
Agenti AI: guida per auditor e management
Paolino Madotto
Intelligentiae s.r.l.
paolino.madotto@intelligentiae.it
intelligentiae
Data enabling Business
Intelligentiae s.r.l.
www.intelligentiae.it
info@intelligentiae.it
ROMA(RM) - ITALY
Paolino Madotto
Chief Executive Officer & Founder
paolino.madotto@intelligentiae.it
+393287243271

intelligentiae

Data enabling Business

Intelligentiae s.r.l. e Ambrogio

Soluzioni innovative di Intelligenza Artificiale

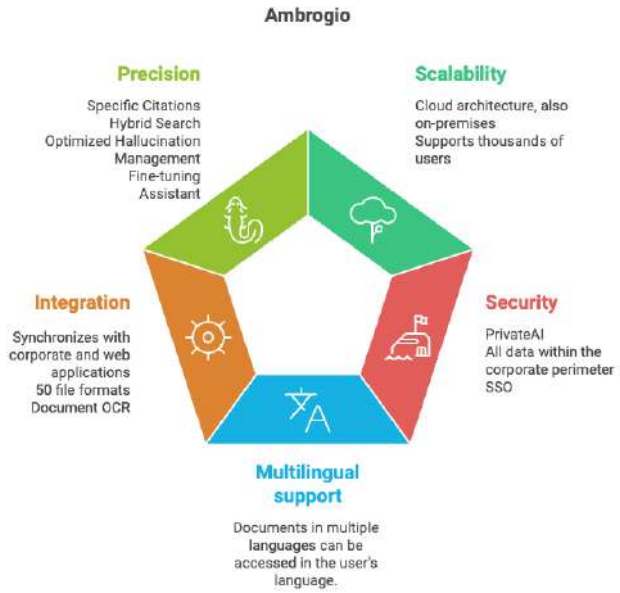
Intelligentiae s.r.l. offre soluzioni avanzate di Intelligenza Artificiale per migliorare l'efficienza delle aziende italiane.

Ambrogio: primo RAG italiano

Ambrogio è il primo sistema italiano di Retrieval-Augmented Generation, progettato per offrire assistenza affidabile e personalizzata in **PrivateAI**.
Scalabile e costruito sulle aziende.

Risposte affidabili e senza allucinazioni

Ambrogio si distingue per la riduzione delle allucinazioni, garantendo risposte sicure e precise agli utenti.



CONTENTS

01

Cos'è un agente AI

Cos'è un Agente AI?

Un agente AI è un sistema software che percepisce l'ambiente, elabora obiettivi e agisce autonomamente per raggiungerli.

Autonomia

Agisce senza intervento umano, prendendo decisioni in base ad obiettivi predefiniti.

Reattività

Risponde in tempo reale a cambiamenti nell'ambiente, adattando il suo comportamento.

Pro-attività

Inizia azioni per raggiungere obiettivi futuri.

Apprendimento

Migliora le prestazioni attraverso l'esperienza.

Architettura Logica di un Agente AI

Il flusso informativo è ciclico, con feedback continui che permettono all'agente di adattarsi e apprendere.

Tipologie di Agenti AI

Reattivi

Reagiscono a stimoli senza mantenere uno stato interno complesso.

Esempio: Chatbot bancari di base per risposte a FAQ.

Deliberativi

Pianificano sequenze di azioni per raggiungere obiettivi.

Esempio: Agenti di trading algoritmico che eseguono strategie.

Ibridi

Combinano reattività e pianificazione per equilibrare efficienza e complessità.

Esempio: Robot di Process Automation Cognitiva (RPA).

Multi-Agente

Sistemi di agenti che collaborano o competono per risolvere problemi.

Esempio: Piattaforme di help-desk con assistenti specializzati.

02

Tecniche base di implementazione

Tecniche di Prompt Engineering

Zero-Shot
L'LLM risponde senza esempi, basandosi solo sulle sue conoscenze.

Few-Shot
Fornisce alcuni esempi di input-output per guidare l'LLM.

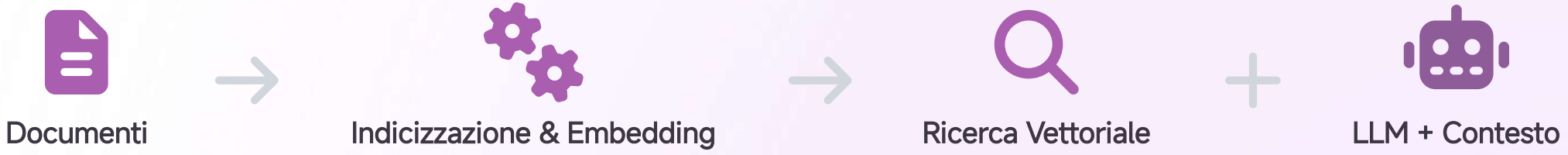
Chain-of-Thought (CoT)
Chiede all'LLM di mostrare il suo ragionamento passo-passo.

Esempio: Estrazione Dati da Policy

"Estrai dal documento: 1) Massimale prestito, 2) Tasso di interesse, 3) Requisiti età. Mostra il risultato in JSON. Se un dato non è presente, scrivi 'N/D'."

Retrieval-Augmented Generation (RAG)

Colma i limiti di memoria dell'LLM integrando una knowledge base aziendale, riducendo le "allucinazioni".



