

MODULE 3

Interacting with LLM APIs

Up until now...

we only used a couple of parameters when calling an LLM:

```
response = self.client.responses.create(  
    model="gpt-5.4-mini",  
    instructions="You classify whether a job posting ...",  
    input="Title: AI Engineer, Description: ...",  
    text={  
        "format": {  
            "type": "json_schema",  
            "name": "job_classification",  
            "schema": CLASSIFY_SCHEMA,  
            "strict": True,  
        },  
        "verbosity": "low",  
    },  
)
```



Your Turn.

Refactor your project to use pydantic models instead of raw JSON schemas

From Fridge to Recipe



1 Snap

Take a quick photo of the inside of your fridge



2 Analyse

A vision model detects the ingredients it can see



3 Cook

Get recipe ideas tailored to what you already have

Passing an Image to an LLM API

Three ways to send an image — pick the one that fits your use case



Web URL

Use when the image is already public on the web



Base64

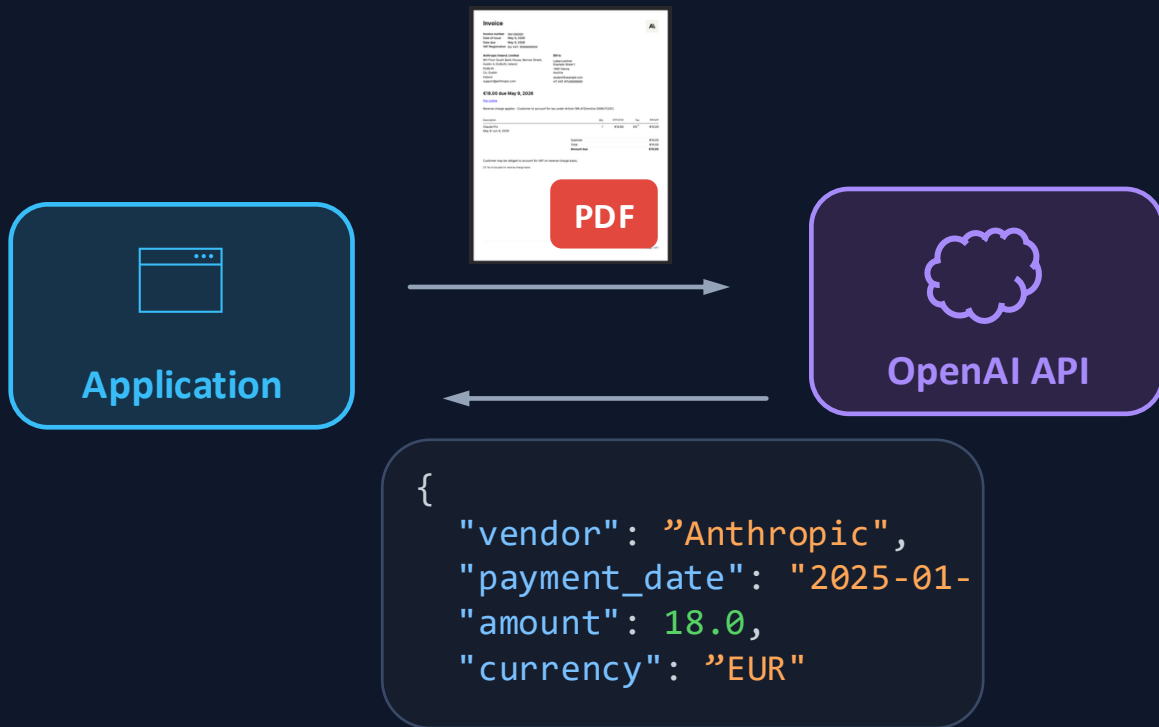
Use for a local image you only need for one request



