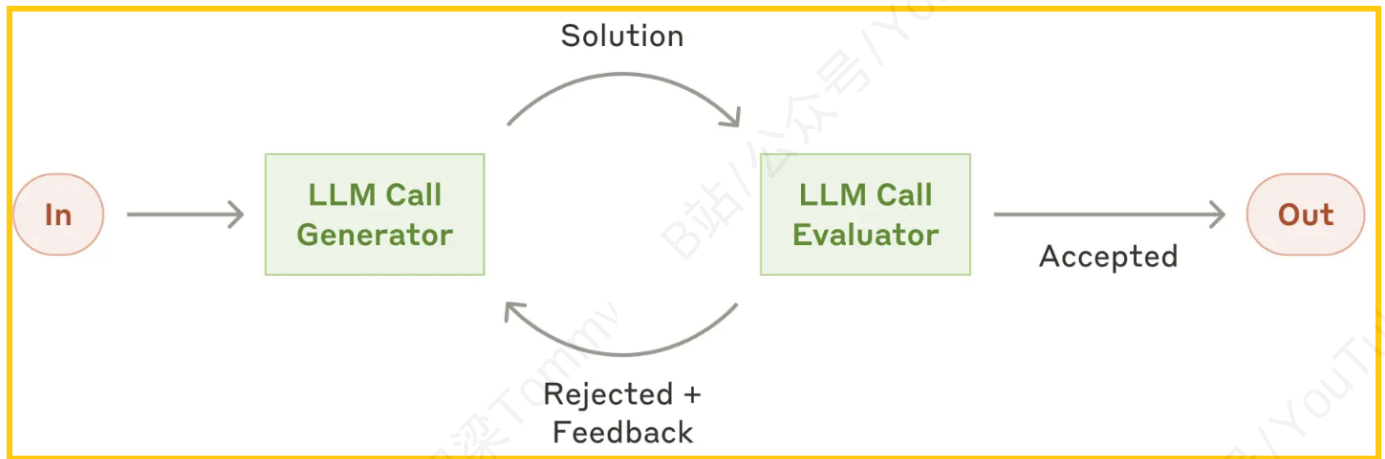




7. Agentic RAG 架构分类

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。

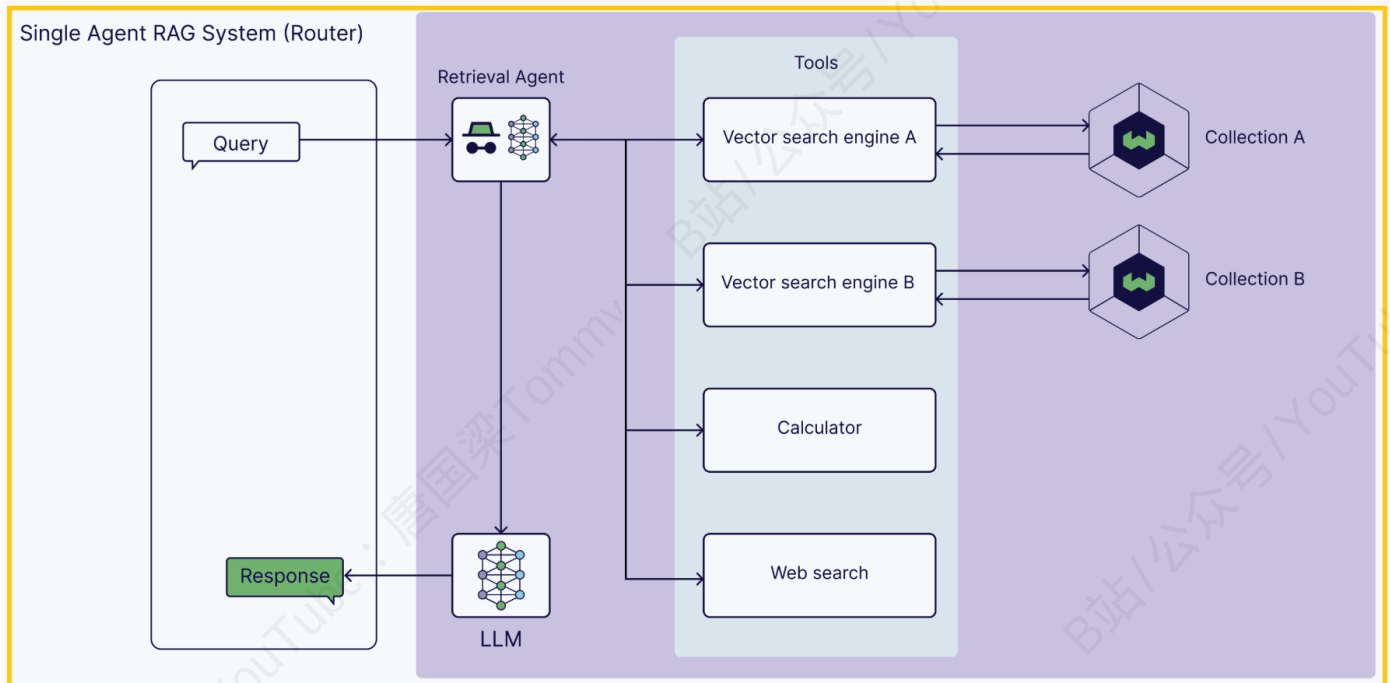
7.1 单个 Agentic RAG

• 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）



- 应用场景示例：

- FAQ系统：可根据问题类型判断是从数据库、网页还是FAQ文档中获取答案。
- 客服Agent：接入多个业务系统（工单、邮件、用户行为日志）提供支持。

- 优点：

- 实现简单，结构清晰。
- 适用于数据源不复杂或智能体任务边界明确的场景。
- 较传统RAG更灵活智能，可动态选择信息源。

- 局限：

- 所有任务都集中在一个Agent上，易造成负载过重。
- 智能体功能杂糅（路由、检索、判断、生成），不利于模块化优化。

7.2 多个 Agentic RAG

- 定义：

Multi-Agent RAG 是更复杂的 Agentic RAG 实现，系统中引入多个功能划分明确的智能体，通过**主控智能体（master agent）**协调各个**专职智能体（specialized agents）**协同完成任务。

