AI and .NET = Great!!

M Ibnu Fadhil
Gravicode

M. IBNU FADHIL

CODER.ENTREPRENEUR.TINKERER

Twitter : @mifmasterz

Founder of Gravicode, AI & IoT Enthusiast

Initiator of Gadgeteer Indonesia & BMC

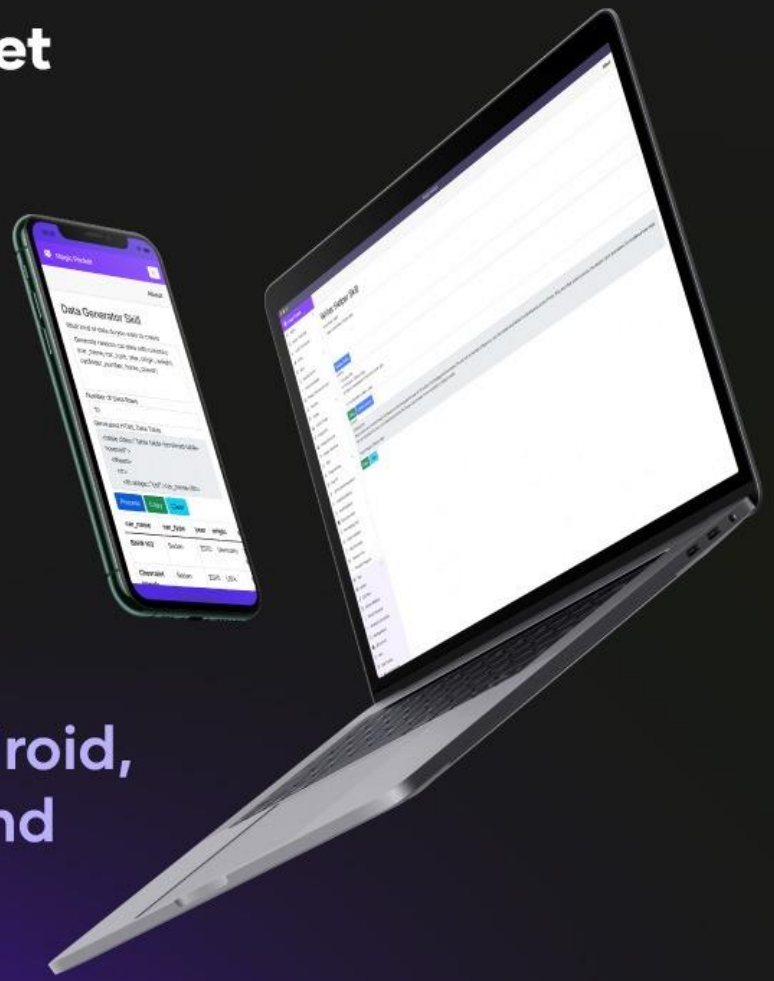
fb.com/netgadgeteerindonesia

fb.com/buitenzorgmakersclub

Contributor in makers.id

Creator of Magic Pocket and Marvelous Bot

Magic Pocket



Get it Now!
Magic Pocket is Available in Android, Windows, iOS and macOS

Get it from Microsoft

GET IT ON Google Play

Download on the App Store

MAGIC POCKET IS DEVELOPED BY

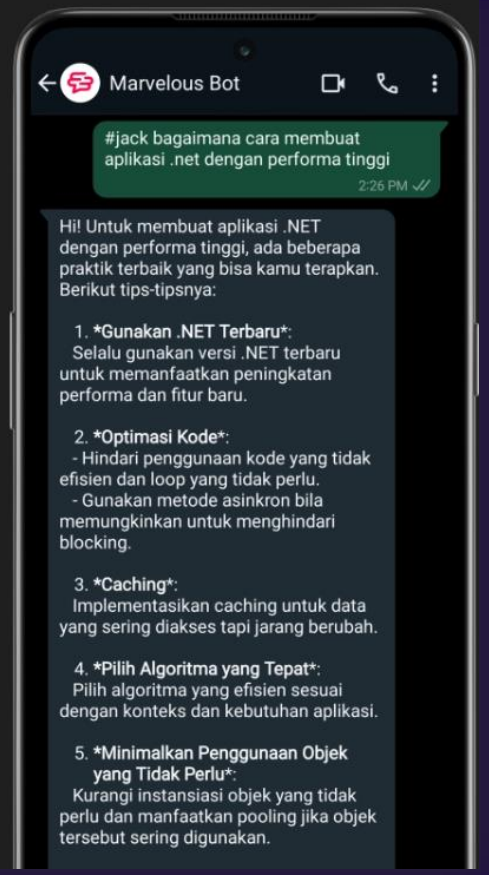
gravicode.studios gravicodestudio gravicode.studios

GRAVI CODE STUDIOS

Get it Now !!

MARVELOUS BOT

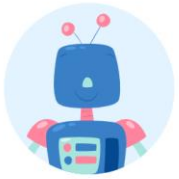
Marvelous Bot (Marbot) adalah virtual AI assistant yang siap memberikan pelayanan informasi kepada siapa saja yang membutuhkannya. Bot ini adalah bentuk upaya kami mendemokratisasi AI agar mudah dimanfaatkan semua kalangan.



Public Bot Personas

#ike

all-the-things you need bot, penyedia segala macam informasi

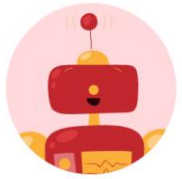


#siti

Teman ngobrol yang ceria

#ustad

Bot penyedia informasi tentang islam secara umum



#jack

Bot yang ahli coding dengan platform .NET

#asep

Bot penyedia informasi, dengan kekhasan berbahasa sunda

Get it Now !!

Agenda

Open AI dotnet SDK

Windows AI

Microsoft.Extensions.AI

Summary & Key Takeaways

01

What is openai-dotnet and who is building it?

.NET developer pain points

Unfamiliar ecosystems

Python or Node.js

REST APIs

Custom wrappers

No SLAs

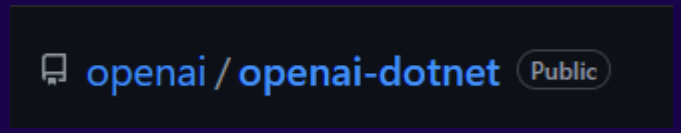
Risk aversion

openai-dotnet is...

