IMPROVE AND MANAGE

Agenti AI: Prospettive, Sfide e Architetture per un Futuro Inclusivo e Sostenibile

Fabio Malpezzi

Executive Tecnologico con oltre 30 anni di esperienza nella trasformazione digitale e nell'implementazione di sistemi ERP globali. Specializzato in migrazioni complesse su Microsoft Dynamics 365 Finance & Operations (F&O), ho guidato con successo numerosi progetti multinazionali, affiancando le aziende nel percorso di innovazione tecnologica e crescita operativa.

Esperienza

Ho gestito e completato otto migrazioni multi-paese a Dynamics 365 F&O, dimostrando competenza approfondita in:

- Reingegnerizzazione e ottimizzazione dei processi aziendali nel manufacturing e nella supply chain
- Integrazione end-to-end tra logistica, contabilità e sistemi CRM
- Change management strategico per favorire l'adozione cross-culturale e interdipartimentale
- Automazione avanzata basata su intelligenza artificiale e implementazione di sistemi predittivi personalizzati

Combino una solida competenza tecnica su Dynamics 365 F&O e intelligenza artificiale con una visione strategica consolidata, documentata sul mio blog Improve & Manage (https://improveandmanage.com). Attraverso il blog condivido analisi approfondite e casi di studio concreti, offrendo alle aziende soluzioni pragmatiche per affrontare le sfide della digital transformation in ambienti altamente complessi.

🌐 Blog Improve & Manage | 🔗 LinkedIn

Sommario

Agenti AI: Prospettive, Sfide e Architetture per un Futuro Inclusivo e Sostenibile 1

1. Contesto e Definizione degli Agenti AI 4
   1.1 Breve inquadramento storico 4
   1.2 Verso gli Agenti AI moderni 4
2. Architettura e Principi di Funzionamento 5
   2.1 Modello di linguaggio (LLM) 5
   2.2 Motore di orchestrazione 5
   2.3 Modulo di ragionamento 5
   2.4 Interfacciamento esterno (tool/plugin) 6
   2.5 Monitoraggio e feedback 6
3. Benefici e Punti di attenzione 7
   3.1 Benefici 7
   3.2 Punti di attenzione 8
4. Protocolli di Colloquio Emergenti 9
5. Impatto Computazionale ed Energetico 10
   5.1 Fattori di Costo 10
   5.2 Soluzioni di Ottimizzazione 10
   5.3 Esempi Pratici 11
   5.4 Sfide di Sostenibilità 11
6. Opportunità, Prospettive Future e Implicazioni Socio-Economiche degli Agenti AI 12
   6.1 Autonomia Crescente 12
   6.2 Modelli Specializzati 13
   6.3 Integrazione con IoT e Robotica 13
   6.4 Standardizzazione 13
   6.5 Formazione 13
   6.6 Diseguaglianze e Accesso alla Tecnologia 13
   6.7 Impatti Economici Sistemici 13
   6.8 Sfide Aperte 14
7. Architetture multi agente 15
   7.1 Architettura Gerarchica 15
   7.2 Architettura con Intervento Umano 15
   7.3 Architettura a Rete 15
   7.4 Architettura Sequenziale 15
   7.5 Architettura di Trasformazione dei Dati 16

7.6 Altri Pattern 16
7.7 Framework decisionale per selezionare l'architettura multi-agente più adatta 16
8. Conclusioni 21
Azioni strategiche per le Aziende e I Policy Maker 23
1. Investire in Formazione e Competenza 23
2. Adottare Soluzioni Open-Source 23
3. Promuovere la Standardizzazione 24
4. Ottimizzare le Risorse Computazionali 24
5. Implementare Governance Etica 24
6. Favorire l'Inclusività 24
9. Fonti e possibili approfondimenti 25

1. Contesto e Definizione degli Agenti AI

Negli ultimi anni, i sistemi di Intelligenza Artificiale (IA) hanno conosciuto una rapida evoluzione, con gli “Agenti AI” che emergono come strumenti capaci di svolgere compiti autonomi o semi-autonomi. L’avvento di modelli di linguaggio di grandi dimensioni (LLM) e di metodologie di prompt engineering ha permesso di collegare tali modelli a sistemi software capaci di interpretare, elaborare e gestire flussi di dati complessi in linguaggio naturale.

1.1 Breve inquadramento storico

La concezione di “agente” come entità software, a volte con componenti hardware, risale agli anni ’90, quando si diffusero i primi studi sui sistemi multi-agente, come il modello BDI (Beliefs, Desires, Intentions). In tali sistemi, un “agente” possedeva stati interni (credenze, desideri e intenzioni) e regole di ragionamento per perseguire determinati obiettivi. L’avvento del machine learning e dell’elaborazione del linguaggio naturale (NLP) ha trasformato profondamente questo scenario: dagli agenti basati su regole statiche if-then si è passati a entità capaci di apprendere dal contesto e di dialogare in modo evoluto con esseri umani o altri sistemi.

1.2 Verso gli Agenti AI moderni

Oggi, un Agente AI integra ragionamento, pianificazione e apprendimento, sfruttando gli LLM (Large Language Model, es. ChatGPT) per comprendere linguaggio naturale, dialogare e decidere su dati non strutturati. Questa capacità semantica fa la differenza rispetto ai software tradizionali, in quanto gli agenti possono interpretare istruzioni complesse, estrarre informazioni da fonti eterogenee (email, PDF, articoli, pagine web) e orchestrare diverse API o tool, adattandosi in modo dinamico senza rigide strutture predefinite.

