


**.NET 9 + AI =**  

张善友

微软最有价值专家

广东智用人工智能应用研究院有限公司 CTO

# Agenda

**ML.NET 4.0**

**TensorFlow.NET 和 TorchSharp**

**ONNX 运行时原生支持**

**OpenAI SDK 集成**

**新的Numerics APIs**

**生成式AI 和.NET 社区**

2022年的程序员



VS

2025年的程序员



# Who will develop AI in products?

2022

Data Scientists

2023

Data Scientists

GenAI  
Enthusiasts

2024

Data Scientists

GenAI  
Enthusiasts

Developers

2025

Data  
Scientists

GenAI  
Enthusiasts

Developers

GenAI Enthusiasts = Hardcore firstcomers and LLM integration community

Developers = Mainstream developers

<https://medium.com/floomai/get-ready-for-ai-gateways-45e4dca22675>

# 全场景开发平台: .NET





# ML.NET 4.0

1. **为 .NET 开发者生成：** 无需离开 .NET 生态系统，便可以使用 C# 或 F# 创建自定义 ML 模型
  2. **使用 AutoML 简化自定义ML：** Model Builder(简单的 UI 工具)和 ML.NET CLI
  3. **使用 TensorFlow & 进行扩展：** 使用其他流行的 ML 框架(TensorFlow、ONNX、Infer.NET 等)并访问更多机器学习场景，如图像分类、物体检测等
- AutoML 增强功能：
    - 用于平衡模型选择的多指标优化
    - 支持 AutoML 中的时间序列预测
  - 用于简化模型部署的新 Infer<T> API
  - 将 TensorFlow 和 ONNX 模型转换为 ML.NET 格式以提高性能
  - 新的Microsoft.ML.GenAI： GenAI包提供一系列流行GenAI模型的torchsharp实现，目标是从相应的Python常规模型加载相同的权重：<https://github.com/dotnet/machinelearning/issues/7169>

# TensorFlow.NET 和 TorchSharp

## TensorFlow vs Pytorch

TensorFlow.NET 正在通过高性能 C# 绑定和自动微分支持进行升级

- 使用 cppSharp 生成的新的高性能 C# 绑定
- 自定义 C# 运算的自动微分支持
- Keras API 完全用 C# 实现，允许无缝的模型定义和训练



TorchSharp 是一个 .NET 库，旨在提供对 PyTorch 支持的库的访问功能。其主要目标是将 libtorch 提供的 API 绑定到张量操作上。作为 .NET 基金会的一员，TorchSharp 让 .NET 开发人员能够利用 libtorch 库，直接在他们的代码中创建和训练神经网络。

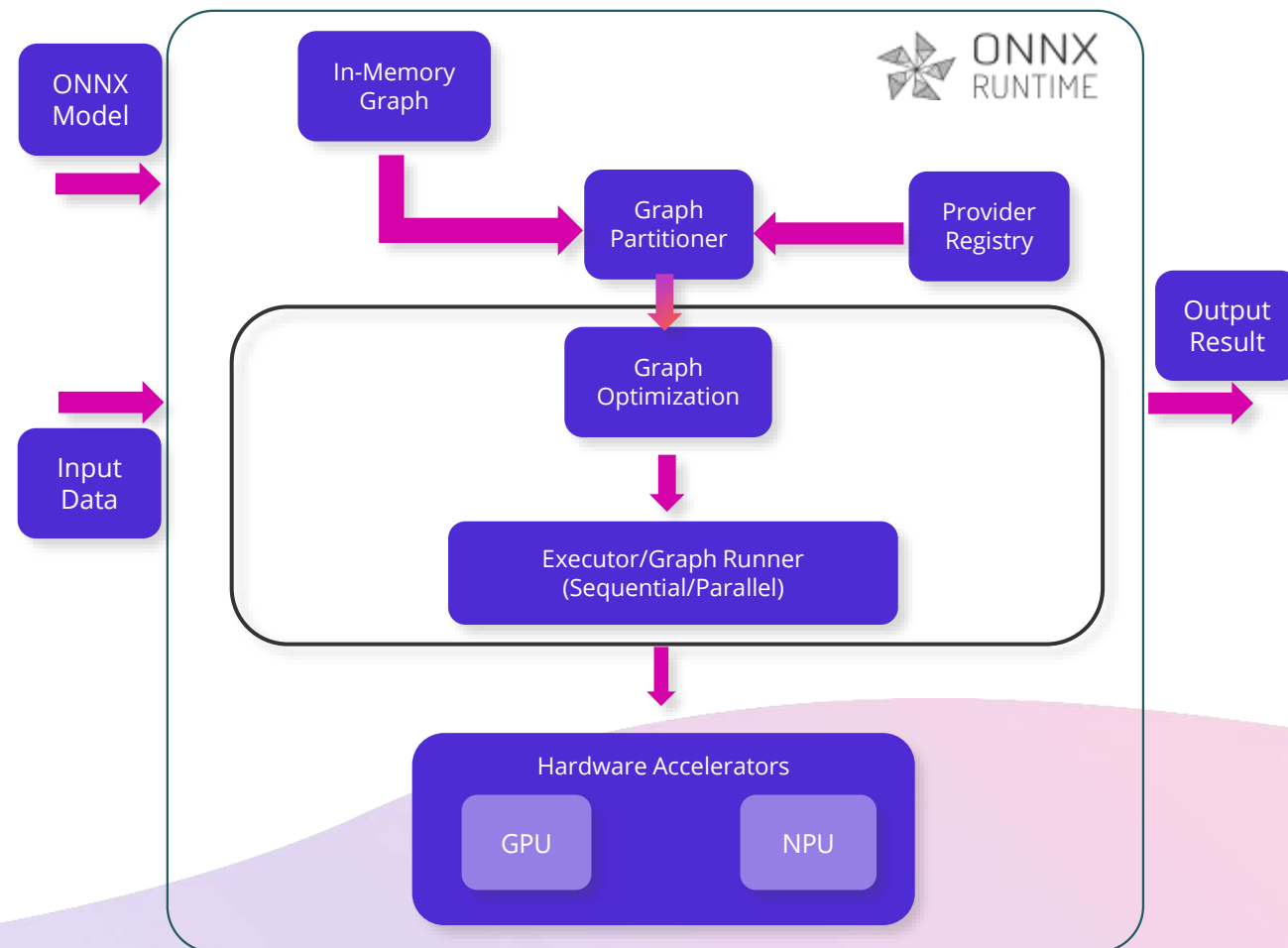
# ONNX 运行时原生支持

**ONNX** = Open and interoperable **file format** for ML and DNN models.

**ONNX Runtime** = Fast and efficient **model inference and training engine** that works across a diverse range of hardware accelerators.

