What is the OpenAI API?

WORKING WITH THE OPENAI API

James Chapman
Curriculum Manager, DataCamp

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

[1] Image Credit: OpenAI

OpenAI, ChatGPT, and the OpenAI API

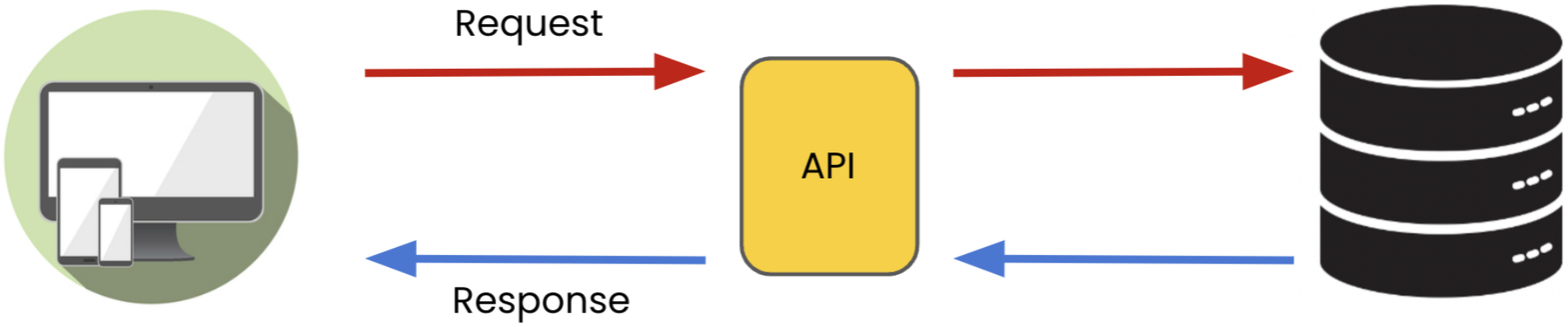
- **OpenAI** = AI R&D company
- **ChatGPT** = AI application
- **OpenAI API** = Interface for accessing OpenAI models

OpenAI, ChatGPT, and the OpenAI API

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

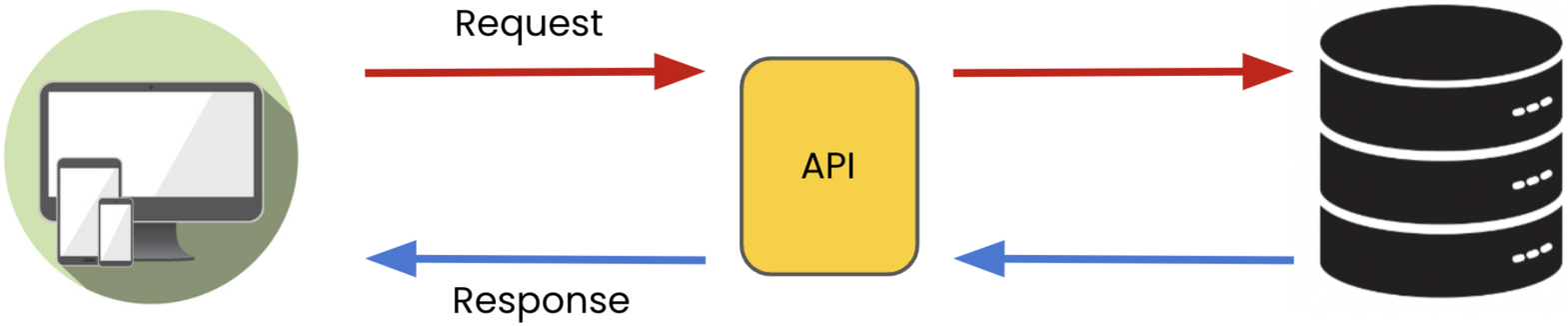
What is an API?

- Application Programming Interface

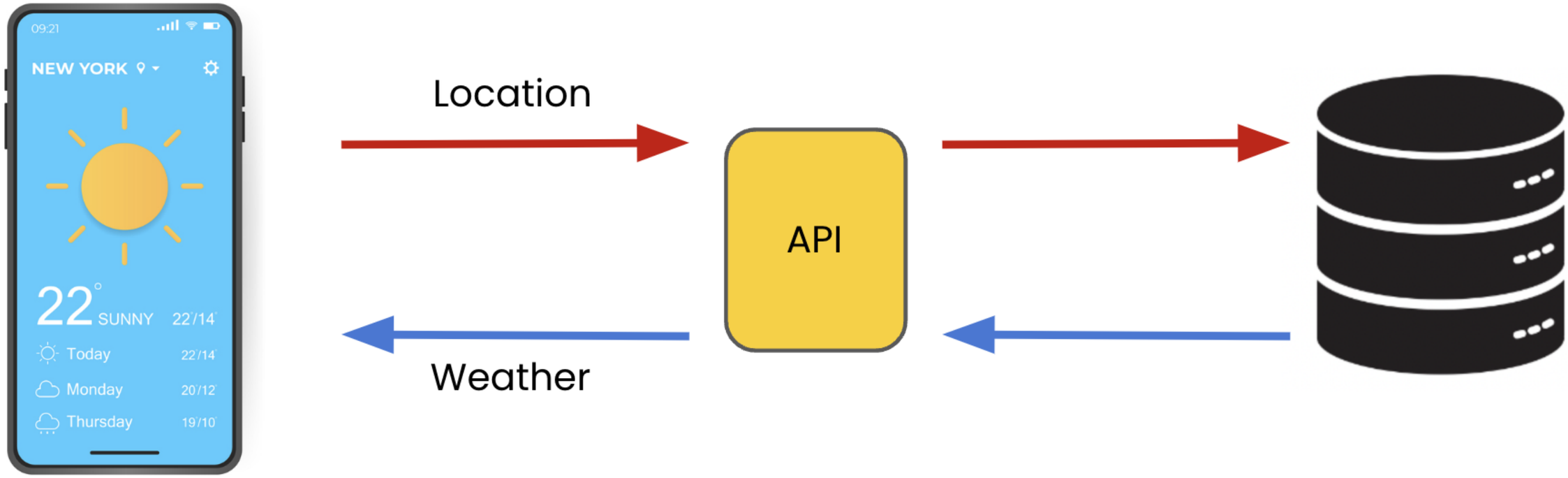


What is an API?

