UNA SERIE SULL'INTELLIGENZA ARTIFICIALE

Io, Gemini e altri quattro

Parte VI

Gli Agenti IA: Quando l'Intelligenza Artificiale Agisce

ROCCO MASSIMILIANO MONDO

ABSTRACT

Nei cinque episodi precedenti abbiamo capito come funziona l'IA, come comunicare con essa e quali modelli LLM scegliere per le nostre esigenze. Ma fino ad ora abbiamo sempre parlato di sistemi reattivi: tu scrivi, l'IA risponde. In questo sesto episodio compiamo un salto qualitativo fondamentale: dall'IA che risponde all'IA che agisce (salteremo dall'intelligenza generativa pura a quella agentica). Gli Agenti IA sono sistemi in grado di pianificare una sequenza di azioni, usare strumenti esterni (ricerca web, API, scrittura di file, esecuzione di codice), interpretare i risultati e adattare il piano in corso d'opera — il tutto con supervisione umana minima. Vedremo come funzionano, quali protocolli li governano, i casi d'uso più concreti e i rischi reali di una tecnologia che, se usata con consapevolezza, può moltiplicare la produttività professionale in modo sorprendente.

Rocco M. Mondo, Giugno 2026

PARTE VI

Gli Agenti IA: Quando l'Intelligenza Artificiale Agisce

IL SALTO CONCETTUALE: DA REATTIVO AD AGENTICO

Immaginate di chiedere ad un assistente umano: «Analizza le bollette dell'energia elettrica degli ultimi sei mesi, calcola la spesa media mensile per fascia oraria, identifica eventuali anomalie rispetto ai consumi storici e preparami un report Excel con grafici.» Un bravo assistente non vi risponde con domande: pianifica, raccoglie i documenti, esegue i calcoli, costruisce il file e ve lo consegna.

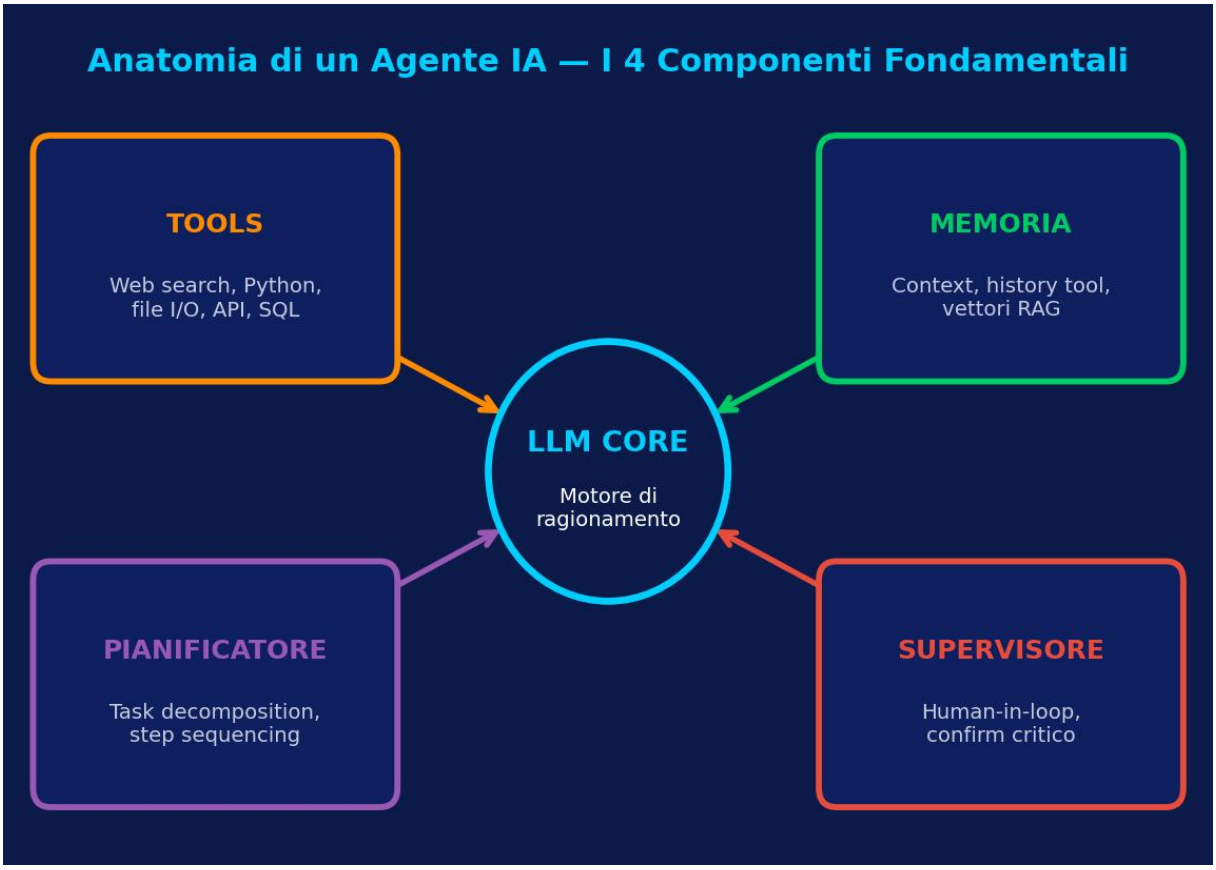
Fino a poco tempo fa un LLM non era in grado di fare questo. Poteva rispondere ad una domanda, suggerire un approccio, ma non eseguire autonomamente una sequenza di azioni nel mondo reale. Con gli ***agenti IA*** questo confine si è dissolto.

