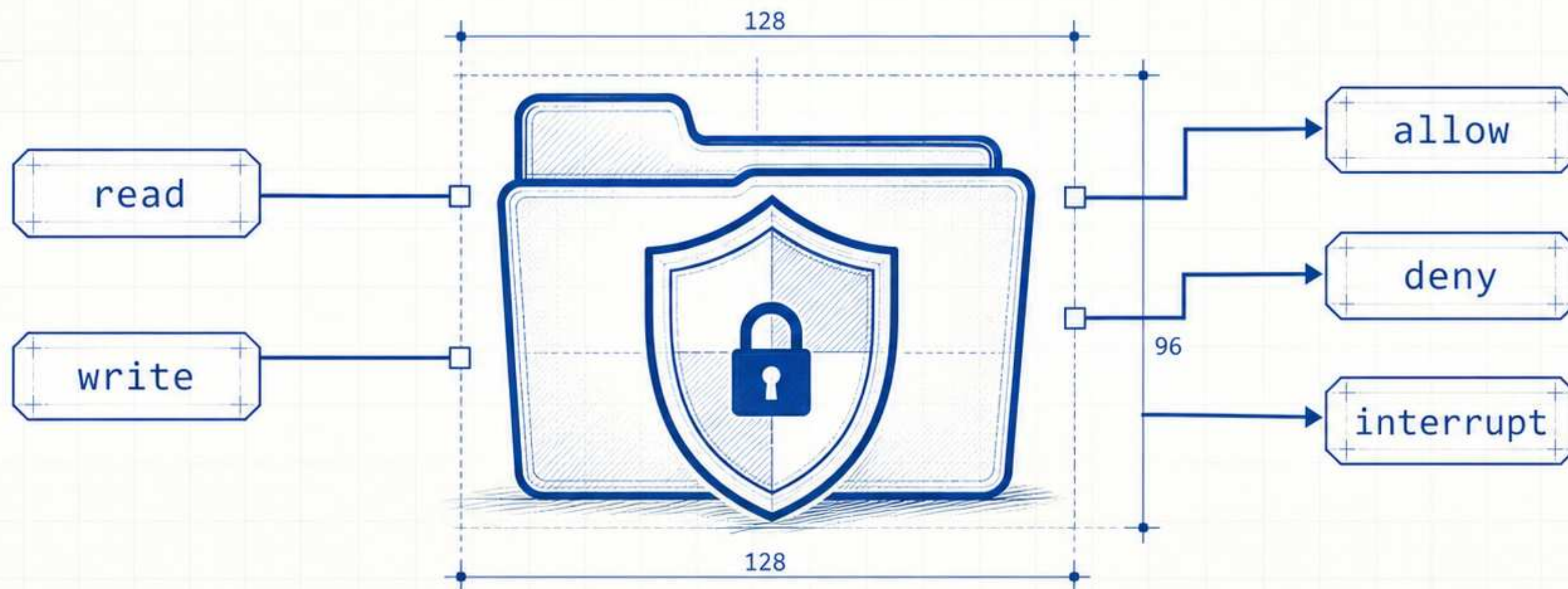
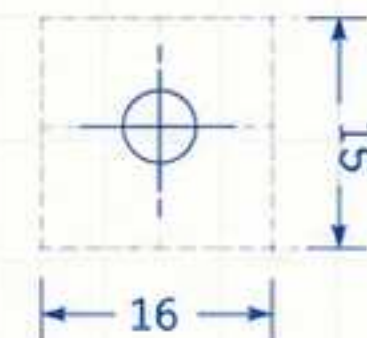


