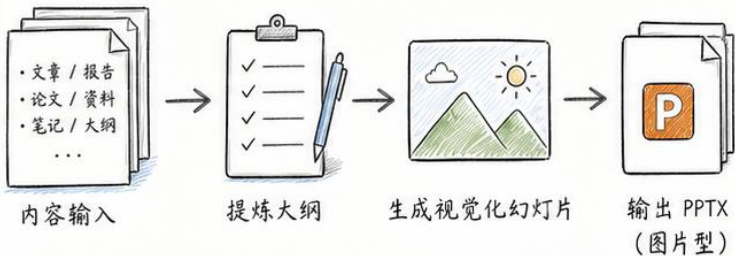


双技能总览：从“生成”到“重建”

两个互补的 PowerPoint 技能，覆盖 PPT 工作流的两端。

技能一：codex-ppt

内容到视觉型 PPT



典型工作流

- ① 理解内容 →
- ② 规划大纲 →
- ③ 选择风格
- ④ 生成样张 →
- ⑤ 批量生成 →
- ⑥ QA 与交付

互补连接

从零开始生成
或从现有页面重建



技能二：image-to-editable-ppt

视觉页面到可编辑 PPT



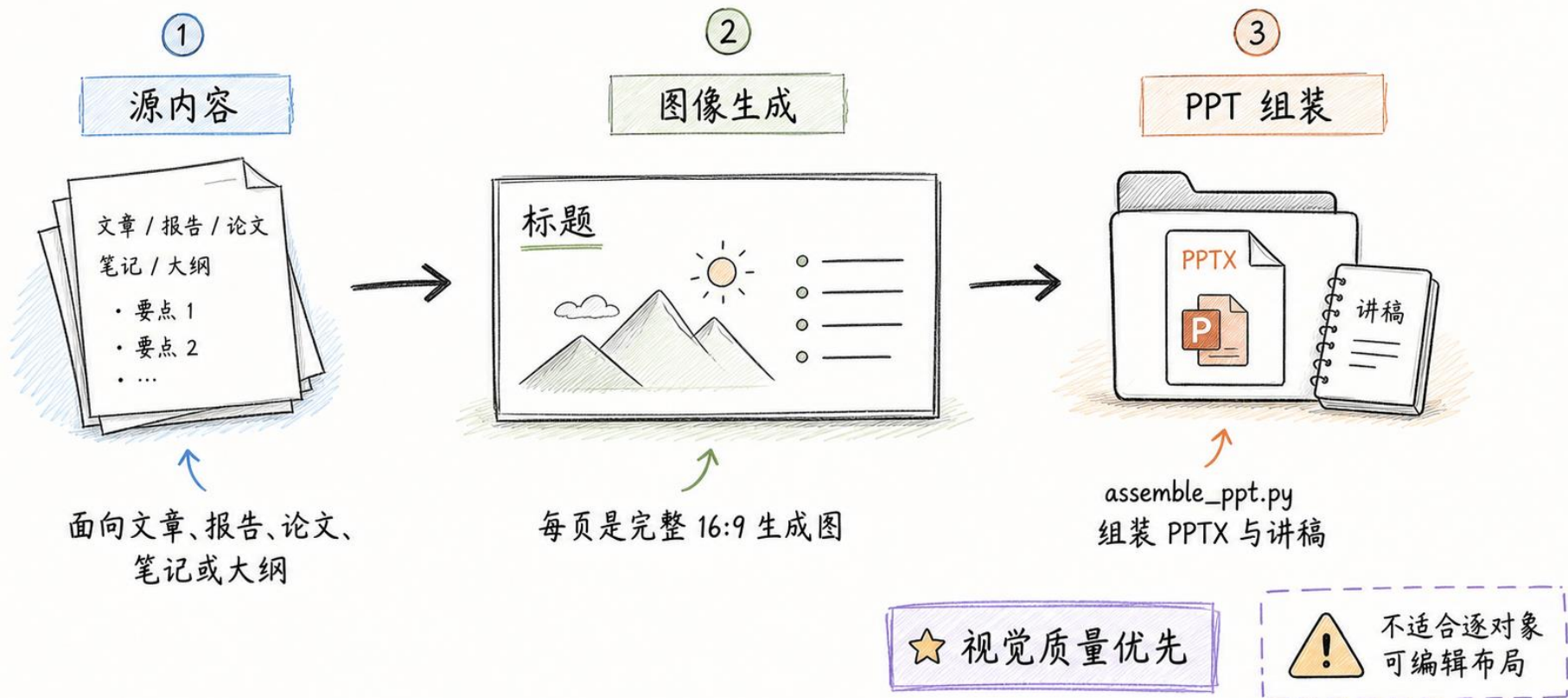
典型工作流

- ① 输入页面 →
- ② 识别解析 →
- ③ 结构重建
- ④ 人工校对 →
- ⑤ 导出可编辑 PPT

要点速览

- ✓ codex-ppt：把内容转化为高质量的视觉化 PPT
- ✓ image-to-editable-ppt：把现有页面重建为可编辑 PPT
- ✓ 覆盖 PowerPoint 工作流的两端，互补协同
- ✓ 前 10 页讲生成，后 10 页讲重建

codex-ppt 的核心定位



codex-ppt 的硬约束



必须

- ✓ 最终 `slide_XX.png` 必须来自选定图片后端
- ✓ 样张通过后可用时必须分派 `slide subagent`
- ✓ 每个 `worker` 只处理一个 `slide job`
- ✓ `dispatch` 和 `result` 状态由脚本记录



