# Liberi di usare l'AI

## Modelli aperti, dati e AI sul proprio computer

Moreno La Quatra

Università Kore di Enna

Liberi di usare una chat.
Ma anche di scegliere cosa c'è dentro?

# Due richieste quotidiane

## Lista della spesa

> "Prepara la **lista della spesa** usando le ricette e quello che ho in dispensa".

Una cena tra amici, senza comprare ciò che abbiamo già.

## Traduzione

> "Traduci in inglese questo **menu per la cena**".

Un testo in italiano, una versione in inglese.

# Che cos'è un LLM?

Un **Large Language Model** è un modello capace di generare testo a partire da un **contesto**.

Richiesta e contesto → Modello → Nuovo testo

Per interagire con il modello usiamo un'**applicazione**, per esempio una chat.

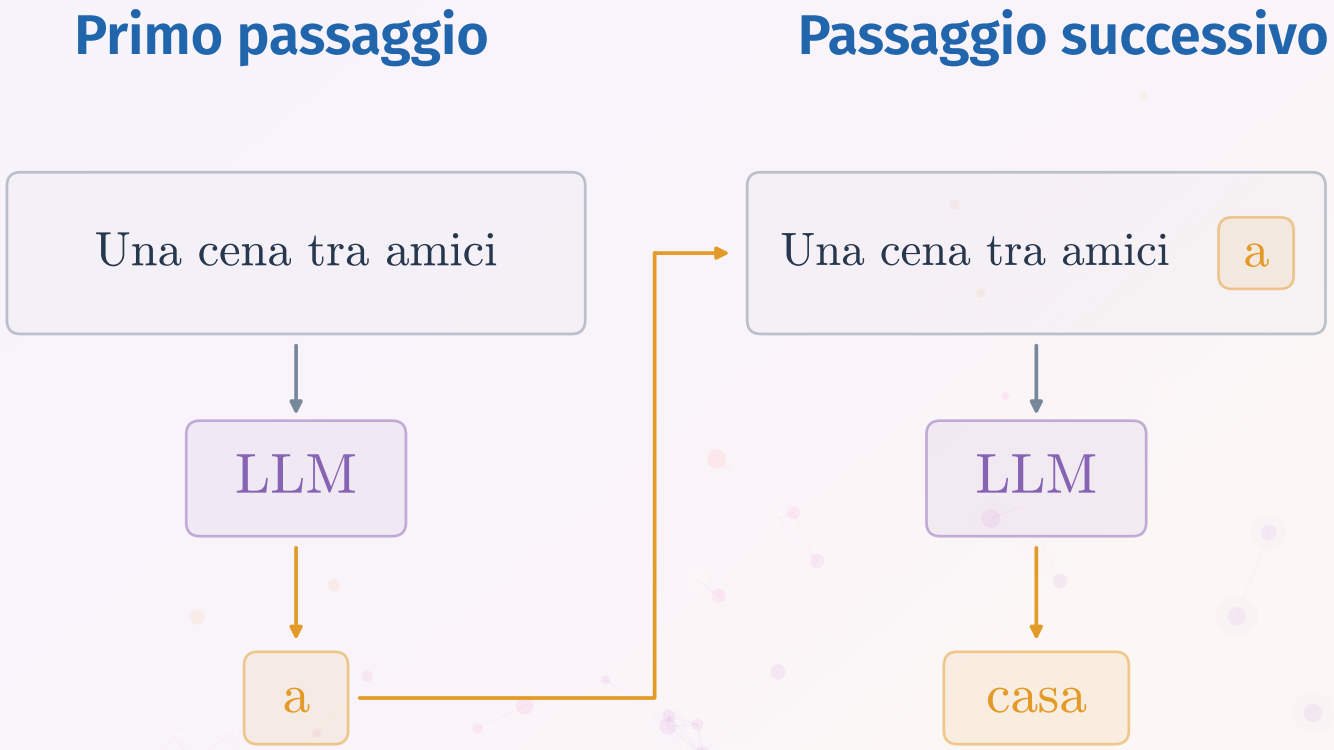
L'applicazione prepara il **contesto** da inviare al modello: la nostra richiesta, eventuali istruzioni, messaggi precedenti e testi o documenti.

Fonte: Olmo 3, scheda del modello

# Come funziona?

Il testo viene trattato come **token**: piccoli pezzi di testo, non necessariamente parole intere.

Ogni nuovo token **si aggiunge al contesto** e viene usato per generare il successivo.

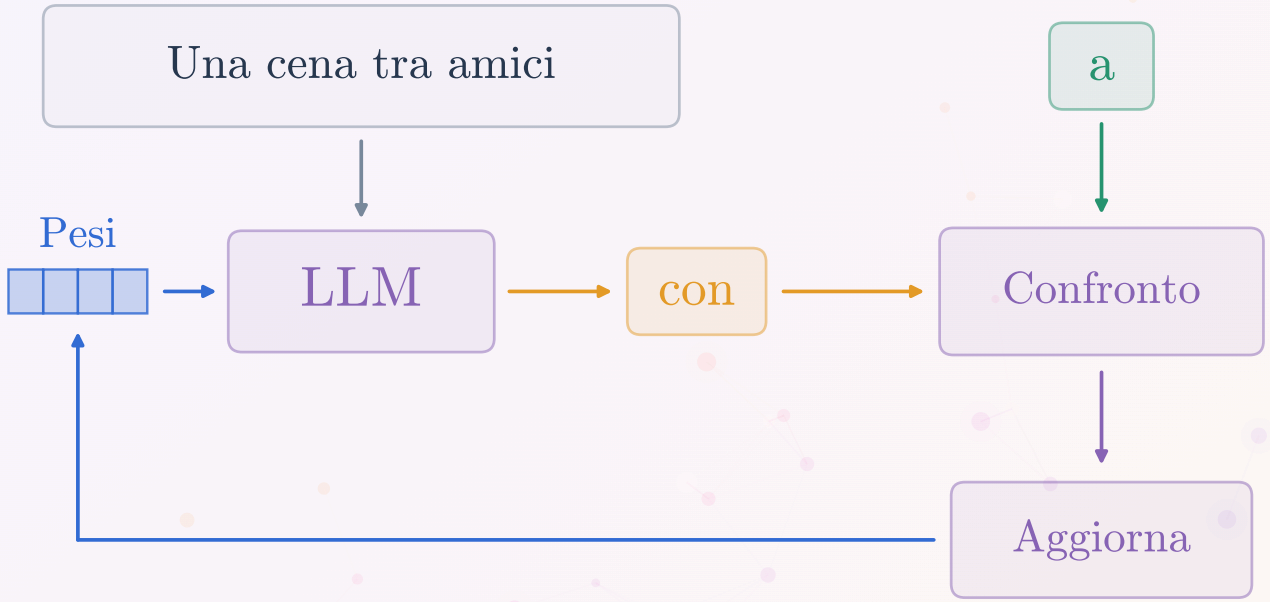



Fonte: Olmo 3, modello e documentazione

# Training

**Training**: il modello impara dai dati modificando i propri parametri (**pesi**).

Confronta il token previsto con quello presente nel testo e usa il confronto per **aggiornare i pesi**.



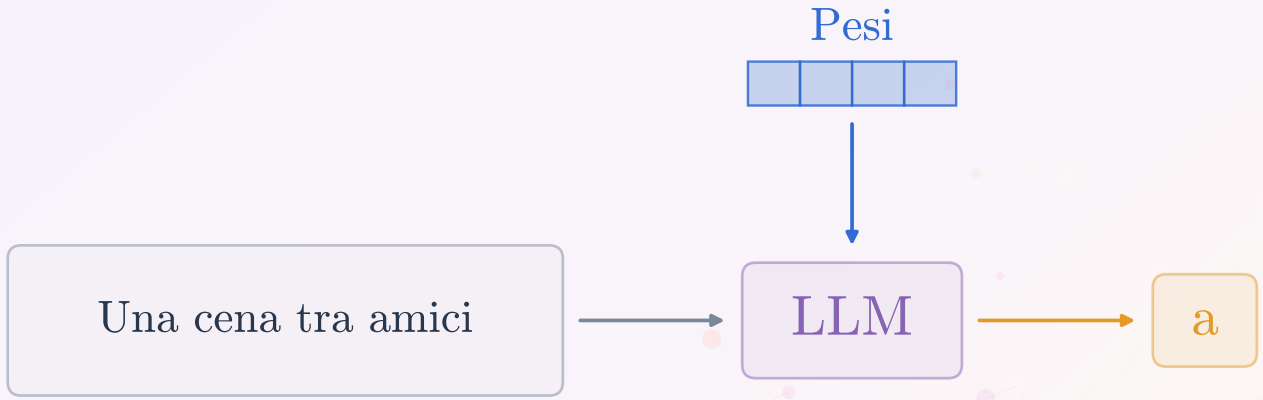

Fonte: Olmo 3, fasi e checkpoint

# Inferenza

**Inferenza**: usiamo un modello che ha già completato il training.

Il modello riceve il **contesto** e usa i **pesi** per generare il token successivo.

