DotNet Liguria

Casi d'uso e architettura di un Copilot personalizzato

Ing. Raffaele Rialdi
Senior Software Architect - Consultant
@raffaeler - raffaeler@vevy.com

Perché aggiungere un Copilot alle nostre app?

Casi d'uso e strategie

Classificazione

- Estrarre categorie, anomalie, sentimenti o decisioni basate su dati non strutturati (testo, audio, immagini, video)

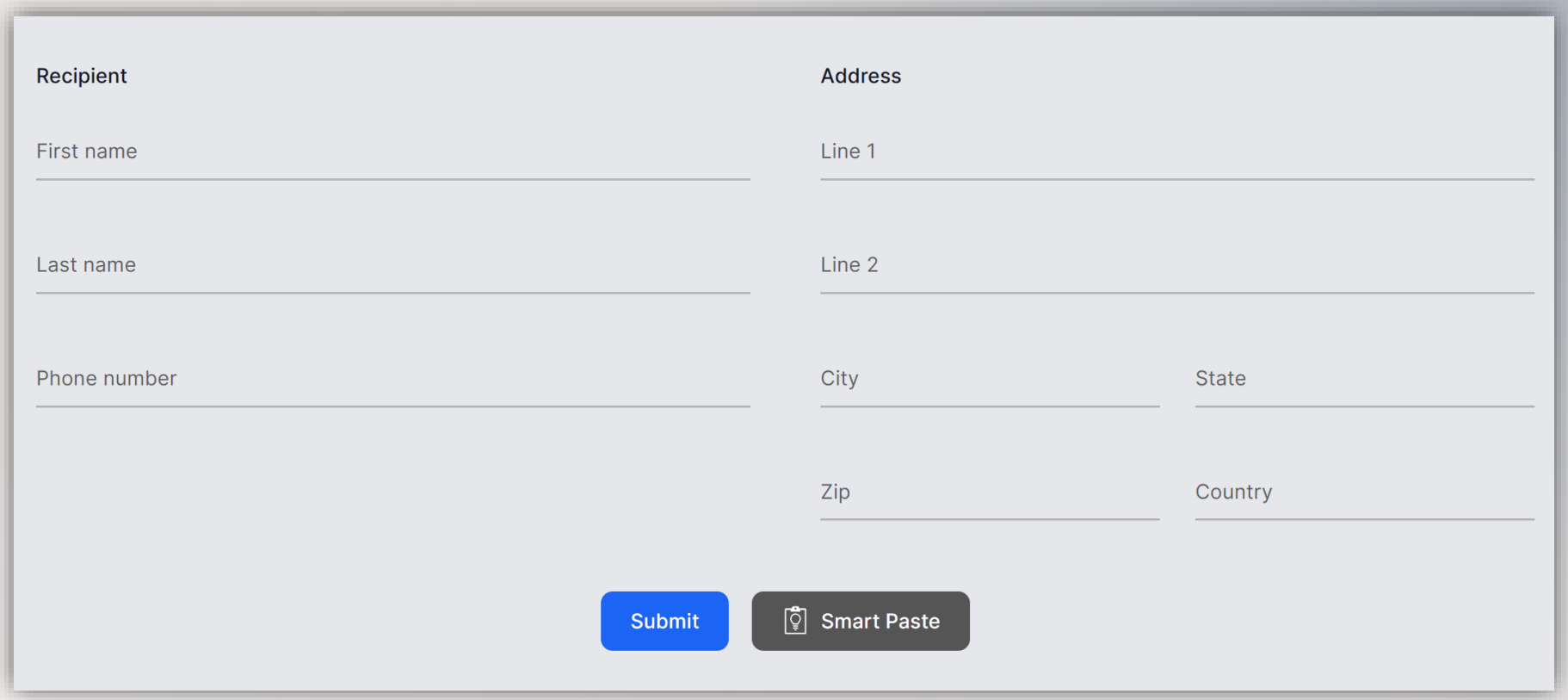
1. «Estrai le categorie da questo testo e riportale in una lista puntata: ....»

2. «Date queste categorie: "...", crea un CSV delle categorie per questo testo ...»

- Quest’ultimo è il modo più efficace di ottenere la classificazione.

