

应用商店

第一部分 总体设计

应用管理员可以在应用商店安装、浏览和管理应用。

第二部分 逻辑设计

一、术语

业务知识网络

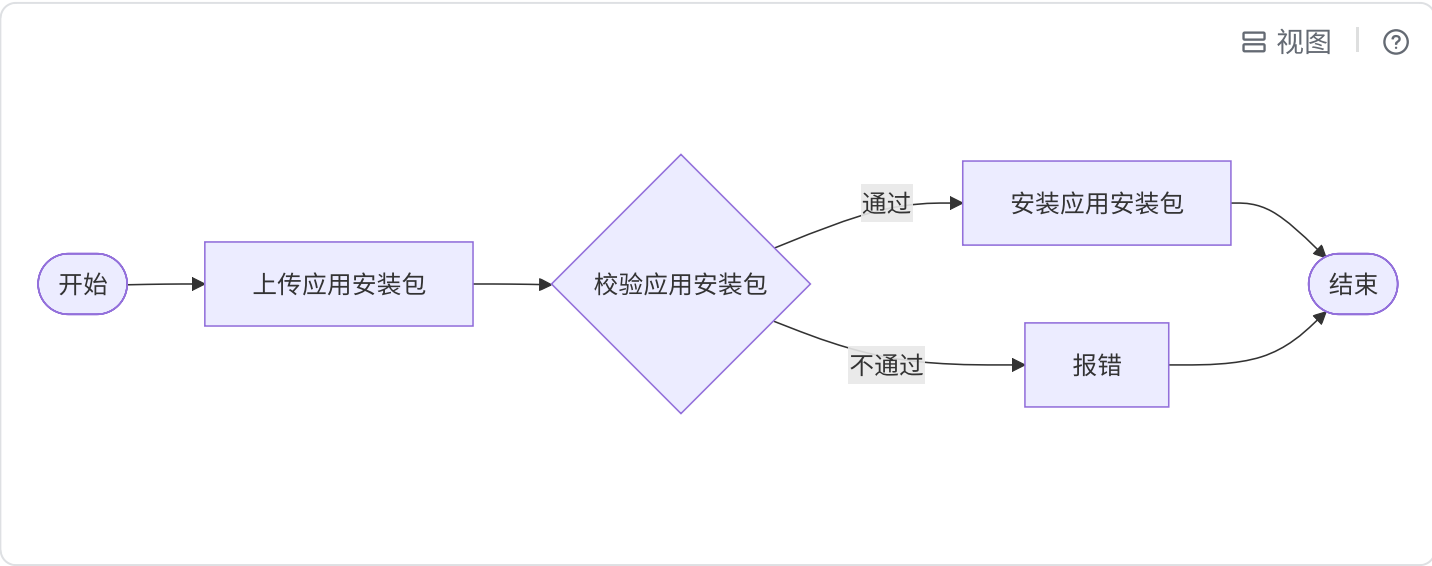
即本体模型。

二、安装应用

2.1 安装应用流程

1、安装应用流程分为两个子流程：

- 上传应用包并校验
- 安装应用包



2.1.1 上传应用包

1、所有的应用信息都从应用包中读取，基本信息包括以下属性：

- 应用名称
- 应用描述
- 应用图标
- 更新者
- 更新时间

2、应用安装包为 ZIP 压缩包，扩展名为： `.dip`。

3、应用包大小限制为 1GB。

4、应用安装包内部结构如下：

```
application.dip
1  my-application/
2  |— application.key
3  |— manifest.yaml
4  |— assets/
5  |   |— icons/
6  |       |— 64.png
7  |— packages/
8  |   |— images/
9  |       |— image1.tar
10 |       |— images2.tar
11 |   |— charts/
12 |       |— chart1.tgz
13 |       |— chart2.tgz
14 |— ontologies/
15 |   |— ontology.json
16 |   |— ontology1.json
17 |— agents/
18 |   |— agent.json
19 |   |— agent1.json
```

