



AI su .NET senza Cloud

LLM locali con Foundry Local e .NET Aspire

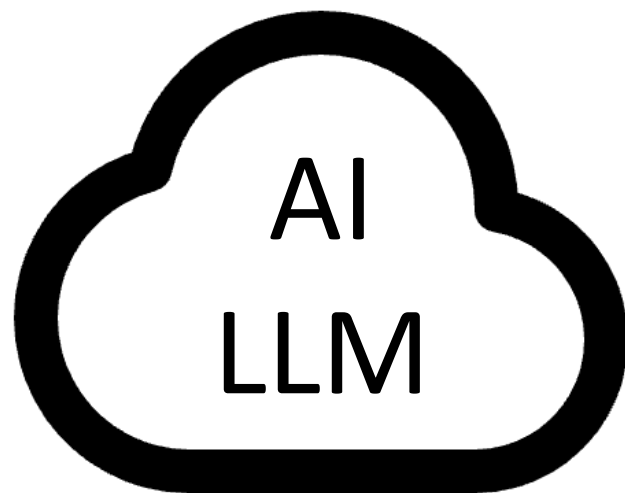




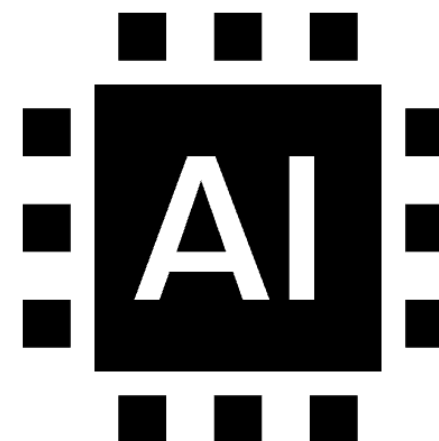
CONSORZIO
UNIVERSITARIO
DI PORDENONE
MOLTIPLICATORE DI VALORE

OVERNET.
upgrade your digital skills

[stesi]
Powered by Innovation



Foundry Local



Ollama



.NET Aspire

LLM in Locale: Motivi

Costo

- Esegui l'AI sul dispositivo locale gratuitamente.

Privacy e Residenza dei Dati

- Mantieni i dati sul dispositivo per garantire la conformità.

Latenza e Reattività in Tempo Reale

- Ad esempio, le applicazioni di gaming richiedono inferenze a bassa latenza.

Funzionalità Offline

- L'AI locale consente il funzionamento in ambienti con connettività limitata o isolati.

Esperienza per gli Sviluppatori

- Nessun problema di quote o limitazioni, il che consente iterazioni più rapide.

Sfide dei LLM Locali

Vincoli Hardware

- L'AI locale richiede spesso CPU/GPU ad alte prestazioni. I dispositivi di fascia bassa faticano con la dimensione dei modelli e la velocità di inferenza.

Dimensione e Compatibilità dei Modelli

- Molti modelli all'avanguardia sono troppo grandi per essere eseguiti efficientemente in locale senza tecniche come la quantizzazione o la distillazione.
- C'è un compromesso tra prestazioni e qualità del modello.

Aggiornamenti e Manutenzione

- Distribuire modelli in un ecosistema hardware eterogeneo significa fornire il modello migliore per l'hardware dell'utente finale.
- Gli sviluppatori devono gestire manualmente aggiornamenti dei modelli, conflitti di dipendenze e ambienti di runtime.

