

YDS-Lab标准目录结构适配Trae方案的修改分析报告

一、核心前提：YDS-Lab标准目录结构核心逻辑回顾

基于前期沟通，YDS-Lab标准目录结构以“智能体定义-部门工作区-工具链-项目管理”为核心框架，原始结构简化如下：

Code block

1

YDS-Lab/

2

|— Struc/

核心结构目录

3

| |— Agents/

智能体定义目录（按部门角色划分）

4

| | |— ceo/

总经理智能体（人设、职责、工具配置）

5

| | |— marketing/

企划部智能体

6

| | |— tech/

技术部智能体

7

| | |— ...（其他部门）

8

| |— Departments/

部门工作区（数据存储与协作）

9

| | |— GeneralOffice/

总经办文档库

10

| | |— MarketingDept/

企划部工作区

11

| | |— ...（其他部门）

12

| |— Tools/

工具链目录（按功能划分）

13

| | |— integration/

跨部门集成工具

14

| | |— security/

安全工具

15

| | |— ...

16

|— Projects/

项目管理目录（按项目编号划分）

17

| |— 001-dewatermark-ai/

示例项目：AI去水印工具

18

| |— ...

19

|— Models/

模型配置目录

20

|— config/

模型参数与适配配置

核心逻辑：按“角色定义-数据存储-工具支撑-项目落地”分层，实现智能体与现实部门的精准映射，支持全流程协作。

二、适配性分析：现有目录结构与Trae方案的匹配度

结合当前Trae适配版方案（智能体配置、协作机制、业务流程），现有目录结构的匹配度及问题如下：

2.1 高匹配度模块（无需修改）

目录模块	匹配的Trae方案核心需求	适配依据
Struc/Agents/	各部门智能体自定义配置（人设、职责、工具集）	方案3.2节“各部门智能体详细配置”中，总经理、前端开发、财务总监等智能体的配置逻辑，与该目录“按角色建子目录”的设计完全一致
Struc/Departments/	Trae共享工作区的数据存储与协作需求	方案3.3节“共享工作区”用于存储各部门输出（如企划部调研报告、技术部代码），与该目录“部门工作区”的定位完全匹配
Projects/	项目交付流程的全周期管理	方案4.2节“项目交付流程”中，从研发到交付的全环节输出（代码、测试报告、交付包），可直接按该目录的项目编号规则存储

2.2 需优化模块（适配Trae特性）

目录模块	现有问题（与Trae方案不匹配点）	优化方向
Struc/Tools/	1. 未体现Trae核心的“MCP协议工具集成”特性；2. 未区分“智能体专属工具”与“公共工具”，与方案3.2节“各智能体工具配置”逻辑脱节	1. 新增“MCP/”子目录，存储MCP协议适配的工具（如Figma MCP、GitHub MCP）；2. 在各部门工具目录下新增“exclusive/”（专属工具）和“public/”（公共工具）
Struc/Agents/[部门]/	现有仅含“define.py”等配置文件，未体现Trae智能体的“触发规则”“输出规范”等核心配置（方案3.2节明确要求）	在各智能体子目录下新增“trigger.yaml”（触发规则配置）和“output_spec.yaml”（输出规范），适配Trae的自动化触发需求
Models/	未区分“Trae轻量模型”与“核心模型”，与方案5.2节“智能体能力优化”中“核心智能体用高性能模型”的逻辑不符	新增“lightweight/”（轻量模型，用于非核心任务）和“core/”（核心模型，如GPT-4 Turbo）子目录，适配不同智能体的模型需求

2.3 需新增模块（补全Trae协作机制）

现有目录缺少支撑Trae “智能体协作流程” 和 “安全管控” 的专属目录，需新增以下模块：

- 1. **Struc/Workflow/**：存储Trae流程自动化引擎的配置文件（如方案4.2节 “项目交付流程” 的流转规则），子目录按流程类型划分（如 “project_delivery/” “operation_promotion/” ）
- 2. **Struc/Security/**：独立于Tools目录，存储Trae数据安全相关配置（如方案5.1节的 “AES加密配置” “权限控制清单” ），避免与工具配置混淆
- 3. **Struc/Log/**：存储智能体协作日志（如方案3.3节 “冲突管控” 的操作日志、方案4.3节 “流程优化” 的数据分析原始数据），支撑迭代优化