Deep Agents 实战

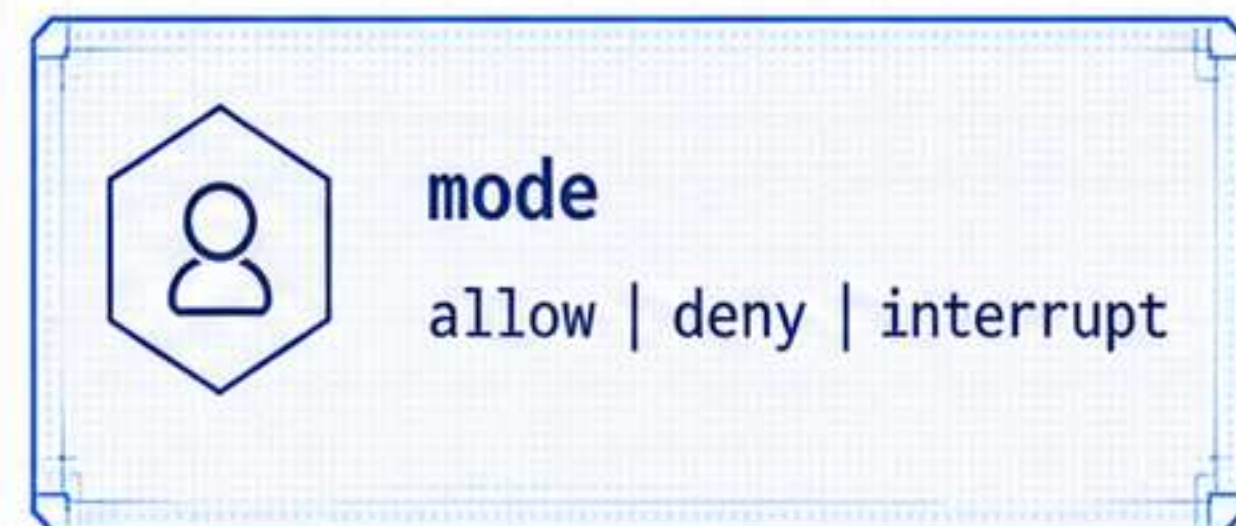