禁止

- ~~✗ 本地绘图~~
- ~~✗ Pillow 生成~~
- ~~✗ HTML 截图冒充~~
- ~~✗ 手动拼贴冒充~~

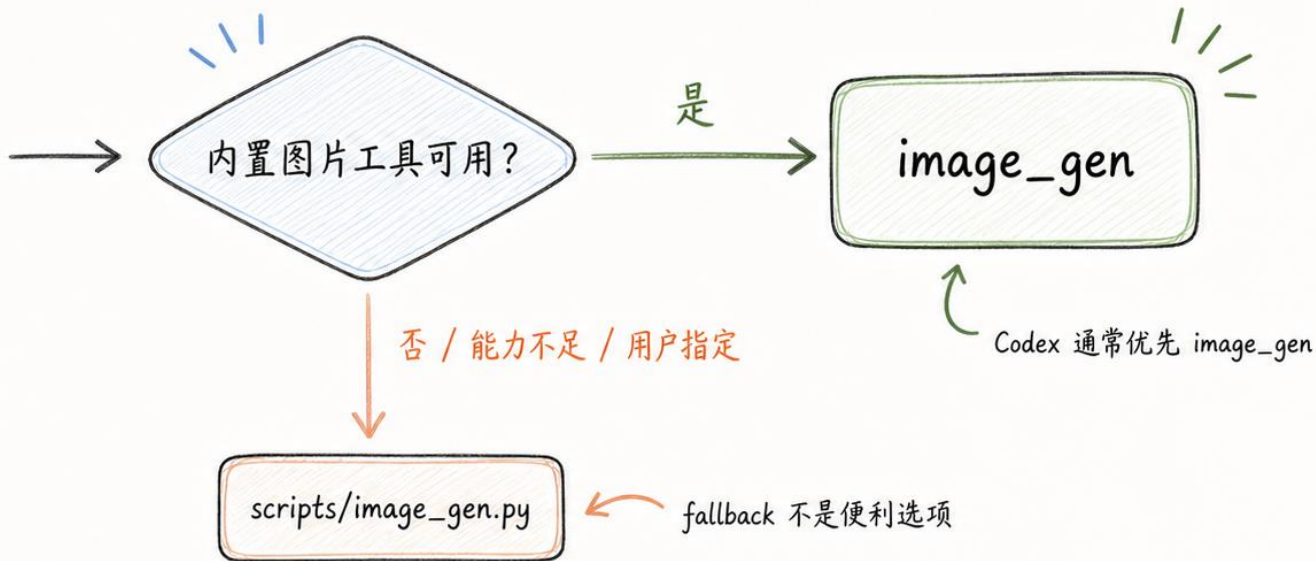


★ 状态以脚本记录为准

图片后端选择逻辑

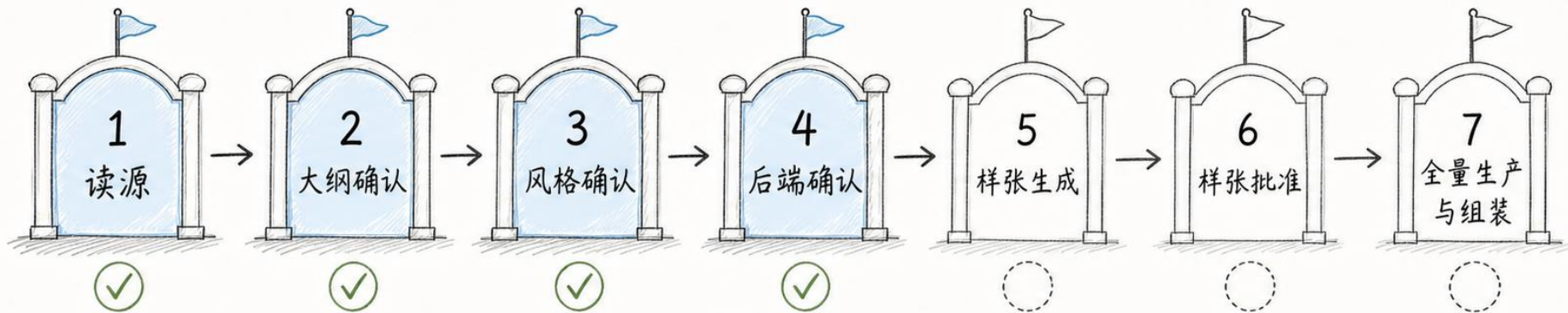


- ✓ 先检查内置图片生成工具
- ✓ Codex 通常优先 image_gen
- ✓ 仅在不可用、能力不足或用户指定时 fallback
- ✓ 后端确认后必须保持一致