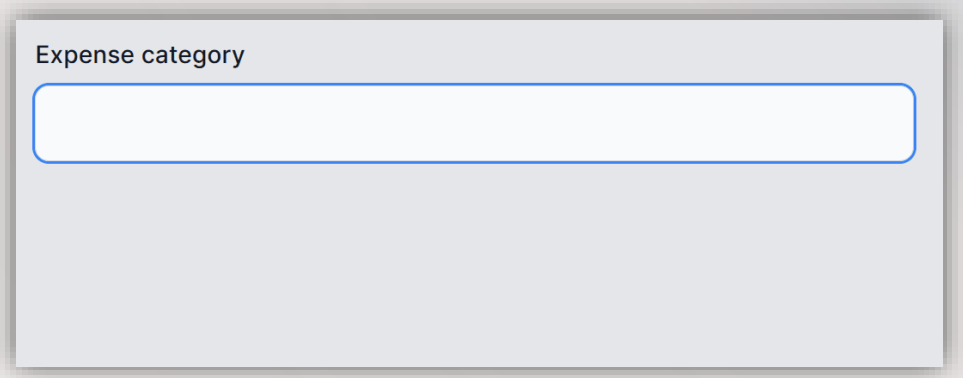
Identificazione dei dati (inserimento Form)

- Viene usato per suddividere il testo catturato con "Copy" nei campi presenti nella form



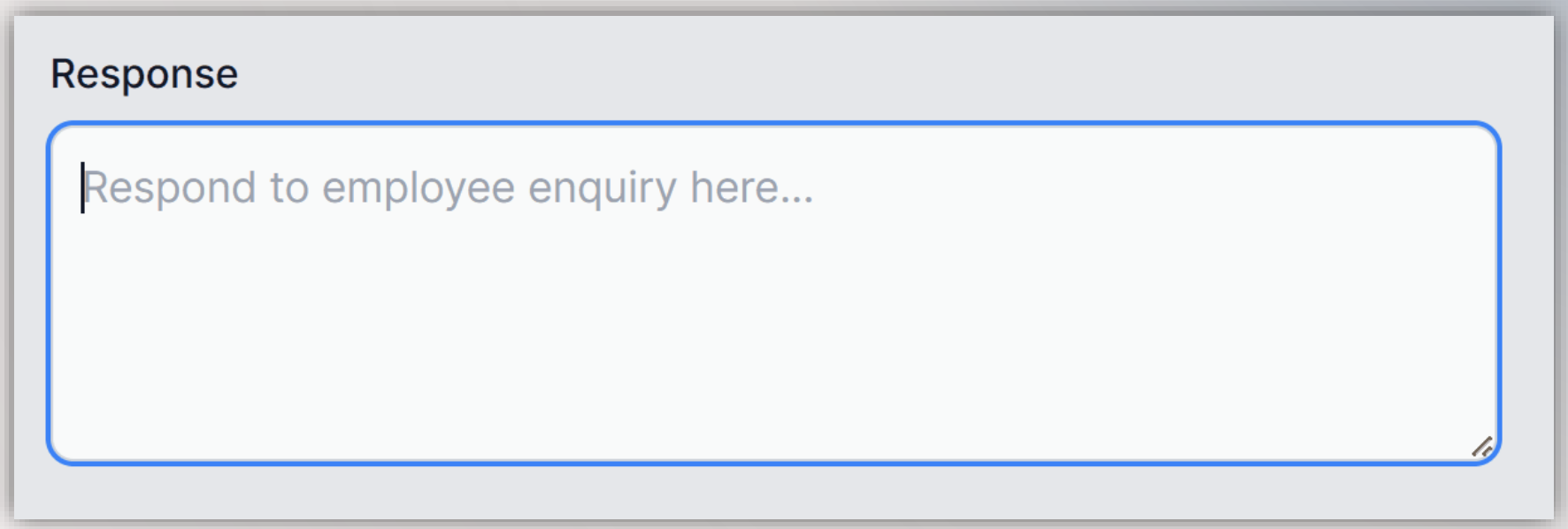
Ricerca Semantica

- Si usano gli Embedding per misurare la distanza tra parole o frasi
- Esempi:
  - Pagina di inserimento delle spese sul sito aziendale
  - Pagina che richiede la categoria della spesa
- Esempio di controllo per cercare le voci di una categoria:
  1. Precalcolare gli Embedding delle parole
  2. Calcolare l'Embedding della parola e cercarla per similarità



Completamento automatico

- Consiste nell'aiutare l'utente a completare il testo che sta scrivendo.
- Email, pagine di feedback, etc.