5、以下是对应用安装包的内部结构的说明：

目录 / 文件	是否必须	说明
application.key	必须	记录应用包的唯一ID，32 位大小写敏感的英文+数字组合，用于校验应用唯一性
manifest.yaml	必须	应用安装包描述文件
assets/icons/64.png	非必须	64x64 的应用图标

packages/	必须	镜像 + Chart 目录
packages/images	必须	服务镜像，以oci-archive OCI 官方标准格式打包
packages/charts	必须	chart包，tgz打包格式
ontologies/	非必须	<p>本体模型的配置文件(json文件由业务知识网络导出)，支持导入多个业务知识网络</p> <p>ontology.json数据结构 example:</p> <div><p>代码块</p><pre>1 components: 2 schemas: 3 CreateAgentRequest: 4 type: object 5 properties: 6 key: 7 type: string 8 description: agent标识 9 maxLength: 50 10 is_built_in: 11 type: integer 12 description: 13 - 是否内置（内部接口有效） 14 - 0: 否 15 - 1: 是 16 example: 0 17 is_system_agent: 18 type: integer 19 description: 是否是系统智能体 20 enum: [0, 1] 21 example: 0 22 name: 23 type: string 24 description: 名字 25 example: agent1 26 maxLength: 50 27 profile: 28 type: string 29 description: 简介 30 maxLength: 100 31 avatar_type: 32 type: integer 33 description: 34 - 头像类型 35 - 1: 内置头像 36 - 2: 用户上传头像</pre></div>

```

37         - 3: AI生成头像
38         example: 1
39     avatar:
40         type: string
41         description: |
42             - 头像信息
43             - 当头像类型为1时，为内置头像标
44 识
45             - 当头像类型为2时，为头像存储信
46 息
47             - 当头像类型为3时，为头像存储信
48 息
49         example: "1"
50     product_key:
51         type: string
52         description: 所属产品标识
53         example: dip
54     config:
55         $ref:
56         '#/components/schemas/AgentConfig'
57     created_by:
58         type: string
59         description: 创建人uid（内部接口有
60 效）
61         example: xxx
62     required:
63         - name
64         - avatar_type
65         - avatar
66         - product_key
67         - config
68
69     AgentConfig:
70         type: object
71         description: data agent配置
72         properties:
73             input:
74                 $ref:
75                 '#/components/schemas/AgentInput'
76             system_prompt:
77                 type: string
78                 description: 系统提示词
79             dolphin:
80                 type: string
81                 description: dolphin语句
82             is_dolphin_mode:
83                 type: integer