LLM Locali



LLaMA^{C++}



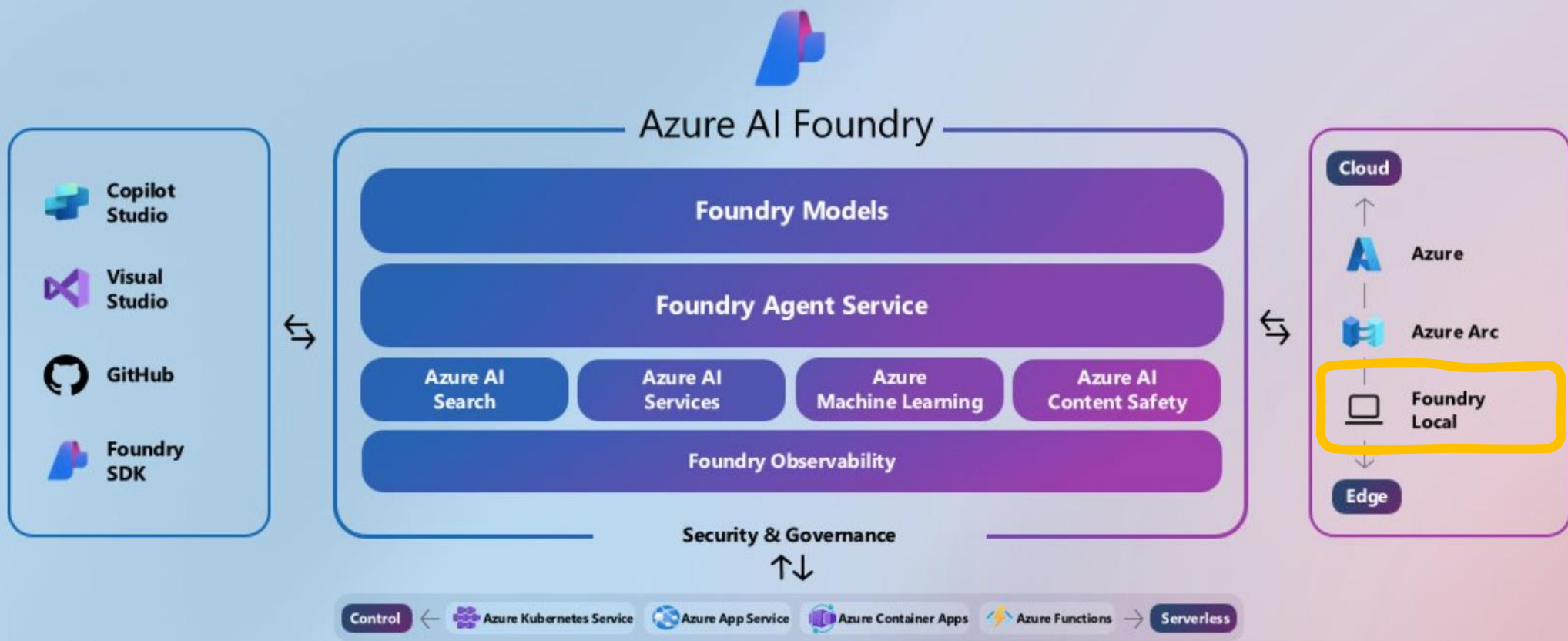
llamafire



Ollama

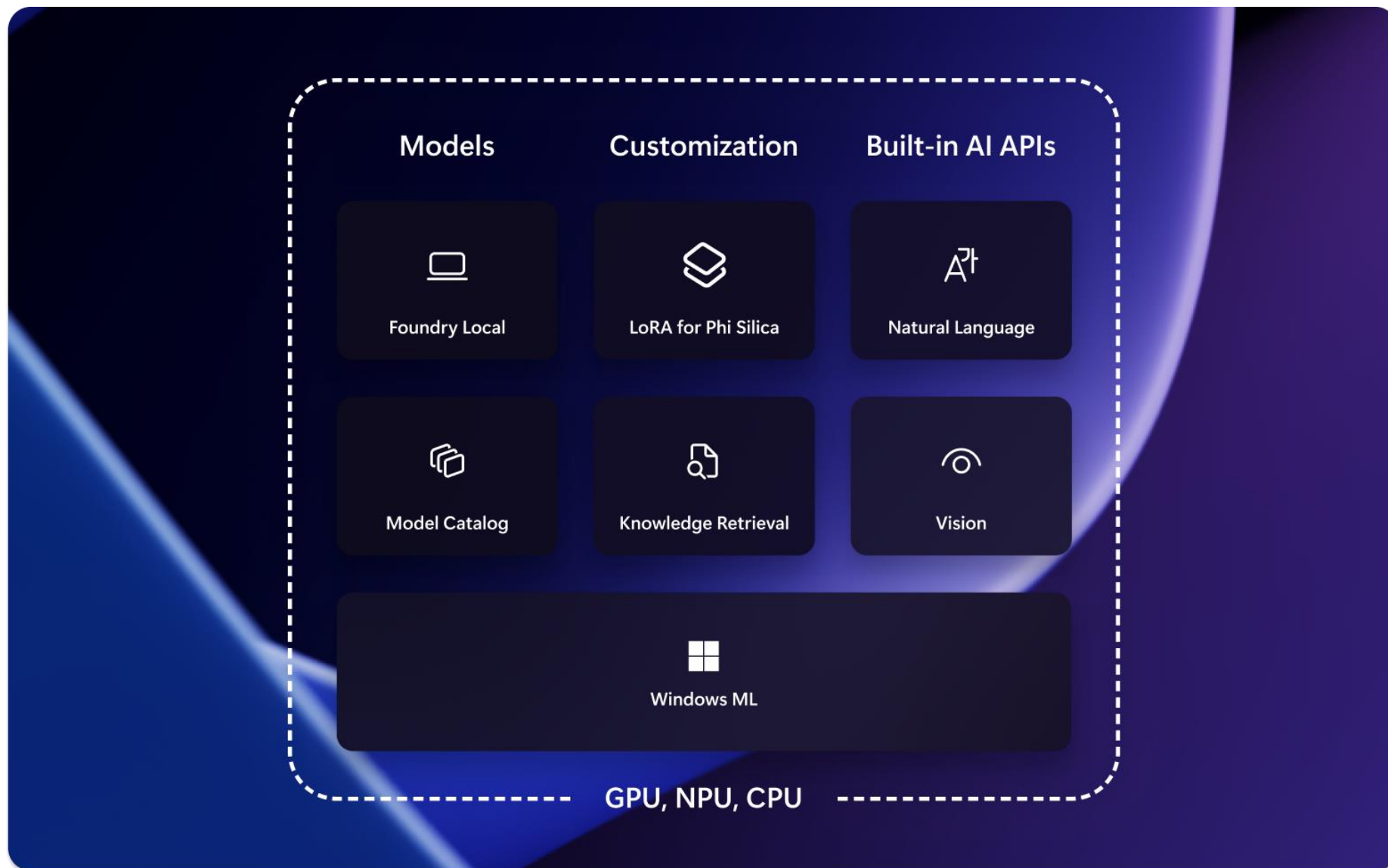


Foundry Local





Windows AI
Foundry



Foundry AI / Foundry Local / Ollama



Azure AI Foundry