Un agente IA è un sistema che combina un LLM con la capacità di usare strumenti (tools use) in un ciclo autonomo di ***pianificazione***, ***esecuzione*** e ***valutazione***. È la differenza tra un esperto che vi dà consigli e un collaboratore che esegue il lavoro.

La svolta è avvenuta in modo rapido: dai primi esperimenti con GPT-4 ***Function Calling*** (2023) alle architetture agentiche native integrate in Claude 3.7, GPT-4o e Gemini 2.0 (2025), il tempo di maturazione tecnologica è stato straordinariamente breve. Oggi gli agenti sono disponibili sia come servizi cloud che come sistemi completamente locali, ***orchestrabili*** da chiunque abbia competenze di programmazione di base.

ANATOMIA DI UN AGENTE IA

Ogni agente IA, indipendentemente dalla piattaforma o dal framework usati, è composto da ***quattro elementi fondamentali*** che lavorano in ciclo continuo. La comprensione di questi componenti è essenziale per progettare ***flussi agentici*** efficaci e per diagnosticare i problemi quando qualcosa va storto. Sotto viene riportato uno schema grafico esemplificativo.



I quattro componenti fondamentali di un Agente IA e le loro interazioni

1. *Il Cervello (LLM)*

Il modello linguistico resta il componente centrale, il motore di ragionamento. Riceve il compito, interpreta il contesto, decide quale strumento usare e valuta i risultati intermedi. Non è solo un generatore di testo: ***in modalità agentica diventa un pianificatore***. La qualità dell'agente dipende in larga misura dalla capacità di ragionamento del modello sottostante. ***Modelli con Extended Thinking*** (come Claude 3.7 in modalità ***Sonnet Extended***) mostrano una significativa riduzione degli errori nelle catene di azioni lunghe.

2. *Gli Strumenti (Tools)*

Sono le capacità operative dell'agente. Ogni strumento è una funzione che l'LLM può invocare quando necessario. Il meccanismo tecnico (chiamato ***Function Calling o Tool Use***) è standardizzato: l'LLM restituisce un file con struttura ***JSON*** con il nome della funzione e i

parametri, il ***runtime*** la esegue e restituisce il risultato nel contesto. Gli strumenti più comuni includono:

- ***Web search***: interrogazione di motori di ricerca in tempo reale
- **Code interpreter**: esecuzione di codice Python in sandbox isolata
- ***File I/O***: lettura e scrittura di file (PDF, Excel, Word, JSON, CSV)
- ***HTTP/API calls***: integrazione con qualsiasi servizio REST
- ***Database queries***: SQL su SQLite, PostgreSQL, MySQL
- ***Email e calendar***: invio/lettura email, creazione eventi (tramite MCP)

