



2024 AI+研发数字峰会

AI+ Development Digital summit

AI驱动研发迈进数智化时代

中国·上海 05/17-18

多场景下智能体应用构建技巧

卢建晖 微软高级云技术布道师

科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



 **K+峰会**  **上海站**
K+ 全球软件研发行业创新峰会
时间: 2024.06.21-22

 **K+峰会**  **敦煌站**
K+ 思考周®研习社
时间: 2024.10.17-19

 **K+峰会**  **香港站**
K+ 思考周®研习社
时间: 2024.11.10-12



K+峰会详情



 **AiDD峰会**  **上海站**
AI+研发数字峰会
时间: 2024.05.17-18

 **AiDD峰会**  **北京站**
AI+研发数字峰会
时间: 2024.08.16-17

 **AiDD峰会**  **深圳站**
AI+研发数字峰会
时间: 2024.11.08-09



AiDD峰会详情



卢建晖

微软高级云技术布道师

专注在人工智能和大数据，微软多年技术大会讲师，微软人工智能黑客松教练，著有基于大模型框架的《Sematic Kernel Cookbook》，

目录

CONTENTS

1. 为什么需要智能体
2. 如何实现智能体
3. 智能体的应用场景
4. 研发场景的实践

PART 01

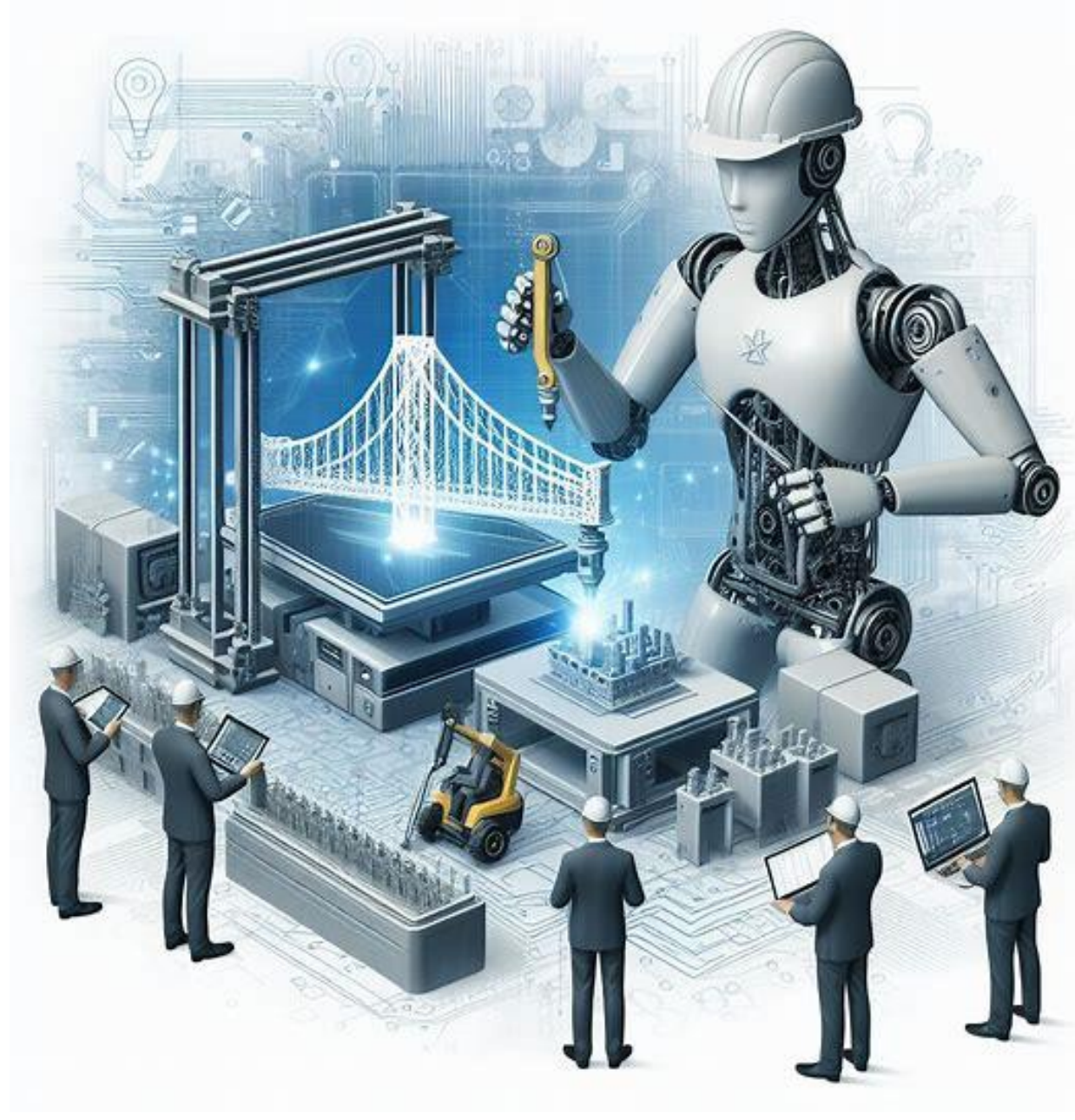
我们为什么需要智能体

▶ Copilot 应用改变了我们的工作方式



▶ 几个关键点

- 怎样写“好”提示工程
- Copilot 之间如何互动
- 解决大模型中的幻觉
- 准确性



▶ 对智能的愿景



PART 02

如何实现智能体

▶ AI Agents 的特点

自主性： AI Agents 能够独立执行任务，无需持续的人工干预或输入。

感知： Agents 功能通过各种传感器（例如摄像头或麦克风）感知并解释他们所处的环境。

高响应性： AI Agents 可以评估环境并做出相应响应以实现其目标。

