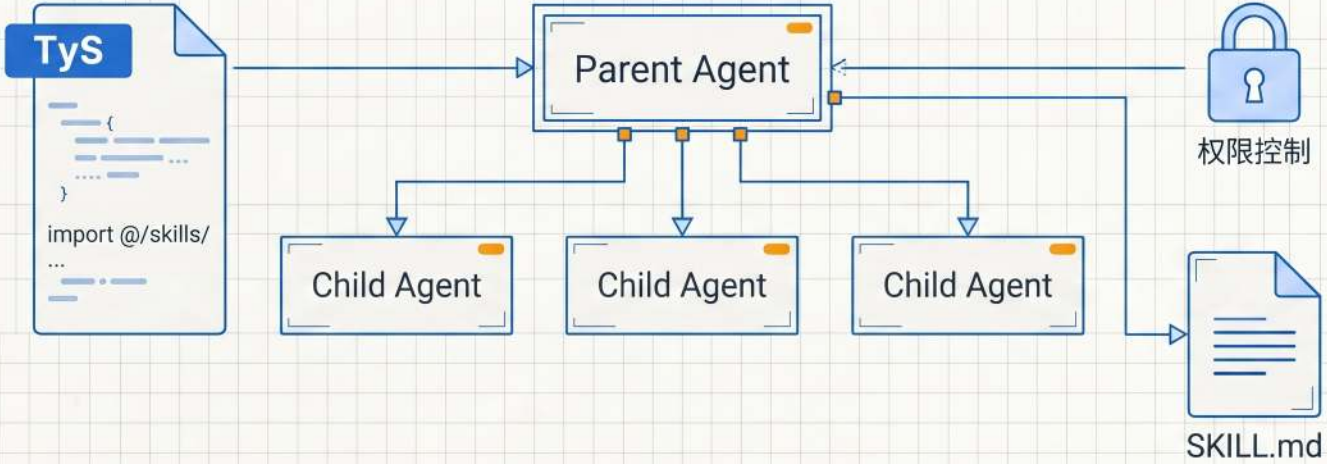
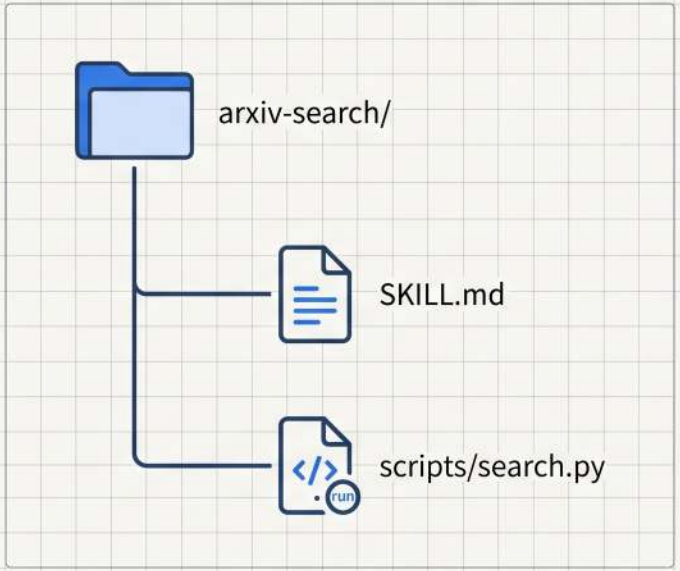
Deep Agents 实战