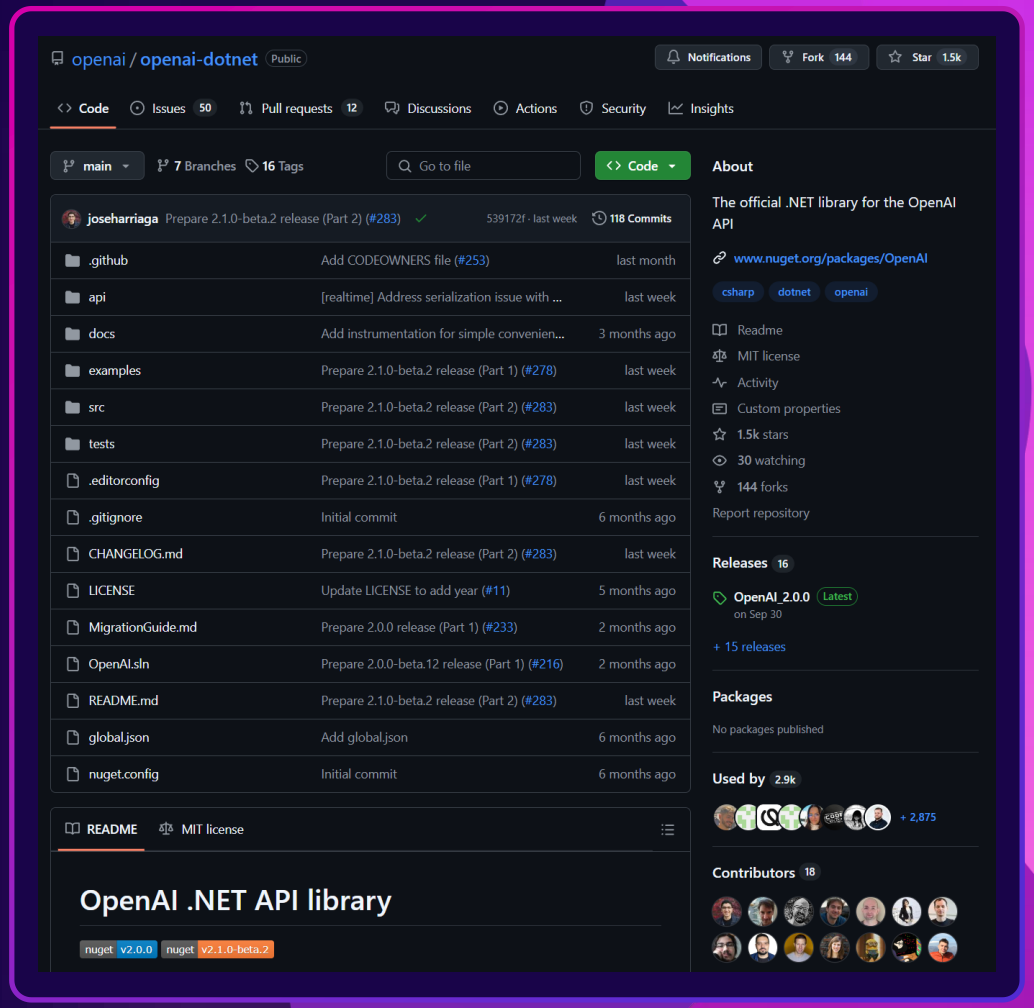
the **official** OpenAI library for .NET,
built by **Microsoft**,
on behalf of **OpenAI**

What does this mean in practice?

It means that the repo is under the OpenAI GitHub org, because they own it (and the same goes for the NuGet package).



The contributors to the repo though are Microsoft employees building the library in the open.



01-A

What are its capabilities?

Complete coverage of the OpenAI REST API

The full gamut of OpenAI's REST API is supported today, including the recently released Realtime API for which we had support on the same day it was publicly announced.

Support for the latest OpenAI models

OpenAI's latest flagship models, including GPT-4o, GPT-4o mini, o1-preview, and o1-mini, are fully supported today. Moving forward, our goal is to provide access to the very latest models as soon as they're publicly available.

Idiomatic and productive

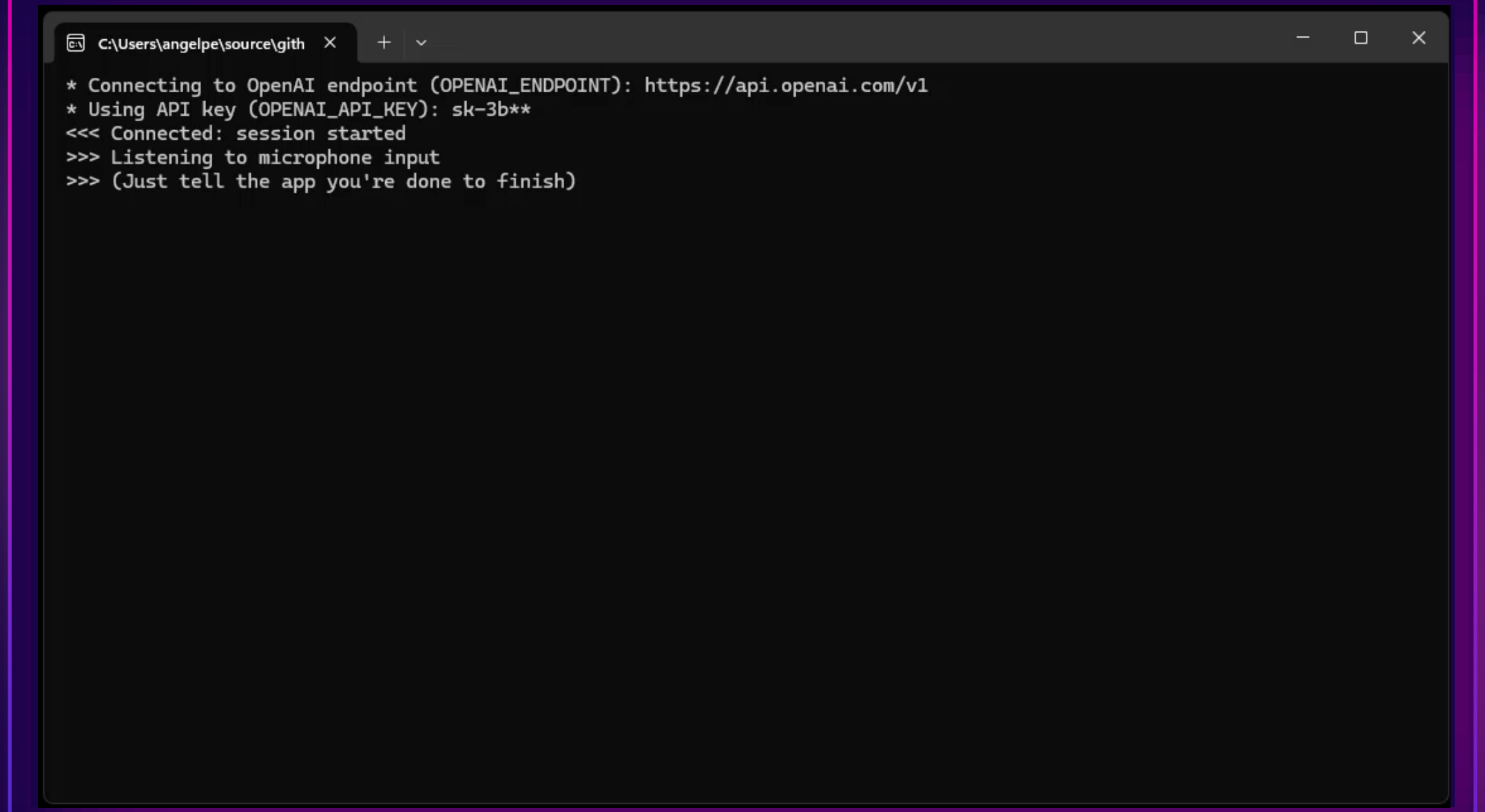
Designed to be intuitive and efficient for C# developers, optimized for real-world tasks. It provides seamless integration with OpenAI's latest features, allowing developers to build intelligent applications with ease and efficiency.

Extensible

Designed with extensibility in mind, allowing the community to build additional libraries on top of it.