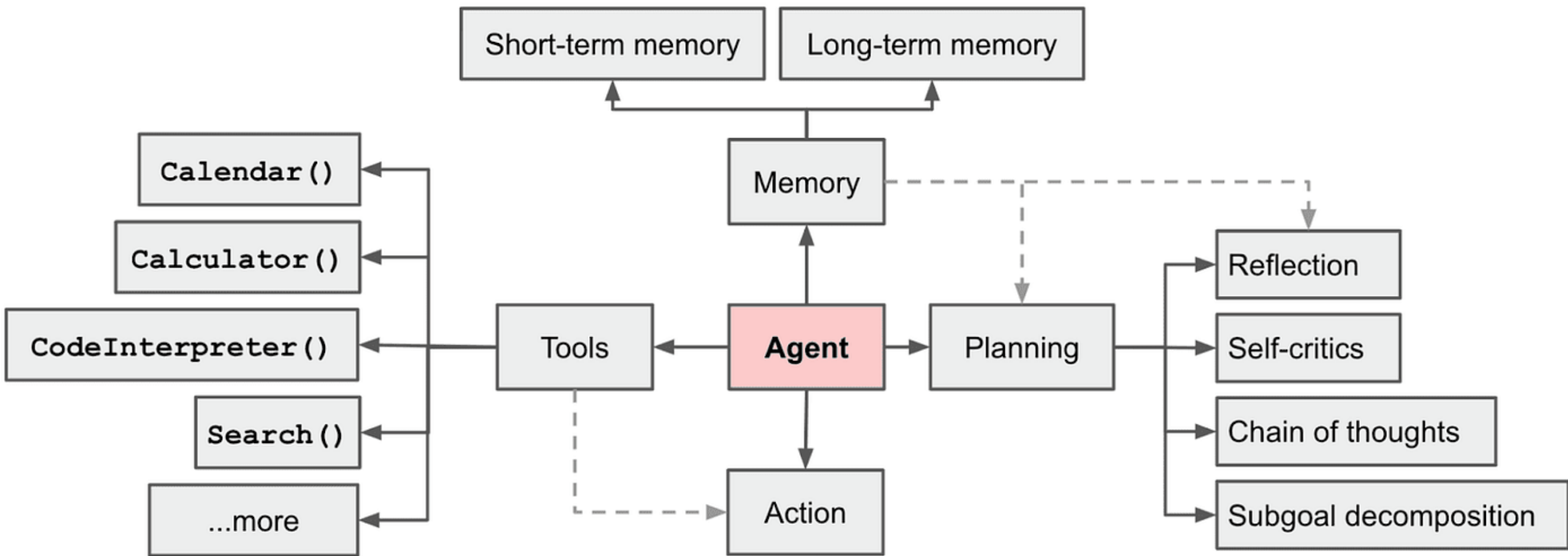
推理和决策： AI Agents 是智能工具，可以分析数据并做出决策以实现目标。他们使用推理技术和算法来处理信息并采取适当的行动。

学习： 他们可以通过机器、深度和强化学习元素和技术来学习和提高自己的表现。

通信： AI Agents 可以使用不同的方法与其他代理或人类进行通信，例如理解和响应自然语言、识别语音以及通过文本交换消息。

面向目标： 它们旨在实现特定目标，这些目标可以预先定义或通过环境的交互来学习。

▶ AI Agent 的结构



[LLM Powered Autonomous Agents | Lil'Log \(lilianweng.github.io\)](https://lilianweng.github.io)

▶ Assistants API 是第一步

Assistants API

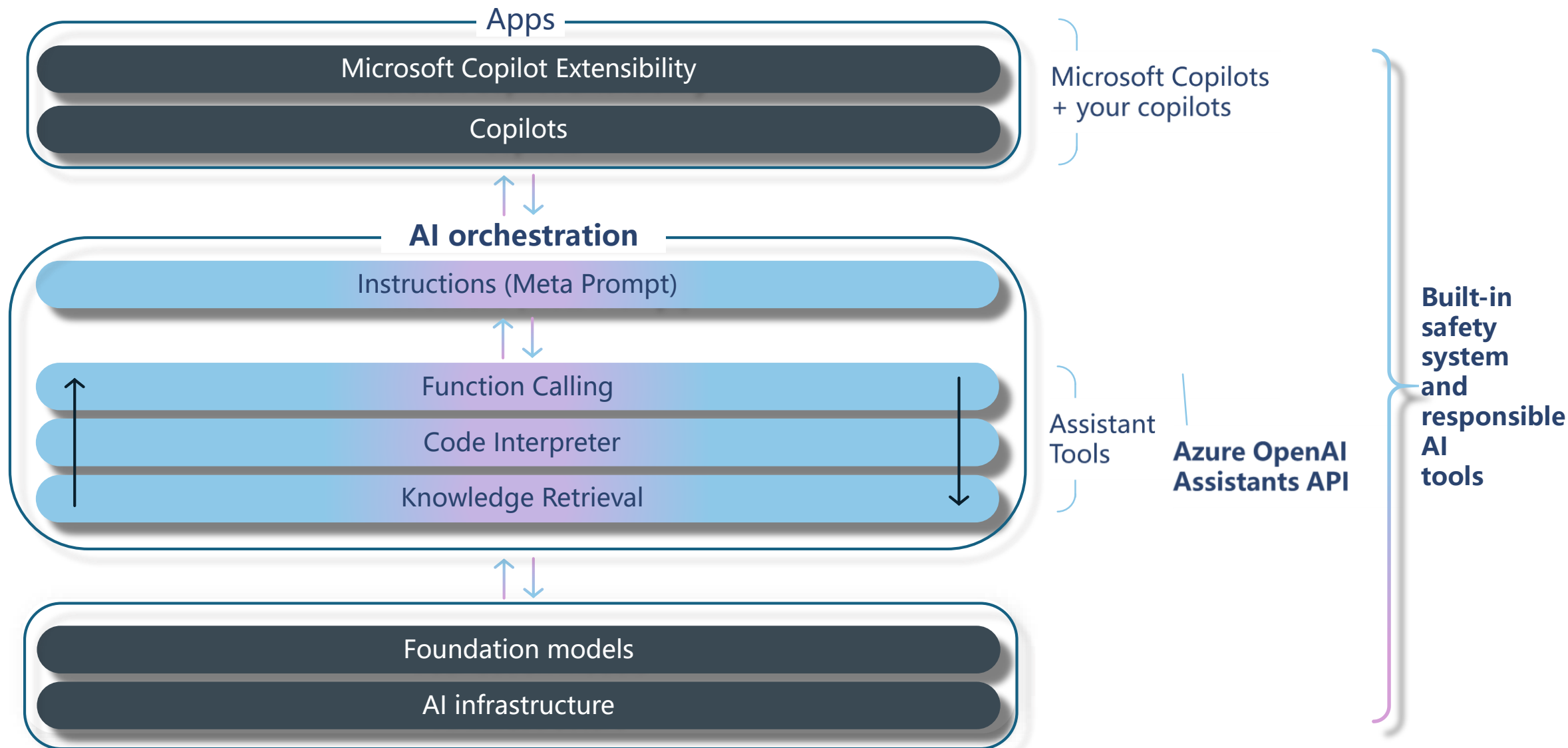
轻松为 Copilot 引入状态：通过单个 API 构建强大、有状态的 Copilot 和类似座席的体验

减少开发工作量：无需多个集成来进行状态管理、上下文关联和聊天线程、检索、代码执行和外部扩展

强大的工具和第三方扩展能力：使开发人员能够并行访问强大的内置工具，例如代码解释器、知识检索以及通过 function calling 的外部扩展

实现突破性的虚拟多智能体场景：设想虚拟智能体不仅可以相互交流，还可以执行任务，并可以通过安全方式来实现智能体间的工作流程

▶ Assistants Stack



▶ Chat Completion API vs Assistant API

Chat Completions API

- 轻巧而强大
- 本质上无状态

Assistants API

- Stateful (内置对话状态管理)
- 访问持久线程
- 访问多种格式的文件。API 处理分块、嵌入存储和创建以及实施矢量搜索*
- 自动管理模型的上下文窗口
- 并行访问多个工具 (每个助手最多 128 个工具), 包括代码解释器
- 使用 function calling 来构建工具

