

# .NET+AI: 技术栈全景 解密



# 目录

## CONTENTS



扫码获取本次分享的课件和代码示例

01 | 破冰

02 | 术语速通

03 | 技术栈全景

04 | 实操演练

# 实现以下功能

你要怎么做？需要多久？



## 地址提取

张三，广东省深圳市南山区大冲城市花园  
A栋1108，181xxxx1212。



## 英语外教APP

开发一个口语对话APP。



## 客服系统

开发一个智能客服系统。

# AI 时代的新解法

编程思维的转变

从“面向语法/过程”（告诉计算机每一步怎么做）到“面向语义/意图”（告诉计算机我想要什么）



## 面向语法编程

基于编程语言的语法  
来编写功能模块  
类似面向遥控器编程。



## 面向语义编程

用自然语言（提示词）  
来驱动功能实现  
类似面向智能音箱编程。



01

破冰：走出三大迷思

# 开场三问：你中招了吗？

三个普遍焦虑，直击传统.NET开发者的AI认知误区



## 必须用Python?

认为AI开发是Python的专利，.NET开发者只能望洋兴叹。



## 生态不成熟?

担心微软AI生态落后，缺乏企业级生产就绪的解决方案。






## 现在学已晚?

觉得AI浪潮已过，现在入场成本高昂且错失良机。




# 迷思一：“搞AI必须用Python？”

真相：Python适合研究，.NET更适合生产。

## Python

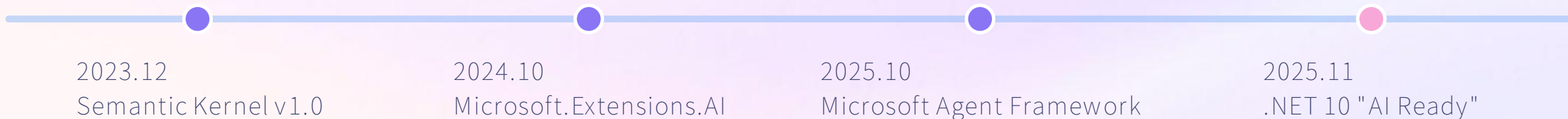
-  性能：解释型语言，执行效率相对较低。
-  类型安全：动态类型，运行时易出错。
-  企业集成：与企业现有技术栈集成需要额外适配。