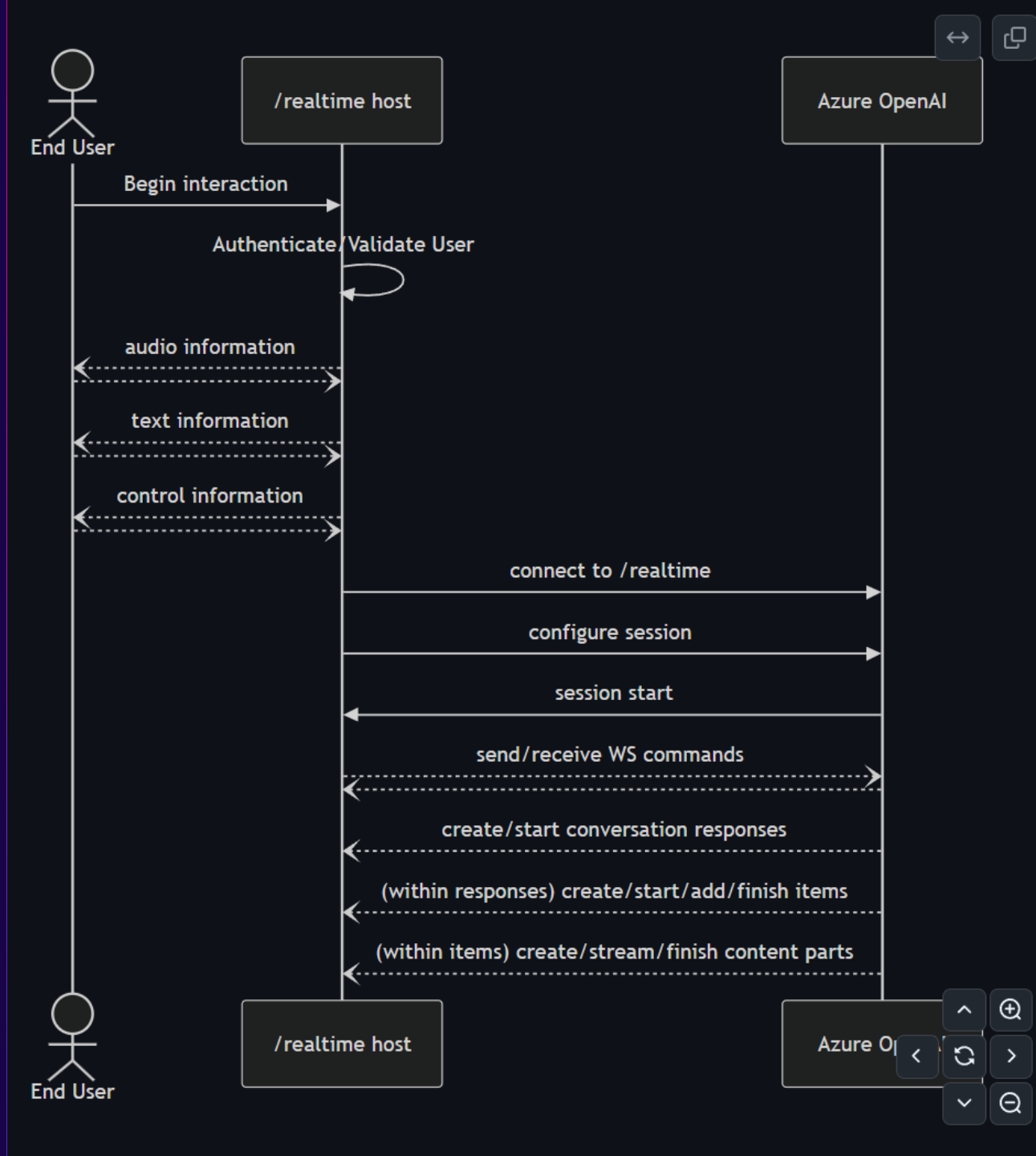
01-B

Demos: Realtime API for audio


C:\Users\angelpe\source\gith
* Connecting to OpenAI endpoint (OPENAI_ENDPOINT): https://api.openai.com/v1
* Using API key (OPENAI_API_KEY): sk-3b**
<<< Connected: session started
>>> Listening to microphone input
>>> (Just tell the app you're done to finish)

Realtime API interactions

Using the Azure OpenAI library for .NET



01-C

Resources and how to get involved

Resources

OpenAI

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

Official OpenAI GitHub repo

https://github.com/Azure-Samples/aoai-realtime-audio-sdk

Sample code for Realtime API for audio demo

https://aka.ms/openai/feature-request

Feature requests for the official OpenAI library

https://aka.ms/openai/bug-report

Support for the official OpenAI library

Azure

https://aka.ms/azsdk/openai

Companion Azure library source code

https://github.com/robch/openai-realtime-chat-with-keyword-cs

Sample code for Realtime API for audio demo

https://aka.ms/azsdk/feature-request

Feature requests for the companion Azure library

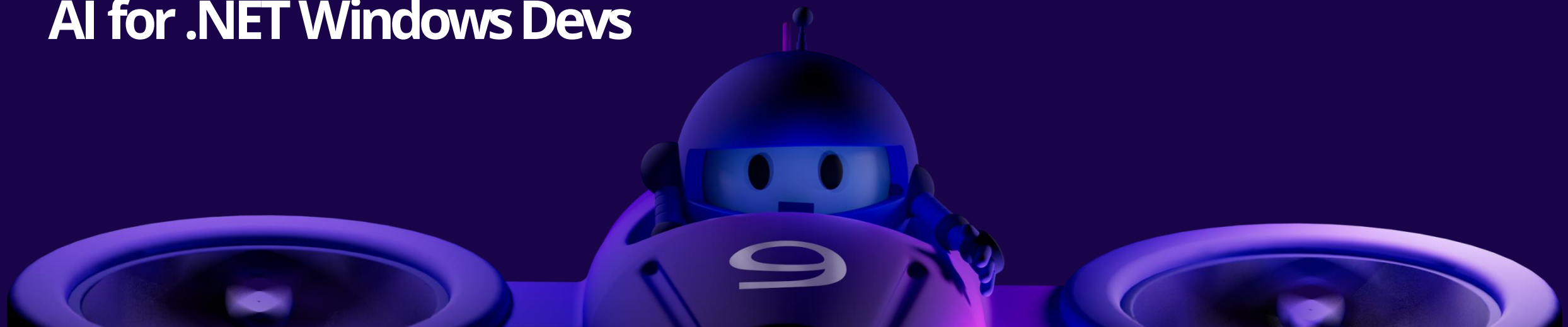
https://aka.ms/azsdk/bug-report

Support for the companion Azure library

02

Windows AI

AI for .NET Windows Devs



Windows Copilot Library

APIS backed by on-device ML Models

Phi Silica and **Text Recognition** will be available in an upcoming WinAppSDK experimental release

Vector Embeddings, **RAG**, **Text Summarization** and more will follow

```
if (!LanguageModel.IsAvailable())
{
    await LanguageModel.MakeAvailableAsync();
}

var languageModel = await LanguageModel.CreateAsync();

var response = await languageModel.GenerateResponseAsync("what's the meaning of life?");
```

