ReAct 智能应用架构深度剖析

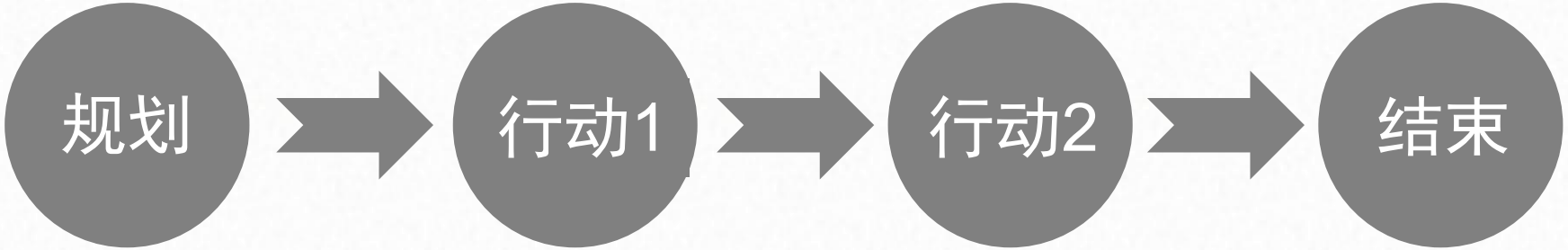
目录

1 Agent 流行的工作方式

01、AI Agent 智能体应用架构深度剖析及关键技术详解

我们希望 Agent 如何工作?

方案一:



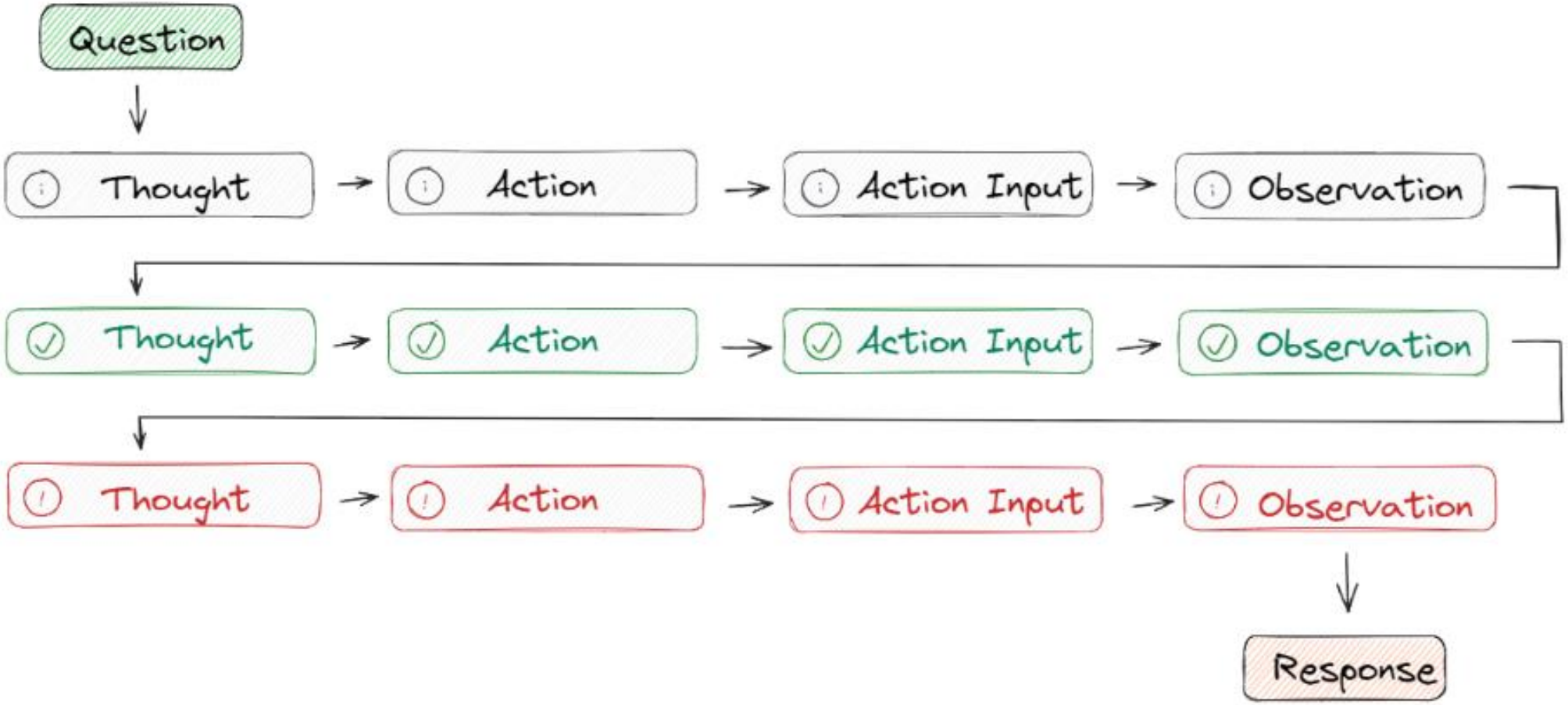
分解优先

方案二:

交错分解

2 手动实现 ReAct 智能应用架构

01、手动实现 ReAct 智能应用架构



02、ReAct 提示词设计-英文

```
prompt = """
You run in a loop of Thought, Action, Observation, Answer.
At the end of the loop you output an Answer
Use Thought to describe your thoughts about the question you have been asked.
Use Action to run one of the actions available to you.
Observation will be the result of running those actions.
Answer will be the result of analysing the Observation

Your available actions are:

calculate:
e.g. calculate: 4 * 7 / 3
Runs a calculation and returns the number - uses Python so be sure to use floating point syntax if necessary

wikipedia:
e.g. wikipedia: Django
Returns a summary from searching Wikipedia

Always look things up on Wikipedia if you have the opportunity to do so.

Example session:

Question: What is the capital of France?

Thought: I should look up France on Wikipedia

Action: wikipedia: France

You should then call the appropriate action and determine the answer from
the result

You then output:

Answer: The capital of France is Paris
"""
```

03、ReaAct 提示词设计-中文

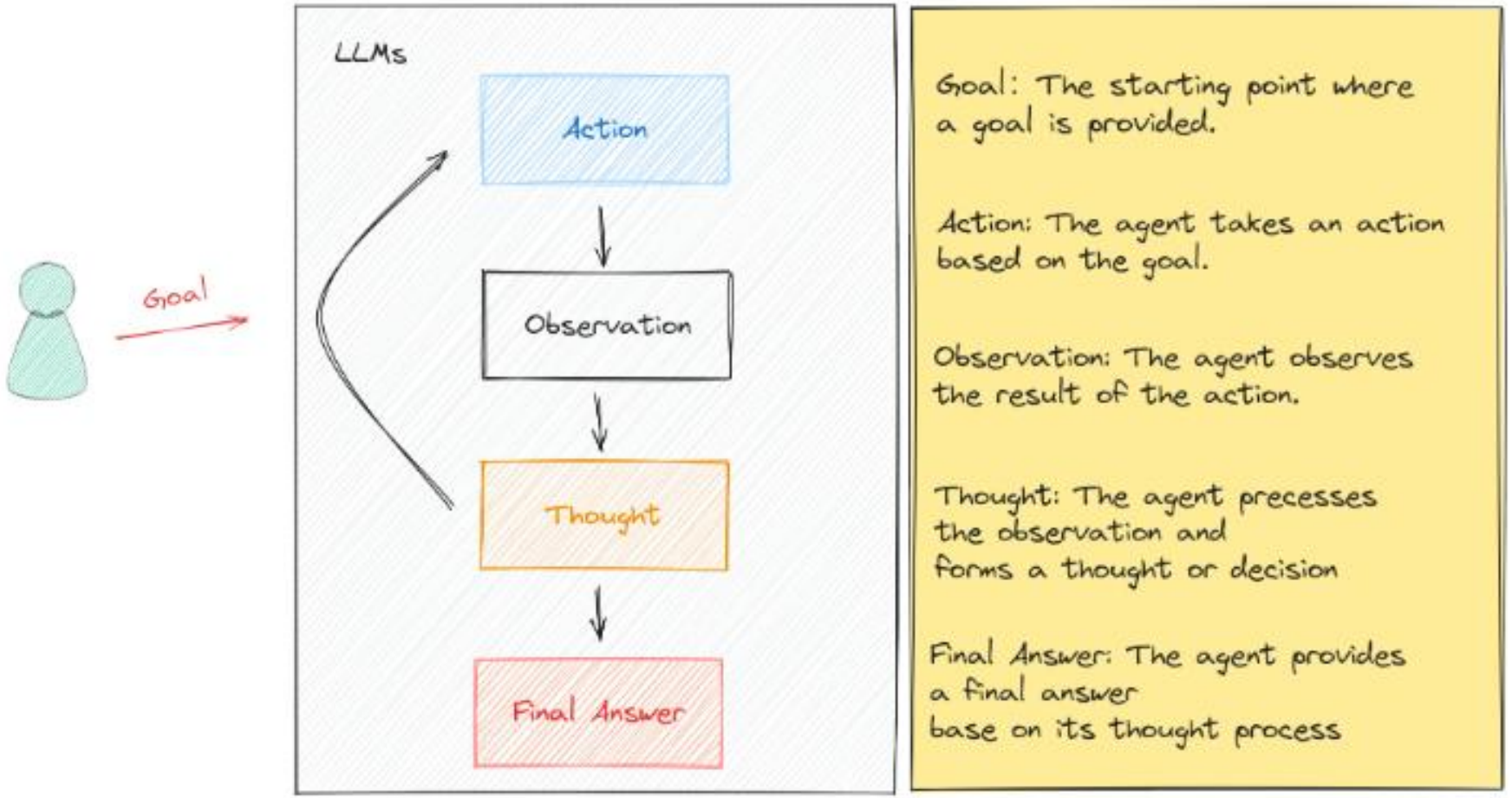
```
prompt = """

您在一个由“思考、行动、观察、回答”组成的循环中运行。
在循环的最后，您输出一个答案。
使用“思考”来描述您对所提问题的思考。
使用“行动”来执行您可用的动作之一。
“观察”将是执行这些动作的结果。
“回答”将是分析“观察”结果后得出的答案。

您可用的动作包括：

calculate（计算）：
例如：calculate: 4 * 7 / 3
执行计算并返回数字 - 使用Python，如有必要请确保使用浮点数语法

wikipedia（维基百科）：
例如：wikipedia: Django
返回从维基百科搜索的摘要

如果有机会，请始终在维基百科上查找信息。

示例会话：

问题：法国的首都是什么？

思考：我应该在维基百科上查找关于法国的信息

行动：wikipedia: France

然后您应该调用适当的动作，并从结果中确定答案

您然后输出：

回答：法国的首都是巴黎

"""
```

指定角色
干活的方式指定

指定工具

给出示例

03、ReAct 应用架构设计



3 基于 ReAct 智能应用架构的客服系统

03、ReAct 应用架构设计