- 架构特点：

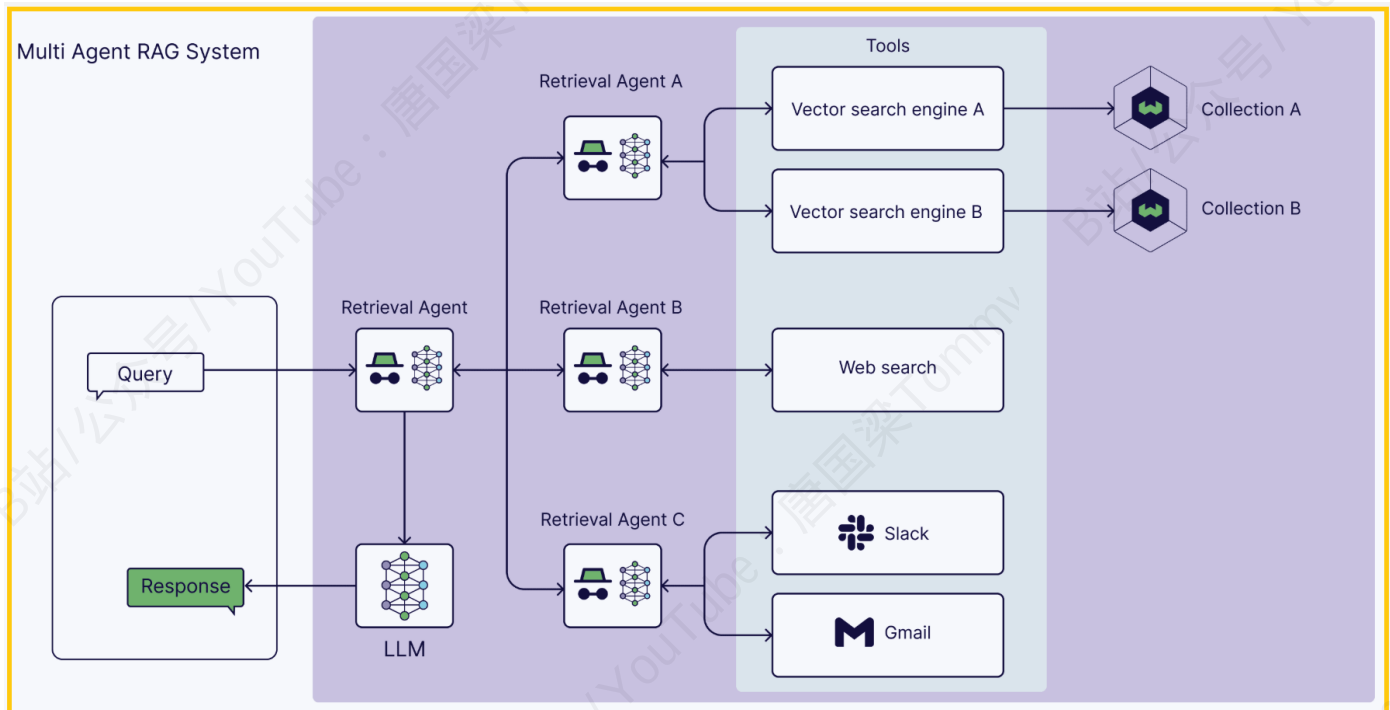
- 主控智能体:

- 负责任务拆解、子任务分配、汇总各个子任务结果。

- 专职智能体:

每个Agent负责特定检索任务, 比如:

- Agent A: 检索内部文档数据
- Agent B: 基于Web执行实时搜索
- Agent C: 调用公司API查邮箱



- 应用场景示例:

- 企业级问答系统: 结合私有知识库、ERP系统数据、外部API、公共网页搜索。
- 多模态信息检索: 文字Agent检索文本, 图像Agent调用图像识别模块, 视频Agent查找相关片段。

- 优点:

- 模块化强: 任务明确、职责清晰, 便于扩展和优化。
- 支持更复杂、多阶段的任务链。
- 各Agent可独立优化, 便于并行处理, 提升效率。

- 局限:

- 架构更复杂, 需要调度与通信机制。
- Agent间协调不当可能造成冲突、冗余或信息损失。
- 成本更高 (模型调用频率上升), 需优化任务拆解逻辑。

7.3 层级式 Agentic RAG

核心理念:

采用多层次 (Hierarchical) 智能体架构, 由上层智能体 (如主控 Agent) 协调下层专职智能体 (如检索 Agent、生成 Agent) 完成复杂任务。

架构特点:

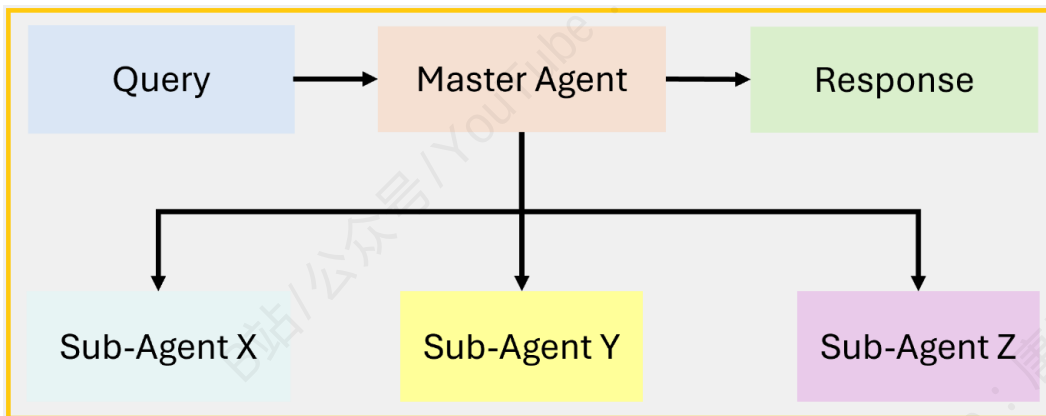
- **主控智能体:** 负责任务分解、子任务分配和结果整合。
- **专职智能体:** 各自承担特定功能, 如检索、生成、验证等。
- **层级协作:** 上层智能体可监督和指导下层智能体的操作。