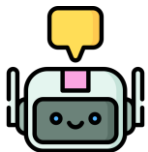
▶ 场景学习: 使用 Assistants API 构建一个旅游应用

分析一下具体工作 - **行业:** 旅行; **领域:** 客户服务; **目标:** 缩短响应时间, 取悦用户

1 参与者



顾客
(需要旅行计划和预订帮助)



助理
旅游顾问
(用户代理)



助理
订舱代理
(专业代理)

2 相关工具

1. 代码解释器



解决复杂的数学问题和高级数据分析

2. 知识检索



RAG
(1) flight_itinerary.txt
(2) Airbnb_bills.pdf
(3) train_ids.csv
(4) stations.csv

3. Function Calling (与多个API集成)



Deutsche Bahn API
(实时火车时刻表)



Graph API or Logic Apps
(read/write to Outlook)

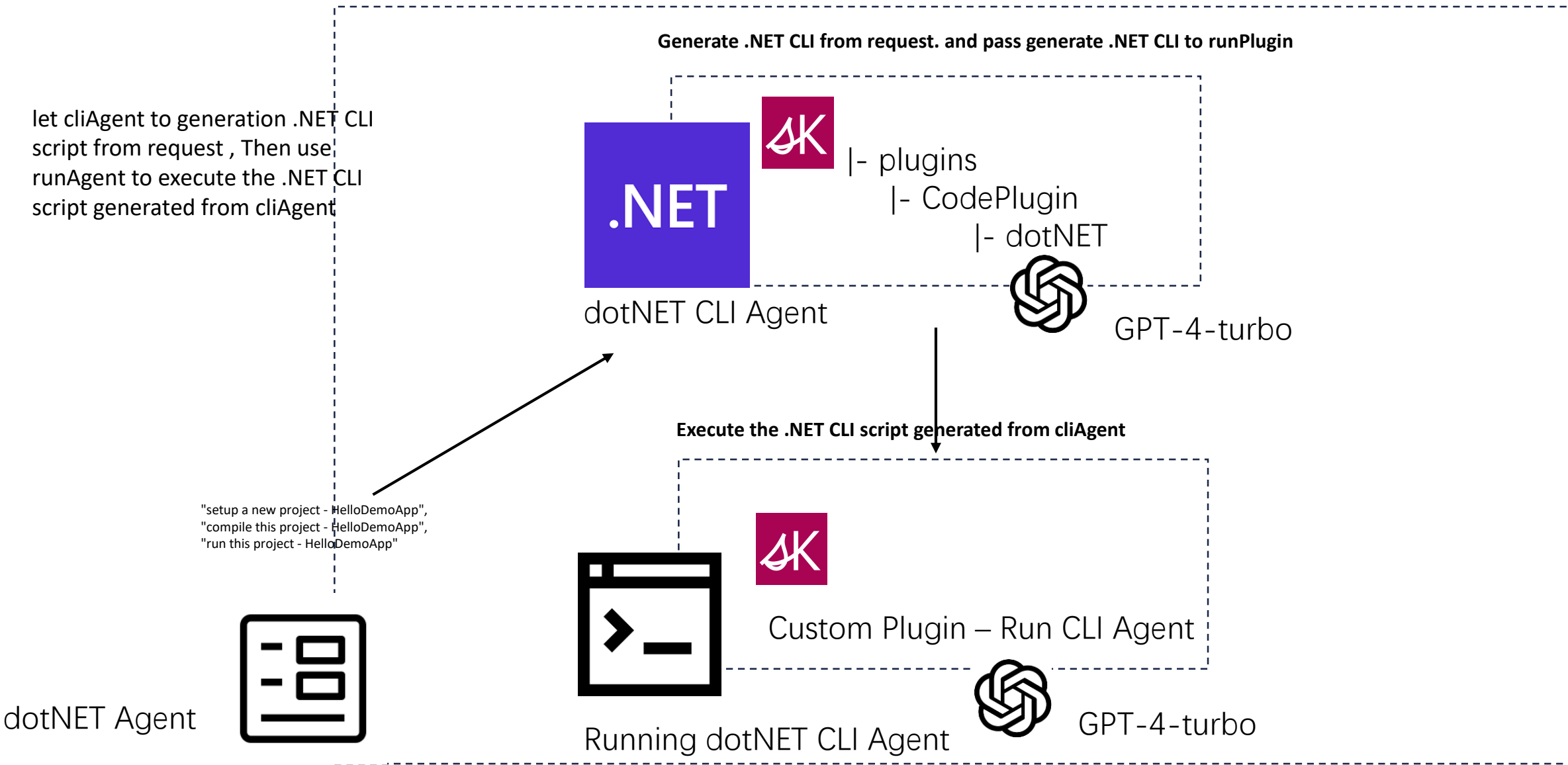
3 助理配置

- 选择型号 (gpt-4-1106-preview)
- 为每个助理定义自定义指令: “您是提供.....的旅行顾问”, “您是火车预订代理.....”
- 激活检索和代码解释器工具
- 添加文件
- 添加功能: 例如 获取旅程、获取实际日期时间、获取最佳价格

4 Multi-Assistant Thread Runs with parallel function calls

- 用户上传航班行程并询问: “建议我旅行日期的每日行程”
 - 助理检索文件, 然后回答问题
- 用户询问 “阻止我的日历中的慕尼黑啤酒节日期”
 - 助理查找音乐节日期的行程并调用 Microsoft Graph API 在用户日历中创建 Outlook 事件
- 用户上传他们的 Airbnb 账单并询问: “我们将是 4 位朋友住在柏林的这家 Airbnb 上。我的费用分摊 + 我的航班费用是多少?”
 - Assistant 调用 Code Interpreter 生成 Python 代码, 然后回答问题。
- 用户询问 “请提供 10 月 27 日当周从 柏林 到 慕尼黑 最便宜的航班”
 - Assistant调用DB API获取火车路线和票价并显示