Vantaggio: Risposte accurate su regolamentazioni interne, prodotti, procedure.

Tool Use e Function Calling

L'agente non genera solo testo, ma può interagire con il mondo esterno chiamando funzioni.

Esempio: Agente chiama servizio scoring creditizio e aggiorna CRM in tempo reale.

Ciclo Autonomo: Percepisci-Pianifica-Agisci

Gli agenti operano in un loop continuo che permette loro di agire e adattarsi in modo autonomo.

Percezione: Raccoglie dati dall'ambiente.

Pianificazione: Elabora strategie per raggiungere gli obiettivi.

Azione: Esegue le azioni previste.

Rivalutazione: Osserva i risultati e aggiorna la strategia.

⚠ Rischio: Possibili loop infiniti se non gestiti da "guard rails".

INSERTO SPECIALE

Agentic in 5 mosse: da prompt a JSON

Paolino Madotto

Intelligentiae s.r.l.

paolino.madotto@intelligentiae.it

CONTENTS

INSERTO SPECIALE

INSERTO SPECIALE

A

Dall’idea al JSON

INSERTO SPECIALE

1. Prompt base e risposta JSON

Il contratto di comunicazione con l'AI

Il Prompt

Chiediamo esplicitamente un output strutturato:

```
"Restituisci solo JSON con
campo 'temperatura'."
```

La Risposta e il Parsing

Il modello restituisce un oggetto JSON, che il codice Python carica e utilizza:

```
# Risposta dell'AI
{"temperatura": 18}

# Codice Python per estrarre il dato
data = json.loads(response)
temp = data["temperatura"] # 18
```

2. Parsing Sicuro e Validazione

INSERTO SPECIALE

Non basta parsare, bisogna validare. Usiamo uno schema JSON per garantire l'integrità dei dati.

1. Definisci Schema

Specifica la struttura e i tipi di dati attesi.

→

2. Valida i Dati

Usa `jsonschema.validate()` per confrontare JSON e schema.

→

3. Gestisci Errori

Se la validazione fallisce, il codice può gestire l'eccezione.

```
try:
    validate(instance=data, schema=schema)
exceptValidationErroras e:
    print(f"Errore: {e}")  # Dato mancante o tipo errato
```

INSERTO SPECIALE

B

Chiamata a funzione tool

INSERTO SPECIALE

3. Descrivere Funzioni come Strumenti (Tools)

La Funzione Python

Una funzione normale, con type hints per chiarezza.

```python
def accendi_luce(stanza: str) -> bool:
    ... # Logica per accendere la luce
    return     True
```

La Descrizione "Tool"

Un dizionario che descrive la funzione all'AI.

```
{"type": "function",
  "function": {
    "name": "accendi_luce",
    "description": "Accende la luce",
    "parameters": { ... }
  }}
```

L'AI usa questa descrizione per capire quando e come chiamare la funzione, restituendo un JSON con `name` e `arguments`.

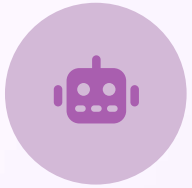
4. Il Ciclo di Invocazione e Callback

INSERTO SPECIALE

Da una richiesta in linguaggio naturale a un'azione eseguita.

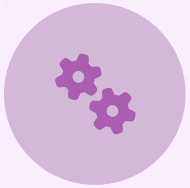
1. Richiesta Utente

"Accendi la luce in cucina"



2. Decisione AI

Restituisce un tool call JSON



3. Esecuzione Codice

La funzione `accendi_luce` viene chiamata

4. Risposta Finale

L'AI genera la frase di conferma

```
# Esempio di codice per l'esecuzione
if tool_call:
  result = func_map[tool_call.name](**json.loads(tool_call.arguments))
  messages.append({"role": "function", "content": str(result)})
```

INSERTO SPECIALE

C

Esempio end-to-end

5. Un Micro-Agente Completo

INSERTO SPECIALE

Unisce tutti i concetti in un flusso di lavoro autonomo.

```
# 1. Definizione dei Tools (Funzioni)
defaccendi_luce(stanza: str) -> bool: ...
defrileva_tempo() -> str: ...
# 2. Descrizione dei Tools per l'AI
tools = [
  {"type": "function", "function": {"name": "accendi_luce", ...}},
  {"type": "function", "function": {"name": "rileva_tempo", ...}}
]
# 3. Mappa delle funzioni
func_map = {"accendi_luce": accendi_luce, "rileva_tempo": rileva_tempo}
# 4. Ciclo di conversazione e esecuzione
messages = [{"role": "user", "content": "Accendi la luce e dimmi che ore sono"}]
whileTrue:
  response = client.chat.completions.create(model="...", messages=messages, tools=tools)
  message = response.choices[0].message
  messages.append(message)
  if message.tool_calls:
    for tool_call in message.tool_calls:
      args = json.loads(tool_call.function.arguments)
      result = func_map[tool_call.function.name](**args)
      messages.append({"role": "tool", "content": str(result)})
  else:
    print(message.content) # Risposta finale!
    break
```

INSERTO SPECIALE

1. Elementi Chiave del Function Calling 🛠️

Il processo si basa su tre componenti principali:

A. La Descrizione della Funzione (Lo Schema)

Questo è il set di istruzioni che tu (il programmatore) fornisci all'LLM. Deve essere conciso, preciso e preferibilmente in formato JSON Schema.