第 10 讲：Skills（下）— 高级用法



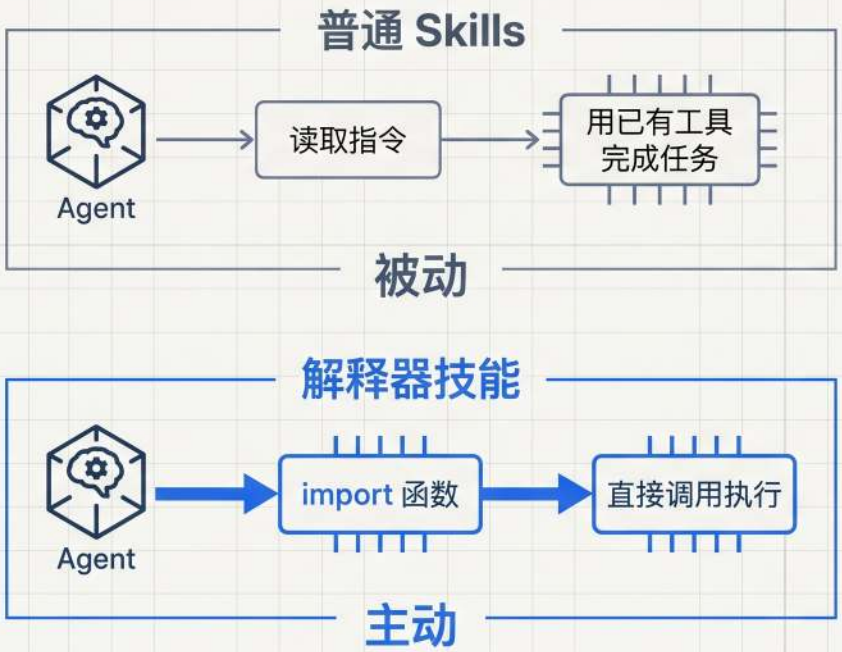
沙箱脚本

scripts/ 目录下的可执行代码



- Skills 可以包含可执行脚本（Python、Bash、JS 等）
- Agent 按照 SKILL.md 中的指令运行脚本
- 读取脚本：任何后端都可以
- 执行脚本：需要沙箱环境（如 Daytona）
- 沙箱外的 Skill 文件需要通过中间件同步进去