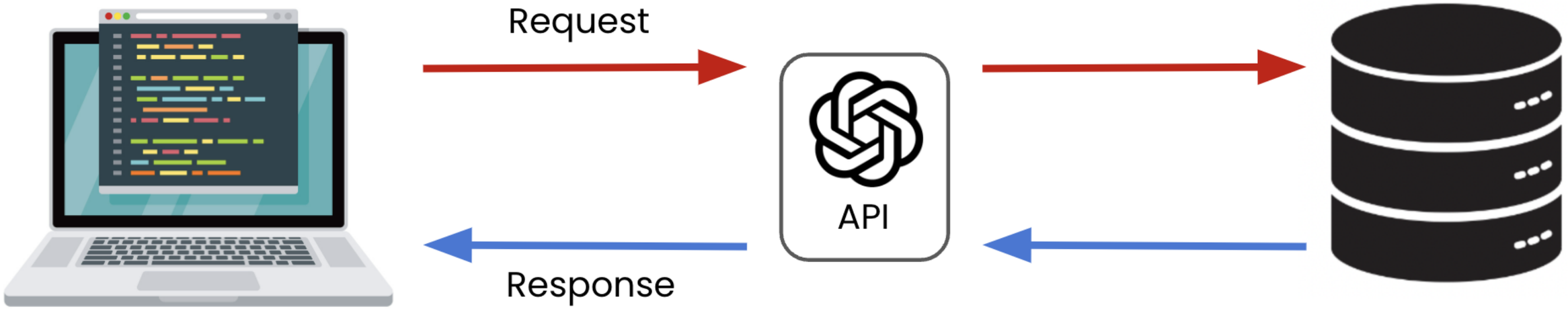
- Application Programming Interface



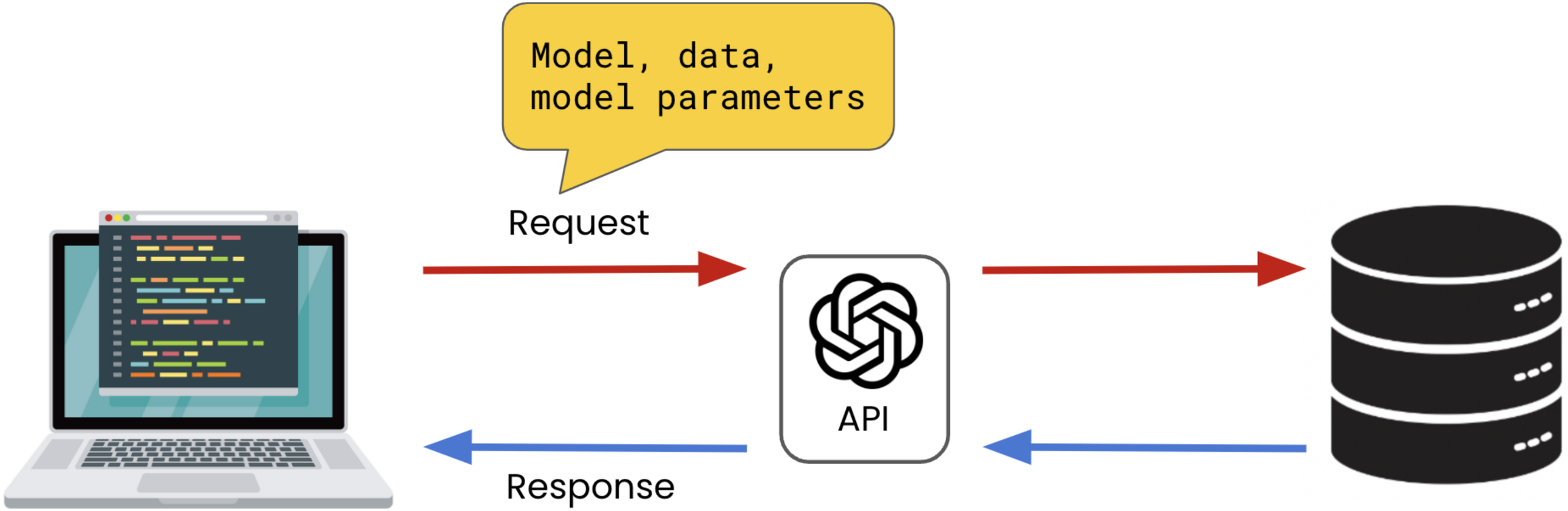
What is an API?



The OpenAI API

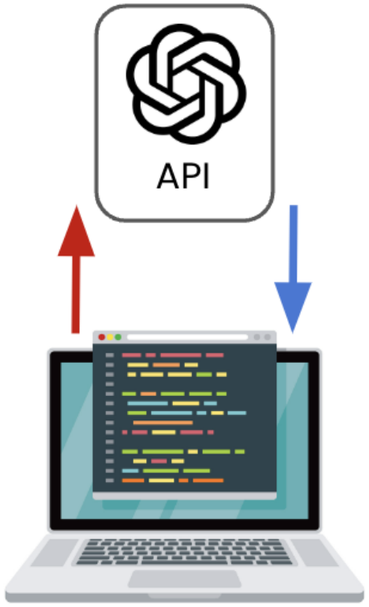


The OpenAI API

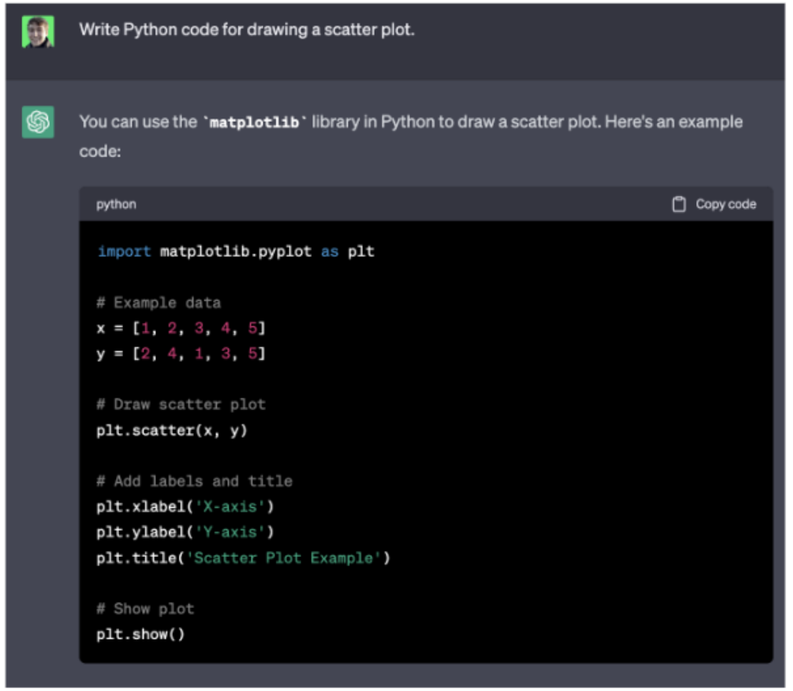


API vs. web interface

- Provides *flexibility* for integrating AI into products, experiences, and processes
- Interact via programming language