Esempio di Descrizione (Schema) per una Funzione Meteo:

Componente	Descrizione
`name`	`ottieni_meteo_attuale`
`description`	Restituisce la temperatura attuale e una descrizione delle condizioni meteorologiche per una data città.
`parameters`	Oggetto JSON che descrive gli argomenti: - `type`: `object` - `properties`: - `citta`: `type: string, description: Il nome della città.` - `required`: `['citta']`

INSERTO SPECIALE

B. Il Prompt dell'Utente

Questa è la richiesta che l'utente pone, che spinge l'LLM a ragionare sulla necessità di usare lo strume

Esempio di Prompt dell'Utente:

Utente: "Che tempo fa oggi a Milano e qual è la capitale dell'Italia?»

C. La Risposta dell'LLM (La Chiamata)

A fronte del prompt e della descrizione fornita (A), l'LLM genera una richiesta di chiamata strutturata per l'azione specifica, ignorando la parte della domanda che può risolvere da solo (come "capitale dell'Italia").

Esempio di Risposta dell'LLM (Output della Chiamata):

```
{
  "tool_calls": [
    {
      "function": {
        "name": "ottieni_meteo_attuale",
        "arguments": {
          "citta": "Milano"
        }
      }
    }
  ]
}
```

INSERTO SPECIALE

2. Il Ciclo di Esecuzione e Risposta Finale 🔄

Dopo che l'LLM ha generato la chiamata (come mostrato nel punto 1.C), il tuo codice di orchestrazione esegue i seguenti passi:

- **Passo 1: Esecuzione dello Strumento**
  Il tuo codice Python intercetta l'output JSON, estrae il nome della funzione (ottieni_meteo_attuale) e gli argomenti (citta: Milano), ed esegue la funzione nel tuo sistema.

  Esempio di Risultato dell'Esecuzione (Funzione Python):
  "La temperatura attuale a Milano è di 12°C con cielo nuvoloso."

- **Passo 2: Invio del Risultato all'LLM**
  Il risultato del Passo 1 viene re-inserito nello storico della conversazione come un messaggio con ruolo "tool" o "function" e l'intera conversazione aggiornata viene inviata nuovamente all'LLM.

- **Passo 3: La Risposta Finale dell'LLM**
  L'LLM riceve il risultato della funzione e lo integra con la conoscenza che possiede per rispondere alla domanda completa.
  Esempio di Risposta Finale all'Utente:
  LLM: "La temperatura attuale a Milano è di 12°C con cielo nuvoloso. Inoltre, ti confermo che la capitale dell'Italia è Roma."

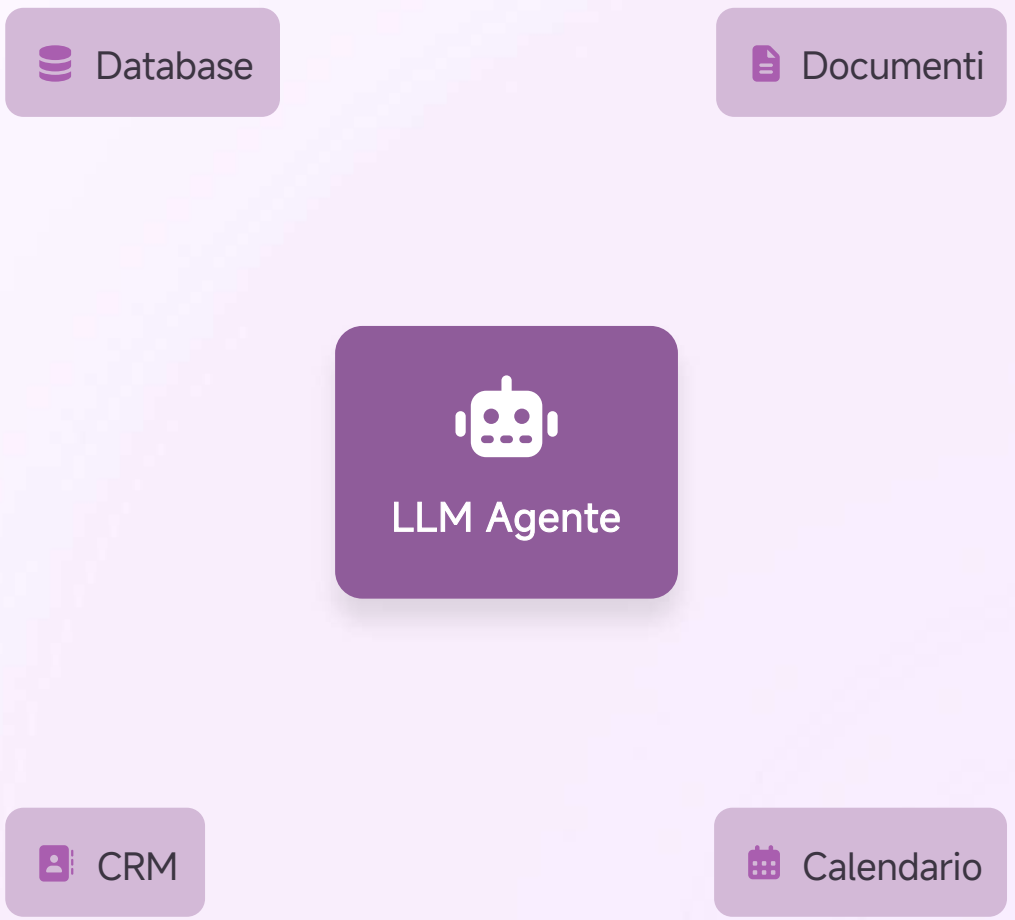
03

Protocolli MCP e A2A

Model Context Protocol (MCP)

Un standard aperto di Anthropic per collegare LLM a dati e strumenti esterni in tempo reale.