DirectML

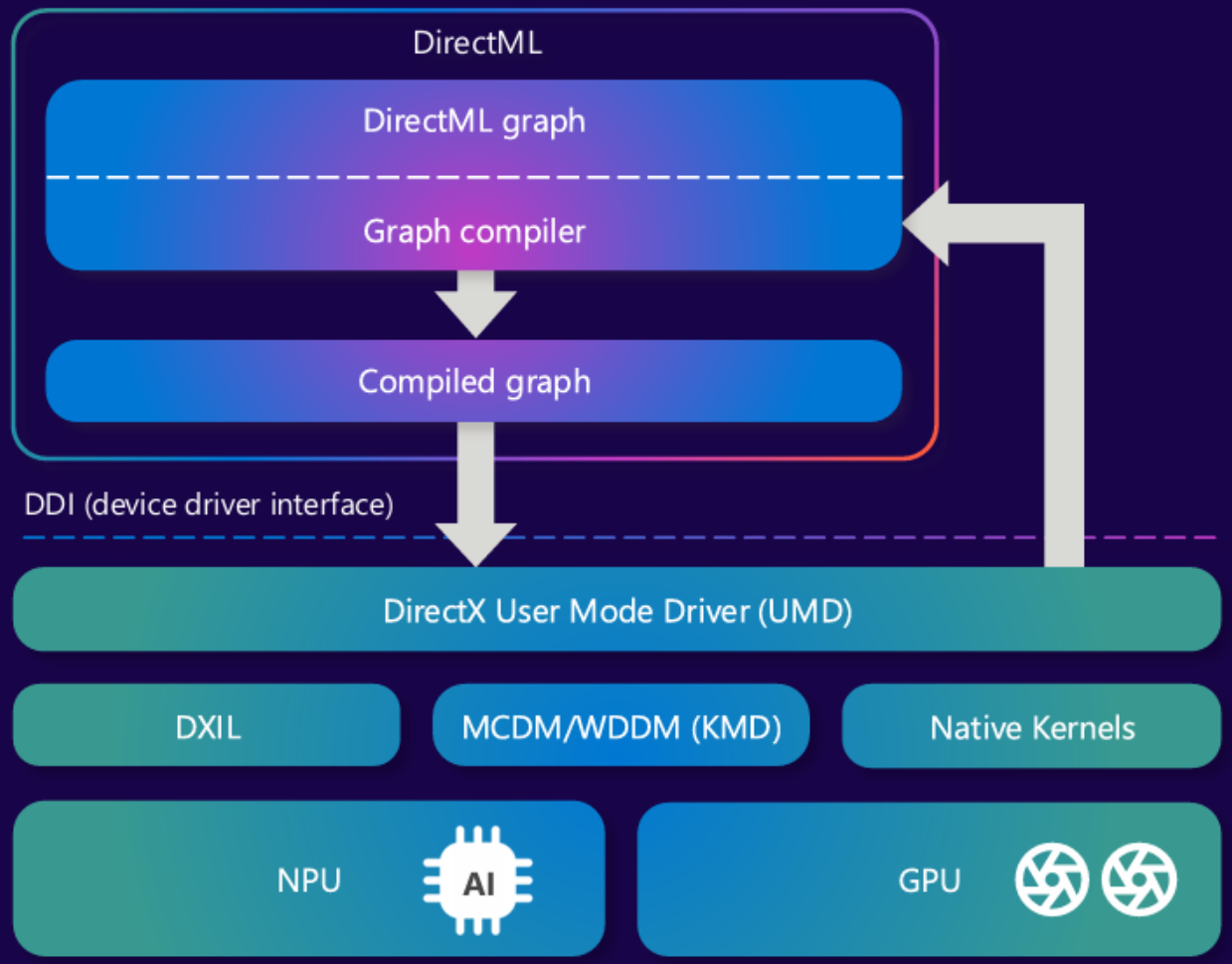
High-performance, hardware-accelerated DirectX 12 library for machine learning on Windows

A single, hardware-abstraction API for optimization and deployment, that scales across hardware

Full GPU support and expanding to include NPUs for full breadth of hardware support with AMD, Intel, Nvidia, and Qualcomm

Support for 4-bit Activation-Aware Quantization (AWQ) for minimal impact on accuracy to enable more models across Windows hardware

Available as NuGet package or as part of Windows 10 SDK or newer



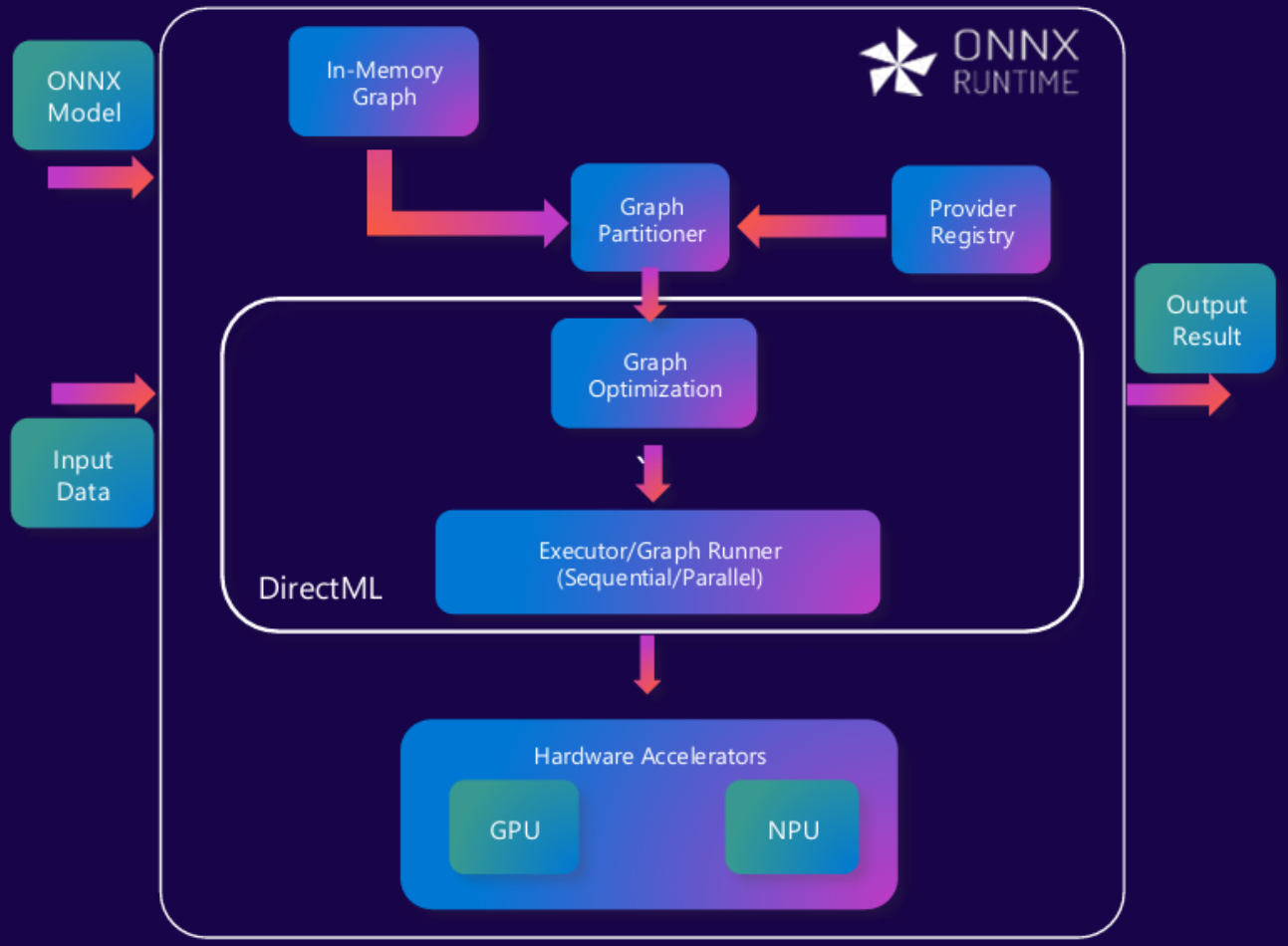
ONNX, ONNX Runtime (ORT), ORT GenAI, Olive

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.