Raccomandazioni

- Il caso tipico è quello dell'E-commerce
- Preparazione:
  - Ogni descrizione del prodotto viene trasformata in Embedding
  - Ogni utente che acquista un prodotto accumula i relativi Embedding nel profilo
- Fase operativa:
  - Un nuovo utente visita la pagina di un prodotto
  - L'Embedding della descrizione viene utilizzata per identificare gli utenti che hanno acquistato prodotti simili (non necessariamente lo stesso)
  - L'applicazione raccomanda gli altri prodotti acquistati da quell'utente.

Trasformazioni del testo

- «Crea un riassunto del testo che segue»
- «Traduci il seguente testo utilizzando uno stile professionale»
- «Riscrivi il seguente testo in modo che possa essere compreso da un bambino di 12 anni»
- «Riscrivi il seguente testo enfatizzando le caratteristiche positive»
- «Riscrivi il seguente testo usando uno stile piratesco»

Aiuto interattivo nelle applicazioni

- L'utente sta utilizzando una applicazione che ancora non conosce a fondo
  - Esempio: un editor di testo
- In una casella di testo l'utente scrive:
  - «voglio cambiare le impostazioni della lingua per questo paragrafo»
  - «apri la finestra per formattare il paragrafo selezionato»
  - «trasforma in tabella questa lista puntata, aggiungendo una colonna vuota in fondo»
- Accessibilità. Aiuto per gli ipovedenti o portatori di altri handicap
  - Gli LLM permettono di utilizzare l'applicazione chiedendo spiegazioni e parlando in modo del tutto naturale
  - Esempio: salva il documento corrente in PDF dentro la cartella Budget2024

Ottenere risposte deterministiche

Agenti

- Gli agenti (chiamati anche Tool o Funzioni) sono degli aiuti che possiamo fornire al modello LLM
- Il software comunica al modello queste funzionalità extra fin dall'inizio
  - Queste funzionalità sono descritte in linguaggio naturale
- Quando il modello identifica un contesto su cui l'agente è "ferrato", lo chiama
  - L'agente è un software che gira nell'ambiente dell'utente
- La risposta dell'agente verrà usata dal modello per produrre la risposta.

Agenti + RAG + Dati su database

- I database aziendali contengono pochi dati testuali e molti numeri
- La loro relazione segue uno "schema", non è testo sequenziale
- Come fare RAG su un Database?
  1. Il prompt chiede dati che il LLM 1 non conosce
     - Insieme al prompt offriamo al LLM 1 due Agenti
  2. LLM 1 prepara una domanda (testuale) per l'Agente 1
  3. L'Agente 1 usa un LLM 2 per generare una "Query SQL"
  4. LLM 1 manda la query SQL all'Agente 2
  5. L'Agente 2 esegue la "Query SQL" ed estrae i dati
  6. LLM 1 ha i dati (RAG) e può finalmente rispondere alla domanda

Esempi di Agenti

- Ottenere informazioni sul PC dell'utente
  - File, log, telemetria, stato di salute delle macchine, etc.
- Fornire dati successivi al training del modello (Bing)
  - Meteo odierno, news, azioni in borsa, etc.
- Eseguire la RAG senza mettere i dati privati sul Cloud
  - I modelli che generano Embedding sono disponibili anche offline
- Pilotare o leggere dati da dispositivi elettronici (IoT industriale/domotica)
  - Ogni tipo di automazione

Sviluppare un Copilot: Architettura e Librerie

https://github.com/raffaeler/llmstarter

Librerie .NET per costruire un Copilot

- HttpClient puro e semplice
  - Pro: Zero dipendenze e possibilità di fruire delle funzionalità più recenti
  - Con: Molto codice da scrivere
- Azure OpenAI SDK
  - Pro: Facile da usare
  - Con: Le API v1 e v2 non possono coesistere (La v2 usa OpenAI SDK internamente)
- OpenAI SDK
  - Pro: Facile da usare
  - Con: Limitata solo ai modelli di OpenAI
- Semantic Kernel
  - Pro: Supporta tanti providers (Azure, OpenAI, HuggingFace, Ollama e altri)
  - Con: Resta sempre indietro sulle funzionalità più recenti
  - Con: Le astrazioni possono fare confusione a chi conosce già le API "native" dei provider