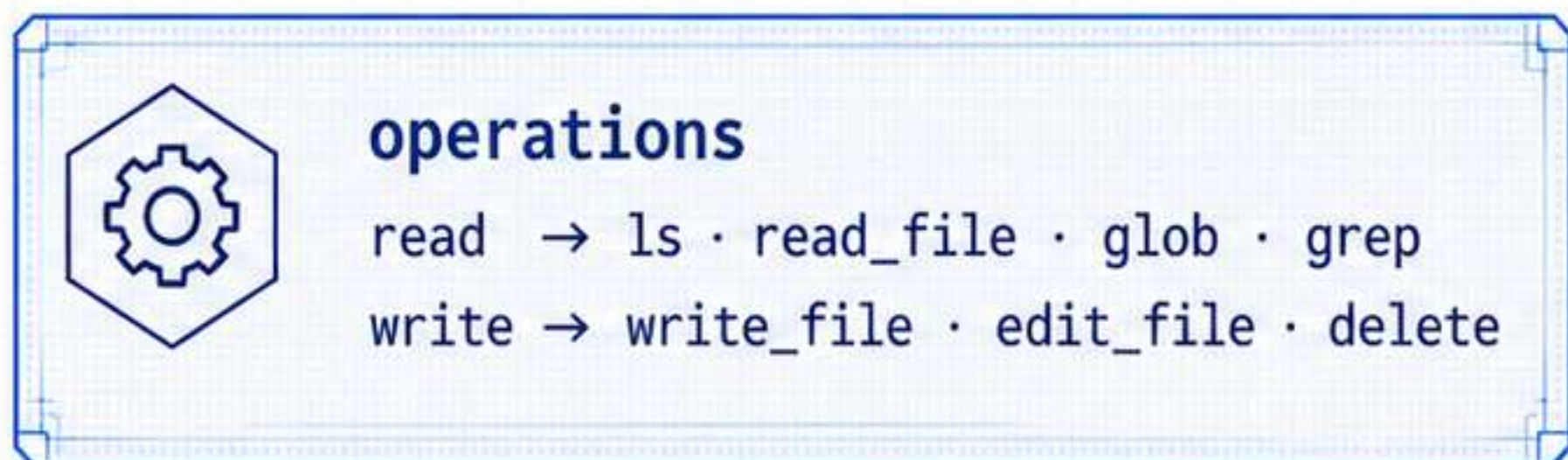
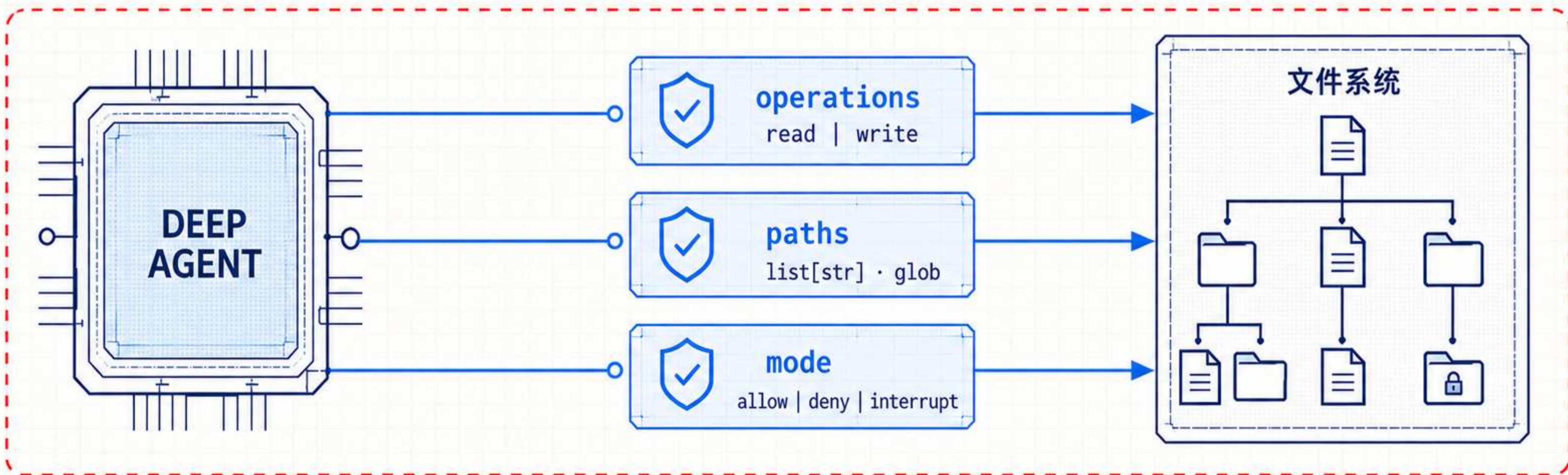
第 15 讲：文件系统权限