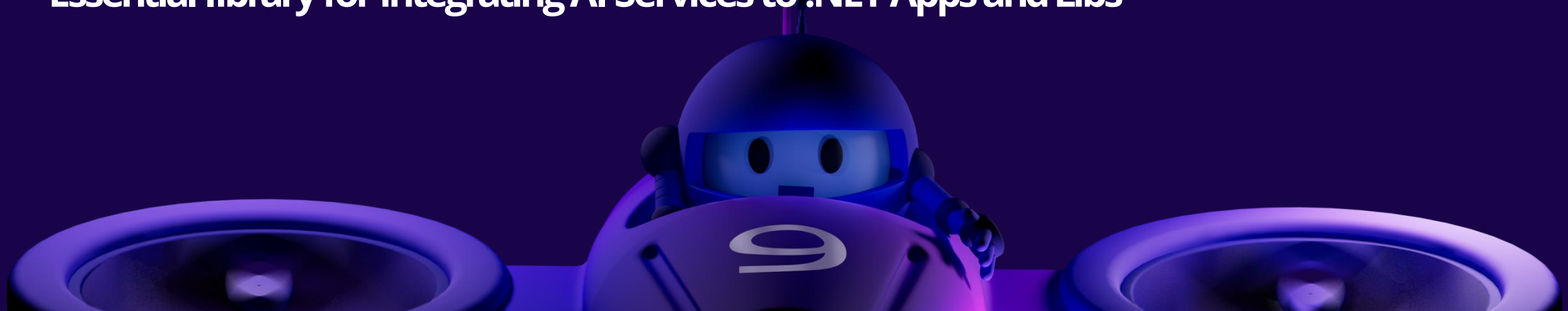


Demo..
Infuse AI in windows Apps with ONNX and Semantic Kernel

03

Microsoft.Extensions.AI

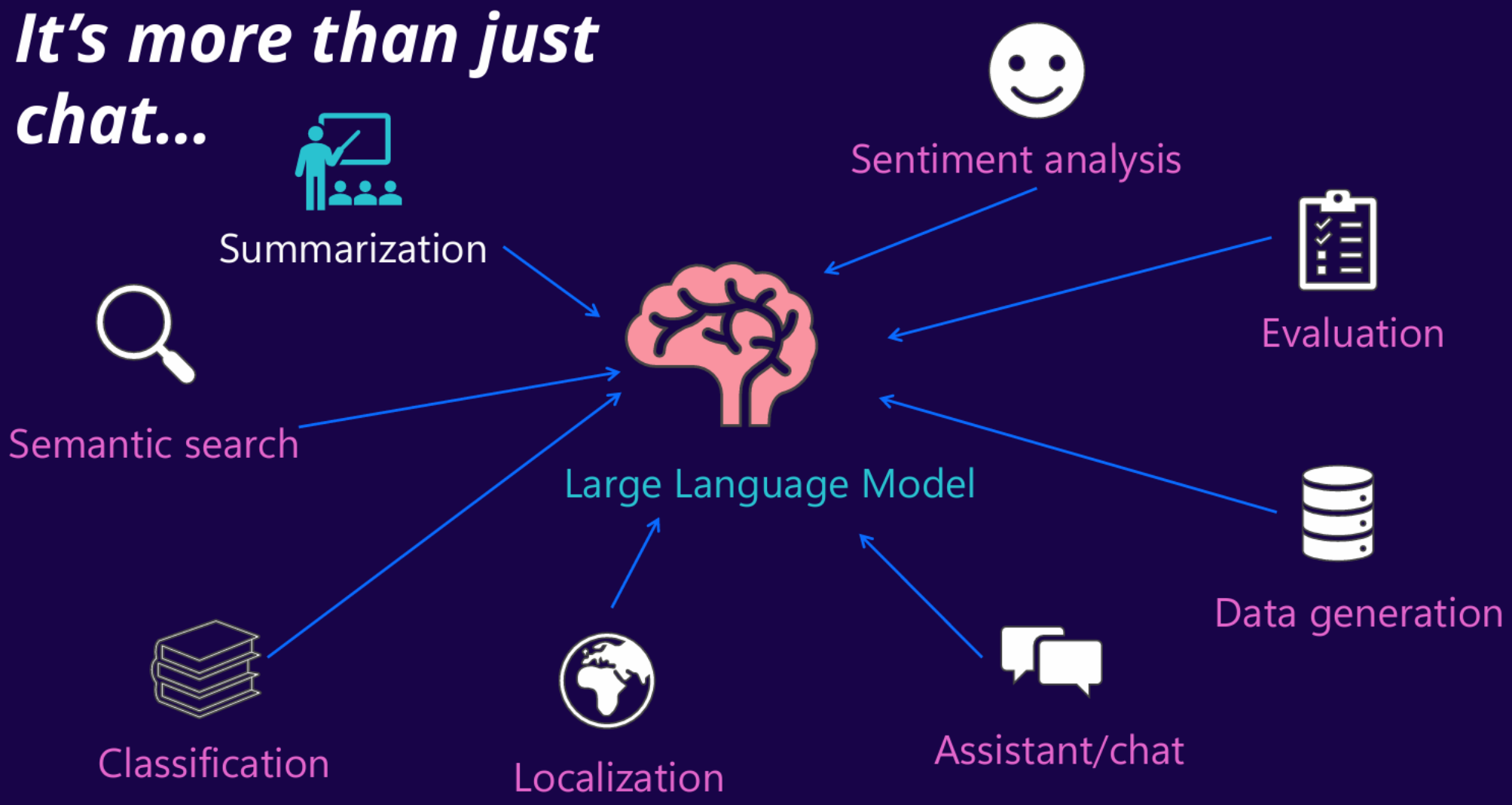
Essential library for integrating AI Services to .NET Apps and Libs



Think about LLM..

Chat-Gpt for example..

It's more than just chat...



```
<PackageReference Include="Azure.AI.OpenAI" Version="2.1.0-beta.1" />
<PackageReference Include="Azure.                              .0" />
<PackageReference Incl    Microsoft.Extensions.AI" Versio    .0.0-preview.9.24525.1" />
<PackageReference Incl    "Microsoft.Extensions.AI.OpenAI"    sion="9.0.0-preview.9.24525.1" />
```

```
    aiClie    Utils.CreateAzureOpenAIClient(    point, key);
IChatClient cha    ent = aiClient.AsChatClien    delId);
```

