

EulerCopilot 赋能openEuler更智能



开放原子开源基金会
OPENATOM FOUNDATION



OpenEuler

开放原子
开源行
MEEUP
原子社区
开源同行

目录

- AI与OS融合技术洞察

- EulerCopilot—智能问答服务

 - 整体介绍

 - 方案

- EulerCopilot—1+1+N智能体框架

 - 洞察

 - 方案

 - Demo演示

- 敬请期待

我们正迈向智能时代

第一次工业革命



第二次工业革命



第三次工业革命



第四次工业革命



回顾历史进程，三次大的工业革命，推动人类社会迈入蒸汽时代、电气时代、互联网时代，每一次革命都深刻改变了世界。

人工智能目前看是人类历史上第四次工业革命，将带给全世界更为深刻的变革，已进入人们生活的方方面面。

大模型开启操作系统智能新时代，操作系统交互、管理和运维模式迎来变革

颠覆 聊天助手



ChatGPT
(GPT 3.5)
2023.2.7

ChatGPT: 一种对话生成模型，可以被用于构建聊天机器人，为用户提供自然、流畅的对话体验

颠覆 生产力工具



MS Copilot 365
(GPT 4)
2023.3.17

MS Copilot 365: copilot 嵌入到办公软件中，文档、PPT、数据分析等内容，自动生成

OS交互变革



Windows
Copilot
2023.5.24

开发工具: Copilot SDK和IDE，提升开发效率
云服务: 文本、代码生成等
Copilot: 大模型内嵌OS, 变革交互方式
TaskMatrix: 基于大语言模型驱动数百万API，构建通用任务平台

OS自动化运维和AI基础设施变革



Ansible
Lightspeed
2023.5.28

LightSpeed: 基于 watsonx.ai 大模型，与辅助编程，Ansible脚本生成和执行
OpenShift AI: AI和MLOPS企业级平台, 提供大模型开发和部署流水线服务；集成IBM watsonx.ai生成式ai服务；与NV软硬件相互认证



k8sgpt
0.7.4 release
2023.7.20

k8sgpt: 通过SRE经验训练的领域大模型，实现K8S集群故障诊断和分析；
Charmed kubeflow: AI和MLOPS开发平台；提供大模型开发和部署流水线；与NV软硬件相互认证

GPT 3.5

- ✓ 响应人类指令
- ✓ 泛化到未知领域
- ✓ 代码生成/理解
- ✓ 强大的内容生成能力

参数量：1750亿

GPT 4

- ✓ 文字和图片信息处理
- ✓ 更强理解模型
- ✓ 较强的角色认定
- ✓ 输入/记忆能力提升

参数量：1.8万亿

GPT4.5

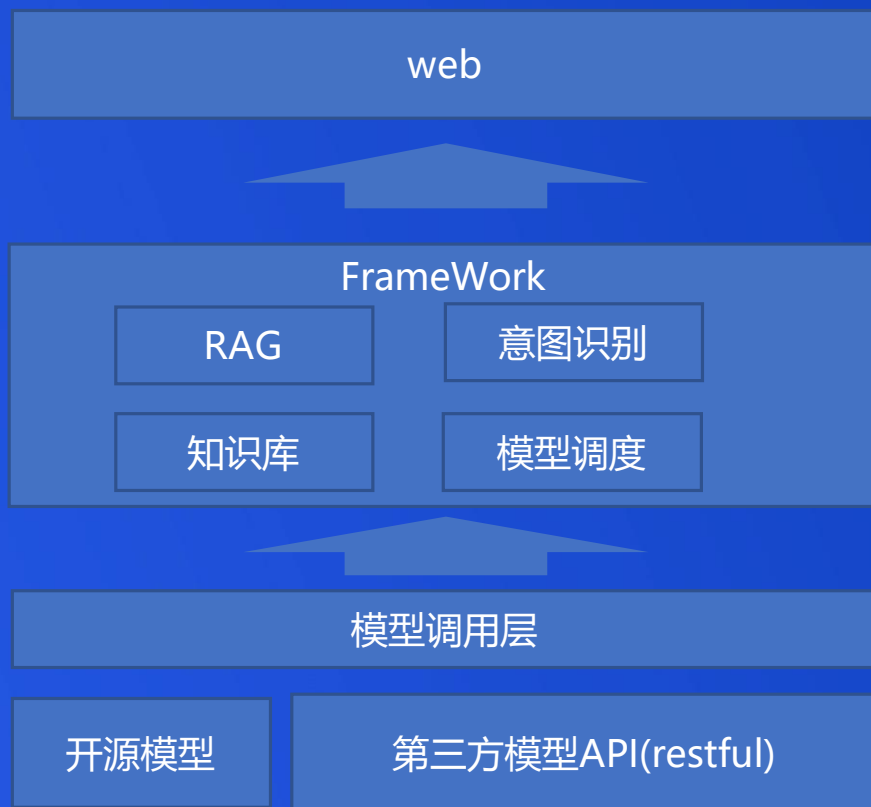
- ✓ 通用物理理解和分析能力
- ✓ 自然语义感知能力
- ✓ 零样本迁移能力
- ✓ 支持多模态输入

