Tool Usage in Kani

Provide Plain Python Code to Models

- Tools are written as plain Python methods and decorated with the special `@ai_function` decorator

- In this example, we give a model access to a real-time weather API

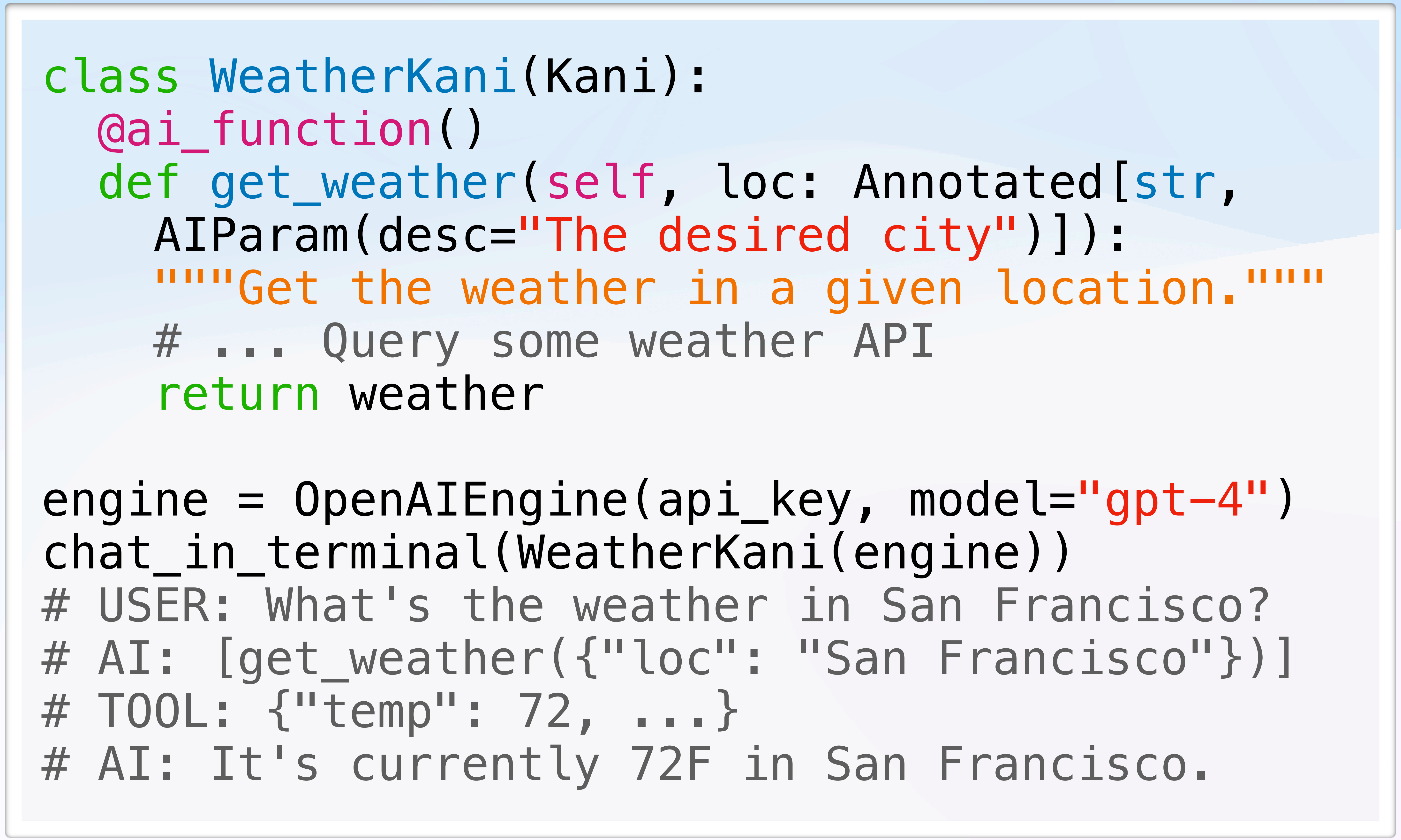
```
class WeatherKani(Kani):
  @ai_function()
  def get_weather(self, loc: Annotated[str,
    AIParam(desc="The desired city")]):
    """Get the weather in a given location."""
    # ... Query some weather API
    return weather

engine = OpenAIEngine(api_key, model="gpt-4")
chat_in_terminal(WeatherKani(engine))
# USER: What's the weather in San Francisco?
# AI: [get_weather({"loc": "San Francisco"})]
# TOOL: {"temp": 72, ...}
# AI: It's currently 72F in San Francisco.
```