!=

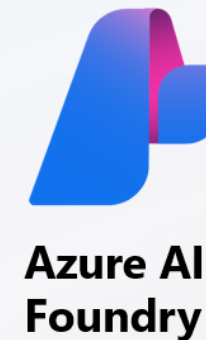
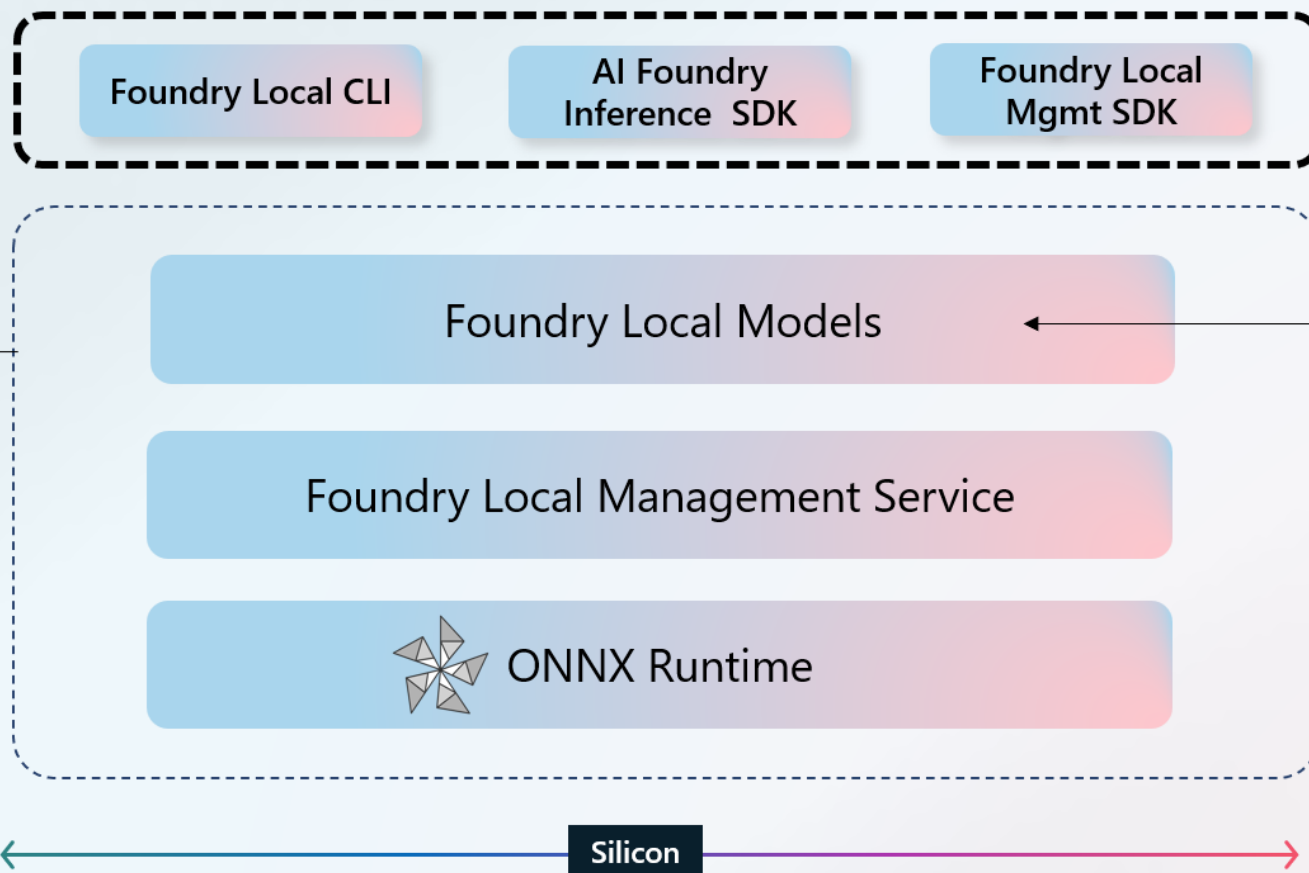


Foundry Local

>



Ollama



Available today on Windows and macOS

Foundry Local

✓ **Offre il modello migliore per l'hardware dell'utente**

- Foundry Local seleziona automaticamente la variante di modello più adatta (CPU, GPU, NPU) in base al dispositivo dell'utente.
- Runtime ottimizzati (ONNX, WebGPU) garantiscono performance elevate anche su hardware modesto.

