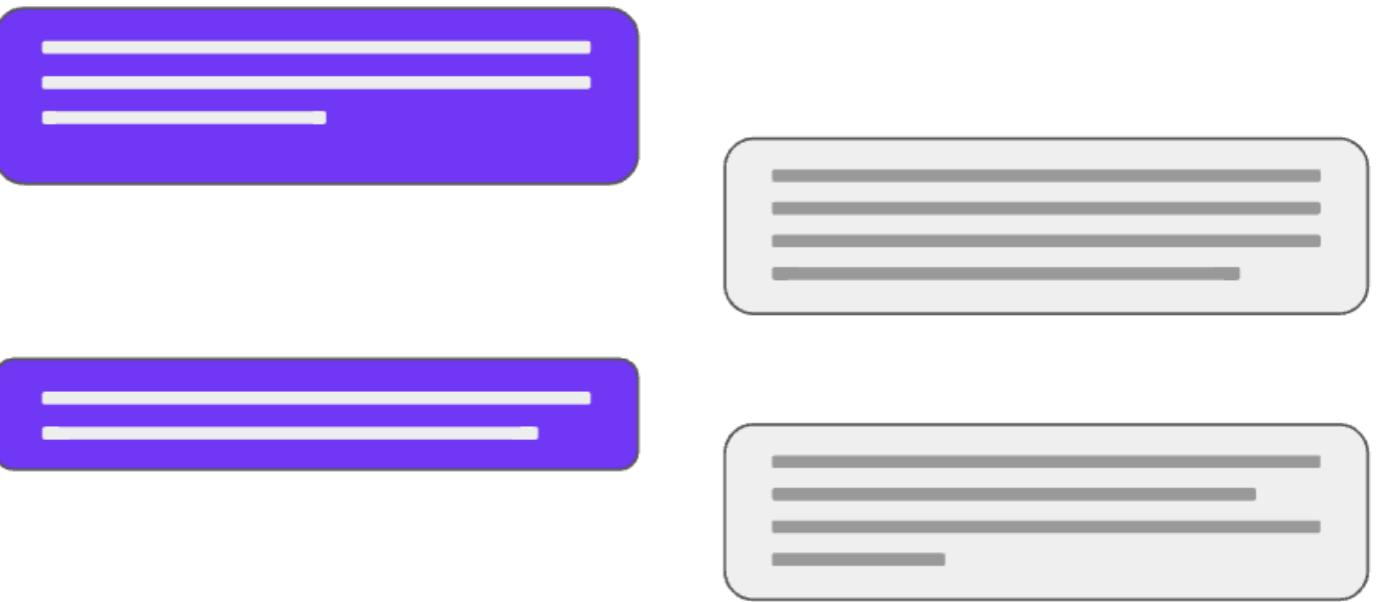
- Receive *continuation* of a prompt
- *Single-turn tasks*
 - Answer questions
 - Classification into categories
 - Sentiment analysis
 - Explain complex topics



```
response = openai.Completion.create(  
    model="gpt-3.5-turbo-instruct",  
    prompt="Insert prompt to complete."  
)
```

Chat

- *Multi-turn conversations*
 - Ideation
 - Customer support assistant
 - Personal tutor
 - Translate languages
 - Write code
- Also performs well on *single-turn* tasks



- Available through `openai.ChatCompletion`
→ Chapter 2

Moderation

- Check content for *violations* of OpenAI's *usage policies*, including:
 - Inciting violence
 - Hate speech
- Can *customize model sensitivity* to specific violations



¹ <https://openai.com/policies/usage-policies> ² <https://platform.openai.com/docs/guides/moderation/overview>

Organization settings

Organization name

Human-friendly label for your organization, shown in user interfaces

DataCamp

- Better management of *access, billing, and usage limits*
- Users can be part of multiple organizations