Gli Agenti AI vengono impiegati in una moltitudine di compiti: dall’automazione del servizio clienti e della gestione delle email, all’analisi di grandi volumi di dati, fino alla pianificazione di attività complesse in contesti logistici o produttivi. Il risultato è un salto concettuale rispetto a sistemi basati su query codificate (SQL, logiche a blocchi) e un’accelerazione della trasformazione digitale in molte realtà aziendali.

2. Architettura e Principi di Funzionamento

Un Agente AI tipicamente si articola in più livelli o moduli, che cooperano per offrire comportamento "intelligente" e capacità di interazione in linguaggio naturale.

2.1 Modello di linguaggio (LLM)

È il "cuore" dell'agente, responsabile dell'elaborazione semantica di testi e interazioni. Questi modelli (ad es. GPT, Claude, Mistral, Grok, Gemini, Deepseek, ecc.) sono reti neurali di grandi dimensioni, addestrate su dataset di testo molto vasti.

- *Prompt e contesto:* L'agente guida il modello di linguaggio attraverso un prompt, ovvero un insieme di istruzioni, esempi e informazioni di contesto fornite in linguaggio naturale. Ad esempio, un prompt potrebbe essere: «Agisci come un assistente di supporto clienti e rispondi alle domande con un tono cortese e professionale». Queste istruzioni, chiamate prompt, sono di solito predefinite al momento della configurazione dell'agente, orientandone il comportamento.
- *Gestione di bias e allucinazioni*: È essenziale monitorare la qualità delle risposte generate, poiché gli LLM possono occasionalmente produrre contenuti inesatti (le cosiddette "allucinazioni") o riflettere pregiudizi presenti nei dati su cui sono stati addestrati, richiedendo attenzione per garantire affidabilità e correttezza.

2.2 Motore di orchestrazione

Coordina i flussi di lavoro, decide come distribuire le richieste tra sottosistemi (tool o plugin) e tiene traccia dei passaggi da compiere. Può attivare, ad esempio, un agente specializzato nell'estrazione di dati da un database o un altro agente dedicato all'analisi semantica di PDF.

2.3 Modulo di ragionamento

Può essere basato su IA simbolica, logica formale o approcci ibridi (reti neurali + regole), utile per

interpretare l'ambiente e gestire compiti più complessi (ad es. pianificazione a lungo termine, risoluzione di problemi multi-step).

2.4 Interfacciamento esterno (tool/plugin)

L'agente si collega a servizi web, API di terze parti, database o sensori IoT, sfruttando meccanismi di configurazione a runtime. In alcuni casi, è lo stesso LLM a generare istruzioni per invocare i tool (funzioni, script, comandi esterni).

2.5 Monitoraggio e feedback

Registra le interazioni, verifica gli output dell'agente, corregge bias o errori sistematici e mantiene log di audit. Questo aspetto diventa essenziale in contesti sensibili (sanità, finanza, pubblica amministrazione), dove l'accuratezza e la tracciabilità delle decisioni sono cruciali.

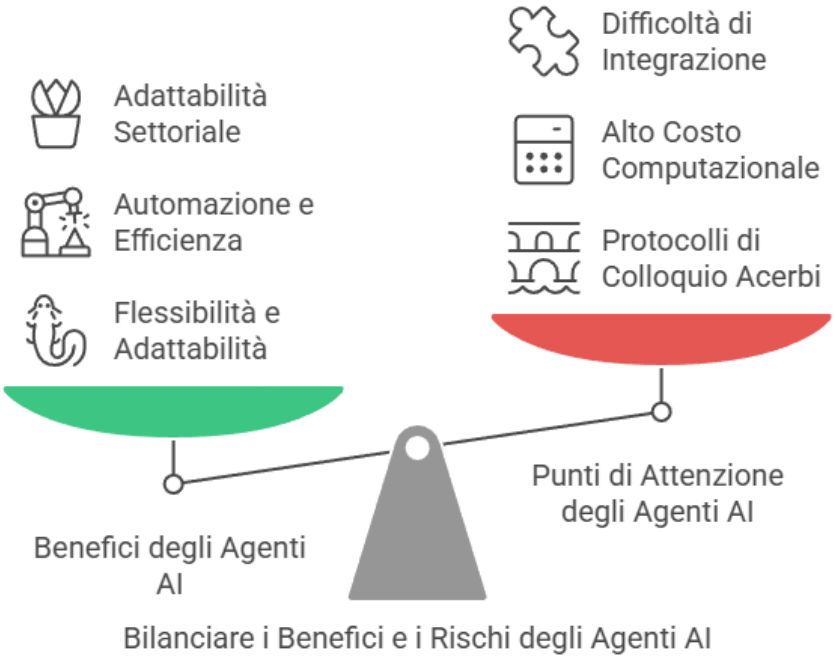
Come i LLM gestiscono il linguaggio e l'importanza dei prompt

Le moderne architetture di reti neurali (come i Transformer) scompongono il testo in "token" per apprendere relazioni contestuali tra frammenti di testo. Il prompt engineering permette di guidare il modello verso risposte più coerenti, definendo ruoli, obiettivi e stile comunicativo. In assenza di un contesto chiaro, l'LLM tende a generare contenuti generici o a fraintendere la richiesta, mettendo in evidenza quanto sia essenziale formulare prompt ben strutturati e supportati da esempi.