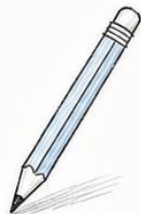


consistency lock: 后端确认后保持一致

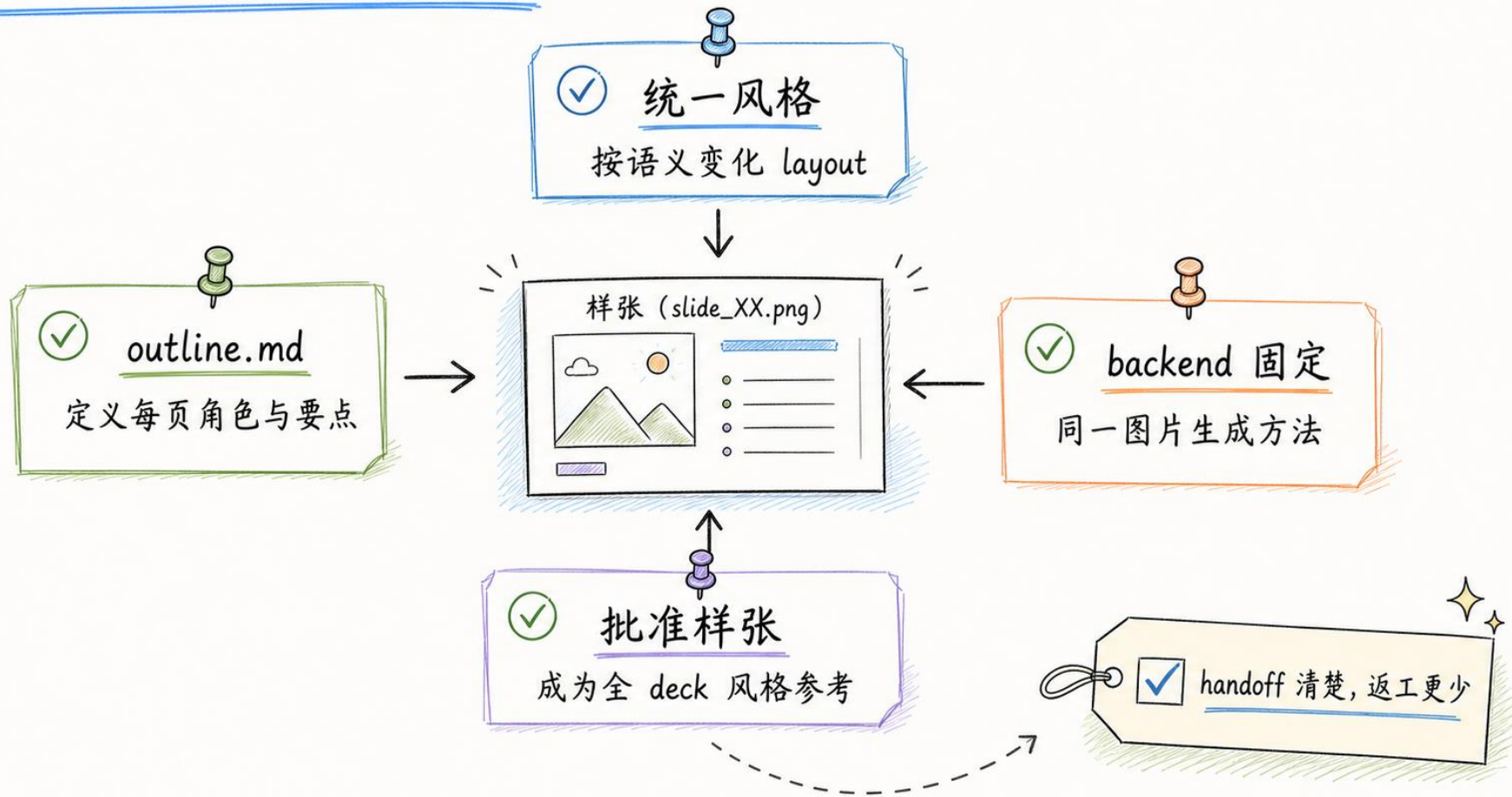
codex-ppt 的阶段门



- 读源内容与资产
- 确认大纲、风格和后端
- 只生成一页样张并等待批准
- 样张通过后才全量生产与组装

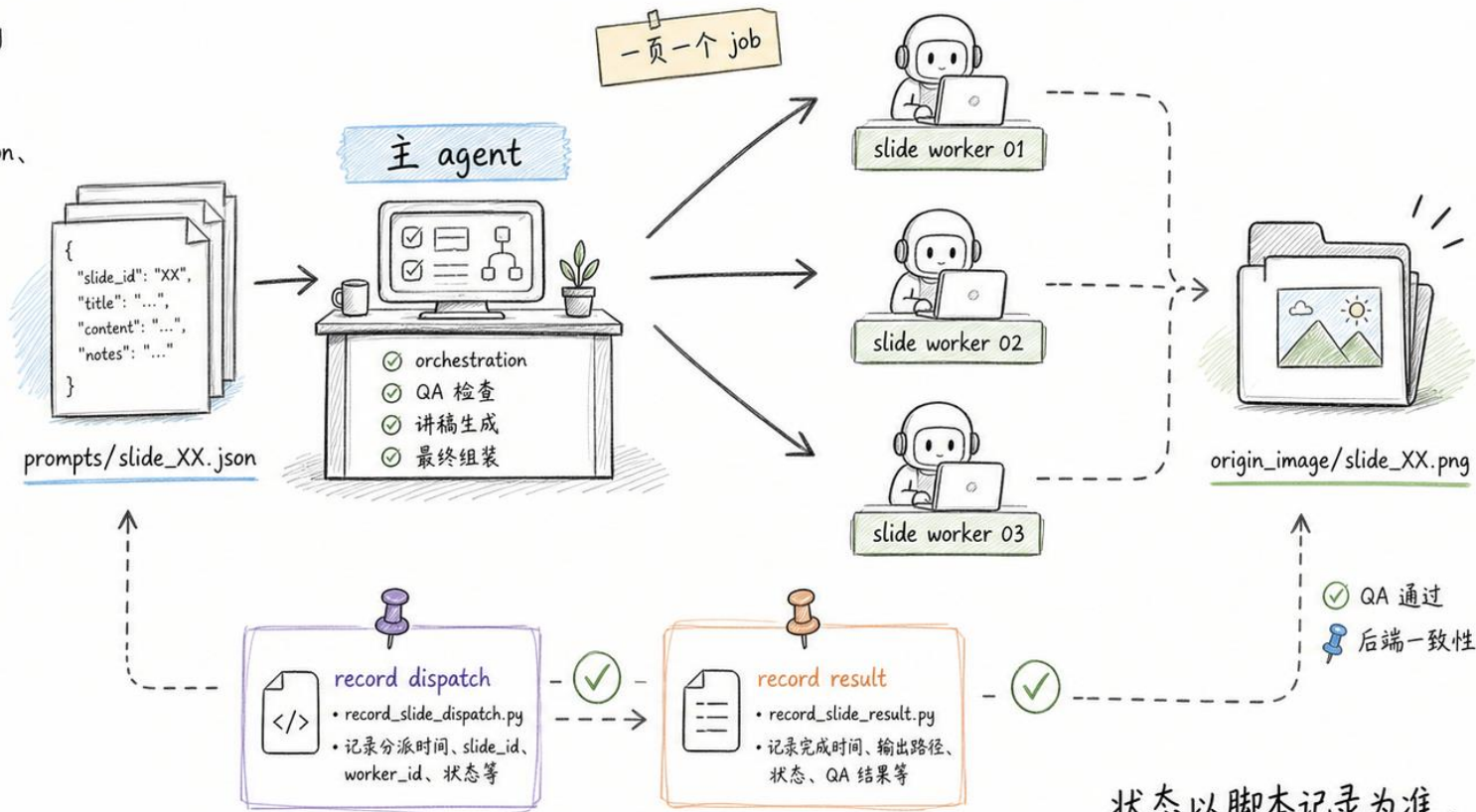


大纲与样张如何控制质量



子任务分派与状态记录

1. prepare_slide_prompts.py 生成每页 JSON job
2. 主 agent 只做 orchestration、QA、讲稿和最终组装
3. 每个 worker 只处理一个 slide job
4. worker 必须使用样张相同的图片生成方法
5. record_slide_dispatch.py 与 record_slide_result.py 是真实状态来源