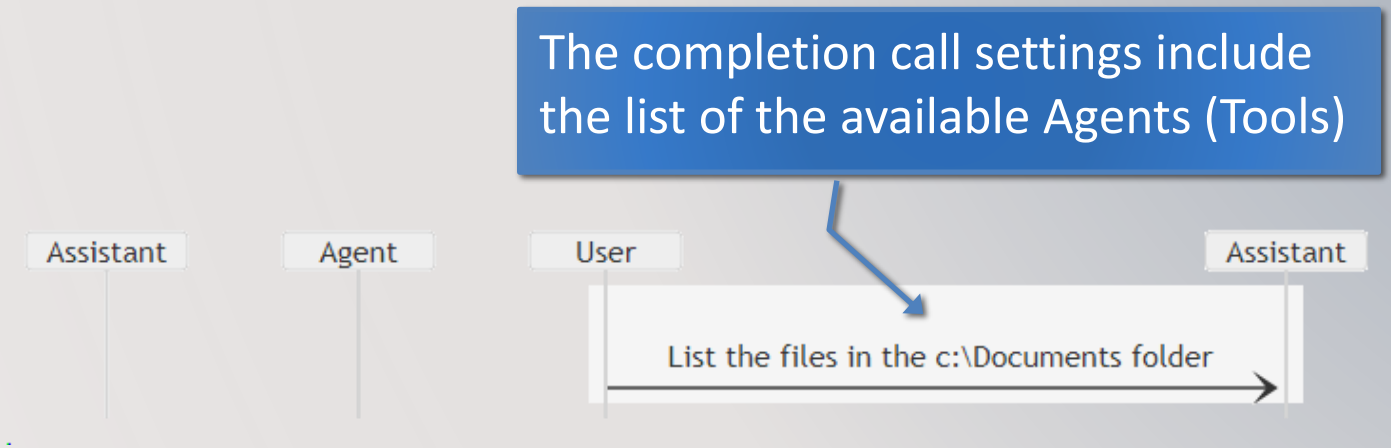
Stateless vs Stateful

- I modelli LLM sono stateless (ottimo per la scalabilità)
  - Il prompt determina lo stato iniziale della rete neurale
  - Il modello produce un "completamento" basato su quello stato
- Questa architettura presenta alcune sfide:
  - Ogni volta dobbiamo sempre rimandare l'intera conversazione
  - Questo implica maggiore latenza dovuta al trasferimento di rete
- Nella v2, OpenAI ha introdotto la Assistants APIs con oggetti Stateful:
  - Threads: rappresentano lo stato persistito cioè la memoria della conversazione
    - Il numero di token (e i costi) non cambiano
  - Assistants: possono usufruire di tool locali (lato server) come file search, code interpreter o user functions
    - Sono usabili a prescindere dagli Assistants.