参数量：xx亿

GPT5

- ✓ 音频和视频处理
- ✓ 更强智能化处理能力
- ✓ 更长时间记忆能力
- ✓ 人工智能模型，自主开发

EulerCopilot—智能问答服务整体介绍



赋能openEuler更智能:

- 构建openEuler统一知识入口, 快速获取信息。
- 自然语言智能交互, 降低openEuler开发及运维门槛。

您好, 我是EulerCopilot

作为您的openEuler的智能助手, 您可以 试着问我关于以下方面的内容哦~



OS领域通用知识

包含Linux常规知识、上游信息和工具链介绍及指导。



openEuler专业知识

包含openEuler社区信息、技术原理和使用等介绍。



openEuler扩展知识

包含openEuler周边硬件特性知识和ISV、OSV相关信息。



openEuler应用案例

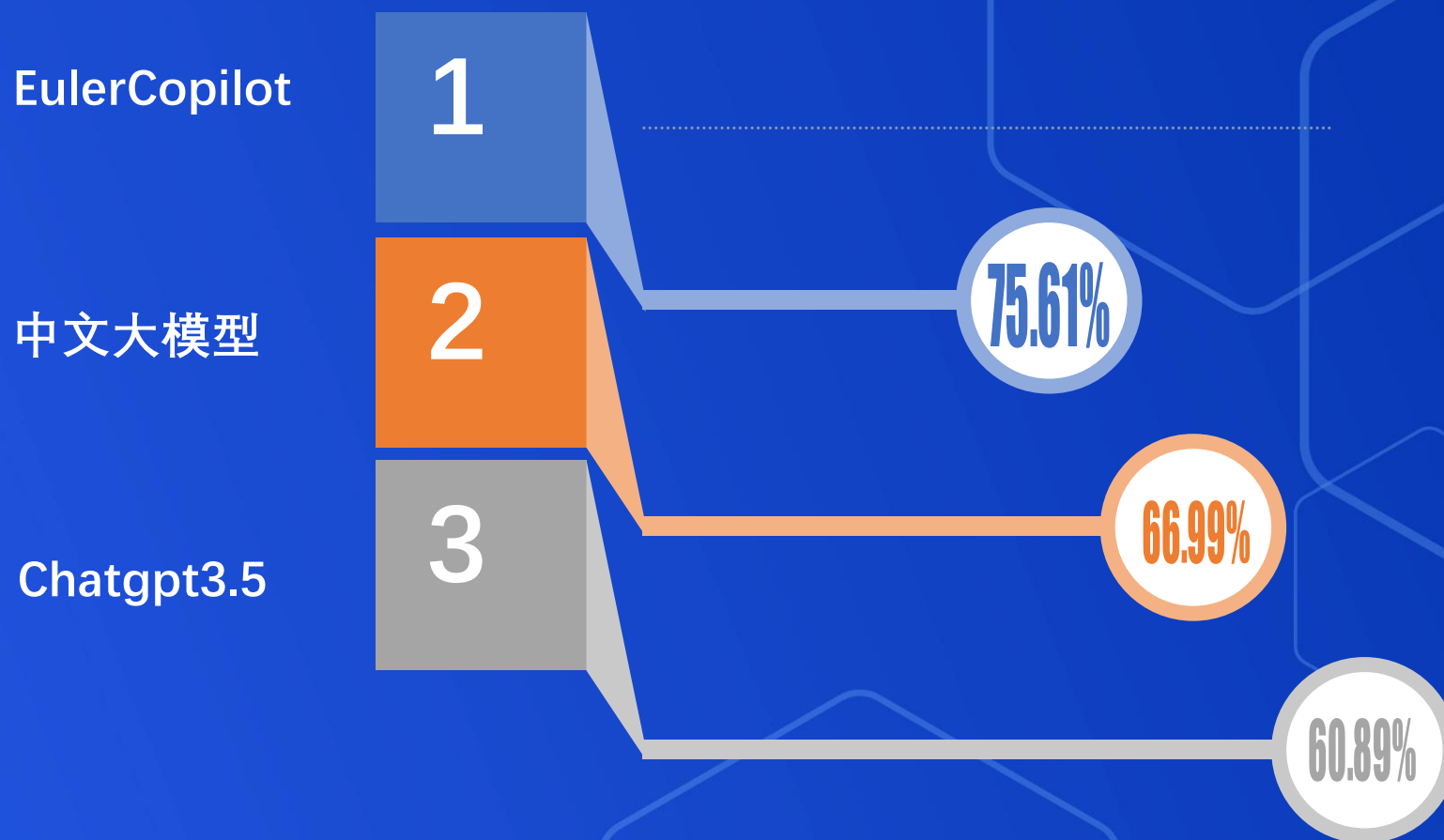
包含openEuler技术案例、行业应用案例。



shell命令生成

帮助用户生成单条命令或复杂命令。

智能问答服务准确率评估—openEuler垂直领域最优



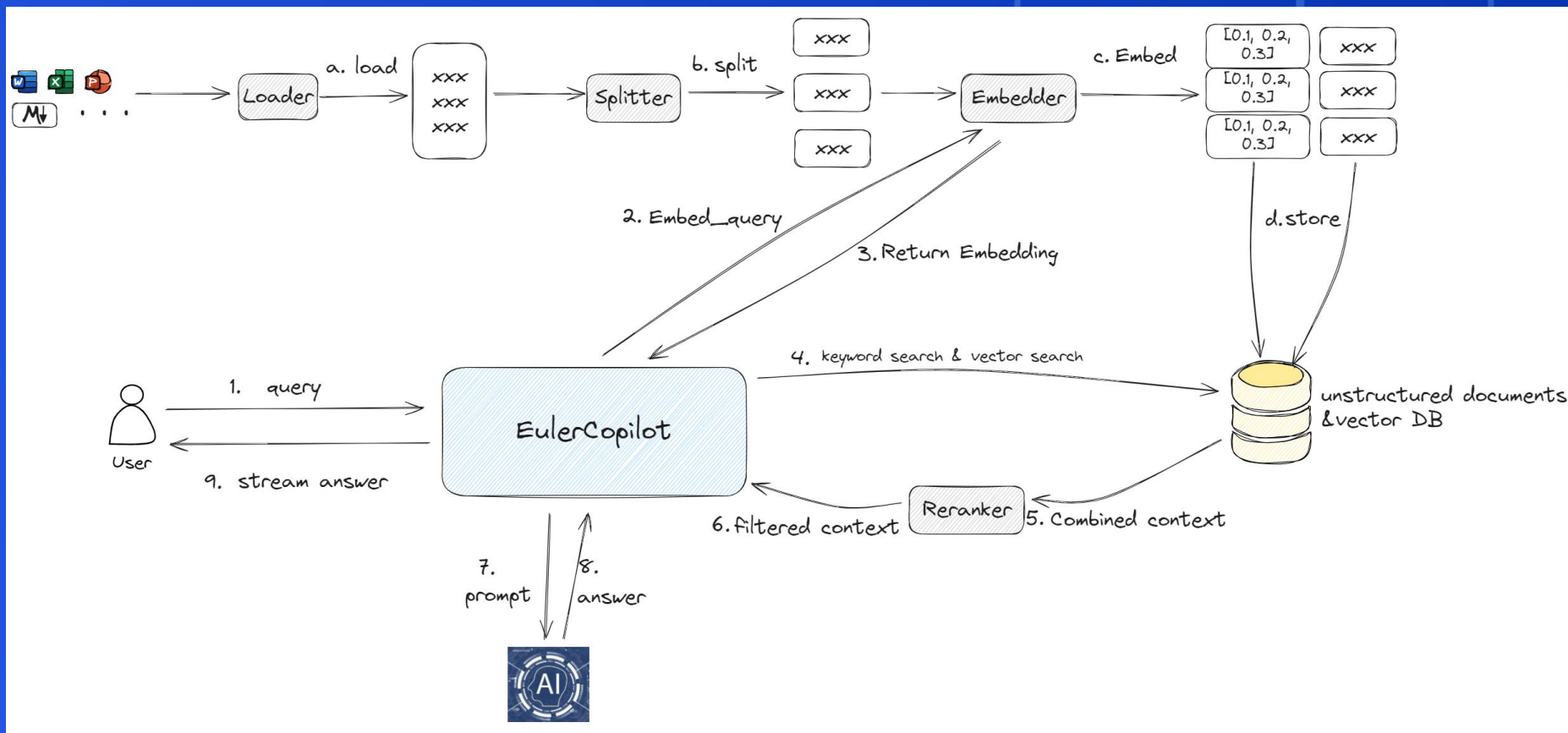
智能问答实现方案—构建可实时感知的领域AI助手

As-is

1. 预训练大模型**不了解领域知识**
2. 大模型训练/微调成本高、时间周期长，知识**更新不及时**
3. 需要**人工检索**多个知识源

To-be

1. 领域知识库构建**领域知识库**充当大模型“海马体”
2. **更新便捷**，及时为大模型提供**最新知识**
3. 检索知识库，通过大模型**归纳汇总**



AI应用达到奇点，从2C辅助应用逐渐深入2B生产核心系统，需要跨越技术鸿沟



工程范式变化

从“标注数据”到“标注接口”

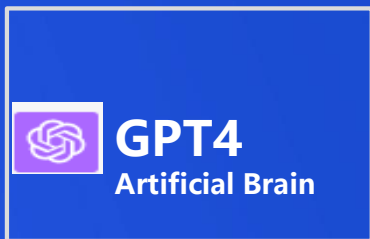
交互范式变化

从面向鼠标的“图形界面交互”到“以自然语言为主流的体感交互”