Che cos'è il "few-shot prompting"?

Il "few-shot prompting" consiste nell'includere nel prompt alcuni esempi che illustrano il formato o lo stile desiderato delle risposte. Ad esempio, si possono fornire al modello un paio di input, ciascuno con la sua risposta ottimale. Questo aiuta l'LLM a "imparare" rapidamente dall'esempio fornito, senza richiedere un addestramento esteso o un vasto dataset. Con pochi esempi mirati (da cui la definizione di *few-shot*), il modello riesce a generalizzare meglio e a produrre risposte allineate alle esigenze dell'utente.

3. Benefici e Punti di attenzione



Bilanciare i Benefici e i Rischi degli Agenti AI

3.1 Benefici

1. *Gestione Semantica dei Dati*
   Grazie alla capacità di interpretare testi non strutturati, gli Agenti AI eccellono nella cosiddetta *semantic data integration*. A differenza dei software di Business Intelligence tradizionali (basati su database relazionali e query rigide), un Agente AI può estrarre concetti chiave da documenti eterogenei e gestire flussi di dati complessi senza la necessità di definire schemi fissi in anticipo.
   - *Esempio pratico*: un Agente AI incaricato di "filtrare e comprendere" migliaia di email di reclamo in un e-commerce, categorizzando automaticamente i problemi e proponendo soluzioni personalizzate. Un classico sistema di BI richiederebbe campi predefiniti e processi di ETL (Extract, Transform, Load) più lunghi da implementare.
2. *Estrema Flessibilità*
   Un Agente AI si "riconfigura" a runtime variando i prompt e i plugin. Se un agente è stato progettato per assistere i clienti, basta cambiare il prompt per focalizzarlo su un dominio diverso (ad es. analisi dati di mercato, segnalazioni IT, processi HR). Questa plasticità riduce il time-to-market e i costi di sviluppo: non occorre riscrivere codice ogni volta, ma solo fornire istruzioni e strumenti adeguati al nuovo compito.
3. *Riduzione del carico umano in processi ripetitivi*
   Automatizzando mansioni come la lettura e la catalogazione di documenti, la prima assistenza al cliente o l'elaborazione di report standard, gli Agenti AI permettono di liberare risorse umane per attività più creative, strategiche o di controllo qualità.
4. *Adattabilità a domini specialistici*
   Mentre i LLM generalisti coprono un ampio spettro di conoscenze, le aziende possono

addestrare (o “raffinare”) modelli specializzati su dataset verticali (es. medicina, finanza, manifattura) ottenendo accuratezza superiore in settori molto tecnici, senza dover predisporre complicati sistemi di regole.

3.2 Punti di attenzione

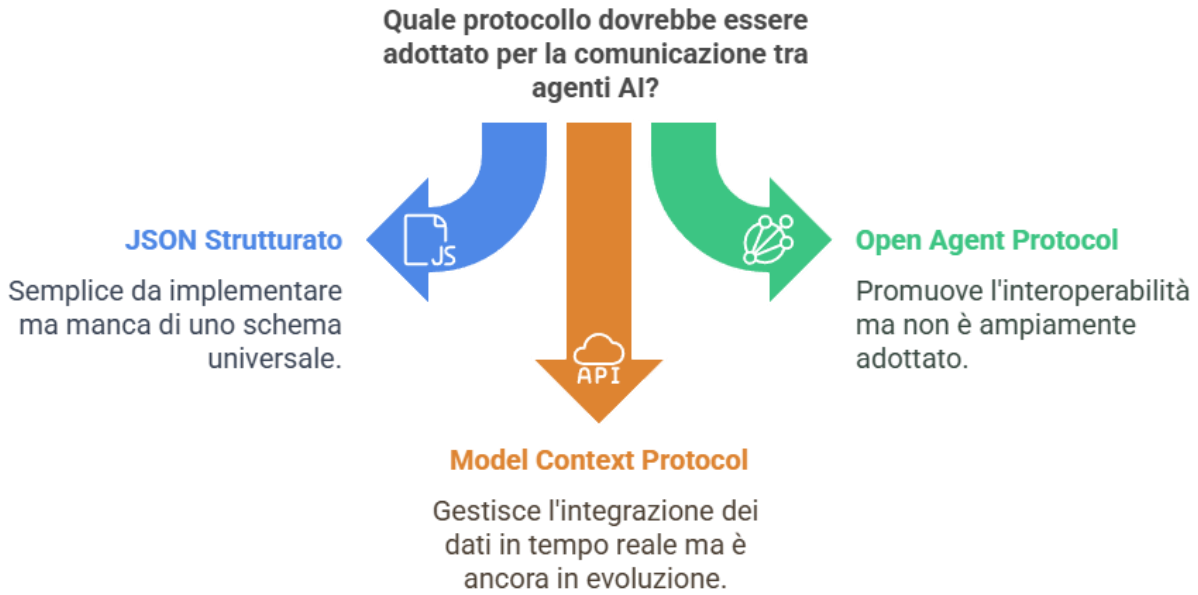
1. *Protocolli di Colloquio ancora acerbi*
   Non esistono ancora standard pienamente condivisi per formati di messaggio, autenticazione e orchestrazione multi-agente. Alcune proposte (JSON strutturato, Open Agent Protocol) stanno emergendo, ma la frammentazione è elevata, costringendo le aziende a implementazioni proprietarie o a ricorrere a framework in rapida evoluzione (LangChain, Auto-GPT, ecc.). Ciò può causare lock-in, costi di integrazione e problemi di interoperabilità.