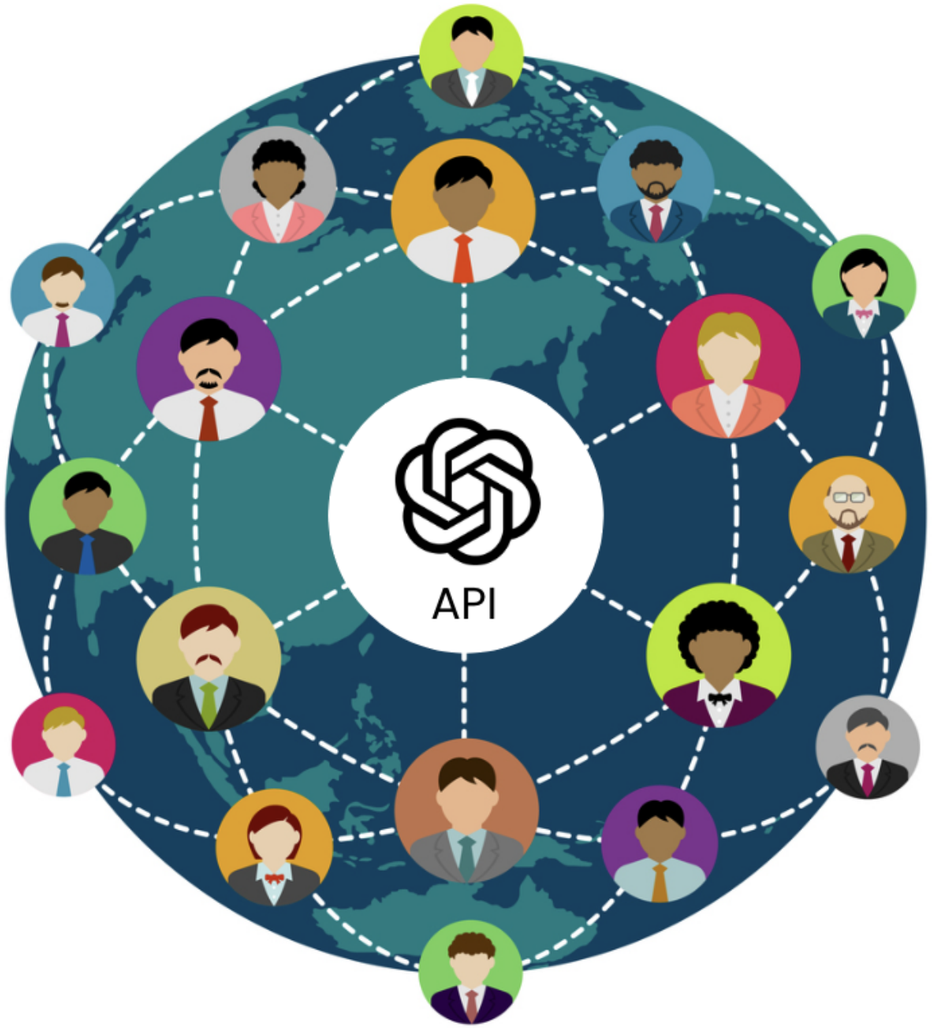


- Very little setup
- Sufficient for *streamlining* individual's workflows



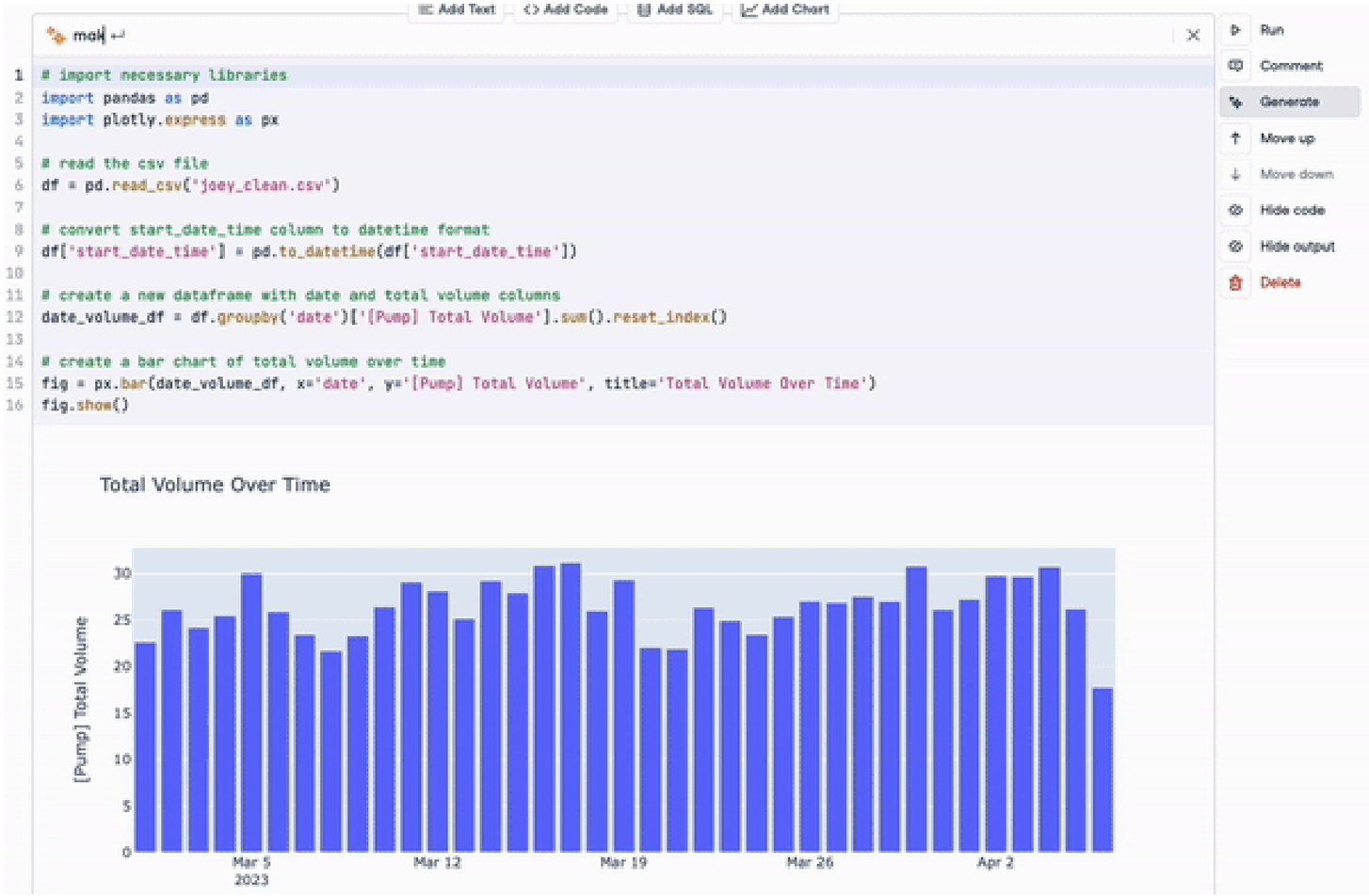
Why the OpenAI API?

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



Building AI applications

DataCamp Workspace = Cloud-based IDE



Let's practice!

WORKING WITH THE OPENAI API

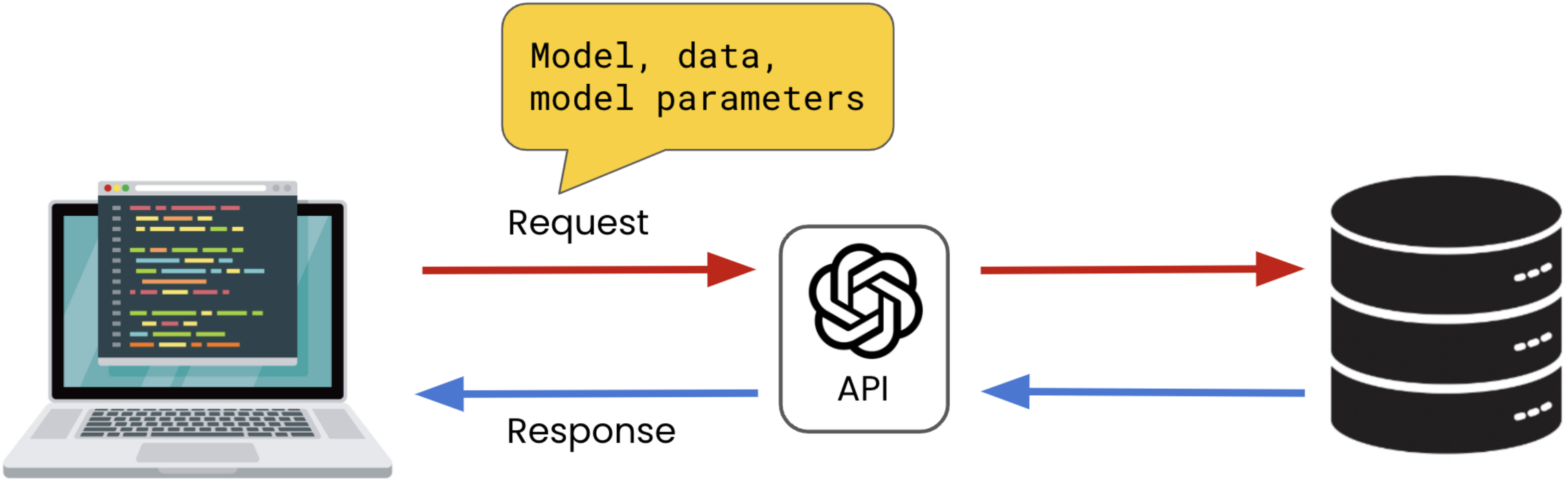
Making requests to the OpenAI API

WORKING WITH THE OPENAI API

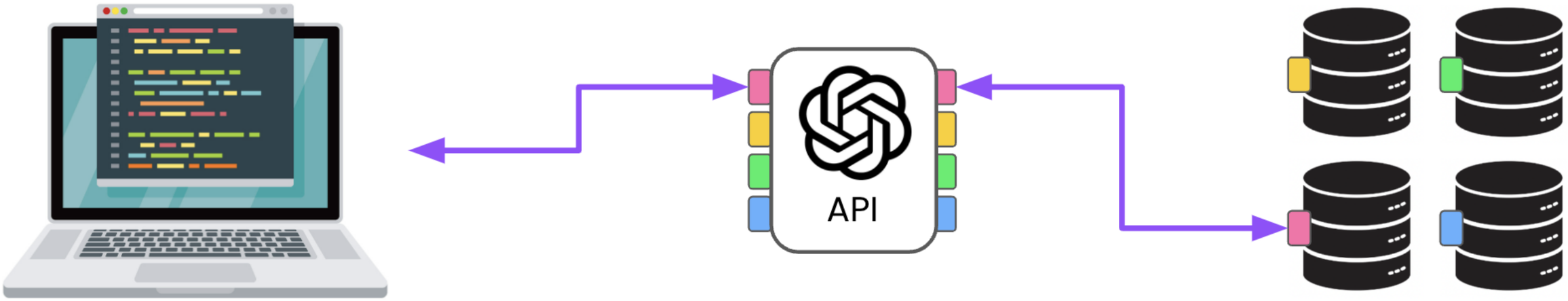
James Chapman
Curriculum Manager, DataCamp