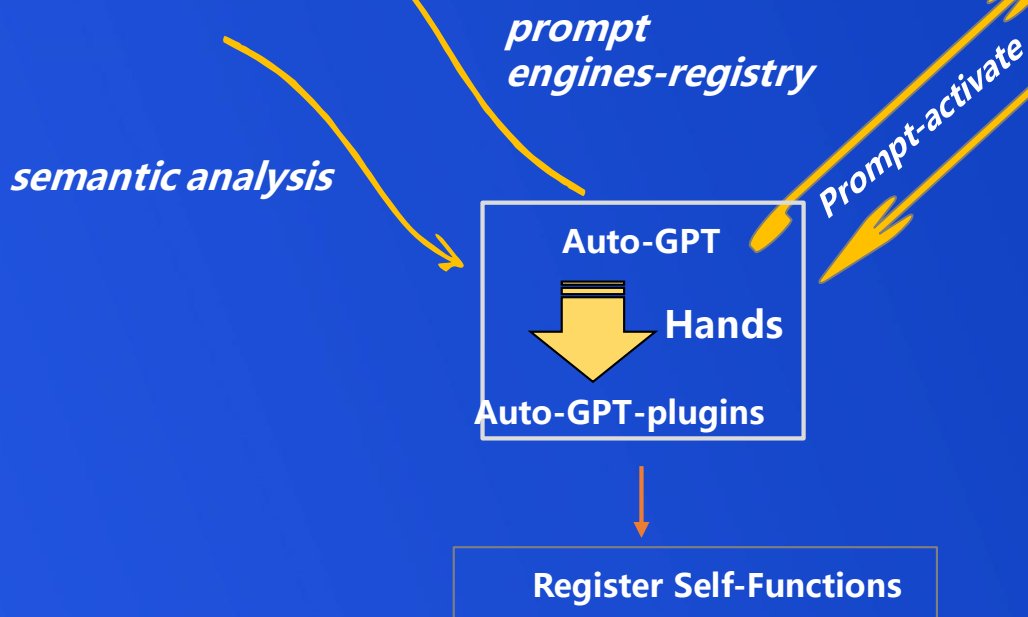
开发范式变化

从面向编程语言的“功能型API”进化到面向人和大模型的“语义级API”

AutoGPT: 提供插件接入能力, 自然语言识别, agent 协助完成事务相关工作



query: 等xxx特性上库后, 启动版本流水线, 如果构建失败, 通知pr相关人员来修改, 重新启动流水线, 测试报告按照上次的格式邮件推送给xxx。



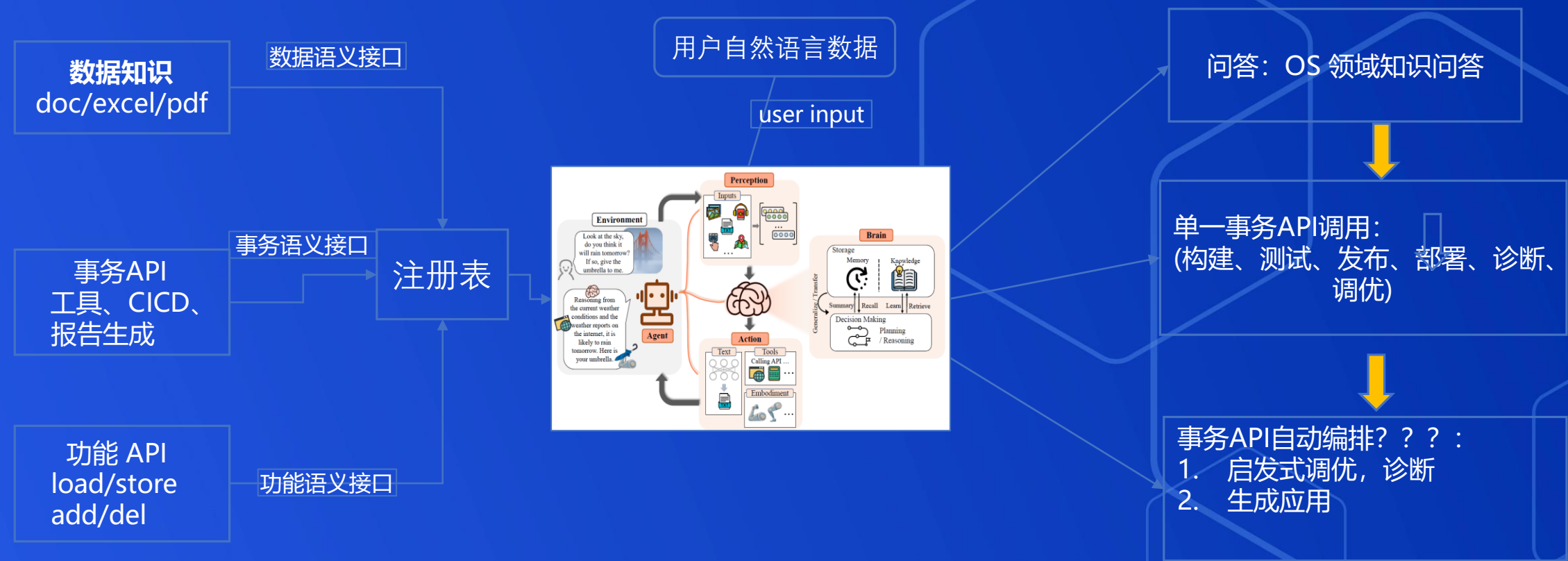
```
class AutoGPTBaiduSearch(AutoGPTPluginTemplate):  
    def __init__(self):  
        super().__init__()  
        self._name = "Baidu-Search-Plugin"  
        self._version = "0.1.0"  
        self._description = (  
            "This plugin performs Baidu searches using the provided query."  
        )  
        self.load_commands = (  
            os.getenv("SEARCH_ENGINE")  
            and os.getenv("SEARCH_ENGINE").lower() == "baidu"  
            and os.getenv("BAIDU_COOKIE")  
        )  
  
    def can_handle_post_prompt(self) -> bool:  
        return True  
  
    def post_prompt(self, prompt: PromptGenerator) -> PromptGenerator:  
        if self.load_commands:  
            # Add Baidu Search command  
            prompt.add_command(  
                "Baidu Search",  
                "baidu_search",  
                {"query": "<query>"},  
                _baidu_search,  
            )
```

```
def _baidu_search(query: str, num_results=8):  
    """  
    Perform a Baidu search and return the results as a JSON string.  
    """  
  
    headers = {  
        'Cookie': os.getenv("BAIDU_COOKIE"),  
        'User-Agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:112.0) Gecko/20100101 Firefox/112.0"  
    }  
    url = f'https://www.baidu.com/s?wd={query}&rn={num_results}'  
    response = requests.get(url, headers=headers)  
    response.encoding = 'utf-8'  
    soup = BeautifulSoup(response.text, 'html.parser')  
    search_results = []  
  
    for result in soup.find_all('div', class_=re.compile("result c-container")):  
        title = result.find('h3', class_='t').get_text()  
        link = result.find('a', href=True)['href']  
        snippet = result.find('span', class_=re.compile("content-right_82s40"))  
        if snippet:  
            snippet = snippet.get_text()  
        else:  
            snippet = ""  
        search_results.append({  
            'title': title,  
            'href': link,  
            'snippet': snippet  
        })  
  
    return json.dumps(search_results, ensure_ascii=False, indent=4)
```