适用场景:

- 需要多步骤推理和复杂任务管理的应用, 如多文档问答、跨领域信息整合等。

优势:

- 提高系统的模块化和可扩展性。
- 增强任务处理的灵活性和准确性。



7.4 纠错型 Agentic RAG

参考来源

1. Corrective Retrieval Augmented Generation
<https://arxiv.org/pdf/2401.15884>

核心理念:

引入反馈机制, 智能体在生成初步回答后进行自我评估, 识别并纠正潜在错误, 提升回答质量。

架构特点:

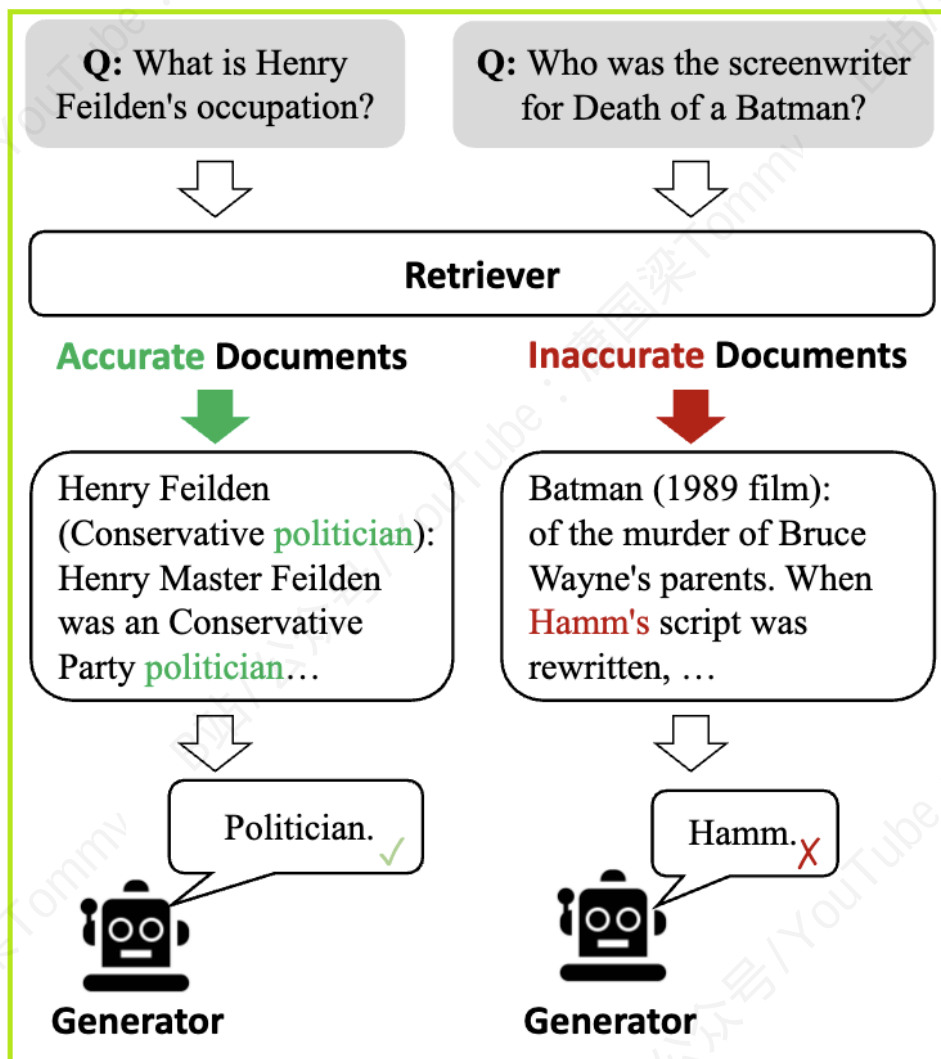
- **初步生成:** 智能体根据检索结果生成初步回答。
- **自我评估:** 引入“批评者 (critic)”模块, 评估回答的准确性和一致性。
- **迭代优化:** 根据评估结果, 智能体对回答进行修正, 直至满足质量标准。