**ORT Generate API (ORT GenAI)** = High performance, **easy-to-use API** for GenAI models

**Olive** = A toolkit for hardware-aware **AI model optimization**.





# ONNX 运行时原生支持

一个专用命名空间 (Microsoft.ML.OnnxRuntime) , 其中包含一个 API 来直接加载和运行 ONNX 模型

- 直接模型加载: `var session = new InferenceSession ("model.onnx") ;`
- 使用 `Span<T>` 和 `Memory<T>` 对输入/输出张量进行高效的内存管理
- 通过统一 API 支持硬件加速 (CPU、GPU、DirectML)

# 新的Numerics APIs

.NET 9 引入了新的数值 API，以实现高效的张量和矩阵运算

- `System.Numerics.Tensor<T>` 用于高效的张量运算
- `System.Numerics.Matrix<T>` 用于矩阵代数
- SIMD 加速线性代数例程
- 与 Nvidia 的 cuDNN 库集成，用于深度学习基元

# OpenAI SDK 集成



OpenAI



## OpenAI library for .NET

### Customer benefits

- 和 OpenAI 合作构建官方的.NET 客户端
- 功能 对齐Python 和 JavaScript/TypeScript 库
- 支持最新的 OpenAI 特性和模型
- 从第一天开始就支持 GPT4o 和 Assistants
- 跨OpenAI 和 Azure OpenAI 的统一体验

<https://github.com/openai/openai-dotnet>

第四范式

特征工程

结构工程

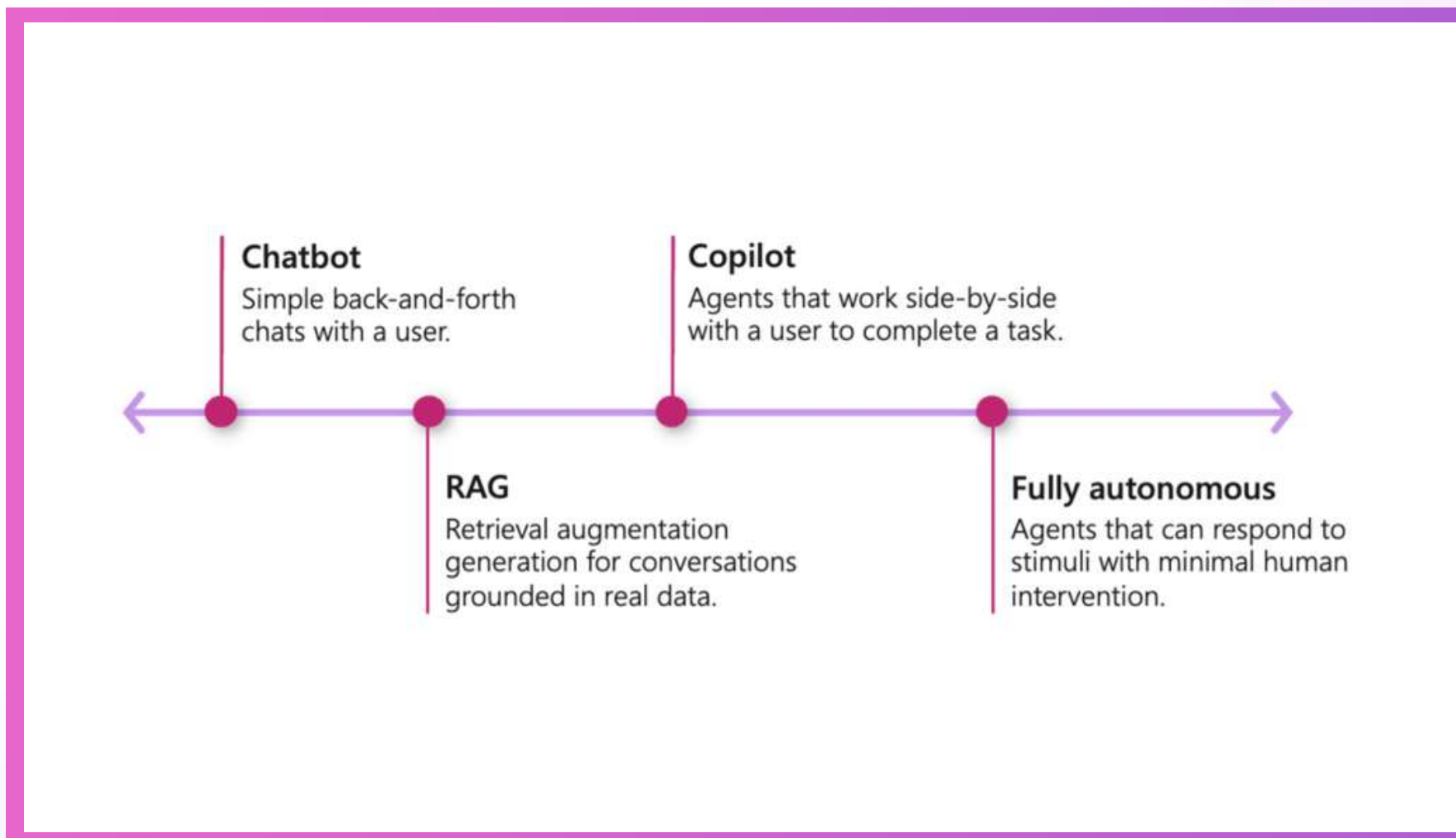
目标工程

提示工程

范式	简介	典型模型	参数规模	模型类别	优点	缺点	硬件需求
特征工程 Feature engineering	通过先验知识来定义规则，使用这些规则来更好地提取出文本中的特征	BOW TF-IDF	100	传统机器学习模型	速度快	特征工程难，工作量大	CPU
结构工程 Architecture engineering	无需特征工程，需要设计一个好的模型结构来自动学习文本中的特征	Word2Vec FastText ELMo	>1M	深度学习模型	速度快，无需特征工程	不适合直接做下游任务，需要人工设计网络结构，样本需求大	GPU单卡
目标工程 Objective engineering	通常不会对模型本身做太多改动，而是在损失函数上做改动，以适应输入数据，预训练 + 微调	BERT GPT	>100M	预训练语言模型	可以直接做下游任务，少量样本即可得到较好效果	速度慢，训练成本高	GPU卡，几十张
“第四范式” 提示工程 Prompt engineering	模型适配任务 → 任务适配模型	T5 GPT3 GPT4	>1B	大模型	无需训练，可零样本学习	速度慢，效果强，依赖模型和提示	GPU卡，千张以上

\* 来自卡耐基梅隆大学博士后刘鹏飞2021年的论文《 Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing 》