约束条件:

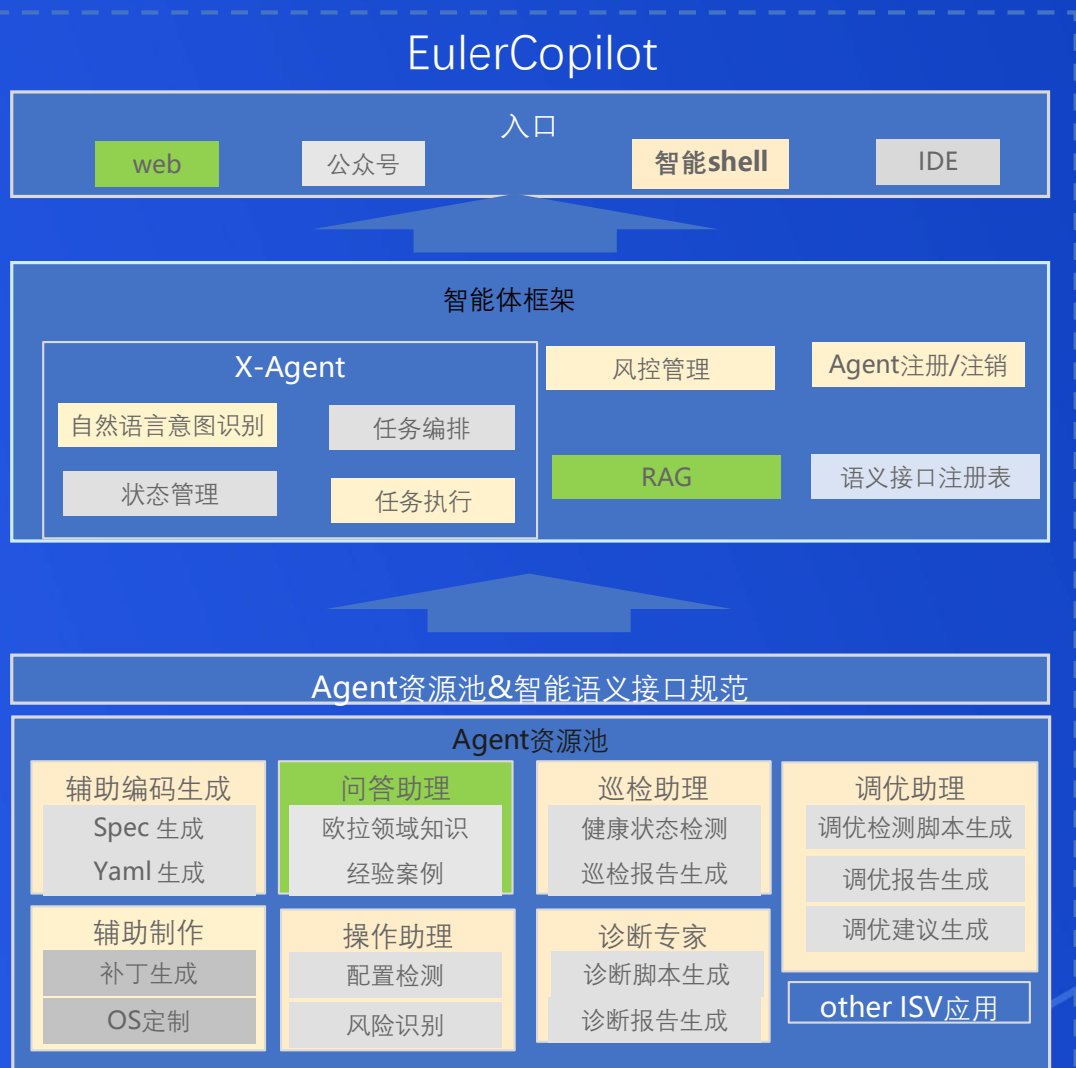
- 1、需要GPT-4同级大模型能力
- 2、需要自己实现hang并注册plugins, 完全需要代码开发来完成Agent调度
- 3、社区活跃度降低, 最近一次发布2023-12

智能体框架设想：从知识问答->单一事务处理->复杂事务编排

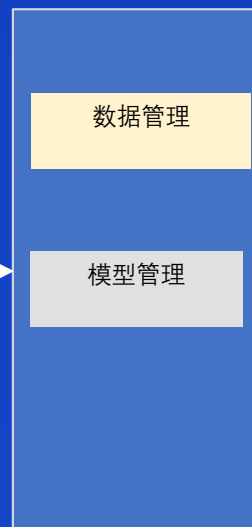


- 关键问题:
- 1、数据标注由大模型完成辅助，接口的描述标注如何生成 EPKG
 - 2、API 编排框架，对对事物API的编排，功能API的编排准确率更高

EulerCopilot: 基于1+1+N智能体架构的AI全栈解决方案



工程服务



特性规划

智能体框架(1)

- **X-Agent:** 构建OS领域模型, 实现agent注册, **复杂目标任务规划分解, 有序执行, 结果反馈;**
- **语义接口注册表:** **将Agent语义信息注册到OS语义接口注册表**