2. *Alto Costo Computazionale ed Energetico*
   L’esecuzione di LLM (specie quelli più grandi come GPT-4) comporta un consumo elevato di risorse (GPU/TPU), incidendo sia sui costi diretti (hosting in cloud o acquisto di hardware specializzato) sia sul piano ambientale. Il modello di addestramento richiede inoltre risorse enormi, anche se, nella maggior parte dei casi, le aziende utilizzano versioni pre-addestrate fornite da terzi.

3. *Difficoltà di integrazione e “cambio di mentalità”*
   Le organizzazioni, per adottare appieno gli Agenti AI, devono formare il personale su concetti come il prompt engineering e le potenzialità/limiti dei LLM. Questo implica un cambiamento culturale, passando dai processi rigidamente strutturati a un approccio più flessibile ma anche più “incerto”. Senza la dovuta formazione e consapevolezza, gli operatori potrebbero commettere errori (es. prompt ambigui) o affidarsi troppo a sistemi non controllati.

4. *Aspetti etici e di responsabilità*
   In settori come la sanità o la finanza, occorre chiarire chi risponde di eventuali errori commessi dall’agente, quali dati sono processati e come viene garantita la conformità alle normative sulla privacy. L’assenza di standard condivisi aggrava il problema, rendendo cruciale un monitoraggio umano (human-in-the-loop) nelle fasi più sensibili.

4. Protocolli di Colloquio Emergenti



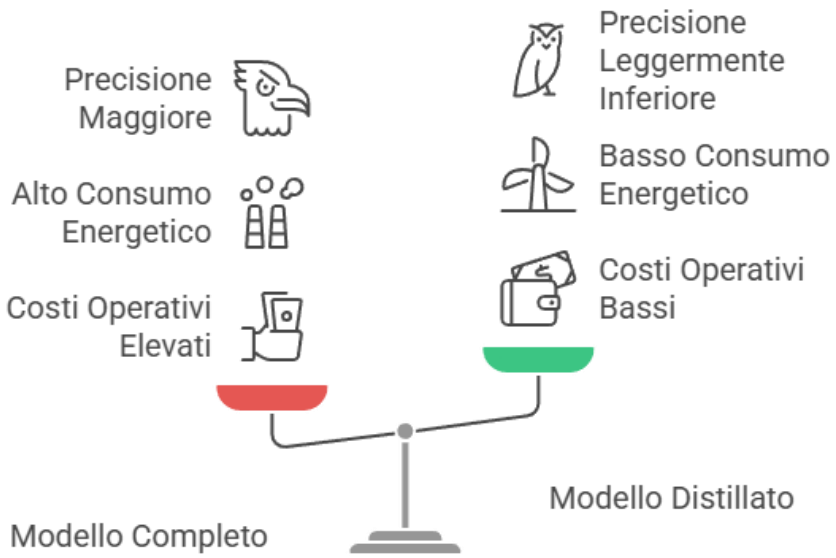
La cooperazione tra più agenti AI, o tra agenti e LLM, richiede protocolli di comunicazione standardizzati, che definiscano la struttura dei messaggi, la semantica dei comandi e le procedure di autenticazione/scambio dati. Tuttavia, tali standard sono ancora in via di maturazione:

- **JSON Strutturato**
  Un approccio semplice consiste nel racchiudere prompt e risposte in formati JSON, con campi come "action", "arguments", "response_type". Framework quali *LangChain* e *Auto-GPT* sperimentano diverse varianti. Il limite è la mancanza di uno schema universale condiviso.
- **Open Agent Protocol (OAP)**
  Propone un linguaggio condiviso per la comunicazione fra agenti, basato su messaggi standardizzati e cicli di richiesta-risposta. L'idea è fornire un ordito comune che faciliti l'orchestrazione e l'interoperabilità, ma non ha ancora ricevuto un'adozione massiccia dai principali provider LLM.
- **Model Context Protocol (MCP)**
  Sviluppato da Anthropic, mira a gestire in modo standardizzato l'integrazione tra modelli AI e fonti dati real-time (API, file system, CRM). Utilizza JSON-RPC 2.0 e protocolli di autenticazione centralizzati. È promettente ma ancora in evoluzione.

Perché la standardizzazione è importante
Come accaduto per i protocolli di rete (TCP/IP, HTTP), la definizione di uno standard condiviso permetterebbe agli agenti di collaborare senza dover sviluppare continuamente adattatori custom. Inoltre, chiarirebbe aspetti di sicurezza, responsabilità e governance, rendendo più agevole l'adozione degli Agenti AI su larga scala.

5. Impatto Computazionale ed Energetico

L'impiego di LLM su vasta scala ha un impatto rilevante in termini di costi, latenza e consumi energetici, sia in fase di training, sia durante l'inferenza.



Bilanciare Costi e Precisione nei Modelli LLM

5.1 Fattori di Costo

- **Hardware**
  Schede GPU avanzate (es. NVIDIA H100) costano circa € 30.000 di euro l'una, e un singolo server può contenerne sino ad 8. In un data center aziendale di grandi dimensioni, con centinaia o migliaia di server, l'investimento in hardware può diventare rapidamente molto elevato.
- **Energia**
  Un server con 8 GPU A100 assorbe 6-7 kW e un intero data center può arrivare a diverse decine di MW. A ciò si sommano i costi di raffreddamento e manutenzione.
- **Cloud**
  Provider come AWS, Azure e GCP offrono GPU a noleggio per 3-5€/ora per singola GPU, con costi mensili facilmente superiori a diverse migliaia di euro se l'agente è attivo 24/7.

