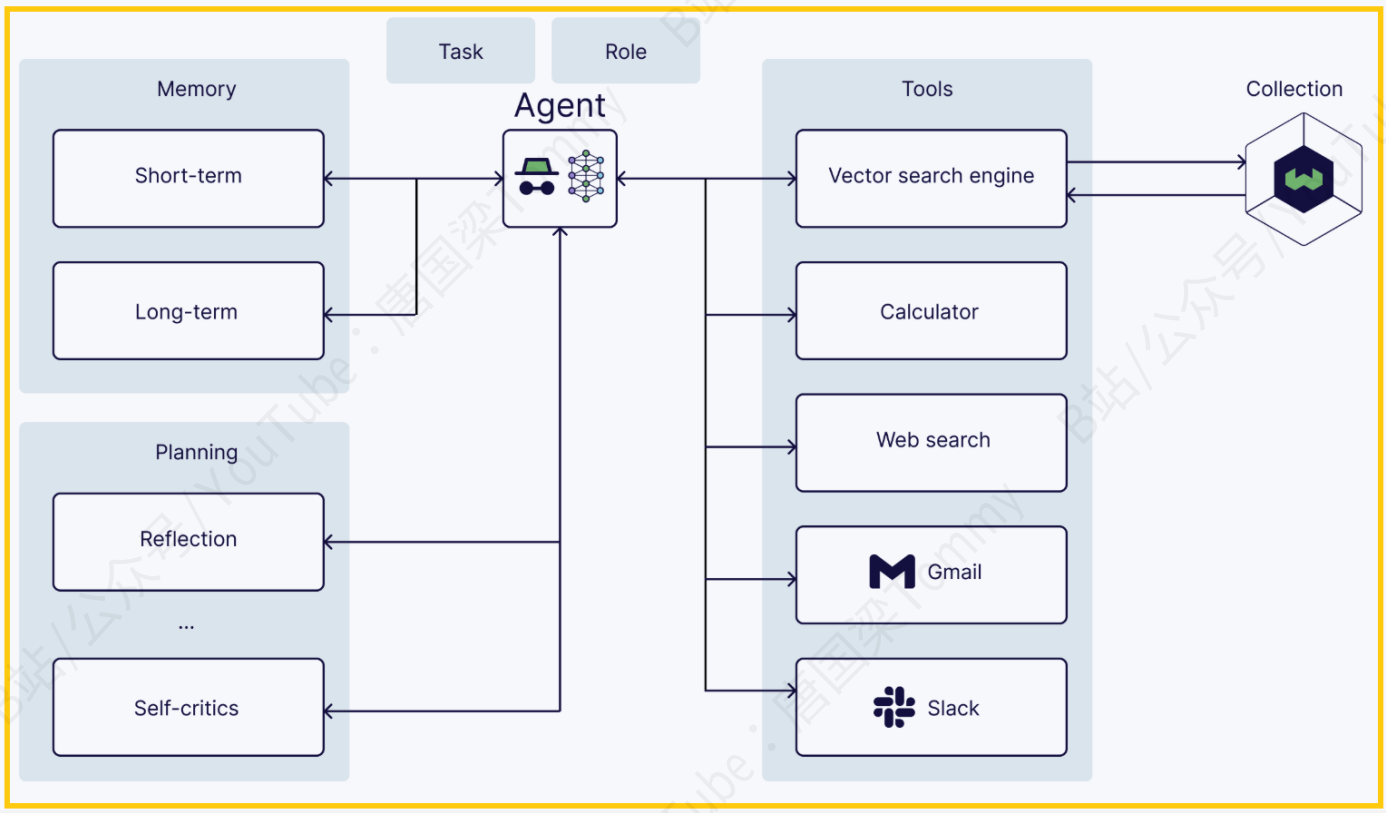
- **规划（反思和自我批评）**：通过反思、查询路由或自我批评来指导智能体的迭代推理过程，确保复杂任务被有效分解。
- **工具（向量搜索、网络搜索、API 等）**：扩展智能体的能力，使其能够访问外部资源、实时数据或进行专业计算。



5. Agent 设计模式

```
# 参考资源
1. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG
https://arxiv.org/pdf/2501.09136v3
```