Supera i limiti della conoscenza statica del modello, abilitando azioni su CRM, database, documenti aziendali tramite un'interfaccia standardizzata.



Architettura e Primitivi di MCP

MCP Host

App come Claude Desktop o IDE

MCP Client & Server

Comunicazione tramite JSON-RPC

Client (Roots, Sampling)

Server (Prompts, Resources, Tools)

Strumenti Locali

Database, filesystem, servizi web

L'LLM, attraverso il client, può selezionare e invocare funzioni esposte dal server.

MCP è la USB dell'AI

Connettore universale

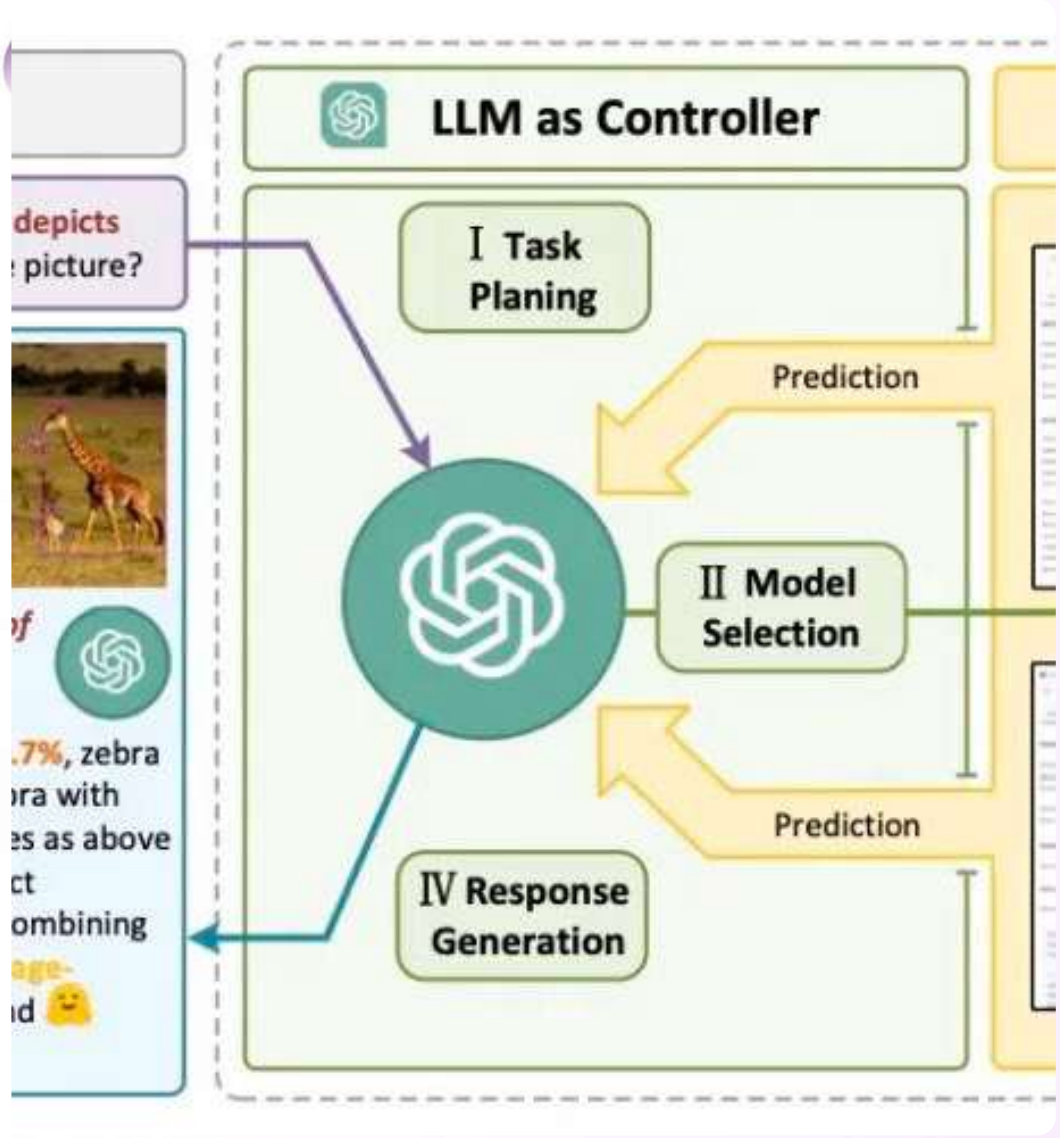
Model Context Protocol (MCP) funge da connettore universale per qualsiasi LLM, consentendo l'integrazione rapida e sicura con dati e strumenti aziendali, esattamente come una USB si collega a qualsiasi PC senza driver custom.

Trasferimento in tempo reale

MCP permette il trasferimento di dati e azioni in tempo reale, garantendo che l'AI possa interagire con il mondo esterno in modo immediato e dinamico, migliorando l'efficienza e la risposta alle richieste.

