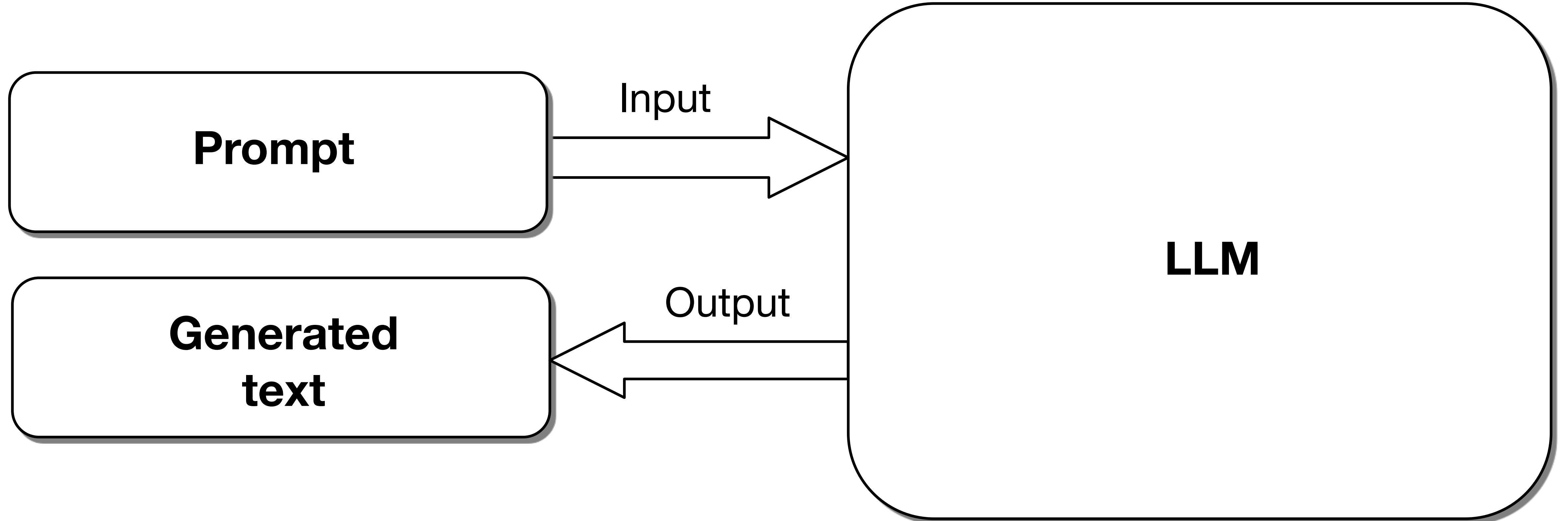


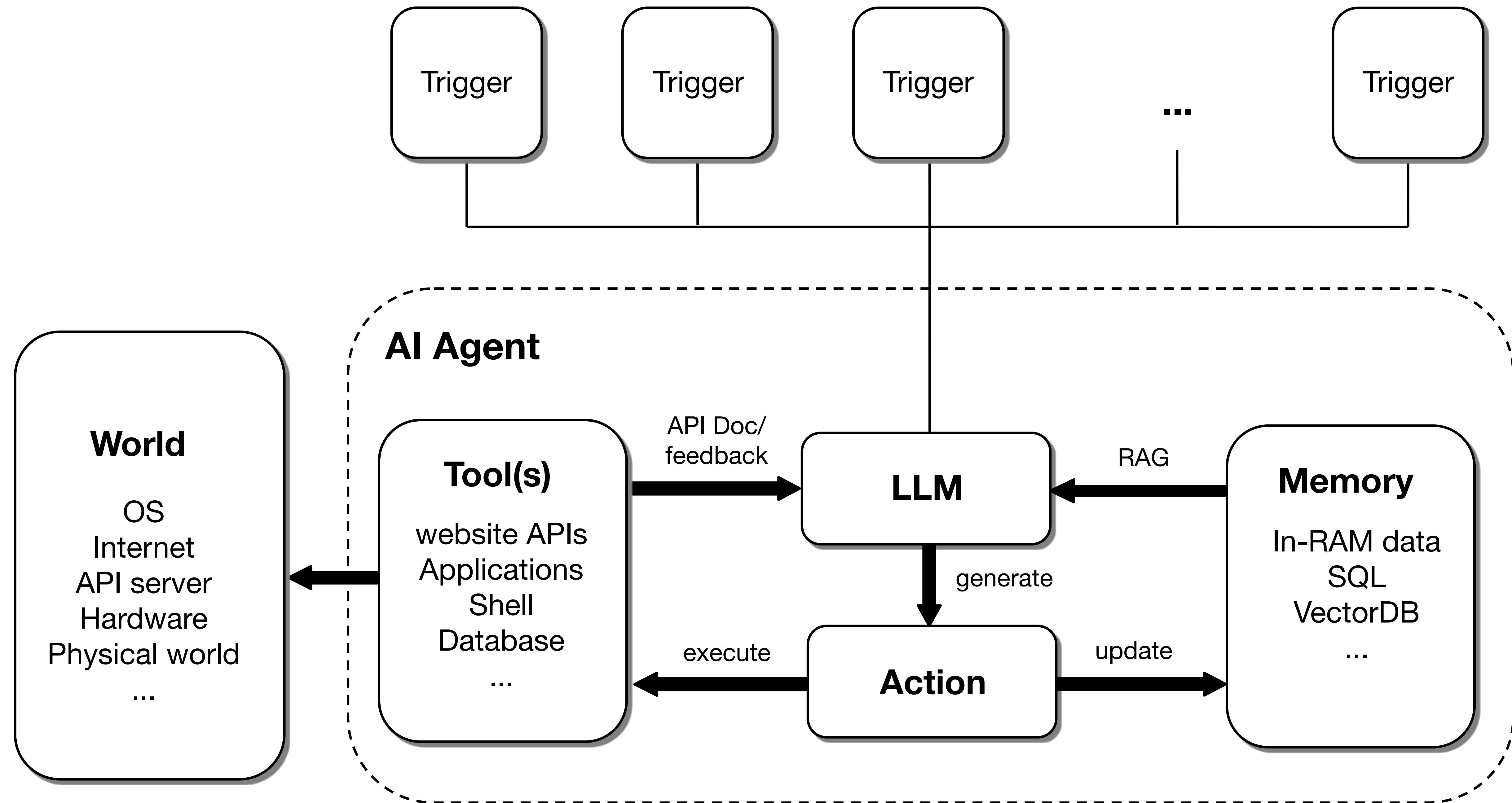
AI-агенты в каждый дом

Алексей Порядин, PythoNN

Основные понятия

- Что такое LLM?
- Что такое AI-agent?





LLM – собеседник, AI-агент – помощник

Example of Project's `urls.py` Configuration

Here's a complete example including the `vulnerabilities` app:

```
python

# project/urls.py

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('forums/', include('forums.urls', namespace='forums')),
    path('vulnerabilities/', include('vulnerabilities.urls', namespace='vulnerabilities')),
    # Add other apps here
]
```

By ensuring that your URL patterns, namespace, and project configuration are correct, you should be able to resolve the `NoReverseMatch` error.



You've hit the Free plan monthly limit for GPT-4o.

Responses will use our basic model until your limit resets August 14, 2024.

Copy code

Get Plus

X

Message ChatGPT

```
1 import Image fr
2 import Link fro
3 import { Descri
4
5 interface Projec
6     title: string
7     description: string
8     images: string
9     link: string;
10 }
11
12 const Project:
13     title,
14     description,
15     images,
16     link,
17 }) => {
18
19     return (
20         <div className="Project">
21             <Descript...
```

What are my open issues?

GitHub Copilot

Let me check for any open issues in your repository.

> Ran list_issues

You have 1 open issues in your repository:

- Issue #4: "Create detailed flow for adding new journal entries to the app"
 - Created on March 21, 2025
 - Assigned to you
 - Description: Create a detailed user flow exploring all the options for adding a new journal entry.