5.2 Soluzioni di Ottimizzazione

- **Quantizzazione e Pruning**
  Riduzione della precisione numerica (es. 16 bit → 8 bit) o rimozione di neuroni/pesi non essenziali, diminuendo la dimensione del modello senza compromettere eccessivamente l'accuratezza.
- **Modelli Distillati**
  Versioni "leggere" (DistilBERT, DistilGPT) che replicano gran parte delle funzionalità dei modelli originali con un numero minore di parametri.

- **Edge Computing**
  Spostamento di alcune parti di calcolo su dispositivi locali, evitando di dover mantenere costantemente un grande modello in cloud.

5.3 Esempi Pratici

- **Servizio clienti**
  Un'azienda con 10.000 interazioni giornaliere passa da un LLM completo su cloud (≈ 500€/giorno) a un modello distillato on-prem (≈ 50€/giorno, con hardware locale). La precisione scende di poco, ma il risparmio è significativo.

- **Monitoraggio IoT**
  Sensori in una fabbrica usano agenti leggeri (pruning + quantizzazione) per l'analisi in tempo reale, riducendo il consumo energetico da 100W a 20W per dispositivo.

5.4 Sfide di Sostenibilità

Le fonti di energia per i data center spesso non sono rinnovabili, con un impatto ecologico (emissioni di CO_2) potenzialmente elevato. Molte aziende stanno valutando hardware più efficiente (ASIC, TPU) o investendo in forniture di energia pulita per mitigare l'impronta ambientale. Tuttavia, i costi iniziali e la complessità di tali soluzioni favoriscono soprattutto le grandi imprese, mentre PMI e startup rischiano di rimanere escluse, rallentando l'adozione inclusiva degli Agenti AI.

6. Opportunità, Prospettive Future e Implicazioni Socio-Economiche degli Agenti AI

Nonostante le sfide, gli Agenti AI offrono un potenziale di trasformazione notevole in svariati settori, prospettando un futuro in cui software e IA si integrano in modo sempre più “umano-centrico” e autonomo.

6.1 Autonomia Crescente

Si delineano scenari di *self-driving agents* capaci di decidere quali task affrontare, in che ordine, e come ottimizzare i risultati con minima supervisione umana. In una filiera logistica, l’agente potrebbe ricalcolare percorsi in tempo reale, gestire scorte e inviare notifiche a fornitori, riducendo fino al 20% i tempi di consegna. Questioni etiche e di responsabilità diventano centrali: chi paga i danni se l’agente sbaglia in modo autonomo?

6.2 Modelli Specializzati

Cresce la necessità di LLM verticali, addestrati su dataset specialistici. Ad esempio, in medicina, un modello addestrato su radiologia può analizzare TAC più accuratamente di un LLM generalista, usando meno risorse computazionali. Questo avvicina le performance di un agente AI a quelle di un esperto umano nel settore, pur mantenendo la flessibilità tipica dei modelli di linguaggio.

6.3 Integrazione con IoT e Robotica

Gli Agenti AI coordinano sensori e robot, assumendo decisioni immediate in situazioni reali. In una catena di montaggio, un robot guidato dall'agente AI può riallineare la produzione in seguito a un guasto di un macchinario, oppure chiedere un intervento umano solo se necessario. La latenza di comunicazione e la robustezza dei sistemi diventano requisiti tecnici cruciali.

6.4 Standardizzazione

Consorzi industriali e progetti open source puntano a definire protocolli e framework condivisi, incentivando la nascita di ecosistemi interoperabili. L'adozione di uno standard di comunicazione e la condivisione di best practice per la sicurezza e la privacy potrebbero sbloccare la vera diffusione di massa degli Agenti AI, rendendoli plug-and-play in vari contesti.

6.5 Formazione

Manager e professional di vari settori dovranno acquisire competenze di base su LLM, prompt engineering e gestione degli Agenti AI. La mancanza di personale formato rischia di frenare l'innovazione o di generare errori di valutazione (es. sovrastimare l'affidabilità di un sistema). L'aggiornamento continuo è quindi una priorità, data la rapidità con cui l'IA evolve.

6.6 Diseguaglianze e Accesso alla Tecnologia

L'adozione degli Agenti AI rischia di accentuare il digital divide, sia tra aziende che tra aree geografiche. Le grandi corporation possono investire in soluzioni proprietarie, mentre le PMI, con solo il 15% di adozione in Europa nel 2024 contro il 62% delle grandi imprese, faticano a tenere il passo. A livello globale, gli hub tecnologici di USA, Cina ed Europa occidentale dominano lo sviluppo, lasciando indietro le economie emergenti. Per mitigare queste disparità, servono politiche di finanziamento per le PMI, risorse condivise come modelli pre-addestrati, e soluzioni tecniche "frugali" che riducano i requisiti computazionali.