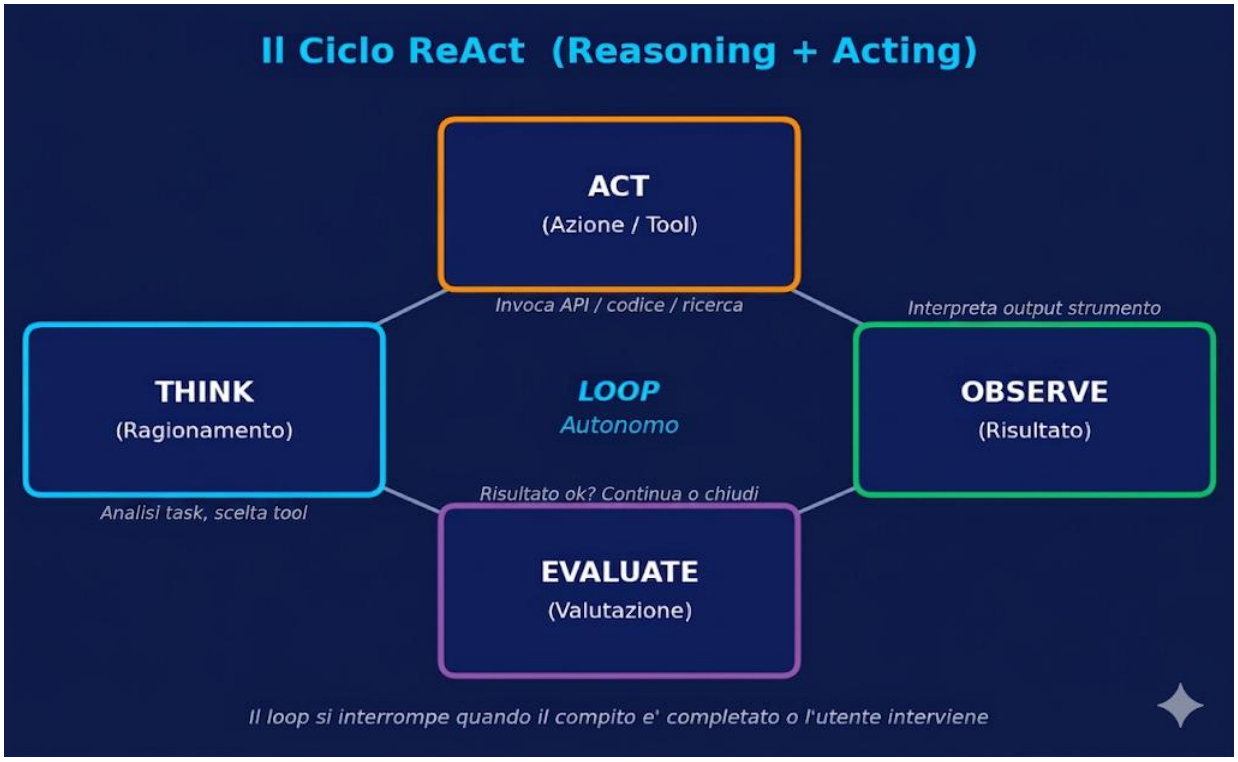
3. La Memoria (Context & State)

L'agente ***mantiene una memoria*** di ciò che ha fatto: i risultati degli strumenti precedenti, i passaggi completati, le informazioni raccolte. Questa memoria alimenta ogni nuova decisione. Tecnicamente esistono tre livelli di memoria: ***la memoria a breve termine*** (context window della sessione corrente), ***la memoria episodica*** (risultati dei tool salvati nel context), e la ***memoria semantica a lungo termine*** (knowledge base vettoriale, implementata tramite RAG — che affronteremo nella Parte VII).

È importante capire e memorizzare questi aspetti, perché verranno ripresi e ampliati nei prossimi episodi e saranno la base teorica per lo sviluppo del ***progetto A.D.A.M***., (scaricate il file pdf con il programma completo su questa serie dedicata all'intelligenza artificiale).

4. Il Ciclo di Lavoro (ReAct Loop)

Il pattern ReAct — Reasoning + Acting, introdotto da Yao et al. nel 2022 — è la base teorica della maggior parte degli agenti moderni. L'agente alterna ragionamento esplicito (cosa devo fare, quale strumento è più adatto?) e azione concreta (invocare il tool, elaborare il risultato), valuta l'output e decide il passo successivo.