15

Tool Usage in Kani

Provide Plain Python Code to Models

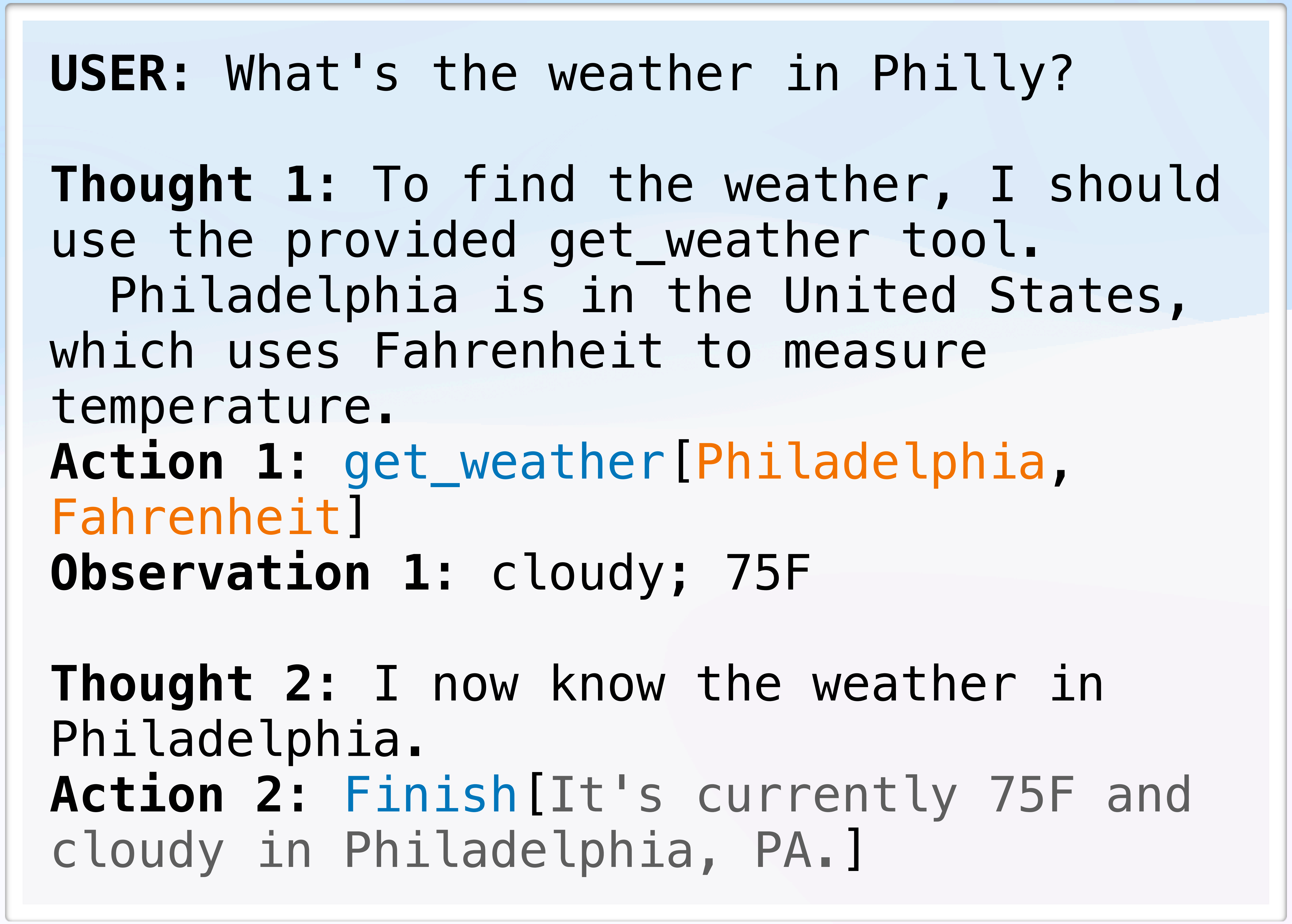
- Tools are written as plain Python methods and decorated with the special `@ai_function` decorator
- In this example, we give a model access to a real-time weather API



```
class WeatherKani(Kani):
  @ai_function()
  def get_weather(self, loc: Annotated[str,
    AIParam(desc="The desired city")]):
    """Get the weather in a given location."""
    # ... Query some weather API
    return weather

engine = OpenAIEngine(api_key, model="gpt-4")
chat_in_terminal(WeatherKani(engine))
# USER: What's the weather in San Francisco?
# AI: [get_weather({"loc": "San Francisco"})]
# TOOL: {"temp": 72, ...}
# AI: It's currently 72F in San Francisco.
```

Tool Prompting

ReAct et al



```
USER: What's the weather in Philly?

Thought 1: To find the weather, I should
use the provided get_weather tool.
  Philadelphia is in the United States,
which uses Fahrenheit to measure
temperature.
Action 1: get_weather[Philadelphia,
Fahrenheit]
Observation 1: cloudy; 75F

Thought 2: I now know the weather in
Philadelphia.
Action 2: Finish[It's currently 75F and
cloudy in Philadelphia, PA.]
```