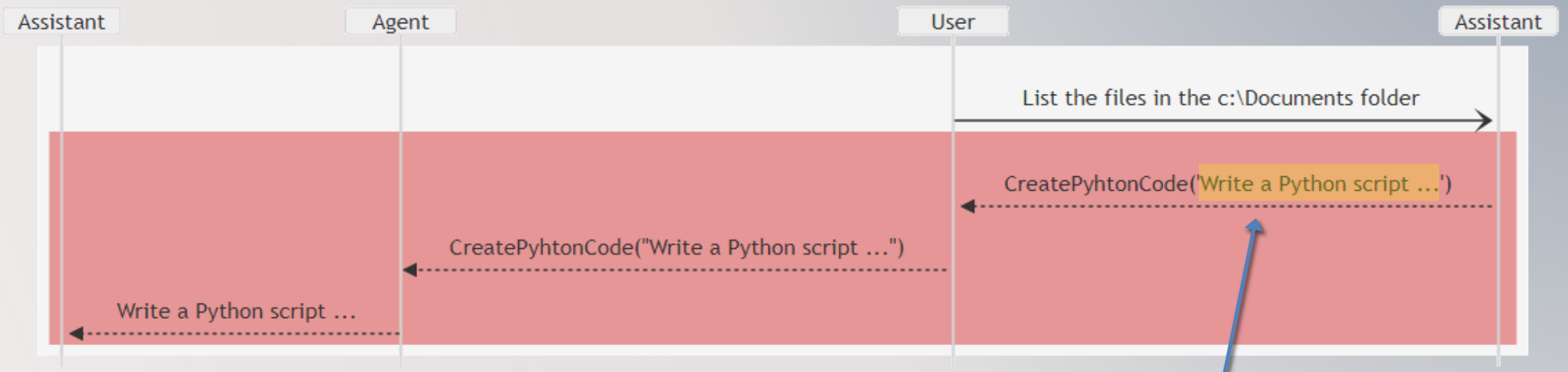
Funzioni / Tools / Agenti

- L'invocazione locale di codice è una callback nel mezzo di una completion
  - Accade tra la domanda (user prompt) e la risposta (Assistant Prompt)
- L'esistenza dei Tool è dichiarata nella richiesta iniziale
  - I parametri sono descritti in uno schema JSON
  - Semantic Kernel astrae questi concetti usando gli attributi .NET
- È vitale descrivere bene la funzione e i suoi parametri
  - Il nome del tool viene usato per spiegare la funzionalità del tool al modello
- Le Eccezioni (messaggio) sono usate dal modello per riprovare o cambiare strategia.
- Gli agenti usano i Prompt su un (altro) modello per ottenere il loro scopo
  - Attenzione alle ricorsioni!