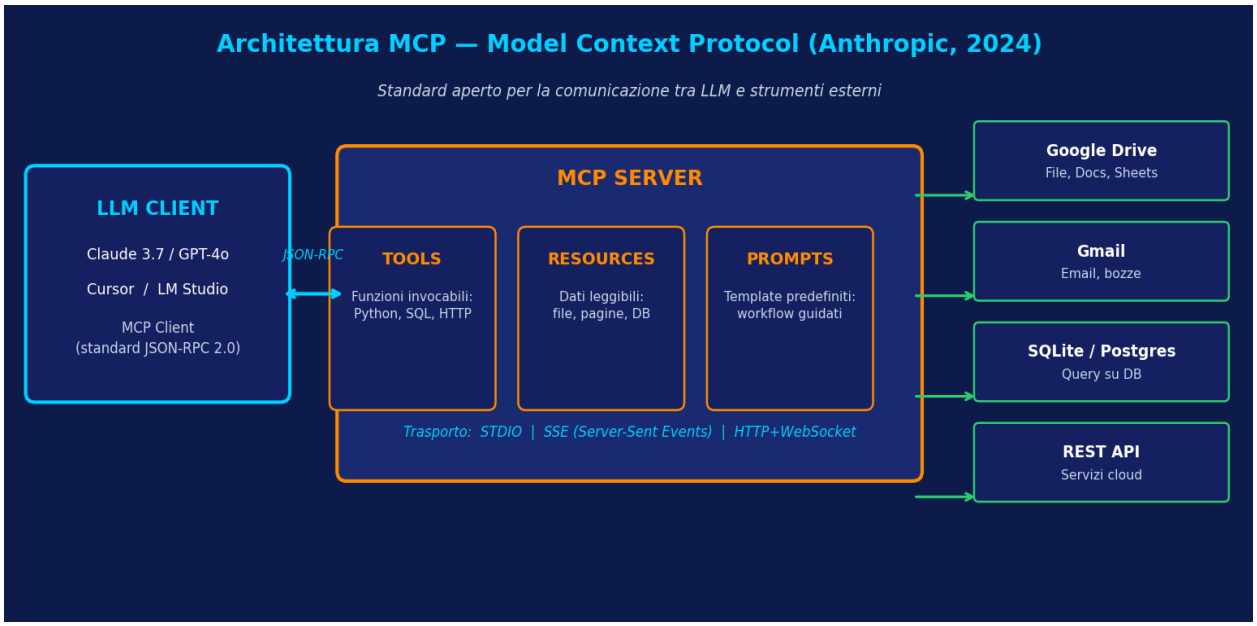
[Nota bene]
*Il pattern **ReAct** (Yao et al., 2022, arXiv:2210.03629) ha dimostrato riduzioni dell'errore fino al 34% su benchmark di problem-solving rispetto a modelli che generano direttamente la risposta finale. La forza è nella trasparenza: ogni passo di ragionamento e ogni azione sono espliciti e tracciabili — fondamentale per la verifica professionale dei risultati.*

IL PROTOCOLLO MCP: IL SISTEMA NERVOSO DEGLI AGENTI

Un tema diventato rapidamente centrale nel mondo degli agenti è il ***Model Context Protocol*** (***MCP***), introdotto da Anthropic a novembre 2024 e adottato dall'intero ecosistema AI nel corso del 2025. Nel maggio 2025 OpenAI ha annunciato il supporto nativo a MCP in GPT-4o, sancendo di fatto lo standard come protocollo universale per la comunicazione tra LLM e strumenti esterni.

MCP risolve un problema concreto: prima del suo avvento, ogni integrazione tra un LLM e un servizio esterno era un ***progetto custom***. Ogni piattaforma aveva la sua API proprietaria, ogni agente andava costruito con codice specifico. MCP standardizza il processo, un server espone

i propri strumenti in un formato standardizzato e qualsiasi client compatibile (Claude, GPT-4o, Cursor, Cline, LM Studio) può usarli senza integrazioni personalizzate.