✓ **Modelli ottimizzati per prestazioni e qualità.**

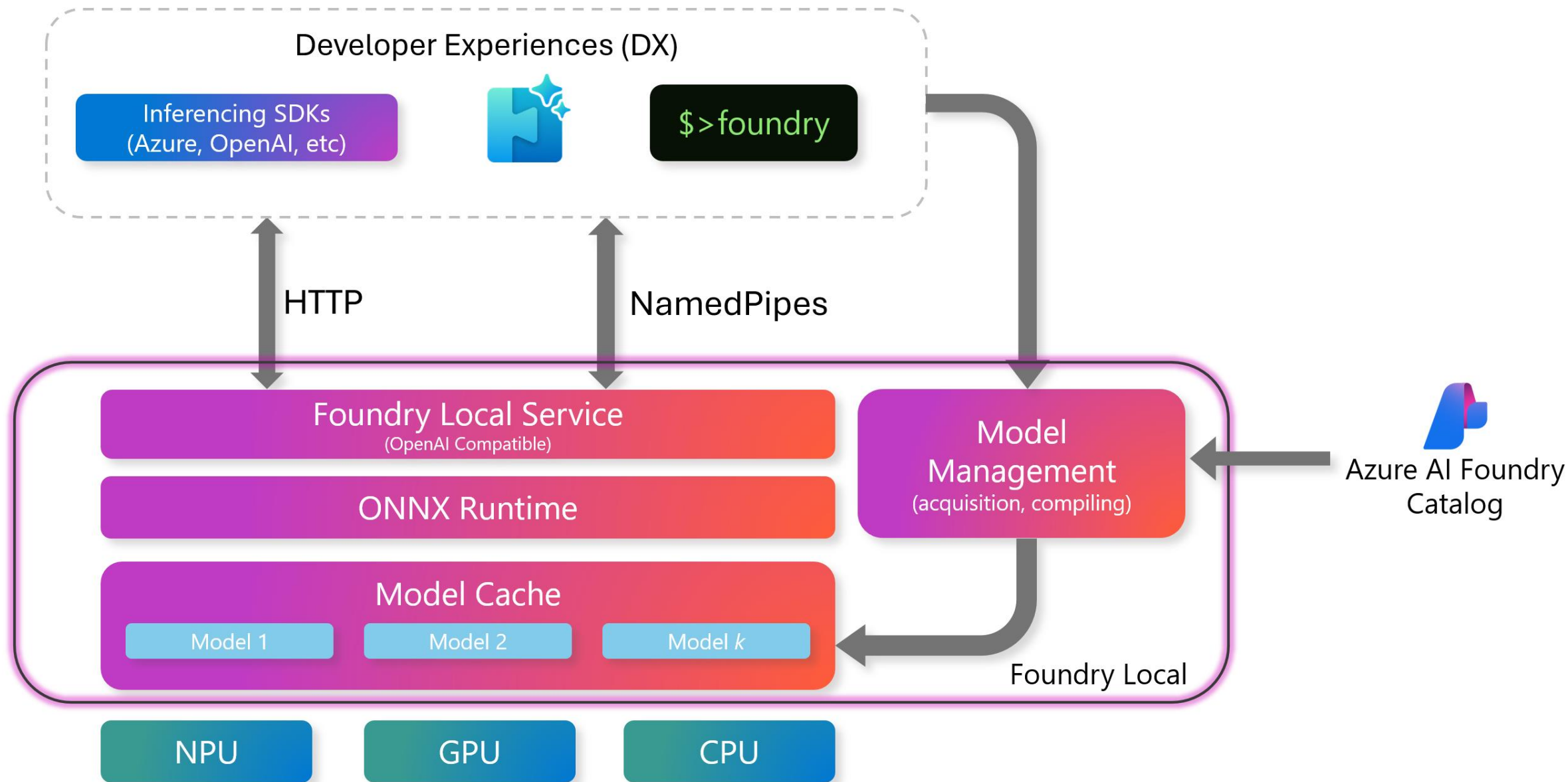
- Supporta modelli ONNX quantizzati avanzati e LLM ottimizzati per dispositivo (es. Phe4, Mitral).
- I modelli vengono scaricati dinamicamente dal catalogo Azure AI Foundry, senza bisogno di bundling.

✓ **Gestione integrata di modelli e servizi**

- Il servizio di gestione modelli si occupa di download, versionamento e caricamento a runtime.
- CLI e SDK semplificano integrazione e gestione del ciclo di vita.

✓ **Si integra con i tuoi strumenti preferiti**

- Compatibile con .NET Aspire, Azure OpenAI, Semantic Kernel, LangChain e altri.
- API compatibili OpenAI per chat e generazione facilitano l'adozione.




Come installare

1. Install Foundry Local

- **Windows:** Open a terminal and run the following command:

Bash

 Copy

```
winget install Microsoft.FoundryLocal
```

- **macOS:** Open a terminal and run the following command: `bash brew tap microsoft/foundrylocal brew install foundrylocal` Alternatively, you can download the installer from the [Foundry Local GitHub repository](#).

2. Run your first model Open a terminal window and run the following command to run a model:

Bash

 Copy

```
foundry model run qwen2.5-0.5b
```

Dove gira

- **Sistema operativo:**

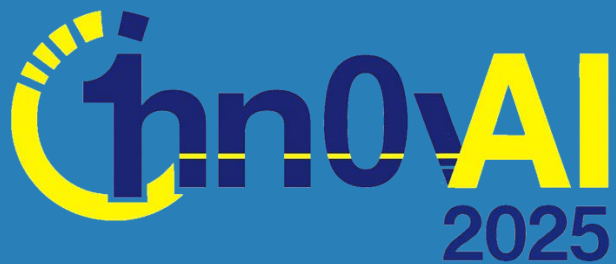
Windows 10 (x64), Windows 11 (x64/ARM), Windows Server 2025, macOS.