适用场景:

- 对回答准确性要求高的应用, 如医疗咨询、法律问答等。

优势:

- 显著降低错误率和幻觉现象。
- 提升用户对系统回答的信任度。



7.5 自适应 Agentic RAG

参考来源

1. Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity

<https://arxiv.org/pdf/2403.14403>

2. <https://www.analyticsvidhya.com/blog/2025/01/agentic-rag-system-architectures/>

核心理念： 智能体根据任务需求和中间结果，动态调整检索策略和工具选择，实现更灵活的任务处理。

架构特点：

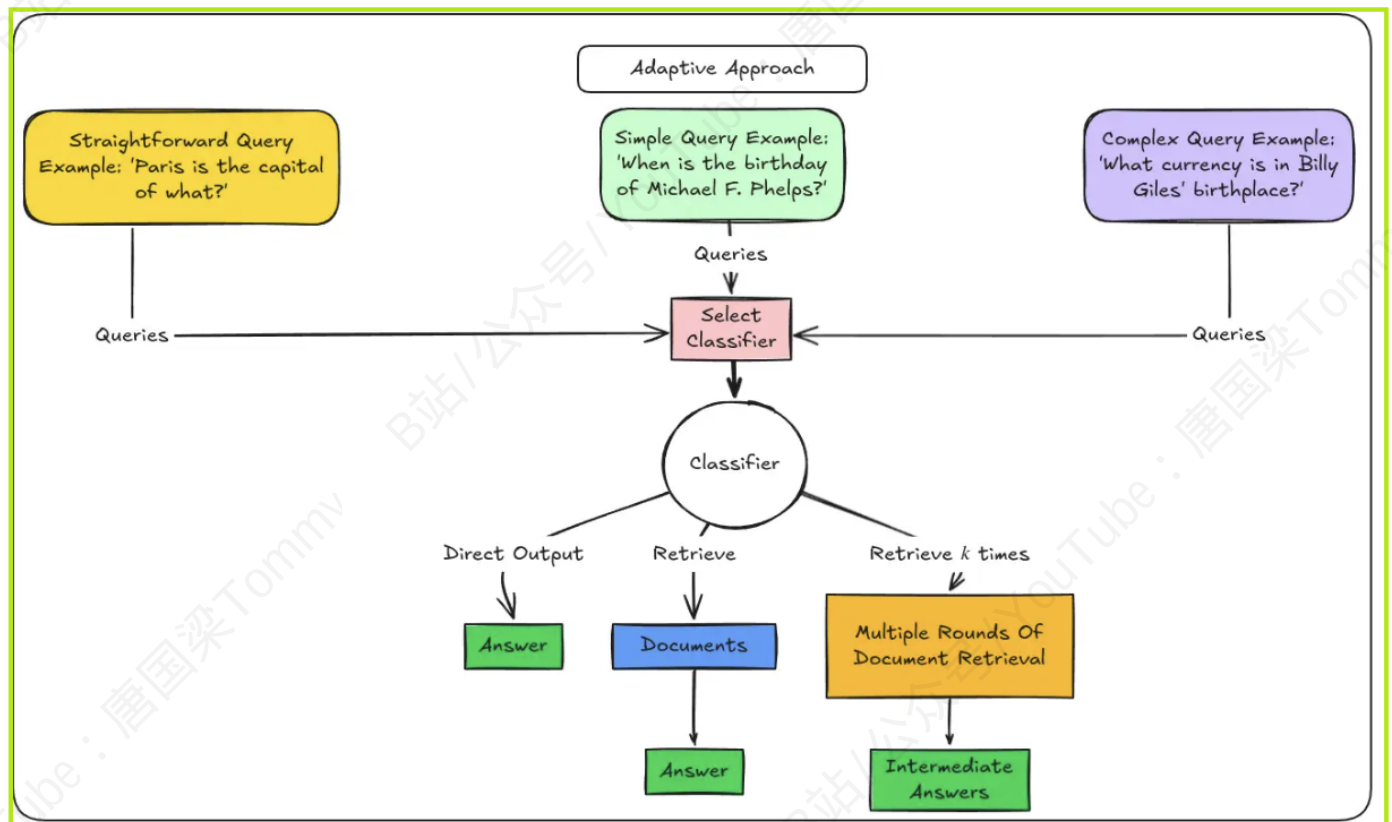
- **动态决策：** 智能体可根据当前上下文，决定是否需要进一步检索或更换工具。
- **多工具集成：** 支持接入多种工具和数据源，如向量数据库、Web 搜索、API 等。
- **实时反馈：** 根据中间结果，实时调整任务执行路径。