6

COME FUNZIONA MCP TECNICAMENTE

Il protocollo si basa su ***JSON-RPC 2.0*** e può operare su tre trasporti: ***STDIO*** (per processi locali), ***SSE*** — Server-Sent Events — (per servizi web), e ***HTTP+WebSocket*** (per architetture distribuite). Un server MCP espone tre tipi di risorse:

Tipo	*Descrizione*	*Esempi tipici*
Tools	Funzioni invocabili dall'LLM	search_web(), run_python(), query_sql(), send_email()
Resources	Dati leggibili dall'LLM	Contenuto file, pagine web, record DB, output API
Prompts	Template di prompt predefiniti	Workflow guidati, istruzioni di sistema riutilizzabili

Nella mia esperienza con Claude, l'integrazione MCP ha trasformato il modo di lavorare: ***Google Drive*** per accedere ai documenti direttamente in chat, ***Gmail*** per analizzare

comunicazioni tecniche, ***Calendar*** per la pianificazione. Non serve più il copia-incolla: l'agente legge, elabora e risponde nel contesto corretto.

[Lo scenario MCP nel 2026]
A maggio 2026 il registry pubblico di server MCP conta oltre 2.000 integrazioni certificate, tra cui strumenti per Jira, Salesforce, SAP, Autodesk, e i principali servizi cloud. L'ecosistema si sta evolvendo verso MCP over HTTP con autenticazione OAuth 2.0, abilitando integrazioni enterprise-grade con requisiti di sicurezza stringenti.

I PRINCIPI FRAMEWORK AGENTICI

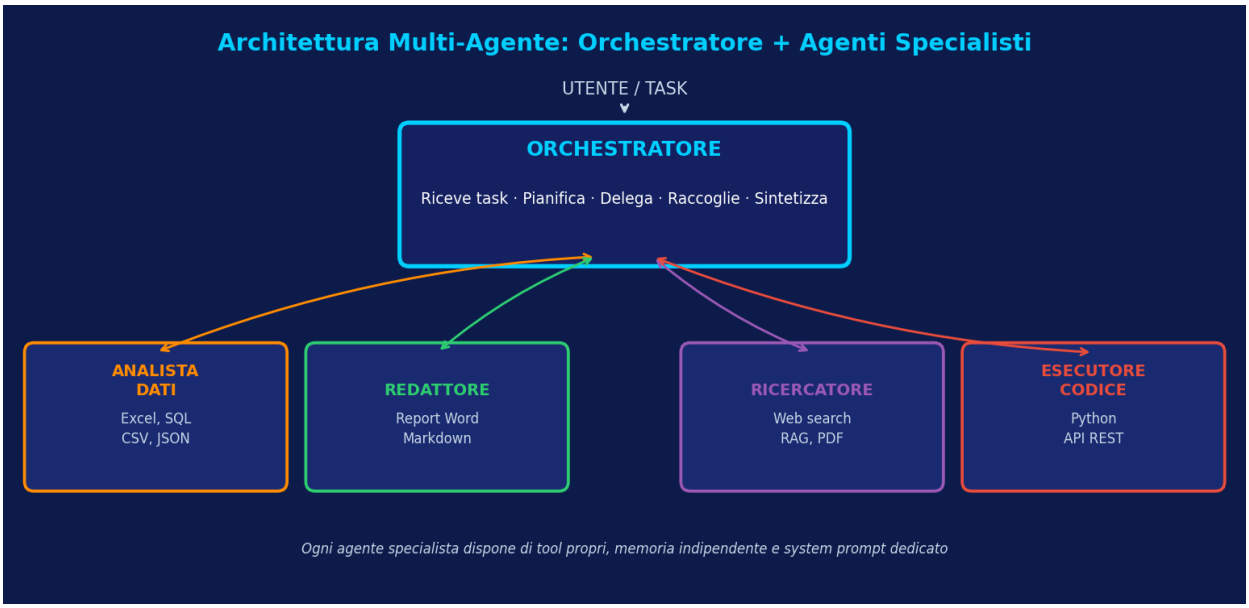
Esistono oggi diversi framework per costruire agenti IA. La scelta dipende dalla complessità del progetto, dalle competenze tecniche disponibili e dal livello di controllo richiesto.

