

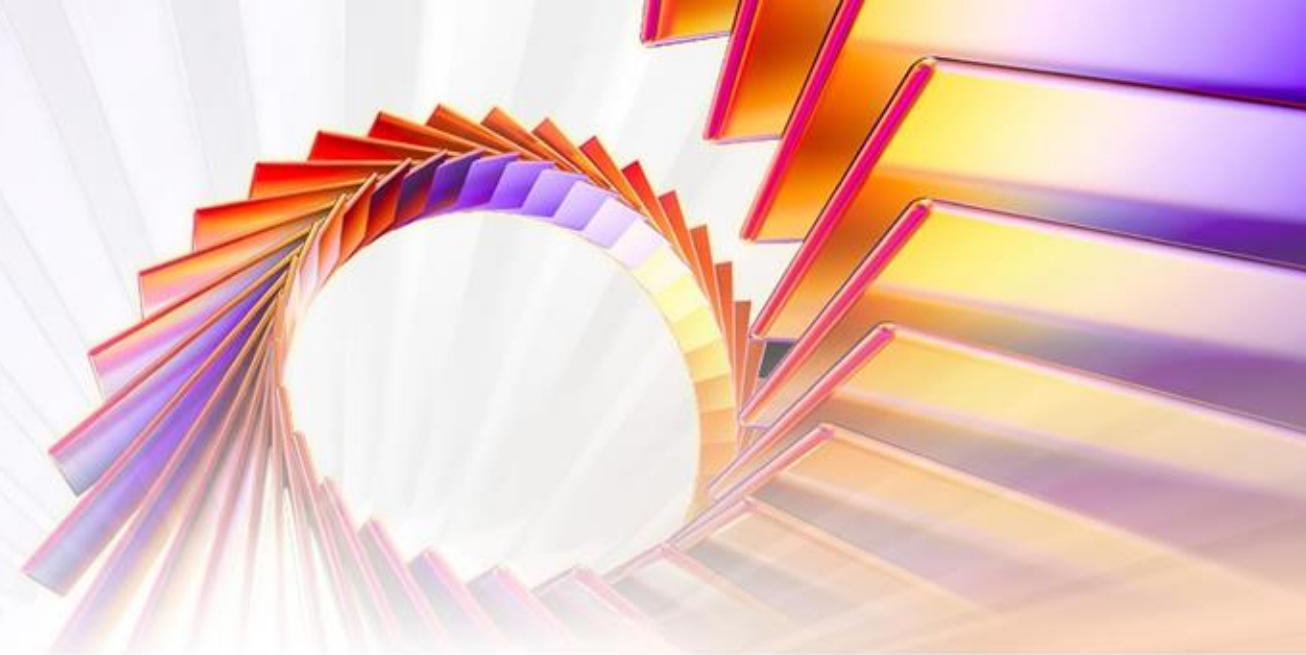
AjDD 8th
2025

| 11月14-15日 | 深圳

第8届 AI+ Development
Digital Summit

AI+研发数字峰会

拥抱AI 重塑研发



用AI Agent与知识库重塑遗留系统迁移范式

张思楚 | /thoughtworks



张思楚

Thoughtworks Technical Principal

海外大项目的离岸交付中心技术负责人,全栈工程师、畅销Web产品架构师、Web专利技术发明人。目前专注于提升软件交付效能,致力于持续集成和持续交付的有效实施,推动适宜的软件效能度量体系的落地。擅长运用AI技术赋能软件开发流程,具备深厚的技术背景,能够引导团队实现AI智能化的开发与运维流程,提升团队整体协作效率和软件质量。

目录

CONTENTS

- I. 背景
- II. 问题/痛点
- III. 解决思路/整体方案
- IV. 具体实现/技术实践
- V. 总结与展望

PART 01

背景

► 您是否也在为企业里的遗留 CMS 发愁？

遗留的 CMS 系统就像一个破败的水桶

到处漏水

已经没办法再装水（内容）



自建 CMS...

遗留系统现代化 = 企业生存关键

- **技术债务**: 积累多年，最终爆发为高成本与商业风险 (Equifax 18亿美元损失)
- **维护困境**: 人才稀缺 + 知识孤岛，一旦流失=系统瘫痪 (每年长15%)
- **安全漏洞**: 缺乏现代标准，合规难保障，数据泄露风险高 (均成本 490万美元)
- **性能低下**: 加载缓慢 → 转化率直降，直接影响收入 (1秒→ 7% 转化率损失)

创新瓶颈：遗留 CMS = 增长阻碍者

- **架构僵化**: 集成困难 → 数据孤岛 & 流程断裂 (45% 集成困难是其CMS首要挑战)
- **渠道局限**: 仅支持桌面，无法覆盖多端 → 重复内容&风险 (2025 移动端 64.35%)
- **缺乏敏捷**: 内容修改依赖开发 → 上市慢，响应慢 (长 35%)

