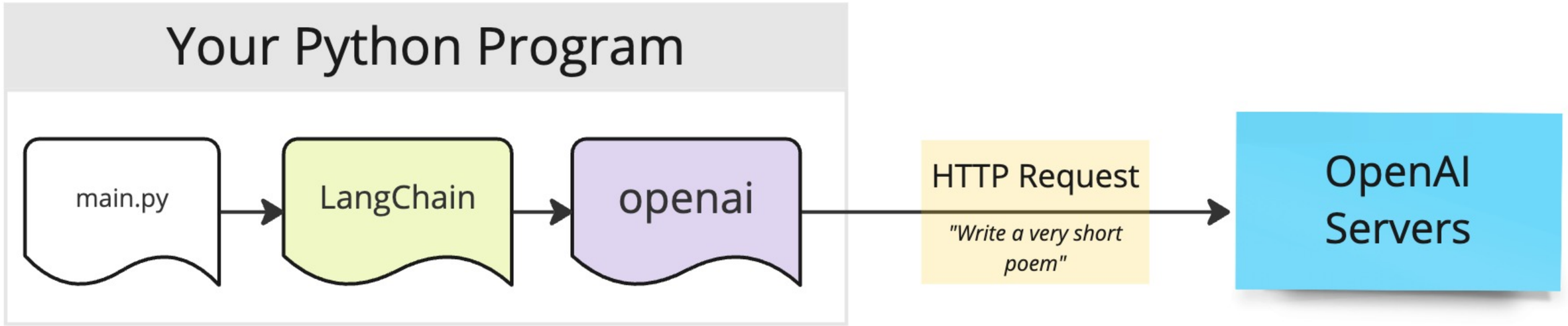
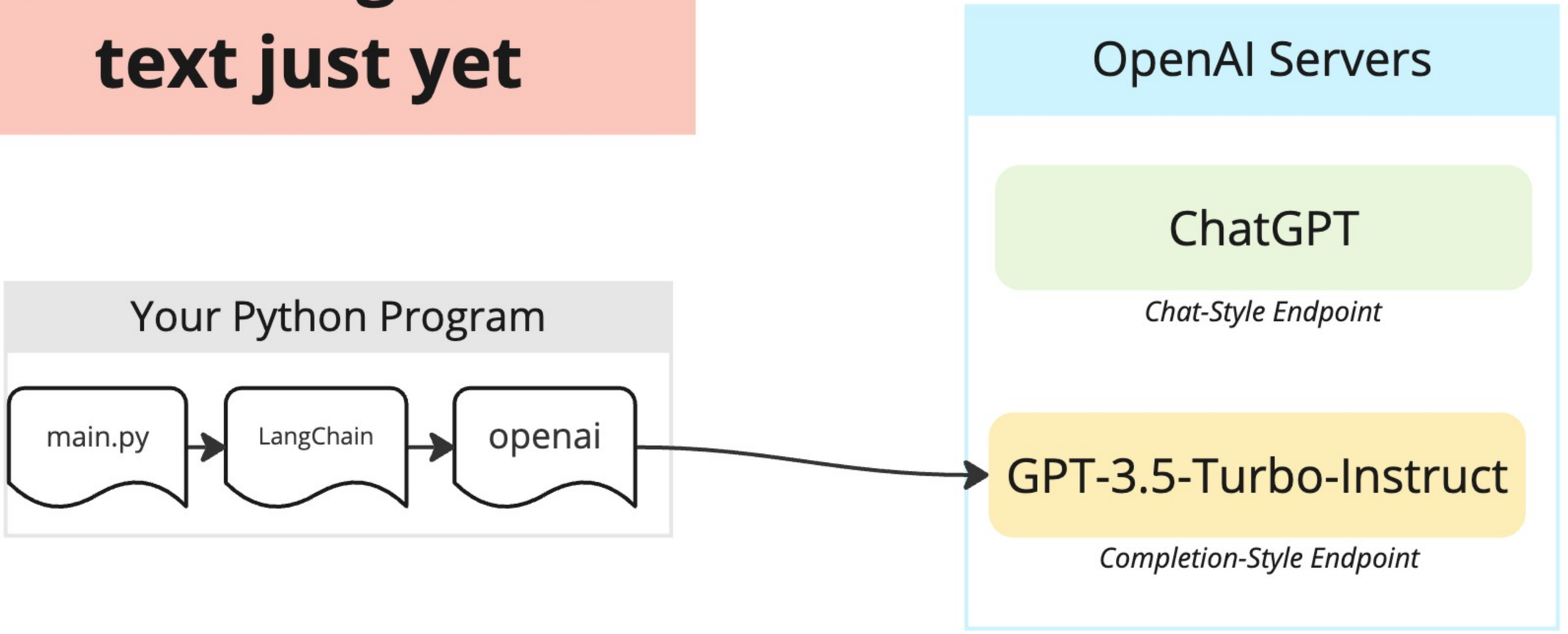

Your Python Program
main.py
LangChain
openai
HTTP Request
"Write a very short poem"
OpenAI Servers