8 Add Context...

Ask Copilot or type / for commands



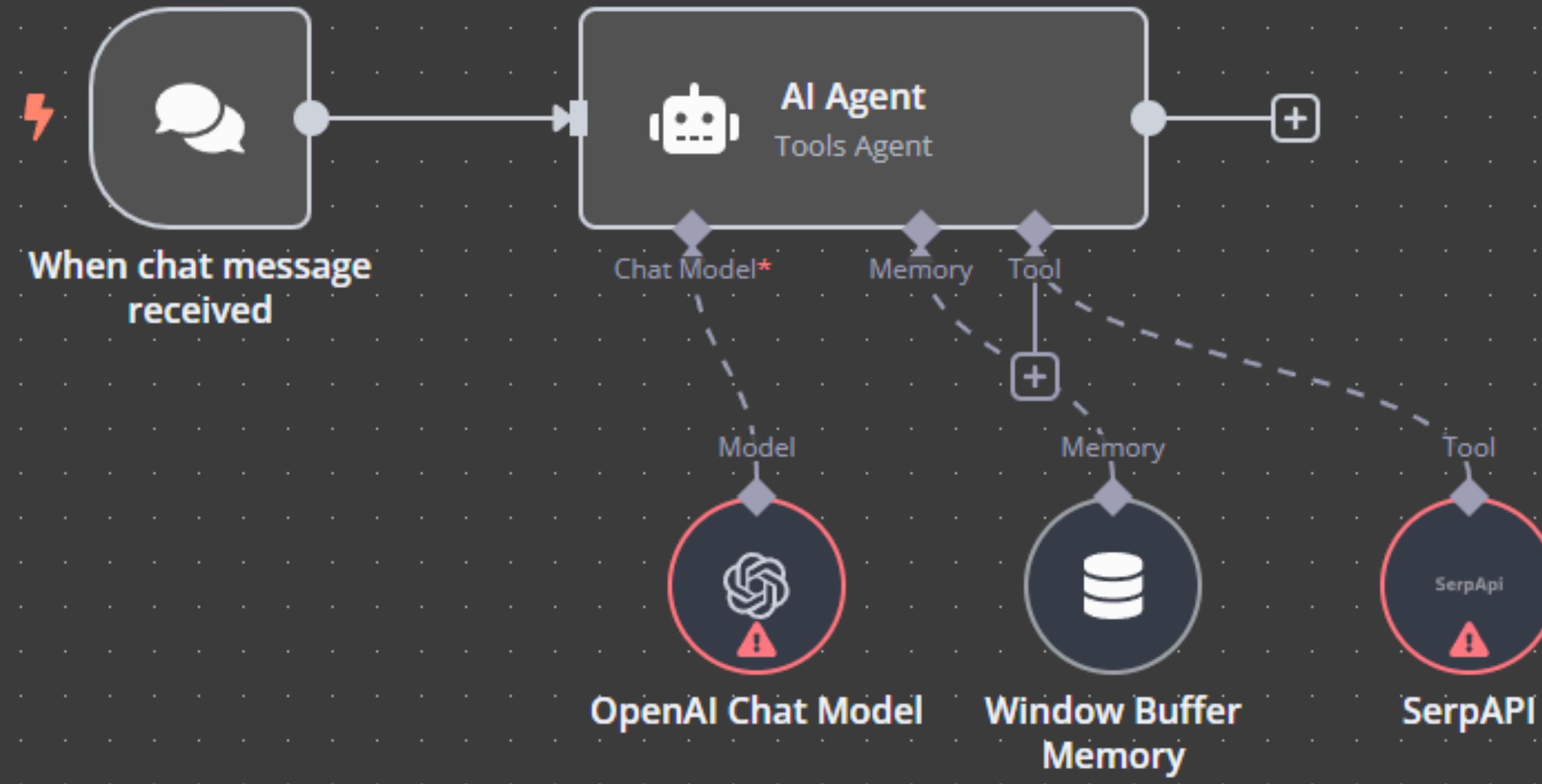
Agent ▾ Claude 3.7 Sonnet ▾ ➤ ▾

Какие есть инструменты?

No Code



n8n



Любимый Python

```
messages = [{}  
    "role": "system",  
    "content": "Ты — помощник, отвечай развернуто и вежливо."  
]
```

```
def send_message(user_input):  
    messages.append({"role": "user", "content": user_input})
```

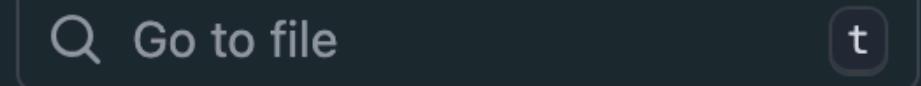
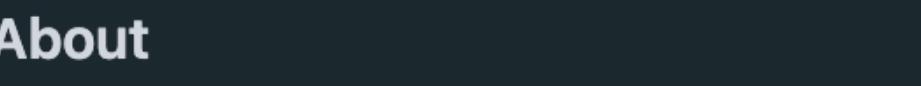
```
completion = openai.chat.completions.create(  
    model="gpt-4o-mini",  
    messages=messages  
)
```

```
assistant_reply = completion.choices[0].message.content  
  
messages.append({"role": "assistant", "content": assistant_reply})  
return assistant_reply
```

```
print("Assistant:", send_message("Привет! Я Python разработчик"))
```

 **langchain** Public

 Watch 774  Fork 19k  Starred 115.7k

 master  988 Branches  979 Tags  Go to file  Add file  Code  About

 **sydney-runkle** and **eyurtsev** feat(langchain): dynamic system prompt ...  4d11877 · 12 hours ago  14,252 Commits

 Build context-aware reasoning applications

LangChain

- Цепочки вызовов (Chains)**
- Шаблоны промптов и форматирование**
- Интеграция инструментов**
- Стандарт в разработке агентов**

```
workflow = StateGraph(state_schema=MessagesState)
```

```
def call_model(state: MessagesState):
    response = llm.invoke(state["messages"])
    return {"messages": response}
```