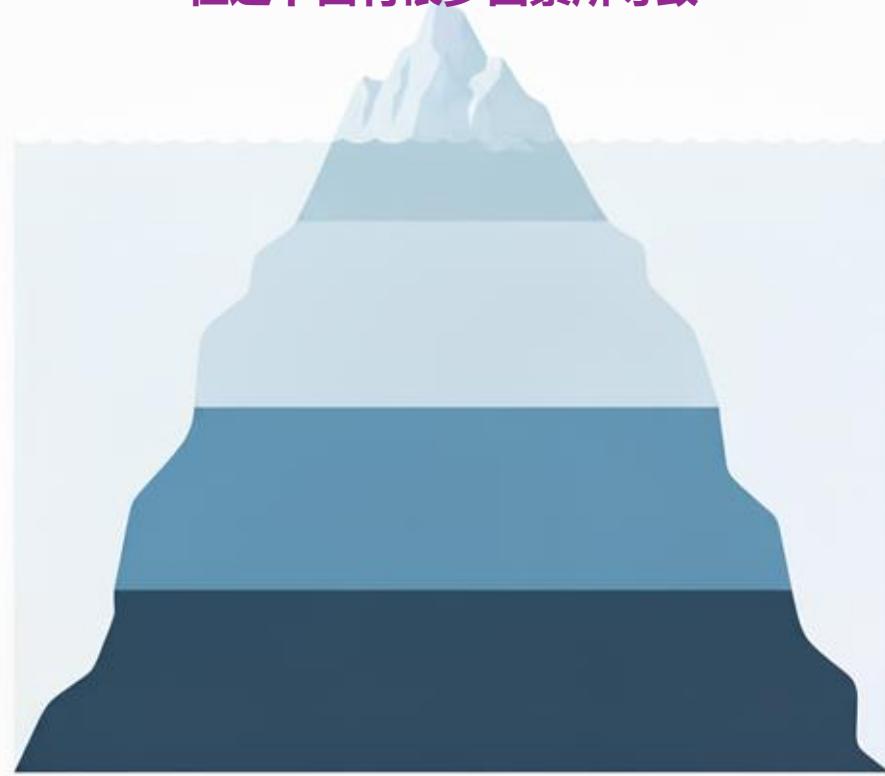
PART 02

问题/痛点 迁移遗留CMS

► 迁移或重建CMS的痛点

根据Gartner等机构的研究，高达 83% 的数据迁移项目要么彻底失败，要么严重超出预算和时间表。项目预算平均超支。

遗留CMS系统的迁移失败只是冰山上的表象
在这下面有很多因素所导致



不可能三角：成本 / 时间 / 质量

- 成本高：逆向工程、复杂数据映射、脚本开发、清洗 (ELT) → 巨大人力投入
- 周期长：动辄 1 年以上，市场已无法接受（平均会延期 30%至41%）
- 质量低：受限时限和预算 → 数据丢失、内容损坏、用户体验差。

常见风险陷阱

- 数据风险：格式/Schema 不兼容 → 数据不完整（40% 的内容过时或重复的、12% 的页面没有任何内部链接 “孤岛页面” ）
- 集成失败：后期发现新CMS与CRM/ERP不兼容
- SEO 重创：URL/元数据缺失，排名一夜清零（一家医疗机构在 48小时内，搜索流量暴跌了 42%，并产生了超过 800个 “404页面未找到” 错误。）

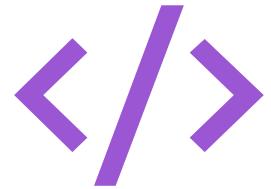


PART 03

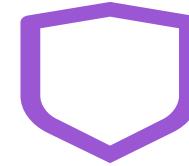
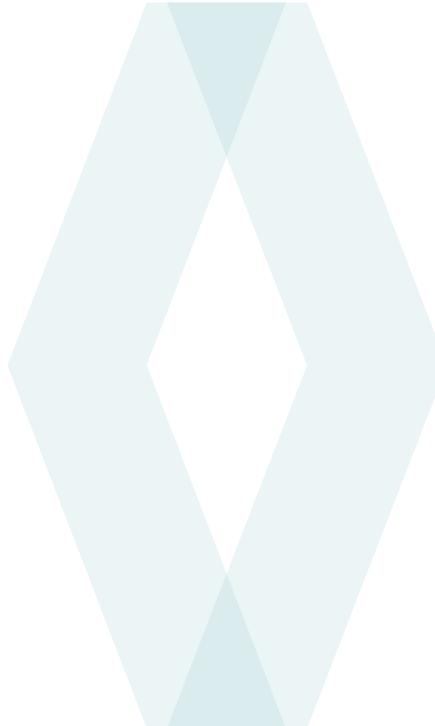
解决思路/整体方案
借助 AI 破局