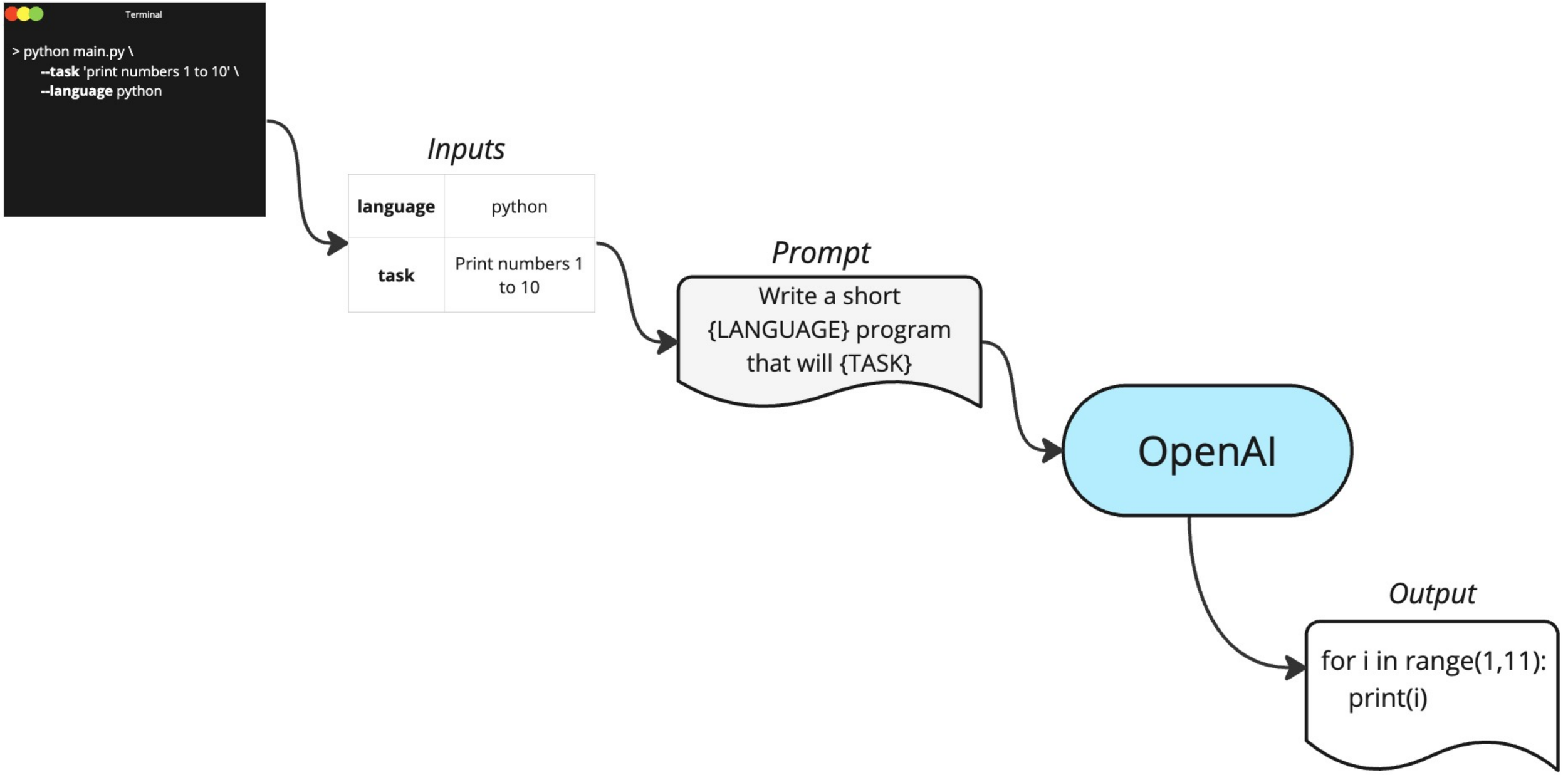
We aren't using ChatGPT to generate text just yet