Agent call flow - Step 1

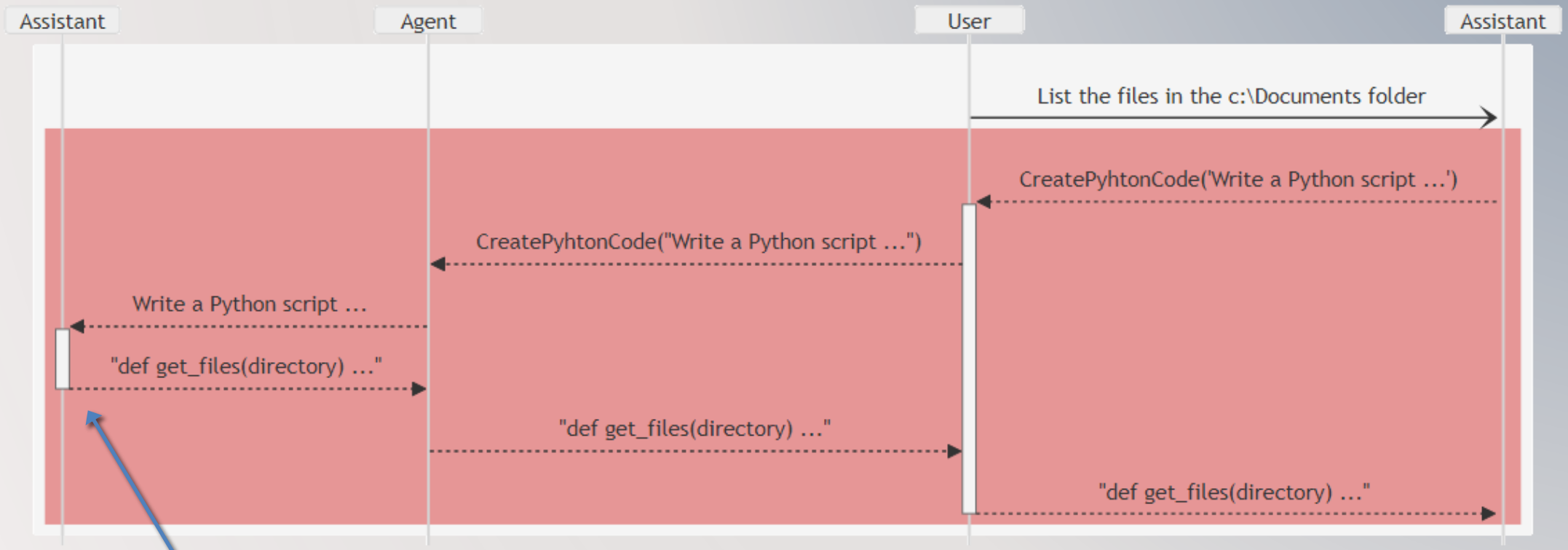


Agent call flow - Step 2



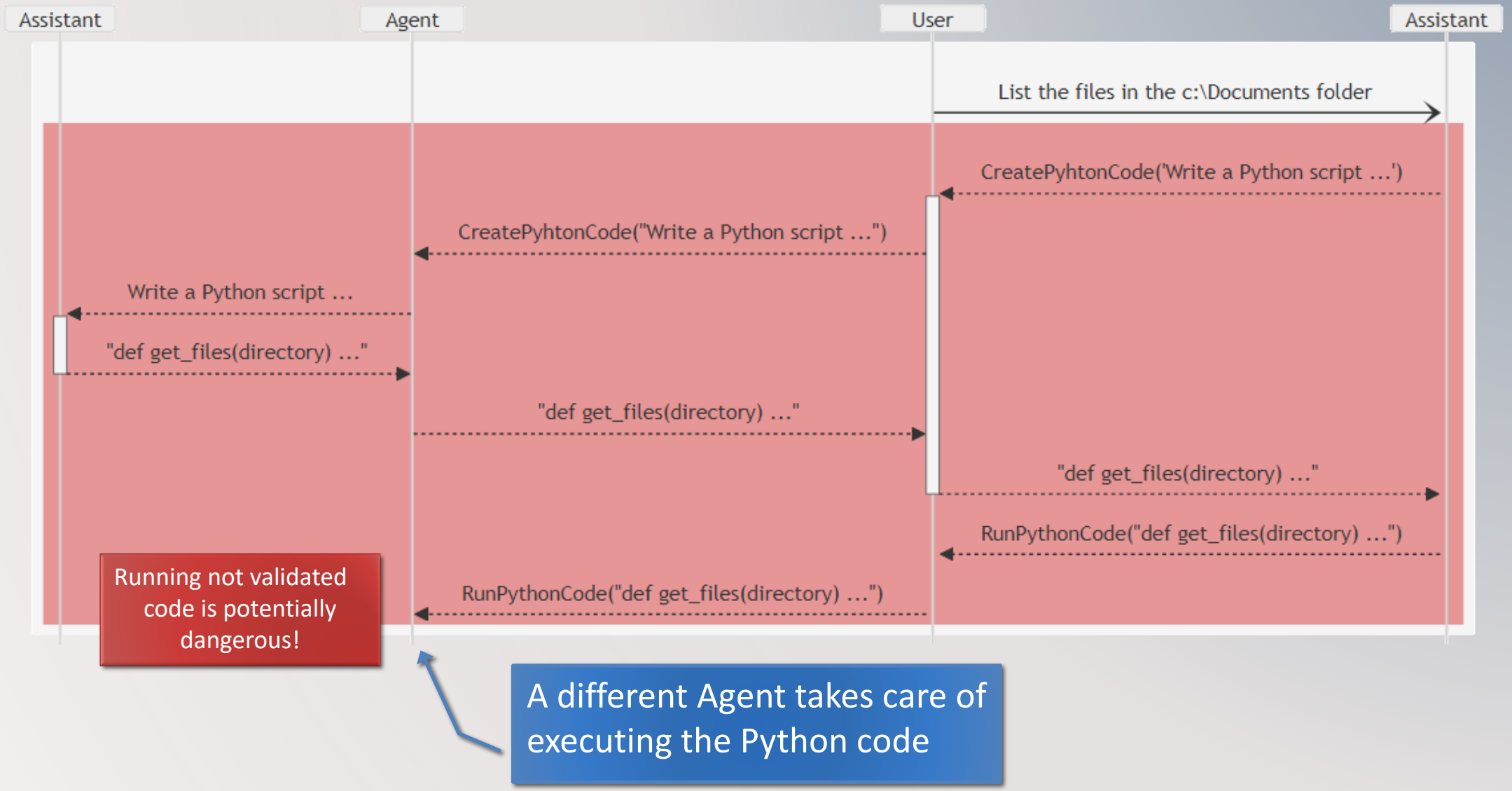
Prompt generated by the Assistant (LLM).
It contains the English textual description of the code to generate