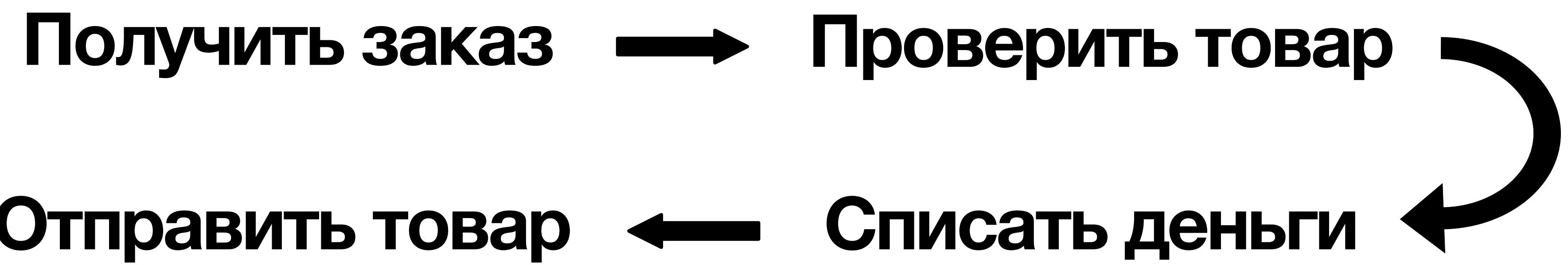
```
workflow.add_edge(START, "model")
workflow.add_node("model", call_model)
```

```
memory = MemorySaver()
app = workflow.compile(checkpointer=memory)
```

```
config = {"configurable": {"thread_id": "bob-conversation-1"} }
```

```
output = app.invoke({"messages": [HumanMessage("Привет! Я Python разработчик")]}, config)
output["messages"][-1].pretty_print()
```

```
output = app.invoke({"messages": [HumanMessage("Кто я такой?")]}, config)
output["messages"][-1].pretty_print()
```



 **langgraph** Public

 Watch 115  Fork 3.3k  Starred 18.8k

 main ▾  150 Branches  411 Tags

 Go to file  t  Add file ▾  Code ▾

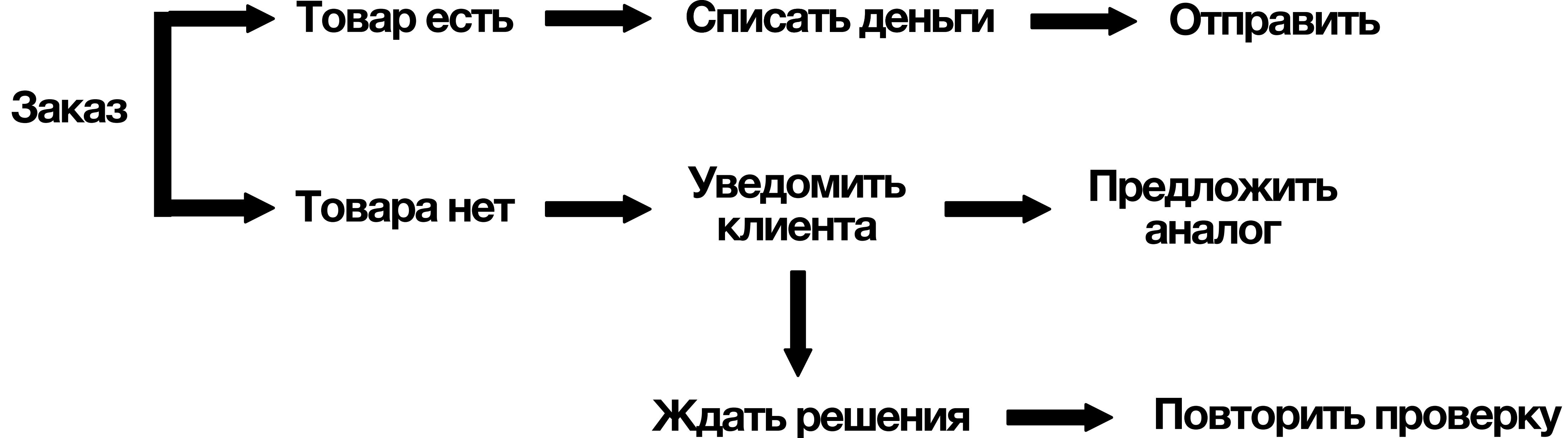
About

 **casparb** and **shahrukh-shaik** fix(langgraph): fix PostgresSaver crashing w...  f0fcfd2 · yesterday  6,279 Commits

Build resilient language agents as graphs.