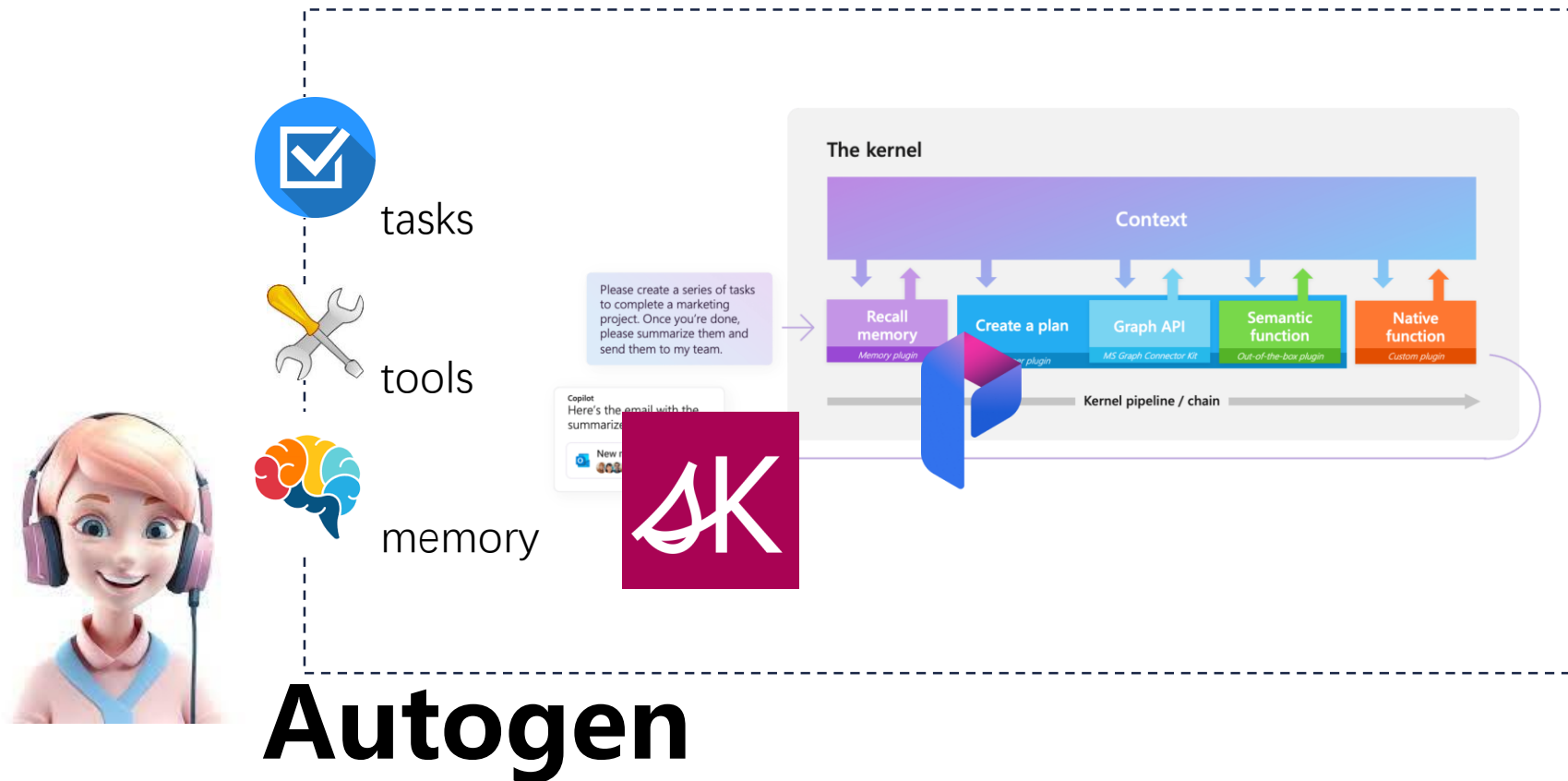
▶ Semantic Kernel 构建智能体



PART 03

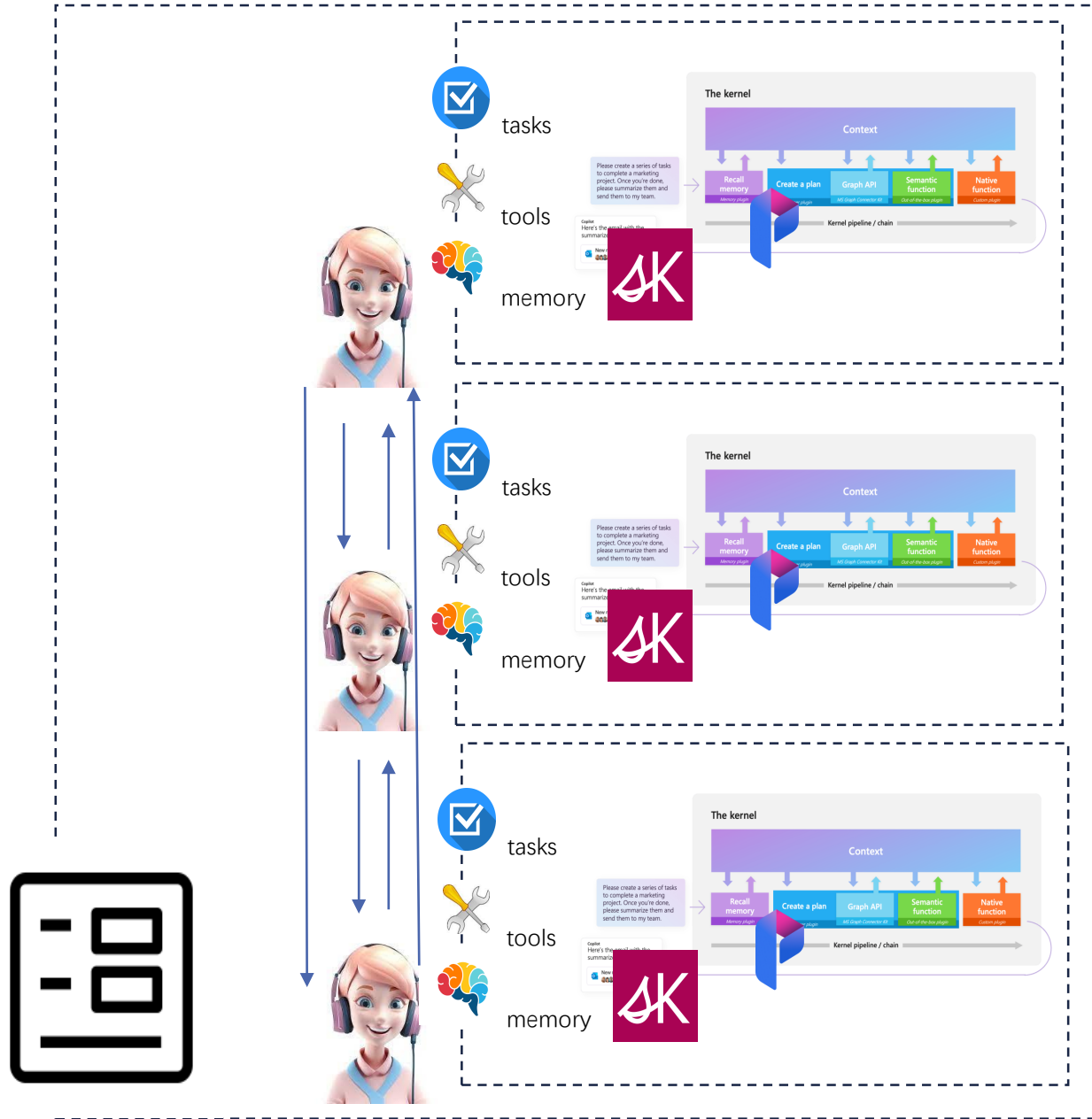
智能体的应用场景