- **Hardware:**

almeno 8 GB di RAM. Consigliato 16 GB di RAM.

- **Accelerazione** (facoltativa):

GPU NVIDIA (serie 2.000 o successive), GPU AMD (serie 6.000 o successive), AMD NPU, Intel iGPU, Intel NPU (32 GB o più di memoria), Qualcomm Esequion X Elite (8 GB o più memoria), Qualcomm NPU o Apple silicon.



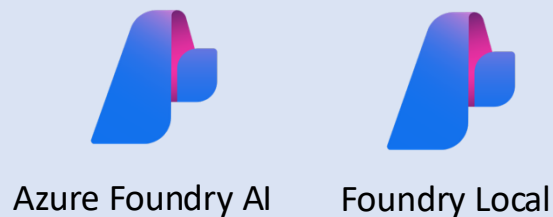
DEMO



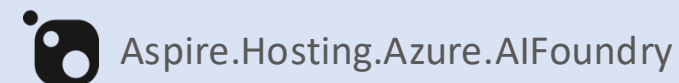
.NET Aspire

Version >= 9.4

Hosting Project



Azure AI Foundry integration (Preview)



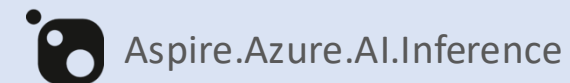
```
var builder = DistributedApplication.CreateBuilder(args);  
var foundry = builder.AddAzureAIFoundry("foundry");  
var chat = foundry.AddDeployment("chat", "Phi-4", "1", "Microsoft");
```

Client Projects

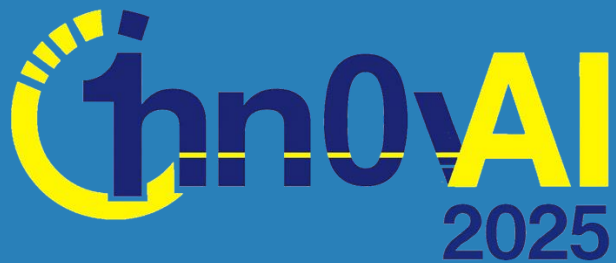


Client Applications

Azure AI Inference integration (Preview)



```
builder.AddAzureChatCompletionsClient(connectionName: "ai-foundry");
```

DEMO

Vantaggi di Aspire e Foundry Local per gli sviluppatori



.NET Aspire

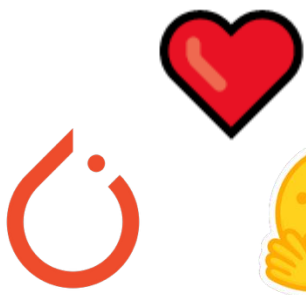


Foundry Local

- Automazione del download e aggiornamento dei modelli semplificata.
- Applicazione client che attende il caricamento dei modelli.
- Gestione automatica delle reference per semplificare l'integrazione.
- Telemetria e log integrati per monitorare le prestazioni in tempo reale e fare debug.
- Transizione facile dallo sviluppo locale ai modelli cloud con Azure AI Foundry in produzione.