# 智能应用范式

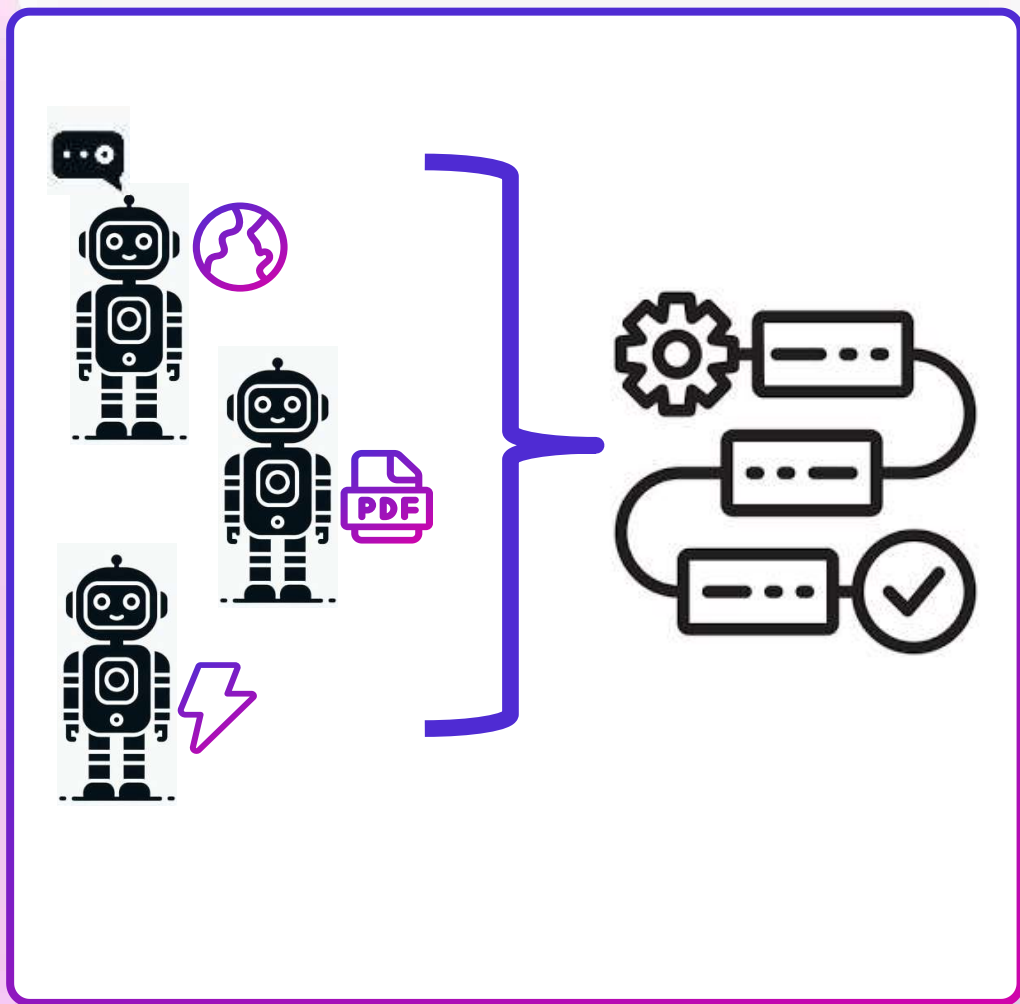






# 基于RAG的智能体

擅长回答有关特定主题或数据的问题



# 组装 智能体 workflow

定义智能体、事件和操作，以便在更复杂的工作流程中组合它们

基础技能

Prompt

提示工程是一个较新的学科，应用于开发和优化提示词（Prompt），帮助用户有效地将语言模型用于各种应用场景和研究领域。

Plugins

提供一堆可能可选择的工具。在应用时，需要结合上下文，生成动态的工具列表，以让 LLM 选择合适的工具。

GPTs

GPTs被定位为ChatGPT的定制版本，允许用户构建适用于特定用途或业务场景的定制化模型。它通过大量的预训练数据进行训练，可根据需求定制适用于不同场景。

GPT APIs

基于云的 API，可以访问 GPT 的高级语言模型。它允许开发人员将自然语言处理 (NLP) 功能集成到他们的应用程序中，并构建智能对话界面

Function Calling

由 LLM 在 API 调用时，检测何时应该调用一个函数，传递输入给函数，并调用这个函数。

Fine Tune

基于通用预训练的模型，通过在特定任务的数据集上进行进一步的训练来微调模型的参数，以使其适应特定任务的要求。

Copilot

使用**提示工程**与**智能体**（Agent）进行交互，和**外部工具**相结合。根据用户意图，结合特别的模式编写 workflow，以自动构建上下文。

RAG/GraphRAG

除了 LLM 本身已经学到的知识之外，通过外挂其他数据源的方式来增强 LLM 的能力，这其中就包括了外部向量数据库等。

AI Agents

使用高级智能体（Agent）自动生成提示来控制预训练模型和外部工具。由 LLM 来自动根据用户意图生成 workflow，并自动控制外部工具。

专业技能

# Schillace 法则（使用LLM创建软件的最佳实践）

**原则一** 若模型能胜任编写任务，便仍需动手，模型会不断提高，而代码则无法如此

**原则二** 以精准为代价，换取更高的杠杆，借助互动缓解风险

**原则三** 代码用于语法和过程:模型用于语义和意图

**原则四** 系统的脆弱程度取决于最脆弱的部分

**原则五** 问得越好，答得越好

**原则六** 不确定性是抛出的异常

**原则七** 文本是一种通用的数据传输协议

**原则八** 对于你来说困难的，对于模型也是困难的

**原则九** 当心“意识的错觉”，模型可以用来反过攻击自身



Ask smart to get smart.

席勒士  
九原则

# .NET 与生成式 AI

C# 已经在人工智能领域发展并站稳脚跟。下面是.NET社区中AI相关资源：

- Semantic kernel: <https://github.com/geffzhang/awesome-semantickernel/>
- Autogen: <https://github.com/microsoft/autogen/tree/main/dotnet>
- Botsharp: <https://botsharp.readthedocs.io/en/latest/>
- AIDotNet: <https://github.com/AIDotNet>
- Elsa-core: <https://github.com/elsa-workflows/elsa-core>
- Senaprc.AI: <https://github.com/Senparc/Senparc.AI>
- Stable Diffusion: <https://github.com/mcmonkeyprojects/SwarmUI>
- StabilityMatrix: <https://github.com/LykosAI/StabilityMatrix>



# Semantic Kernel – 语义内核

<https://github.com/microsoft/semantic-kernel>

<https://github.com/geffzhang/awesome-semantickernel>

最新版本 1.20

Semantic Kernel (SK) 加速了利用 AI 的应用程序和服务的开发，封装了常见的 AI 应用程序设计模式：

- 提示工程
  - Prompt Chaining & Prompt + Code Chaining
  - Chain of Thought (CoT)
  - Zero-shot / Few-shot
- 语义记忆索引和存储，上下文记忆检索
- 技能定义、托管、发现
- 自然语言处理，意图检测
- 多模型和多模态。



## SK 中文技术社区



# 满足语义AI的轻量级内核

提示词链接、递归推理、总结、零/少样本学习、上下文记忆、长期记忆、嵌入、语义索引、规划、访问外部知识库/数据...