► 颠动 AI 的两个阶段

AiDD 8th
2025



逆向工程
分析现有系统

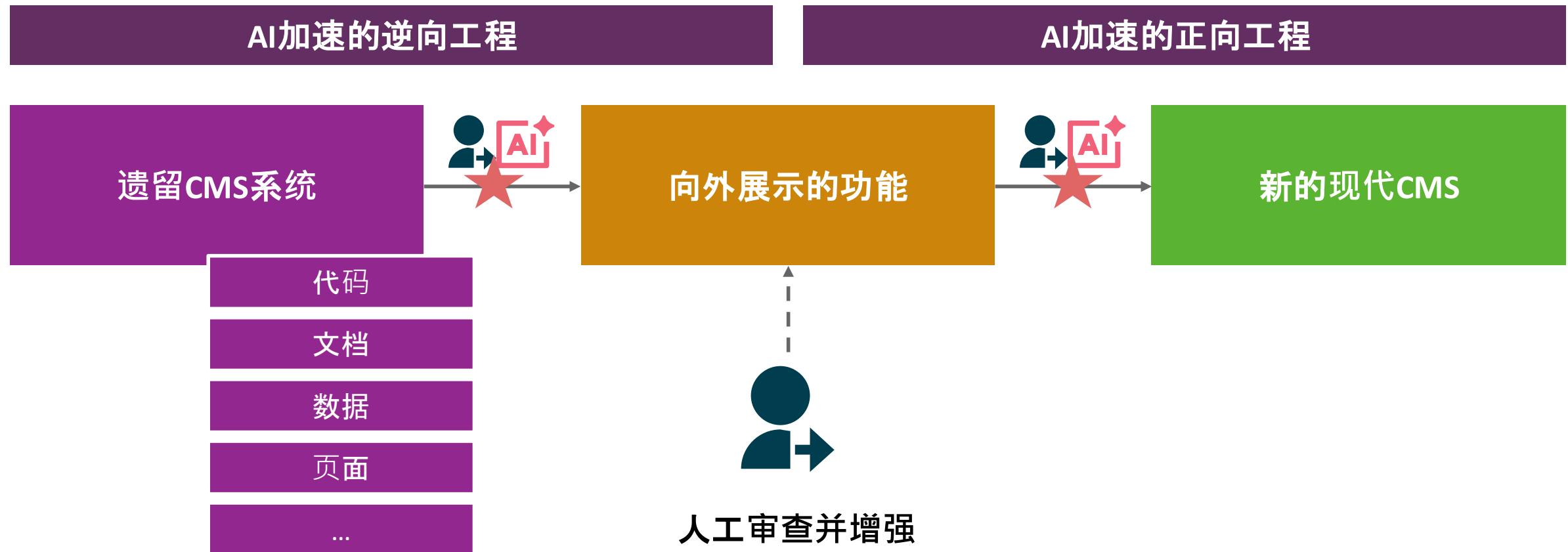


正向工程
功能重写



► 在迁移之前先整理好现有状态

GenAI 它正在以多种不同的形式发挥作用



► 用AI Agent与知识库重塑CMS遗留系统迁移范式

AjDD 8th
2025

20多年的遗留 CMS

外部可见的
展现内容 HTML

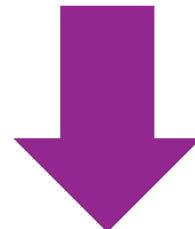
★ AI 赋能
代码结构问题 → 内容处理问题

转换成现代 CMS 可以
直接消费的 Markdown



逐个迁移并完成

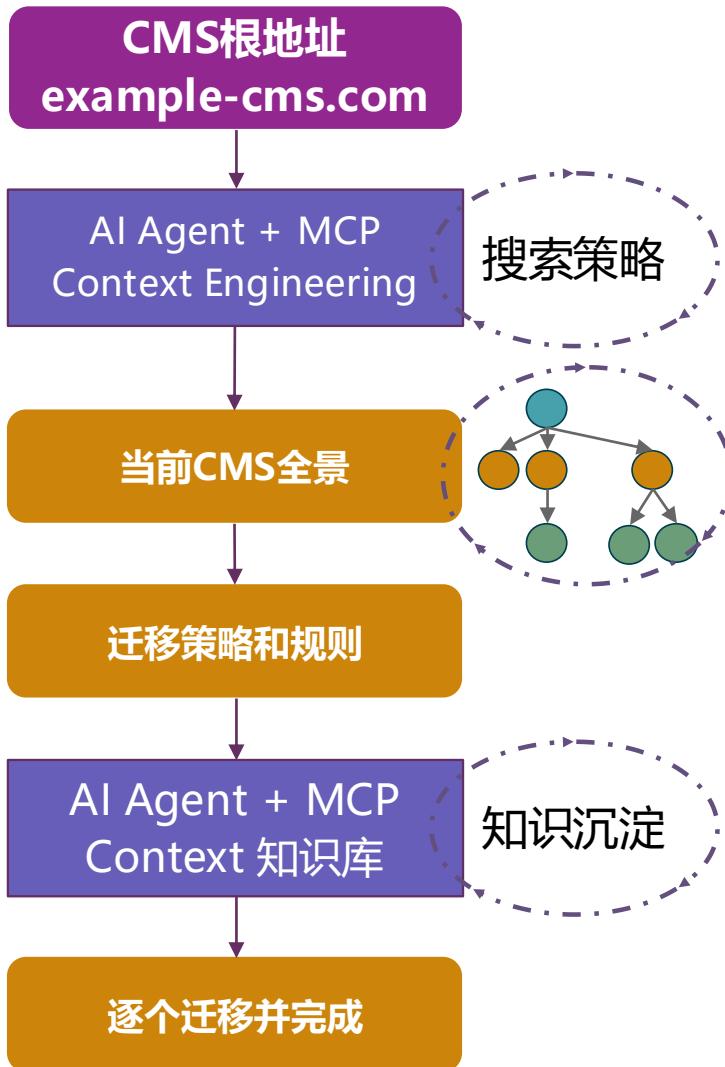
- 运行了20多年的CMS系统，其代码和数据库早已陈旧难懂，传统的ETL迁移几乎不可行，因此被视为“黑盒”。
- 传统做法往往依赖逆向工程去解析内部结构，但代价高、周期长且风险大。