Organization ID

Identifier for this organization sometimes used in API requests

Save

¹ <https://platform.openai.com/account/org-settings>

Organizations

```
import openai

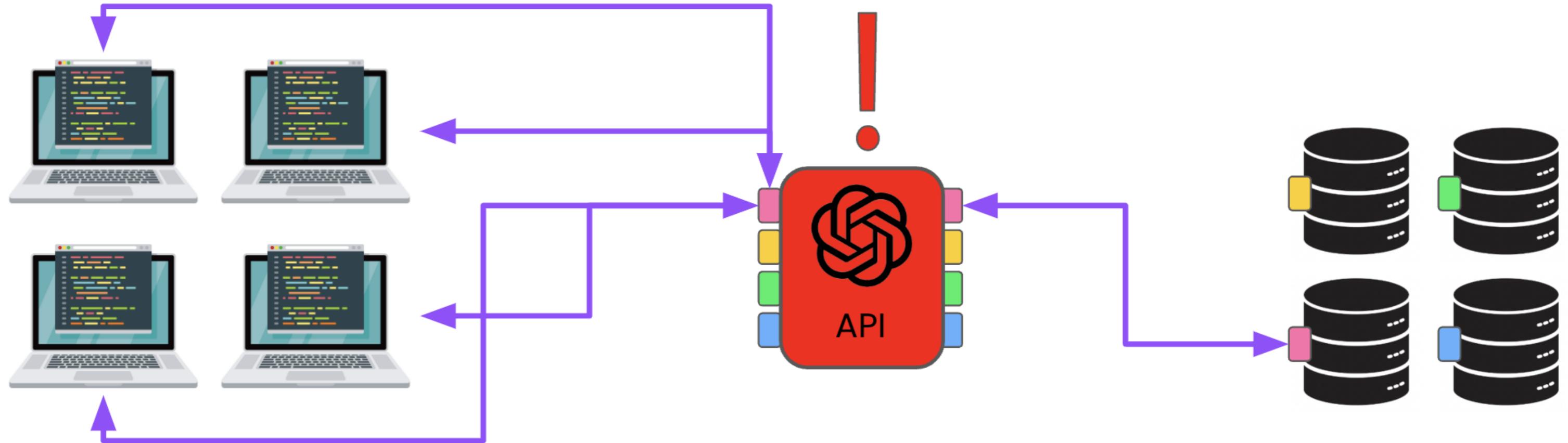
openai.api_key = "ENTER YOUR KEY HERE"
openai.organization = "ENTER ORG ID"

response = openai.Completion.create(
    model="gpt-3.5-turbo-instruct",
    prompt="What is the OpenAI API?"
)

print(response)
```