datacamp

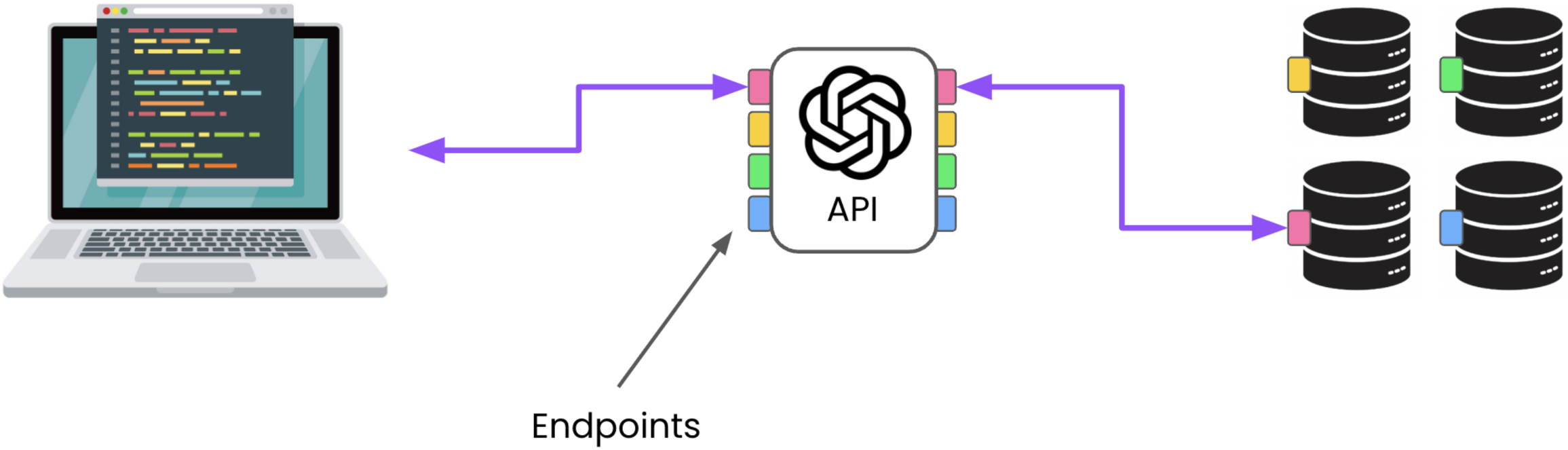
APIs recap...



API endpoints



API endpoints



API endpoints

- **Endpoints** → API access point designed for specific interactions

API authentication

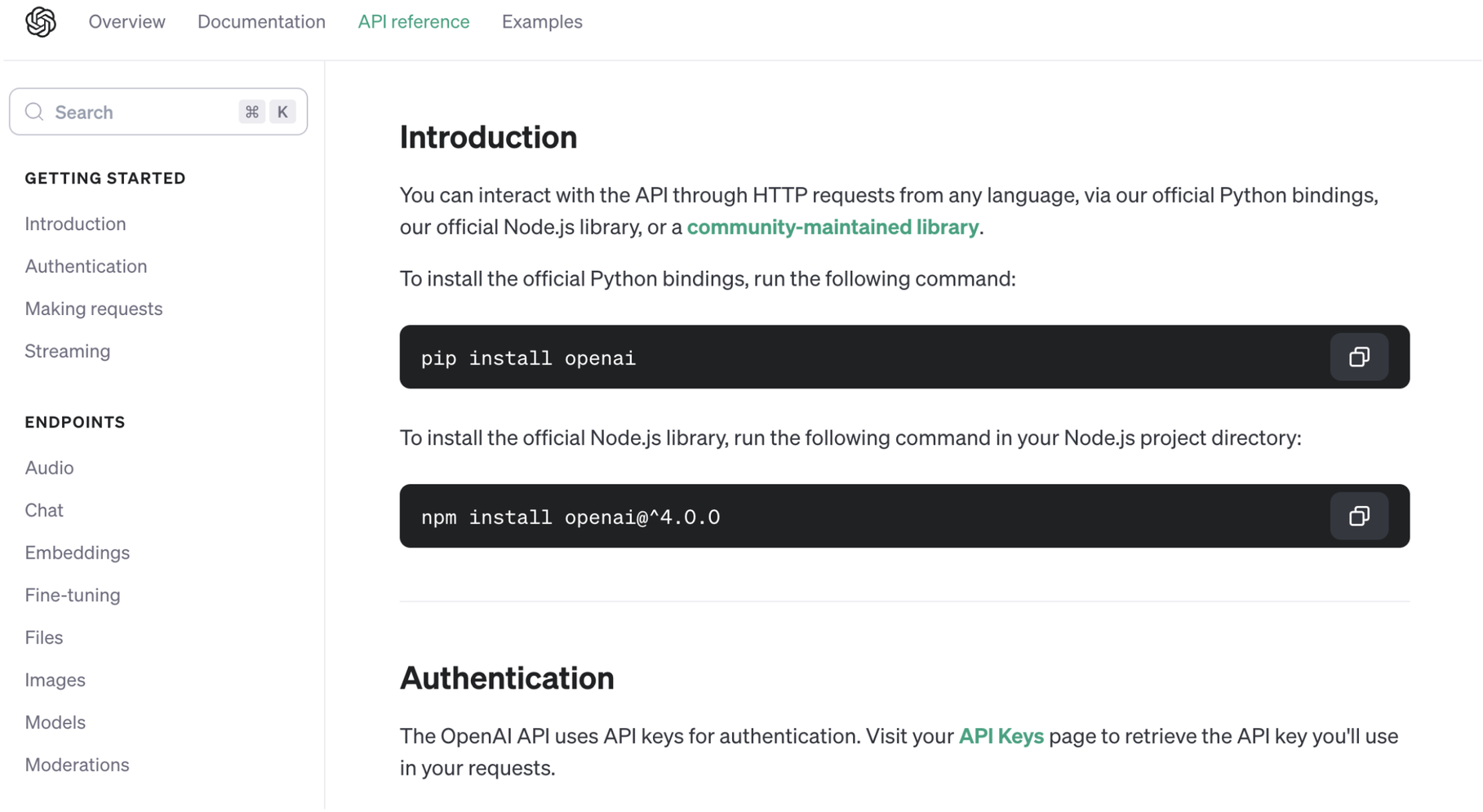
- **Authentication** → Controls on access to API endpoint services (often unique **key**)