Plug-and-play sicuro

Grazie a MCP, gli agenti AI possono essere implementati in modo sicuro e tracciabile, con policy di accesso e rollback ben definite, garantendo un ambiente plug-and-play affidabile per qualsiasi sistema aziendale.



Agent-to-Agent Protocol (A2A)

Un standard di Google per la comunicazione tra agenti AI autonomi, indipendentemente dal vendor o dal framework.

Permette agli agenti di collaborare, delegare task e negoziare l'interazione senza condividere memoria o tool interni.

Funzionalità chiave:

- Capability Discovery (scoperta)
- Task Management (gestione)
- UX Negotiation (negoziazione)

Funzionamento di A2A: Agent Cards & Task

Esempio: Agente Marketing genera lead e delega a Agente Vendite tramite A2A per la chiusura.

MCP vs. A2A: Integrazione Verticale vs. Orizzontale

MCP

Agente → Strumento

Integrazione **verticale**. Connette un agente a tool e dati specifici, come un "driver" universale.

A2A

Agente → Agente

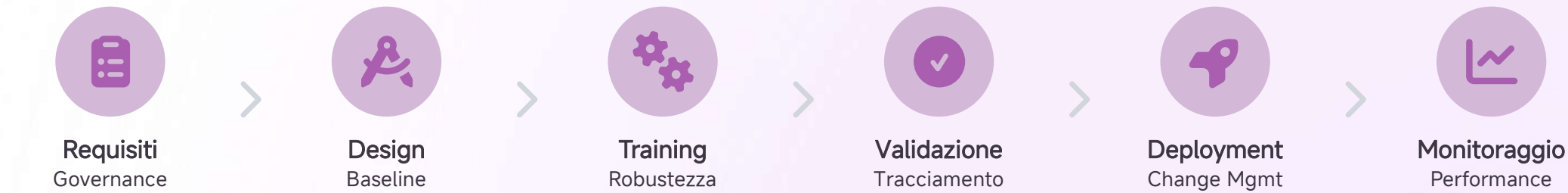
Integrazione **orizzontale**. Coordina più agenti specializzati, come un "protocollo email" per AI.

Nelle architetture complesse, i due protocolli **coesistono**, riducendo il codice custom e favorendo la scalabilità.

04

Fondamenti per auditor

Ciclo di Vita dell'Agente e Controlli Chiave



Metriche di Performance e Fairness

KPI di Performance

- Precision & Recall
- Latency & Throughput
- Tasso di completamento task

KPI di Fairness

- Fairness Score
- Disparate Impact
- Equal Opportunity

Queste metriche devono essere campionate e documentate per fornire evidenza agli audit.

Tracciabilità e Explainability

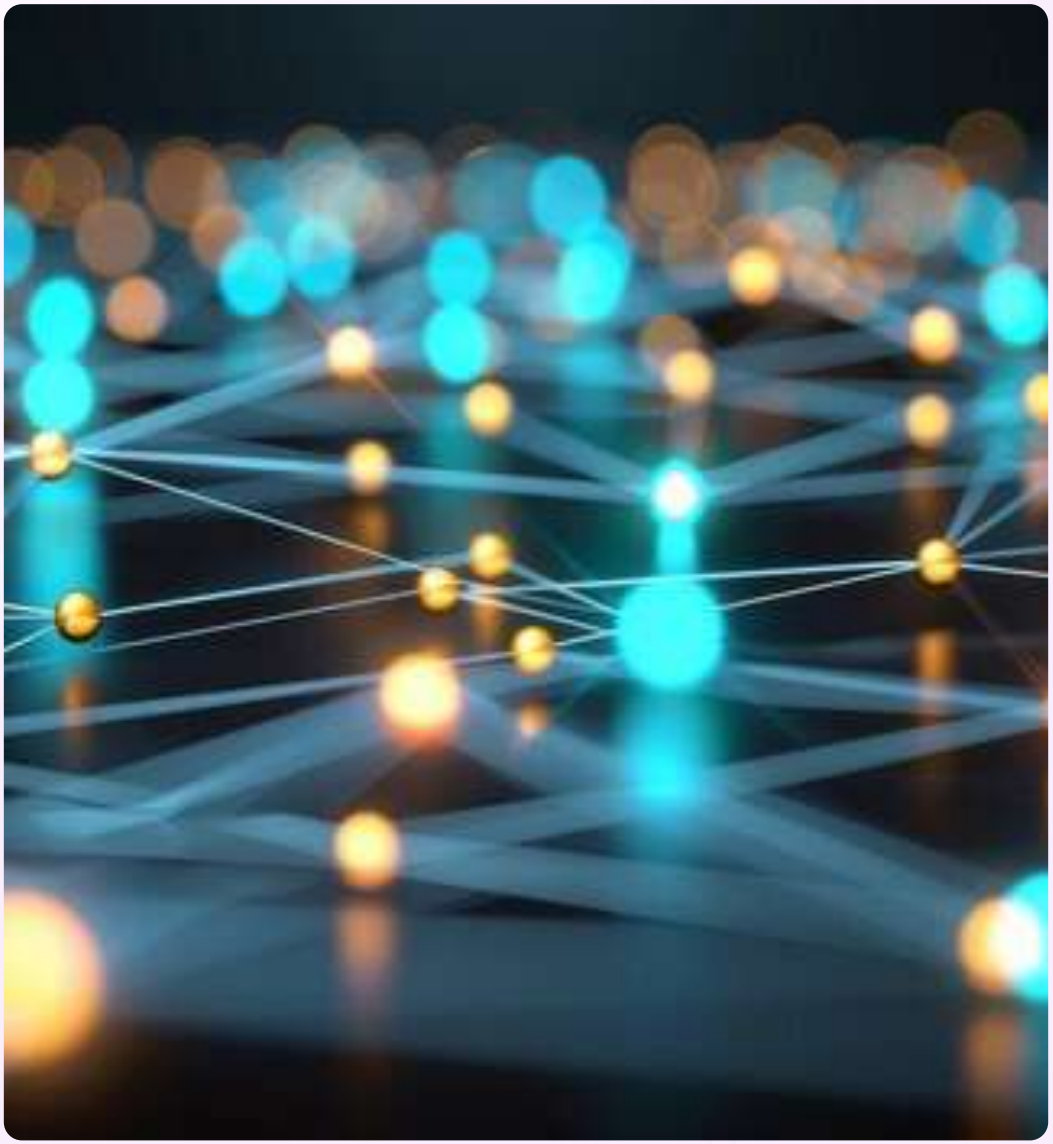
Per garantire la fiducia e la conformità, è cruciale poter ricostruire e spiegare le decisioni dell'agente.