Common Building Blocks

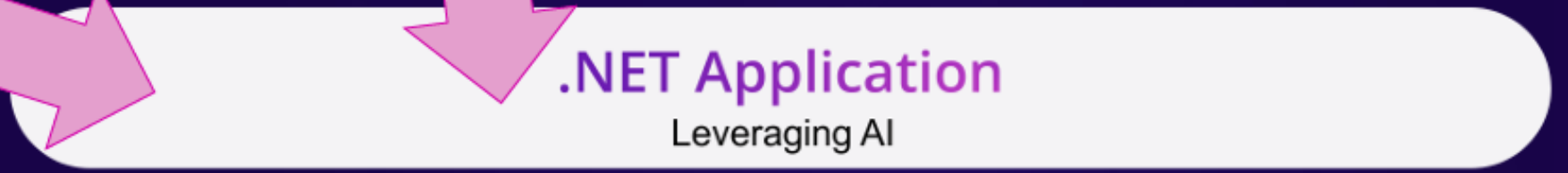
- Chat
- Streaming
- Embeddings

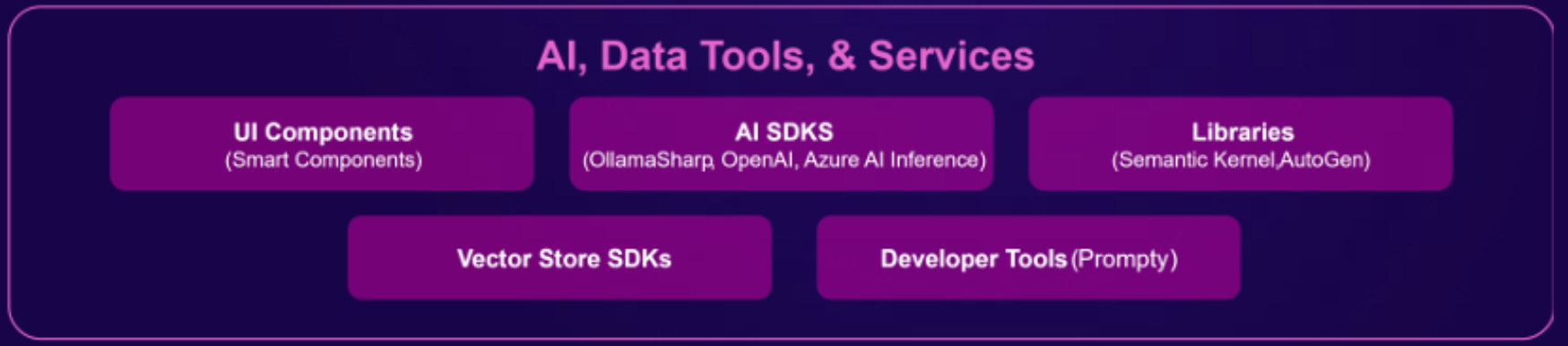
Middleware

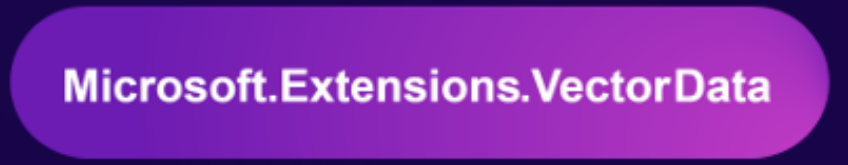
- Cache
- Telemetry
- Filter

Integrations

- OpenAI
- GitHub Models
- Ollama







Want to change?

```
IChatClient chatClient = new OllamaApiClient(new Uri(endpoint), modelId);
IEmbeddingGenerator<string, Embedding<float>> embeddingGenerator =
    new OllamaApiClient(new Uri(endpoint), embeddingsId);
```

Prompt time!

```
You are part of a customer support ticketing system.
Your job is to write brief summaries of customer support interactions.
This is to help support agents understand the context quickly so they can help
the customer efficiently.

Here are details of a support ticket.

${messages}

Write a summary that is up to 30 words long, condensing as much distinctive
information as possible

Summary:
""";
```

2 references

```
public async Task<ChatCompletion> GenerateLongSummaryAsync(string input)
{
    var prompt = GetLongSummaryPrompt(input);
    var response = await _chatClient.CompleteAsync(prompt);
    return response;
}
```

Demo..

Chat, ChatHistory, ChatStreaming, Function Calling, Embedding, Caching, DI, Telemetry with OpenAI, Ollama

RAG Time!

Retrieval
Augmented
Generation

Source (PDF)

Text

Embeddings

(Vectors)

LLM

Semantic Search

VectorDB

```
IEmbeddingGenerator<string, Embedding<float>> embeddingGenerator =
    new OllamaApiClient(new Uri(endpoint), embeddingsId);
```

```
// Configure product manual service
var vectorStore = new InMemoryVectorStore();
var productManualService = new ProductManualService(embeddingGenerator,
    vectorStore, useOpenAIEmbeddings);
```

```
// [1] Parse (PDF page -> string)
var pageText = GetPageText(page);

// [2] Chunk (split into shorter strings on natural boundaries)
var paragraphs = TextChunker.SplitPlainTextParagraphs([page.Text], 200);

// [3] Embed (string -> embedding)
var paragraphsWithEmbeddings = await _embeddingGenerator
    .GenerateAndZipAsync(paragraphs);
```

```
// [4] Save
var manualChunks =
    paragraphsWithEmbeddings.Select(p => new ManualChunk
    {
        ProductId = docId,
        PageNumber = page.Number,
        ChunkId = ++paragraphIndex,
        Text = p.Value,
        Embedding = p.Embedding.Vector.ToArray()
    });
```

```
{
  "ChunkId": 2460,
  "ProductId": 179,
  "PageNumber": 3,
  "Text": "Misuse and Abuse",
  "Embedding": [0.02627289, 0.066441216, 0.04378815, 0.03978883,
```

Demo..

RAG + Semantic Search

AI Evaluation

Traditional Software vs AI Evaluations

	Traditional Software	AI
Outputs	Predictable, Objective	Unpredictable, Subjective (Vibes-based)
Patterns	TDD, Assertions	Thresholds, Variable
Tools	Integrated (VS, CI/CD)	Isolated, Disparate

Built-in Evaluators & Extensible APIs

```
IEvaluator rtcEvaluator = new RelevanceTruthAndCompletenessEvaluator(options);
IEvaluator coherenceEvaluator = new CoherenceEvaluator();
IEvaluator fluencyEvaluator = new FluencyEvaluator();
IEvaluator groundednessEvaluator = new GroundednessEvaluator();
IEvaluator answerScoringEvaluator = new AnswerScoringEvaluator();
```

```
public interface IEvaluator
{
    0 references | 0 changes | 0 authors, 0 changes
    IReadOnlyCollection<string> EvaluationMetricNames { get; }

    0 references | 0 changes | 0 authors, 0 changes
    ValueTask<EvaluationResult> EvaluateAsync(
        IEnumerable<ChatMessage> messages,
        ChatMessage modelResponse,
        ChatConfiguration? chatConfiguration = null,
        IEnumerable<EvaluationContext>? additionalContext = null,
        CancellationToken token = default(CancellationToken));
}
```