API usage costs

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

[1] https://openai.com/pricing

API documentation



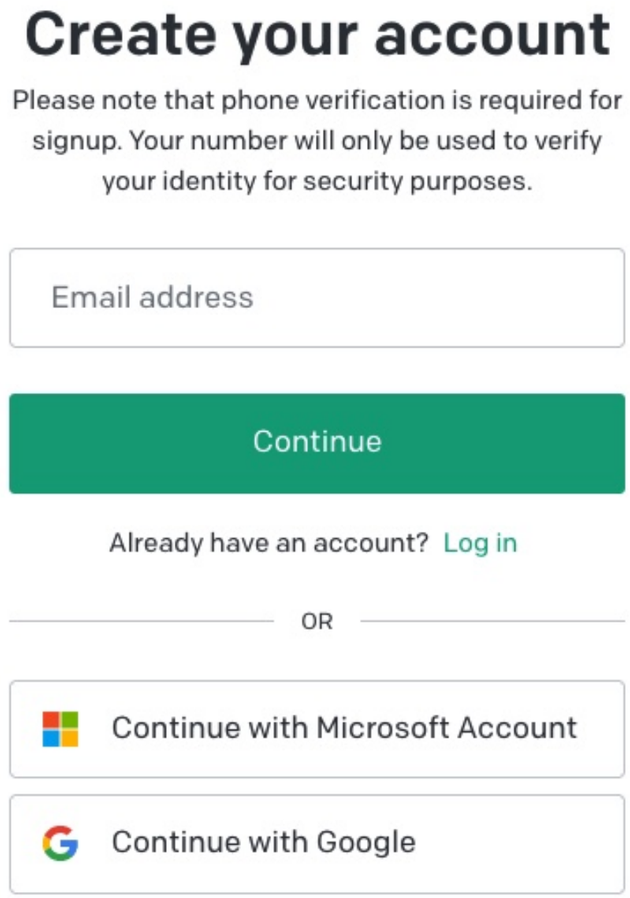
[1] https://platform.openai.com/docs/api-reference

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

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)

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

Making a request

```
from openai import OpenAI

client = OpenAI(api_key="ENTER YOUR KEY HERE")

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

print(response)
```

The response

```
Completion(id='cmpl-8SPzX0HgYV1emmoGN6MYjjaRJzkBR', choices=[CompletionChoice(finish_reason='length', index=0,
logprobs=None, text='\n\nThe OpenAI API is an application programming interface provided by OpenAI, a')],
created=1701783787, model='gpt-3.5-turbo-instruct', object='text_completion', system_fingerprint=None,
usage=CompletionUsage(completion_tokens=16, prompt_tokens=7, total_tokens=23))
```

Additional spacing:

```
Completion(id='cmpl-8S0D6VZBpM8Vy0fZf6atE71b1Zdvm',
           choices=[CompletionChoice(finish_reason='length', index=0, logprobs=None,
                                     text='\n\nThe OpenAI API is an application programming interface provided by OpenAI, a')],
           created=1701684684,
           model='gpt-3.5-turbo-instruct',
           object='text_completion',
           system_fingerprint=None,
           usage=CompletionUsage(completion_tokens=16, prompt_tokens=7, total_tokens=23))
```

Interpreting the response

```
print(response.choices)
```

```
[CompletionChoice(finish_reason='length', index=0, logprobs=None,
  text='\n\nThe OpenAI API is an application programming interface provided by OpenAI, a')]
```

Interpreting the response

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

```
CompletionChoice(finish_reason='length', index=0, logprobs=None,
  text='\n\nThe OpenAI API is an application programming interface provided by OpenAI, a')
```

Interpreting the response

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

```
The OpenAI API is an application programming interface provided by OpenAI, a
```

Converting the response into a dictionary

```
print(response.model_dump())
```

```
{'id': 'cmpl-8S0KeXQsQVl9i9CagkHZjb2Ujhy2A',
 'choices': [{'finish_reason': 'length', 'index': 0, 'logprobs': None,
              'text': '\n\nThe OpenAI API is an application programming interface provided by OpenAI, a'}],
 'created': 1701685152,
 'model': 'gpt-3.5-turbo-instruct',
 'object': 'text_completion',
 'system_fingerprint': None,
 'usage': {'completion_tokens': 16, 'prompt_tokens': 7, 'total_tokens': 23}
}
```

Let's practice!