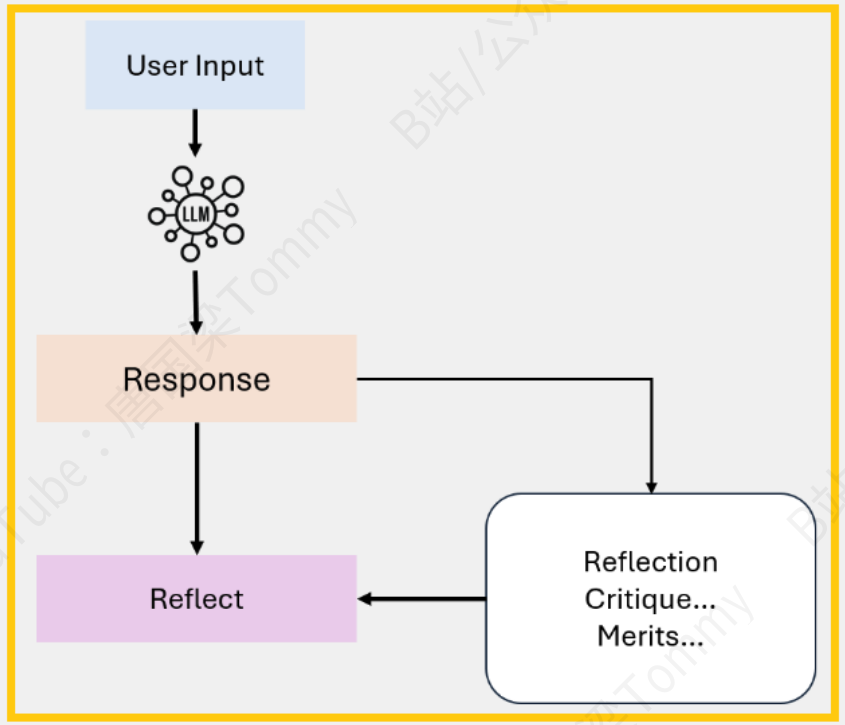
智能体设计模式为构建具备自主性、适应性和协作能力的智能系统提供了结构化的方法。这些模式使智能体能够在复杂、多变的环境中高效运行，提升系统的准确性和可扩展性。本文重点介绍四种关键模式：

1️⃣ 反思 (Reflection)

反思模式使智能体能够自我评估并持续优化其输出。通过引入自我反馈机制，智能体可以识别并修正错误、不一致或低效的部分，从而提高整体性能。在实践中，反思通常包括以下步骤：

- **自我评估**：智能体对其输出进行分析，判断其准确性、风格和效率。
- **反馈整合**：将评估结果纳入后续迭代，优化输出质量。
- **外部辅助**：借助单元测试、网络搜索等工具增强评估过程。