```

```
78         description: 是否是dolphin模式
79         enum: [0, 1]
80         default: 0
81     data_source:
82         $ref:
83         '#/components/schemas/DataSource'
84     skills:
85         $ref:
86         '#/components/schemas/Skills'
87     llms:
88         type: array
89         items:
90             $ref:
91             '#/components/schemas/LLMItem'
92     is_data_flow_set_enabled:
93         type: integer
94         description: 是否启用数据流设置
95         enum: [0, 1]
96         example: 0
97     opening_remark_config:
98         $ref:
99         '#/components/schemas/OpeningRemarkConfig'
100     preset_questions:
101         type: array
102         description: 预设问题列表
103         items:
104             type: object
105             required:
106                 - question
107             properties:
108                 question:
109                     type: string
110                     description: 问题内容
111     output:
112         $ref:
113         '#/components/schemas/AgentOutput'
114     built_in_can_edit_fields:
115         type: object
116         description: 内置可编辑字段配置
117         additionalProperties:
118             type: boolean
119     memory:
120         type: object
121         properties:
122             is_enabled:
123                 type: boolean
124             description: 是否启用记忆
```

```
120         related_question:
121             type: object
122         properties:
123             is_enabled:
124                 type: boolean
125                 description: 是否启用相关问题
126         plan_mode:
127             type: object
128         properties:
129             is_enabled:
130                 type: boolean
131                 description: 是否启用规划模式
132         required:
133             - input
134             - llms
135             - output
136
137     AgentInput:
138         type: object
139         description: agent输入配置
140         properties:
141             fields:
142                 type: array
143             items:
144                 type: object
145                 properties:
146                     name:
147                         type: string
148                         description: 字段名
149                     type:
150                         type: string
151                         description: 字段类型
152             is_temp_zone_enabled:
153                 type: integer
154                 description: 是否启用临时区
155                 enum: [0, 1]
156             temp_zone_config:
157                 type: object
158                 description: 临时区配置
159                 properties:
160                     name:
161                         type: string
162                     tmp_file_use_type:
163                         type: string
164                     max_file_count:
165                         type: integer
166                     single_file_size_limit:
```

```
167             type: integer
168         single_file_size_limit_unit:
169             type: string
170         support_data_type:
171             type: array
172             items:
173                 type: string
174         allowed_file_categories:
175             type: array
176             items:
177                 type: string
178
179     DataSource:
180         type: object
181         description: 数据源配置
182         properties:
183             doc:
184                 type: array
185                 items:
186                     type: object
187             kg:
188                 type: array
189                 items:
190                     type: object
191             metric:
192                 type: array
193                 items:
194                     type: object
195             kn_entry:
196                 type: array
197                 items:
198                     type: object
199             knowledge_network:
200                 type: array
201                 items:
202                     type: object
203             advanced_config:
204                 type: object
205                 properties:
206                     doc:
207                         type: object
208                     kg:
209                         type: object
210
211     Skills:
212         type: object
213         description: 技能配置
```

```

214         properties:
215             tools:
216                 type: array
217                 items:
218                     type: object
219                     properties:
220                         tool_id:
221                             type: string
222                         tool_box_id:
223                             type: string
224                         tool_input:
225                             type: object
226                             nullable: true
227                         intervention:
228                             type: boolean
229             agents:
230                 type: array
231                 items:
232                     type: object
233                     properties:
234                         agent_key:
235                             type: string
236                         agent_version:
237                             type: string
238                         agent_input:
239                             type: object
240                             nullable: true
241                         intervention:
242                             type: boolean
243             mcps:
244                 type: array
245                 items:
246                     type: object
247                     properties:
248                         mcp_server_id:
249                             type: string
250
251         LLMItem:
252             type: object
253             properties:
254                 is_default:
255                     type: boolean
256                 description: 是否默认
257             llm_config:
258                 $ref:
259
260     '#/components/schemas/LLMConfig'
```



```
260 LLMConfig:
261     type: object
262     description: LLM配置
263     properties:
264         id:
265             type: string
266             description: 模型ID
267         name:
268             type: string
269             description: 模型名称
270         model_type:
271             type: string
272             description: 模型类型
273         temperature:
274             type: number
275             description: 温度参数
276         top_p:
277             type: number
278         top_k:
279             type: integer
280         frequency_penalty:
281             type: number
282         presence_penalty:
283             type: number
284         max_tokens:
285             type: integer
286             description: 最大token数
287
288 OpeningRemarkConfig:
289     type: object
290     description: 开场白配置
291     required:
292         - type
293     properties:
294         type:
295             type: string
296             description: 开场白类型（固定/动态）
297         enum: ["fixed", "dynamic"]
298         fixed_opening_remark:
299             type: string
300             description: 固定开场白
301         dynamic_opening_remark_prompt:
302             type: string
303             description: 动态开场白提示语
304
305 AgentOutput:
```

```
306     type: object
307     description: 输出配置
308     properties:
309         variables:
310             type: object
311             properties:
312                 answer_var:
313                     type: string
314                 doc_retrieval_var:
315                     type: string
316                 graph_retrieval_var:
317                     type: string
318             default_format:
319                 type: string
320             description: 默认输出格式
321
322     CreateAgentResponse:
323         type: object
324         properties:
325             id:
326                 type: string
327                 description: agent id
328             version:
329                 type: string
330                 description: |
331                     - agent版本
332                     - 未发布data agent的版本为: v0
333         required:
334             - id
335             - version
336
337     AgentFactoryError:
338         type: object
339         description: 错误信息（统一格式）
340         required:
341             - error_code
342             - description
343             - solution
344             - error_link
345             - error_details
346         properties:
347             error_code:
348                 type: string
349                 description: 业务错误码
350             description:
351                 type: string
352                 description: 错误描述
```