Agent call flow - Step 3

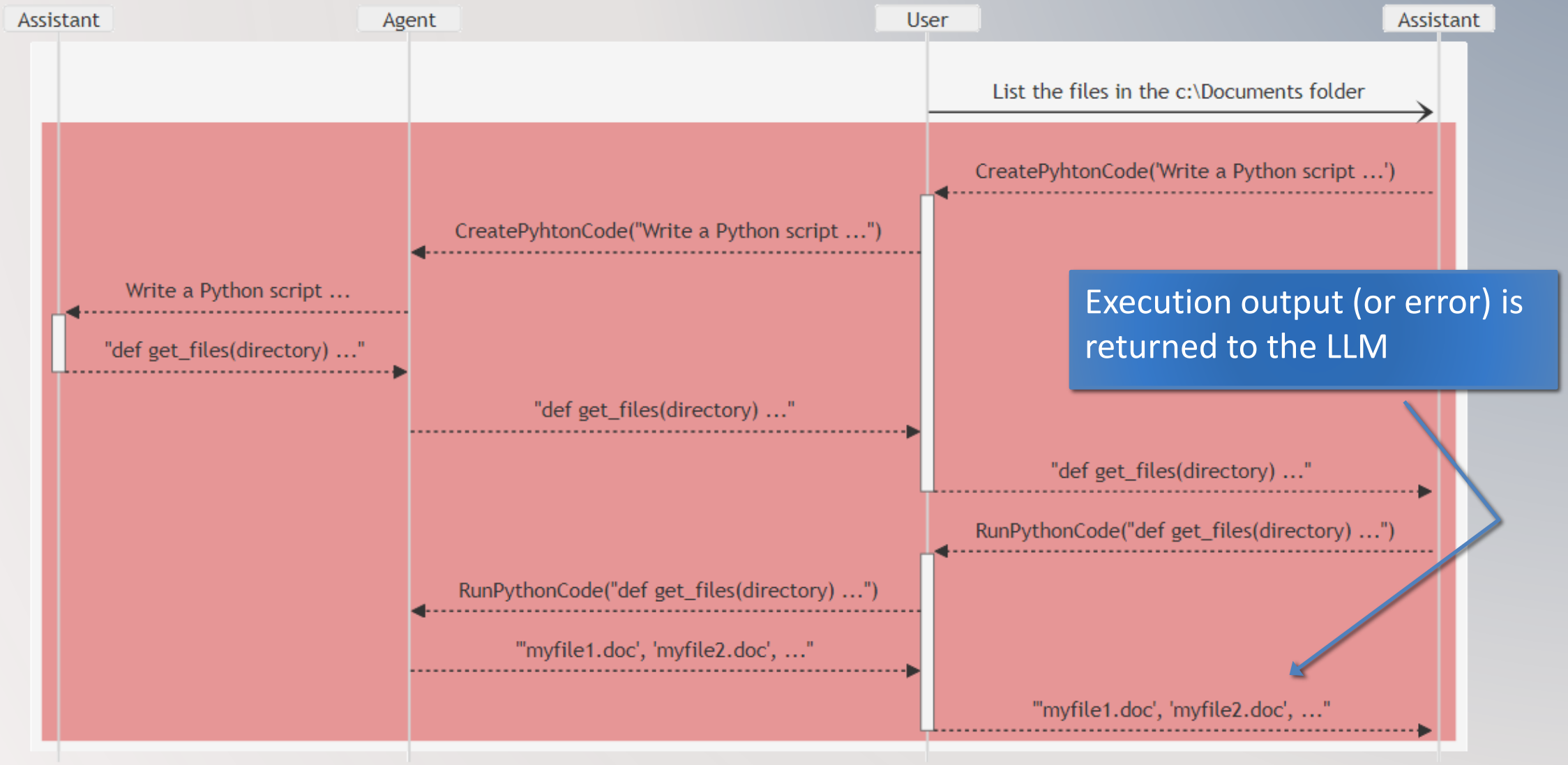


Another Assistant (LLM) generates the Python code based on the English description of the first assistant