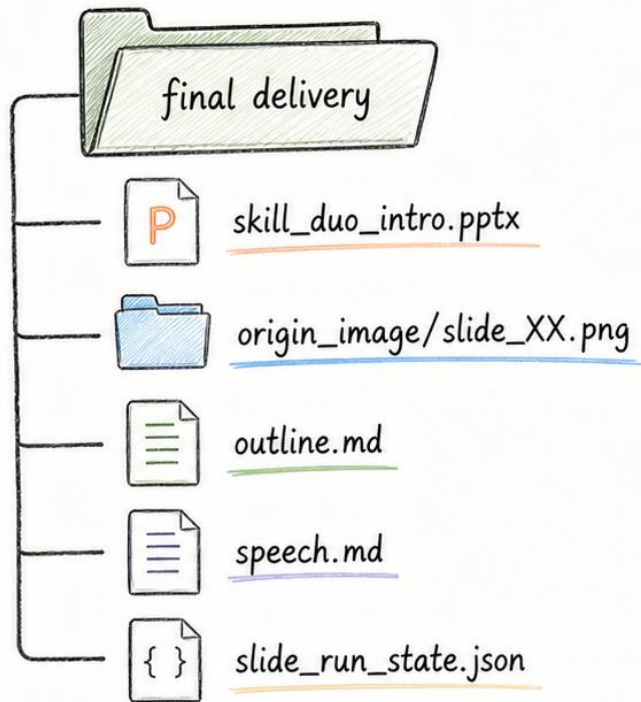


状态以脚本记录为准。

QA 与最终交付物

QA 检查清单

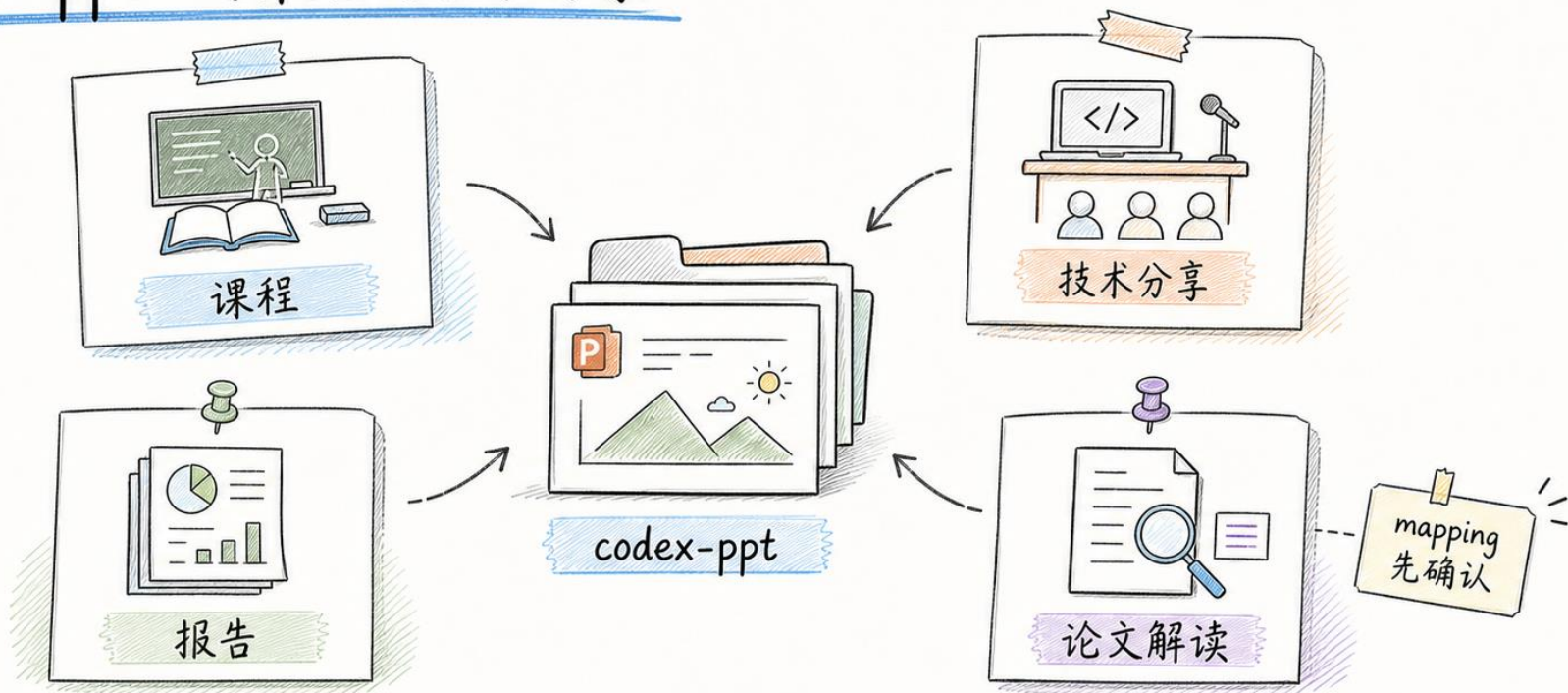
- ✓ 检查文字清晰、中文不乱码
- ✓ 确认内容匹配大纲、元素不重叠
- ✓ speech.md 写入 speaker notes
- ✓ 交付 PPTX、图片、大纲、讲稿和 job 状态