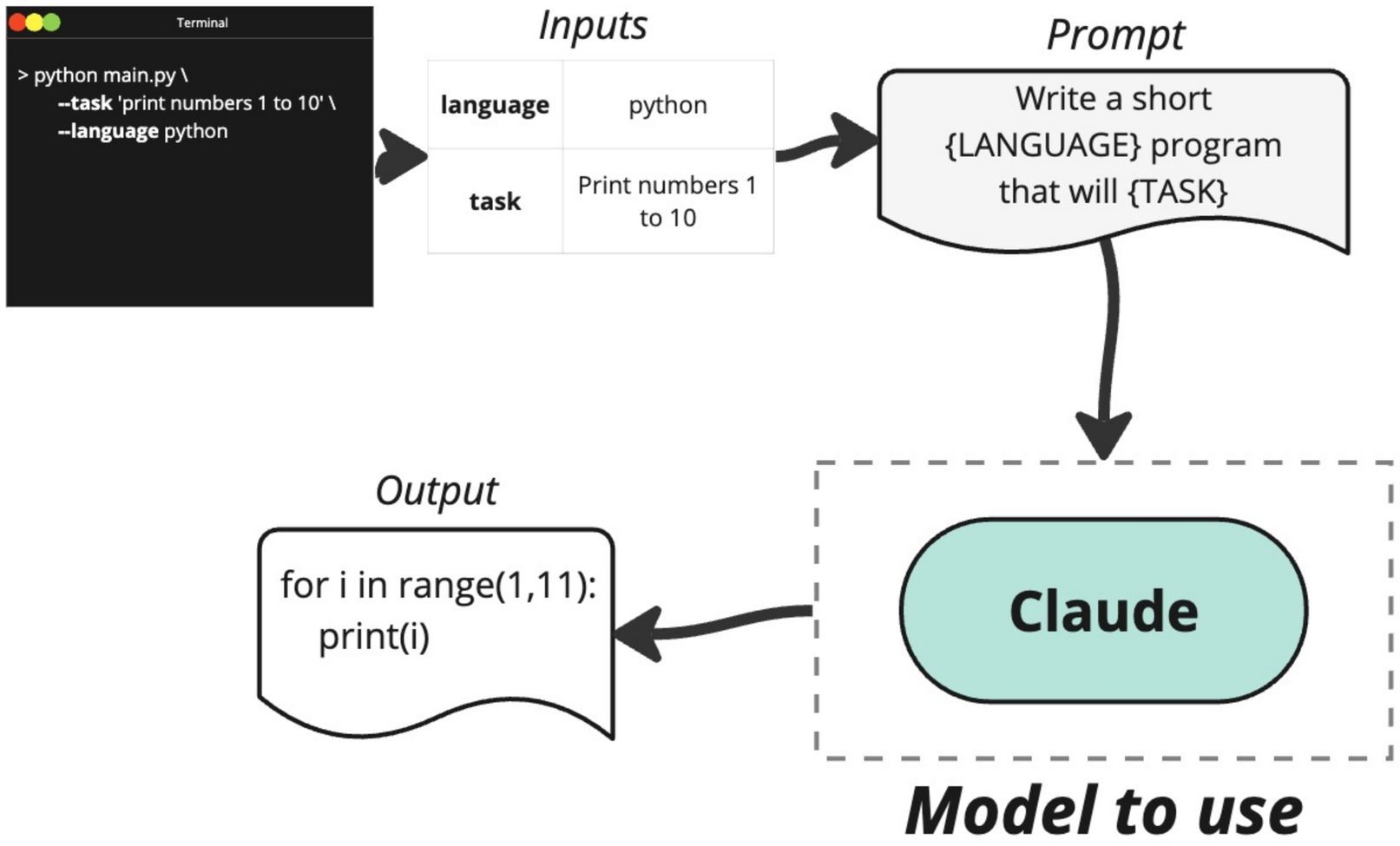


Two Big Goals of LangChain

Provide tools to automate each step of a text generation pipeline

Make it easy to connect tools together

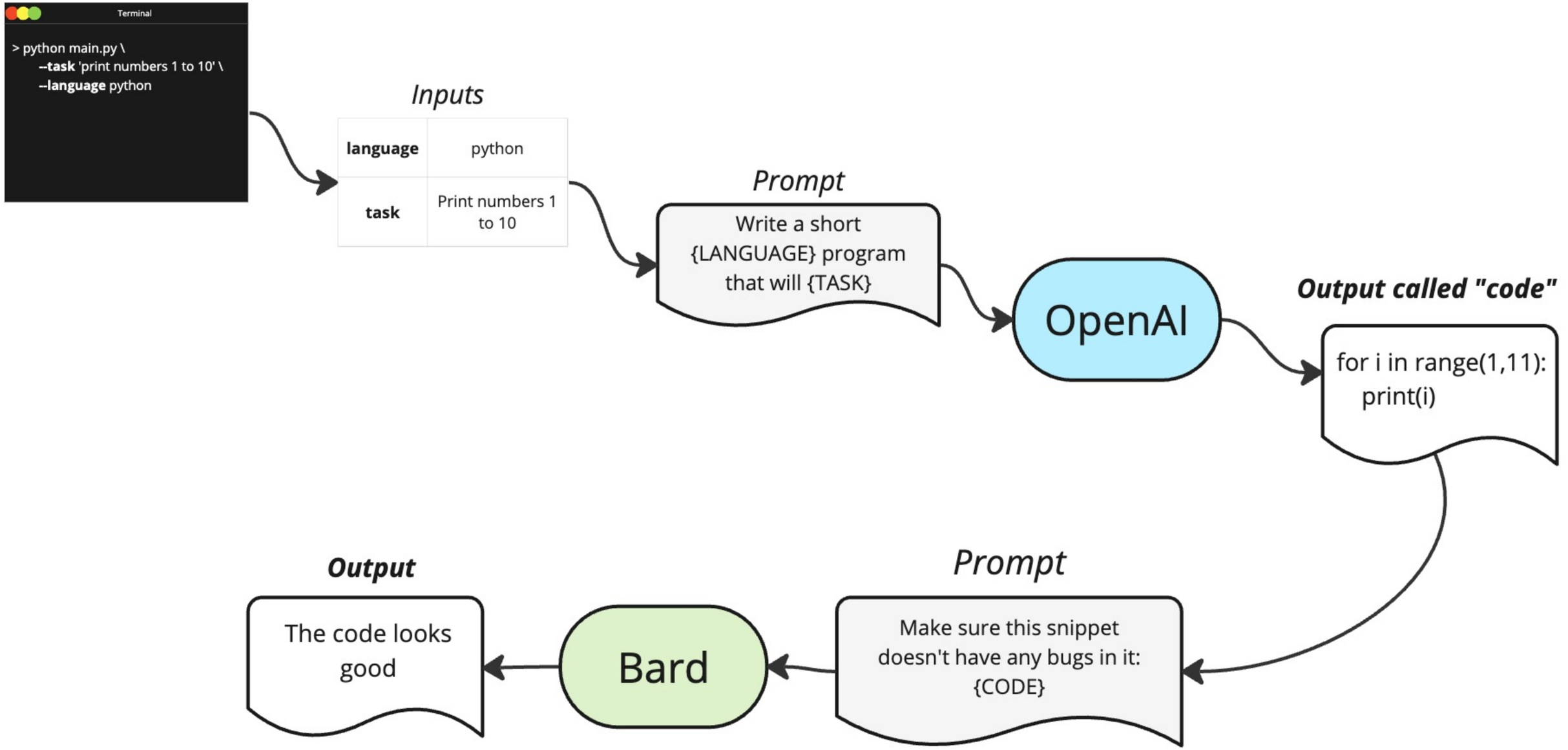

Terminal
> python main.py \
--task 'print numbers 1 to 10' \
--language python
Inputs
language
python
task
Print numbers 1 to 10
Prompt
Write a short {LANGUAGE} program that will {TASK}
OpenAI
Output
for i in range(1,11):
print(i)