## .NET

-  性能：编译型 + JIT，企业级性能表现。
-  类型安全：强类型系统，开发期捕获错误。
-  企业集成：无缝集成 现有.NET生态与工具链。

## 迷思二：“微软AI生态不成熟？”

从 Semantic Kernel 到 Microsoft Agent Framework 的演进之路



2年 生态逐步完善

全线 企业级稳定

完整 技术栈覆盖

## 迷思三：“学习成本太高？”

真相：你不是从零开始，而是站在现有技能的肩膀上。

Python开发者  
3个月+

学语言 + 学框架 + 学AI

VS

.NET开发者  
1个月

仅需学习AI框架本身

你的C#技能可直接复用：

依赖注入 → AI服务注册

中间件 → AI管道处理

异步编程 → AI流式响应

02

术语速通：读懂AI黑话

# AI核心术语速通：基础层四件套

30秒建立最小可用知识框架



## LLM (大语言模型)

读了整个互联网的超级助手，提供基础AI能力。



## Prompt (提示词)

AI时代的编程语言，用于控制AI行为。



## Token (令牌)

AI世界的计量单位，决定了使用成本。



## Context Window (上下文窗口)

AI时代大模型的短期记忆



# AI核心术语速通：应用层四把钥匙

每把钥匙对应一个具体的企业痛点



## Function Calling

大模型的外挂系统，让AI从“只能说”到“能做事”。  
让AI能调用外部函数/工具，是实现Agent的核心。



## Agent (智能体)

从回答到解决问题的跨越，能自主规划和执行任务。



## Embedding (嵌入)

把文字变成坐标点，用于语义搜索和相似度计算。



## RAG (检索增强)

给AI配个图书馆，解决AI不知道企业知识的问题。



## Fine-tuning (微调)

给通用AI做岗位培训（用自己的数据训练模型），让它成为领域的专家。

# AI核心术语速通：工程层四件套

聚焦落地，用驾驶隐喻构建叙事



Prompt Engineering  
写好“需求文档”的艺术



MCP  
“插件商店”式工具连接

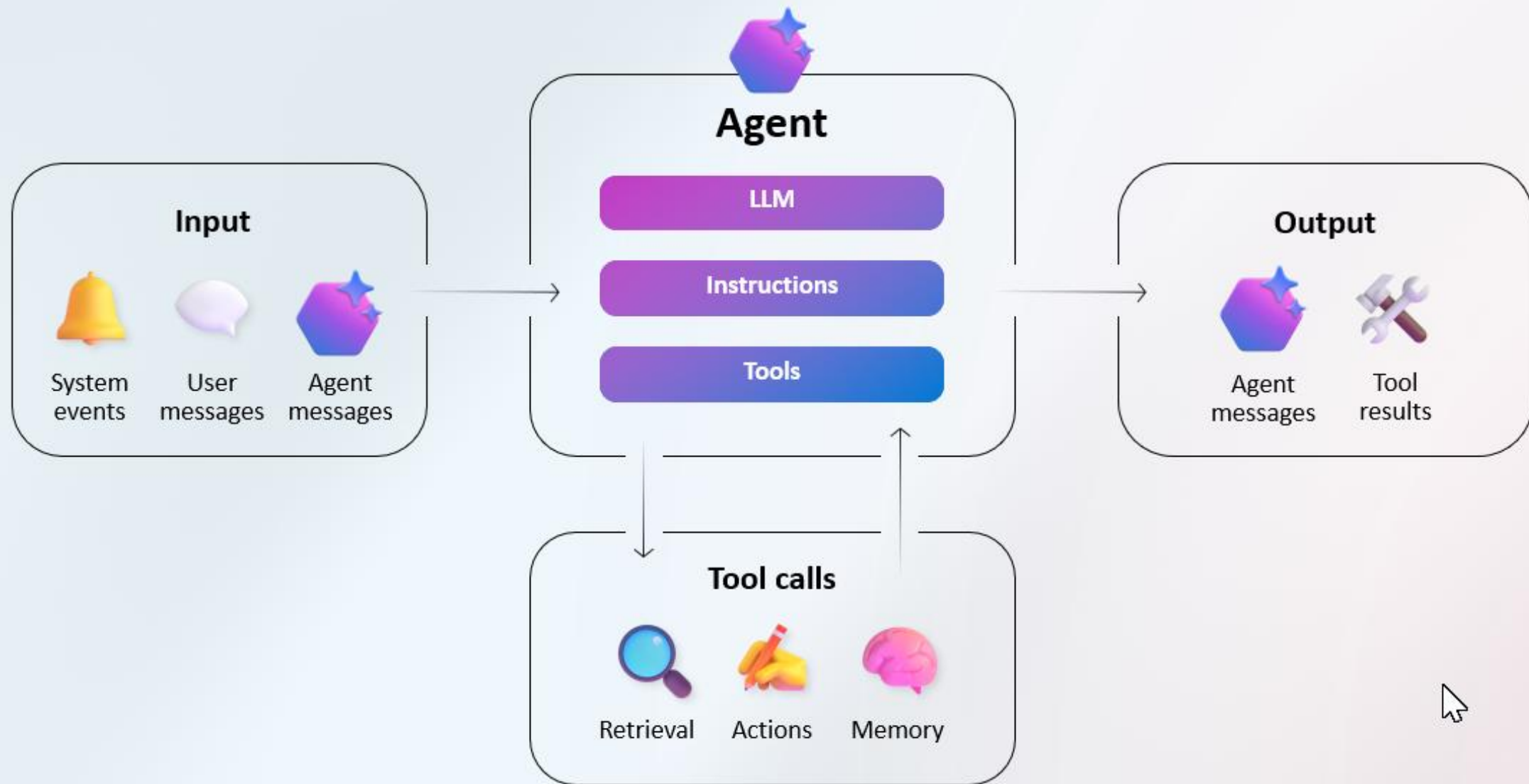


A2A  
“车队”式协作



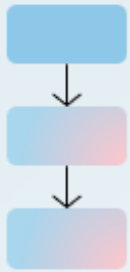
Workflow  
“导航图”式流程编排