三、最终优化后目录结构（Trae适配版）

Code block

```
1  YDS-Lab/
2  |—— Struc/                                # 核心结构目录（优化后）
3  |   |—— Agents/                          # 智能体定义目录（新增配置文件）
4  |   |   |—— ceo/                        # 总经理智能体
5  |   |   |   |—— define.py              # 人设与职责配置
6  |   |   |   |—— prompt.py             # 提示词配置
7  |   |   |   |—— trigger.yaml          # 触发规则（如1小时内拆解任务）
8  |   |   |   |—— output_spec.yaml      # 输出规范（如决策报告格式）
9  |   |   |—— marketing/                  # 企划部智能体（同上述结构）
10 |   |   |—— tech/                      # 技术部智能体（同上述结构）
11 |   |   |—— ...（其他部门）
12 |   |—— Departments/                   # 部门工作区（未修改）
13 |   |   |—— GeneralOffice/             # 总经办文档库
14 |   |   |—— MarketingDept/            # 企划部工作区
15 |   |   |—— ...（其他部门）
16 |   |—— Tools/                         # 工具链目录（新增MCP适配）
17 |   |   |—— MCP/                      # MCP协议工具集成
18 |   |   |   |—— figma_mcp/            # Figma MCP配置
19 |   |   |   |—— github_mcp/          # GitHub MCP配置
20 |   |   |—— integration/              # 跨部门集成工具
21 |   |   |—— security/                 # 基础安全工具
22 |   |   |—— [部门]/                   # 部门工具（分专属/公共）
23 |   |   |   |—— exclusive/            # 部门专属工具（如技术部的编译工具）
24 |   |   |   |—— public/              # 公共工具（如通用报表工具）
25 |   |—— Workflow/                     # 新增：流程自动化配置
26 |   |   |—— project_delivery/         # 项目交付流程规则
27 |   |   |—— operation_promotion/     # 运营推广流程规则
28 |   |—— Security/                     # 新增：数据安全配置
29 |   |   |—— encryption.yaml          # 加密规则配置（AES-256）
```

```
30 | | └─ permission.yaml # 权限控制清单
31 | | └─ Log/           # 新增：协作与操作日志
32 | |   └─ collaboration/ # 智能体协作日志
33 | |   └─ process/       # 流程执行日志
34 | └─ Projects/          # 项目管理目录（未修改）
35 |   └─ 001-dewatermark-ai/ # 示例项目：AI去水印工具
36 |   └─ ...
37 | └─ Models/            # 模型配置目录（优化后）
38 |   └─ core/            # 核心模型（如GPT-4 Turbo）
39 |   └─ lightweight/     # 轻量模型（用于非核心任务）
40 |   └─ config/           # 模型参数配置
```

四、修改必要性总结与实施建议

4.1 修改必要性

✓ 现有YDS-Lab标准目录结构**需要针对性修改**，核心原因：1. 原始结构未适配Trae的MCP协议、流程自动化等核心特性；2. 缺少智能体协作日志、安全配置等支撑模块，无法落地方案中的风险管控和流程优化机制；3. 智能体配置文件不完整，无法体现Trae的触发规则、输出规范等关键配置。

4.2 实施建议

- 分阶段实施：**

第一阶段（基础搭建期第1个月）：完成Struc/Agents/的配置文件新增（trigger.yaml等）和Models/的目录拆分，支撑智能体初始化配置；
- 第二阶段（基础搭建期第2个月）：**新增Struc/Workflow/和Struc/Security/，配置核心流程规则和安全策略；
- 第三阶段（试点验证期第5个月）：**优化Struc/Tools/的MCP适配，根据试点项目需求补充工具配置。
- 兼容性保障：**保留原始目录的核心逻辑（如Agents/、Departments/），仅新增或优化子目录，避免对历史配置的大幅改动；同时在根目录新增“directory_change_log.md”，记录目录修改轨迹，便于团队追溯。
- 文档同步：**在各新增/优化目录下同步新增“README.md”，明确目录用途、配置规范（如trigger.yaml的字段说明、MCP工具的适配步骤）及责任人，适配团队协作需求。
- 验证机制：**每个阶段修改完成后，通过“目录配置校验清单”（含目录完整性、配置文件规范性、与Trae方案匹配度三项指标）进行自检，确保修改落地有效。

（注：文档部分内容可能由 AI 生成）