6.7 Impatti Economici Sistemici

Sul fronte economico, gli Agenti AI promettono un incremento della produttività globale dell'1,5% annuo (Goldman Sachs, 2023), con un impatto potenziale di 7 trilioni di dollari sul PIL mondiale entro il 2030. Tuttavia, questo beneficio si scontra con il rischio di concentrazione del potere economico nelle mani di poche big tech, alimentato da economie di scala e dinamiche winner-takes-most. La sostenibilità rimane un ulteriore nodo: bilanciare efficienza economica, impatto ambientale e distribuzione equa dei benefici sarà cruciale per un'adozione responsabile.

6.8 Sfide Aperte

L'evoluzione degli Agenti AI richiede di affrontare ostacoli complessi che spaziano dalla gestione delle risorse all'adattamento delle strutture sociali e aziendali. Per sfruttare appieno il loro potenziale, è necessario rispondere a esigenze diverse, tra cui:

- Economiche: garantire che i benefici della tecnologia siano accessibili a tutti, evitando barriere che possano favorire oligopoli o concentrare il potere economico in poche mani;
- Etiche: assicurare audit rigorosi, trasparenza e meccanismi di verifica per prevenire bias e scelte non comprensibili;
- Tecniche: garantire scalabilità e robustezza degli agenti in ambienti dinamici, riducendo al contempo costi e latenza;
- Organizzative: trasformare la cultura aziendale per superare la resistenza al cambiamento, gestendo i costi di transizione e la redistribuzione del lavoro umano.

7. Architetture multi agente

In molti casi, la soluzione più efficace per compiti complessi è la collaborazione di più agenti, ciascuno specializzato in un'area (ricerca dati, analisi, interazione con l'utente, ecc.). Ciò replica la suddivisione del lavoro tipica dei team umani, aumentando flessibilità e resilienza.

7.1 Architettura Gerarchica

Un agente supervisore coordina agenti specializzati, aggregando risultati. È indicata per scenari strutturati, ma può diventare un collo di bottiglia.

- *Esempio*: un supervisore di una banca gestisce richieste di analisi creditizia, delegandole a (a) agente per storico dati interni, (b) agente per rating esterni, (c) agente per email del cliente. Infine, compone un report unico.

7.2 Architettura con Intervento Umano

Prevede che un operatore verifichi e approvi determinate azioni sensibili, garantendo compliance e sicurezza.

- *Esempio*: in un ospedale, un agente AI propone una terapia sulla base di cartelle cliniche e linee guida mediche, ma un medico deve convalidarla prima dell'applicazione. Ciò riduce i rischi in settori regolamentati.

7.3 Architettura a Rete

Agenti paritari collaborano senza supervisore. È un modello resiliente, ma complesso da sincronizzare man mano che gli agenti aumentano.

- *Esempio*: in una supply chain globale, agenti dislocati nei magazzini (inventario), trasporti (camion e navi) e punti vendita (domanda locale) dialogano fra loro per ottimizzare stock e flussi logistici.

7.4 Architettura Sequenziale

Pipeline lineare in cui ogni agente usa l'output del precedente.

- *Esempio*: in un'agenzia marketing, (1) un agente recupera dati di campagne pubblicitarie, (2) un altro calcola il ROI, (3) un terzo genera il report finale da consegnare al cliente.

7.5 Architettura di Trasformazione dei Dati

Agenti dedicati ad arricchire o normalizzare i dati prima che vengano utilizzati da altri agenti.

- *Esempio*: un agente di "data enrichment" etichetta sentiment ed emozioni nelle recensioni, un altro effettua l'anonimizzazione per rispettare la GDPR, prima di passare il tutto a un agente di analisi.

7.6 Altri Pattern

- **Loop**: l'agente o il gruppo di agenti iterano finché non raggiungono una certa qualità nell'output (es. correzione bug in un software).
- **Parallel**: agenti in parallelo per ridurre i tempi di elaborazione.
- **Router**: un agente "router" classifica le richieste e le smista agli agenti specialisti.

7.7 Framework decisionale per selezionare l'architettura multi-agente più adatta

Scegliere l'architettura multi-agente più appropriata per un determinato contesto richiede un'analisi attenta delle caratteristiche di ogni modello e dei requisiti specifici del sistema. Ogni architettura porta con sé punti di forza e limiti, che si riflettono in aspetti come scalabilità, flessibilità, resilienza, complessità implementativa, necessità di supervisione umana, efficienza computazionale, latenza decisionale e coerenza dell'output. Per orientarsi nella decisione, è utile partire da alcune domande di fondo. Quali sono gli obiettivi principali del sistema: velocità, accuratezza o trasparenza? Ci sono vincoli normativi o di compliance da rispettare? Quanto intervento umano è accettabile o necessario? Poi, vanno considerate le risorse a disposizione: l'infrastruttura tecnologica, le competenze del team e il budget per implementazione e manutenzione. Un altro aspetto cruciale è l'escalabilità futura: come evolverà il sistema? Quali volumi di dati o richieste dovrà gestire? In molti casi, la soluzione più efficace potrebbe essere un approccio ibrido, ad esempio combinando una struttura gerarchica con intervento umano per decisioni critiche, o integrando un router centrale in un'architettura a rete per ottimizzare le richieste.

Quando Preferire Ciascuna Architettura

1. Architettura Gerarchica
Descrizione: Un modello centralizzato in cui un agente supervisore coordina gli altri, garantendo ordine ed efficienza.