# What is an agent?

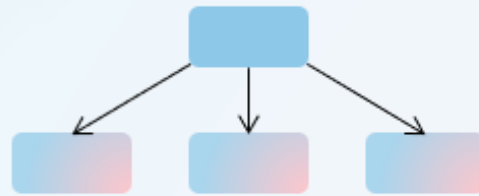


# Multi-Agent Orchestration Patterns

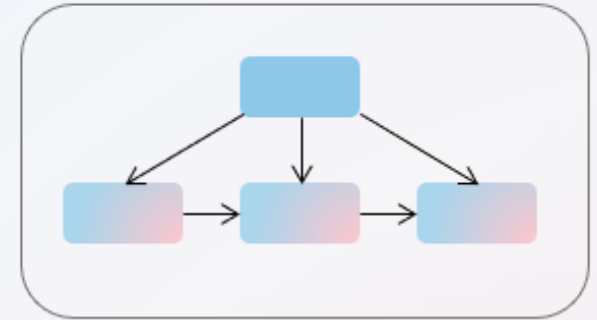
Sequential



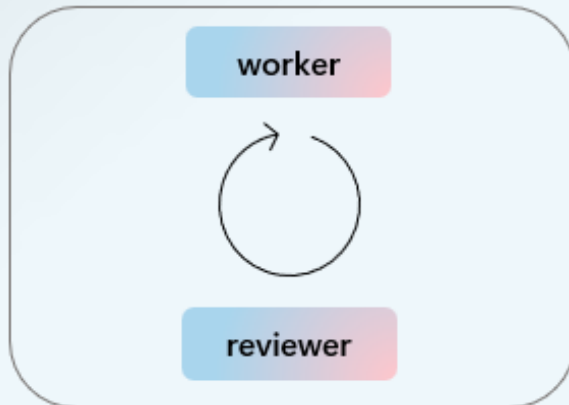
Concurrent



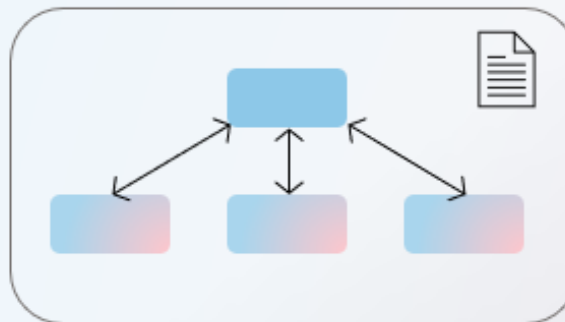
Handoff



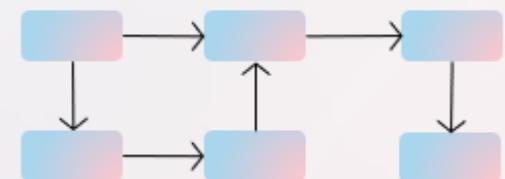
Group Chat



Magentic



Workflow Process



# 技术演进路线图：从引擎到落地



LLM是引擎 (提供AI能力)



Prompt是方向盘 (控制AI行为)



Function Calling是外挂 (让AI能干活)



Agent是司机 (自主完成任务)



A2A是车队 (团队协作)



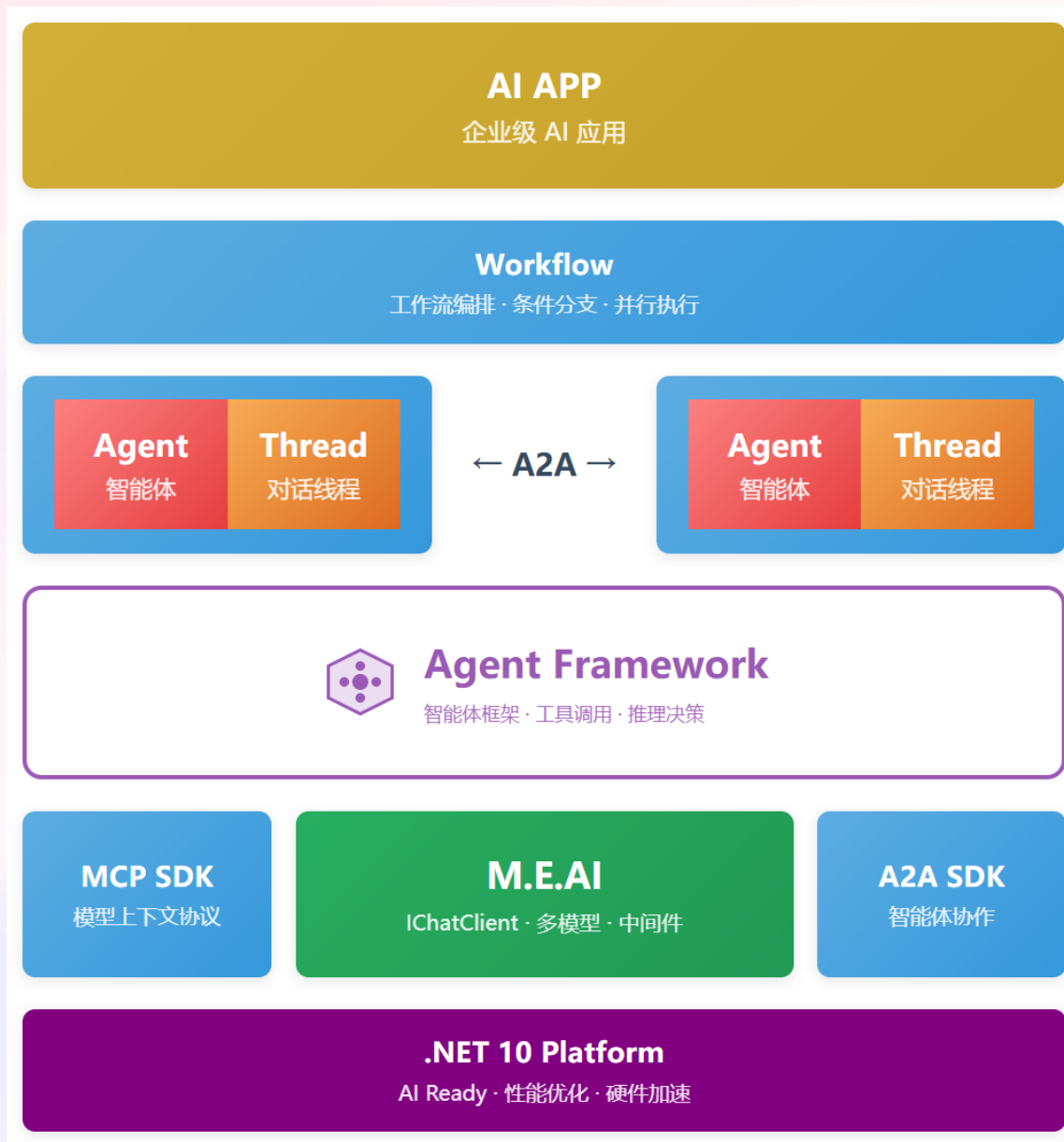
Workflow是导航 (流程编排)