LangGraph

- Новый уровень: агент как граф состояний
- Каждый узел = шаг (LLM, API, проверка условий)
- Поддержка циклов и ветвлений
- Больше контроля, меньше галлюцинаций



```
from langgraph.graph import StateGraph
```

```
class State(dict):  
    pass
```

```
workflow = StateGraph(State)
```

```
def hello_node(state: State) -> State:  
    return {"message": "Hello, LangGraph!"}
```

```
workflow.add_node("hello", hello_node)  
workflow.set_entry_point("hello")
```

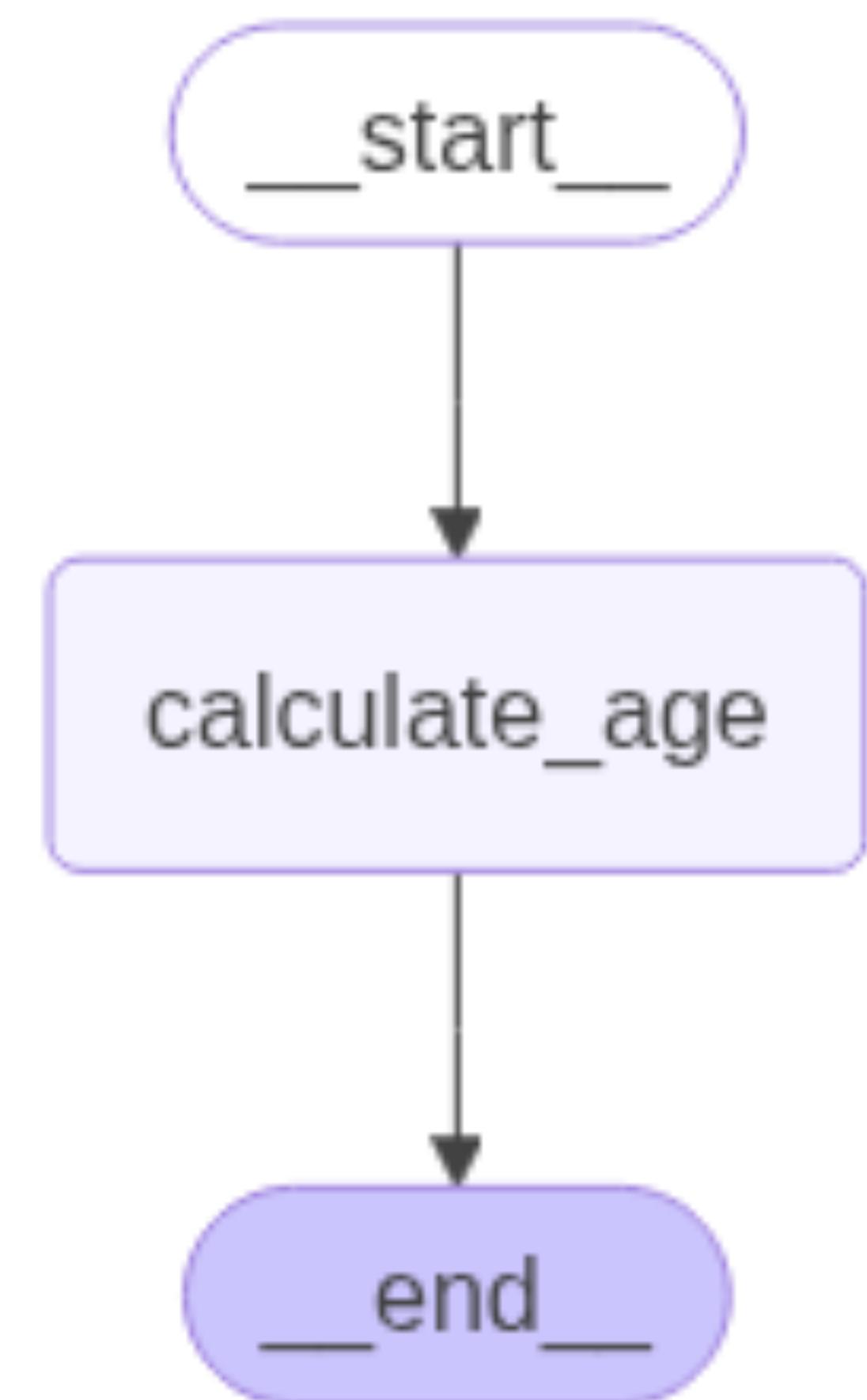
```
from langchain_openai import ChatOpenAI
```

```
model = ChatOpenAI(model="gpt-4o-mini")
```

```
def llm_node(state: State) -> State:  
    resp = model.invoke(state["input"])  
    return {"llm_output": resp.content}
```

```
workflow.add_node("llm", llm_node)  
workflow.set_entry_point("llm")
```

```
app = workflow.compile()  
print(app.invoke({"input": "Привет, кто ты?"}))
```



```
def search_tool(state: State) -> State:  
    q = state["llm_output"]  
    return {"search_result": f"Результаты поиска для: {q}"}
```

```
workflow.add_node("search", search_tool)
```

```
workflow.add_edge("llm", "search")
```

```
app = workflow.compile()
```

```
result = app.invoke({"input": "Что такое LangGraph?"})  
print(result)
```

```
# Пример вывода:  
# {  
#   'llm_output': 'LangGraph — это фреймворк...',  
#   'search_result': 'Результаты поиска для: LangGraph — это фреймворк...'  
# }
```