Tracciabilità

Log immutabili (prompt, risposta, azioni), hash decisionali, versionamento modello.

Explainability (XAI)

Tecniche come LIME o SHAP per giustificare decisioni (es. scarto prestito).



Requisiti Normativi e Standard

GDPR (Art. 22)
Diritto alla non discriminazione automatizzata e profilazione.

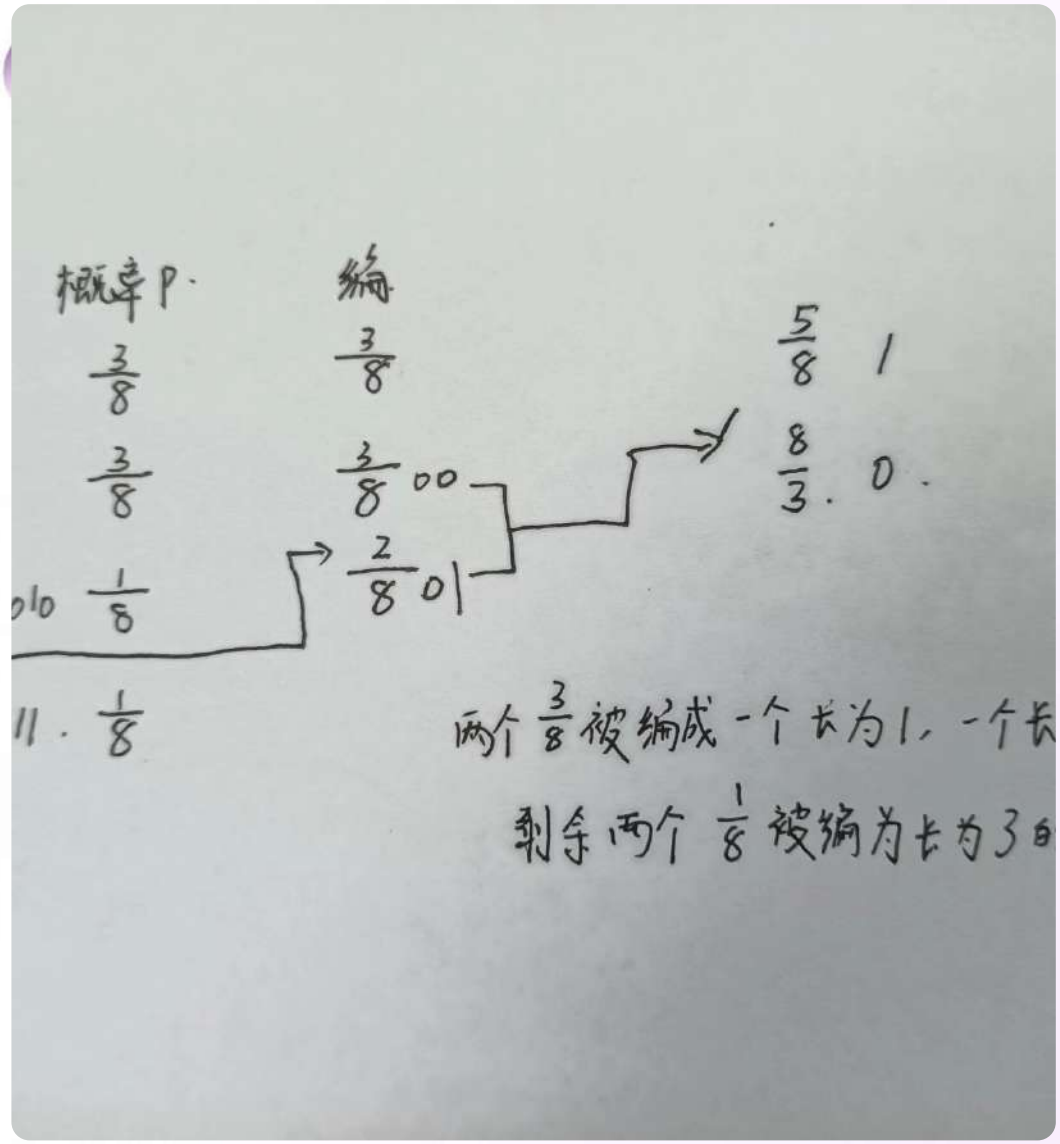
AI Act Europeo
Classificazione del rischio AI e relativi obblighi.