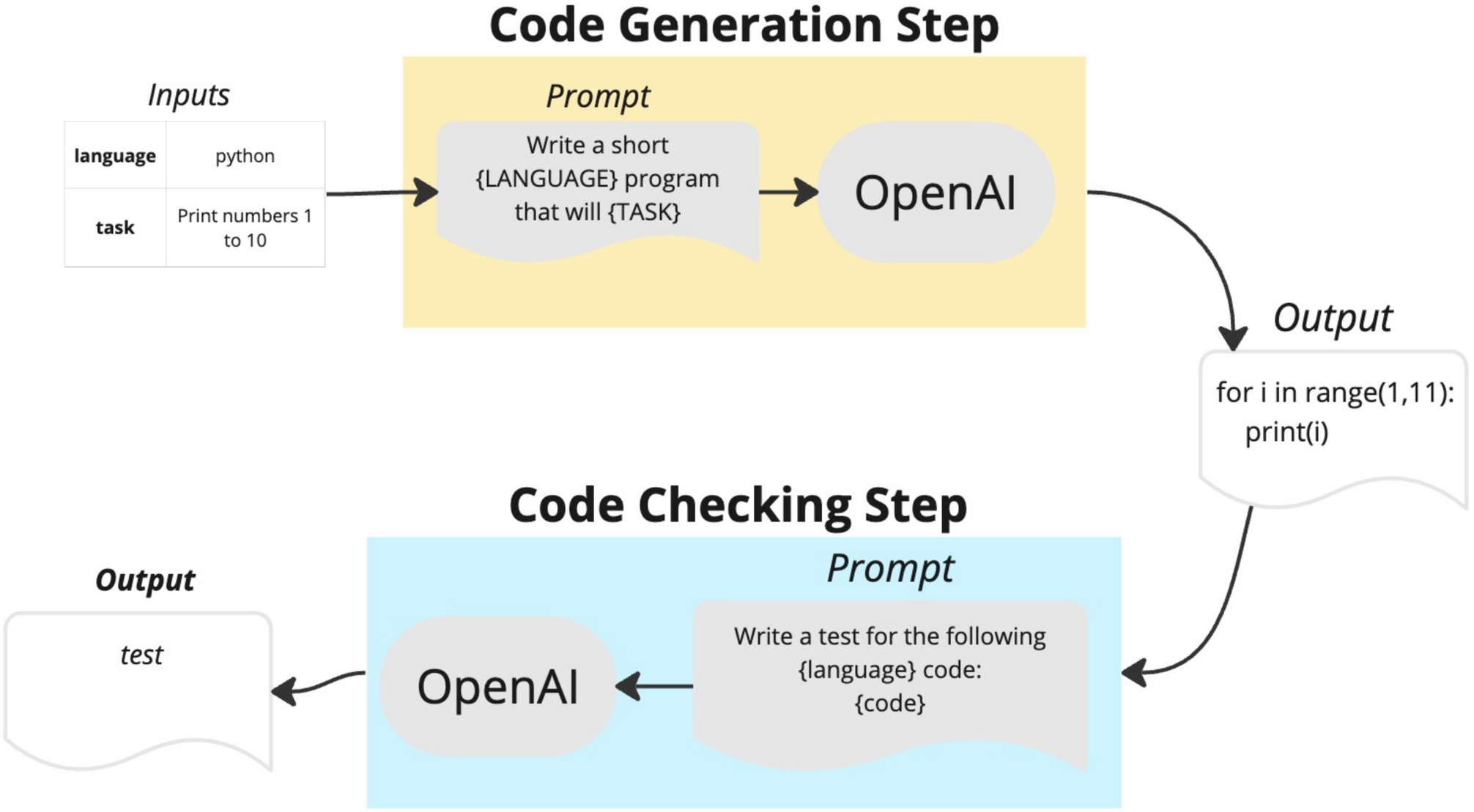


Different language models we might have access to

OpenAI

Llama


Terminal
> python main.py \
--task 'print numbers 1 to 10' \
--language python
Inputs
language
python
task
Print numbers 1 to 10
Prompt
Write a short {LANGUAGE} program that will {TASK}
OpenAI
Output called "code"
for i in range(1,11):
print(i)
Prompt
Make sure this snippet doesn't have any bugs in it: {CODE}
Bard
Output
The code looks good


Code Generation Step
Inputs
language
python
task
Print numbers 1 to 10
Prompt
Write a short {LANGUAGE} program that will {TASK}
OpenAI
Output
for i in range(1,11):
print(i)
Code Checking Step
Prompt
Write a test for the following {language} code:
{code}
OpenAI
Output
test