▶ 单智能体



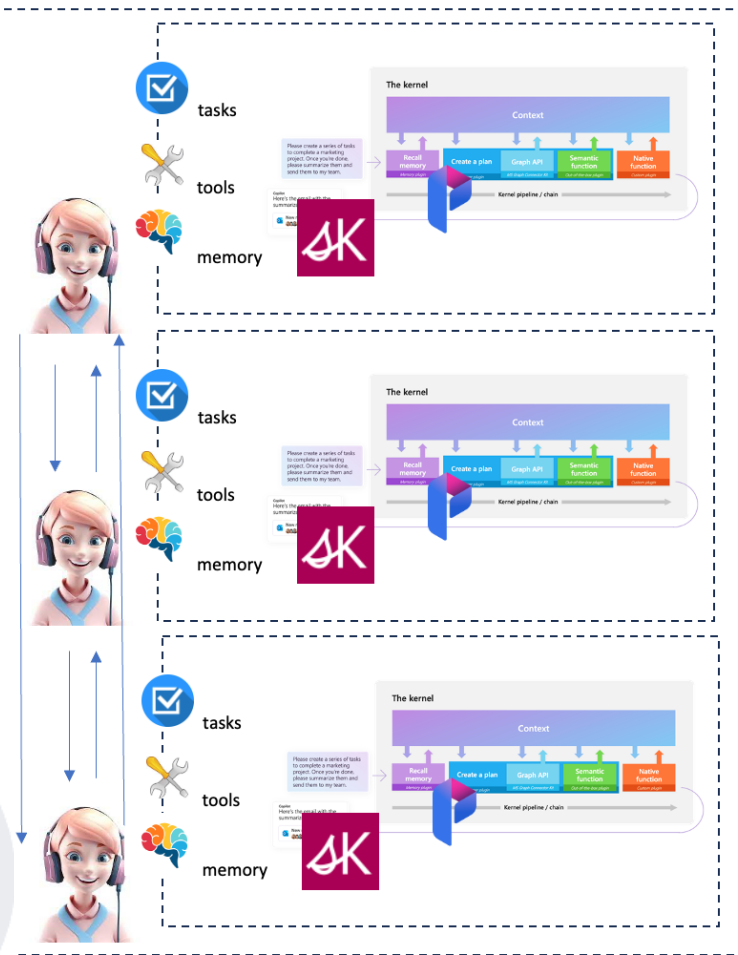
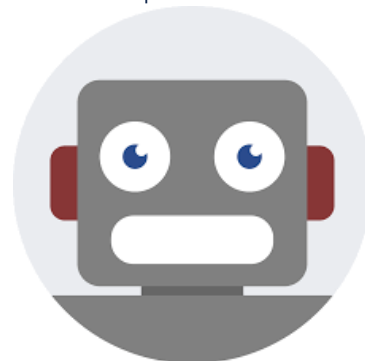
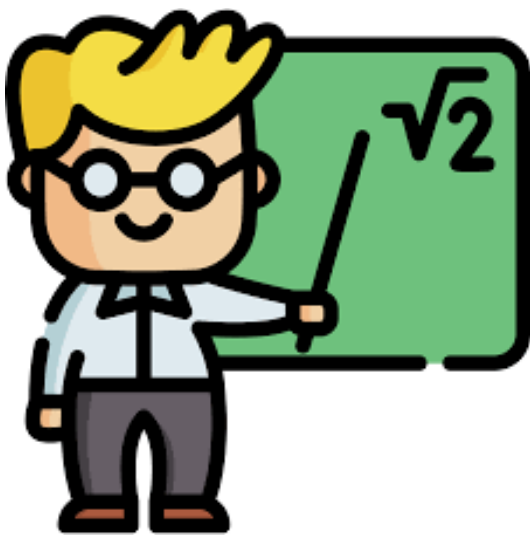