- Quando preferirla: Questo approccio si rivela ideale nei contesti organizzativi dove le linee di responsabilità e rendicontazione sono ben definite, per compiti complessi che richiedono un coordinamento centralizzato, quando serve mantenere una visione d'insieme del processo o in ambienti con elevati requisiti di conformità e controllo.
- Esempi di applicazione ideale: Si adatta perfettamente a sistemi di gestione finanziaria e creditizia, a procedure di approvazione multi-livello in contesti aziendali e alla gestione di supply chain complesse che coinvolgono numerosi fornitori.

- Limitazioni: Il punto debole sta nell'agente supervisore, che può diventare un collo di bottiglia o rappresentare un single point of failure se non funziona correttamente, rendendo il sistema meno adattabile a cambiamenti rapidi dell'ambiente operativo.

2. Architettura con Intervento Umano
Descrizione: Un sistema che integra agenti AI con la supervisione umana, ideale per garantire responsabilità e fiducia.

- Quando preferirla: È la scelta migliore in settori altamente regolamentati dove la responsabilità legale è imprescindibile, per decisioni ad alto rischio con significative implicazioni etiche, nelle fasi iniziali di adozione degli agenti AI in un'organizzazione o quando la fiducia degli utenti gioca un ruolo critico.
- Esempi di applicazione ideale: Funziona al meglio nei sistemi di supporto decisionale in ambito medico, nei processi di approvazione di prestiti o mutui e nei sistemi di monitoraggio della sicurezza pubblica.
- Limitazioni: La scalabilità è limitata dalla disponibilità di supervisori umani, la latenza nelle decisioni aumenta a causa dell'attesa dell'intervento umano e c'è il rischio di introdurre bias umani nel processo decisionale.

3. Architettura a Rete
Descrizione: Un approccio decentralizzato in cui gli agenti collaborano senza un controllo centrale, favorendo resilienza e adattabilità.

- Quando preferirla: Si adatta perfettamente a sistemi distribuiti geograficamente, quando serve alta resilienza senza un single point of failure, per problemi che traggono vantaggio da multiple prospettive o in ambienti dinamici che richiedono un adattamento continuo.
- Esempi di applicazione ideale: È ideale per sistemi di monitoraggio e gestione di smart grid energetiche, per l'ottimizzazione di reti logistiche e trasporti globali e per ecosistemi IoT complessi con sensori distribuiti.
- Limitazioni: La maggiore complessità di implementazione e debug può rappresentare una sfida, così come le possibili incongruenze nei dati tra agenti diversi e gli elevati consumi computazionali legati alla comunicazione tra agenti.

4. Architettura Sequenziale
Descrizione: Un flusso lineare in cui ogni agente elabora dati in sequenza, perfetto per processi strutturati.

- Quando preferirla: È indicata per processi lineari ben definiti con chiare dipendenze, quando i dati devono subire trasformazioni successive, nei casi in cui la tracciabilità del processo è fondamentale o per workflow in cui ogni fase richiede competenze specialistiche.
- Esempi di applicazione ideale: Si presta a processi di approvazione documentale, a pipeline di analisi dati o reportistica e a lavorazioni industriali sequenziali.
- Limitazioni: L'elevata dipendenza dall'affidabilità di ogni singolo agente la rende vulnerabile, con una bassa tolleranza ai guasti che può bloccare l'intera catena in caso di fallimento di un agente, e una difficoltà nell'adattarsi a requisiti variabili.

5. Architettura di Trasformazione dei Dati
Descrizione: Un sistema progettato per elaborare e trasformare grandi volumi di dati, ottimizzando l'input per analisi successive.

- Quando preferirla: È la scelta ottimale per l'elaborazione di grandi volumi di dati eterogenei, quando è necessario normalizzare o arricchire i dati prima dell'analisi, in contesti con requisiti di privacy o conformità normativa e per preparare dati destinati a machine learning o analisi avanzate.
- Esempi di applicazione ideale: Si applica bene all'elaborazione di dati sensibili nel rispetto della GDPR, al preprocessing di dati per analisi predittiva e all'integrazione di fonti dati eterogenee per business intelligence.
- Limitazioni: C'è il rischio di perdere informazioni durante le trasformazioni, la difficoltà nel tracciare la provenienza dei dati attraverso multiple trasformazioni e la necessità di aggiornamenti continui in caso di cambiamenti nelle fonti dati.

6. Architettura Loop
Descrizione: Un modello iterativo che ripete i processi per affinare i risultati, ideale per ottimizzazioni progressive.

- Quando preferirla: È perfetta per compiti che richiedono un miglioramento iterativo della qualità, in processi creativi o di ottimizzazione progressiva, quando serve un controllo qualità rigoroso o per problemi che necessitano di approssimazioni successive.
- Esempi di applicazione ideale: Si rivela utile nel debugging di codice software, nell'ottimizzazione di modelli matematici o simulazioni e nei processi di editing e miglioramento di contenuti creativi.
- Limitazioni: Esiste la possibilità di loop infiniti se i criteri di terminazione non sono ben definiti, accompagnata da elevati consumi di risorse computazionali per le iterazioni multiple e da tempi di completamento difficili da prevedere.

7. Architettura Parallela
Descrizione: Un sistema che esegue più compiti simultaneamente, massimizzando velocità e scalabilità.

- Quando preferirla: È ideale per problemi facilmente parallelizzabili con task indipendenti, quando la velocità di elaborazione è una priorità, in presenza di ampie risorse computazionali disponibili e per gestire picchi di carico o grandi volumi di richieste.
- Esempi di applicazione ideale: Funziona al meglio nell'analisi di grandi dataset in tempo reale, nell'elaborazione parallela di richieste utente in piattaforme ad alto traffico e nelle simulazioni scientifiche complesse.
- Limitazioni: La maggiore complessità nella sincronizzazione e aggregazione dei risultati può essere un ostacolo, insieme agli elevati requisiti di infrastruttura hardware e al rischio di incongruenze nei risultati tra agenti paralleli.