Two Big Goals of LangChain

Provide tools to automate each step of a text generation pipeline

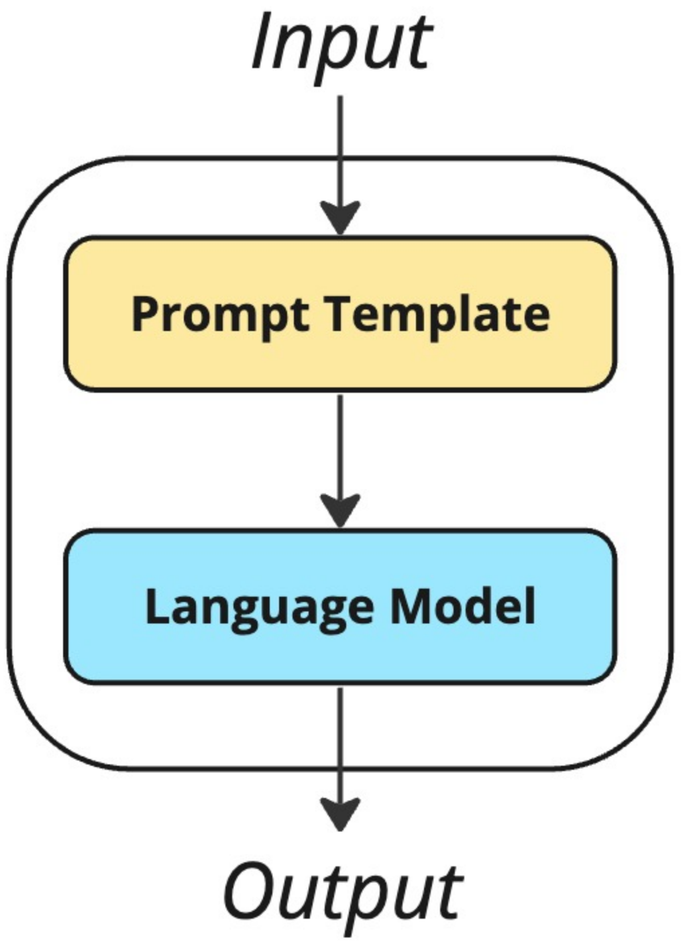
Make it easy to connect tools together

Python class provided by LangChain

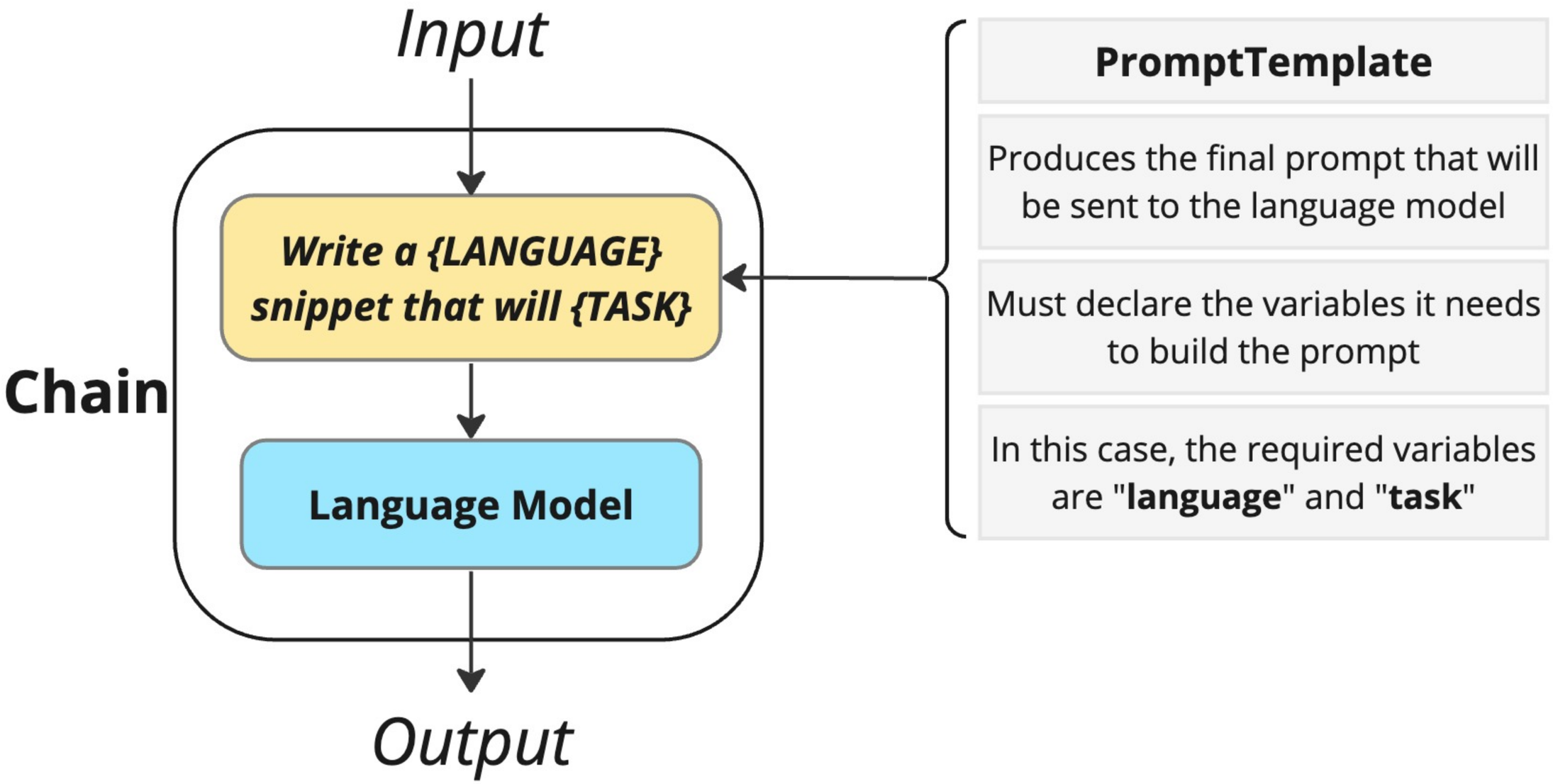
We use **Chains** to make reusable text-generation pipelines