WORKING WITH THE OPENAI API

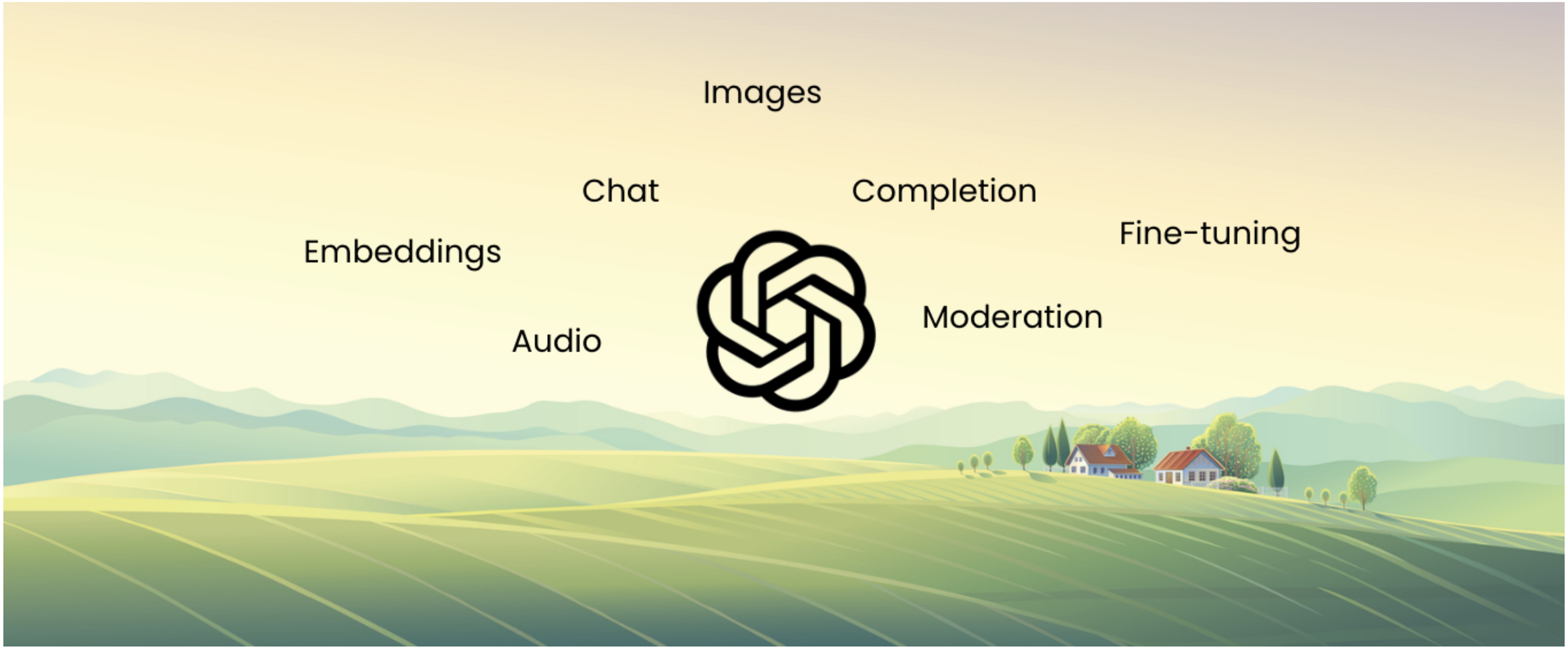
The OpenAI developer ecosystem

WORKING WITH THE OPENAI API

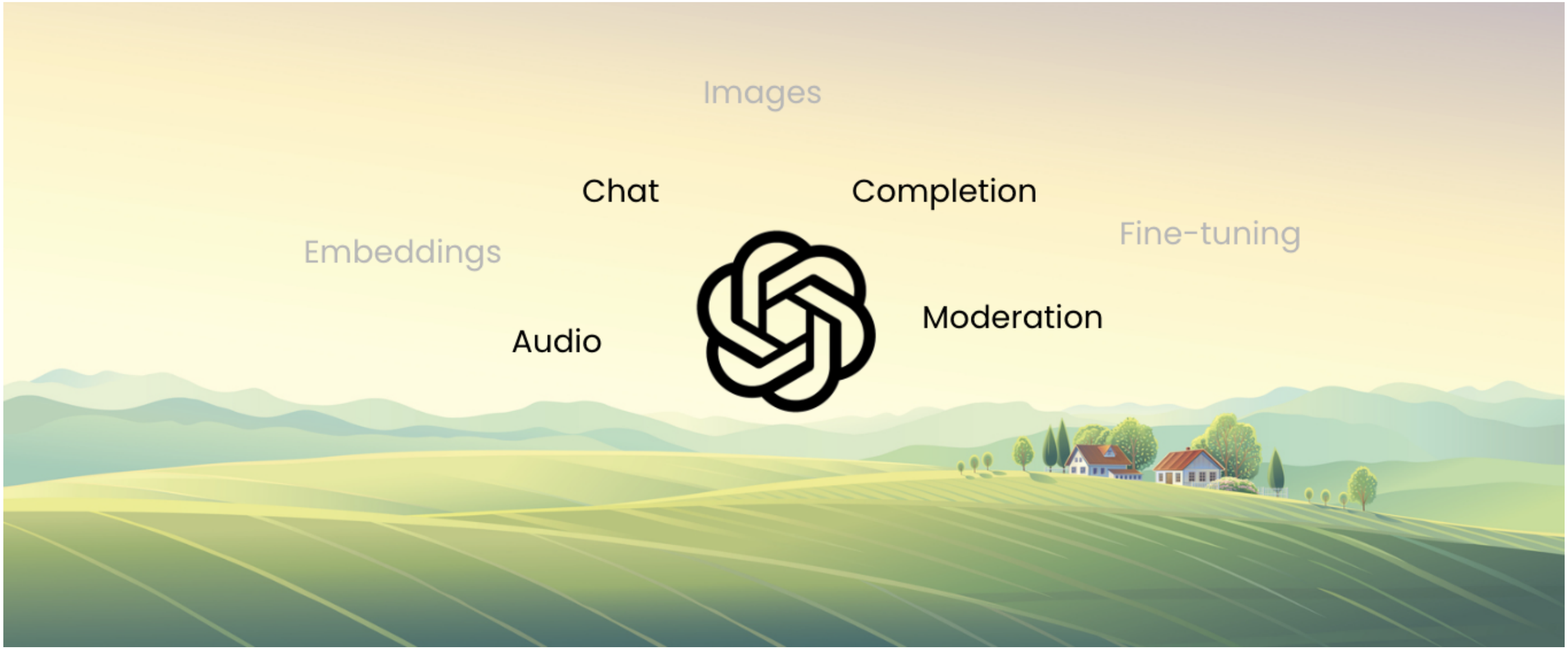
James Chapman
Curriculum Manager, DataCamp