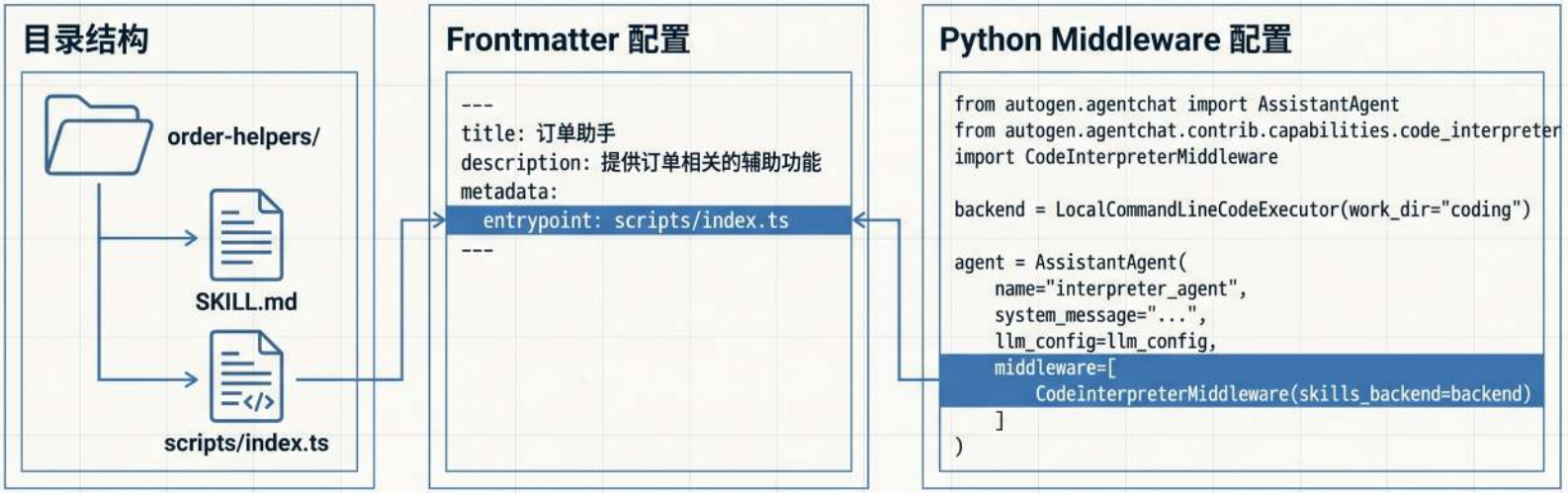
解释器技能
让 Agent import 经过测试的辅助函数



- 普通 Skills：Agent 读取指令，用已有工具完成任务（被动）
- 解释器技能：Agent 可以 import 模块中的函数，直接调用（主动）
- 适用场景：需要确定性执行的逻辑
  - 数据转换、格式验证、聚合计算
  - Agent 导入经过测试的辅助函数，而非每次从头生成

解释器技能设置

metadata.entrypoint + CodeInterpreterMiddleware



- Step 1：在 Frontmatter 的 metadata 中添加 **entrypoint**，指向入口文件
- Step 2：配置 CodeInterpreterMiddleware，使用相同的后端
- Agent 通过 await import("**@/skills/order-helpers**") 导入模块

解释器技能实战

Agent 在解释器代码中 import 并调用

源代码定义

TypeScript 入口文件 (scripts/index.ts)

```ts
// scripts/index.ts
export function groupByStatus(orders: Order[]): Record<string, Order[]> {
  // implementation of the function
  // ...
}
```

import : '@/skills/order-helpers'

Agent 调用

Agent 解释器中的 import 和调用

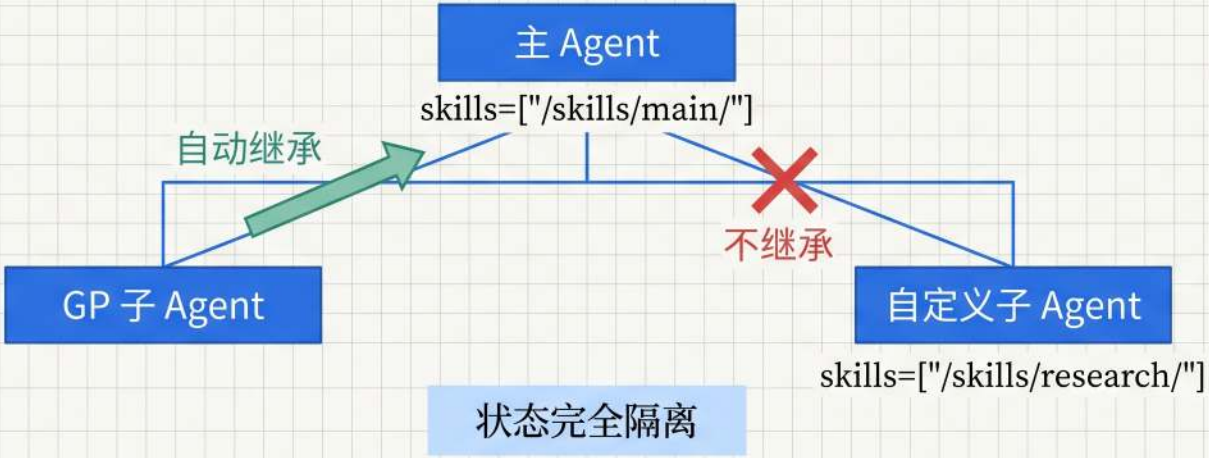
```ts
// Agent 解释器执行的代码
const { groupByStatus } = await import("@/skills/order-helpers");

// 调用函数
const orders = [....]; // 假设的订单数据
const result = groupByStatus(orders);

console.log(result); // 确定性执行结果
```

关键要点

- TypeScript 入口文件导出函数（如 groupByStatus）
- Agent 在解释器中通过 import 语句动态加载技能
- 确定性执行：每次调用结果一致，不依赖 LLM 生成，确保稳定性和可预测性

Skills 与子 Agent

GP 子 Agent 继承，自定义子 Agent 不继承



- 主 Agent：通过 skills=[...] 配置
- General-purpose 子 Agent：自动继承主 Agent 的 Skills
- 自定义子 Agent：不继承，需在定义中指定 skills 参数
- Skill 状态在 Agent 之间完全隔离