- 不再试图解析黑盒内部，而是从外部可见的展现内容（HTML）入手。
- 借助大语言模型擅长的内容理解与重构能力，将原本难解的“代码结构问题”转化为“内容处理问题”。HTML -> Markdown
- 不仅绕过了底层依赖，更展示了AI在传统工程难题中的独特价值，为企业验证了AI能驱动业务增长的不同可能性。

用AI Agent与知识库重塑CMS遗留系统迁移范式

AjDD 8th
2025



不逆向代码库和数据库，而将重点转移到用户可见的 CMS 页面内容，将“代码库和数据库”问题转化为大语言模型擅长的“内容理解与重构”问题，从而解耦底层代码依赖。

- **第一式-AI当“侦探”：**Agent 协同和MCP调用，自动生成高效的爬取策略，更专注关键内容，将复杂任务分解成可执行的子任务。
- **第二式-可视化编排：**利用 Langflow 的可视化界面，通过拖拽将 Agent 与 MCP，构建稳健、可验证、可维护、可扩展的自动化流水线。
- **第三式-智能“翻译官”：**AI Agent 可理解 DOM 语义，而非简单替换标签，从而生成干净、结构化的 Markdown 内容，并重构关键信息。
- **第四式-经验即规则：**通过构建 Context 知识库，将人工美学和编辑经验（如标题格式、图片处理、关键词高亮）转化为 AI 可执行的语言指令。

成本低、周期短、质量高：

- 迁移周期从一年缩短到四个月，程序转换从 40 周缩短到 10 周，测试时间缩短 85%。
- 70% 的迁移工作由 AI 完成，20% AI 辅助，仅10%需人工手动修改。开发效率提升 3 到 4 倍。
- 将人类智慧转化为持续进化的知识库，确保内容风格和质量保持一致，实现效率和质量增长。