适用场景：

- 任务需求多变或信息来源多样的应用，如实时新闻摘要、跨领域研究等。

优势：

- 增强系统的灵活性和适应性。
- 提高任务处理的效率和准确性。



7.6 图结构 Agentic RAG

参考来源

<https://neo4j.com/blog/developer/graphrag-and-agentic-architecture-with-neoconverse/>

核心理念： 利用图数据库表示知识结构，智能体通过图遍历和多跳推理，实现更深层次的信息检索和理解。

架构特点：

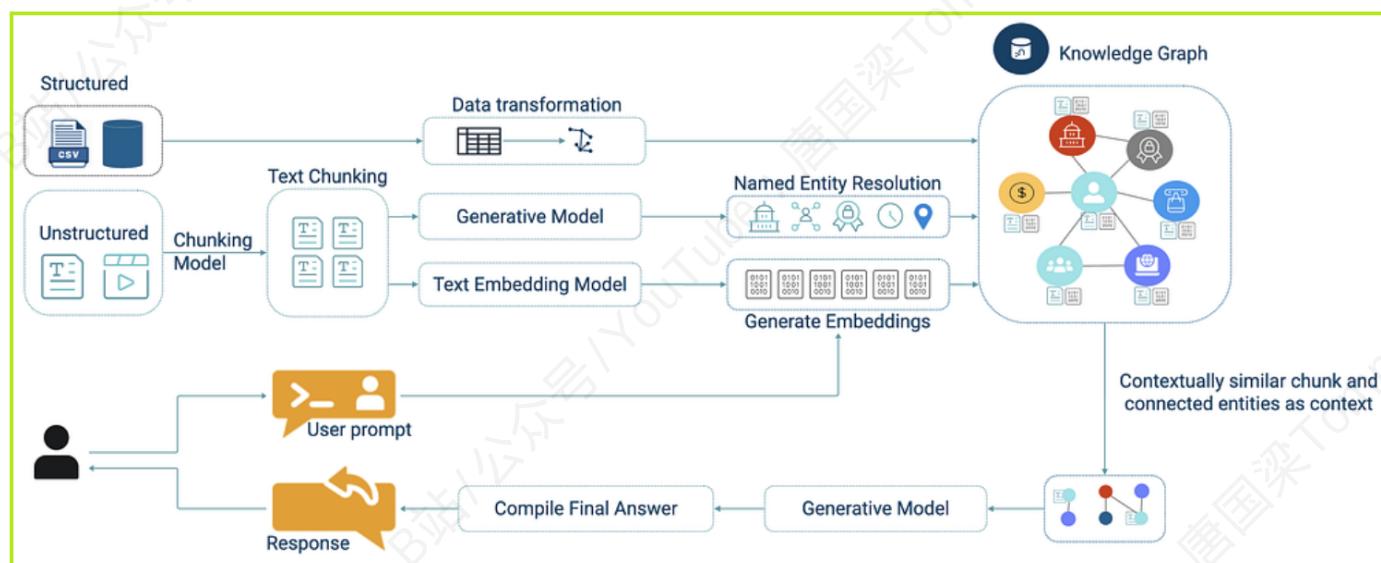
- **知识图谱：** 将实体及其关系表示为图结构，支持复杂的关系查询。
- **多跳推理：** 智能体可沿图中的路径进行多步推理，获取间接相关的信息。
- **混合检索：** 结合向量检索和图检索，提升信息检索的全面性。

