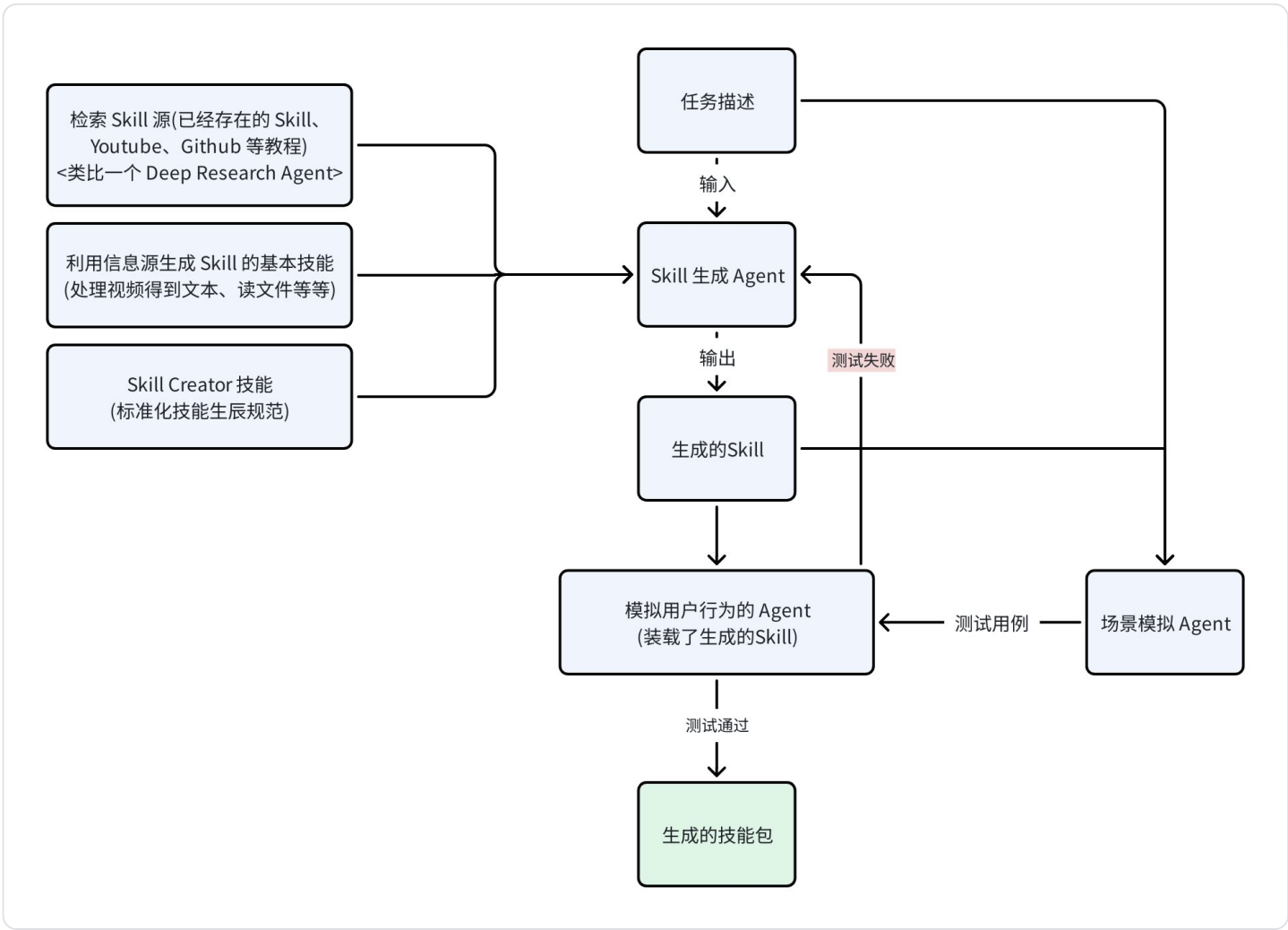


Anything2Skill（多 Agent 架构）

1. 总体目标

Anything2Skill 的目标是：用户只需输入一个“任务描述”，系统就能自动生成一份可复用、可交付的“技能包”。技能包包含标准化的 Skill 表达（例如技能说明、步骤、约束、依赖工具/环境等），并且通过自动化测试验证其可用性。



总体架构图

2. 输入与主流程（从任务到技能包）

任务描述（输入）

用户提交自然语言任务（如“我需要一个能帮我按照毕业论文要求排版的技能”等），作为系统的唯一入口输入。

Skill 生成 Agent（核心中枢）

Skill 生成 Agent 接收任务描述后，会综合利用三类能力来源来完成 Skill 的“生成与组织”：

- **技能检索能力**：优先检索已有 Skill 资产与外部知识源（如已有 Skill 库、YouTube 教程、GitHub 项目/教程等），类似一个“Deep Research Agent”，用于快速找到可复用的高质量解法。
- **信息源处理能力**：把各种信息源转换为可用的技能材料（例如视频转文字、文档/文件读取与结构化抽取等），为后续 Skill 生成提供可靠上下文。
- **Skill Creator 能力**：把抽取到的内容转成“标准化技能表示”，包括步骤化流程、输入输出、边界条件、依赖、失败处理、可测试性等。

生成的 Skill（输出产物）

Skill 生成 Agent 输出一个初版 Skill，并准备进入验证阶段。

3. 测试与迭代闭环（确保技能可用）

场景模拟 Agent（测试用例生成器）

场景模拟 Agent 根据“任务描述”和“生成的 Skill”，构造可执行的**测试用例**（测试输入、环境假设、预期结果、边界情况等），用于验证 Skill 是否能在目标场景下完成任务。

模拟用户行为的 Agent（执行器，装载 Skill）

该 Agent 会“加载生成的 Skill”，并在测试用例驱动下模拟真实用户执行流程：调用工具、按步骤操作、观察中间状态与最终输出，从而判断 Skill 是否可执行、是否稳定、是否满足预期。

测试结果：

- **测试通过**：生成的 Skill 被认定可用，进入最终交付阶段。
 - **测试失败**：失败信号回流到 **Skill 生成 Agent**，触发重新生成/补全/修复（例如补充缺失步骤、修正前置条件、增强异常处理、改写工具调用方式等），再进入下一轮测试，形成迭代闭环。
-

4. 最终交付

生成的技能包

当 Skill 在场景模拟与执行验证中通过测试后，系统输出“技能包”作为最终成果。技能包面向复用与分发，通常包含：

- 标准化 Skill 定义
- 可复现的测试用例与验证方式