# Elsa Workflows + Semantic Kernel

<https://github.com/elsa-workflows/elsa-core/tree/main/src/modules>

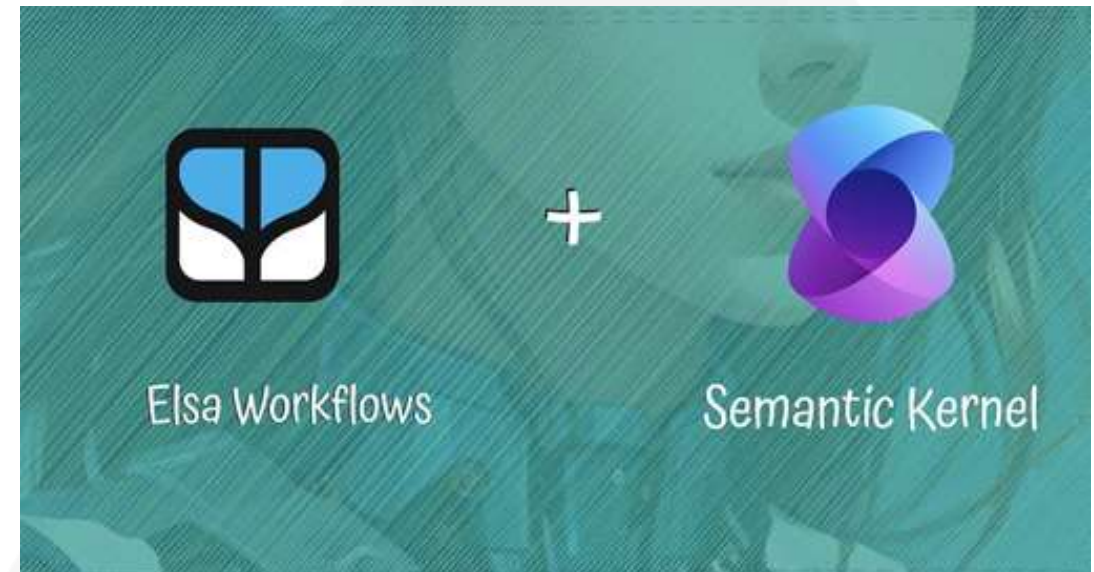
elsa-core / src / modules /

Add file ...

lahma and sfmskywalker Upgrade Quartz and some warning fixes (#5949)

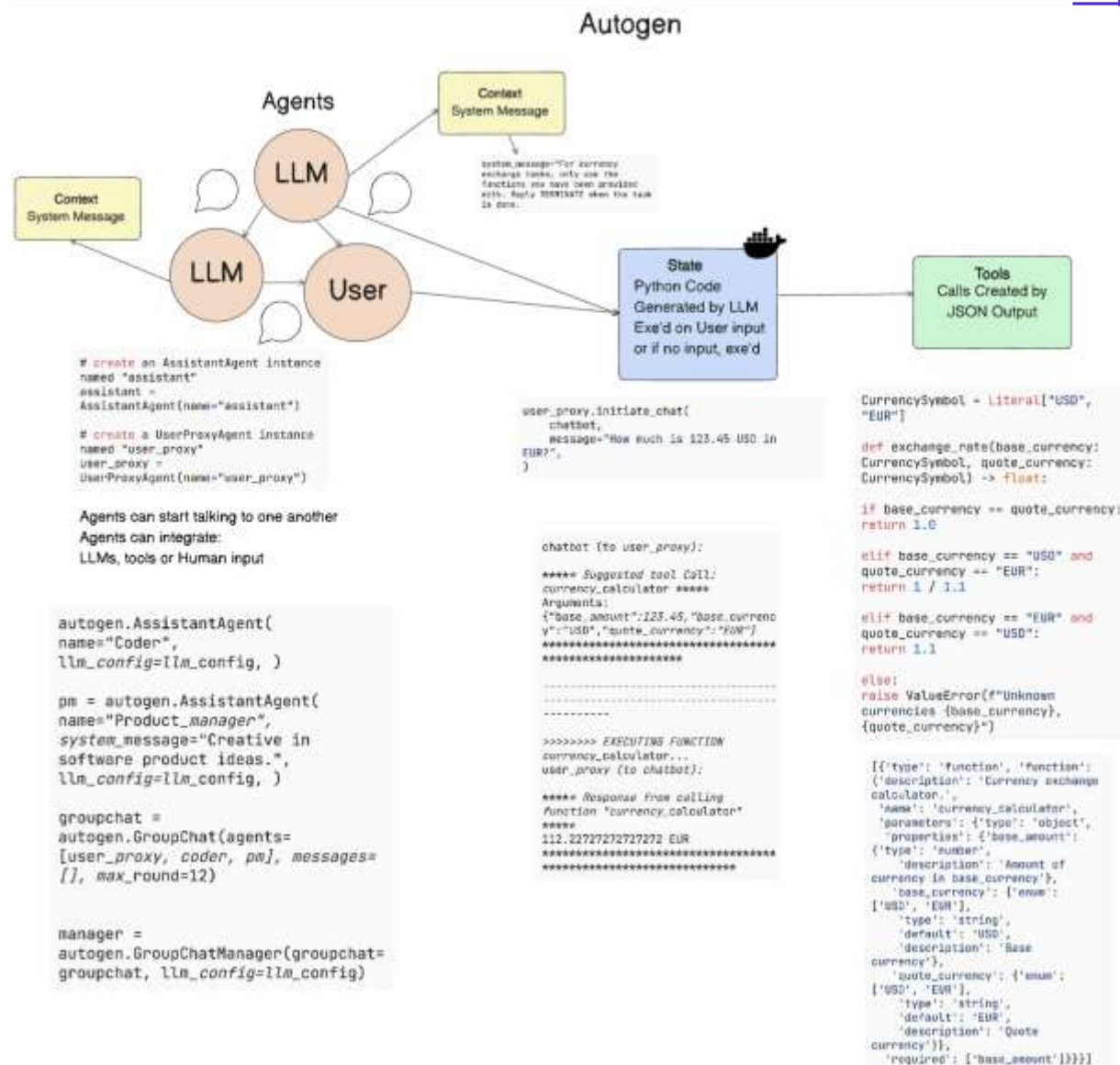
12f947c · 4 days ago History

Name	Last commit message	Last commit date
...		
Elsa.Agents.Activities	Change ActivityKind from Task to Job	3 weeks ago
Elsa.Agents.Api	Secrets API (#5967)	5 days ago
Elsa.Agents.Core	Update agent management features and refine configuration	3 weeks ago
Elsa.Agents.Models	Add Agents Module with Semantic Kernel Integration (#5937)	3 weeks ago
Elsa.Agents.Persistence.EntityFrameworkCore.MySql	Secrets API (#5967)	5 days ago
Elsa.Agents.Persistence.EntityFrameworkCore.SqlServer	Secrets API (#5967)	5 days ago
Elsa.Agents.Persistence.EntityFrameworkCore.Sqlite	Secrets API (#5967)	5 days ago
Elsa.Agents.Persistence.EntityFrameworkCore	Secrets API (#5967)	5 days ago
Elsa.Agents.Persistence	Add Agents Module with Semantic Kernel Integration (#5937)	3 weeks ago
Elsa.Alterations.Core	Merge remote-tracking branch 'origin/patch/3.2.x'	3 weeks ago
Elsa.Alterations.MassTransit	Add alteration notifications and management (#5097)	6 months ago