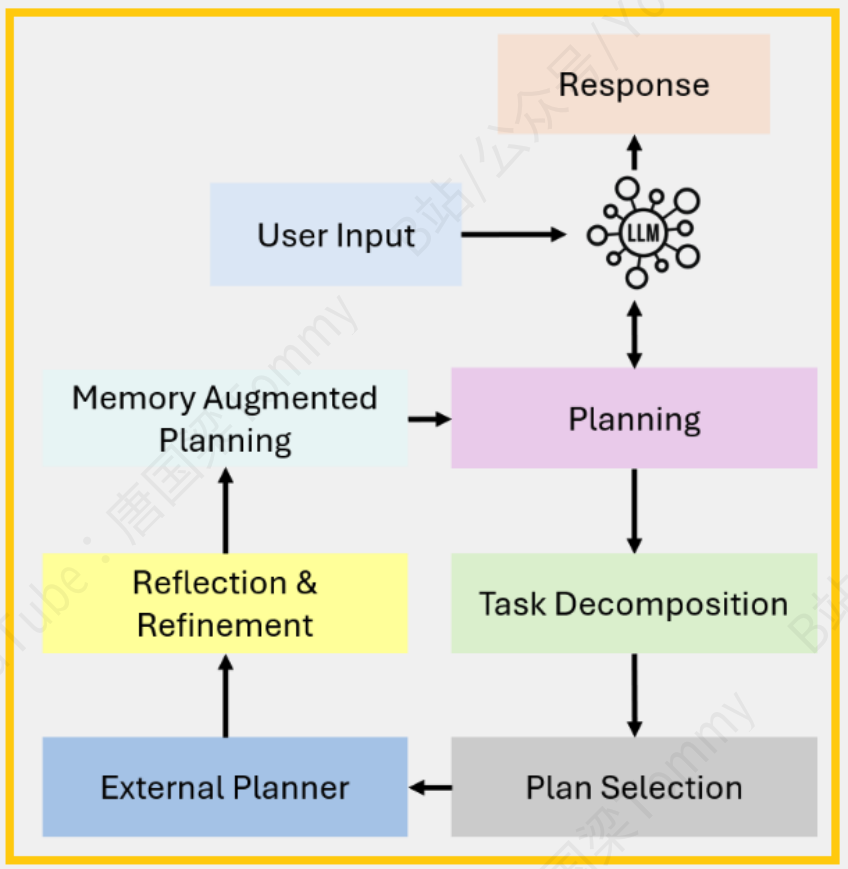
在多智能体系统中，反思可以通过角色分工实现，例如一个智能体生成输出，另一个进行审查和反馈。研究表明，反思机制能够显著提升智能体的表现，尤其在复杂任务中表现尤为突出。



2️⃣ 规划 (Planning)

规划模式使智能体能够将复杂任务分解为更小、可管理的子任务，并制定执行策略。这对于需要多步推理或在动态环境中操作的任务尤为重要。关键特性包括：

- **任务分解**：将整体目标拆分为具体步骤，明确每一步的目标和方法。
- **动态调整**：根据环境变化实时更新计划，确保任务的顺利完成。
- **协同执行**：在多智能体系统中，各智能体根据规划协同工作，提高效率。

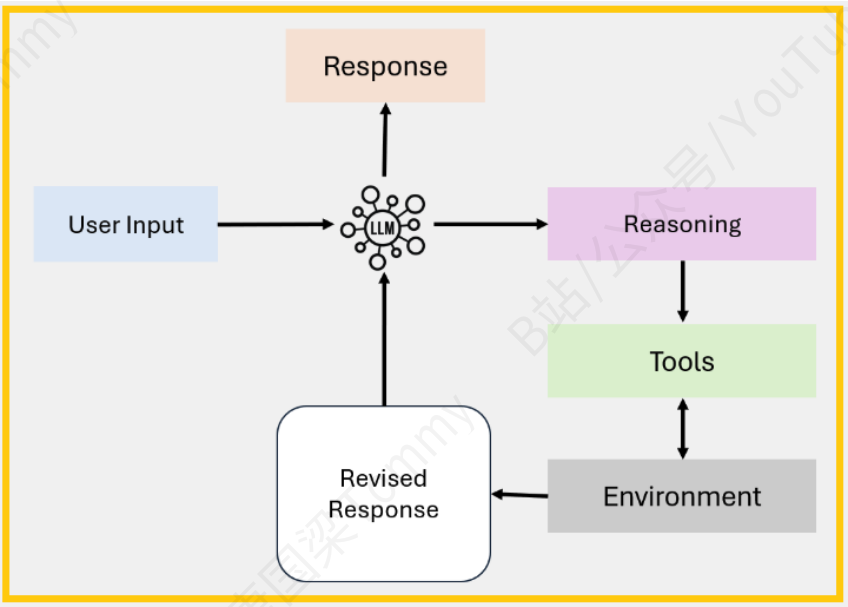


3️⃣ 工具使用 (Tool Use)

工具使用模式扩展了智能体的能力，使其能够利用外部资源（如API、数据库、计算引擎）执行超出其原生功能的任务。这包括：

- **信息检索**：通过网络搜索获取最新数据。
- **数据处理**：使用编程语言（如Python）进行复杂计算或数据分析。
- **系统集成**：与其他软件系统交互，实现任务自动化。