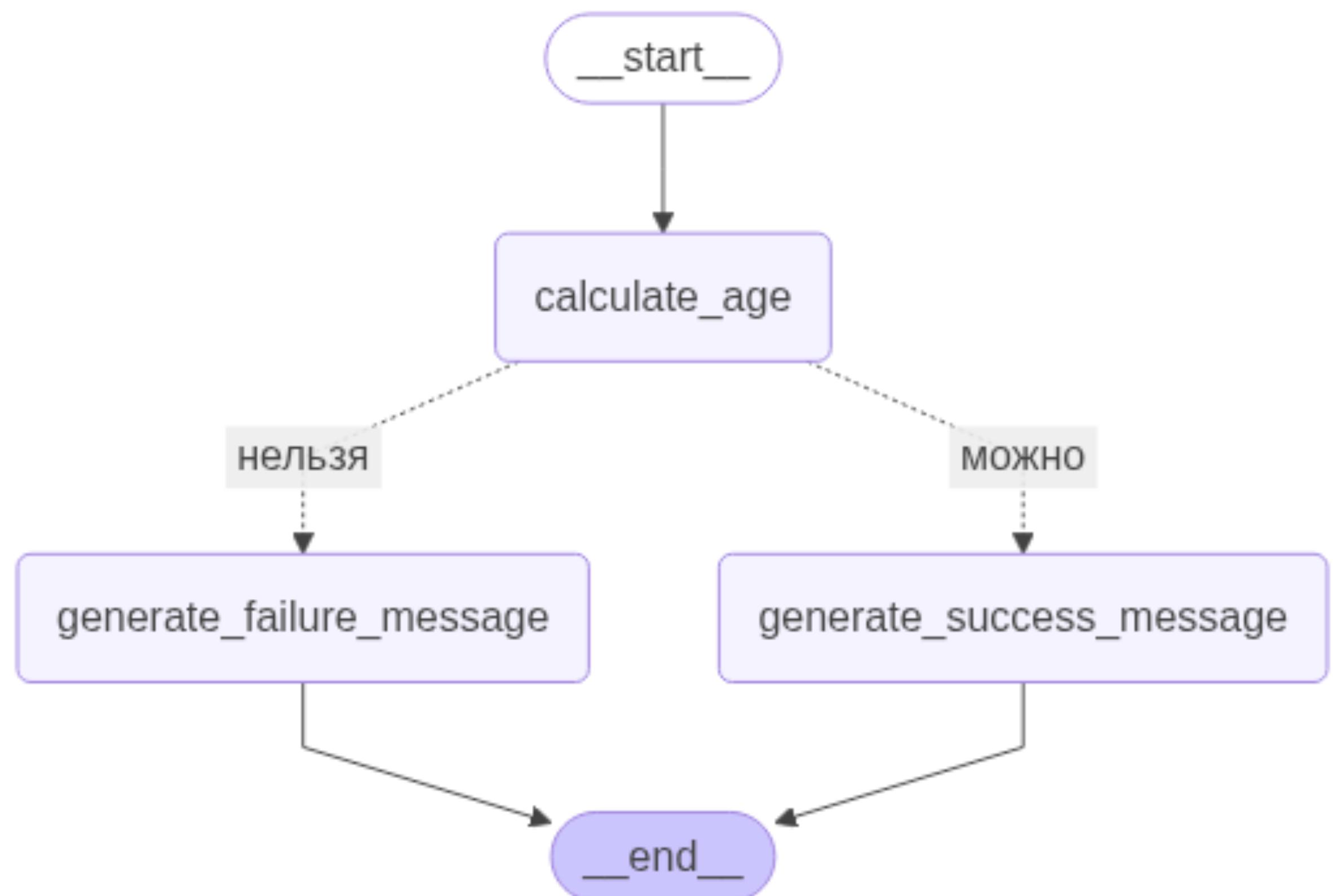
```
from langgraph import Graph
from langgraph.nodes import ToolNode, LLMNode

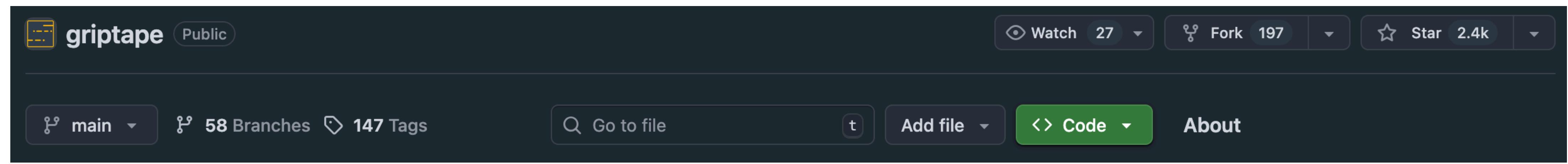
search_node = ToolNode(
    name="web_search",
    tool="SerpAPI", # абстракция для поиска
    input_key="query",
    output_key="results"
)

llm_node = LLMNode(
    name="pick_joke",
    prompt_template="Возьми из {{results}} одну короткую шутку."
)

graph = Graph()
graph.add_nodes([search_node, llm_node])
graph.connect(search_node, llm_node)

graph.run({"query": "programming one-liner joke"})
```