03

.NET+AI 技术栈全景

# .NET AI技术栈全景

自底向上，层层递进，构建企业级AI应用





# .NET 10 - AI Ready 平台

这不是简单的版本迭代，而是为AI时代打造的 平台级升级。



AI推理性能优化：针对大模型推理场景进行运行时优化。

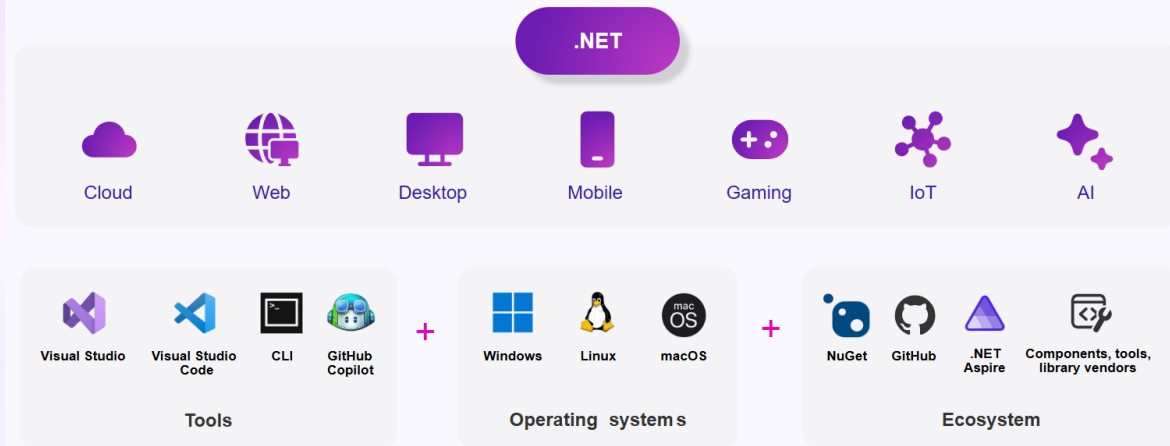


GPU/NPU硬件支持：原生支持异构计算，加速AI任务。



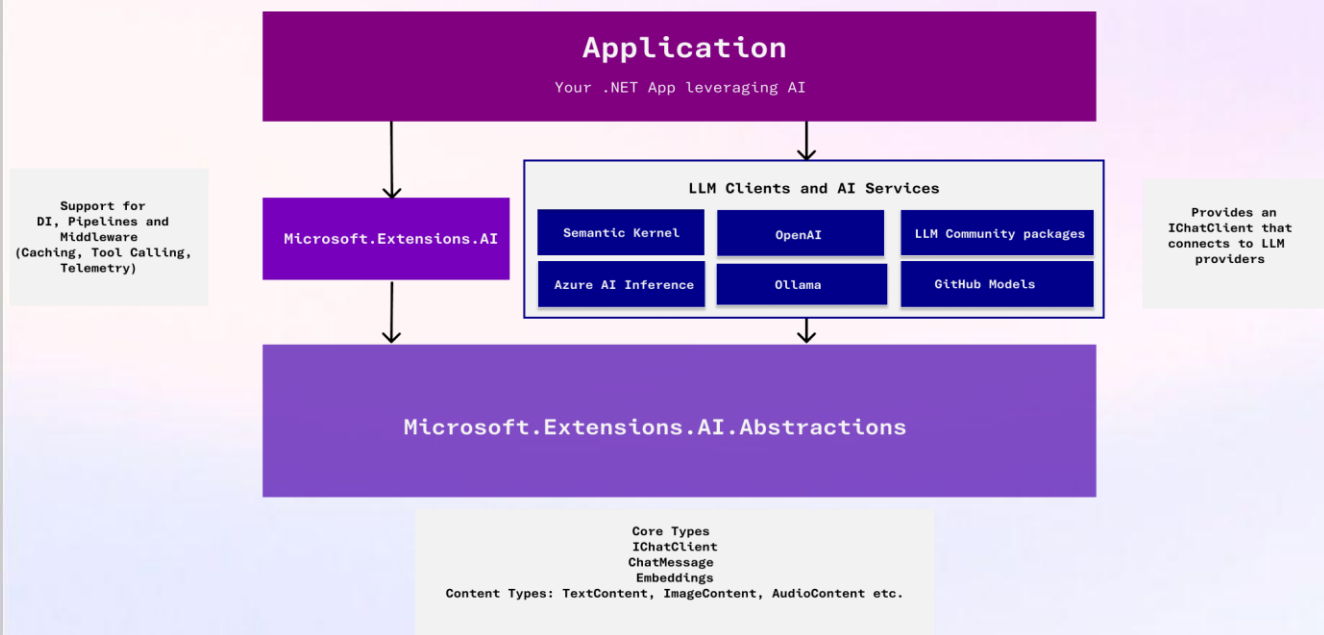
AI调试与追踪：提供从代码到模型的全链路调试工具。

## Build anything with a unified platform

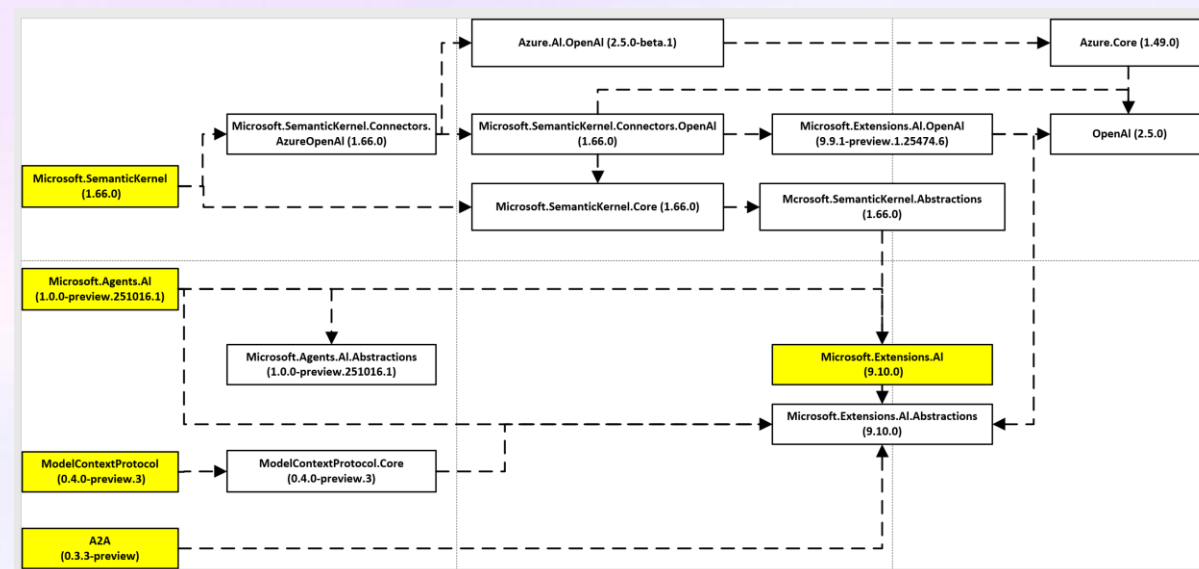


# M.E.AI - 统一抽象层

实现“一次编写，多模型运行”。



- ➡ 多模型支持：OpenAI、Azure、DeepSeek无缝切换。
- ≡ 中间件管道：日志、缓存、重试等AOP能力。
- ⚡ 流式响应：支持AI的打字机效果，提升用户体验。