多智能体



Autogen

混合智能体场景



Autogen

▶ 博客助理的思考

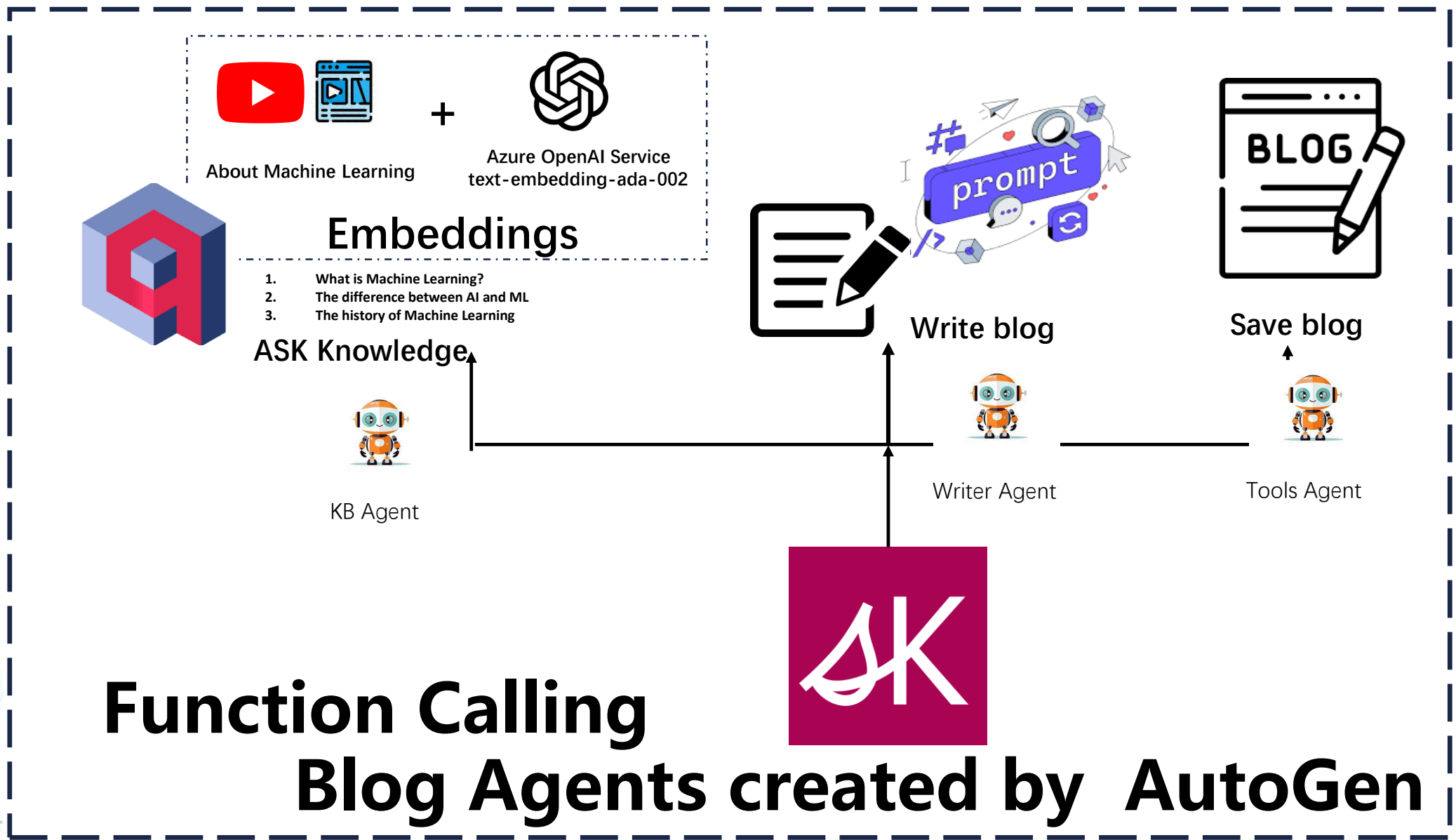
Write about Machine Learning Blog



1. What is Machine Learning?
2. The difference between AI and ML
3. The history of Machine Learning

Writer





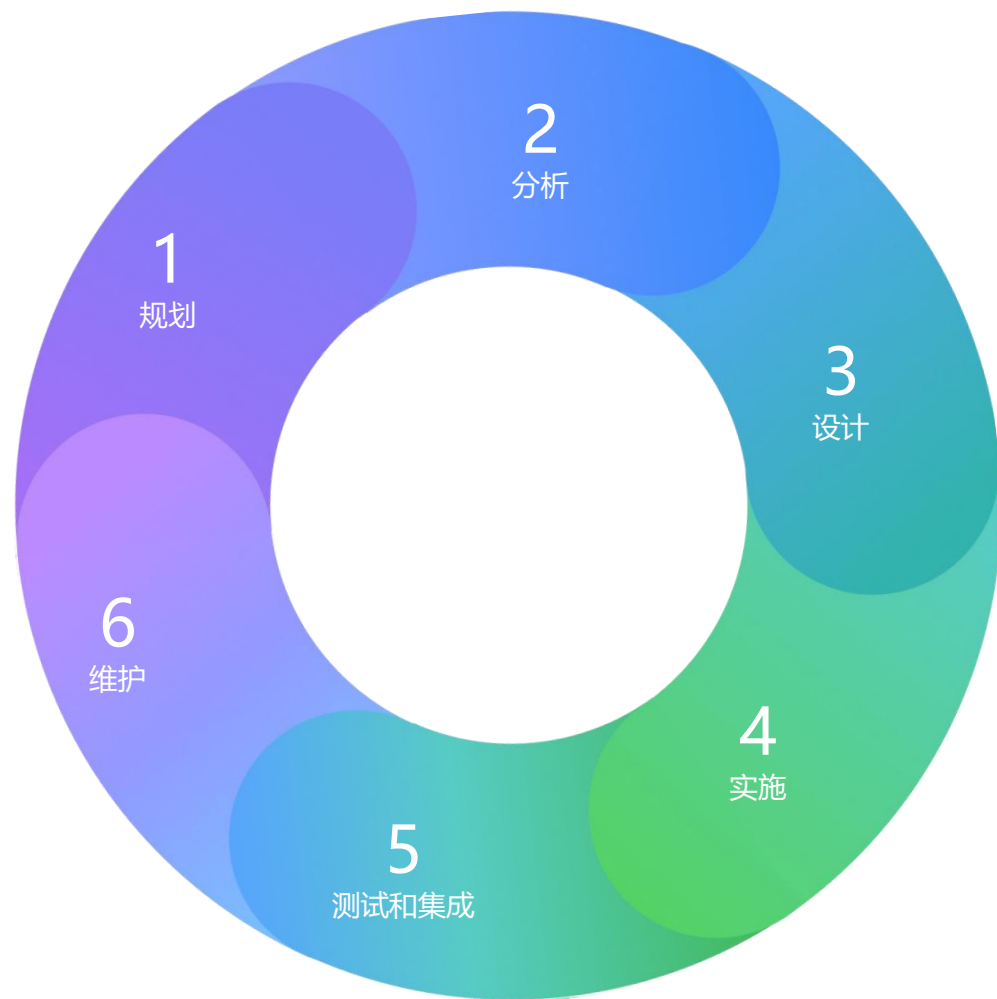
Function Calling

Blog Agents created by AutoGen

PART 04

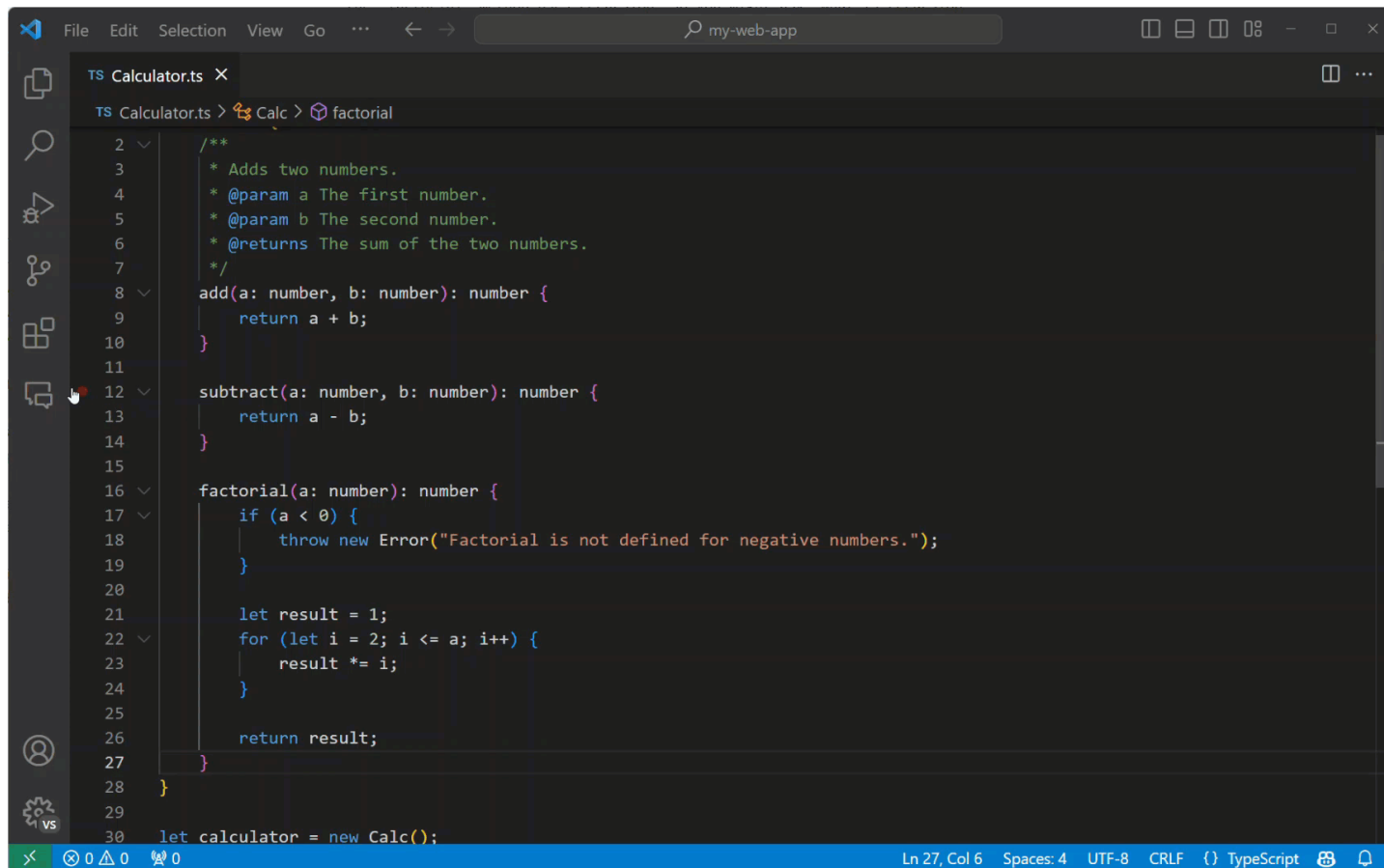
研发场景的实践

▶ 在 AI 加持下让我们的研发更顺畅了



▶ GitHub copilot chat 中的 agent

Copilot 聊天智能体可以让 Copilot 自行查找所需的上下文，而无需您的参与
VS Code 中内置了三个代理：
@terminal
@vscode
@workspace



```
TS Calculator.ts X
TS Calculator.ts > Calc > factorial
2  /**
3   * Adds two numbers.
4   * @param a The first number.
5   * @param b The second number.
6   * @returns The sum of the two numbers.
7   */
8   add(a: number, b: number): number {
9     return a + b;
10  }
11
12  subtract(a: number, b: number): number {
13    return a - b;
14  }
15
16  factorial(a: number): number {
17    if (a < 0) {
18      throw new Error("Factorial is not defined for negative numbers.");
19    }
20
21    let result = 1;
22    for (let i = 2; i <= a; i++) {
23      result *= i;
24    }
25
26    return result;
27  }
28  }
29
30  let calculator = new Calc();
```

▶ GitHub copilot chat agent

GavinGu07
create a word add-in

GitHub Copilot

Creating a Word add-in involves several steps. Here's a step-by-step guide:

- 1. Set up your development environment:** Ensure you have Node.js and npm installed. You also need to install Yeoman and the Office Add-in project template which can be done using the command `npm install -g yo generator-office`.
- 2. Create the add-in project:** Use Yeoman to create the add-in project. You can do this by running `yo office` in your terminal. This will prompt you to answer several questions about your add-in (name, type, etc.). For the type of add-in, choose `Office Add-in Task Pane` project using `jQuery`.
- 3. Explore the project:** The project created by Yeoman includes several files. The most important ones are `manifest.xml` which describes the add-in, and `./src/taskpane/taskpane.html`, `./src/taskpane/taskpane.css`, and `./src/taskpane/taskpane.js` which are the HTML, CSS, and JavaScript files for the task pane of your add-in.
- 4. Modify the add-in:** You can modify the HTML, CSS, and JavaScript files to change the appearance and behavior of your add-in.
- 5. Test the add-in:** You can test your add-in by sideloading it in Word. To do this, you first need to start the local web server by running `npm start` in your terminal. Then, in Word, go to the Insert tab, click My Add-ins > Add a Custom Add-in > Add from File, and then navigate to the `manifest.xml` file in your project.

Here's the code for creating the add-in project:

```
# Install Yeoman and the Office Add-in project template
npm install -g yo generator-office

# Create the add-in project
yo office
```

Remember to choose `Office Add-in Task Pane` project using `jQuery` when prompted for the type of add-in.