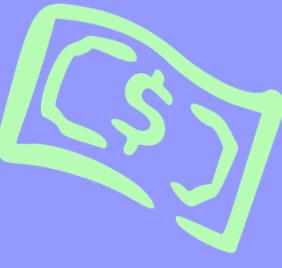
Griptape

- Поддерживает DAG, а также последовательные конвейеры и автономные агенты.
- Есть building blocks для инструментов и работы с данными. Это расширяет творческие возможности LLM, но накладывает строгие ограничения на то, какие данные он может видеть напрямую.
- Расширяет творческие возможности LLM, но накладывает строгие ограничения на то, какие данные он может видеть напрямую.

Где взять токены?

- OpenAI
- Antropic
- OpenRouter
- GigaChat

MONTHLY COST BREAKDOWN



HOURLY RATES

Human Agents
\$18/HR

AI Agents
\$12/HR

HUMAN TALK TIME PER MONTH

60-75%
of total hours

= 103-130 talk
time hours per
month

AI TALK TIME PER MONTH

100%
of total hours

= 173 talk time
hours per
month

TOTAL COST PER 173 HOURS OF TALK TIME

Human Agents:
~\$3100

AI Agents:
~\$2100

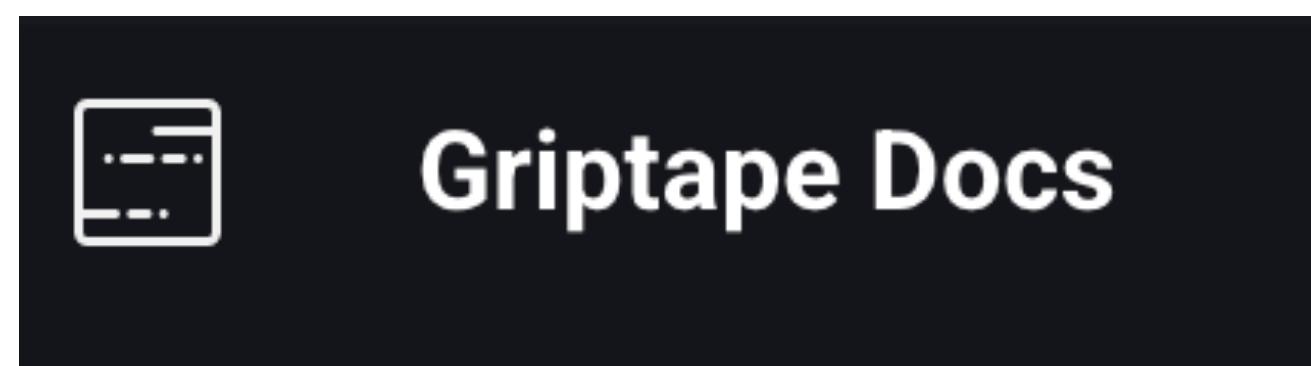
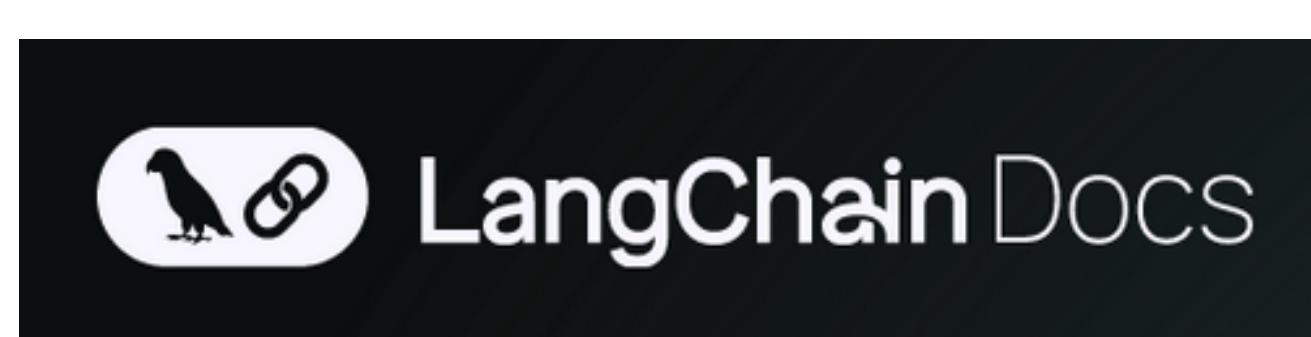
Где не надо применять?