I **pesi restano invariati**. Il nuovo token si aggiunge al contesto.




Riferimento: Olmo 3, modello e documentazione

# L'applicazione non è il modello

## L'applicazione

È il **programma che usiamo** per interagire con il modello.

Riceve le nostre richieste, prepara ciò che viene inviato al modello e ci mostra il risultato.

## Il modello

È il componente che, dopo il training, viene **eseguito per generare** la risposta.

Può essere usato da applicazioni diverse, se sono compatibili.

La stessa applicazione può usare modelli diversi.

Riferimenti: Hermes, provider e strumenti · llama.cpp, inferenza e server

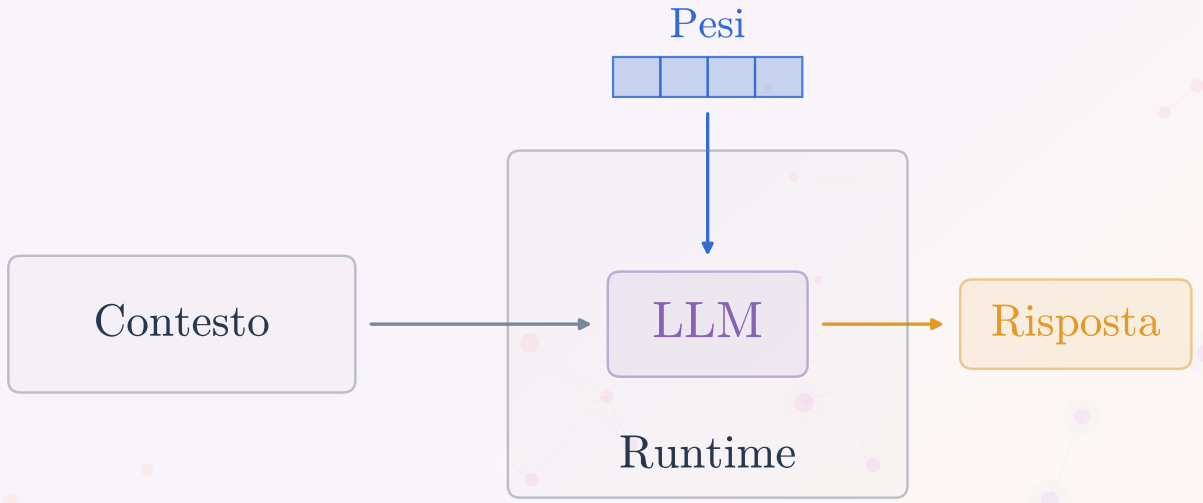

Applicazione
può cambiare
LLM 1
LLM 2
LLM 3

# Il runtime di inferenza

Il **runtime di inferenza** è il programma che **carica i pesi ed esegue il modello**.

Riceve il **contesto** ed esegue i calcoli necessari per generare la risposta.

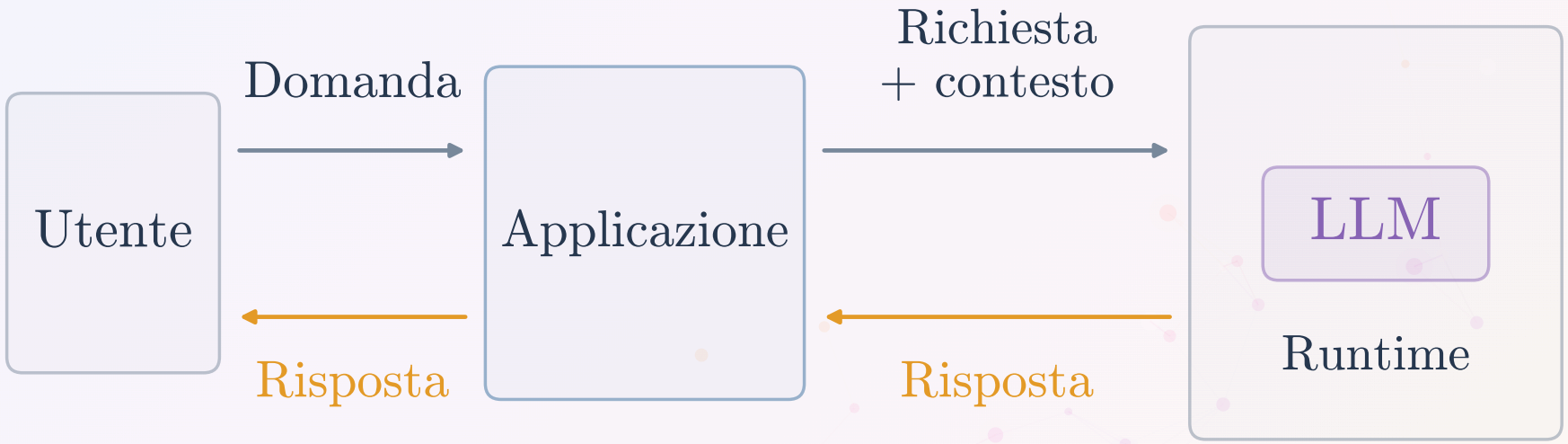
Come **LibreOffice Writer** apre un file `.odt`, un runtime come **Ollama** permette di usare i pesi di un modello.



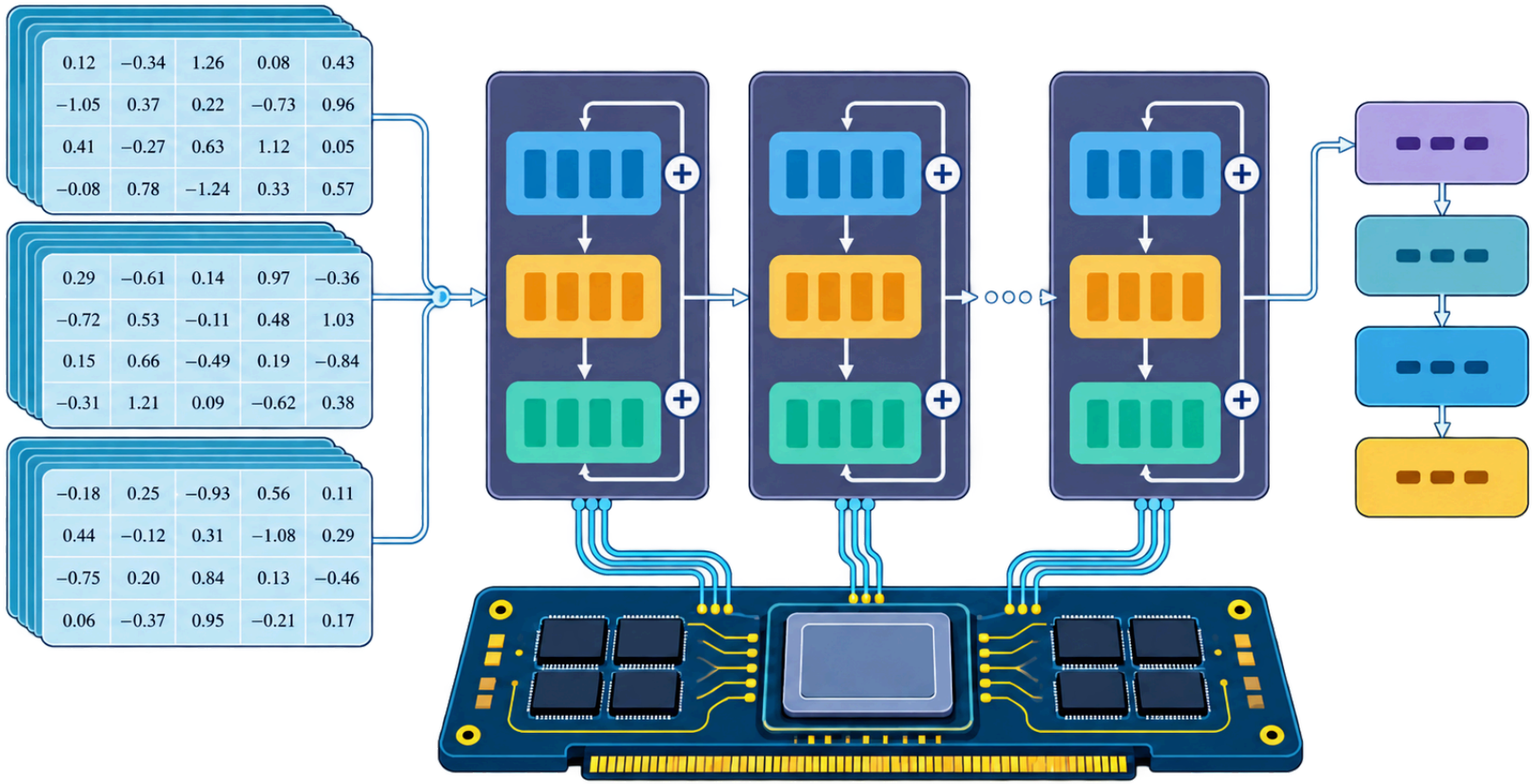

Riferimenti: Ollama · llama.cpp · LibreOffice, formato ODT

# Applicazione e runtime

L'**applicazione** prepara la richiesta; il **runtime** esegue il modello.