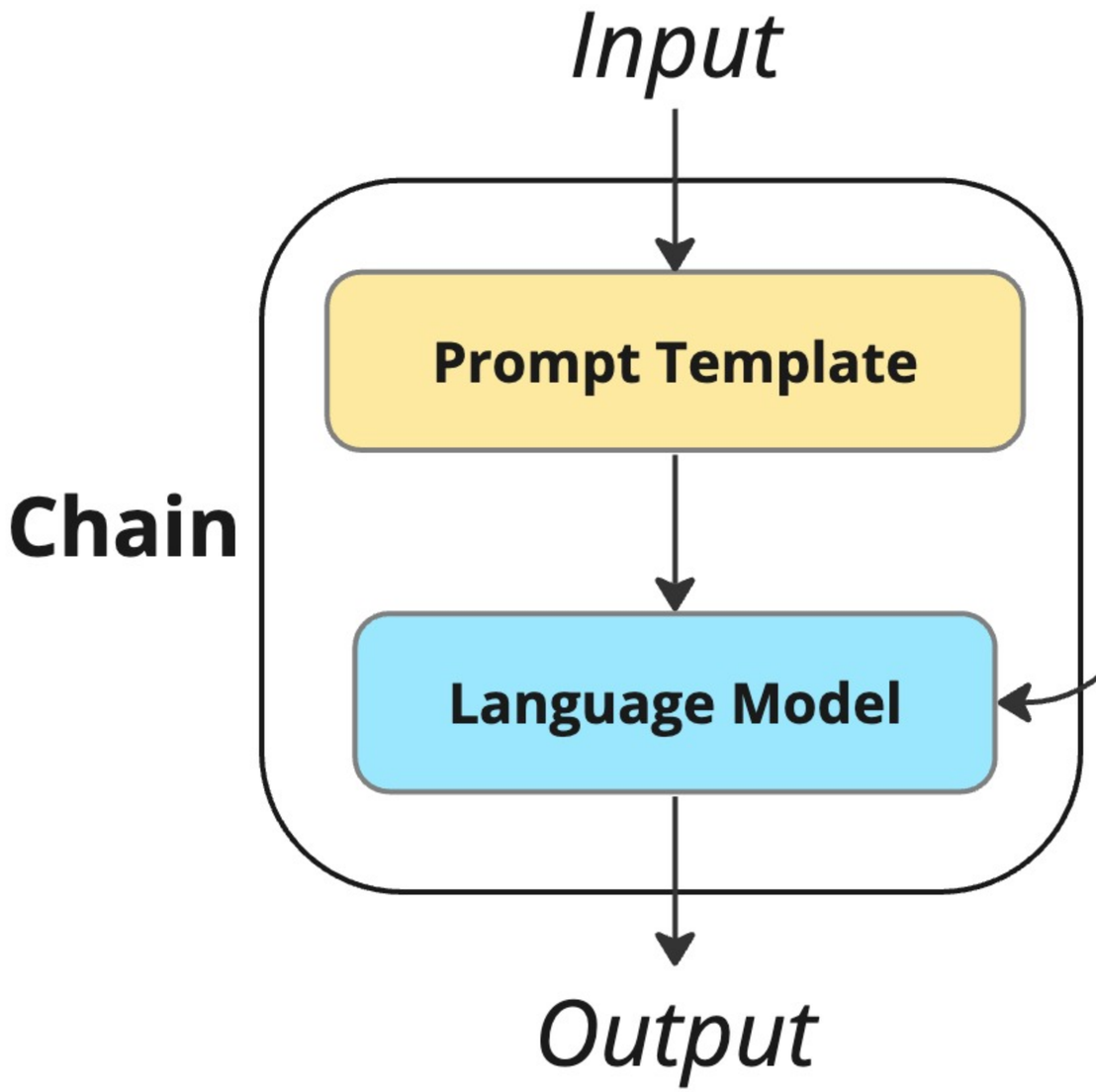
Chains can be connected together to make a more complex pipeline