# Microsoft AutoGen

<https://github.com/microsoft/autogen/tree/main/dotnet>



AutoGen 的主要重点是对话。代理既可对话又可定制。

**可对话** - LLM 可以开始并继续与另一个 LLM 的对话以完成任务。这是通过创建并向他们提供特定的系统消息来完成的。

**可定制** - 代理不仅可以定义为 LLM，还可以定义为用户或工具。作为开发人员，您可以定义一个负责在完成任务时与用户交互以获得反馈的组件。此反馈可以继续执行任务，也可以停止执行任务。

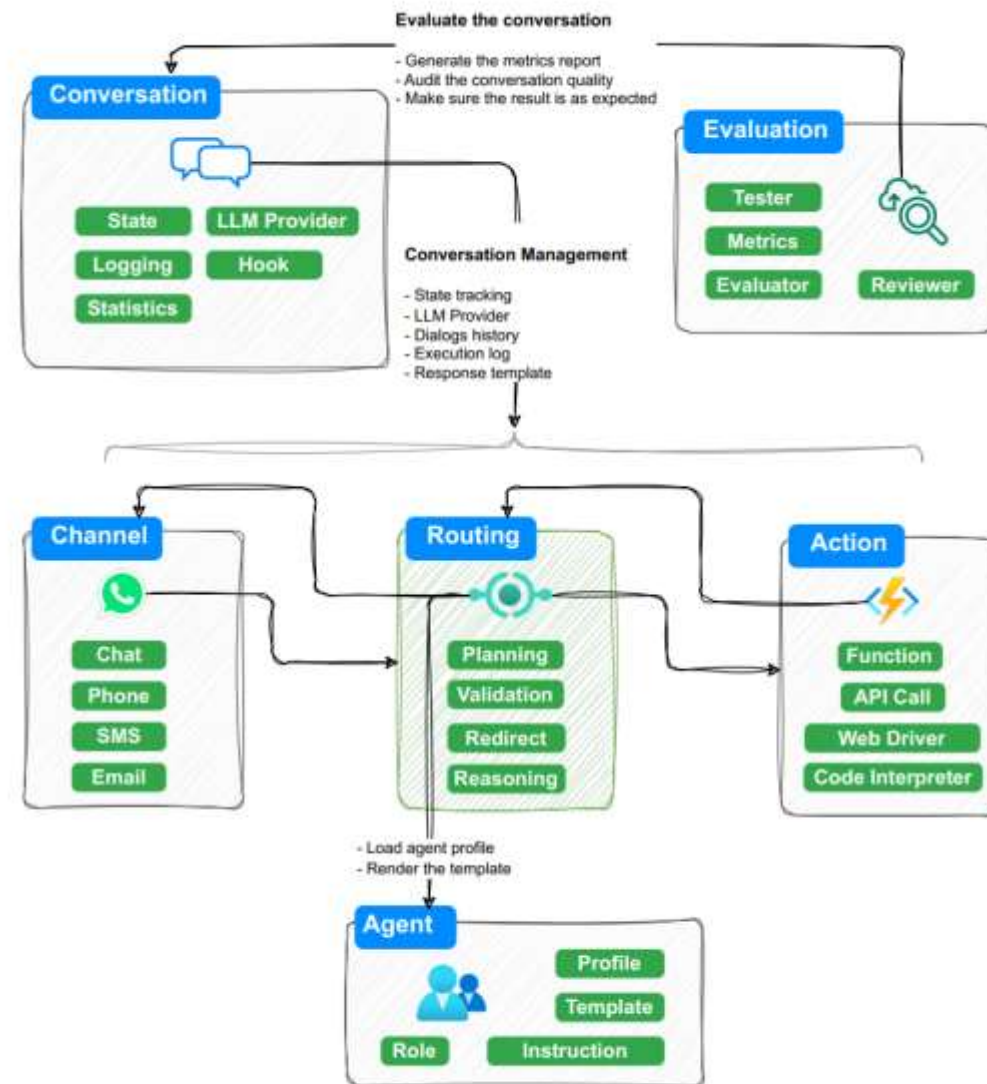
**状态和工具** - 为了更改和管理状态，Agent 会生成 Python 代码来完成任务



# Botsharp 介绍

Botsharp 是一个基于Apache 2.0的 .NET 开源项目，使企业开发者能够高效地将会话集成到现有业务系统中：

1. 内置多代理和具有状态管理的对话系统。
2. 支持多种LLM规划方法以处理不同任务。
3. 内置与RAG (Retrieval-Augmented Generation) 相关的接口，基于内存的向量搜索。
4. 支持多个 LLM 平台 ChatGPT 3.5 / 4.0, PaLM 2, LLaMA 2, HuggingFace, MetaGPT, 讯飞星火等
5. 允许具有不同职责的多个代理合作完成复杂任务。
6. 在一个地方构建、测试、评估和审查您的LLM代理。
7. 内置用SvelteKit编写的Web实时聊天用户界面。
8. 抽象标准的富内容数据结构。与微信、Facebook Messenger、Slack和Telegram等流行消息渠道集成。
9. 提供RESTful开放API和WebSocket实时通信。





# 开箱即用的工作台

<https://botsharp.azurewebsites.net/>

The screenshot displays the BotSharp AI Agent Router interface. On the left is a dark sidebar with navigation links: Dashboard, Router, Evaluator, Agents, Conversations, Knowledge Base, Plugins, and Settings. The main area is titled 'ROUTER' and shows a workflow diagram: a 'User Request' box connects to a 'Router (PizzaBot)' box, which then branches into three agent boxes: 'Agent (Order Inquiry)', 'Agent (Ordering)', and 'Agent (Payment)'. On the right, there are two configuration panels. The 'Agent Settings' panel includes fields for Data Directory (agents), Template Format (liquid), LLM Provider (azure-openai), and LLM Model (gpt-35-turbo). The 'Routing Settings' panel includes fields for Agent Id (01fcc3e5-9af7-49e6-ad7a-a780bd12dc4a) and Planner (NaivePlanner). The footer contains the text '2023 © SciSharp STACK' and 'Design & Develop by open source community'.