Riferimenti: Applicazioni e strumenti · Runtime di inferenza


0.12 −0.34 1.26 0.08 0.43
−1.05 0.37 0.22 −0.73 0.96
0.41 −0.27 0.63 1.12 0.05
−0.08 0.78 −1.24 0.33 0.57
0.29 −0.61 0.14 0.97 −0.36
−0.72 0.53 −0.11 0.48 1.03
0.15 0.66 −0.49 0.19 −0.84
−0.31 1.21 0.09 −0.62 0.38
−0.18 0.25 −0.93 0.56 0.11
0.44 −0.12 0.31 −1.08 0.29
−0.75 0.20 0.84 0.13 −0.46
0.06 −0.37 0.95 −0.21 0.17

# Quali informazioni riceve il modello?

### Incolliamo il testo

"Prepara la lista della spesa: **[ricette e dispensa]**".

Il testo è nella richiesta e diventa **parte del contesto**.

### Indichiamo una cartella

"Leggi i file nella cartella `cena/` e prepara la lista della spesa".

Abbiamo indicato **dove sono i file**, non il loro contenuto.

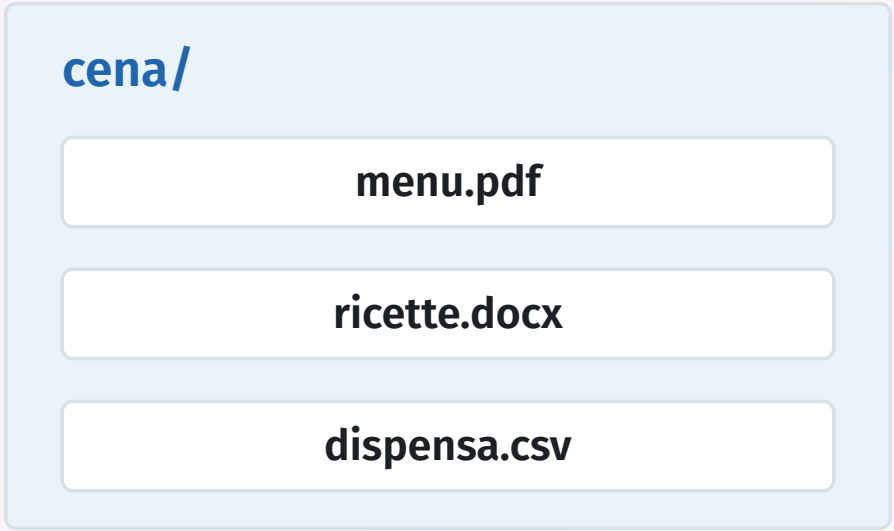
Come portiamo il contenuto dei file **nel contesto del modello**?

# Dai file al contesto

"Leggi ricette e dispensa e prepara la **lista della spesa**".

Indicare `cena/` **non fornisce al modello il contenuto** dei file.

L'**applicazione** può usare **strumenti** per leggerli e aggiungere il contenuto al **contesto**.



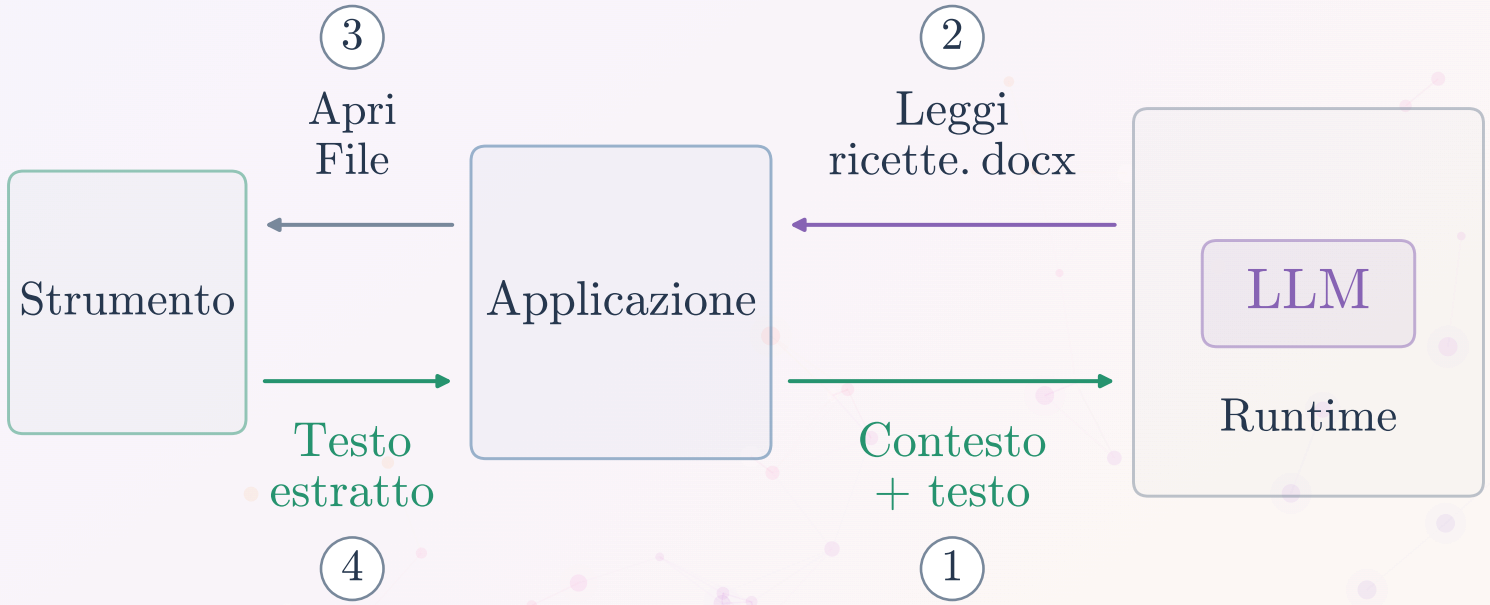

Riferimento: Hermes, strumenti

# Gli strumenti

Nel **contesto** il modello trova la richiesta e i percorsi dei file disponibili.

Chiede all'**applicazione** di leggerli. L'applicazione attiva lo **strumento** adatto: lettura di PDF, DOCX o CSV.

Il contenuto estratto, di solito **testo**, torna nel **contesto** inviato al modello.



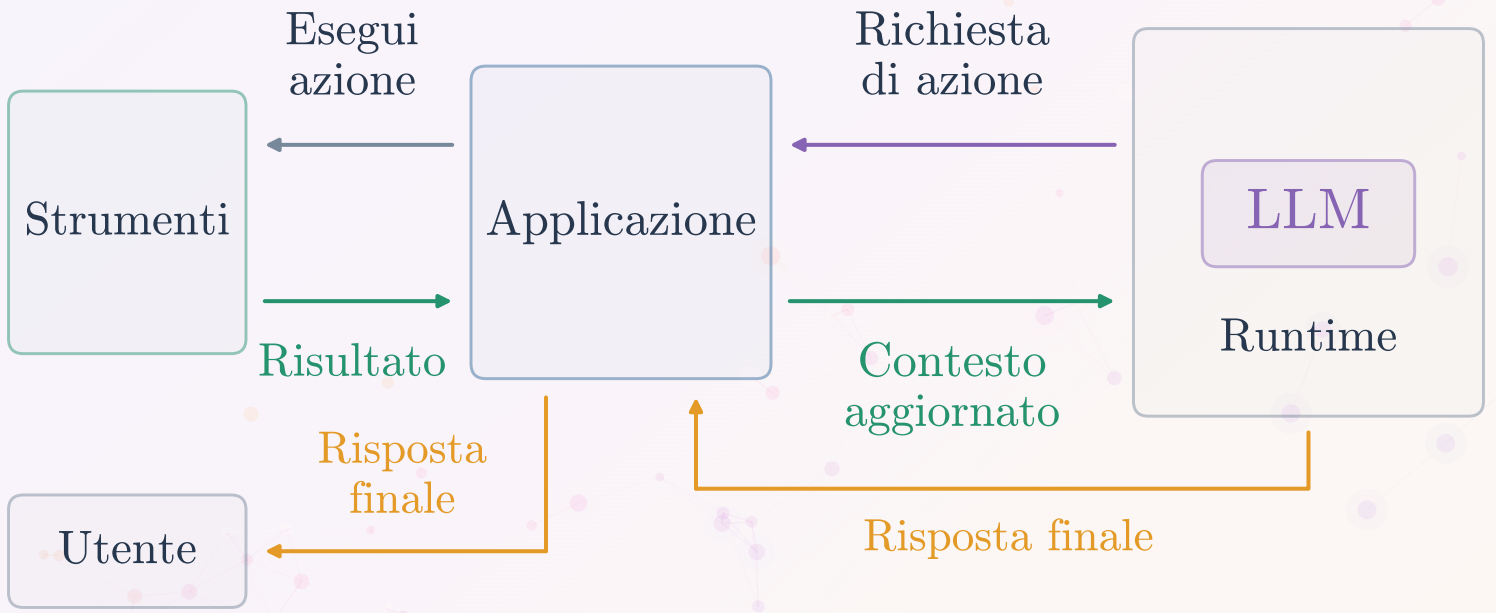

Riferimento: Strumenti e lettura dei file

# Un agente al lavoro

Per preparare la lista della spesa servono **più operazioni**, non una sola lettura.