例如，AutoGen框架允许智能体调用多种工具，增强其在数据分析、代码生成等方面的能力。

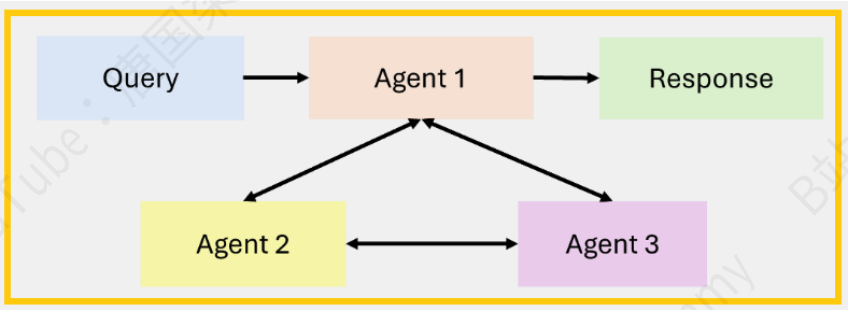


4️⃣ 多智能体协作 (Multi-Agent Collaboration)

多智能体协作模式通过多个智能体的协同工作，实现任务的专业化处理和并行执行。其核心优势包括：

- **角色分工**：不同智能体承担特定子任务，提高整体效率。
- **信息共享**：智能体之间交换中间结果，确保任务的一致性和连贯性。
- **弹性扩展**：系统可根据需求动态调整智能体数量和功能，适应不同规模的任务。

研究显示，多智能体协作在企业应用中表现出色，能够显著提升任务完成率和系统响应速度。



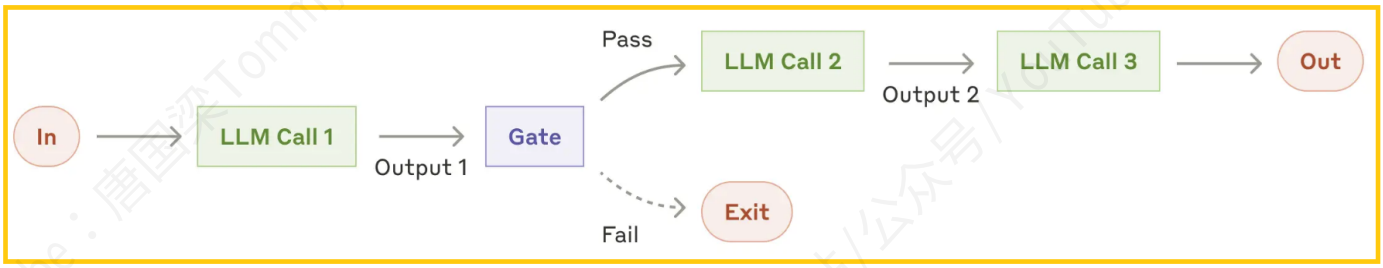
---

6. Agent 工作流模式

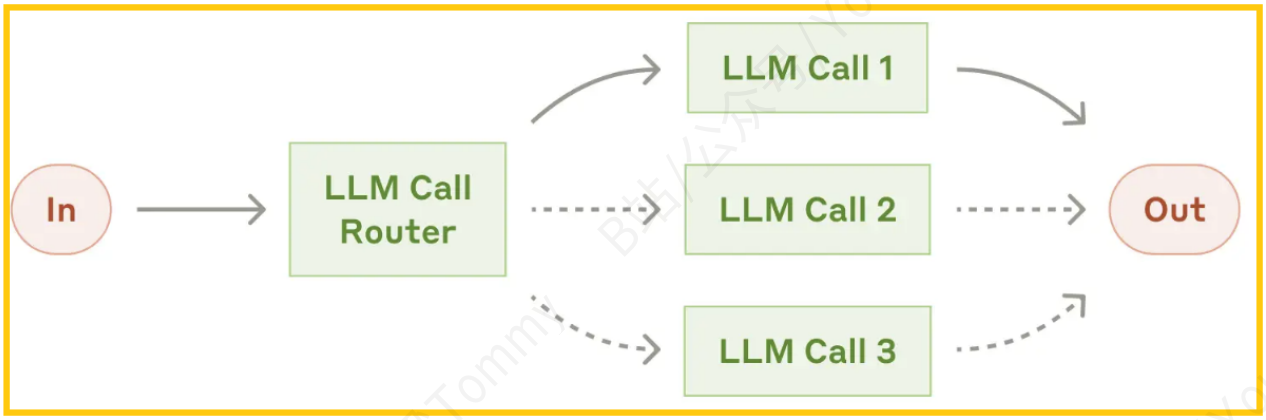
```
# 参考Anthropic定义
https://www.anthropic.com/engineering/building-effective-agents
```