适用场景：

- 需要理解复杂关系和上下文的应用，如学术研究、法律分析等。

优势：

- 提升信息检索的深度和广度。
- 增强系统的推理能力和上下文理解能力。



7.7 文档流程式 Agentic RAG

参考来源

<https://www.llamaindex.ai/blog/introducing-agentic-document-workflows>

核心理念： 通过智能体自动化处理文档相关任务，实现从文档解析、信息提取到结构化输出的全流程自动化。

架构特点：

- **文档解析：** 智能体解析文档结构，识别关键信息。
- **信息提取：** 结合 RAG 技术，从文档中提取相关信息。

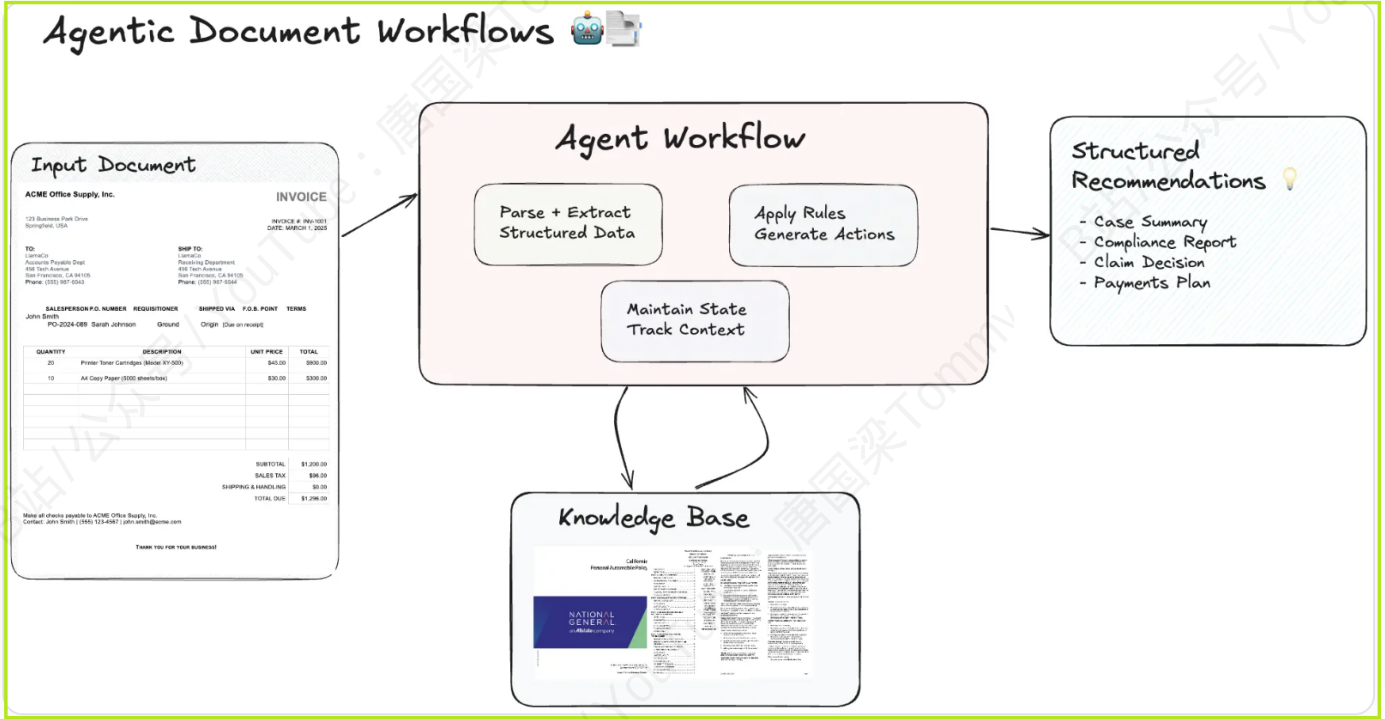
- **结构化输出:** 将提取的信息组织成结构化格式, 便于后续处理。

适用场景:

- 需要处理大量文档的应用, 如合同审查、报告生成、表单填写等。

优势:

- 显著提高文档处理的效率和准确性。
- 减少人工干预, 降低操作成本。



8. RAG vs Agentic RAG 总结比较

在 RAG 领域, 人们正从“静态检索-后生成”迈向“具备规划、调用多工具、可自我反思的 **Agentic RAG**”。后者把整条 pipeline 交给 AI Agent 来主导: 按需拆解任务、调用外部工具、多轮检索-推理-评估-再检索, 并可在多智能体协作中处理实时数据和复杂决策。

8.1 核心差异一览

维度	传统 RAG	Agentic RAG
检索手段	单一向量数据库或搜索 API，静态一次性调用	可在多库/多模态源间多轮检索，检索策略由智能体动态调整
工具生态	基本无工具或仅嵌入检索	内建函数调用 / Toolformer-式自动用工具（计算器、SQL、代码执行等）
推理方式	单步“检索→生成”	规划-推理-反思-迭代的多步骤 reasoning（Self-Reflective / Recursive RAG）