Skill 权限控制

用 FilesystemPermission 控制 Agent 对 Skills 的访问

mode="deny"：禁止写入 → 共享只读 Skills（知识库）

共享 Skills 只读

mode="interrupt"：写入时暂停审批 → 需要 Checkpointer

写入时审批

mode="allow"（默认）：Agent 可自由读写

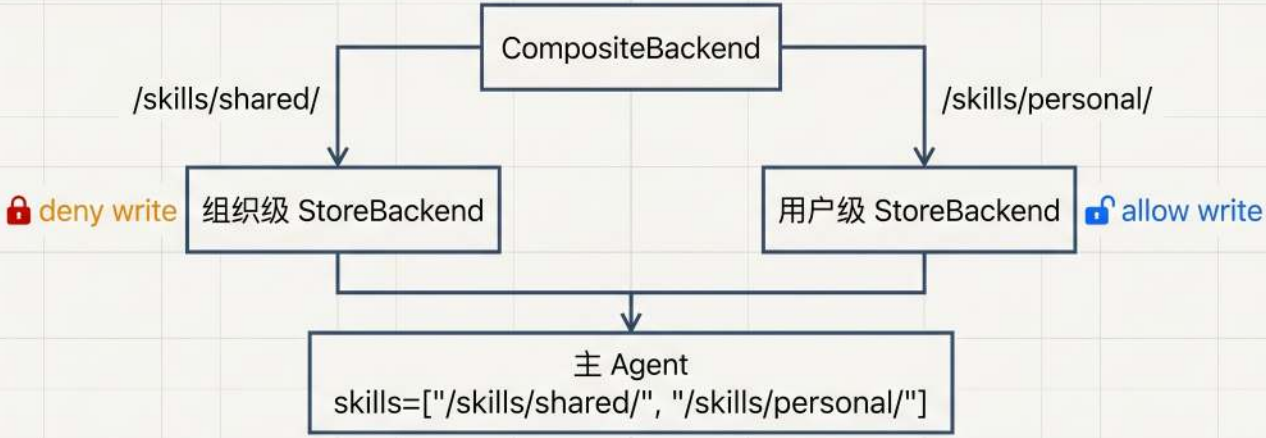
自由读写

典型场景

- 共享 Skills 只读 + 个人 Skills 可写

共享 + 个人 Skills

CompositeBackend 路由不同路径到不同 Store



- `/skills/shared/` → 组织级 StoreBackend（deny 写入）
- `/skills/personal/` → 用户级 StoreBackend（允许写入）
- 主 Agent 同时加载两个源：`skills=["/skills/shared/", "/skills/personal/"]`
- Agent 可以维护个人 **Skills**，但不能修改共享知识库

Skills、Memory 与 Tools

三种为 Agent 提供上下文或能力的机制

Skills

- 按需发现
- Agent 判断相关时读取
- SKILL.md
- 特定任务工作流

Memory

- 始终加载
- 启动时加载
- AGENTS.md
- 项目规范、编码风格

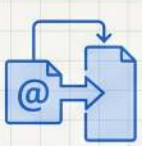
Tools

- 每轮可用
- 绑定到 Agent 的函数
- 单一原子操作

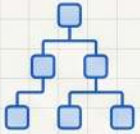
判断：

所有对话都需要 → **Memory** → 特定任务 → **Skills** → 执行操作 → **Tools**

Skills 完整回顾



import


继承树


锁

- 沙箱脚本：scripts/ 下的可执行代码，需要沙箱环境
- 解释器技能：metadata.entrypoint + import @/skills/，确定性执行
- 子 Agent 规则：GP 继承、自定义不继承、状态完全隔离
- 权限控制：deny（只读）/ interrupt（审批）/ allow（自由）
- 共享 + 个人 Skills：CompositeBackend 路由不同路径

下一讲：Human-in-the-Loop — 构建安全的人机协作流程