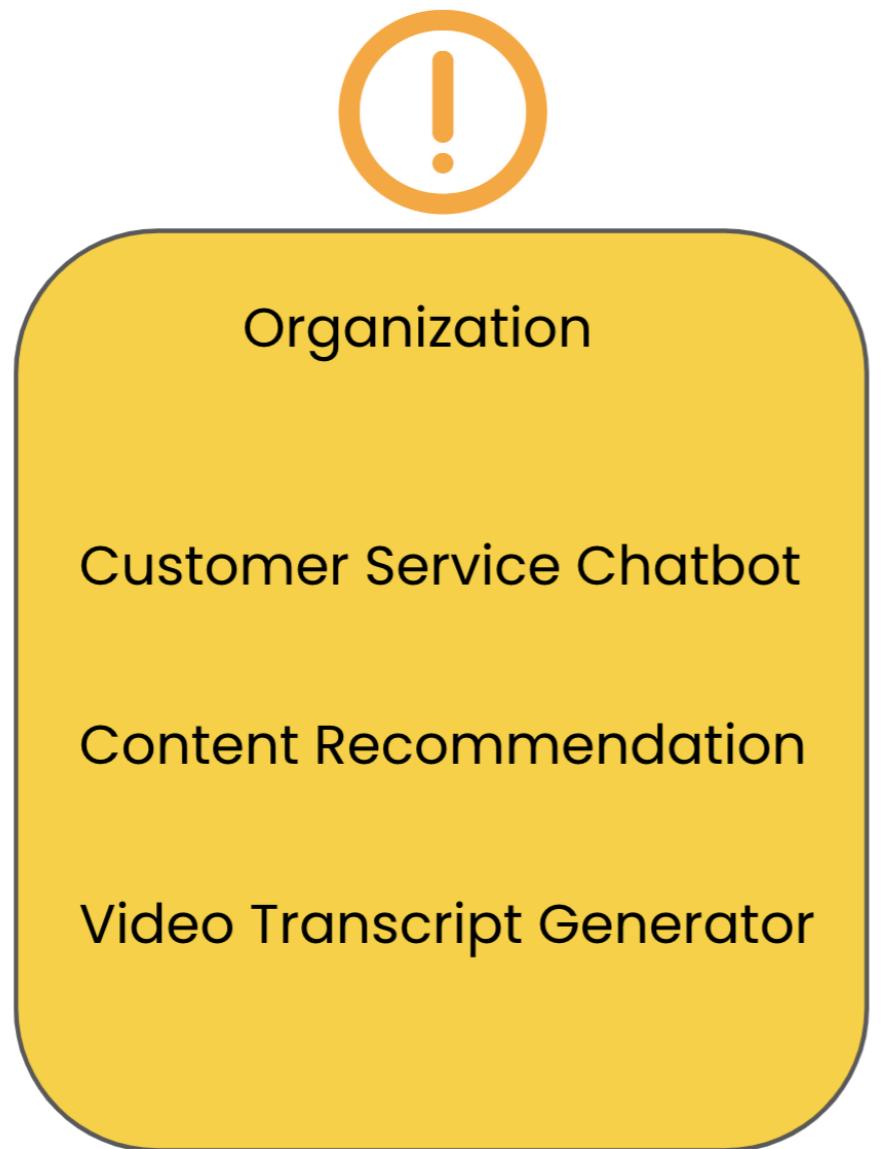
Rate limits

- Cap on *frequency* or *size* of API requests

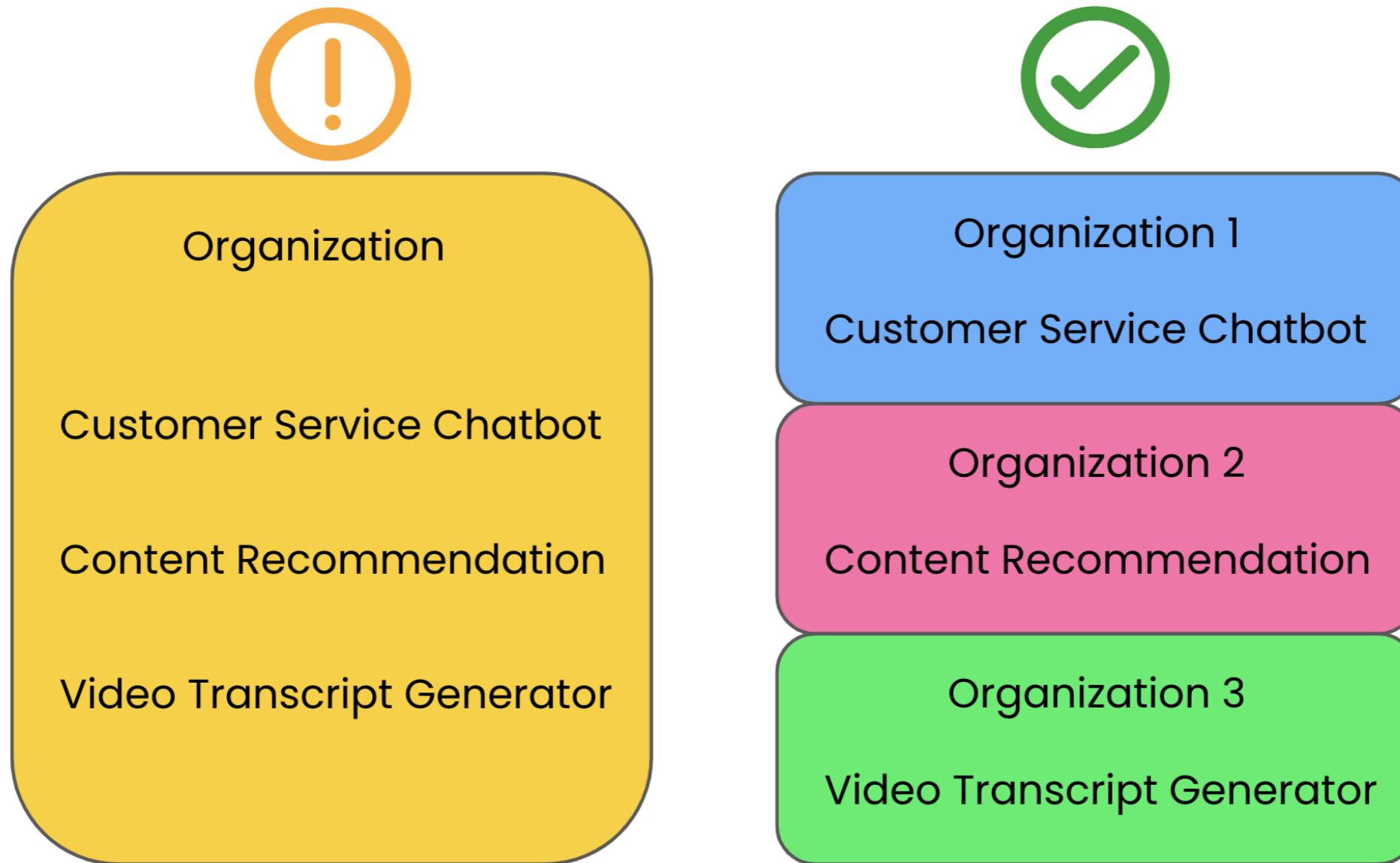


¹ <https://platform.openai.com/docs/guides/rate-limits>

Organization structure

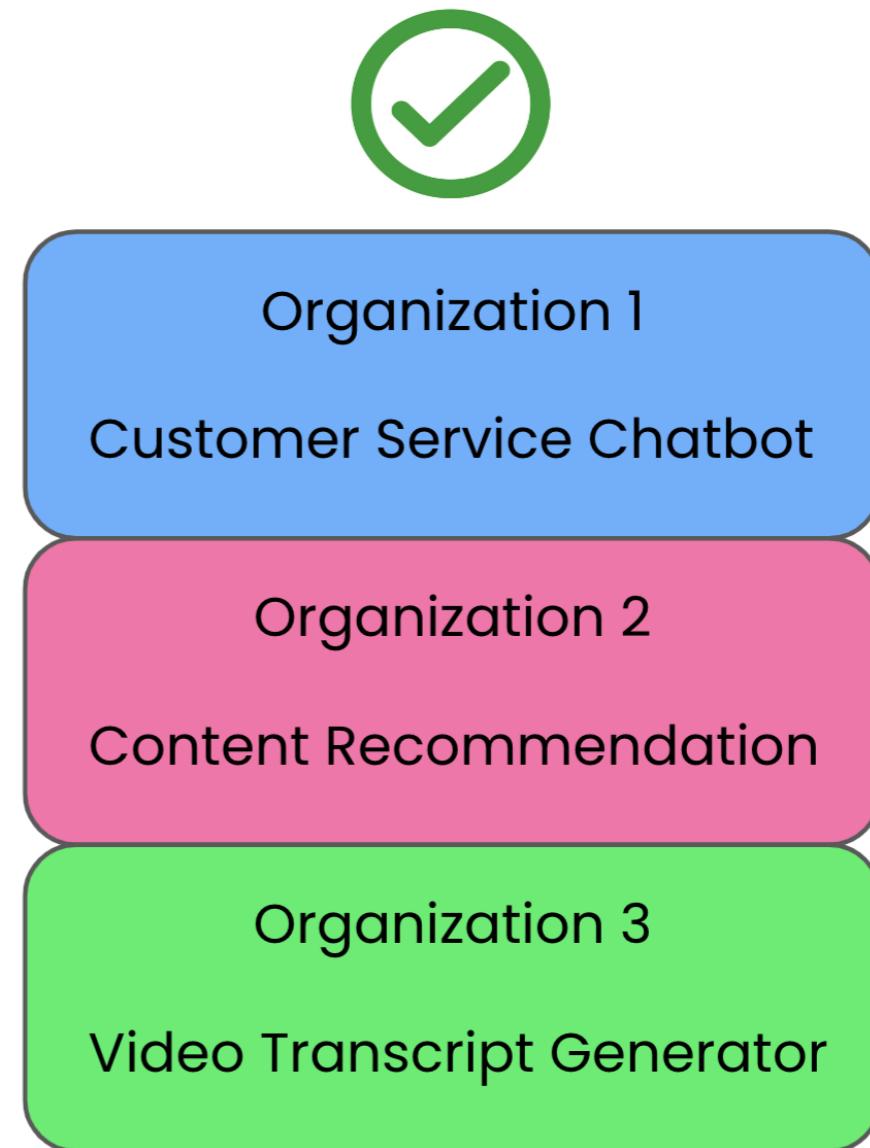


Organization structure



Organization structure

- Distributes requests across organizations
 - Reduced likelihood of hitting rate limits
- Removes *single failure point*
 - Issue with one organization doesn't break all features
- Better usage and billing management



Let's practice!

WORKING WITH THE OPENAI API