QA 通过 → 组装交付



codex-ppt 的最佳使用场景



✓ 内容明确, 希望快速统一视觉

✓ 适合整页视觉表现

✓ 复杂 source image 需先确认 mapping

✓ 课程、报告、技术分享、论文解读

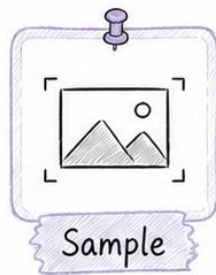
codex-ppt 的一句话总结

内容到视觉型 PPT 的生产线

✓ 强项是统一风格
与整页视觉

✓ 关键风险是
跳过确认或伪造后端

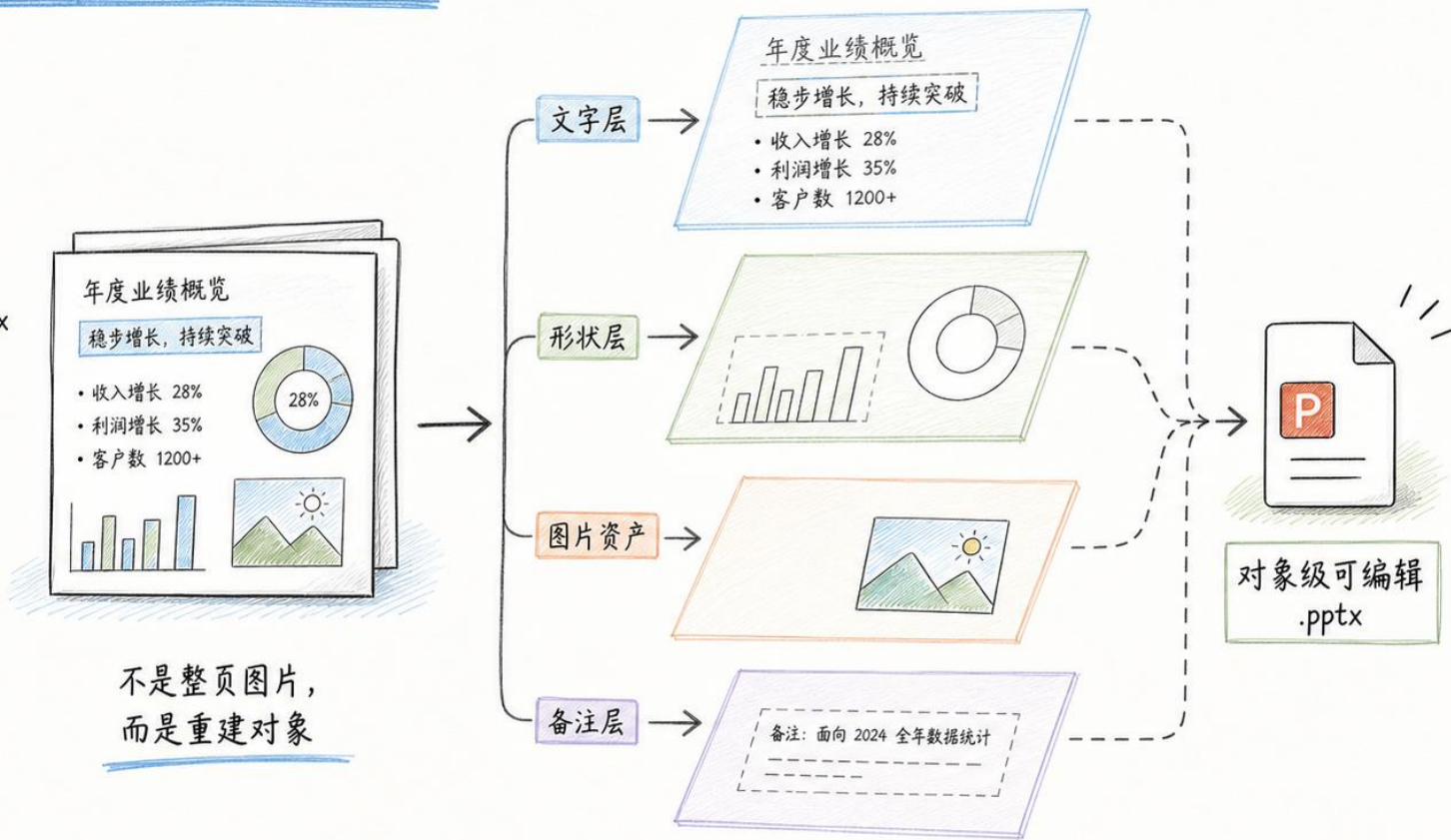
✓ 正确姿势：
大纲、风格、后端、样张



下一部分：
image-to-editable-ppt
 workflow

image-to-editable-ppt 的核心定位

1. 面向图片、PDF、
图片版 PPT/PPTX
2. 输出对象级可编辑 .pptx
3. 可读文字成为原生
text box
4. 复杂视觉元素成为
独立资产



输入与输出边界

支持的输入类型



第 N 页输入
对应第 N 页输出

每次运行生成独立目录



输出交付物



- 支持单图、多图、PDF、图片版 PPT/PPTX
- 第 N 页输入对应第 N 页输出
- speaker notes 按页保留
- 输出 final PPTX、manifest、preview、validation

Notes 保真线

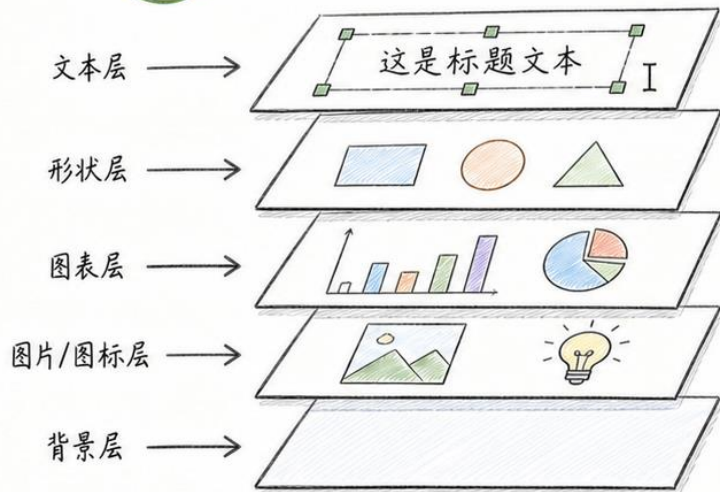
- speaker notes 按页保留
输入第 N 页的 notes → 输出第 N 页的 notes

可编辑率优先的默认取舍

1. 宁可视觉略粗糙，也不要整页 raster 冒充
2. source.png 加文本覆盖是失败模式
3. 隐藏或透明文本不算可编辑文字
4. 非文字复杂对象默认独立资产化



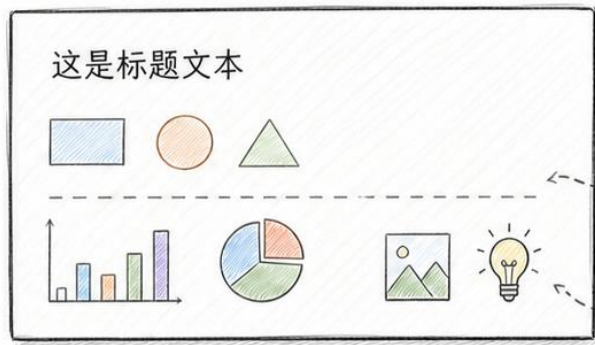
真·可编辑对象层



每一层都是可编辑对象，可调整、可选中



假·可编辑截图层 (不要这样做)