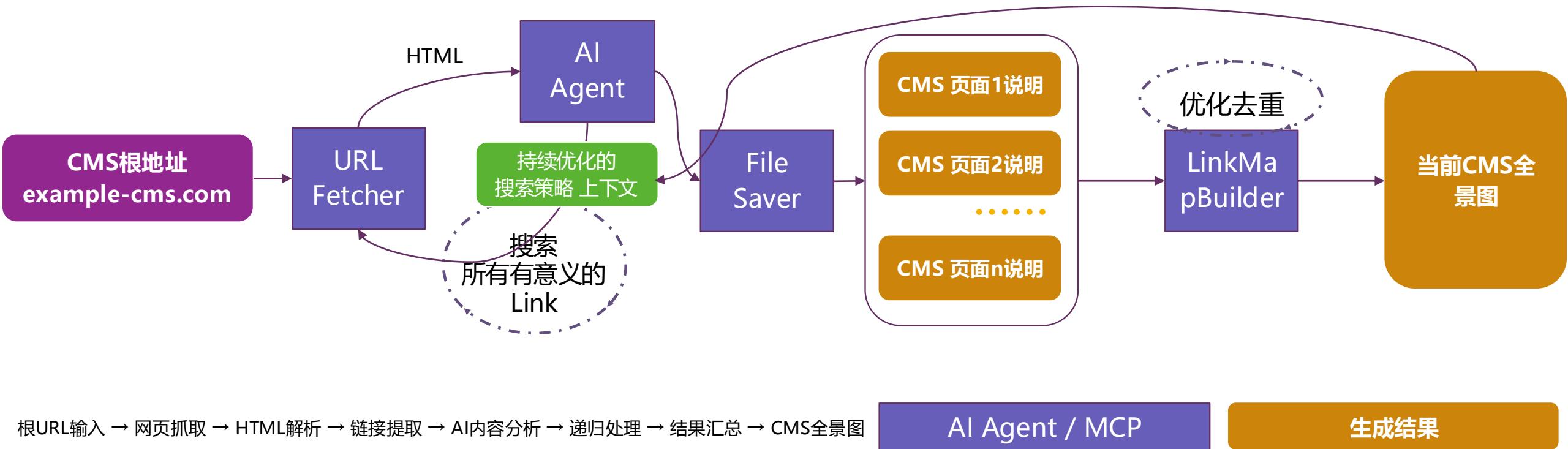
完善的迁移策略保证了数据一致、集成健壮、SEO稳定！

PART 04

具体实现/技术实践 迁移范式

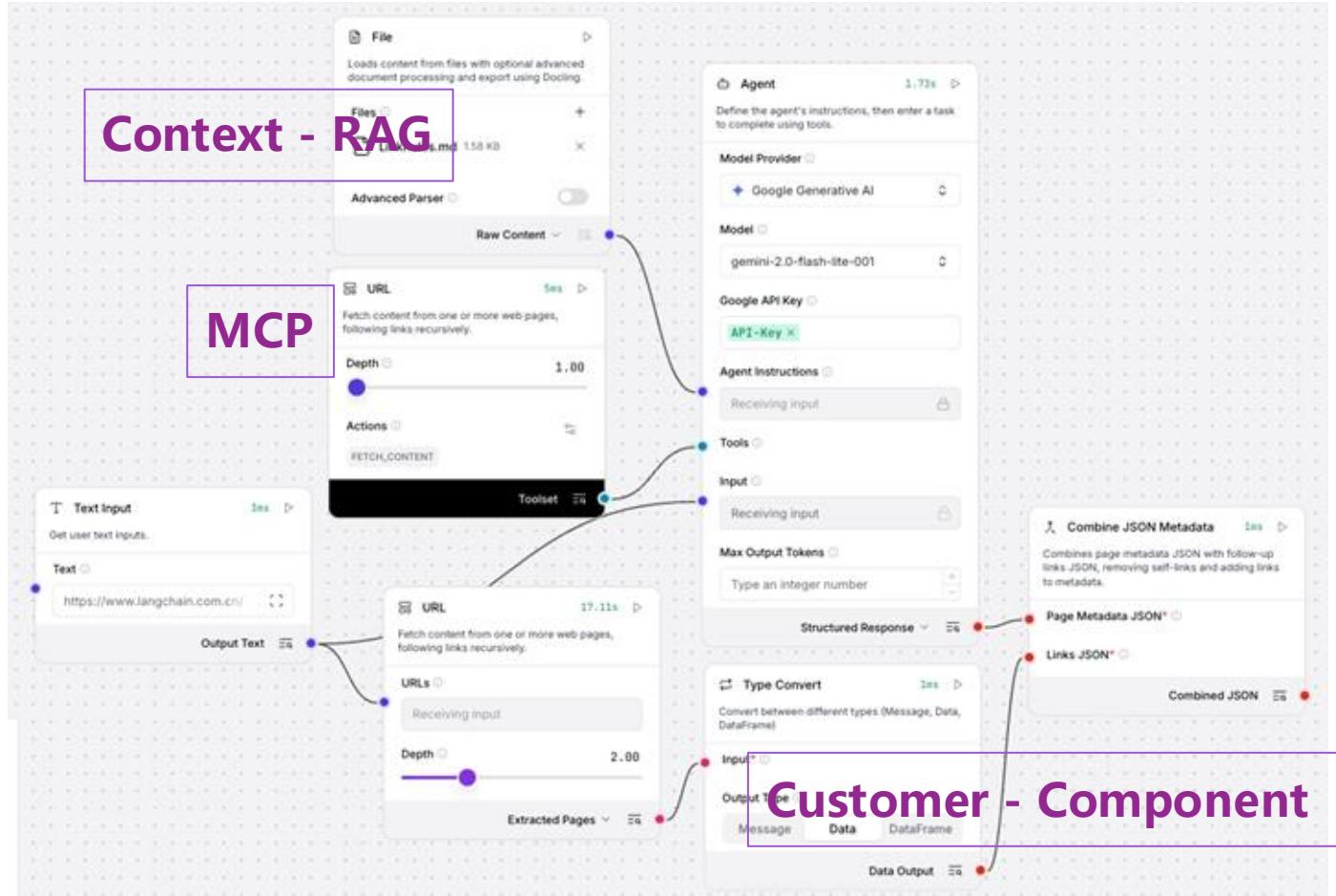
► 第一式 AI 当“侦探”