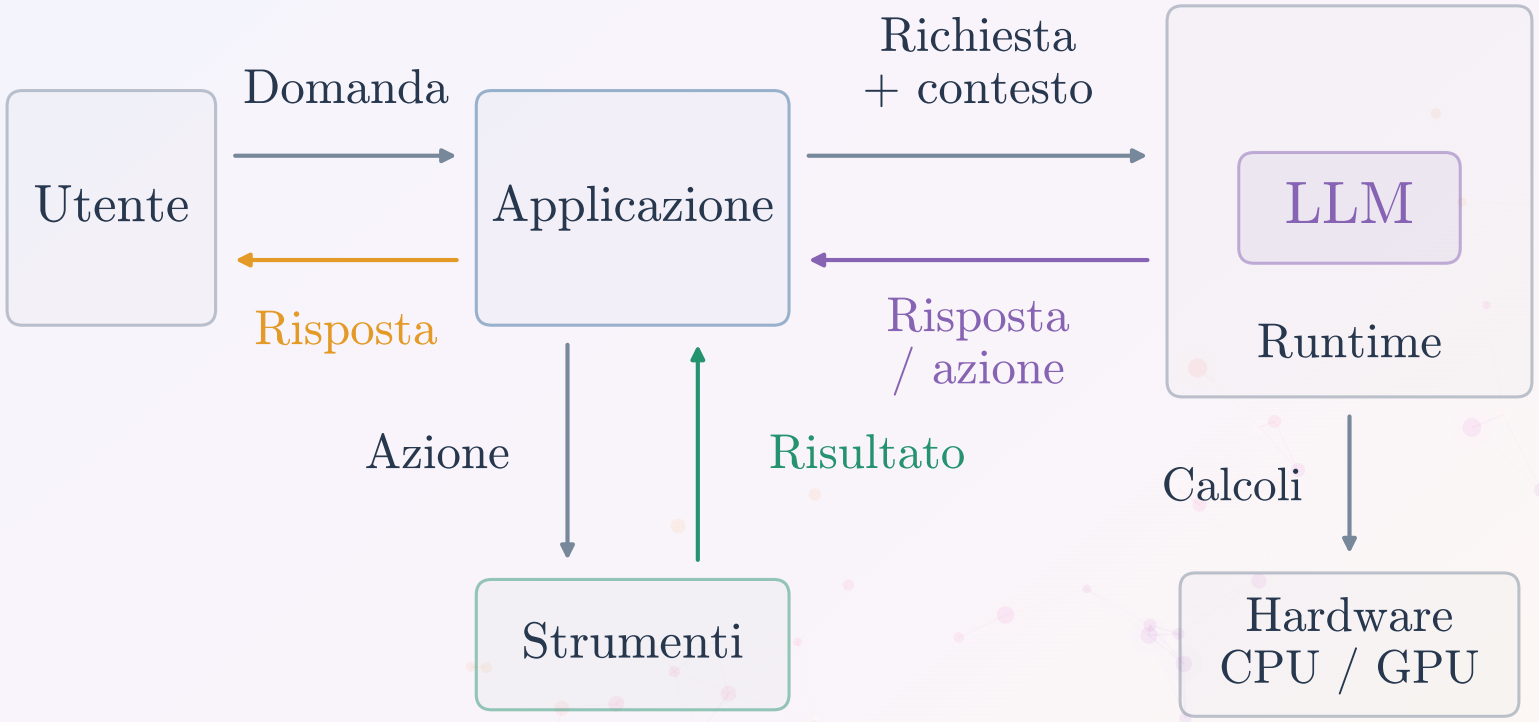
Un **agente** è un sistema che usa questo ciclo per portare avanti il compito.

Il modello può chiedere un'altra operazione oppure generare la risposta finale, che l'**applicazione mostra all'utente**.




Riferimenti: Agenti e strumenti · Ciclo di esecuzione

# I componenti insieme



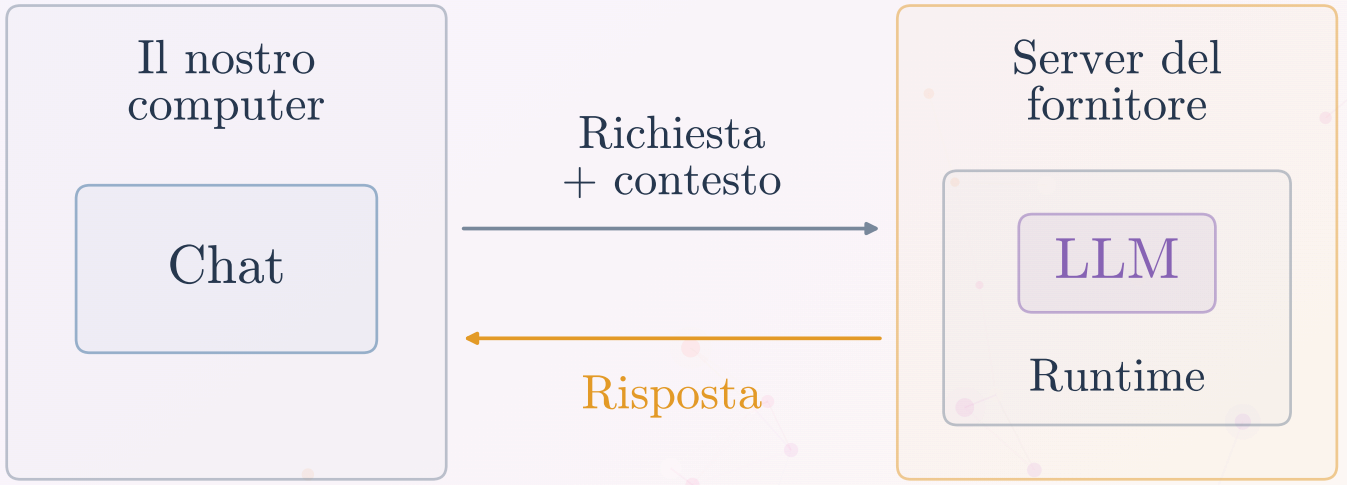

Riferimenti: Runtime di inferenza · Applicazione e strumenti

# Quando usiamo una chat online

Quando scriviamo a **ChatGPT o Gemini**, usiamo una chat nel browser.

La chat invia **richiesta e contesto** ai server dove il **runtime esegue il modello**.

Il modello **non gira sul nostro computer**.



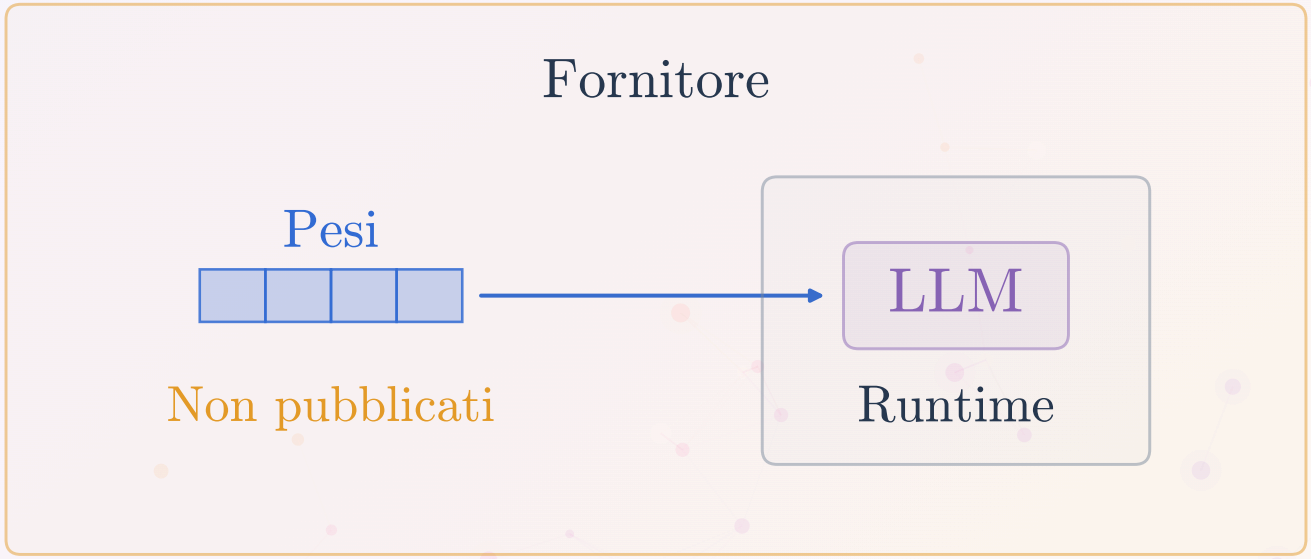

Riceviamo la risposta, **non i pesi né tutti i dettagli del modello**.