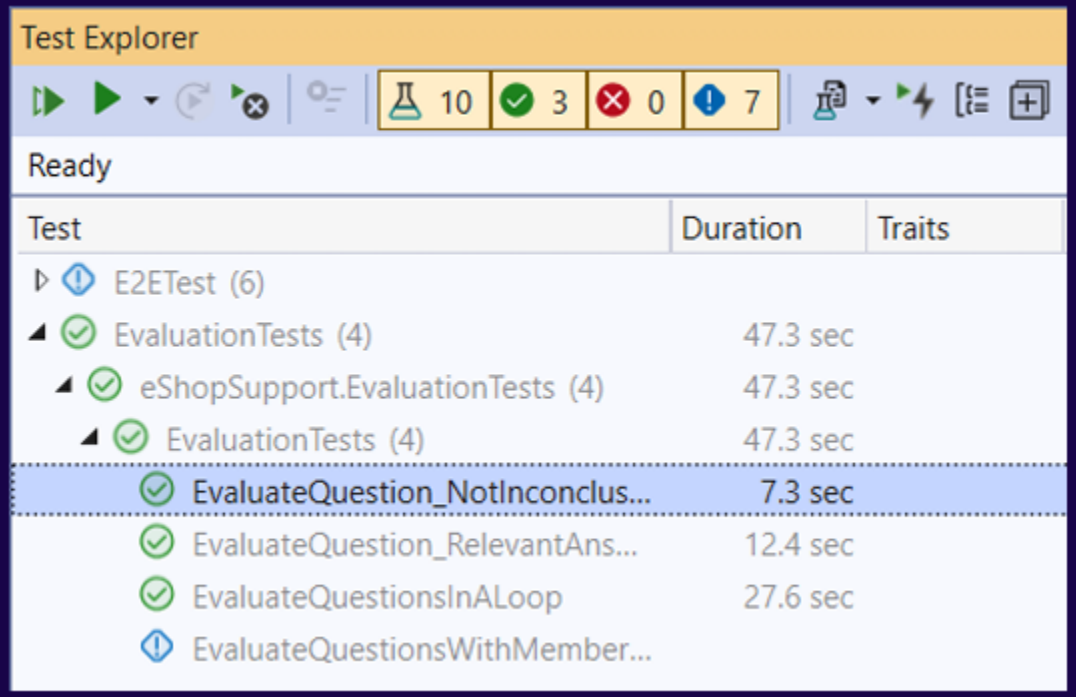
Evaluator Description

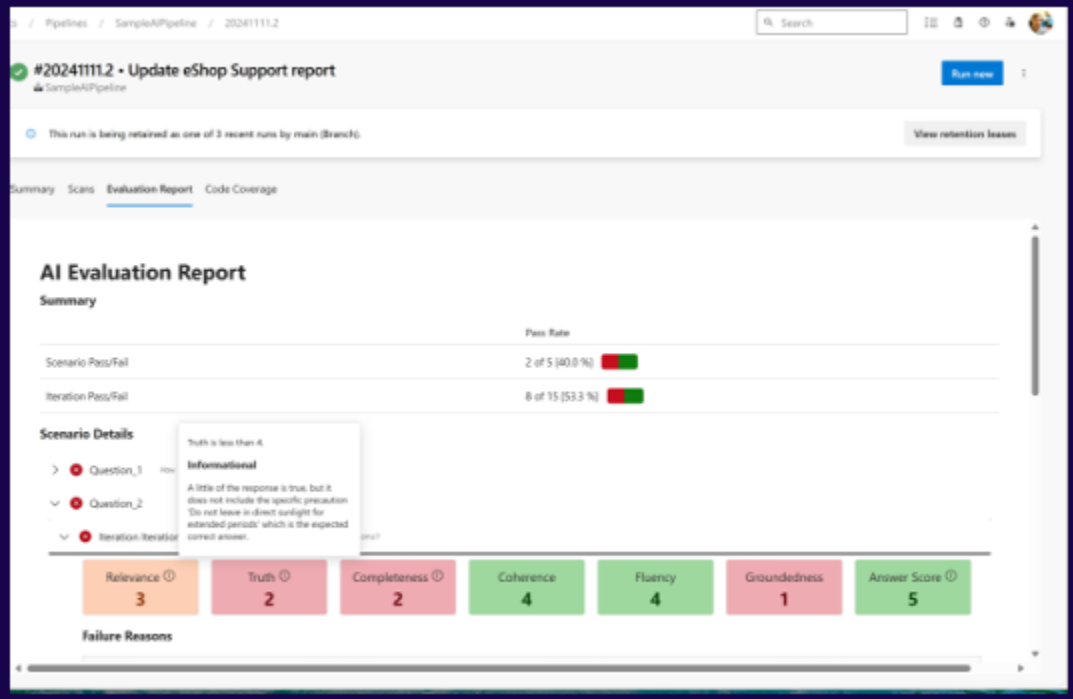
Recommended scenario	Evaluator Type	Why use this evaluator?	Evaluators
Retrieval-augmented generation question and answering (RAG QA), summarization, or information retrieval	AI-assisted (using language model as a judge)	Groundedness, retrieval, and relevance metrics form a "RAG triad" that examines the quality of responses and retrieved context chunks	**Groundedness** Measures how well the generated response aligns with the given context, focusing on its relevance and accuracy with respect to the context. Groundedness Pro Detects whether the generated text response is consistent or accurate with respect to the given context. Retrieval Measures the quality of search without ground truth. It focuses on how relevant the context chunks (encoded as a string) are to address a query and how the most relevant context chunks are surfaced at the top of the list. **Relevance** Measures how effectively a response addresses a query. It assesses the accuracy, completeness, and direct relevance of the response based solely on the given query.
Generative business writing such as summarizing meeting notes, creating marketing materials, and drafting emails	AI-assisted (using language model as a judge)	Examines the logical and linguistic quality of responses	**Coherence** Measures the logical and orderly presentation of ideas in a response, allowing the reader to easily follow and understand the writer's train of thought. **Fluency** Measures the effectiveness and clarity of written communication, focusing on grammatical accuracy, vocabulary range, sentence complexity, coherence, and overall readability.
Natural language processing (NLP) tasks: text classification, natural-language understanding, and natural-language generation	AI-assisted (using language model as a judge)	Examines a response against a ground truth, with respect to a query.	**Similarity** Measures the similarity by a language model between the generated text and its ground truth with respect to a query.
NLP tasks: text classification, natural-language understanding, and natural-language generation	Natural language processing (NLP) metrics	Examines a response against a ground truth.	F1 Score , BLEU , GLEU , METEOR , ROUGE Measures the similarity by shared n-grams or tokens between the generated text and the ground truth, considering precision and recall in various ways.

Integrate Existing and New Patterns

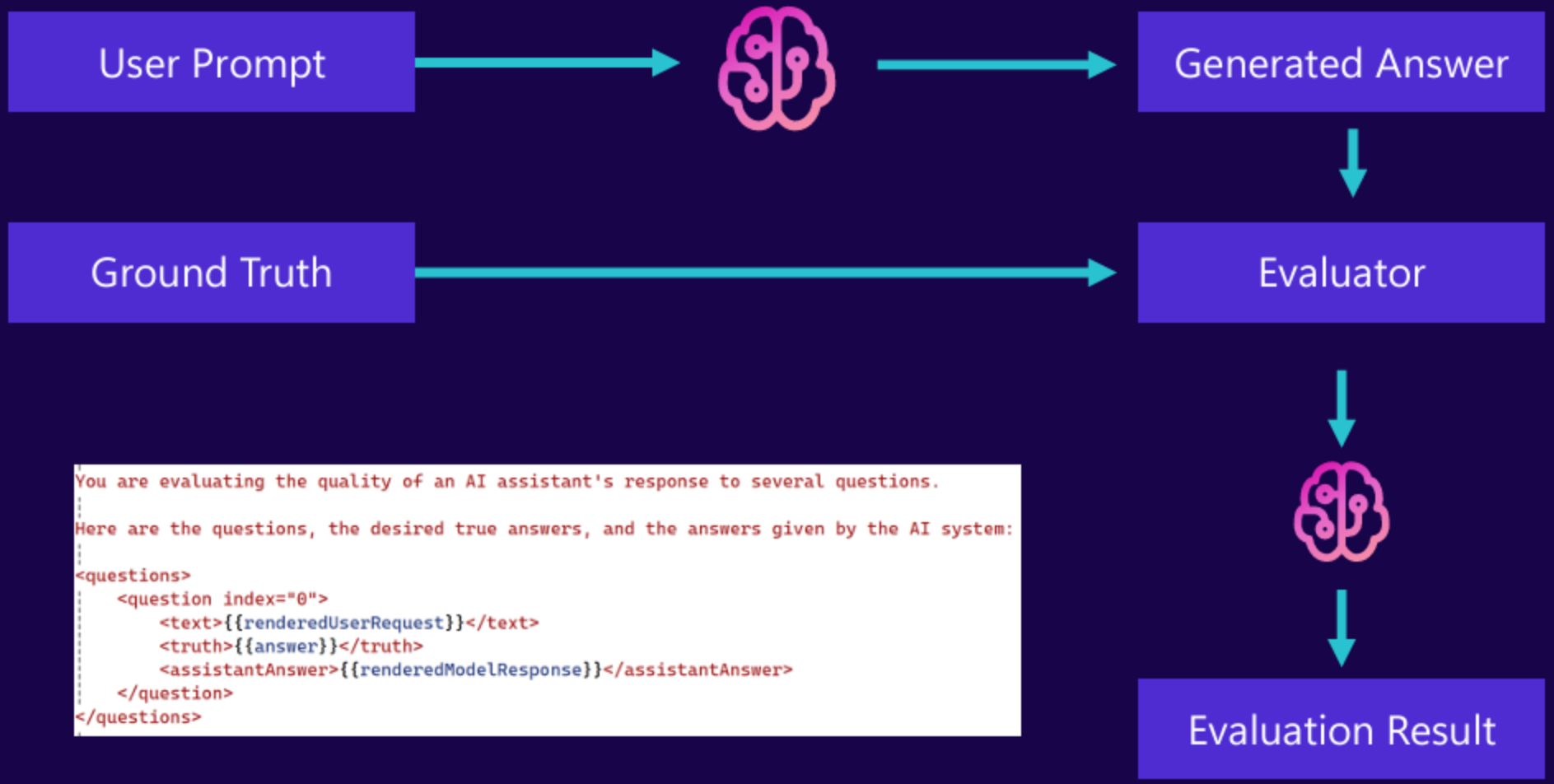
```
[Fact]
| 0 references | 0 changes | 0 authors, 0 changes
public async Task EvaluateQuestion_RelevantAnswer()
{
    var question = new EvalQuestion
    {
        QuestionId = 1,
        ProductId = 106,
        Question = "Is it machine washable?",
        Answer = "No. You should wash it by hand",
    };

    string[] metricNames = ["Relevance"];

    var evalResult =
        await EvaluateQuestion(question, reportingConfiguration, 1, CancellationToken.None);

    Assert.True(
        evalResult
            .Metrics
                .TryGetValue("Relevance", out EvaluationMetric? relevance) &&
        relevance.Interpretation?.Rating >= EvaluationRating.Good,
        $"{relevance.Interpretation?.Reason}");
}
```