- Agent 协同和MCP调用，自动生成高效的爬取策略，更专注关键内容，将复杂任务分解成可执行的子任务。



► 第二式 可视化编排

- 利用 Langflow 的可视化界面，通过拖拽将 Agent 与 MCP，构建稳健、可验证、可维护、可扩展的自动化流水线。



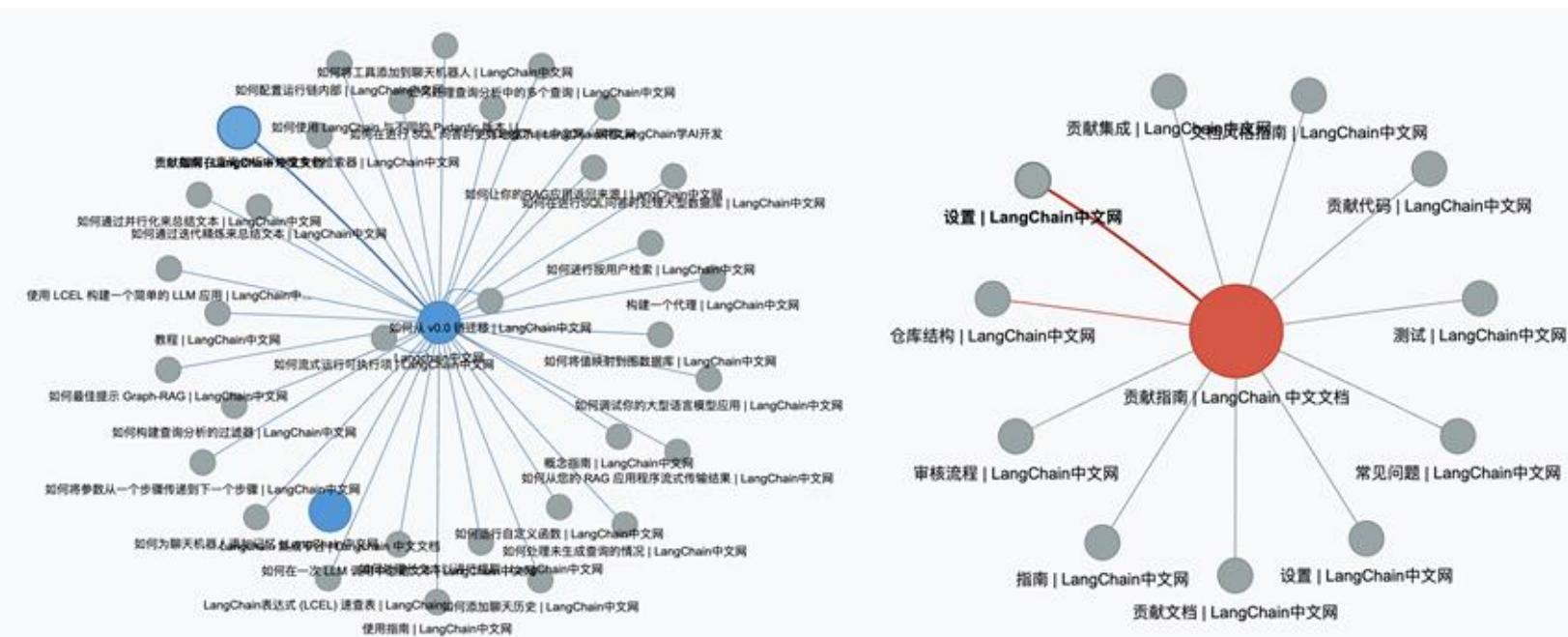
```

text      tree
{
  "url": "https://www.langchain.com.cn/",
  "title": "Langchain中文网",
  "summary": "Langchain中文网，提供Langchain相关的中文文档、教程、示例和社区支持。",
  "category": "其他",
  "type": "静态页面",
  "priority": "高",
  "date": "2023-10-27",
  "keywords": [
    "Langchain",
    "中文",
    "文档",
    "教程",
    "示例"
  ],
  "features": "无",
  "links": [
    {
      "url": "https://www.langchain.com.cn/docs/how_to/query_no_queries/",
      "title": "如何处理未生成查询的情况 | LangChain中文网",
      "description": "有时，查询分析技术可能允许生成任意数量的查询—包括没有查询！在这种情",
      "content_type": "text/html",
      "language": "en"
    },
    {
      "url": "https://www.langchain.com.cn/docs/how_to/summarize_stuff/",
      "title": "如何在一次 LLM 调用中总结文本 | LangChain中文网",
      "description": "大型语言模型可以从文本中总结和提炼所需的信息，包括大量文本。在许多情",
      "content_type": "text/html",
      "language": "en"
    }
  ]
}

```

这里以 www.langchain.com.cn 为示例来迁移

► 第二式 可视化编排



节点信息编辑

选择要编辑的节点:

如何处理未生成查询的情况 | LangChain中文网

编辑节点: 如何处理未生成查询的情况 | LangChain中文网

基本信息:

可编辑字段:

title: 如何处理未生成查询的情况 | LangChain中文网

用户备注:

description: 有时，查询分析技术可能允许生成任意数量的查询——包括没有查询！在这种情况下，我们的整体链需要检查查询分析的结果，然后决定是否调用检索器。

优化图片，删除链接。

summary: 如何处理未生成查询的情况，说明页面。

自定义标签(用逗号分隔):

img_improve; links_remove

category: 错误处理。

保存修改

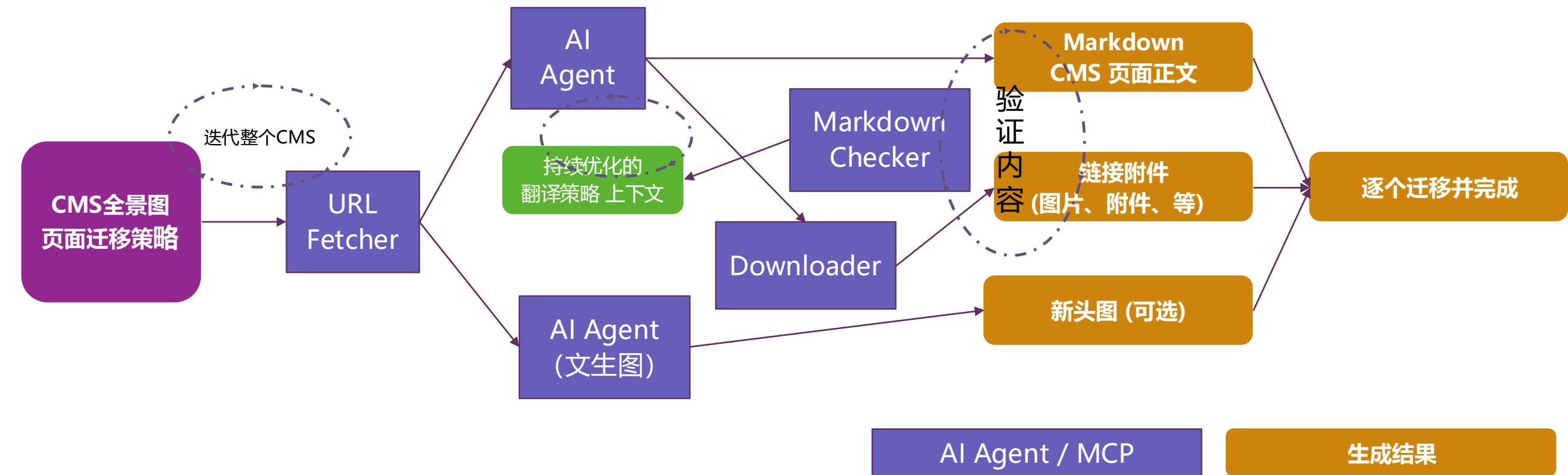
type: 文档节点。

这里以 www.langchain.com.cn 为示例来迁移

- 迭代式的审查、更新页面转换策略，会提升 CMS 全景图的信息质量。
- 一次性递归、大爆炸的方式、不能充分利用 AI 的能力，阶段性的、迭代式的将 CMS 全景图作为上下文送给 Agent 会得到更好的分析结果。
- 人工审查、人工标注、并预设迁移策略，能有效保证迁移成果。
- 让AI做AI擅长的事情（页面分析），让程序做程序擅长的事情（Url整合）。

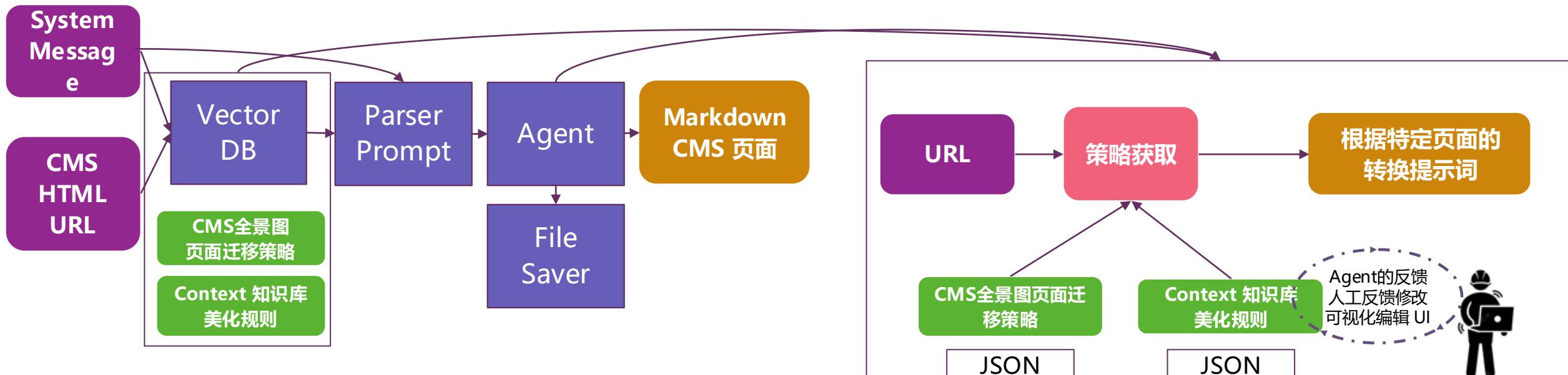
► 第三式 智能“翻译官”