Riferimenti: OpenAI, rapporto GPT-4 · Google, Gemini Apps

# Un modello chiuso

In questo caso, il **modello proprietario** viene eseguito sui server del fornitore.

- Inviamo **richiesta e contesto**, riceviamo risposte.
- Non possiamo eseguirlo sul nostro computer.



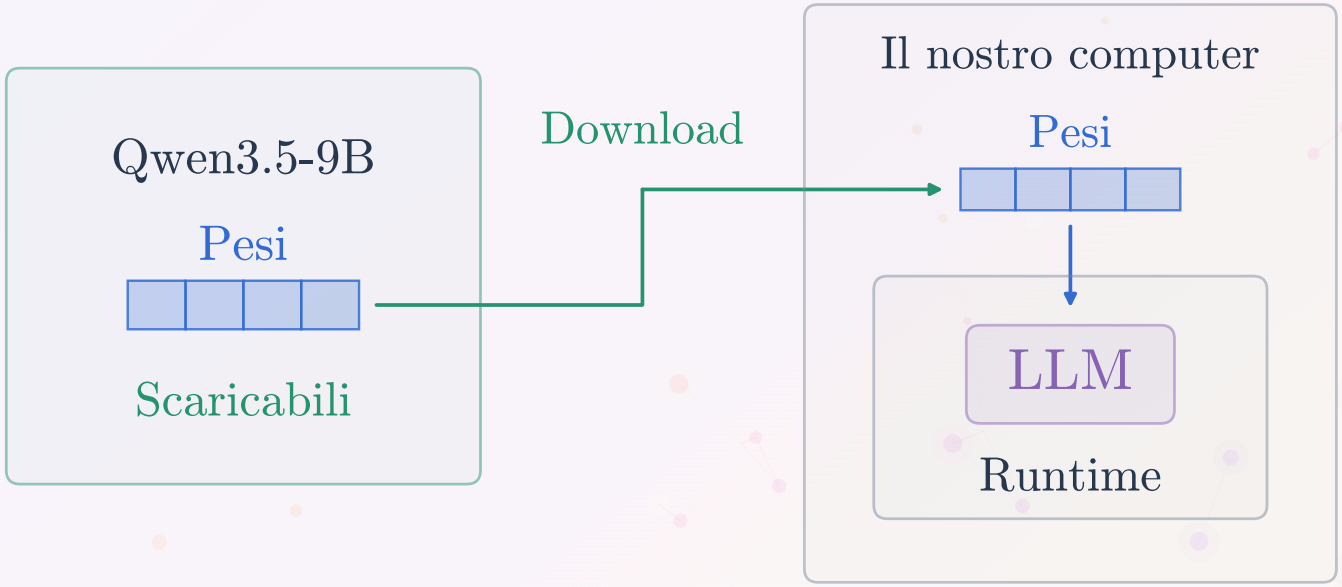

Riferimenti: GPT-4, rapporto tecnico · Dati inviati a Gemini Apps

# Modelli con pesi scaricabili

Con i modelli **open weights** possiamo **scaricare una copia dei pesi**.

Possiamo usare un **runtime compatibile** sul nostro computer, se ha risorse sufficienti.

Un esempio: **Qwen3.5-9B**.



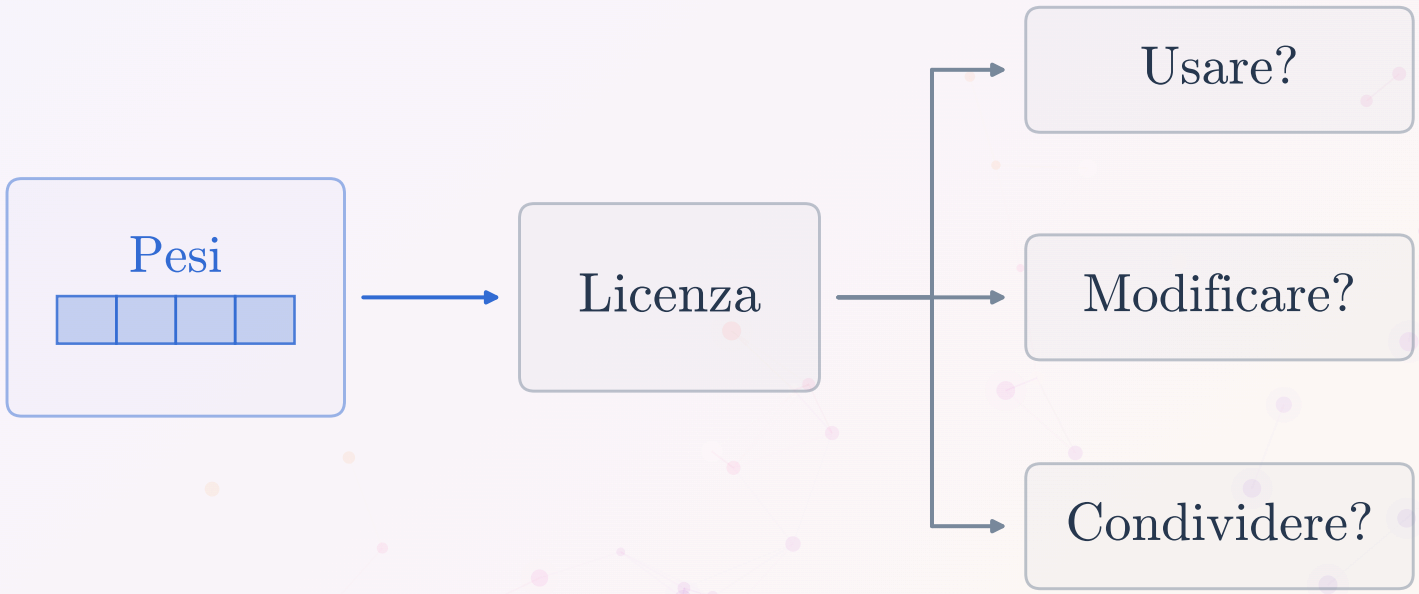

Possiamo scaricare i pesi. **La licenza stabilisce come possiamo usarli**.

Riferimenti: Qwen3.5-9B, uso locale · Licenza dei pesi

# Che cosa possiamo fare con questi pesi?

Abbiamo una copia dei pesi. Possiamo **usarli, modificarli e condividere una versione modificata**?

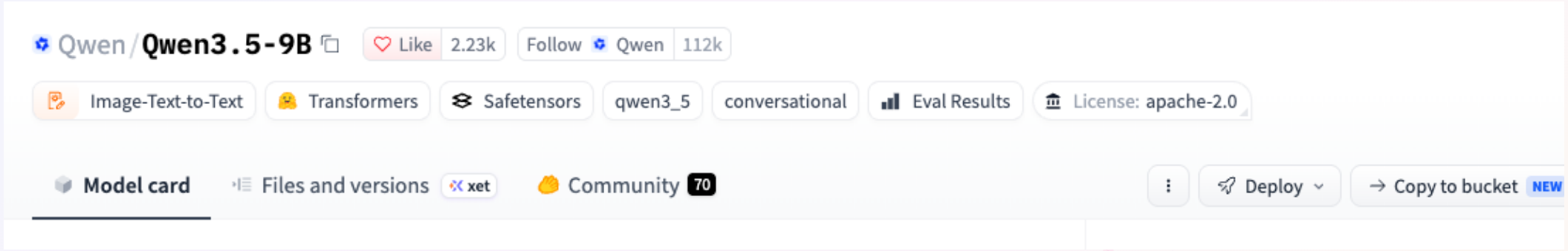
La **licenza** indica quali usi sono consentiti e a quali condizioni.




Riferimenti: Qwen3.5-9B, licenza · OSI, libertà e condizioni

# Dove leggiamo le condizioni?

Nella *model card* di **Qwen3.5-9B** troviamo **License: apache-2.0**.




La licenza chiarisce i **permessi**. Per studiare il modello servono anche i **materiali giusti**.