Works with existing Tooling





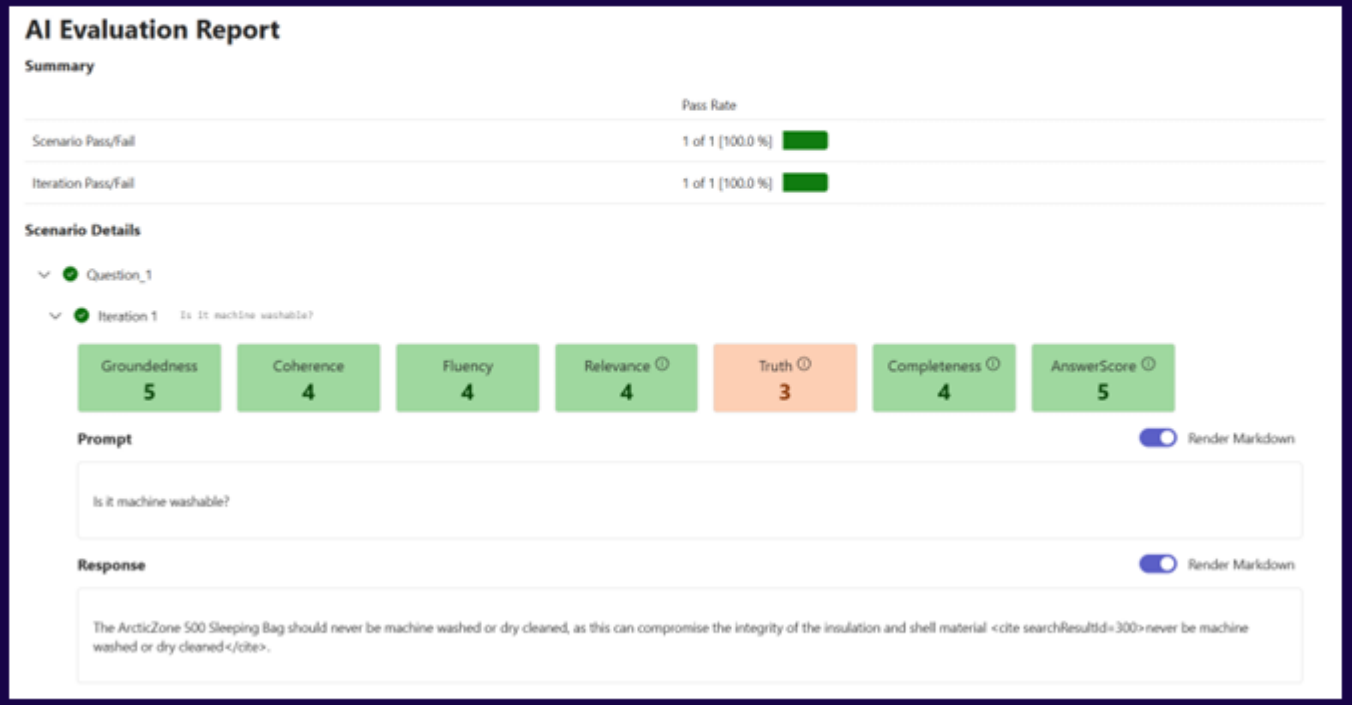
AI Evaluations




```
You are evaluating the quality of an AI assistant's response to several questions.

Here are the questions, the desired true answers, and the answers given by the AI system:

<questions>
    <question index="0">
        <text>{{renderedUserRequest}}</text>
        <truth>{{answer}}</truth>
        <assistantAnswer>{{renderedModelResponse}}</assistantAnswer>
    </question>
</questions>
```

Report Generation

```
dotnet aieval report --path C:\src\eShopCache\ --output eval-report.html
```



Demo..
AI Evaluations

04

Summary & Key Takeaways

Summary & Key Takeaways

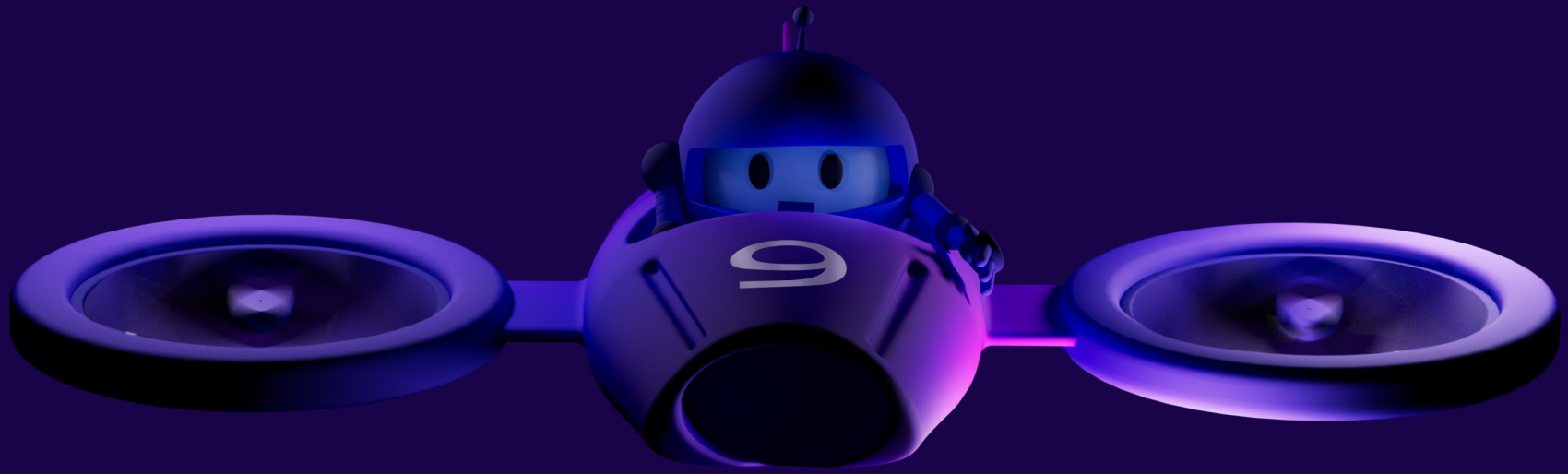
Official: Open AI SDK for .NET from Microsoft

Windows AI
WinML
Copilot Library
DirectML

Microsoft.Extensions.AI

Get .NET 9

Download .NET 9
aka.ms/get-dotnet-9



Resources

Docs: *aka.ms/dotnet/ai/docs*

Samples: *aka.ms/dotnet/ai/samples*

Thank you