EBA Guidelines
Requisiti ICT, outsourcing e gestione del rischio.

ISO 23894
Gestione del rischio per sistemi di AI.

05

Rischi emergenti



Rischi: Allucinazioni e Data Poisoning

Allucinazioni

L'LLM genera risposte plausibili ma false, portando a decisioni errate (es. disposizione non corretta).

Data Poisoning

I dati di training vengono corrotti da fonti esterne, introducendo backdoor o bias. Necessita di robuste pipeline di data validation.

Rischi: Bias Discriminatori e Impatto Reputazionale

Bias nel Modello

Diniego ingiustificato di credito basato su genere, etnia, ecc.

Danno Reputazionale

Scandali, sanzioni AGCM, cause civili.

Mitigazione

Test di fairness periodici e panel di clienti.

Rischi: Cybersecurity e Prompt Injection

La superficie di attacco si espande con l'uso di agenti che interpretano input liberi.

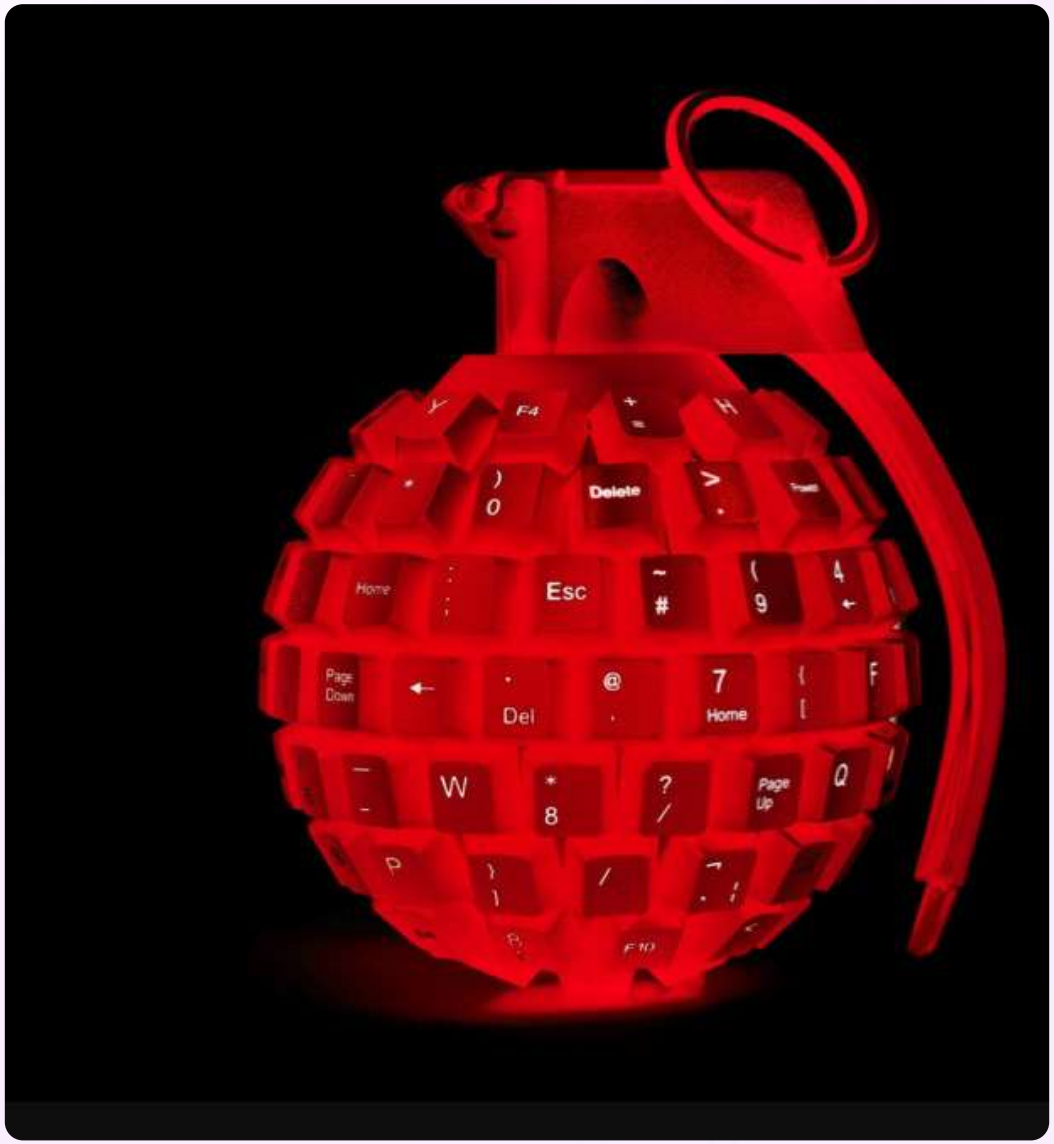
Prompt Injection

Input malevoli che alterano il comportamento dell'agente per estrarre dati o eseguire azioni non autorizzate.

Jailbreak

Tecniche per bypassare le policy etiche e di sicurezza dell'LLM.

Mitigazione: Sanitizzazione input, least privilege, red-team, anomaly detection.



Rischi: Dipendenza e Perdita di Competenze

L'over-reliance sull'AI può portare a un deterioramento delle capacità umane e a un debito tecnico.