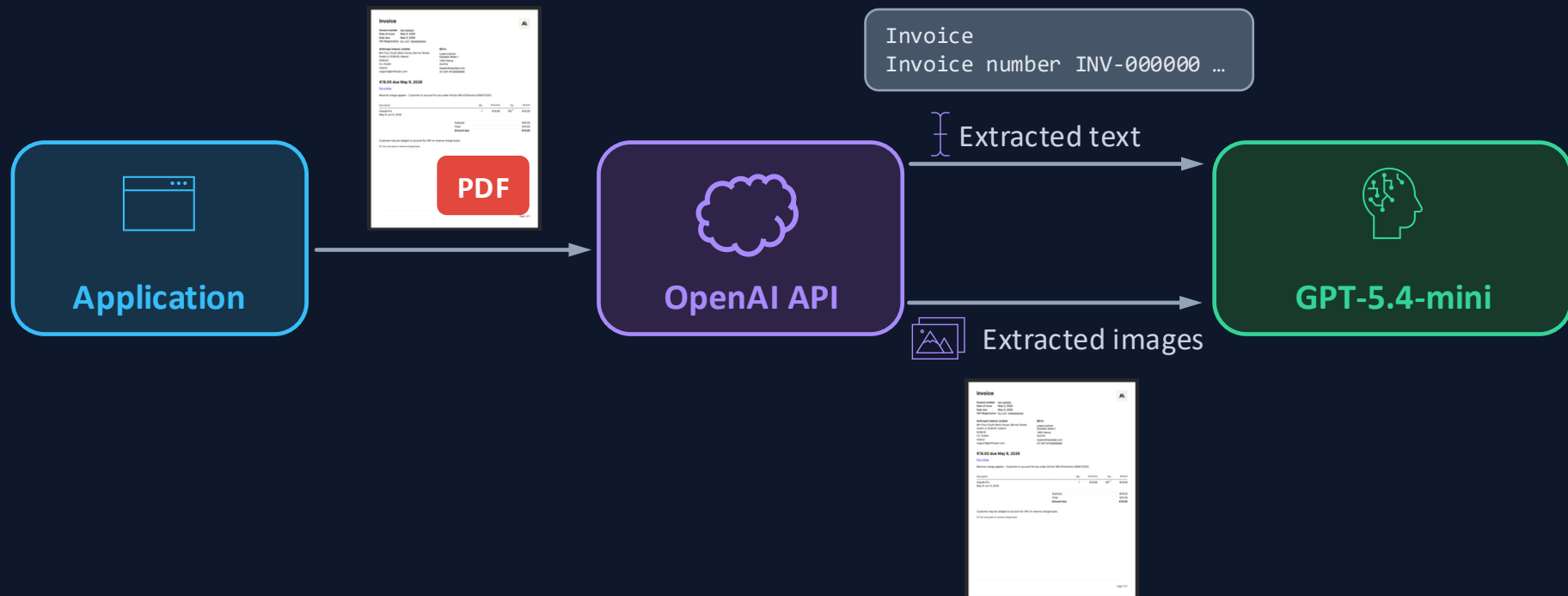
File upload

Use to reuse the same image across multiple requests

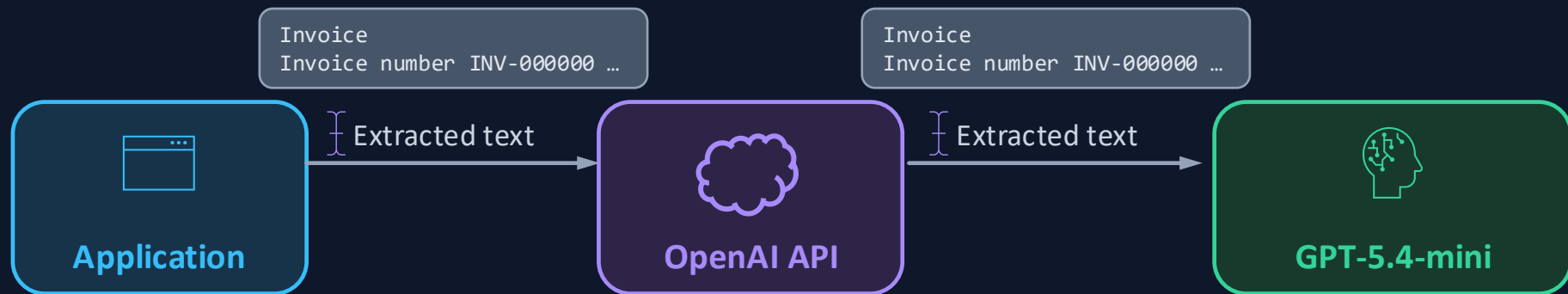
Invoice Classification and Extraction



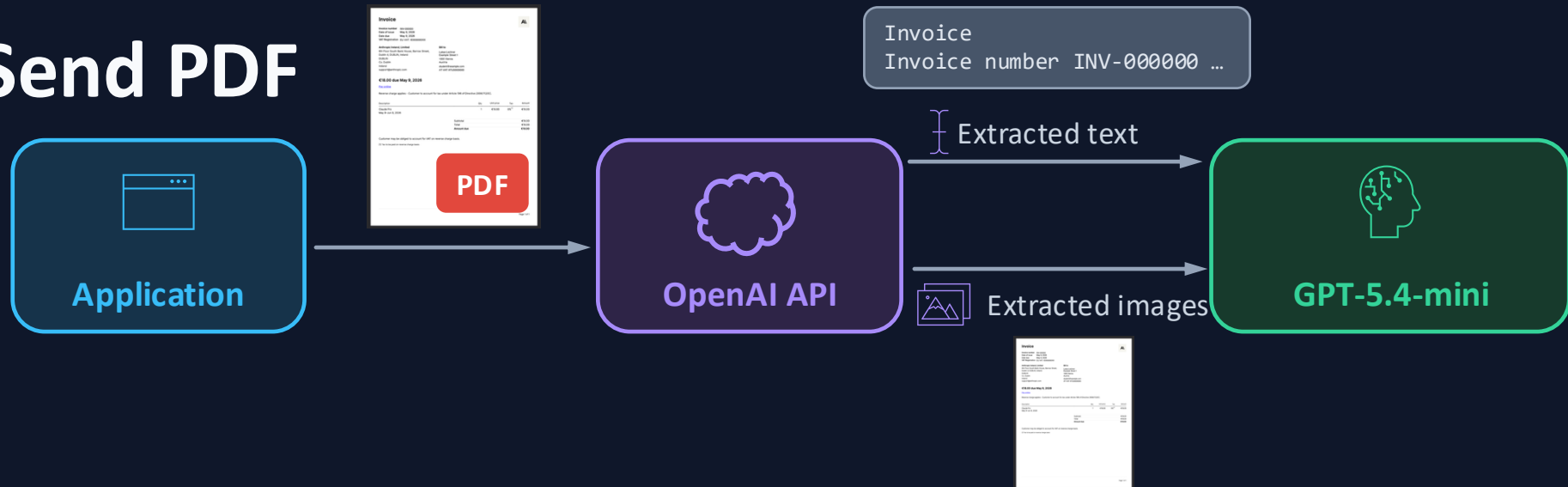
Processing a PDF with the OpenAI API



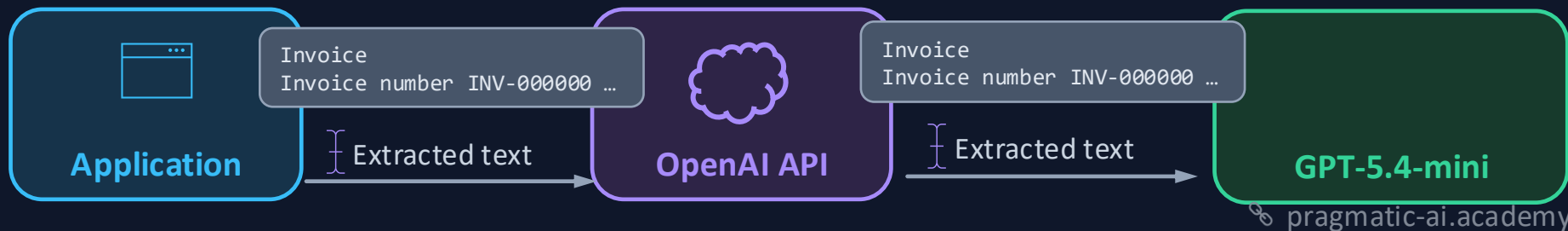
Processing a PDF with the OpenAI API



Send PDF



Send extracted Text



Passing a File to an LLM API

Three ways to send a file — pick the one that fits your use case



Web URL

Use when the file is already public on the web



Base64

Use for a local file you only need for one request



File upload

Use to reuse the same file across multiple requests