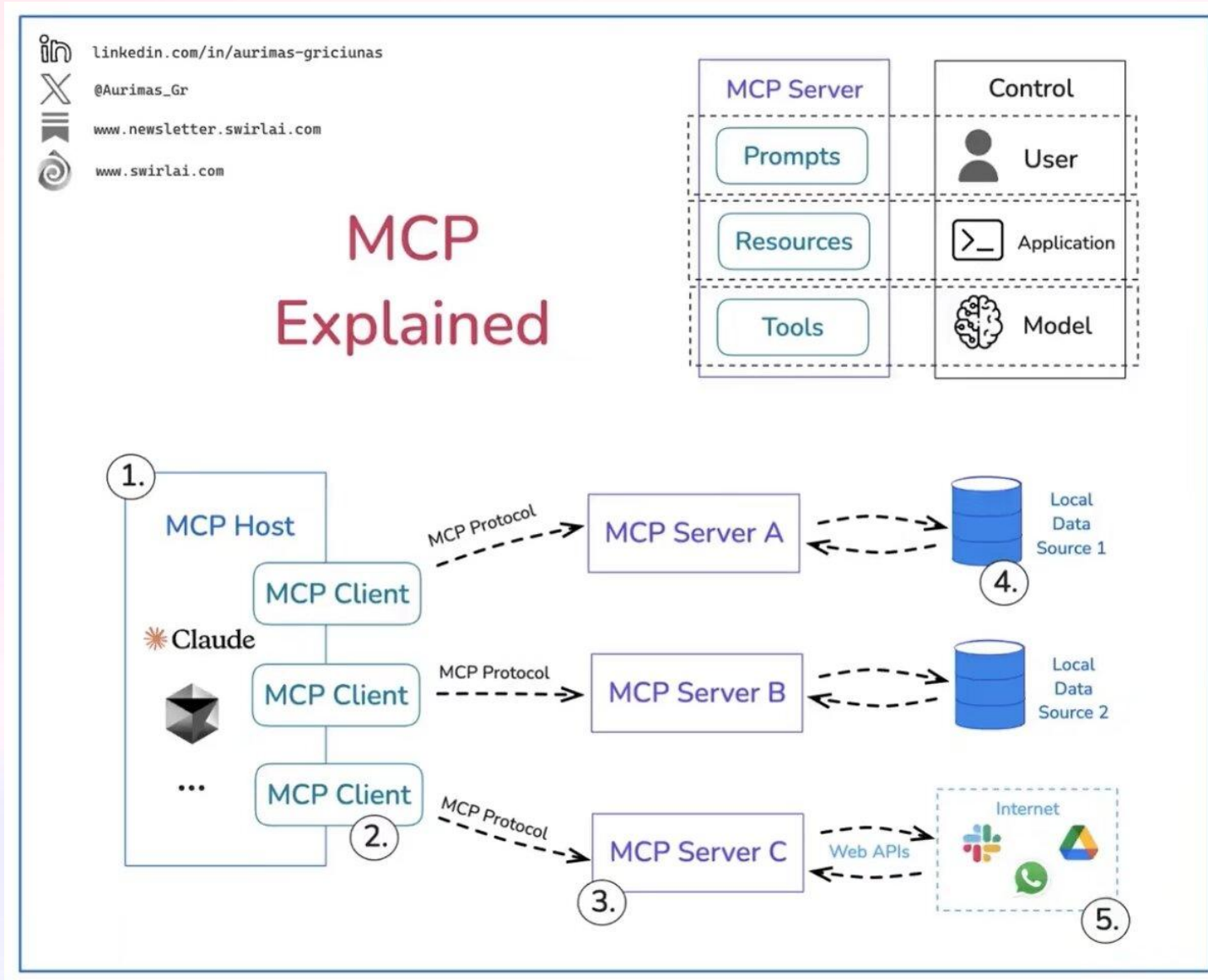


<https://github.com/dotnet/extensions>

# MCP - 工具连接

Model Context Protocol，让AI连接外部工具的“USB接口”。

通过统一协议，AI可以安全、标准化地调用外部数据和工具，从“只会聊天”变成“能干活”。



# A2A：智能体协作

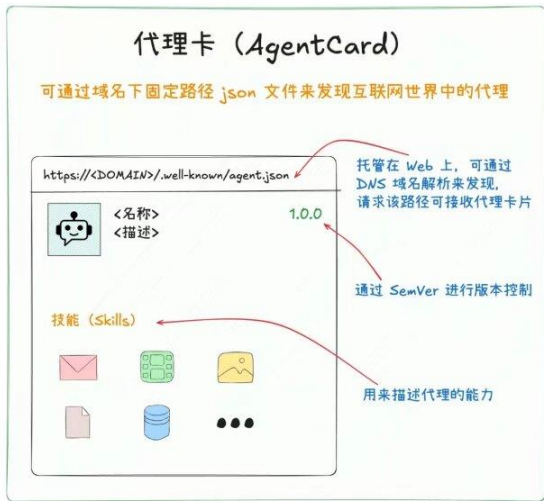
A2A, Agent to Agent, 让多个智能体像团队一样协作，解决单个AI无法完成的复杂任务。

通过统一协议，Agent可以安全、标准化地与外部Agent通信和协助。

<https://github.com/a2aproject/a2a-dotnet>

## Agent2Agent 简介 @lencx

文档: <https://google.github.io/A2A>  
GitHub: <https://github.com/google/A2A>





# AG-UI: 用户代理交互协议

AG-UI, Agent-User Interaction Protocol, 用户代理交互协议, 由CopilotKit公司推出的, 旨在标准化AI智能体 (Agent) 与用户界面 (UI) 之间进行实时、双向、结构化通信的开放标准。其核心是定义了一个共享的事件模式, 用于:

- 实时流式响应
- 工具调用和结果处理
- 代理与用户界面之间的共享状态更新
- 将用户界面组件与代理活动同步

<https://dojo.ag-ui.com/microsoft-agent-framework-dotnet>

<https://github.com/a2aproject/a2a-dotnet>



AG-UI Interactive Dojo

INTEGRATIONS

Microsoft Agent Framework... ▾

VIEW

Preview

</> Code

Docs

DEMOS

Agentic Chat

Chat with your Copilot and call frontend tools

ChatToolsStreaming

Backend Tool Rendering

Render and stream your backend tools to the frontend.

Agent StateCollaborating

Human in the loop

Plan a task together and direct the Copilot to take the right steps

HITLInteractivity

Tool Based Generative UI

Haiku generator that uses tool based generative UI.

Generative ui (action)Tools

Shared State between agent and UI

A recipe Copilot which reads and updates collaboratively

Agent StateCollaborating

dojo.ag-ui.com

Microsoft Agent Framework (.NET)

Integration ID: microsoft-agent-framework-dotnet

# Microsoft's agent frameworks

Orchestrate agents with AutoGen, Semantic Kernel, Foundry SDK and M365 Agents SDK