FS.PERM.015
SCHEMA v1.0
BLUEPRINT



文件能力就是副作用入口

权限配置必须回答三个问题



在入口函数中启用文件权限

`create_deep_agent(..., permissions=[...])`



```
from deepagents import FilesystemPermission, create_deep_agent

agent = create_deep_agent(
    model=model,
    backend=backend,
    permissions=[
        FilesystemPermission(
            operations=["write"],
            paths=["/**"],
            mode="deny",
        )
    ],
)
```

这条规则如何生效

operations=["write"]

仅匹配写操作: write_file / edit_file / delete

paths=["/"]**

匹配所有绝对路径

mode="deny"

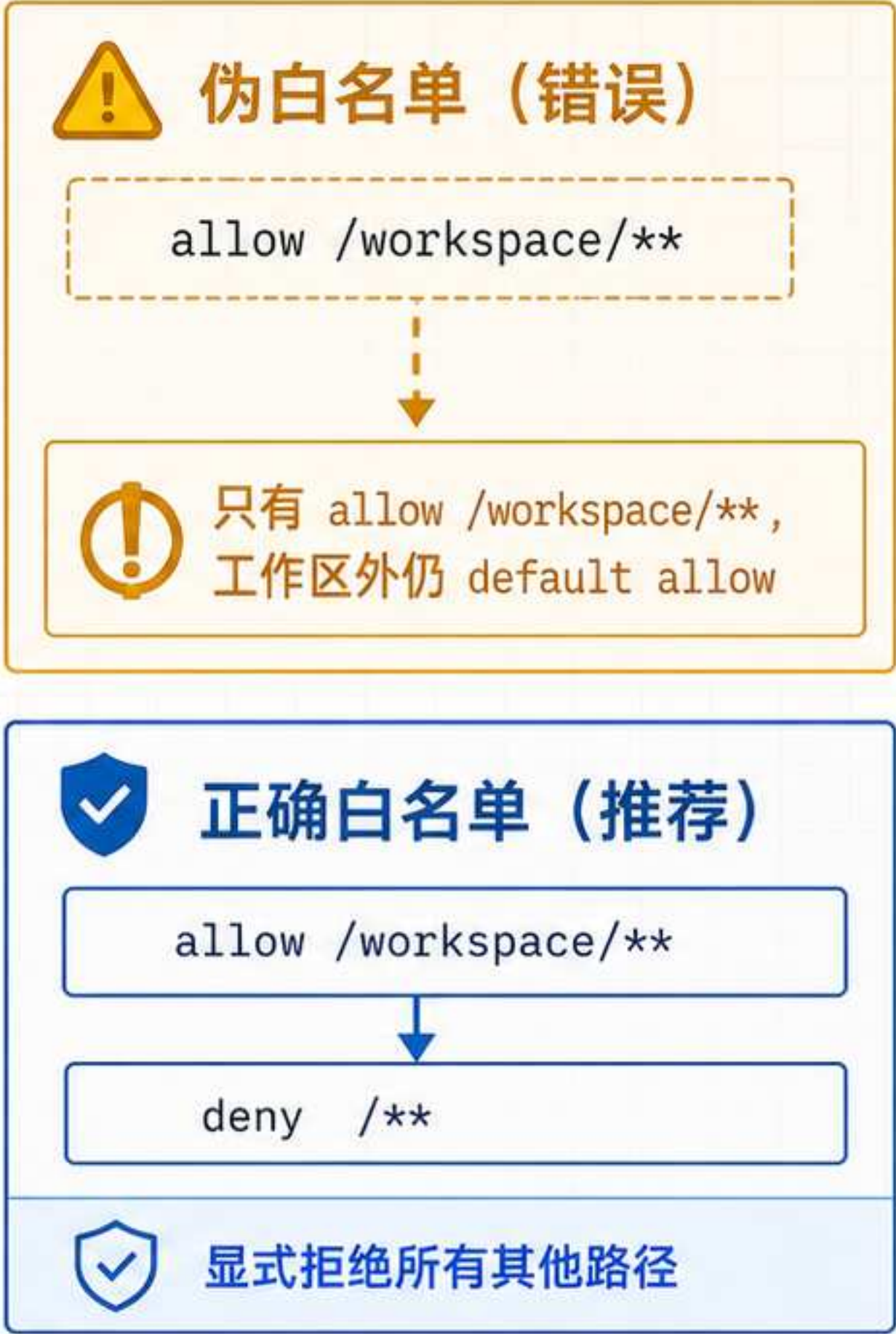
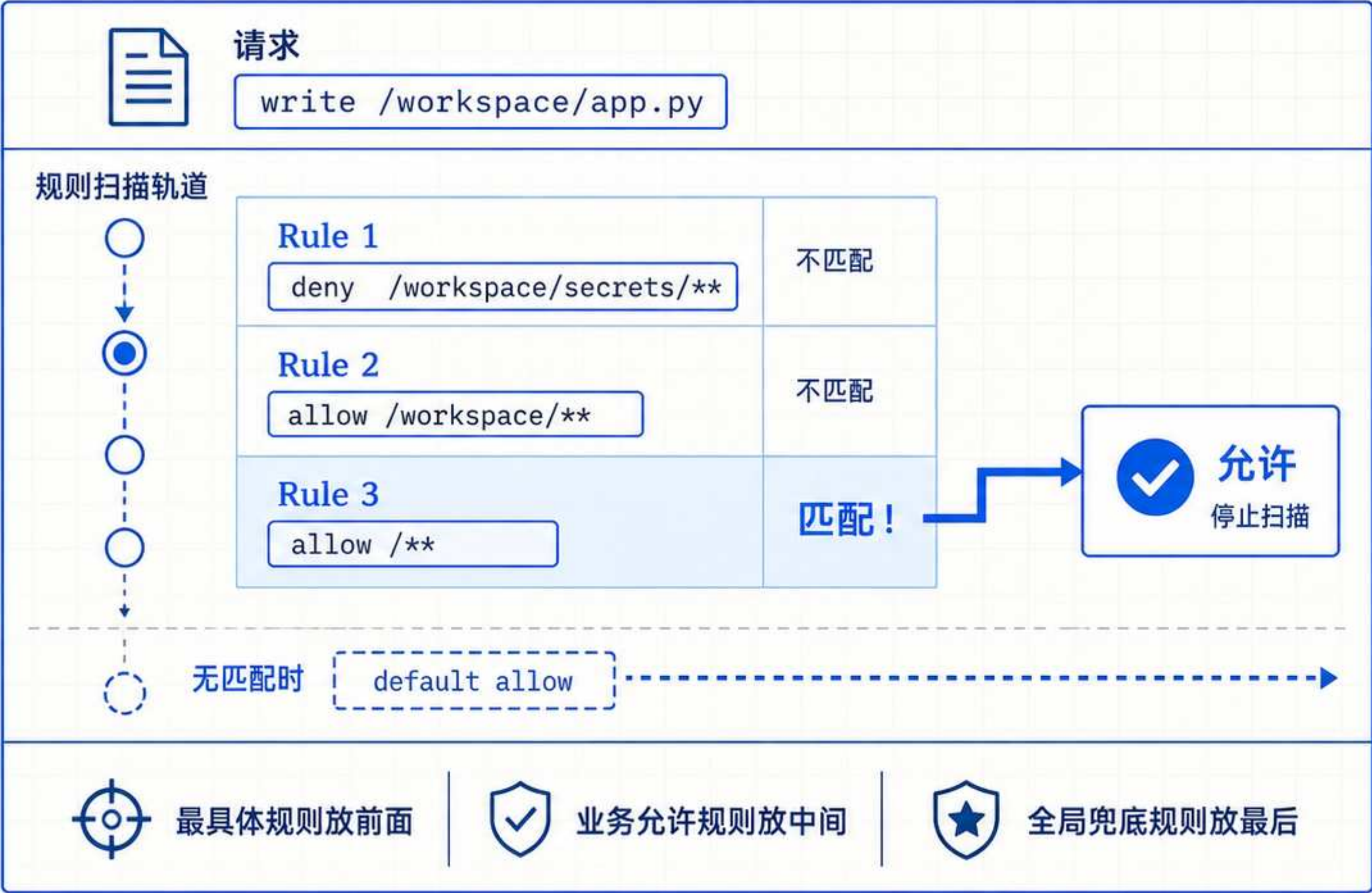
命中规则后直接拒绝