2023 © SciSharp STACK

Design & Develop by open source community

# AIDotNet社区介绍



**AIDotNet** 是由一群热爱人工智能（**AI**）和 **DotNet** 技术的开发者组成的开源组织。我们致力于创建更智能的 **AI** 智能体。我们的项目大多采用 **Apache License 2.0** 许可，允许任意商用。我们鼓励您参与项目开发，并欢迎您加入我们的社区。

# 社区项目介绍

项目名称	链接	描述
AntSK	<a href="https://github.com/AIDotNet/AntSK">https://github.com/AIDotNet/AntSK</a>	基于.Net8+AntBlazor+SemanticKernel 和 KernelMemory 打造的AI知识库/智能体，支持本地离线AI大模型。可以不联网离线运行。支持aspire观测应用数据
FastWiki	<a href="https://github.com/AIDotNet/fast-wiki">https://github.com/AIDotNet/fast-wiki</a>	基于.NET8+React+LobeUI实现的企业级智能客服知识库
Thor(雷神托尔)	<a href="https://github.com/AIDotNet/Thor">https://github.com/AIDotNet/Thor</a>	Thor提供了大部分的AI模型兼容OpenAI的接口格式，并且将所有模型的实现单独成类库打包成SDK使用，可快速使用入门，也可以使用Thor的服务部署成独立的AI中转服务，在Thor中提供了基本的用户管理和权限管理，并且支持多模型转换，以便提供给服务OpenAI的API风格。
GraphRag.Net	<a href="https://github.com/AIDotNet/GraphRag.Net">https://github.com/AIDotNet/GraphRag.Net</a>	参考GraphRag使用 Semantic Kernel 来实现的dotnet版本，可以使用NuGet开箱即用集成到项目中
ThorChat	<a href="https://github.com/AIDotNet/ThorChat">https://github.com/AIDotNet/ThorChat</a>	这个是移植lobechat，将next替换成纯静态项目，后台使用.NET 8提供WebApi支持

# Senparc.AI

**Senparc.AI** 致力于建设一套服务于 AI 落地的工具包和开发者生态。

## 使命

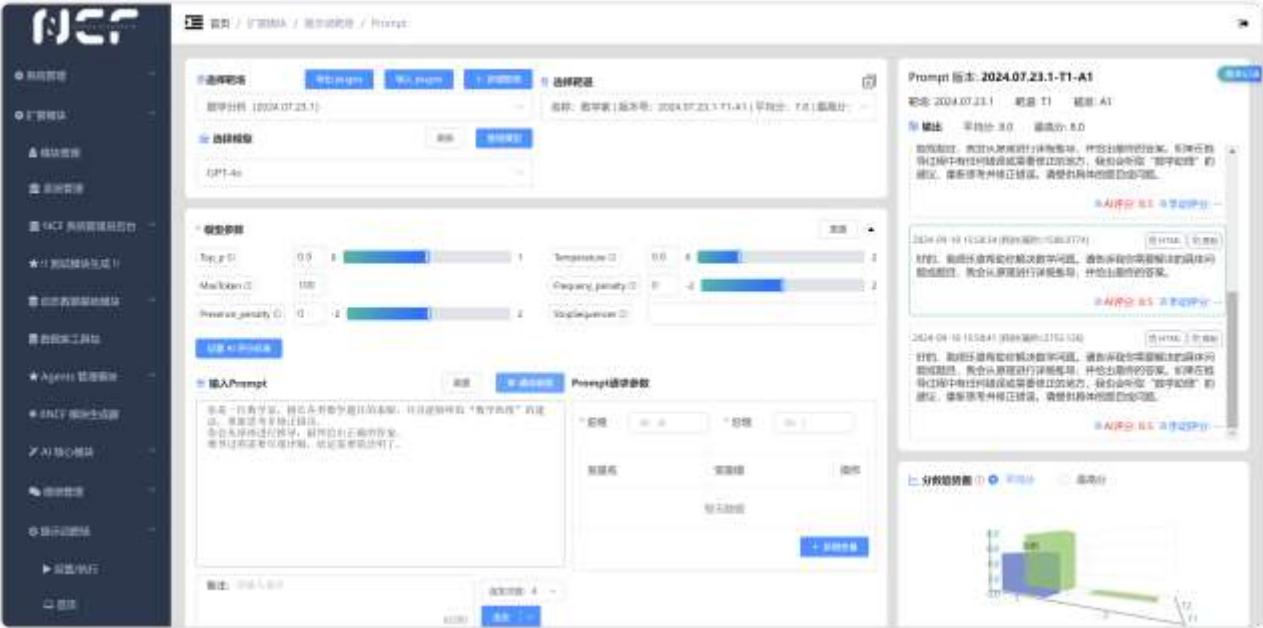
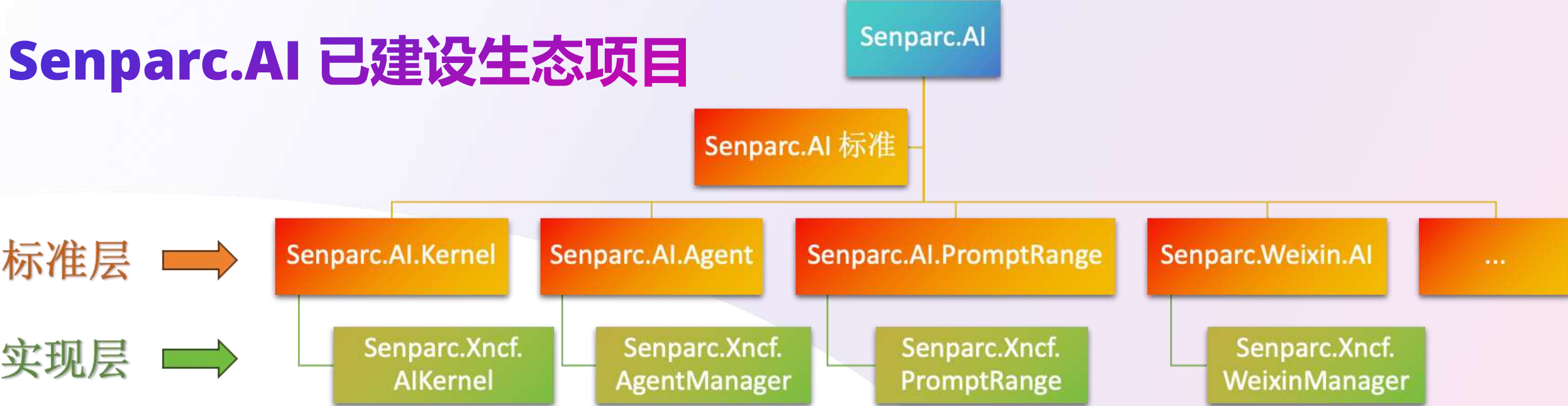
帮助开发者和 AI、大模型消费者，获得更安全、更高效、更经济、有稳定的模型和数据服务，深入赋能落地解决方案。

Senparc.AI: <https://github.com/Senparc/Senparc.AI>

NCF Template: <https://github.com/NeuCharFramework/NCF>

NCF Package Source: <https://github.com/NeuCharFramework/NcfPackageSources>

# Senparc.AI 已建设生态项目



Senparc.AI 已建设完成包括 AIkernel、Agent（智能体）、Prompt（提示词）、终端（微信）在内的一系列基础标准，并在这些标准基础上，使用 NeuCharFramework（NCF）的模块化能力进行模块化的实现，做到开箱即用、即插即用。例如，通过 AIKernel、PromptRange、WeixinManager 三个模块的组合，即可 0 代码成微信机器人程序，使用少量代码即可实现更多扩展功能。

示例：PromptRange（提示词靶场模块）



# Senparc.AI Demo

[https://mp.weixin.qq.com/s/Pz6lBxdgPD52V2NaD\\_GZaA](https://mp.weixin.qq.com/s/Pz6lBxdgPD52V2NaD_GZaA)

盛小嗨



https://localhost:44311/Admin/AiKernel/Index?uid=795D12D8-580B-40F3-A6E8-A5D9D2EA...