- AI Agent 可理解 DOM 语义，而非简单替换标签，从而生成干净、结构化的 Markdown 内容，并重构关键信息。



► 第四式 经验即规则

- 通过构建 Context 知识库，将人工编辑的美学经验转化为可执行规则（如：标题使用主动语态并含疑问句、正文中产品名称自动加粗、图片调整到合适宽度并居中等），实现自动化“美化”。



AI Agent / MCP

生成结果

预处理脚本

PART 05

总结与展望

► 案例总结

成本低、周期短、质量高：

- 迁移周期从一年缩短到四个月，程序转换从 40 周缩短到 10 周，测试时间缩短 85%。
- 70% 的迁移工作由 AI 完成，20% AI 辅助，仅10%需人工手动修改。开发效率提升 3 到 4 倍。
- 将人类智慧转化为持续进化的知识库，确保内容风格和质量保持一致，实现效率和质量增长。

完善的迁移策略保证了数据一致，API Gateway、绞杀者架构设计、集成健壮、SEO 稳定！

```
HANDLE ptrSpdUpDown;
HANDLE ptrUpDownDraw;
HANDLE ptrDrawSpd; //申请指向信号量的句柄
ptrSpdUpDown = ::CreateSemaphore(NULL, 0, 1, NULL);
ptrUpDownDraw = ::CreateSemaphore(NULL, 0, 1, NULL);
ptrDrawSpd = ::CreateSemaphore(NULL, 1, 1, NULL); //实例化三个信号量
bOnOff=true;//进程状态控制量 开启三个线程
m_tWriteSpeed=AfxBeginThread(WriteSpeed,
&m_sWriteSpeed,
THREAD_PRIORITY_NORMAL,
0,
CREATE_SUSPENDED);
m_tWriteSpeed->ResumeThread();
m_tWriteUpDown=AfxBeginThread(WriteUpDown,
&m_sWriteUpDown,
THREAD_PRIORITY_NORMAL,
0,
CREATE_SUSPENDED);
m_tWriteUpDown->ResumeThread();
m_tDraWindow=AfxBeginThread(DrawWindow,
&m_sDrawWindow,
THREAD_PRIORITY_NORMAL,
0,
CREATE_SUSPENDED);
m_tDraWindow->ResumeThread();
```

格式优化



```
1 HANDLE ptrSpdUpDown;
2 HANDLE ptrUpDownDraw;
3 HANDLE ptrDrawSpd; //申请指向信号量的句柄
4 ptrSpdUpDown = ::CreateSemaphore(NULL, 0, 1, NULL);
5 ptrUpDownDraw = ::CreateSemaphore(NULL, 0, 1, NULL);
6 ptrDrawSpd = ::CreateSemaphore(NULL, 1, 1, NULL); //实例化三个信号量
7
8 bOnOff=true; //进程状态控制量 开启三个线程
9 m_tWriteSpeed=AfxBeginThread(WriteSpeed, &m_sWriteSpeed, THREAD_PRIORITY_NORMAL, 0, CREATE_SUSPENDED);
10 m_tWriteSpeed->ResumeThread();
11 m_tWriteUpDown=AfxBeginThread(WriteUpDown, &m_sWriteUpDown, THREAD_PRIORITY_NORMAL, 0, CREATE_SUSPENDED);
12 m_tWriteUpDown->ResumeThread();
13 m_tDraWindow=AfxBeginThread(DrawWindow, &m_sDrawWindow, THREAD_PRIORITY_NORMAL, 0, CREATE_SUSPENDED);
14 m_tDraWindow->ResumeThread();
```

根据内容生成文章头图



- 产品名称自动加粗
- 图片调整到合适宽度并居中
- 自动根据文章内容建议文章分类
- 站外引用链接使用斜体字
- 自动优化表格列宽
-

► 总结与展望

- 让程序专注程序需要解决的事情，让 AI 专注 AI 擅长的事情。
- RAG/Context Engineering 是实践，需要根据自己的场景灵活定制。
- 为 AI 反馈设计友好的人工介入路径，将事半功倍。
- AI 辅助并加速迁移，但是必要的架构设计如：绞杀者模式依然重要。
- 需求一次性讲清楚比后续不断对话更有效。
- 充分利用 AI 的特长，会改变开发流程，导致团队结构变化。“小团队”成为可能。





第8届 AI+研发数字峰会

AI+ Development Digital Summit

感谢聆听！

扫码领取会议PPT资料