结果：所有写操作被拒绝；读取因未匹配此规则而默认允许

规则顺序决定最终结果

first-match-wins; 无匹配时默认允许



四种策略覆盖大多数场景

从最小副作用开始，再按职责放开

① 全局只读



read: /**, write: none

② 工作区白名单



/workspace



/**

read: /workspace, write: /workspace

③ 共享知识只读、用户记忆可写



/policies



/memories

read: /policies, write: /memories

④ 拒绝所有文件访问



read: none, write: none

interrupt 把敏感写入交给别人

暂停 → 审查 → approve / edit / reject → 恢复



需要 deepagents $\geq 0.6.8$



需要 checkpointer



恢复必须复用同一 thread_id

子 Agent 默认继承，显式配置整体替换

不是追加，也不是与父规则取交集



未配置：继承父 Agent 权限



已配置：完整替换父规则



子规则必须独立闭合



inherit



replace ☒



append

$\text{allow} = \text{parent.allow} \cup \text{child.allow}$
 $\text{resources} = \text{parent.resources} \cup \text{child.resources}$



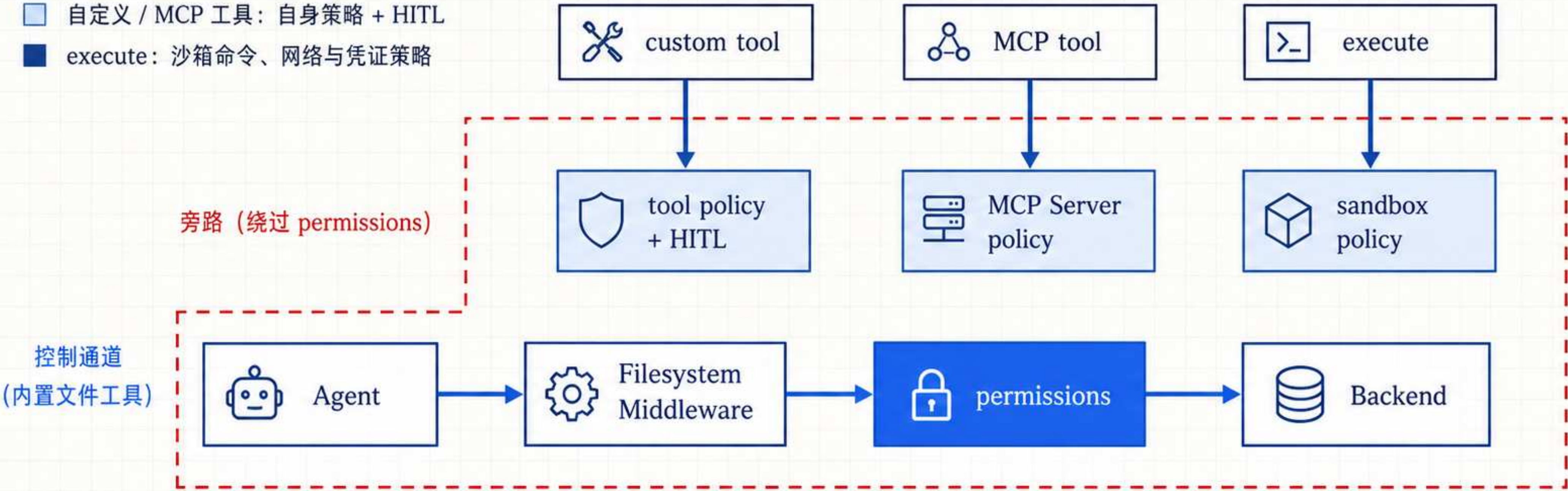
intersection

$\text{allow} = \text{parent.allow} \cap \text{child.allow}$
 $\text{resources} = \text{parent.resources} \cap \text{child.resources}$