协作模式	单一 LLM	多智能体分工：规划 Agent、检索 Agent、评估 Agent 等
实时数据	依赖预构建索引，难以对接流数据	支持 API / streaming 源，实时查询与融合
结果质量控制	无内置校验，幻觉由开发者外部兜底	Agent 级重检索、分级与自评估，显著降低幻觉率
适用场景	FAQ、静态知识库问答	金融行情解读、数据分析助手、企业 Copilot 等需要决策与写作的一体化 workflows

8.2 能力深度对比

1 工具与函数调用

Agentic RAG 通过 OpenAI Function Calling、LangChain Tool 等机制，把 LLM 生成的结构化调用参数映射到外部 API；智能体可以自动决定何时调用计算器、SQL、WebSearch 等工具，从而突破单纯检索的上限。

2 多轮推理与自我反思

- **Recursive / Self-Reflective 回路**：模型先生成答案，对答案或检索结果自评打分，不足时重写查询或补充文档，再次生成，直至评分达标。
- **多级排序-质检**：检索阶段可引入二阶段 reranker；生成后用专门的 Critic Agent 追问事实依据，过滤幻觉。

3 多智能体协作

Pathway、LangGraph 等框架提供基于消息流的多 Agent 流程编排：Planner 负责分解任务，Retriever 负责查询，Analyzer 进行数据处理，Writer 完成最终输出——提高可伸缩性与容错性。

4 实时与多模态数据融合

实时行情或 IoT 流水通过流检索组件接入；同时可在文本外加入图像、表格检索，对结果进行跨模态融合，形成“动态-多模态 RAG”新形态。

Agentic RAG 把**检索增强**与**智能体规划**结合，补齐了传统 RAG 在多步骤推理、实时数据与任务自动化上的短板，是构建下一代企业 Copilot、智能决策助理的关键范式。通过分层代理、工具调用、反思循环与严格安全治理，可以在保证成本可控的前提下，显著提升系统的灵活性、准确性与扩展能力。