**AutoGen**

State-of-the-art  
multi-agent  
research SDK



**Semantic Kernel**

OSS AI service  
and agent  
orchestrator



**Foundry SDK**

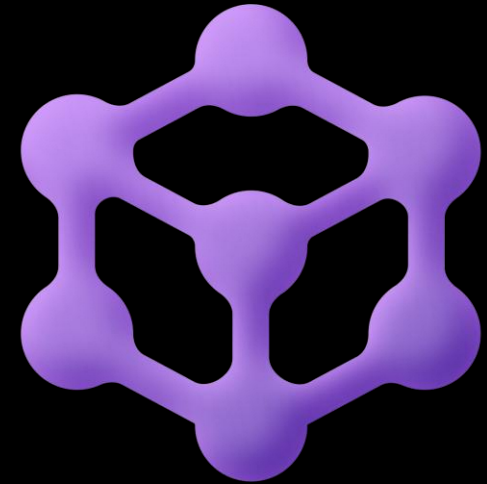
API for Foundry  
Agent Service



**M365 Agent SDK**

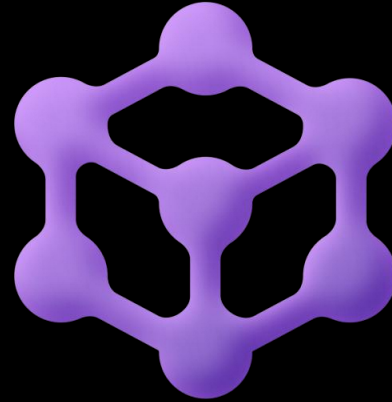
API for Foundry  
Agent Service

Unified Agent  
Development



**Microsoft Agent  
Framework**





# Microsoft Agent Framework

## AI and Agent Orchestration

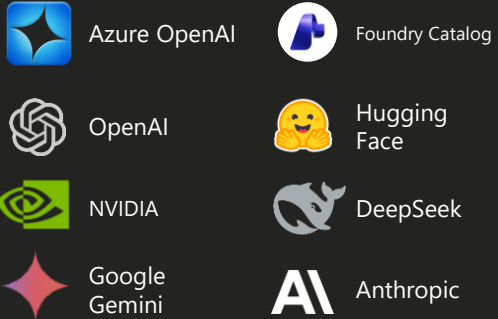


.NET



Python

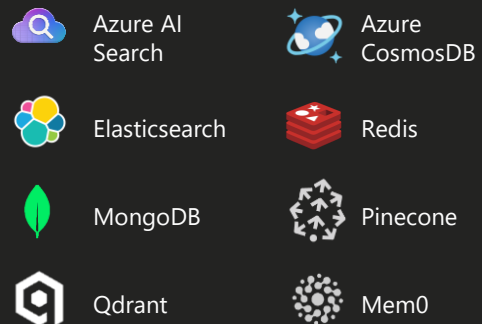
### AI Services



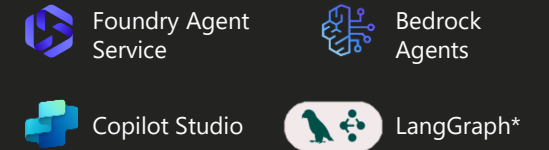
### Local models



### Memory Services



### Agent Services



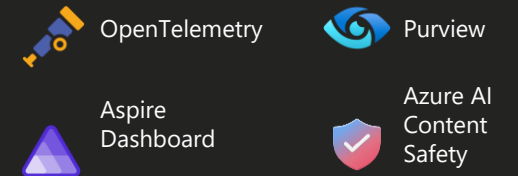
### Plugins



### UI Frameworks

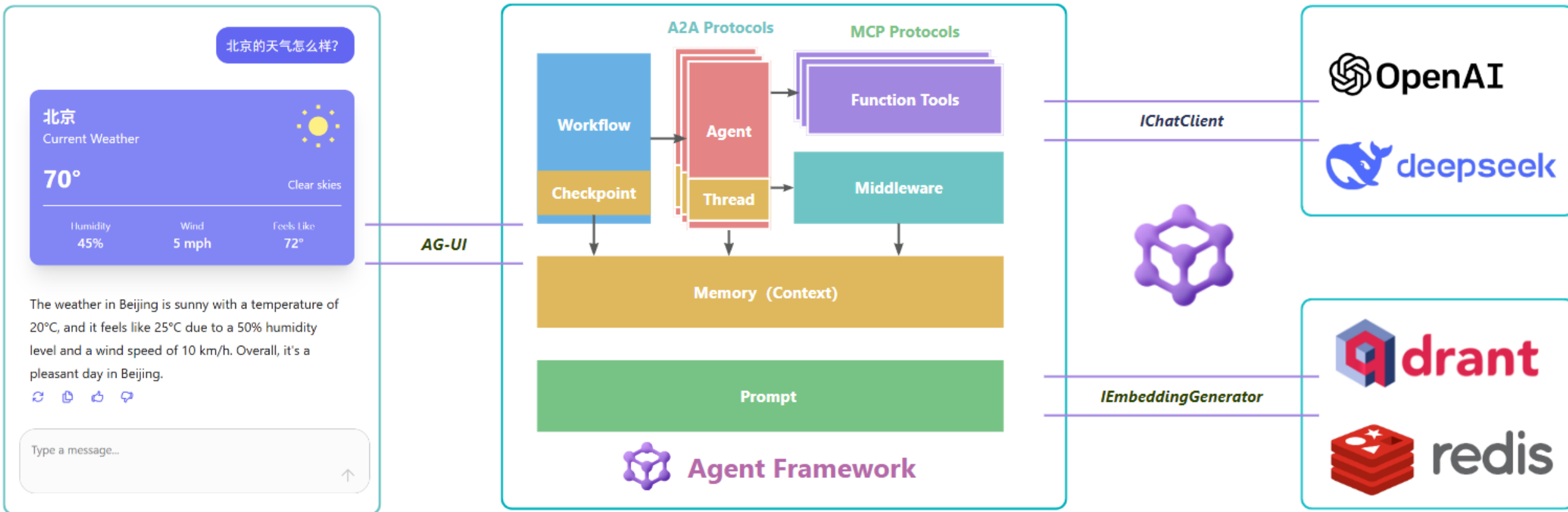


### Filters and telemetry

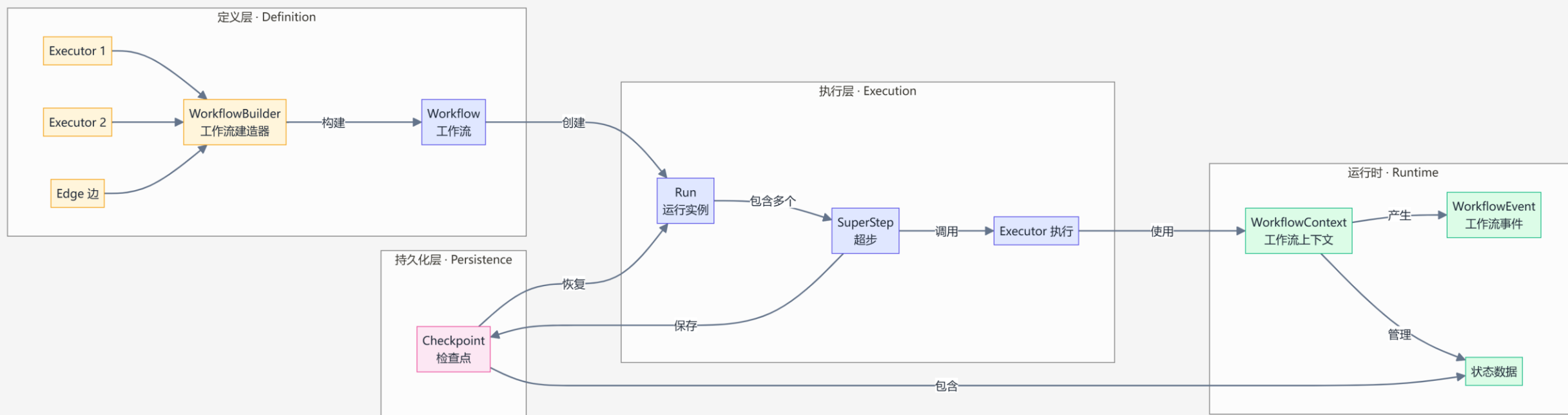


# MAF - 智能体框架

MAF = AutoGen (研究成果) + Semantic Kernel (生产能力), 企业级Agent首选。



# MAF - Workflow



# MAF - DevUI

DEV-UI：提供了面向开发者的Agent/Workflow 交互式测试的网页界面，方便开发者对其进行可视化调试、测试和迭代优化。

The screenshot displays the Dev UI interface for an "Email Spam Detector" workflow. The interface is divided into several sections:

- Header:** Shows "Dev UI v1.0.0b251204" and a dropdown menu for "Email Spam Detector".
- Workflow Diagram:** A visual representation of the 4-step email classification workflow. The steps are:
  - email\_preprocessor** (EmailPreprocessor): Output is a JSON object with "original\_message": "Hi there, are you interested in".
  - spam\_detector** (SpamDetector): Output is a JSON object with "email\_content": { "original\_message": "Hi there, are you interested in" }.
  - spam\_handler** (SpamHandler) and **legitimate\_message\_ha...** (LegitimateMessageHandler): Both receive input from the spam\_detector. The spam\_handler output is a JSON object with "original\_message": "Hi there, are you interested in".