× 整页 raster 截图，不是可编辑对象

× 在上面加文本框覆盖，依然是截图

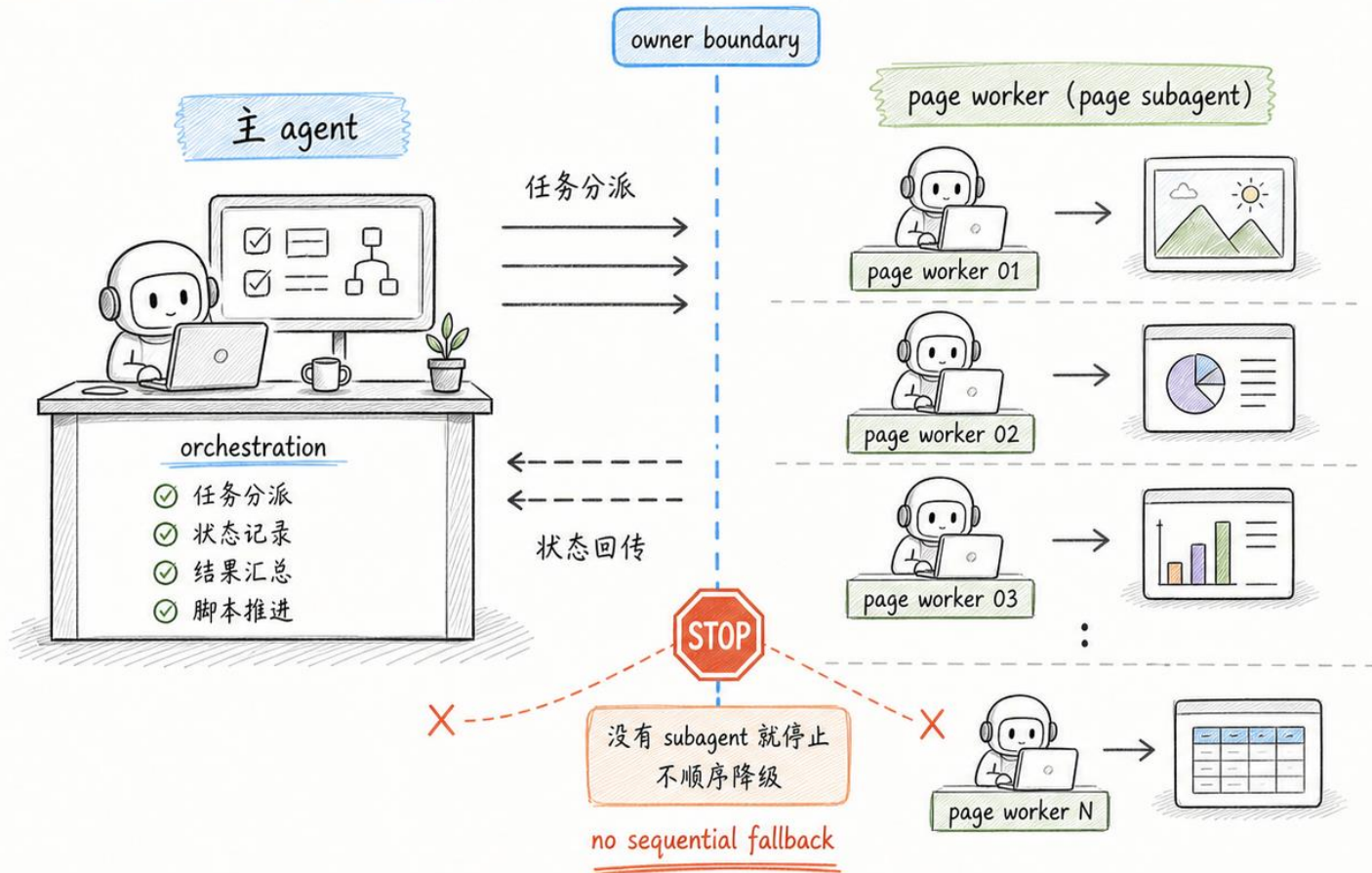
× 隐藏或透明文本不算可编辑文字



目标：真正的可编辑率，而不是“看起来像 PPT”。

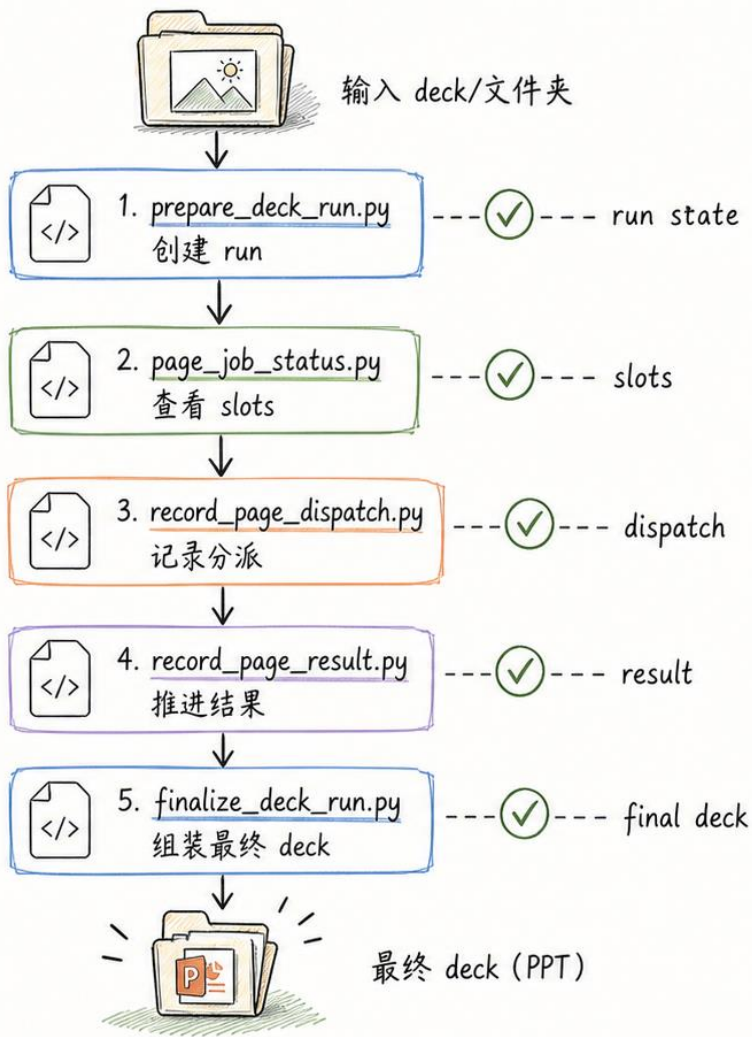
page subagent 是唯一页面重建者

1. 每个来源页面都由 page subagent 重建
2. 主 agent 只做 orchestration
3. 没有 subagent 就停止, 不顺序降级
4. 状态必须由脚本推进