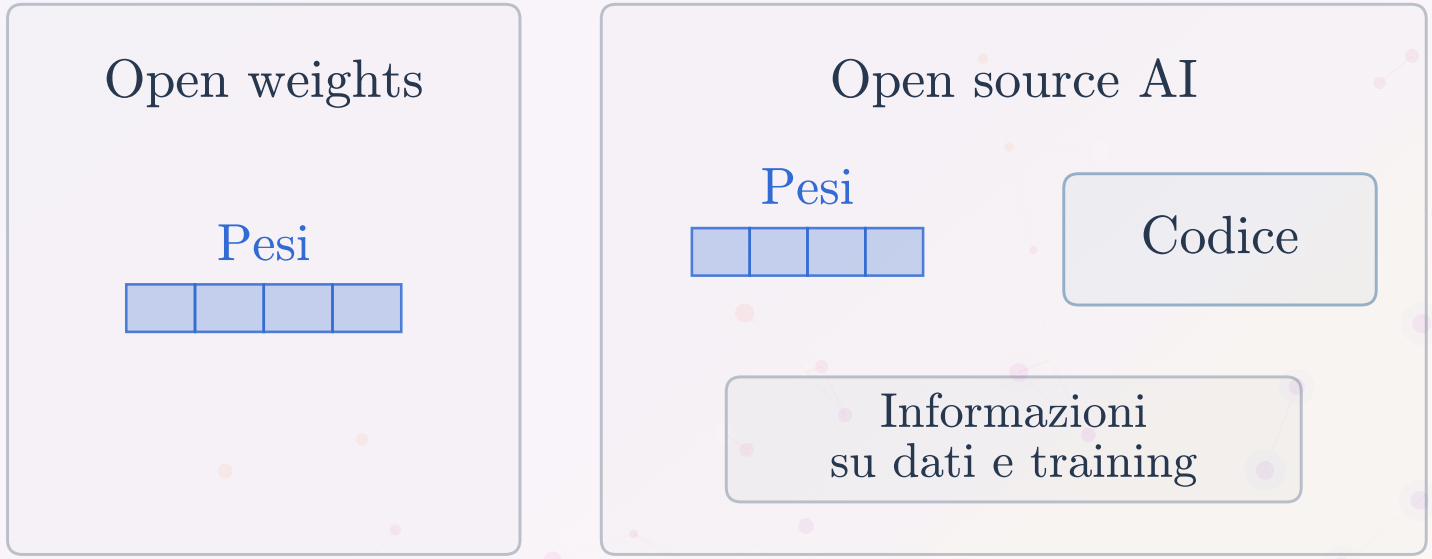
Fonti: Scheda del modello · Testo della licenza

# Open weights e open source AI

**Open weights**: i pesi sono disponibili per eseguire il modello.

Nel riferimento **Open Source AI di OSI**, servono anche **codice e informazioni su dati e training**.

Materiali e condizioni devono permettere di **usare, studiare, modificare e condividere il sistema**.

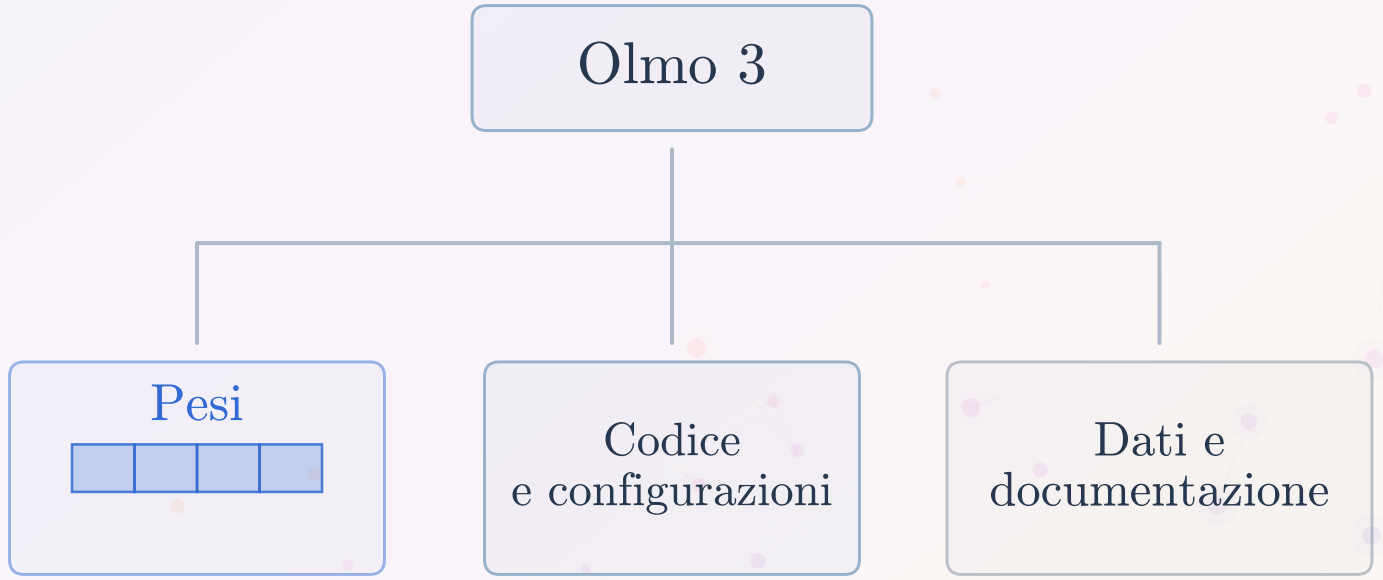



Fonti: OSI, Open Source AI Definition 1.0 · OSI, informazioni sui dati

# Olmo 3

**Olmo 3** pubblica materiali per studiare **anche come si è svolto il training**.

Per usare un modello resta poi una scelta pratica: **dove eseguirlo**.

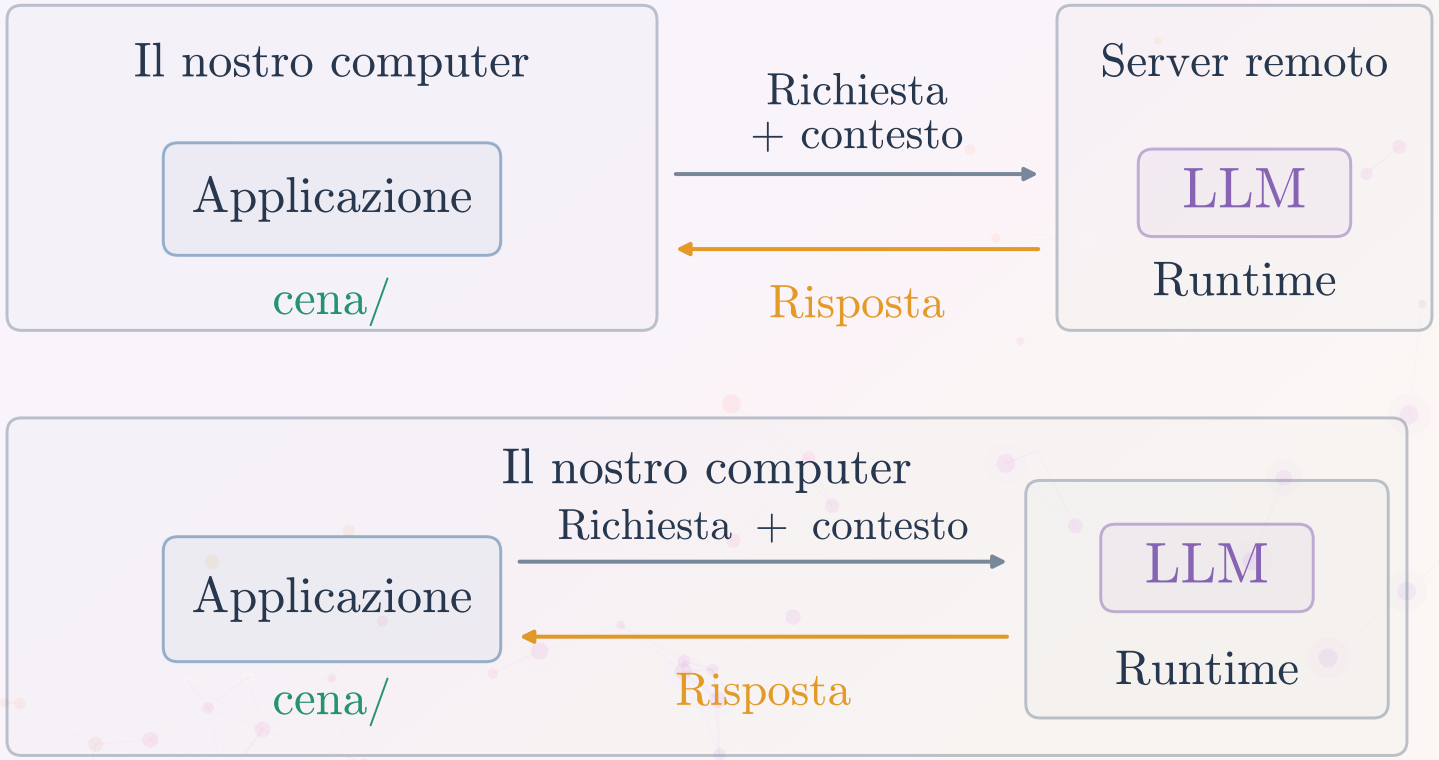



Fonti: Olmo 3 · OLMo-core · Dolma 3 Pool

# Dove passano i file della cena?

Torniamo a **ricette e dispensa**: il loro contenuto entra nel contesto.

Con un runtime **remoto**, quel contesto lascia il computer. Con un runtime **locale**, l'inferenza avviene sul computer.