点此搜索

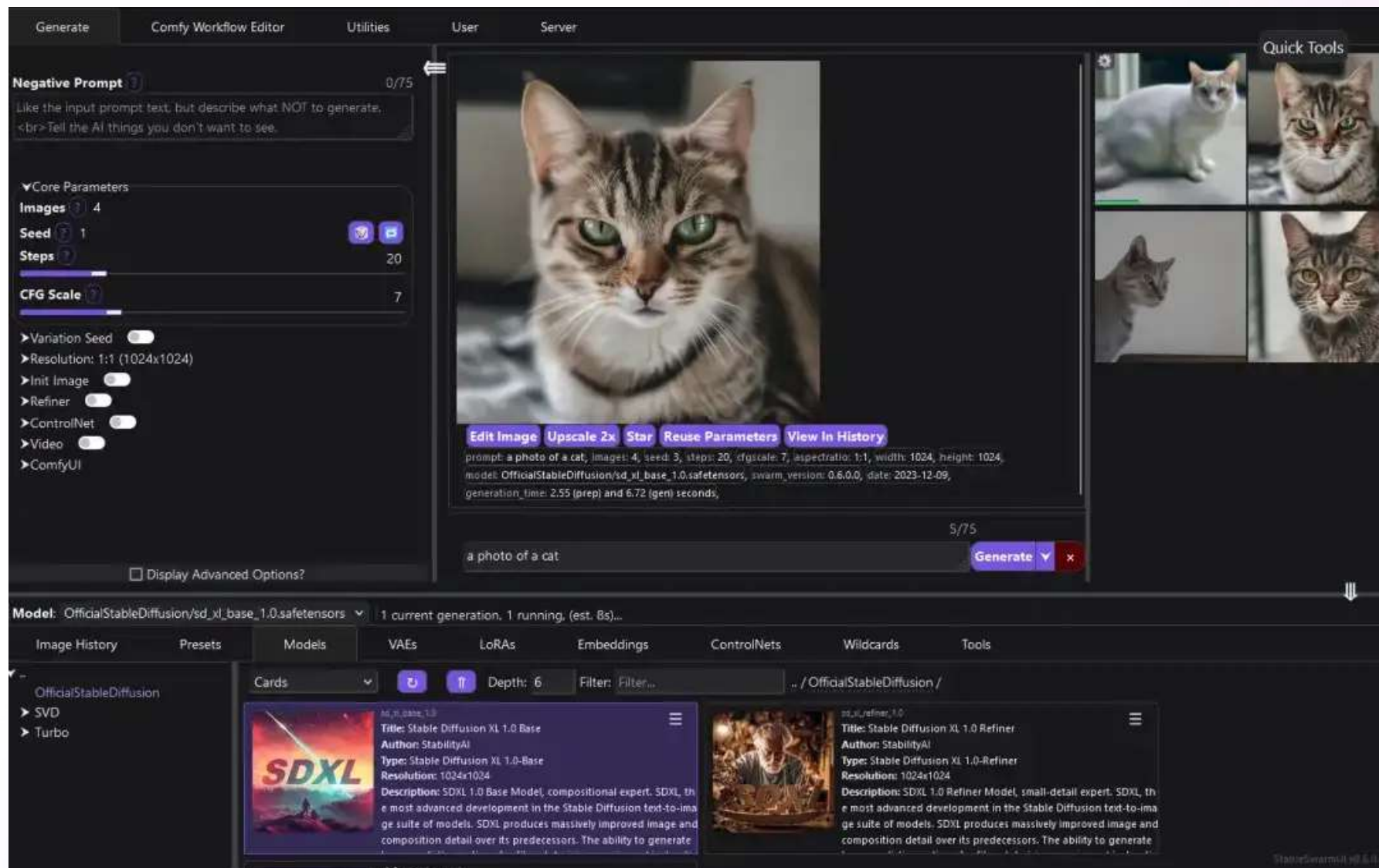
首页 / 扩展模块 / AI 核心模块 / 列表

+ 添加 + 导入 NeuChar AI 云端模型助力

别名	模型名称	部署名称	Endpoint	平台	模型类型	Api Key	备注	添加时间	操作
gpt-4o	gpt-4o	gpt-4o	https://www.neuchar.com/7289	NeuCharAI	Chat	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
gpt-3.5-turbo-instruct	gpt-3.5-turbo-instruct	gpt-3.5-turbo-instruct	https://www.neuchar.com/7289	NeuCharAI	Chat	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
gpt-4-32k	gpt-4-32k	gpt-4-32k	https://www.neuchar.com/7289	NeuCharAI	Chat	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
gpt-3.5-turbo	gpt-3.5-turbo	gpt-3.5-turbo	https://www.neuchar.com/7289	NeuCharAI	Chat	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
text-embedding-ada-002	text-embedding-ada-002	text-embedding-ada-002	https://www.neuchar.com/7289	NeuCharAI	TextEmbedding	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
gpt-4	gpt-4	gpt-4	https://www.neuchar.com/7289	NeuCharAI	Chat	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>
text-davinci-003	text-davinci-003	text-davinci-003	https://www.neuchar.com/7289	NeuCharAI	TextCompletion	*****18f3	从 NeuChar AI 导入 (DevId:7289)	2024/9/22 14:34:58	<a href="#">编辑</a> <a href="#">删除</a>