  - final\_processor** (FinalProcessor): Receives input from both the spam\_handler and the legitimate\_message\_ha... component.
- Execution Timeline:** A list of execution steps with timestamps and status:
  - email\_preprocessor**: 10:57:28 AM, completed.
  - spam\_detector**: 10:57:31 AM, completed.
  - spam\_detector**: 10:57:41 AM, completed (Run #2).
  - spam\_handler**: 10:57:45 AM, completed.
  - final\_processor**: 10:57:49 AM, completed.
  - Workflow Complete**: 10:57:49 AM, completed.
- Final Output:** A text box showing the final result: "Email classified as SPAM (human-verified). AI detected: SPAM (confidence: 0.70). Human reviewer: spam. Spam indicators: spam\_keywords: ['offer', 'click here', 'urgent'], suspicious\_patterns. Action: Message quarantined for review. Processing time: 3.7s".
- Events:** A list of 26 events (46 raw) showing the sequence of events during execution, including "workflow\_event.completed", "output\_item.done", and "output\_item.added".
- Buttons:** "Run Again" and "Deployment Guide" buttons are visible at the bottom.

<https://github.com/microsoft/agent-framework>

## 总结：三大认知刷新



.NET不仅能做AI  
而且可以做的很好

而且更适合生产环境，性能、安全、集成度全面领先。



.NET AI 生态  
已成熟

全线企业级稳定性，生产就绪。



学习门槛比想象  
低的多得多

现有C#技能直接复用，1个月即可上手，  
站在自己的肩膀上。

04


实战演示：要比想象的简单


# .NET+AI 公开资料包



## .NET+AI 技术全景解密

💡 推荐指数: ★★★★★

 [.NET+AI: 技术全景解密.pdf](#) 12.4MB


 [智能体设计模式.pdf](#) 5.5MB



## :: .NET+AI | MAF 入门指南

💡 推荐指数: ★★★★★

下载该Polyglot Notebook 课件，带你迅速体验从M.E.AI基础到Agent开发和Workflow编排。

 [maf-workshop.zip](#) 12.1MB



# 行动号召

AI 并不可怕，可怕的是身边的人比你更会使用AI。-- 黄仁勋

AI不会取代.NET开发者，但会AI的 .NET 开发者会走的更远。

1

第1周：基础  
Hello M.E.AI

2

第2周：进阶  
MCP&A2A

3

第3周：实战  
Agent & Workflow

4

第4周：应用  
项目实战

让我们一起， 拥抱.NET + AI的新时代！

# 感谢您的观看

# THANK YOU