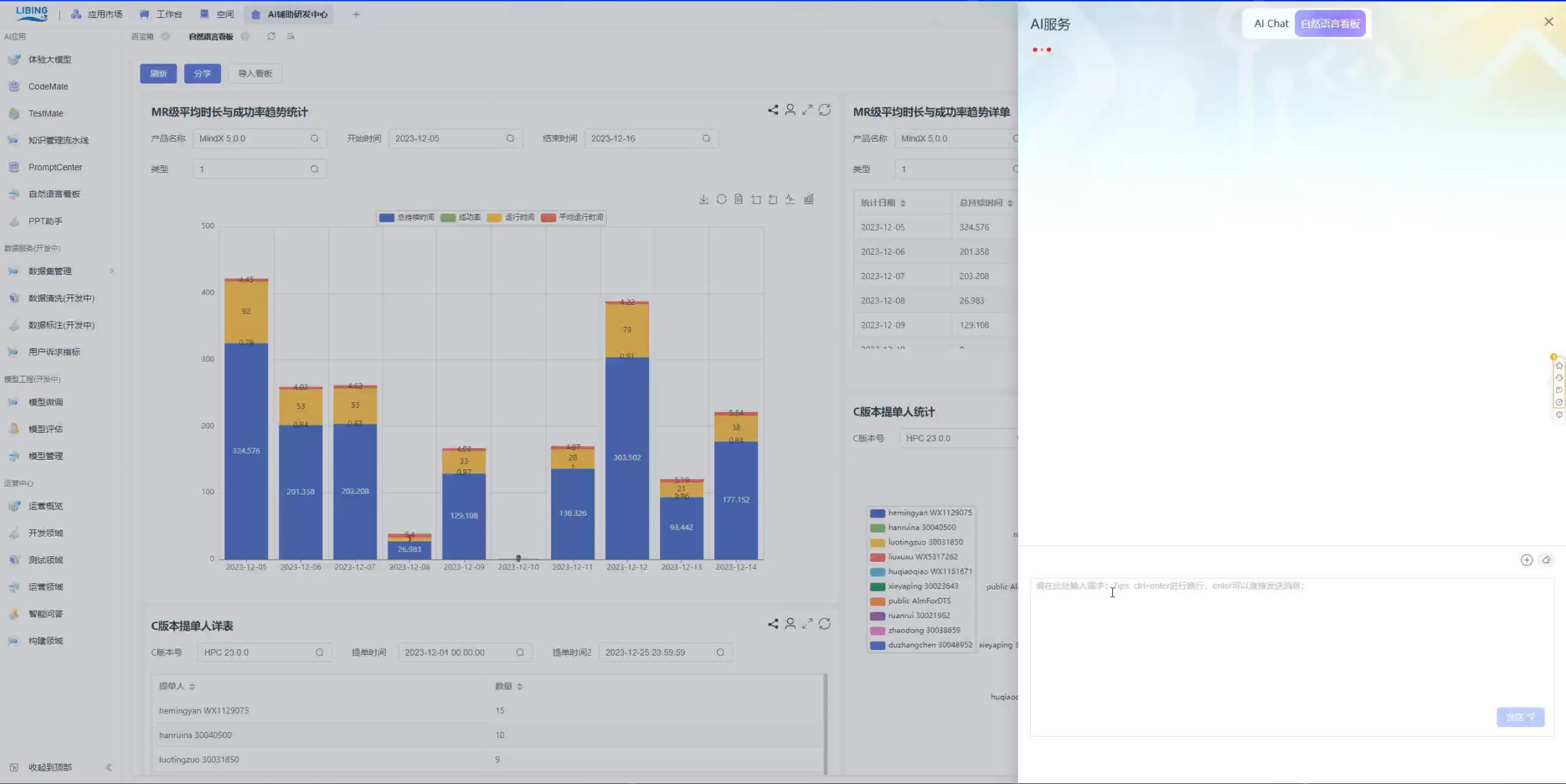
智能语义接口规范(1)

- **兼容openapi语法:** 提供一套新的规范, 同时兼容已有常见语法
- **零代码开发:** 开发者只需专注于生成语义接口描述信息

Agent (N: ISV应用) :

- **智能语义接口描述:** 根据语义接口规范, 通过一个配置文件描述当前Agent的能力以及调用方式。
- **ISV实体:** 工具、webserver等

Demo演示



敬请期待



24.03
版本
发布

新特性

贡献

智能问答:

- OS领域通用知识
- openEuler专业知识
- shell命令生成
- openEuler扩展知识
- openEuler应用案例

.....

openEuler官网在线访问

1+1+N智能体框架:

- 智能shell
- 智能运维
- 智能调优
- openEuler版本专家
- 社区事务
- 智能看板

.....

SIG组运作:

- 成立openEuler智能化SIG组
- 签署CLA, 参与EulerCopilot贡献

.....



THANKS