		<pre>353 solution: 354 type: string 355 description: 解决方法 356 error_link: 357 type: string 358 description: 错误链接 359 error_details: 360 description: 361 - 错误详情（比如当某些数据不存在 时，这里可以包含不存在的id） 362 - 任意类型 363 nullable: true 364</pre>
agents/	非必须	Data Agent 智能体的配置文件，支持导入多个业务知识网络 json数据结构和创建agent接口一致

6、manifest.yaml 示例如下：

```
manifest.yaml

1  manifest_version: 1
2  name: 智能故障分析
3  micro-app:
4      name: intelligent_fault_analysis
5      entry: /intelligent_fault_analysis
6      headless: false
7  version: 1.0.0
8  category: DIP for ITOps
9  description: |
10     这是一个 DIP for ITOps 运维大脑应用
```

7、manifest.yaml 文件的说明如下：

字段	是否必须	说明
manifest_version	必须	固定为 1，主要是用来向前兼容 manifest.yaml 本身的解析方式
name	必须	应用的名称
micro-app.name	必须	微应用的包名
micro-app.entry	必须	微应用的入口路径,false显示导航头，true不显示导航头

micro-app.headless	非必须	微应用是否有导航头，false：默认有导航头，true：不显示导航头
version	必须	语义版本号，更新应用包时必须确保新应用包版本号 > 已安装版本号
category	非必须	应用所属目录，在应用列表中会按目录进行分组。为空则分到“未分组”下
description	非必须	应用的描述

8、上传应用包时，需要进行以下验证：

```
1 flowchart LR
2   A([开始]) --> B{ 是否完整? }
3   B -- 是 --> C{ ID 是否一致? }
4   C -- 是 --> E{ 新版本号 > 旧版本号 }
5   E -- 是 --> G[ 更新应用 ]
6   E -- 否 --> H[ 报错：版本号必须大于 ]
7   C -- 否 --> F[ 上传新应用 ]
8   B -- 否 --> D[ 报错：应用安装包不完整 ]
9
10  D --> O([结束])
11  G --> O
12  H --> O
13  F --> O
14
```

吕 视图 | ?

2.1.2 安装应用包

1、上传应用包并校验通过后，会自动触发安装应用包的流程。

2、安装应用会在后台执行以下操作：

- 解压应用包
- 推送镜像和 Chart 到部署工作台
- 安装 Chart
- 创建业务知识网络，即导入应用包中的 `ontology/ontology.json`
- 创建 Agent，根据 `agent/agent.json` 的配置进行创建
- 启动容器

三、浏览应用

3.1 浏览应用

1、应用市场显示的应用属性包括：应用名称、应用描述、应用图标。

2、应用管理员可以对应用进行的操作包括：

- 配置
- 运行
- 卸载

四、查看应用配置

4.1 查看配置项

1、查看应用配置项时，由于配置项内容较多且部分配置项比较复杂，因此需要采用抽屉组件显示配置项信息。

2、可查看的应用配置包括：基本信息、业务知识网络、智能体。

3、每项应用配置并没有严格的依赖关系，查看配置项时可以自由切换配置项查看里面的内容。

4、如果应用包内不包含某项配置，则在查看应用配置时，仍然显示该配置项标题，但该配置内容提示：不需要配置 { 配置项目名 }，如：不需要配置业务知识网络。

4.2 业务知识网络

1、业务知识网络即本体模型，用户可以以图的结构查看本体模型，支持放大、缩小、拖动画布，但不支持拖动点和边。

2、应用管理员可以通过链接跳转到 ADP 平台配置数据视图映射。

4.3 智能体

1、智能体配置信息包括：

- 智能体名称（只读）
- 智能体描述（只读）
- 智能体提示词（只读）
- 智能体技能列表（只读）

2、应用管理员可以通过链接跳转到 ADP 平台查看智能体的详细配置。

五、运行应用

1、管理员配置完应用之后，可以运行应用来进行测试。

2、仅当所有必要配置项都保存成功时，才可以运行应用。如果有配置项未保存或失效，则禁用应用操作并且在应用按钮右侧使用警告 ICON，悬浮提示“当前应用配置失效，请先检查应用配置”。

六、卸载应用

1、卸载应用前提醒管理员：“卸载应用后，相关配置和数据将被清除，用户将无法使用应用。是否继续?”

第三部分 原型