Riferimenti: Ollama, inferenza locale e cloud · Applicazioni e provider

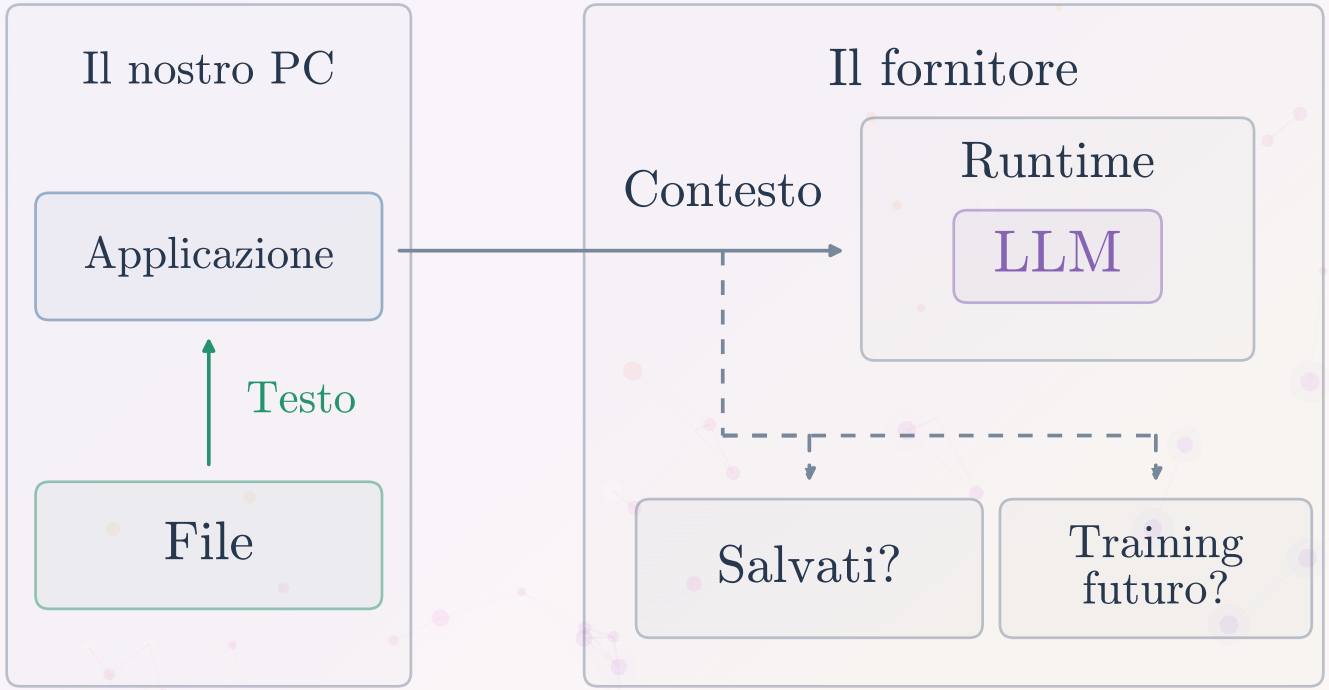
# In remoto, affidiamo i dati al fornitore

Il fornitore riceve il **contesto**, che può contenere anche **dati personali estratti dai file**.

Oltre a generare la risposta:

- **Conserva** questi dati? Per quanto tempo?
- Li usa per il **training di altri modelli**?

Dipende da **servizio, piano e impostazioni**: dobbiamo verificarlo prima di inviare i dati.



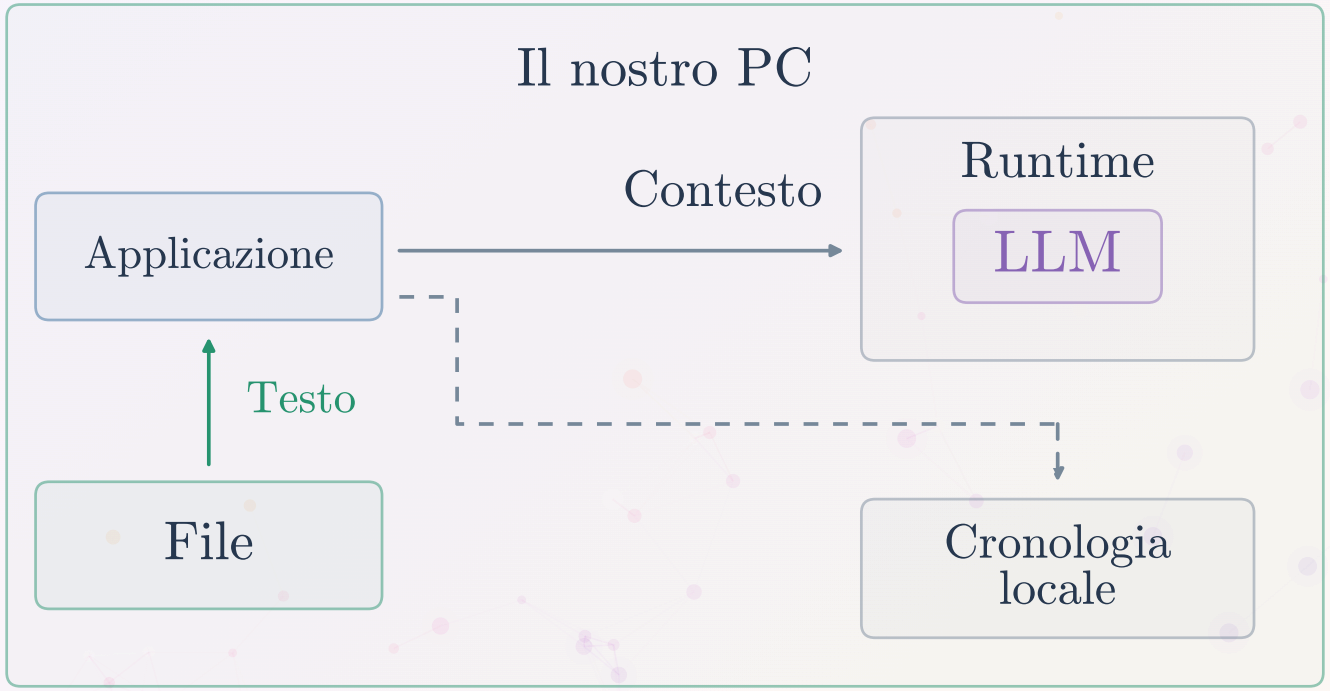

Riferimenti: Trattamento dei dati in Gemini Apps · Inferenza locale e cloud

# In locale, teniamo i dati sul nostro PC

Con i **pesi scaricabili**, possiamo eseguire il modello sul nostro computer.

**File, contesto e risposte restano nel PC.** Decidiamo noi che cosa conservare e cancellare.

Per lavorare senza inviare dati all'esterno, anche **applicazione e strumenti devono usare solo risorse locali**.




Riferimenti: Esecuzione locale · Accessi degli strumenti

# Demo Time!

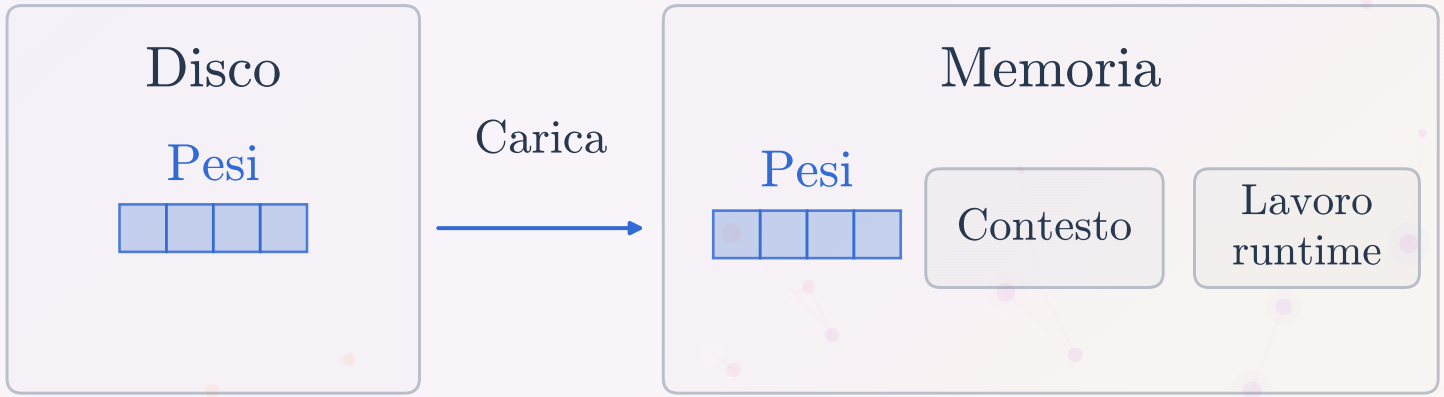


# Cosa serve sul nostro computer?

Modello **già addestrato**.

**Programmi compatibili**, memoria libera e tempi accettabili sul nostro compito.

Il file occupa spazio **sul disco**. Durante l'uso serve memoria anche per **contesto e lavoro del runtime**.




Riferimenti: llama.cpp, esecuzione dei modelli · Memoria durante l'esecuzione

# Applicazioni e agenti open source

Scegliamo l'applicazione con cui dare istruzioni e lavorare sui file.

| Progetto | Per cosa usarlo | Licenza |
|---|---|---|
| Hermes Agent | Conversazioni, file e strumenti. **Nella demo**. | MIT |
| Pi | Un agente da terminale minimale ed estensibile. | MIT |
| Codex CLI | Lavorare sul codice: lettura, modifiche e test. | Apache-2.0 |

Questi programmi usano un modello tramite un runtime: non sono il modello.