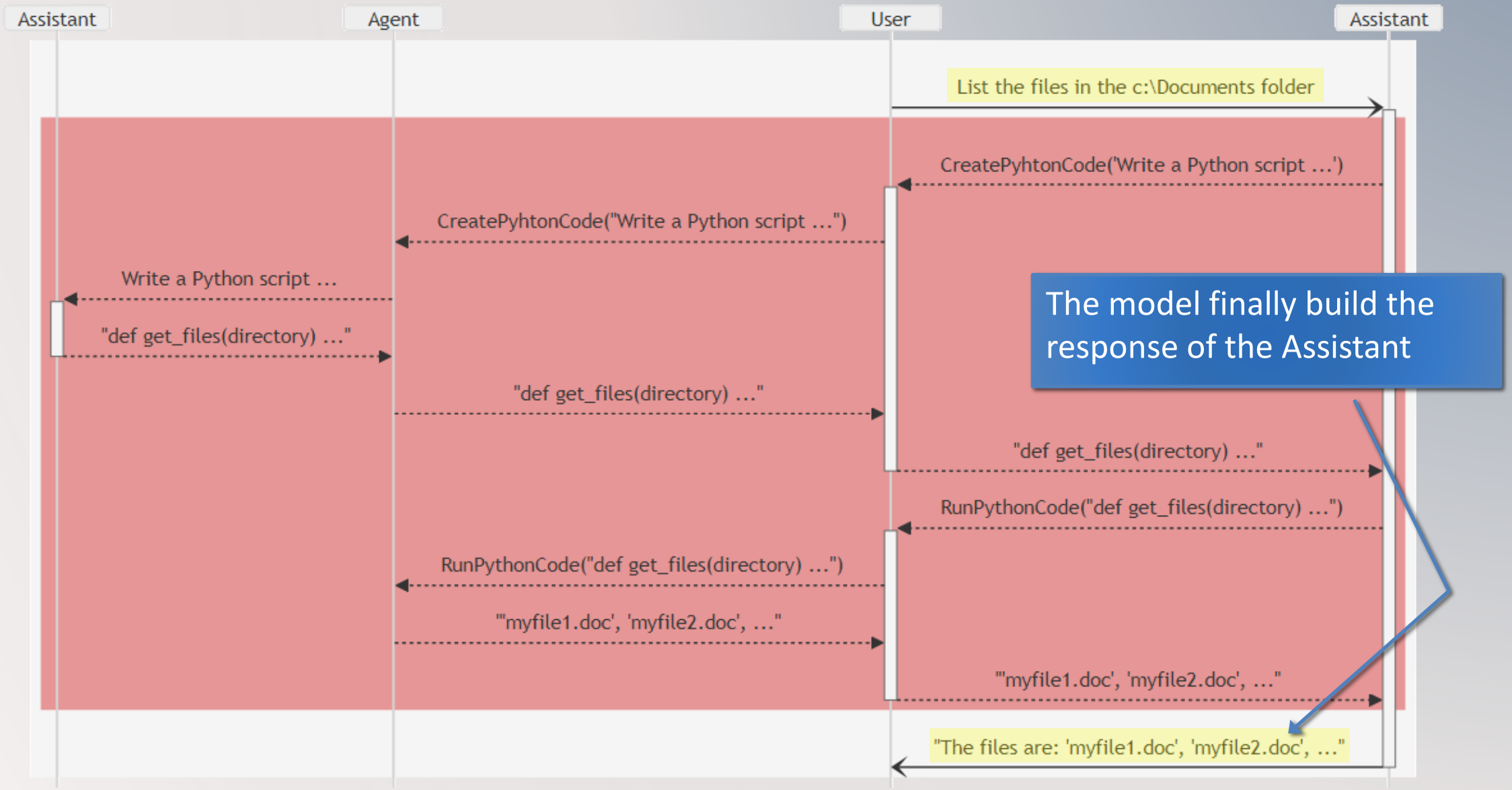
Agent call flow - Step 4



Agent call flow - Step 5



Agent call flow - Final completion



Copilot e Agenti all’opera

Per concludere

- Le similarità (Ricerca semantica) è un ottimo punto di partenza
- Non costruite un CoPilot generico. La genericità lo rende inutile!
- I Copilot che usano agenti:
  - Sono affidabili e possono fornire dati deterministici quando richiesto.
  - Rendono il Copilot più semplice da testare quando arrivano nuovi modelli
- Ricordate di strutturare Prompt di qualità.

Long questions?
Let's meet 1: 1 later!

Thank you!

@raffaeler
raffaeler@vevy.com

Generated offline with AI using Stable Diffusion

Prompt:

create a picture of a crowd of happy smiling people in a room cheering at the speaker

.

Long questions?
Let's meet 1: 1 later!

Thank you!

@raffaeler
raffaeler@vevy.com

Generated offline with AI using Stable Diffusion

Prompt:

create a picture of a crowd of happy smiling people in a room cheering at the speaker

Long questions?
Let's meet 1: 1 later!

Thank you!

@raffaeler
raffaeler@vevy.com

Generated offline with AI using Stable Diffusion

Prompt:

create a picture of a crowd of happy smiling people in a room cheering at the speaker

Long questions?
Let's meet 1: 1 later!

Thank you!

@raffaeler
raffaeler@vevy.com

Generated offline with AI using Stable Diffusion

Prompt:

create a picture of a crowd of happy smiling people in a room cheering at the speaker.

https://tinyurl.com/dotnetliguria

Domande e feedback

Modulo di Feedback

https://tinyurl.com/dotnetliguria