<https://www.neuchar.com/Developer/AiApp>

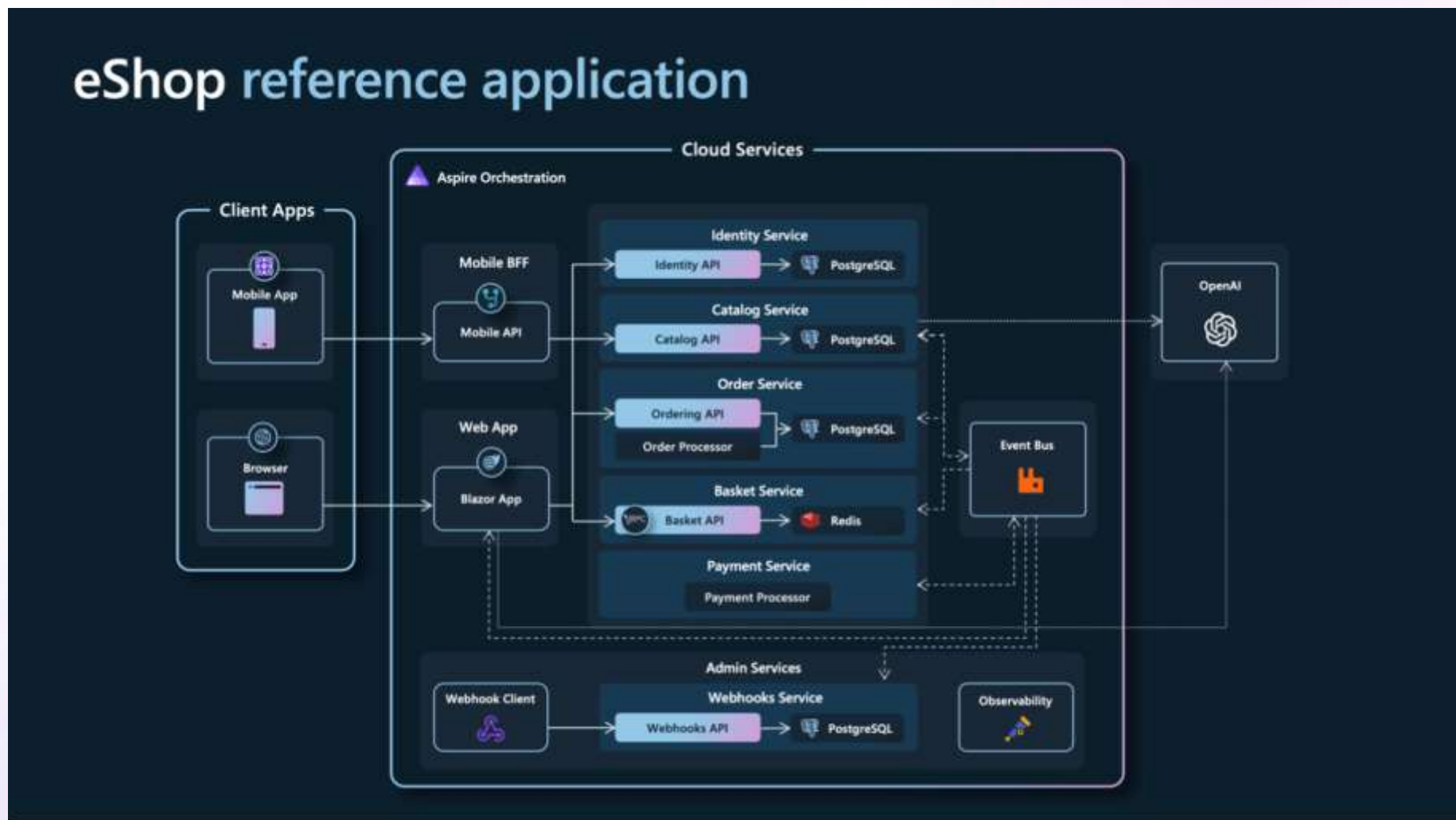
# Stable Diffusion客户端



<https://github.com/mcmonkeyprojects/SwarmUI>

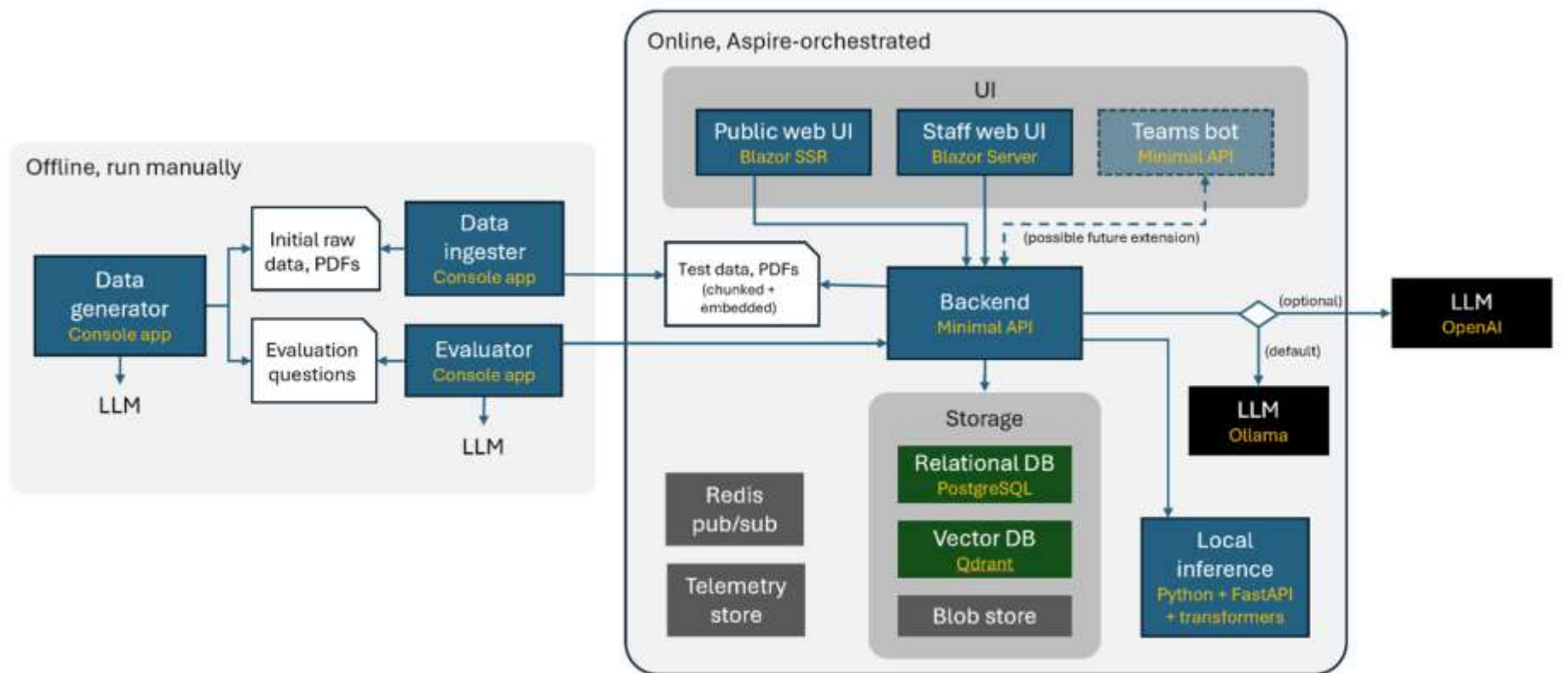
# 示例程序eshop和eshopsupport

基于 .NET Aspire 的参考电商应用，展示了服务架构在构建现代在线购物平台中的应用。项目涵盖产品目录、购物车和订单管理等核心电商功能，同时提供本地开发和部署的支持。



# 示例程序eshop和eshopsupport

在 .NET 中构建 AI 解决方案的示例应用，特别是生成式 AI。该项目演示了一个使用服务架构的电商网站的客户支持应用。





# 做产业变革的支点

## Make AI Happen