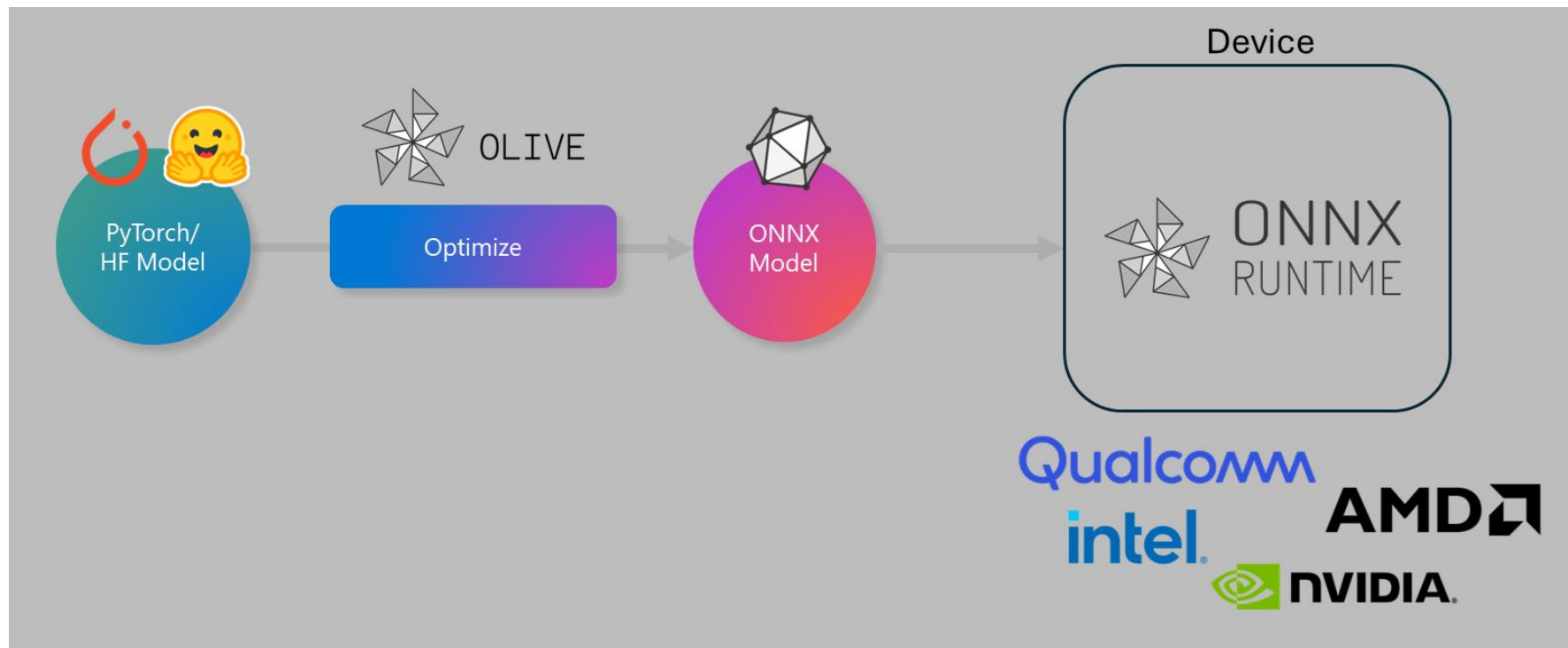


Foundry Local



PyTorch

Safetensor



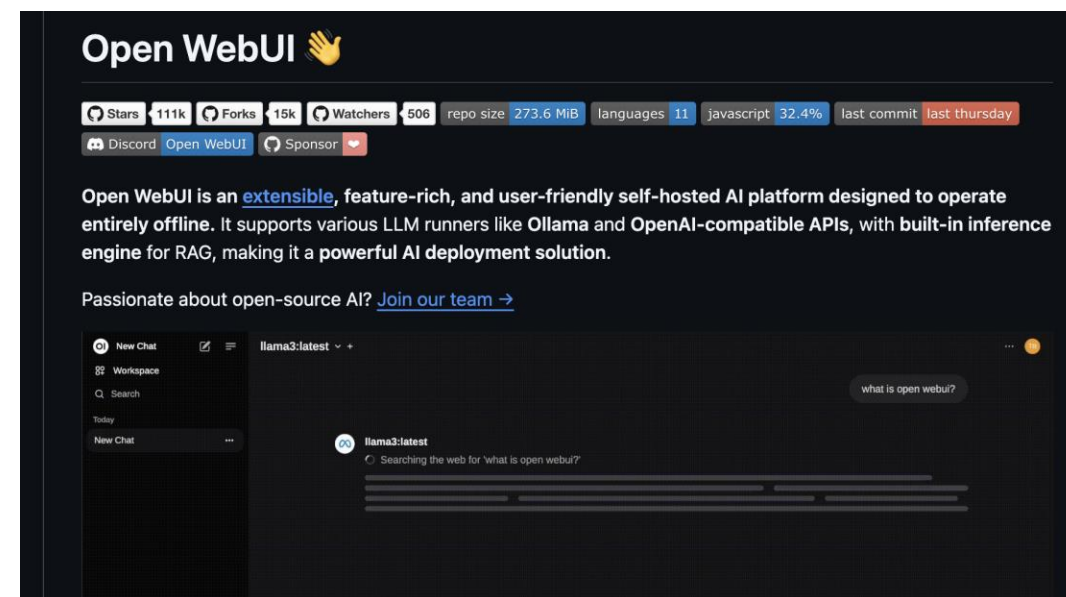
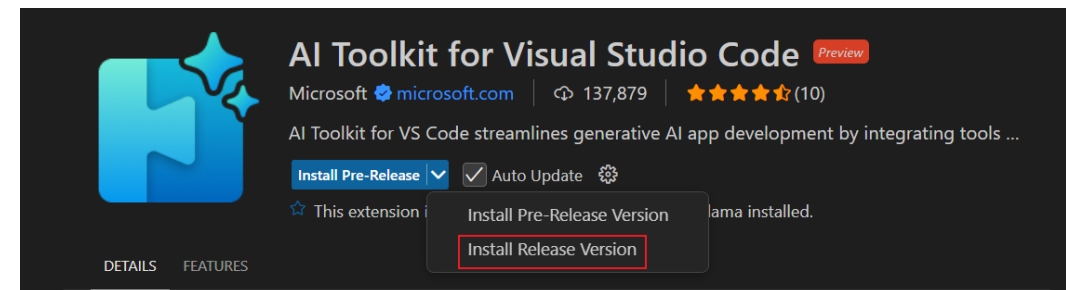
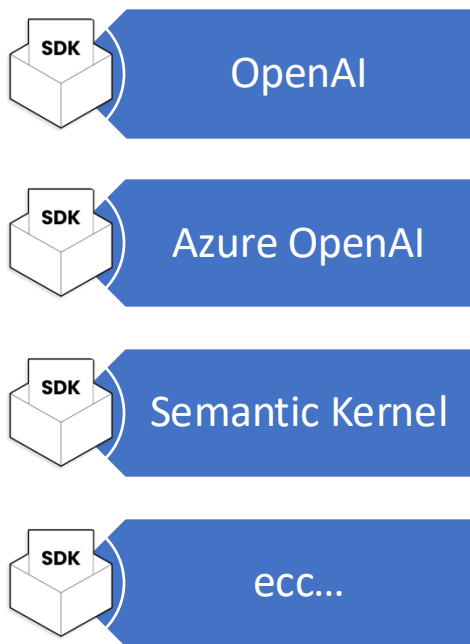
```
olive auto-opt `
  --model_name_or_path meta-llama/Llama-3.2-1B-Instruct `
  --trust_remote_code `
  --output_path models/llama `
  --device cpu `
  --provider CPUExecutionProvider `
  --use_ort_genai `
  --precision int4 `
  --log_level 1
```

<https://learn.microsoft.com/it-it/azure/ai-foundry/foundry-local/how-to/how-to-compile-hugging-face-models?tabs=Bash>



Use with

Foundry Local



Risorse utili

- <https://learn.microsoft.com/en-us/azure/ai-foundry/foundry-local/>
- <https://learn.microsoft.com/en-us/dotnet/aspire/azureai/azureai-foundry-integration?tabs=dotnet-cli>
- <https://github.com/bortolin/AspireFoundryLocalExample>
- <https://github.com/bortolin/FoundryLocalExample>
- <https://www.youtube.com/watch?v=K4rOILCzkl4>
- <https://www.youtube.com/watch?v=TLcyybeF2to&t=1795s>
- https://www.youtube.com/watch?v=HNRrHyq_GP8



Marco Bortolin

email: m.bortolin@hunext.com

twitter: @marcobortolin

<https://github.com/bortolin>

<https://www.linkedin.com/in/marcobortolin>