Framework	*Sviluppatore*	*Punto di Forza*	*Ideale per*
LangChain	LangChain Inc.	Ecosistema enorme, doc. ricca	Prototipazione rapida, RAG
LangGraph	LangChain Inc.	Flussi multi-agente con stato	Workflow complessi, agenti cooperativi
AutoGen	Microsoft Research	Conversazione multi-agente	Ricerca iterativa, task complessi
CrewAI	Community	Agenti con ruoli definiti	Team virtuali specializzati per dominio
Semantic Kernel	Microsoft	Integrazione .NET / Python	Applicazioni enterprise, Azure
Claude + MCP	Anthropic	Standard aperto, privacy	Uso professionale con dati sensibili

[Il mio approccio]
*Per i progetti legati all'ingegneria elettrica preferisco **Claude + MCP**: più controllabile, trasparente e adatto a dati tecnici riservati. Per sperimentazione e prototipazione uso **LangChain**, che ha la community più attiva. Per workflow multi-step complessi sto esplorando **LangGraph**, che gestisce il controllo del flusso in modo esplicito e verificabile.*

ARCHITETTURA MULTI-AGENTE

Un'evoluzione significativa rispetto all'agente singolo è ***l'architettura multi-agente***, in cui più agenti specializzati collaborano sotto la supervisione di un ***orchestratore***. Il vantaggio principale è la specializzazione: ogni agente è ottimizzato per un tipo specifico di compito, con il proprio system prompt, i propri tools e la propria memoria.



Architettura Multi-Agente: orchestratore centrale e agenti specialisti indipendenti

L'orchestratore riceve il task dall'utente, lo decompone in sotto-task e li assegna agli agenti specialisti. Ogni agente esegue autonomamente la propria parte, restituisce il risultato all'orchestratore, che assembla la risposta finale. Questo pattern è particolarmente efficace per:

- ***Task complessi*** che richiedono competenze eterogenee (analisi dati + scrittura + ricerca)
- ***Parallelizzazione***: più agenti lavorano simultaneamente su sotto-task indipendenti
- ***Specializzazione di dominio***: ogni agente ha il system prompt ottimizzato per il suo ruolo
- ***Scalabilità***: si aggiungono nuovi specialisti senza modificare l'orchestratore

[Nota Tecnica]
*I framework come **LangGraph** e **AutoGen** supportano esplicitamente i pattern multi-agente con gestione dello stato condiviso, meccanismi di handoff tra agenti e modalità di consenso per task critici. La comunicazione tra agenti avviene tipicamente tramite messaggi strutturati nel context condiviso dell'orchestratore.*

CASI D'USO REALI: DALL'IDEA ALL'APPLICAZIONE

La parte più interessante degli agenti è quella concreta. Descrivo tre scenari reali nel contesto professionale dell'ingegneria elettrica, con dettaglio tecnico sufficiente a rendere ciascuno replicabile.

Caso 1 - Analisi Automatica delle Bollette Elettriche

Ho costruito un flusso agentico che: legge i PDF delle bollette mensili Edison, estrae i consumi per fascia oraria (F1, F2, F3), calcola il costo medio mensile per fascia, confronta i dati con i mesi precedenti, identifica anomalie di consumo superiori al 15% rispetto alla media storica e genera automaticamente un file Excel con dashboard e grafici. Il flusso usa tre strumenti: un parser PDF (pdfplumber), un motore di calcolo Python e un generatore Excel (openpyxl). L'LLM orchestra il tutto, gestisce gli errori di parsing e aggiunge il commento interpretativo finale.

[Risultato pratico]
Quello che prima richiedeva 45 minuti di lavoro manuale ora richiede 3 minuti di supervisione. Il prompt di orchestrazione: «Sei un analista energetico esperto. Leggi le bollette Edison allegate, estrai i consumi per fascia oraria, calcola i KPI richiesti, identifica anomalie e genera il report Excel. Per ogni anomalia spiega la causa più probabile e suggerisci un'azione correttiva.»

Caso 2 — Verifica Normativa Automatica

Un secondo caso d'uso riguarda la verifica della conformità normativa di un progetto elettrico. L'agente legge i file di progetto (schema unifilare in PDF, relazione tecnica), identifica i componenti citati, interroga una knowledge base locale con le norme CEI pertinenti tramite

RAG (come vedremo nella Parte VII), verifica la corrispondenza tra le caratteristiche dei componenti e i requisiti normativi, e produce un report di conformità con eventuali non-conformità evidenziate. Questo tipo di agente non sostituisce la responsabilità professionale del progettista, ma riduce drasticamente il tempo di verifica e il rischio di dimenticare un requisito normativo.