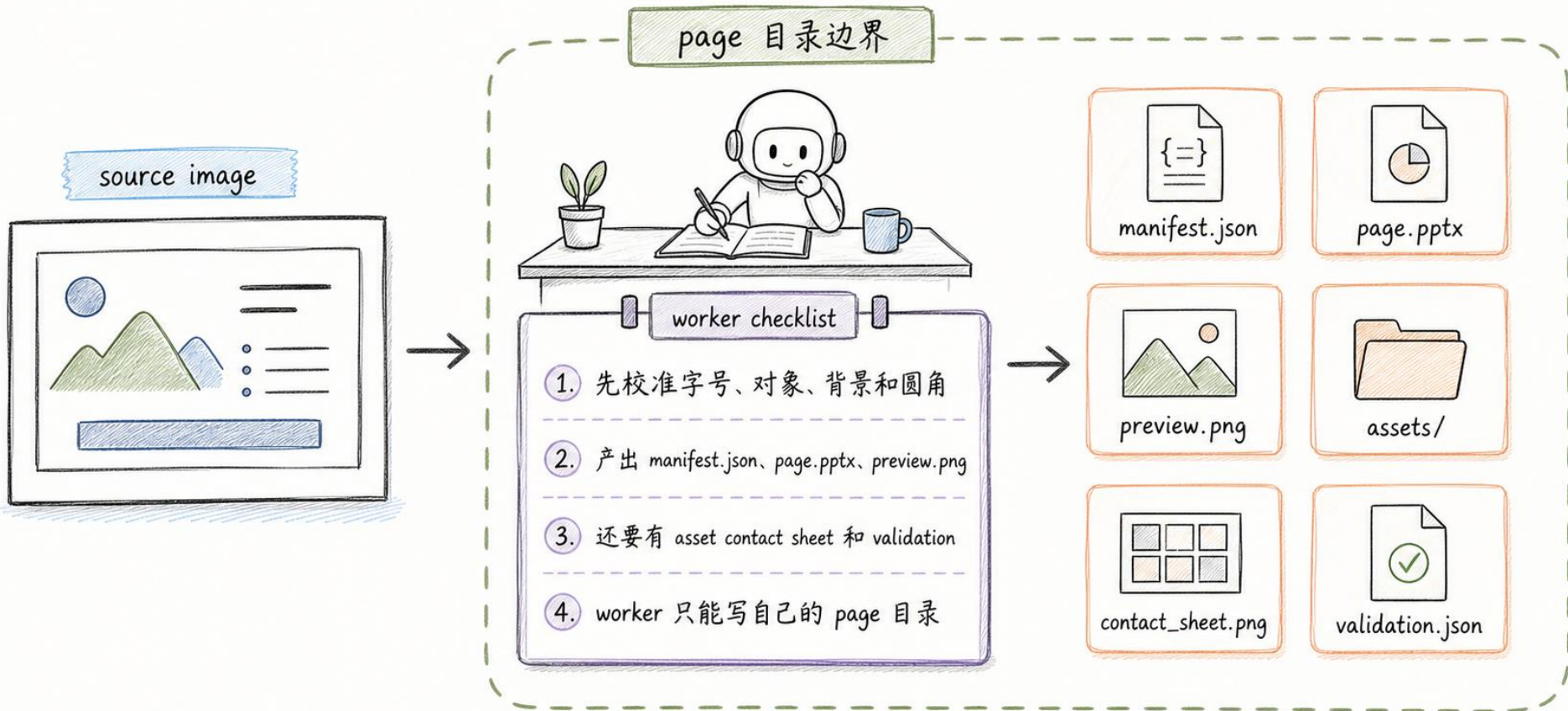


默认运行流程

1. prepare_deck_run.py 创建 run
2. page_job_status.py 查看 slots
3. record_page_dispatch.py 记录分派
4. record_page_result.py 推进结果
5. finalize_deck_run.py 组装最终 deck

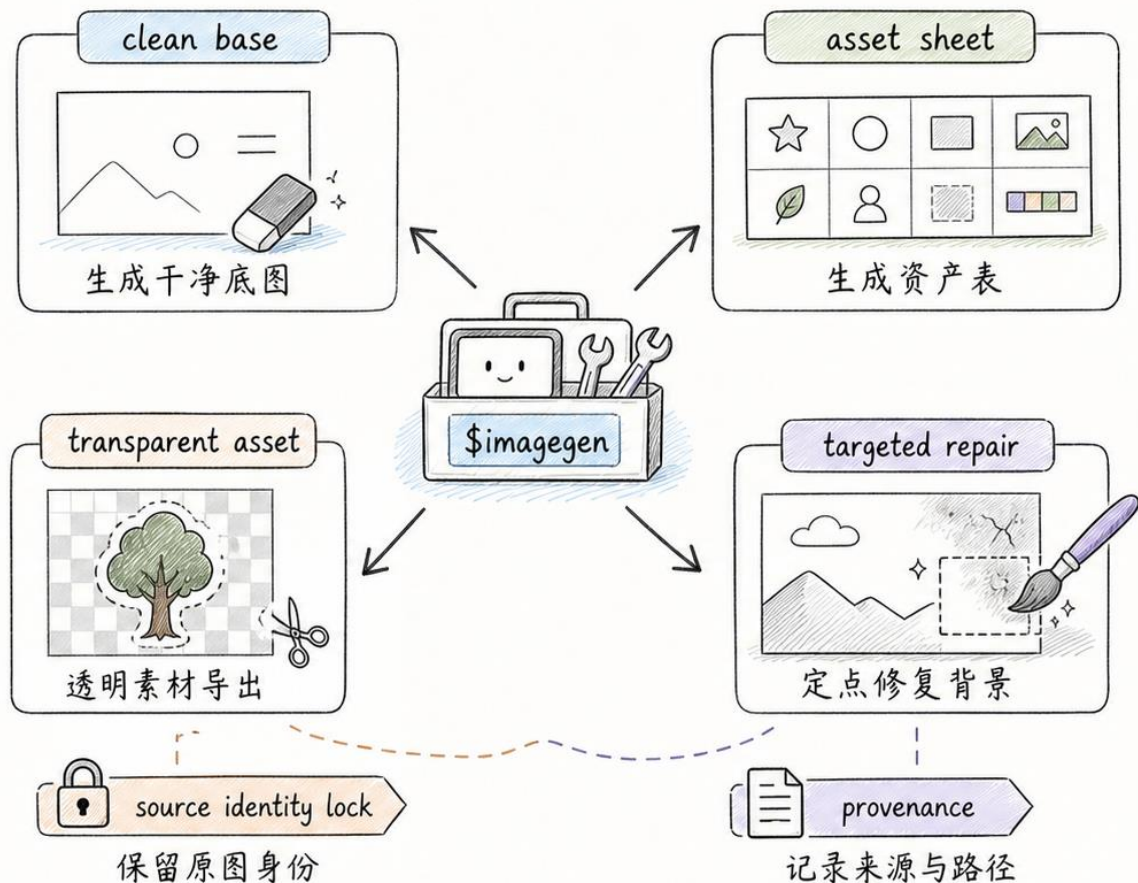


页面 worker 的重建清单



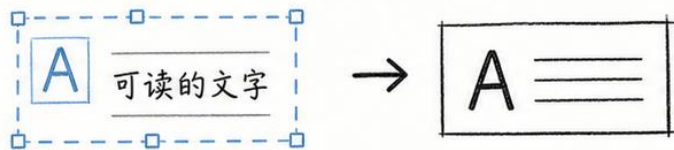
\$imagegen 在重建里的角色

- ✓ 生图、改图、背景修复和 asset sheet 必须用 \$imagegen
- ✓ 默认走 built-in image_gen
- ✓ 复杂背景修复要保留 source identity
- ✓ 生成资产复制到 page 目录并记录 provenance