datacamp

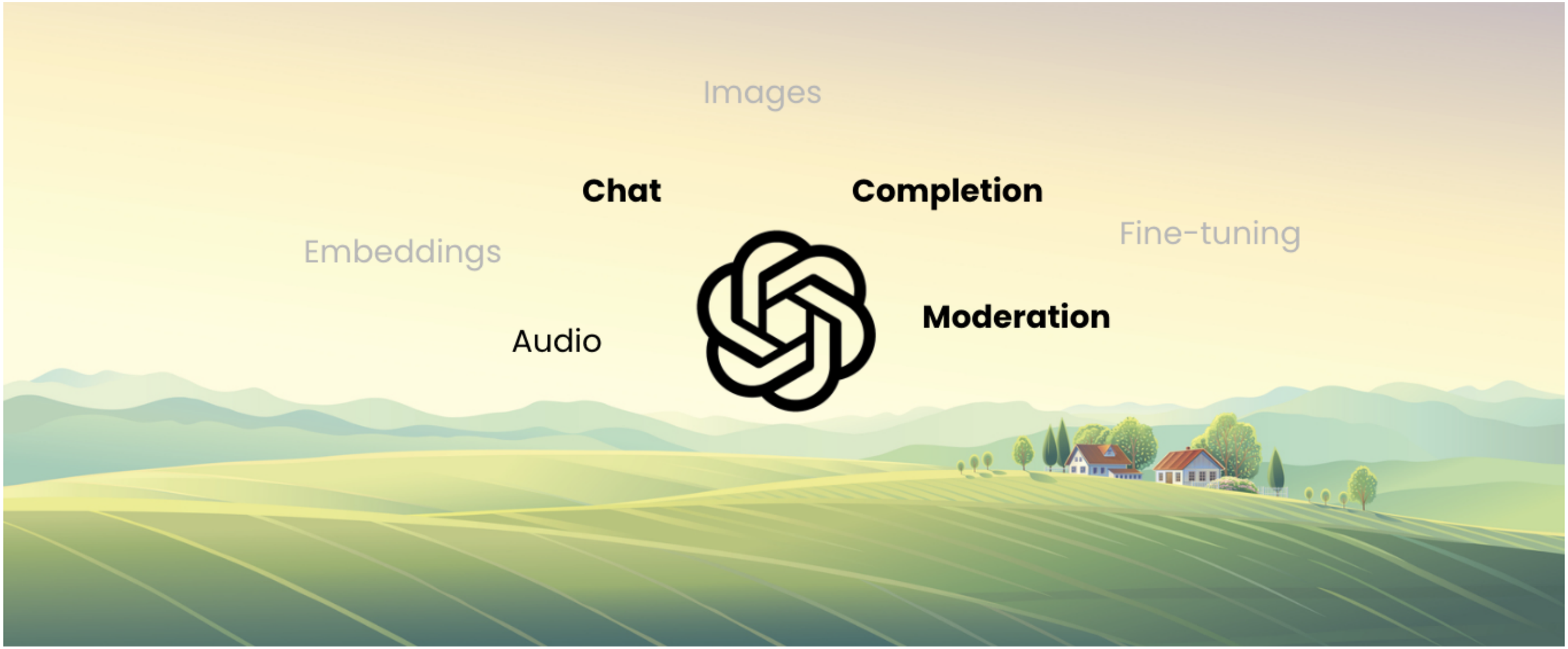
The OpenAI landscape



The OpenAI landscape

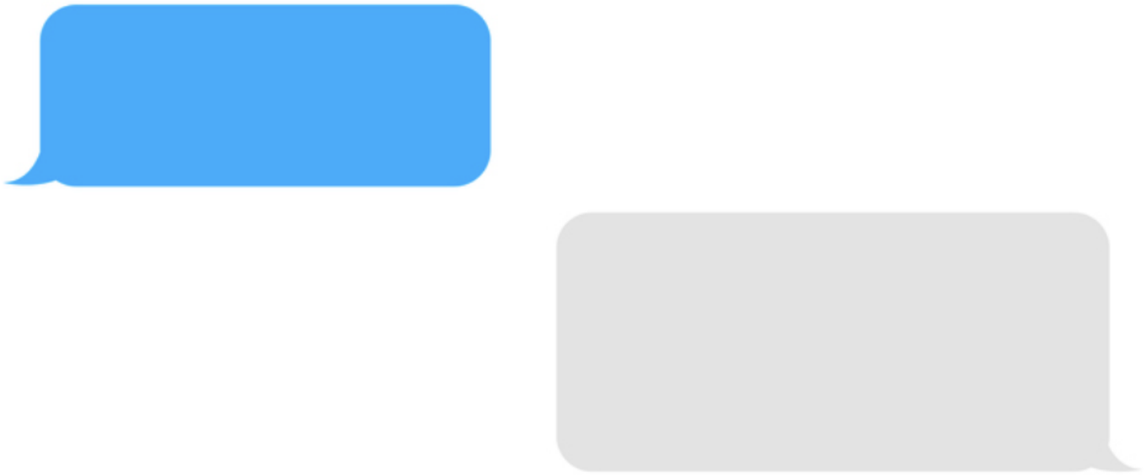


The OpenAI landscape



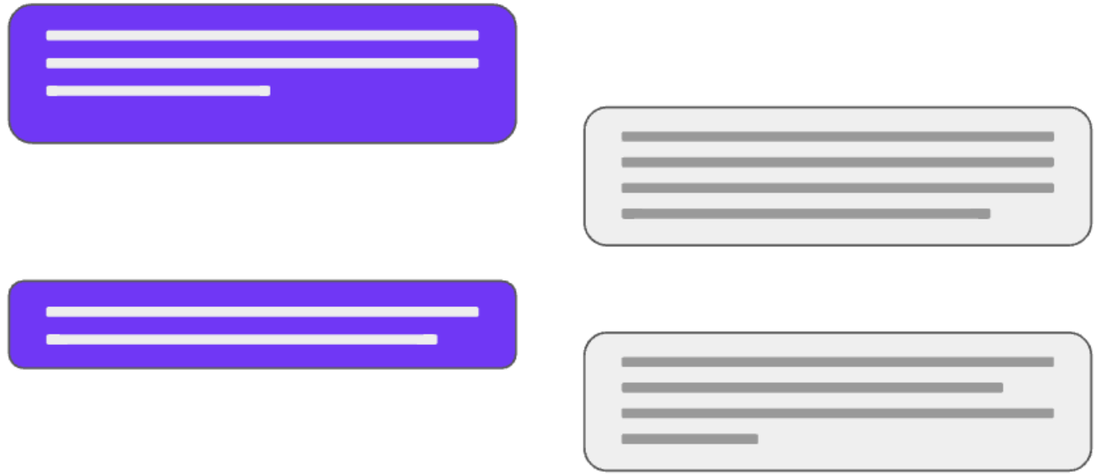
Completions

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



Chat

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

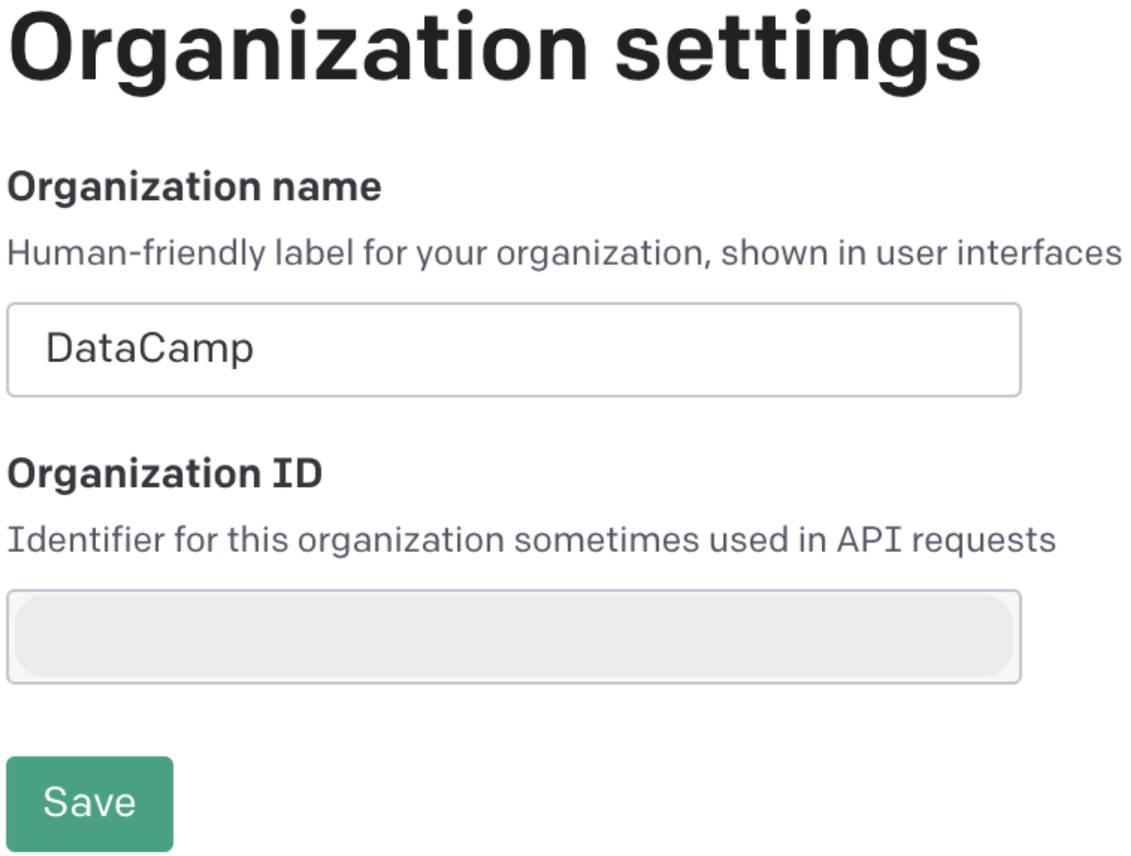


- Chat completions → **Chapter 2**

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



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

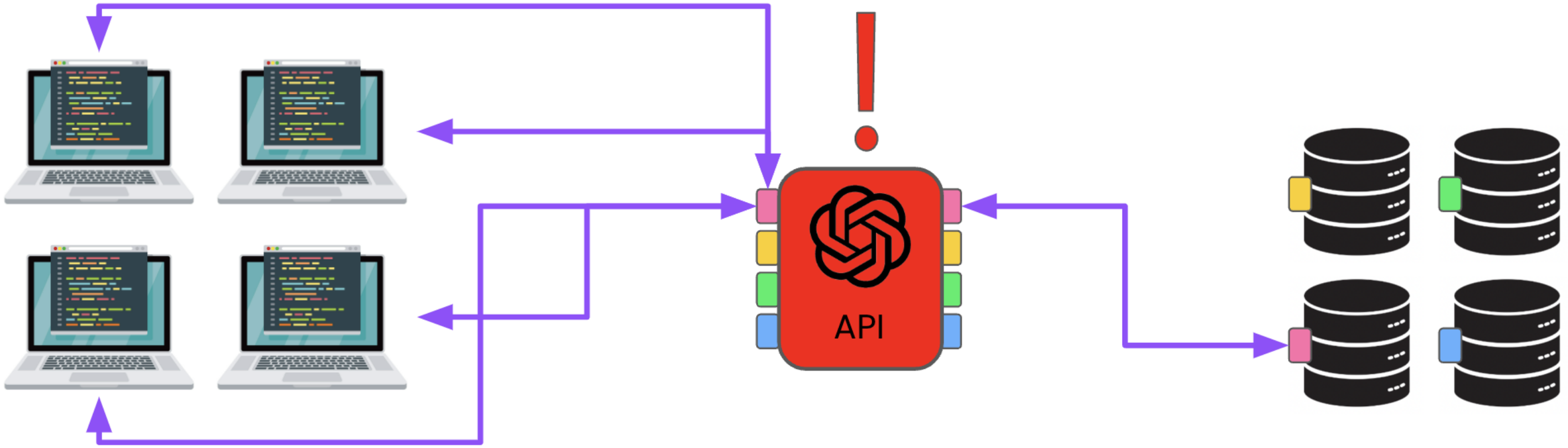
[1] https://platform.openai.com/account/org-settings

Organizations

```
from openai import OpenAI

client = OpenAI(api_key="ENTER YOUR KEY HERE",
                organization = "ENTER ORG ID")

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

print(response)
```