A chain wraps up a *PromptTemplate* and an *LLM*



Chain

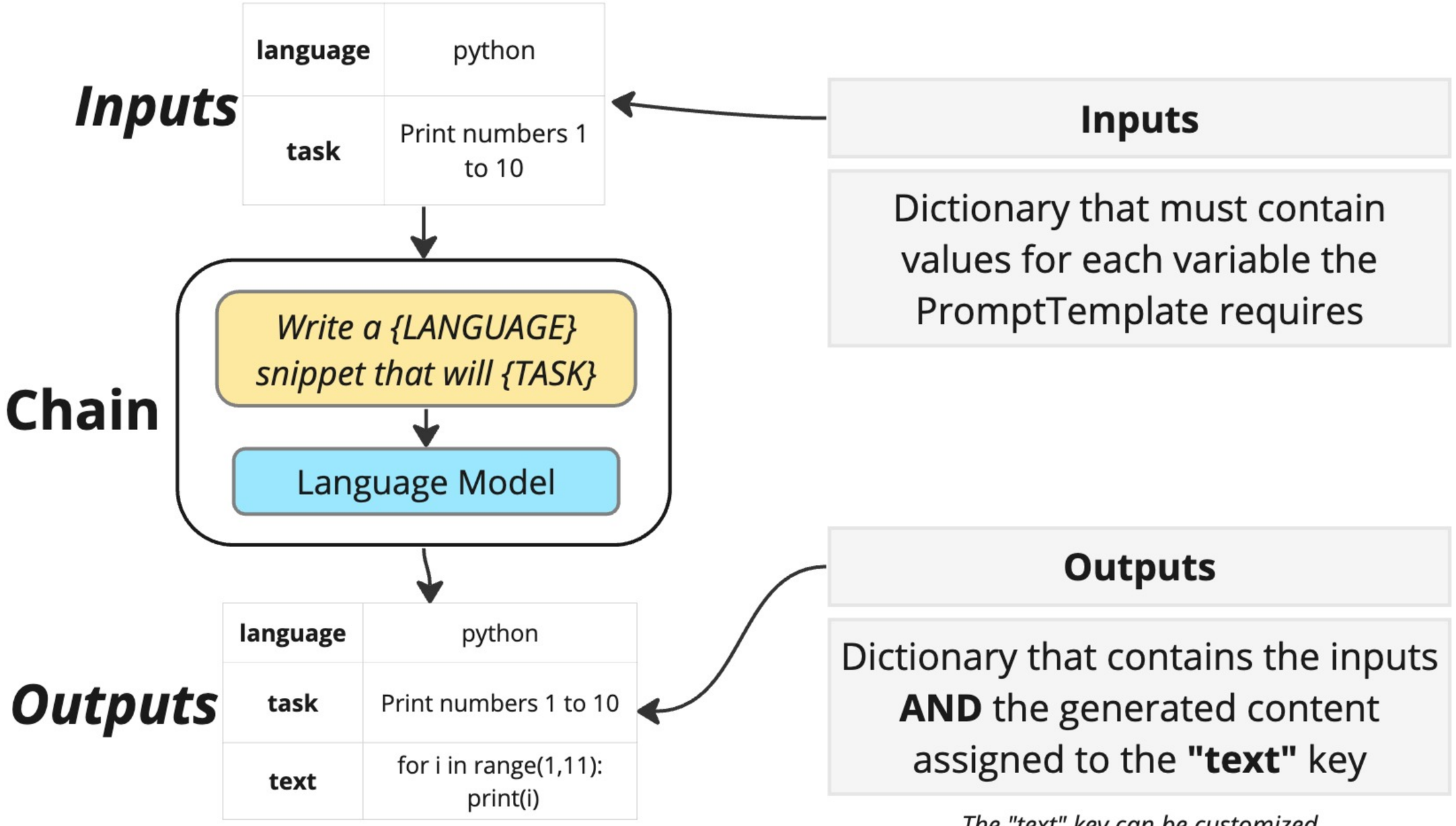

Input
Write a {LANGUAGE} snippet that will {TASK}
Chain
Language Model
Output
PromptTemplate
Produces the final prompt that will be sent to the language model
Must declare the variables it needs to build the prompt
In this case, the required variables are "language" and "task"