智能体工作流模式构建 LLM 应用程序，以优化性能、准确性和效率。根据任务的复杂性和处理需求，有不同的适用方法：

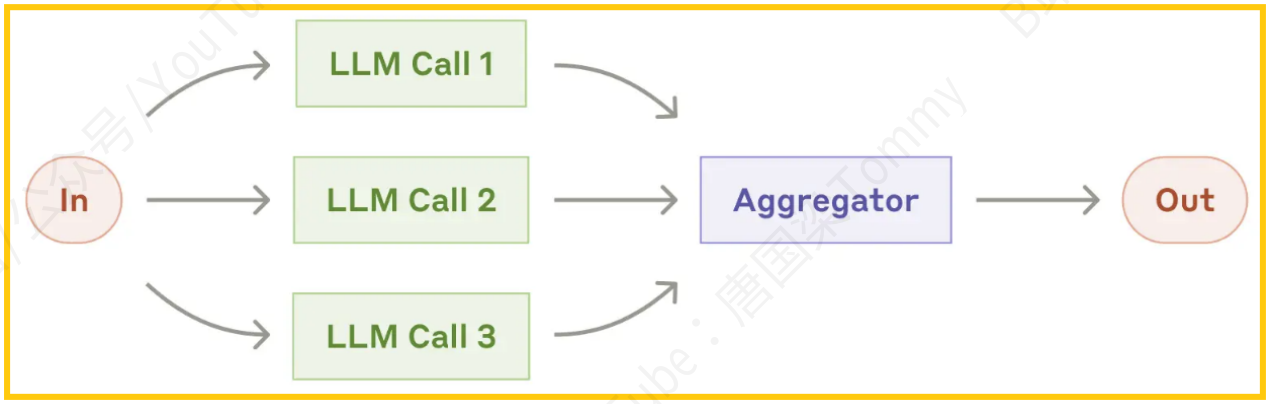
1️⃣ **提示链 (Prompt Chaining):** 将复杂任务分解为多个步骤，每个步骤都以前一步骤为基础。这是一种结构化的方法，通过简化每个子任务来提高准确性，但顺序处理可能增加延迟。



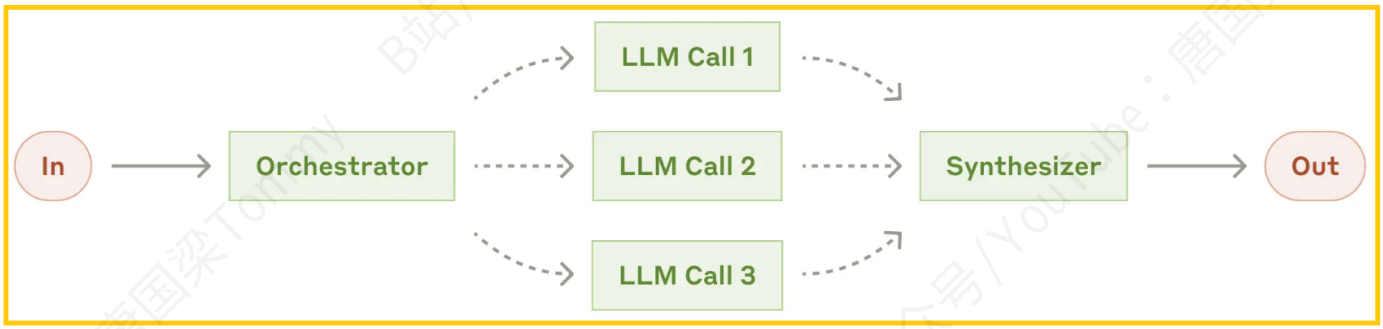
2️⃣ **路由 (Routing):** 对输入进行分类，并将其定向到适当的专业提示或流程。这确保了不同类型的查询或任务得到单独处理，提高了效率和响应质量。适用于不同输入需要不同处理策略的场景，例如客户服务查询分类。