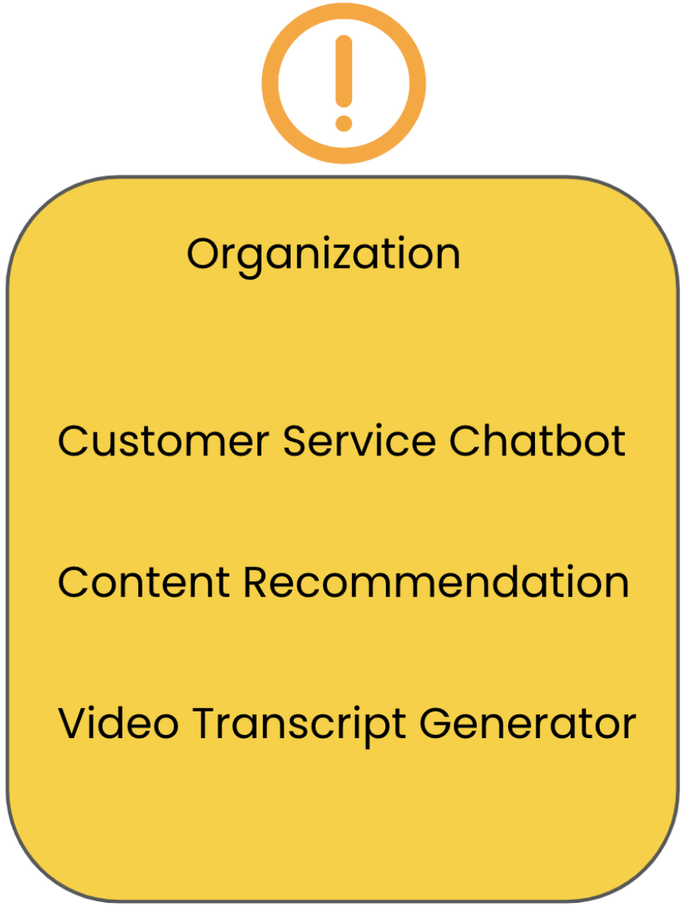
Rate limits

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

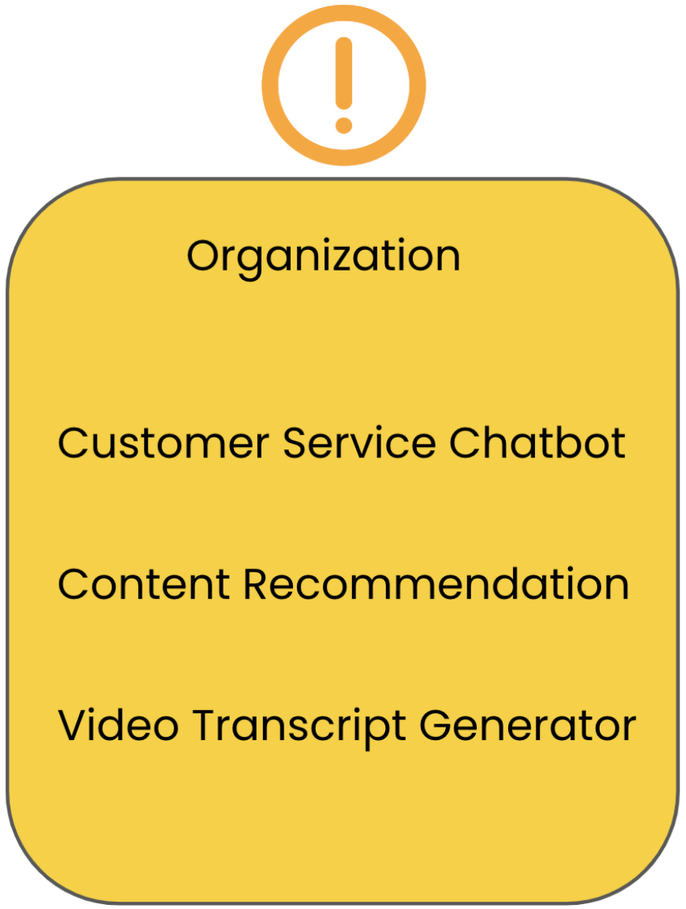


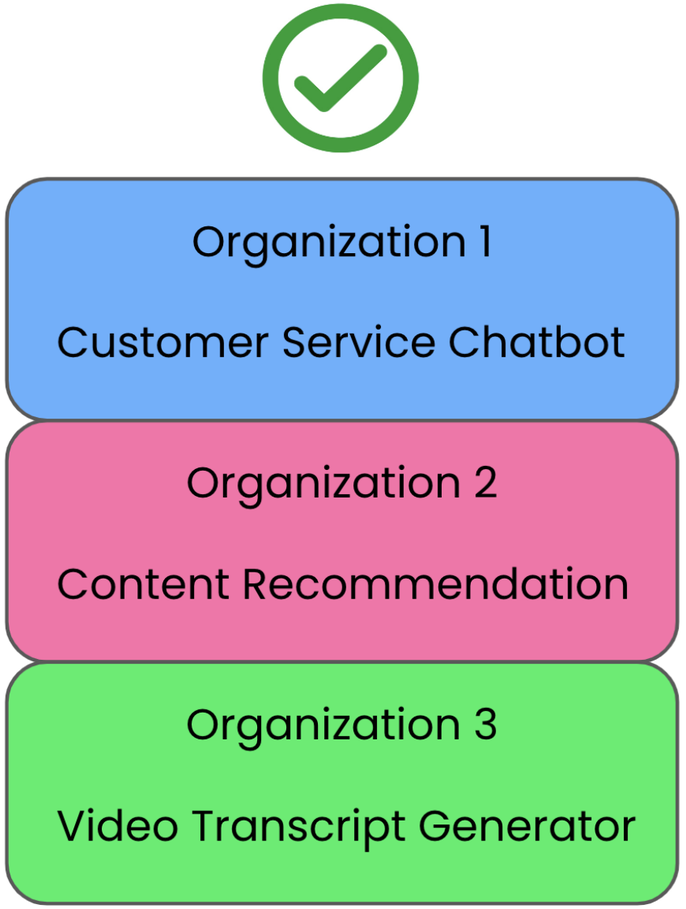
[1] https://platform.openai.com/docs/guides/rate-limits

Organization structure



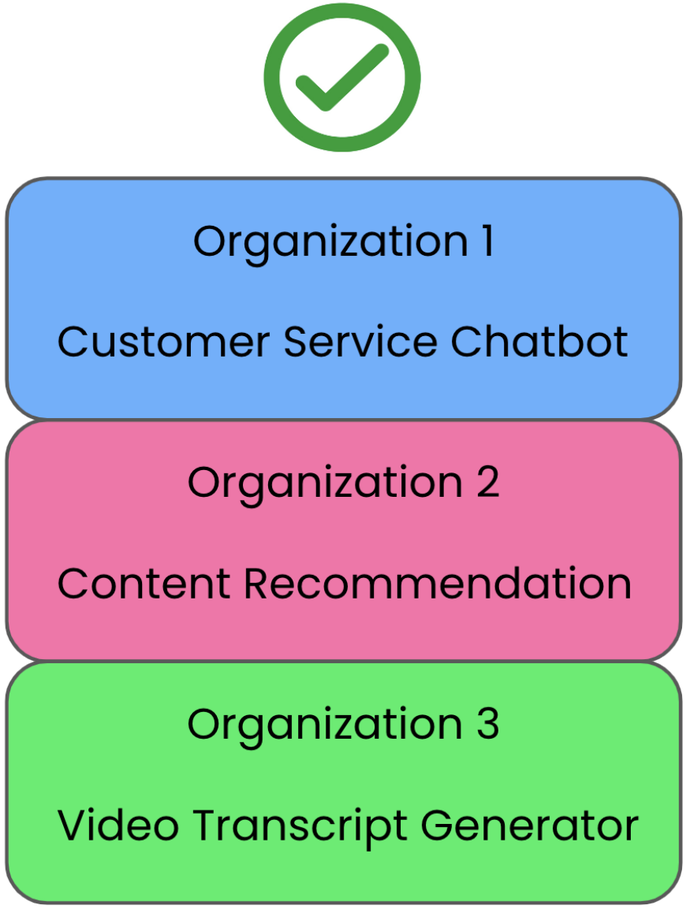
Organization structure





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