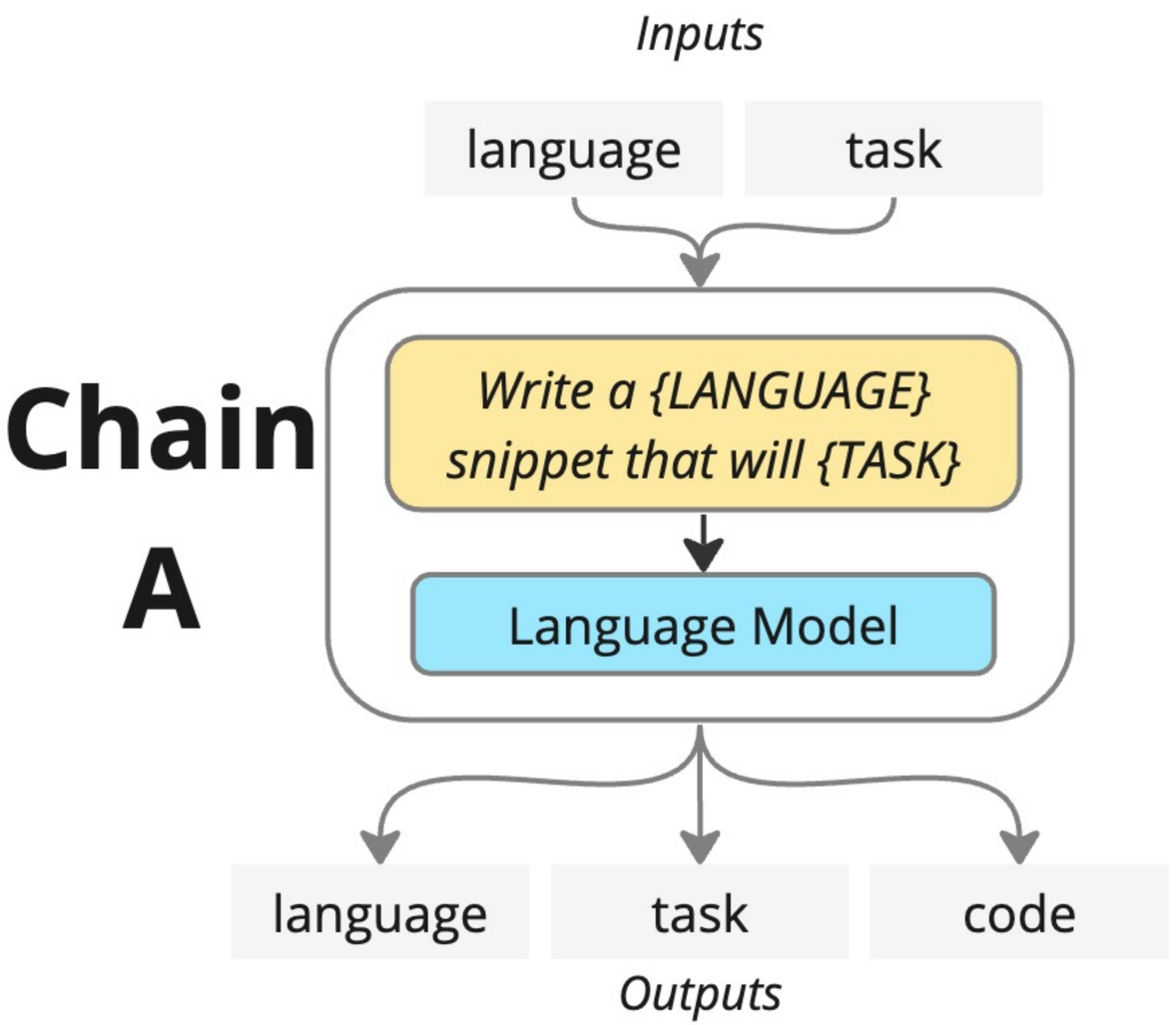


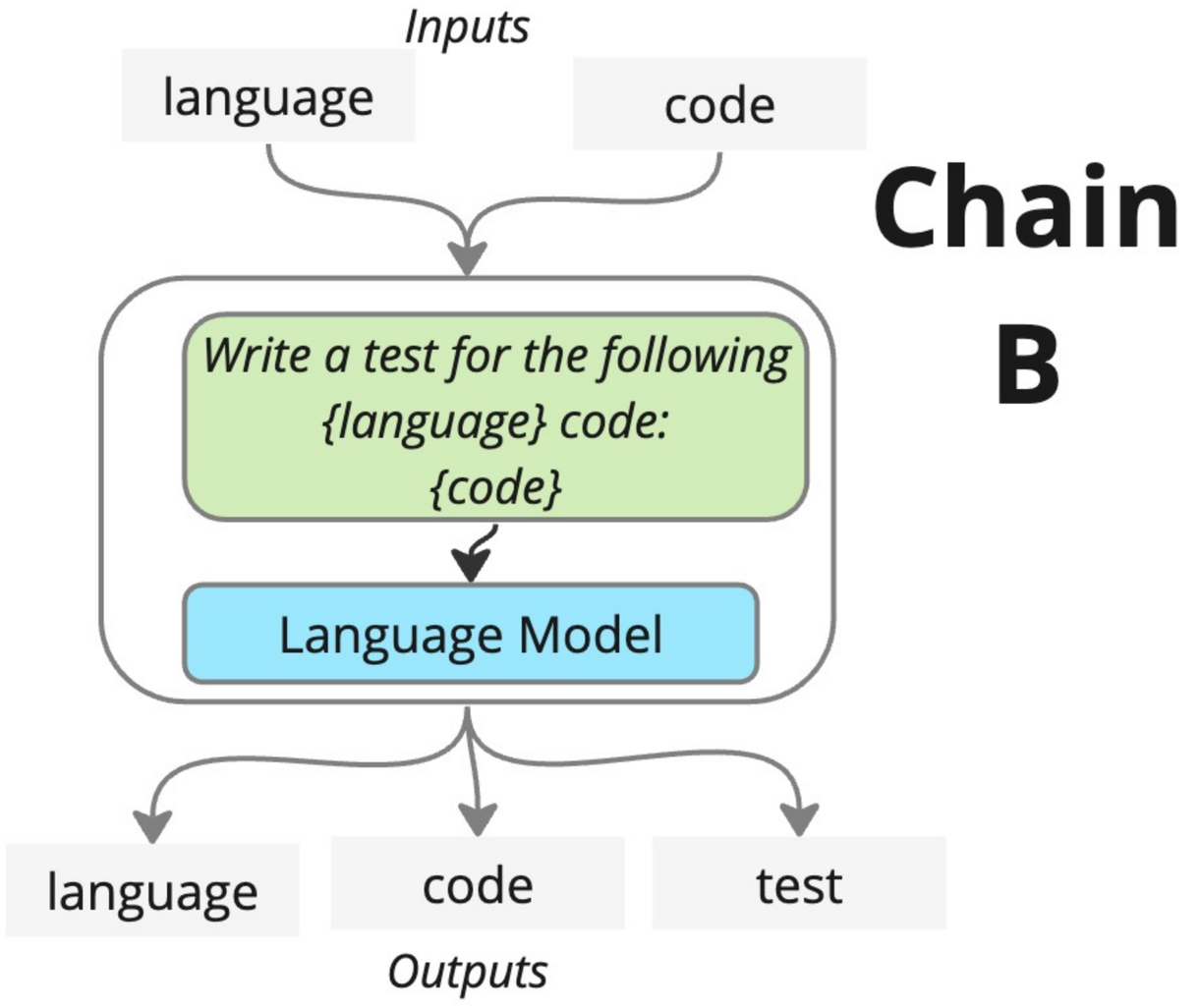
LanguageModel

The LLM we want to use for this text generation pipeline

Can be ChatGPT, Bard, or really anything that produces text


Inputs
language
python
task
Print numbers 1 to 10
Chain
Write a {LANGUAGE} snippet that will {TASK}
Language Model
Outputs
language
python
task
Print numbers 1 to 10
text
for i in range(1,11): print(i)
Inputs
Dictionary that must contain values for each variable the PromptTemplate requires
Outputs
Dictionary that contains the inputs AND the generated content assigned to the "text" key
The "text" key can be customized


Chain A
Inputs
language
task
Write a {LANGUAGE} snippet that will {TASK}
Language Model
language
task
code
Outputs


Chain B
Inputs
language
code
Write a test for the following {language} code: {code}
Language Model
language
code
test
Outputs