- Логика простая, контекст минимален
- Персональные данные
(Иключение: локальные модели)
- Нужна 100% точность
- Люди дешевле и лучше

AI Agents IN ACTION

Micheal Lanham

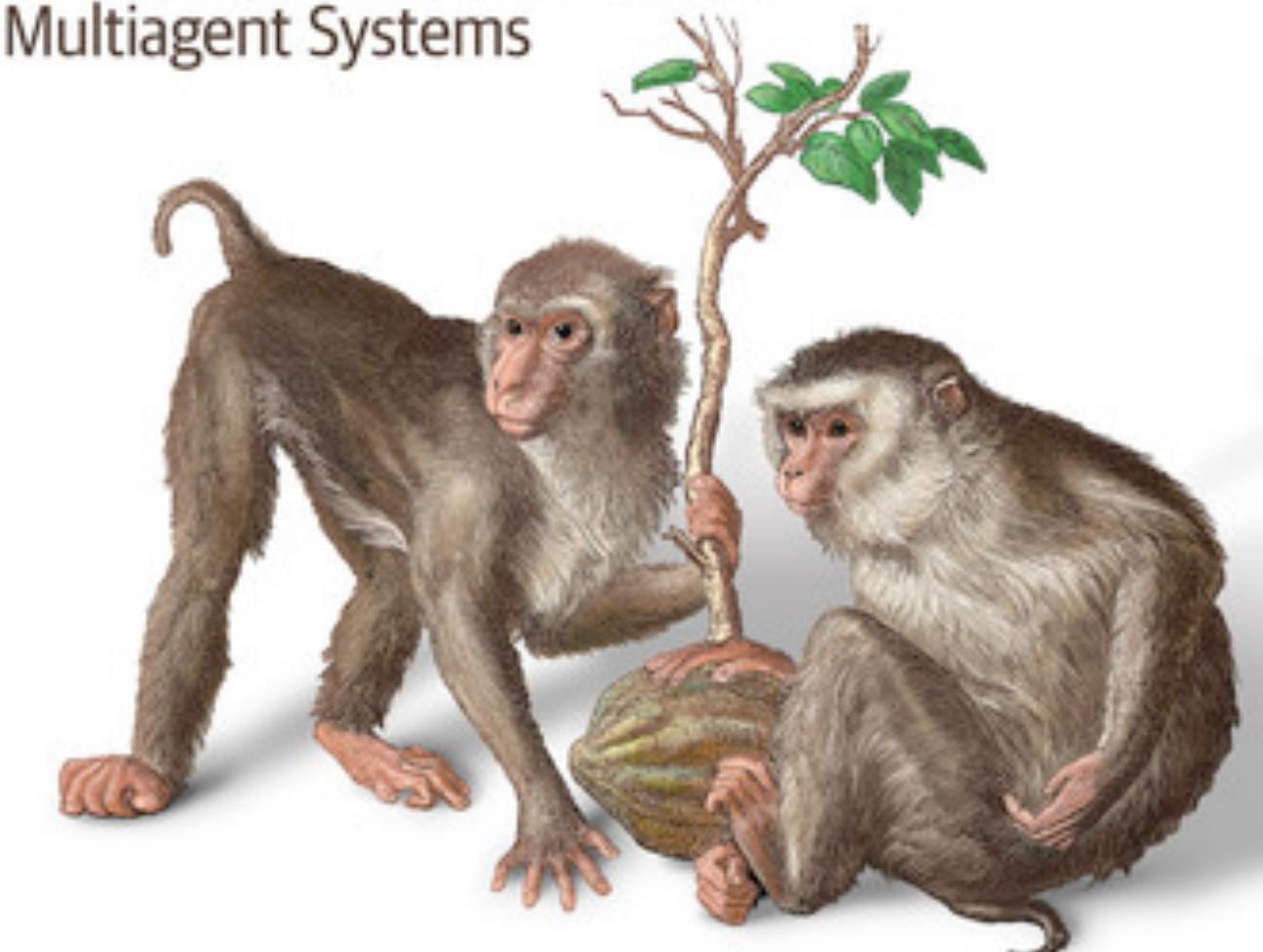
MANNING



O'REILLY®

Building Applications with AI Agents

Designing and Implementing
Multiagent Systems



Michael Albada

**Создавайте агентов
и зарабатывайте миллионы**