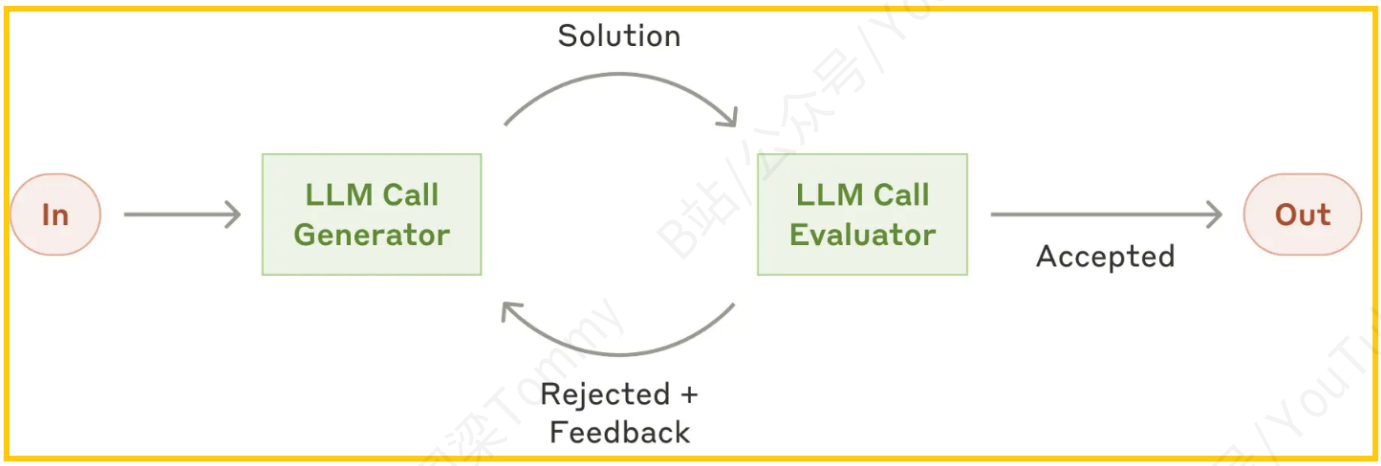
3 **并行化 (Parallelization):** 将任务分解为独立进程并行运行，从而减少延迟并提高吞吐量。可分为分段（独立子任务）和投票（多个输出以提高准确性）。适用于任务可独立执行以提高速度，或多个输出可提高置信度的场景。



4 **协调者-工作者 (Orchestrator-Workers):** 协调者模型动态地将任务分解为子任务，分配给专业工作者模型，并汇总结果。与并行化不同，它能适应不同的输入复杂性。适用于需要动态分解和实时适应的任务，其中子任务不是预定义的。



5 **评估者-优化者 (Evaluator-Optimizer):** 通过生成初始输出并根据评估模型的反馈进行优化，迭代地改进内容。当迭代优化可以显著提高响应质量，特别是在存在明确评估标准的情况下，这种方法是有效的。



7. Agentic RAG 架构分类

```
# 参考资源
1. What is Agentic RAG
https://weaviate.io/blog/what-is-agentic-rag
```

Agentic RAG 系统可以根据其复杂性和设计原则分为不同的架构框架。

7.1 单个 Agentic RAG

- **定义：**

  Single-Agent RAG 是 Agentic RAG 的最基础形式，系统中只引入一个智能体（Agent）来执行任务，主要用于处理多数据源检索任务。

- **架构特点：**
  - **核心角色**：一个智能体负责整个任务流程，包括检索、工具选择、上下文判断与答案生成。
  - **路由能力**：该智能体可以作为“路由器”（Router），根据查询内容动态决定从哪个知识源或工具中提取信息。
  - **接入工具**：支持接入多个外部工具或数据源，如：
    - 向量数据库（如 Weaviate）
    - Web 搜索引擎
    - 内部API（如Slack、邮箱等）