GavinGu07
@workspace /new a word add-in

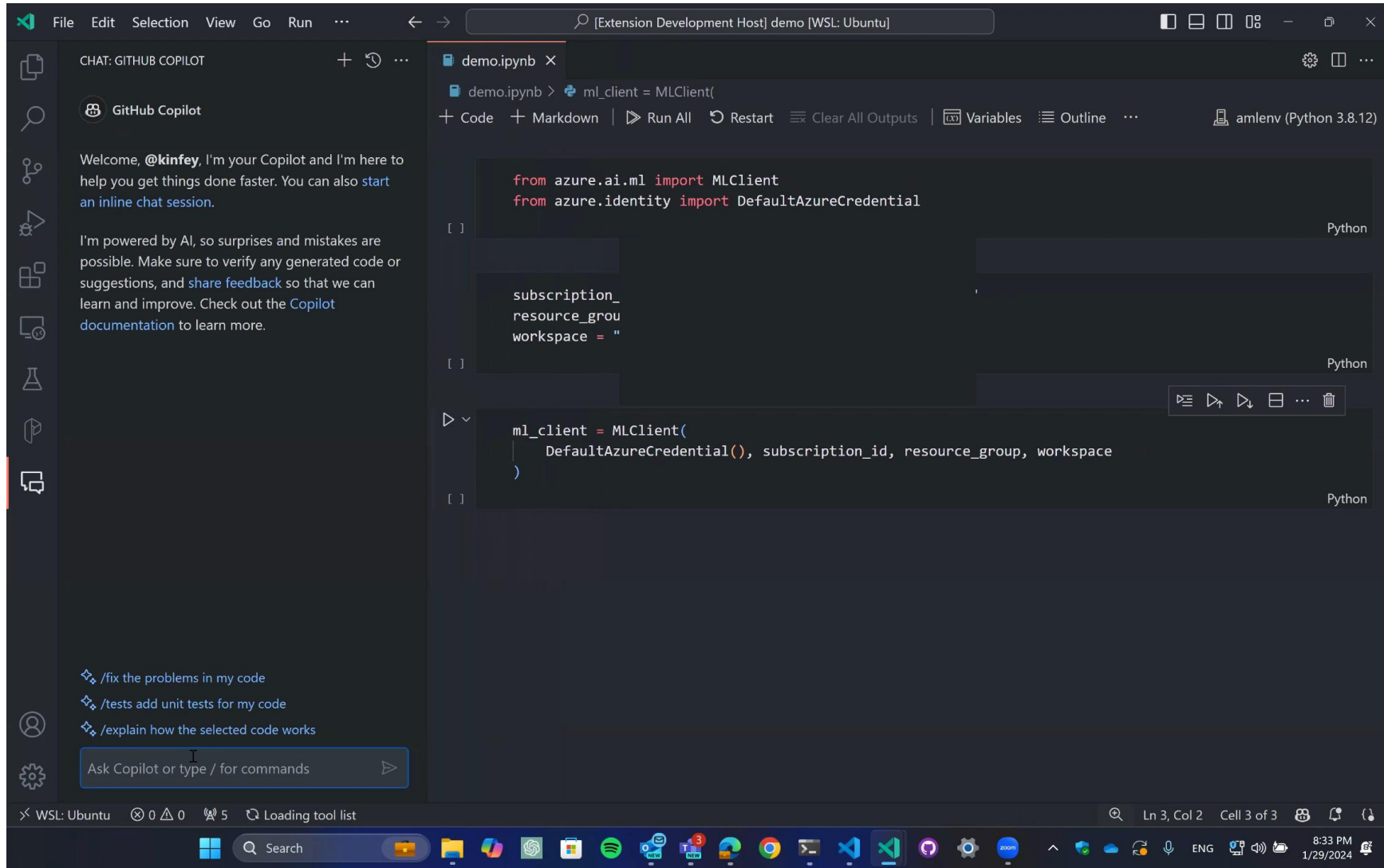
GitHub Copilot used @workspace /new

Sure, here's a proposed directory structure for a Word Add-in project:

```
└─ src
  └─ taskpane
    ├── taskpane.html
    ├── taskpane.js
    └─ taskpane.css
  └─ commands
    ├── commands.html
    ├── commands.js
    └─ commands.css
  └─ functions
    ├── functions.html
    ├── functions.js
    └─ functions.json
  └─ assets
    └─ icon-32.png
  └─ manifest.xml
  └─ package.json
  └─ README.md
```

Create Workspace

自定义 GitHub copilot agent



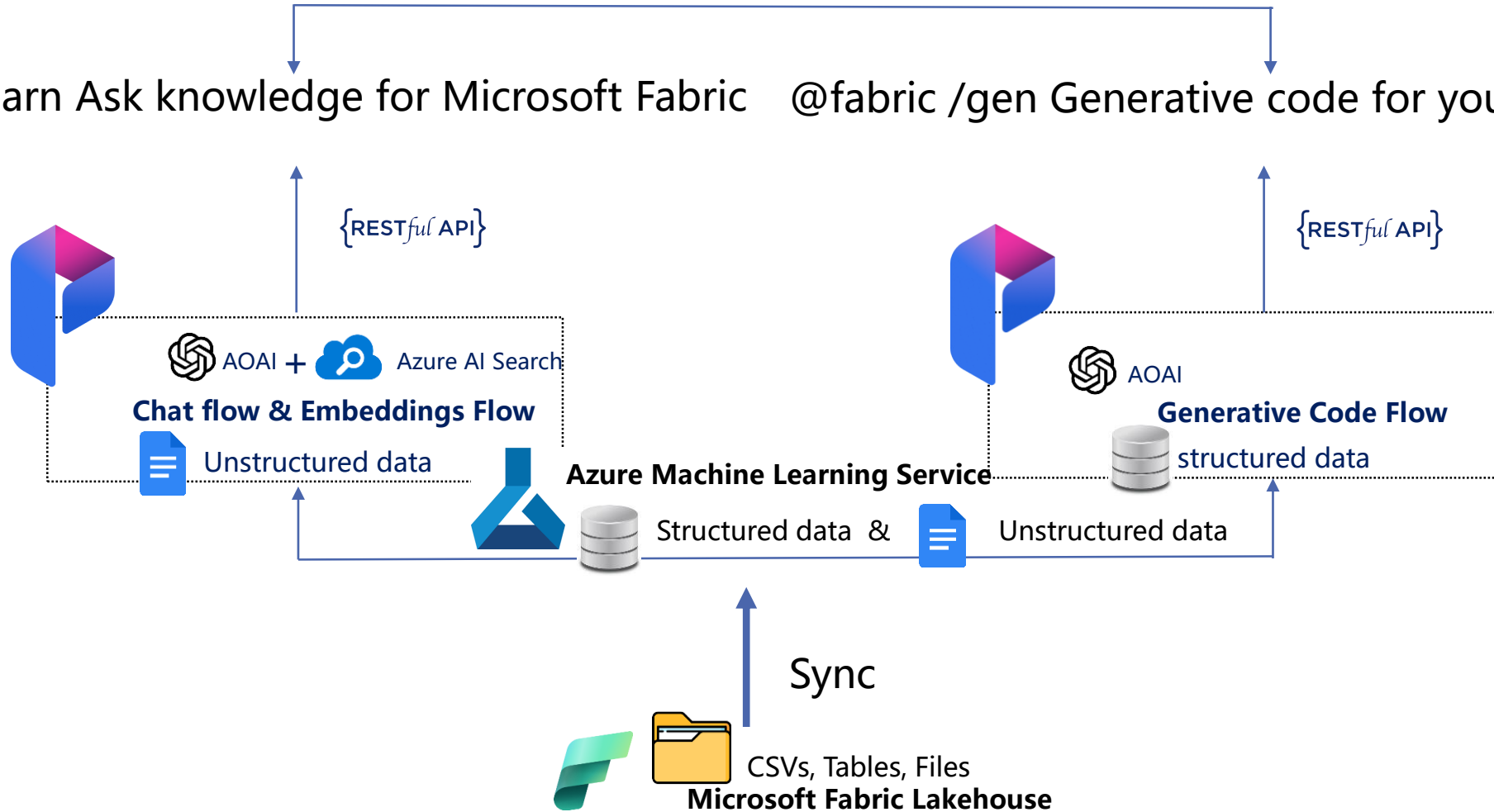


架构



GitHub Copilot Chat Copilot API

@fabric /learn Ask knowledge for Microsoft Fabric @fabric /gen Generative code for your structured data



科技生态圈峰会 + 深度研习



—1000+ 技术团队的选择



 **K+峰会**  **上海站**
K+ 全球软件研发行业创新峰会
时间: 2024.06.21-22

 **K+峰会**  **敦煌站**
K+ 思考周®研习社
时间: 2024.10.17-19

 **K+峰会**  **香港站**
K+ 思考周®研习社
时间: 2024.11.10-12



K+峰会详情



 **AiDD峰会**  **上海站**
AI+研发数字峰会
时间: 2024.05.17-18

 **AiDD峰会**  **北京站**
AI+研发数字峰会
时间: 2024.08.16-17

 **AiDD峰会**  **深圳站**
AI+研发数字峰会
时间: 2024.11.08-09



AiDD峰会详情

THANKS