# Runtime di inferenza open source

Il runtime **carica i pesi ed esegue il modello**.

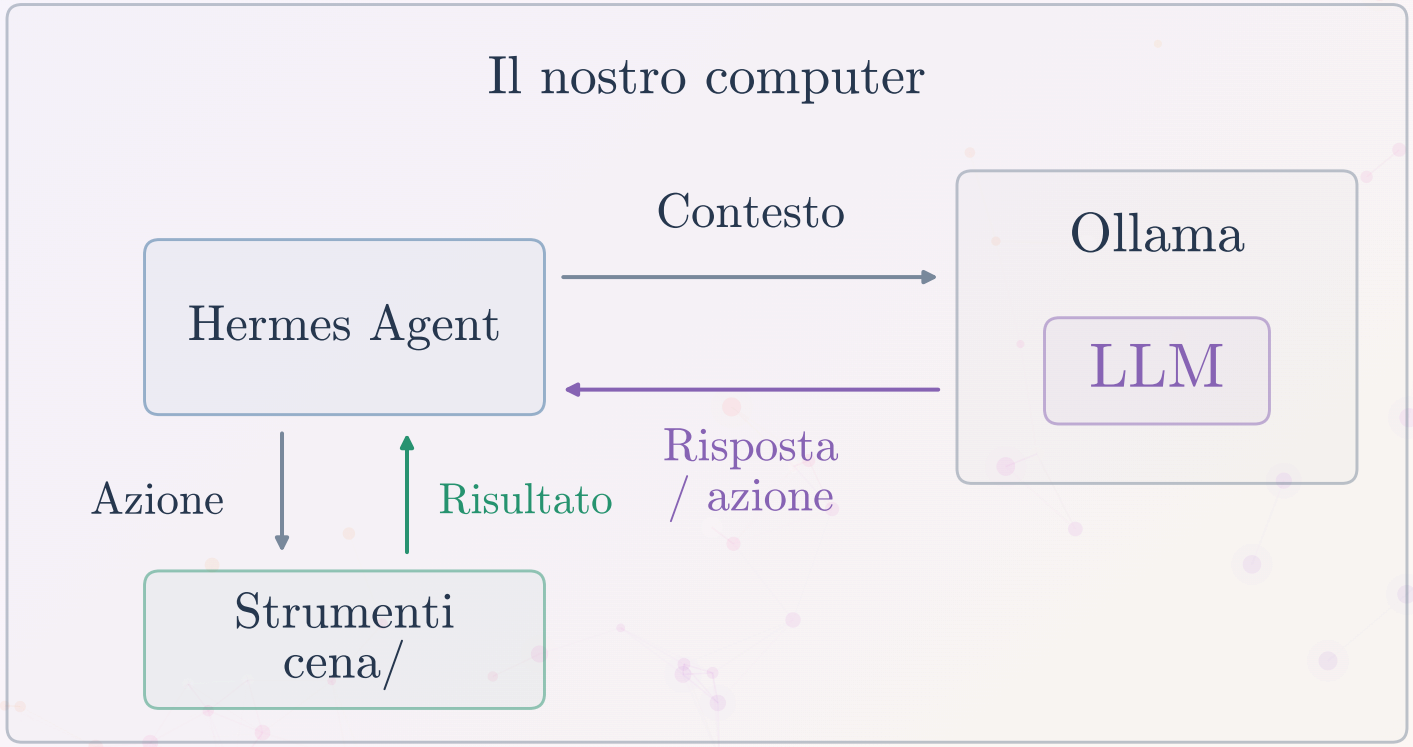
| Progetto | Per cosa usarlo | Licenza |
|---|---|---|
| Ollama | Gestire ed eseguire modelli. **Nella demo**. | MIT |
| llama.cpp | Eseguire modelli su hardware diversi. | MIT |
| vLLM | Eseguire modelli e servire più richieste contemporanee. | Apache-2.0 |
| DwarfStar / ds4 | Eseguire modelli selezionati su hardware supportato. | MIT |

# La configurazione della demo

**Hermes Agent** gestisce la conversazione e gli strumenti per i file.

**Ollama** carica ed esegue il modello localmente.

Usiamo la combinazione **già provata per questa demo**.




Riferimenti: Hermes Agent · Strumenti · Ollama

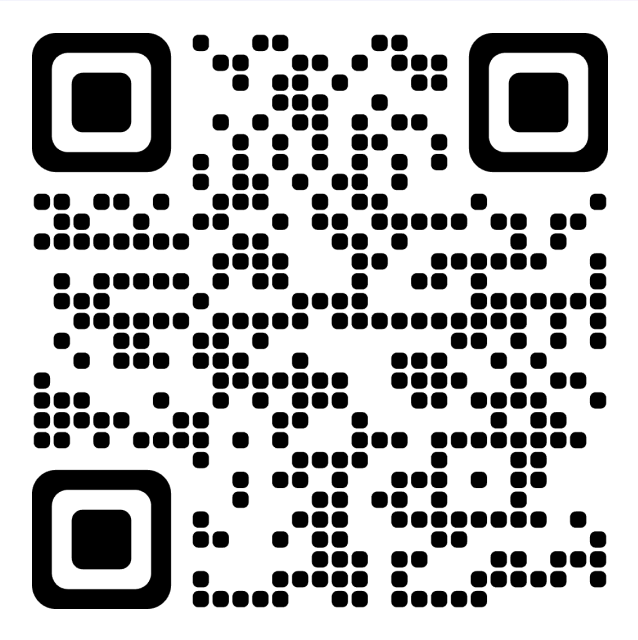


# Quindi quando torno a casa?

1. **Parti dal tuo hardware.** Scegli un modello compatibile con memoria, GPU e sistema operativo.
2. **Scegli l'app per il tuo lavoro.** Attività generali e file: **Hermes**. Codice: **Pi / Codex CLI**.
3. **Scegli un runtime compatibile:**
   - **Uso personale:** Ollama / llama.cpp.
   - **Più utenti o un servizio aziendale:** vLLM.
   - **Ottimizzazioni per modelli recenti selezionati:** ds4, su hardware supportato.

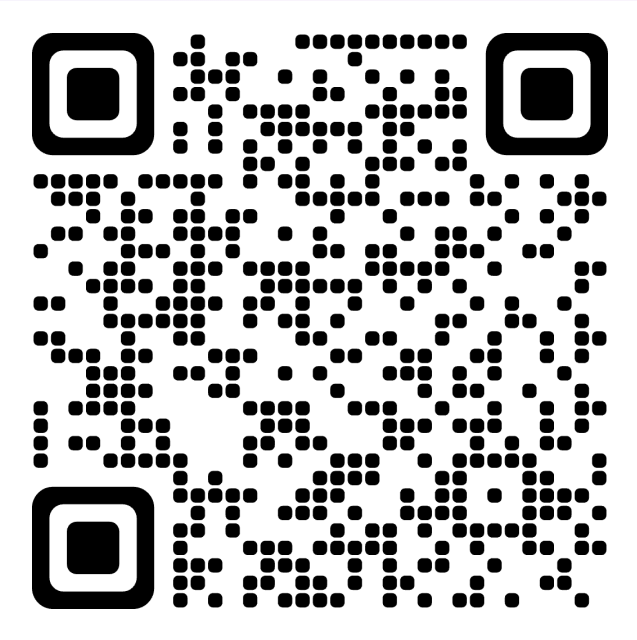
# Slide, risorse e guide

## Slide del talk



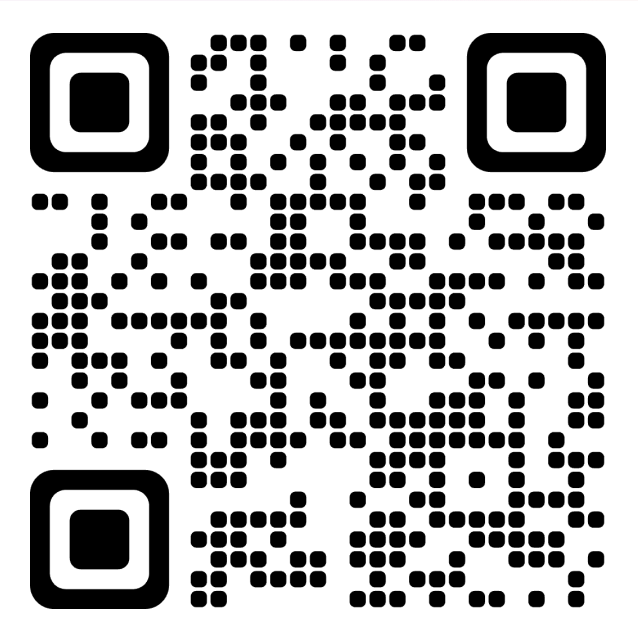
Presentazione HTML

## Risorse



File demo e riferimenti

## Guida



Linux e macOS

# Grazie!



mlaquatra.me

moreno.laquatra@unikore.it