权限只能守住内置文件工具

custom tool、MCP 与 execute 都要单独设防

- 内置文件工具：permissions
- 自定义 / MCP 工具：自身策略 + HITL
- execute：沙箱命令、网络与凭证策略



路径规则 $\neq$ 通用安全沙箱

上线前验证的是安全保证

不仅要测允许路径，更要证明所有兜底边界

工程验收清单



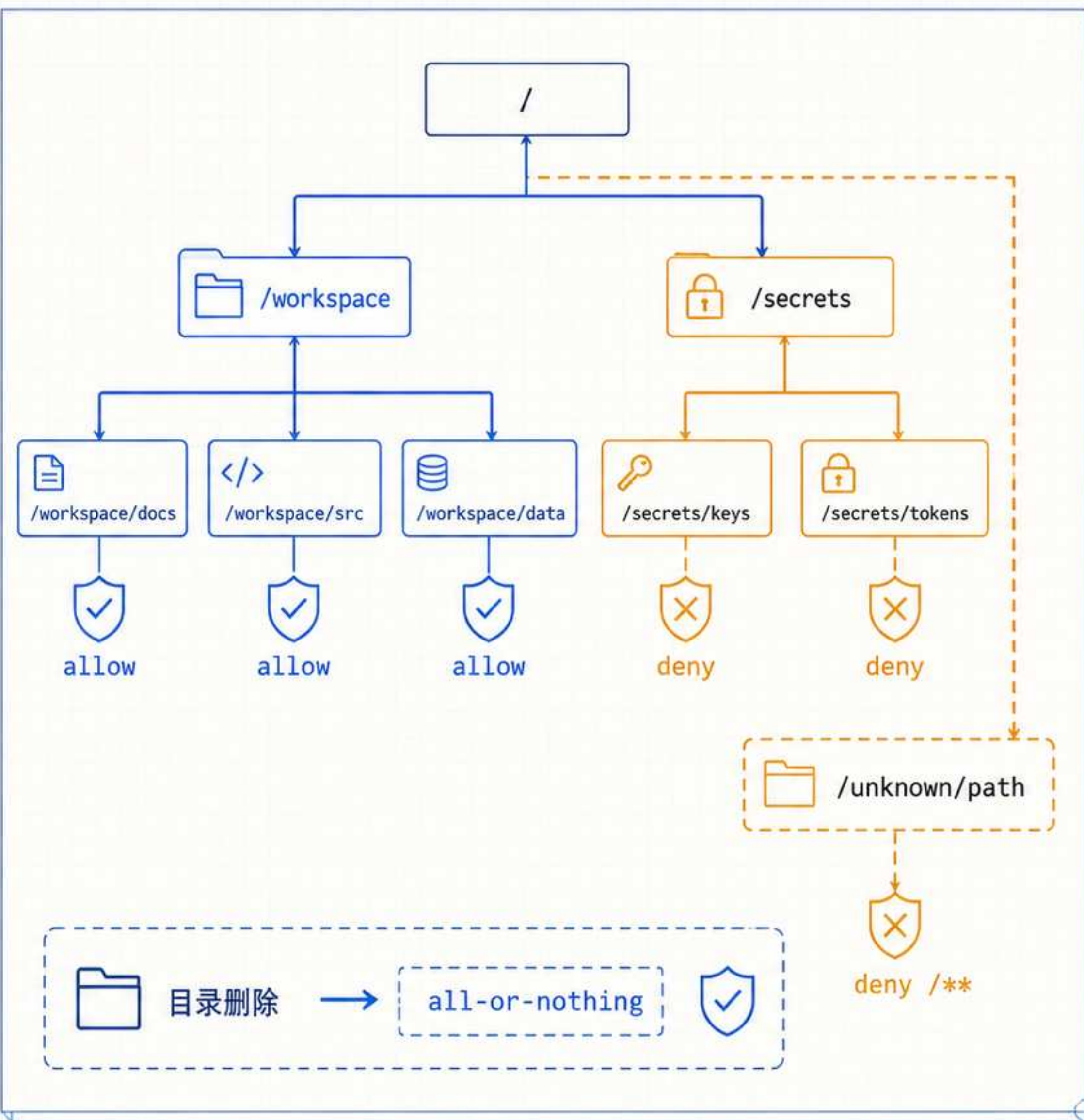
敏感路径：读 / 写 / 编辑 / 删除



陌生路径：末尾兜底是否生效



目录删除、子 Agent、interrupt、旁路



把最小权限写成可审查配置

稳定路径基线 · 少量人工审批 · 动态策略 · 沙箱隔离

- 先建立权限基线，再按 Agent 职责缩小边界