Operatore Umano
Critical thinking

Over-Reliance sull'AI
Fiducia cieca, riduzione del controllo

Perdita Competenze
Technical debt

Mitigazione: Procedure manuali di fallback, training umano-centrico, log di override.

Rischi Emergenti: MCP e A2A

I nuovi protocolli introducono una superficie di attacco estesa che richiede controlli specifici.

Rischi MCP

Endpoint mal configurati possono esporre dati aziendali sensibili o permettere azioni non autorizzate.

Rischi A2A

Lo scambio di artifact tra agenti può trasferire informazioni riservate o dare eccessiva delega.

Mitigazione Comune

Autenticazione rigorosa (OAuth 2.0), cifratura flussi, audit di Agent Cards, controllo degli ambiti di delega.

06

Workflow agentici

Da PDF a piano lavoro

Ricezione del documento PDF

Il workflow inizia con la ricezione di un documento PDF. Un agente specializzato in OCR viene attivato per estrarre il testo dal documento. Questo processo è automatico e garantisce che i dati siano estratti in modo accurato e pronto per l'elaborazione successiva.

Elaborazione del testo

Dopo l'estrazione del testo, un agente NLP analizza il contenuto per classificare le informazioni e identificare le entità rilevanti. Questo passaggio è cruciale per garantire che i dati siano strutturati e pronti per l'inserimento nel gestionale aziendale.

Inserimento nel gestionale

L'agente connector si occupa di mappare i dati estratti e classificati nei campi appropriati del gestionale aziendale. Questo processo è eseguito in modo sicuro e tracciabile, garantendo l'integrità dei dati.

Creazione del piano di lavoro

Infine, l'agente planner genera il piano di lavoro e la documentazione amministrativa necessaria. Ogni passo del workflow è un micro-servizio autonomo che si coordina tramite messaggi asincroni, garantendo una tracciabilità completa e la possibilità di rollback in caso di errori.

Ruolo degli agenti specializzati

Agenti specializzati

Ogni agente ha un obiettivo specifico: l'agente OCR estrae testo con alta precisione, l'agente NLP classifica contenuti e identifica entità chiave, l'agente connector mappa campi su API del gestionale, e l'agente planner genera sequenze di task e documentazione amministrativa.

Comunicazione asincrona

Gli agenti si comunicano tramite un bus di messaggi asincroni. Questo approccio permette una grande flessibilità e riduce l'accoppiamento tra i componenti del sistema, facilitando l'integrazione e il mantenimento.

Integrazione e monitoraggio

Workflow engine

La workflow engine registra le capability di ogni agente e gestisce la fault-tolerance con retry e timeout. Questo garantisce che il processo sia robusto e affidabile, anche in caso di errori o interruzioni.

Monitoraggio e dashboard

Il sistema include un dashboard che mostra lo stato dei task, i KPI del tempo di ciclo e la percentuale di errori. Questo permette una visibilità completa sul processo e facilita il monitoraggio e l'ottimizzazione.

Transazioni compensative

Gli aggiornamenti del gestionale avvengono in transazioni compensative, garantendo la coerenza dei dati anche in caso di rollback. Questo approccio è essenziale per mantenere l'integrità dei dati aziendali.

07

Opportunità strategiche

Opportunità: Efficienza e Riduzione Costi

L'automazione intelligente porta a un risparmio significativo e a un rapido payback period.

- **KYC:** Riduzione del 70% del tempo di onboarding.
- **Back-office:** Abbattimento ticket di routine.
- **Scalabilità:** Operazioni notturne senza straordinario.

Payback Period: 8-12 mesi

Opportunità: Iper-Personalizzazione del Cliente

Analisi Spesa
Pattern e obiettivi di vita

Profilo Cliente
Propensione al rischio, fabbisogni

Prodotto Tailor-Made
Offerta personalizzata

Aumento Cross-Sell

Riduzione Churn

Miglior NPS

Necessità di consenso granular e profilazione lecita.

Opportunità: Innovazione di Prodotto

Gli agenti AI abilitano nuovi servizi a valore aggiunto, creando canali di revenue e differenziazione competitiva.

- **Piano di Risparmio Dinamico:** Collegato a eventi macro.
- **Assistente Fiscale in Tempo Reale:** Consigli personalizzati.
- **Concierge Finanziario 24/7:** Servizio premium fee-based.

Opportunità: Decisioni Data-Driven in Tempo Reale

Gli agenti integrano flussi di dati eterogenei per ottimizzare le decisioni in millisecondi.

Vantaggio in **risk management** e capitale regolamentare, con audit trail continuo.

Opportunità Strategiche: MCP e A2A

L'adozione di standard aperti accelera la trasformazione digitale e crea nuovi modelli di business.

Accelerazione Time-to-Market

Librerie MCP pronte riducono l'integrazione (es. CRM) da mesi a giorni. A2A consente un ecosistema di agenti specializzati senza lock-in vendor.

Nuovi Modelli di Business

Creazione di un marketplace interno di capacità AI monetizzabili. Possibilità di offrire servizi AI come nuovo canale di revenue.

GRAZIE 🙏


intelligentiae
Data enabling Business
Intelligentiae s.r.l.
www.intelligentiae.it
info@intelligentiae.it
ROMA(RM) - ITALY
Paolino Madotto
Chief Executive Officer & Founder
paolino.madotto@intelligentiae.it
+393287243271