对象决策规则

1. 文本 Text



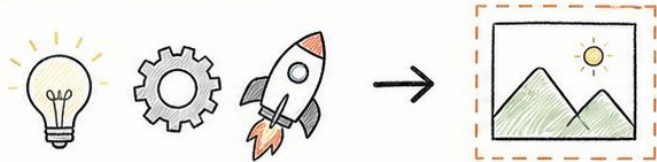
可读文字成为 text box

2. 结构 Shape



基础几何可用原生 shape

3. 前景 Asset



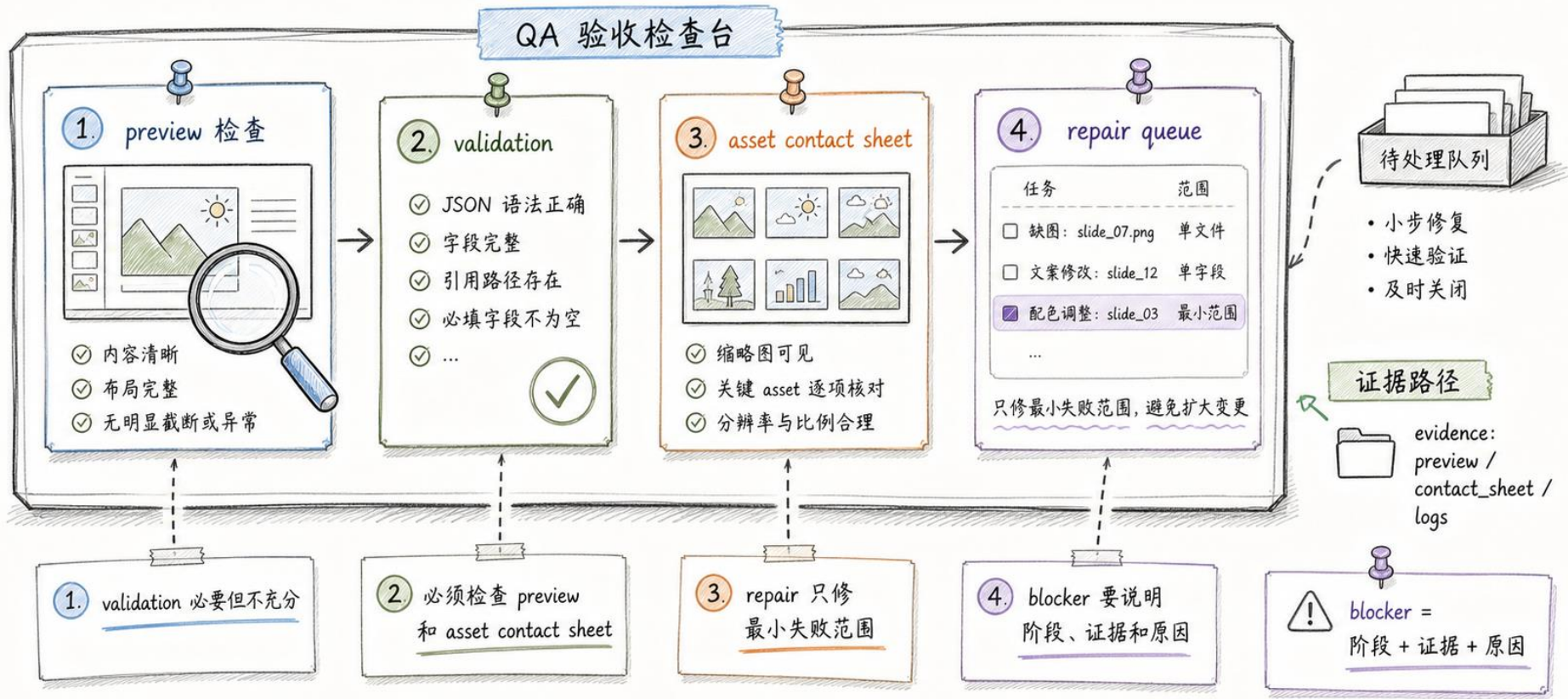
复杂图标默认独立资产

4. 背景 Background



复杂遮挡用 restoration

QA、repair 与验收标准



两个技能如何互补

codex-ppt

- 内容出发
- 生成视觉型 deck
- 重风格一致
- 适合新作

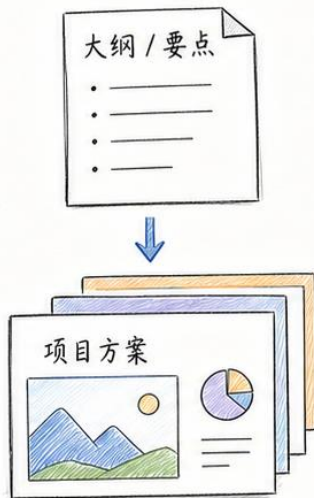
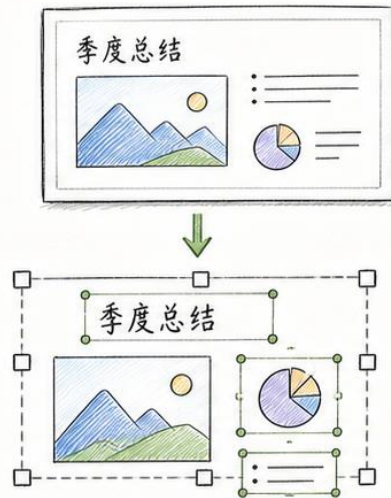


image-to-editable-ppt

- 页面出发
- 重建对象型 deck
- 重可编辑率
- 适合改旧稿



生成与重建，覆盖 PPT workflow 两端。