Caso 3 — Generazione di Documentazione Tecnica

Il terzo caso è quello che uso più frequentemente: un agente che, partendo dai dati di un calcolo illuminotecnico (superfici, altezze, valori di illuminamento, tipologia di lampade), genera automaticamente la relazione tecnica completa in formato Word, con sezioni standardizzate, tabelle di risultato, verifica normativa rispetto alla UNI EN 12464-1 e note esplicative. Il formato segue esattamente la struttura della Guida al Calcolo Illuminotecnico, disponibile sul sito **mondoroccomassimiliano.it**. Il guadagno di tempo è stimabile in 60-70% rispetto alla redazione manuale, con un livello di consistenza formale superiore.

Human-in-the-Loop: Il Controllo Professionale

Un tema spesso sottovalutato nella progettazione di sistemi agentici è il ruolo del supervisore umano. ***L'automazione non è un valore assoluto***: in molti contesti professionali è fondamentale mantenere il controllo su ogni azione critica dell'agente.

Il pattern ***Human-in-the-Loop*** (HITL) prevede punti di controllo espliciti nel flusso agentico, in cui l'esecuzione si interrompe e attende la conferma dell'operatore prima di procedere. Questo è obbligatorio per azioni irreversibili come l'invio di email, la scrittura su database di produzione, la modifica di file firmati digitalmente.

Esistono tre livelli di supervisione, con ***trade-off*** diversi tra autonomia e sicurezza:

Livello HITL	*Descrizione*	*Quando usarlo*
Full Autonomy	Agente esegue tutto senza interruzioni	Task a basso rischio, output reversibili
Confirm Critical	Pausa su azioni irreversibili o ad alto impatto	Invio email, scrittura DB, firma documenti
Step-by-Step	Approvazione umana a ogni passo del loop	Primo utilizzo, flussi critici, test

HUMAN-IN-THE-LOOP: IL CONTROLLO PROFESSIONALE

- Errori a cascata: se il primo passo del flusso è sbagliato, gli errori si propagano e si amplificano. Un dato estratto male da un PDF può produrre calcoli completamente errati in tutti i passi successivi. La soluzione è la validazione esplicita degli output intermedi.
- Azioni irreversibili: un agente con accesso alla posta può inviare email, uno con accesso al filesystem può cancellare file. Prima di abilitare questi strumenti, è essenziale definire confini chiari di operatività tramite policy esplicite nel system prompt.
- Allucinazione strumentale: l'LLM può «inventare» il risultato di uno strumento se l'output è ambiguo. La validazione esplicita dei risultati intermedi è obbligatoria nei flussi critici.
- Prompt injection: un agente che legge contenuti da fonti esterne (web, email, file) può essere manipolato da testo malevolo che contiene istruzioni camuffate da contenuto. Le architetture sicure isolano il context agentico dal contenuto non fidato.
- Dipendenza dall'API: gli agenti cloud dipendono dalla disponibilità del servizio. Per applicazioni mission-critical è opportuno prevedere modalità di fallback e logging completo delle azioni.

[Regola pratica]
Iniziate sempre con agenti in modalità 'sola lettura': leggono, analizzano, propongono, ma non scrivono né inviano nulla senza conferma esplicita. Solo dopo aver verificato la qualità degli output in decine di casi reali considerate l'automazione completa. Il principio del minimo privilegio, fondamentale in sicurezza informatica, si applica anche agli agenti IA.

CONCLUSIONE

Gli agenti IA rappresentano la frontiera più concreta e immediatamente produttiva dell'intelligenza artificiale generativa. Non sono fantascienza: sono strumenti disponibili oggi, accessibili anche senza competenze di programmazione avanzate, che possono moltiplicare la produttività su compiti ripetitivi e ad alta intensità informativa.

La chiave è la consapevolezza: un agente non è un sostituto del giudizio professionale, ma un ***amplificatore della capacità di lavoro***. Funziona tanto meglio quanto più precisa e strutturata è la sua orchestrazione — e avete indovinato: anche qui ***il prompt engineering*** della Parte IV torna ad essere il fattore determinante.

Nel prossimo episodio affronteremo una delle sfide più concrete dell'IA professionale: come dare a un LLM accesso alla nostra conoscenza privata — i nostri documenti, le nostre normative, i nostri archivi tecnici — senza compromettere la privacy. La risposta si chiama ***RAG: Retrieval Augmented Generation.***