8. Architettura Router
Descrizione: Un approccio che instrada le richieste agli agenti più adatti, ottimizzando l'uso delle risorse.

- Quando preferirla: Si adatta a sistemi con tipologie di richieste molto diverse tra loro, quando è necessario instradare efficacemente le query agli agenti più adatti, in contesti con carico di lavoro variabile e picchi di traffico o per ottimizzare l'uso di risorse specializzate.
- Esempi di applicazione ideale: È perfetta per sistemi di supporto clienti multicanale, per piattaforme di servizi cloud con varietà di API e per servizi di risposta a query in domini diversi, sfruttando competenze multi-expertise.
- Limitazioni: La dipendenza critica dalla qualità dell'algoritmo di routing è un punto debole, insieme alla complessità nella gestione delle code e dei carichi di lavoro e alla necessità di un monitoraggio costante per evitare instradamenti errati.

Considerazioni per la Scelta dell'Architettura Ottimale

1. Valutazione dei requisiti di business:
    - Quali sono gli obiettivi principali del sistema? (velocità, accuratezza, trasparenza)
    - Quali vincoli normativi o di compliance esistono?
    - Quale livello di supervisione umana è accettabile o necessario?
2. Analisi delle risorse disponibili:
    - Quale infrastruttura tecnologica è disponibile?
    - Quali competenze tecniche possiede il team?
    - Qual è il budget per implementazione e manutenzione?
3. Considerazioni di escalabilità:
    - Come si prevede che crescerà il sistema nel tempo?
    - Quali volumi di dati e richieste dovrà gestire?
    - Con quali altri sistemi dovrà integrarsi?
4. Approccio ibrido: In molti casi reali, l'approccio più efficace è combinare elementi di diverse architetture in un sistema ibrido. Ad esempio:
    - Un'architettura gerarchica con elementi di intervento umano per decisioni critiche
    - Un sistema a rete con router centralizzato per ottimizzare il routing delle richieste
    - Una pipeline sequenziale con elaborazione parallela in alcune fasi computazionalmente intensive

Ogni architettura è valutata su una scala da 1 a 5, dove 1 rappresenta la peggiore prestazione e 5 la migliore

Architettura	**Scalabilità**	**Flessibilità**	**Resilienza**	**Complessità implementativa**	**Supervisione umana**	**Efficienza computazionale**	**Latenza decisionale**	**Coerenza output**
Gerarchica	Media (3/5)	Media (3/5)	Bassa (2/5)	Media (3/5)	Media (3/5)	Alta (4/5)	Media (3/5)	Alta (4/5)

Architettura	Scalabilità	Flessibilità	Resilienza	Complessità implementativa	Supervisione umana	Efficienza computazionale	Latenza decisionale	Coerenza output
Con intervento umano	Bassa (2/5)	Media (3/5)	Alta (4/5)	Bassa (2/5)	Alta (5/5)	Media (3/5)	Bassa (2/5)	Alta (5/5)
A rete	Alta (4/5)	Alta (4/5)	Alta (5/5)	Alta (4/5)	Bassa (2/5)	Bassa (2/5)	Media (3/5)	Media (3/5)
Sequenziale	Media (3/5)	Bassa (2/5)	Bassa (2/5)	Bassa (2/5)	Media (3/5)	Alta (4/5)	Bassa (2/5)	Alta (4/5)
Trasformazione dati	Alta (4/5)	Alta (4/5)	Media (3/5)	Media (3/5)	Bassa (2/5)	Media (3/5)	Media (3/5)	Alta (4/5)
Loop	Bassa (2/5)	Media (3/5)	Media (3/5)	Media (3/5)	Media (3/5)	Bassa (2/5)	Bassa (2/5)	Alta (5/5)
Parallela	Alta (5/5)	Media (3/5)	Alta (4/5)	Media (3/5)	Bassa (2/5)	Alta (5/5)	Alta (5/5)	Media (3/5)
Router	Alta (4/5)	Alta (5/5)	Alta (4/5)	Alta (4/5)	Bassa (2/5)	Alta (4/5)	Alta (4/5)	Media (3/5)

8. Conclusioni

Gli Agenti AI rappresentano una convergenza innovativa tra modelli linguistici di grandi dimensioni (LLM) e capacità di orchestrazione avanzata, progettati per automatizzare compiti complessi con un approccio che va oltre la semplice esecuzione meccanica. Grazie alla loro abilità di interpretare dati in modo semantico e di adattarsi dinamicamente attraverso l'uso di prompt, questi sistemi offrono una gestione flessibile e contestualizzata delle informazioni. Ad esempio, un agente AI potrebbe analizzare report finanziari, estrarre insight rilevanti e suggerire strategie di investimento, il tutto rispondendo a istruzioni naturali e personalizzate. Tuttavia, questa potenza non è priva di ostacoli: i protocolli di comunicazione tra agenti sono ancora immaturi, i costi energetici per il loro funzionamento restano elevati e la necessità di risorse computazionali avanzate può limitarne l'accesso. La vera forza degli Agenti AI emerge nella collaborazione multi-agente, dove i compiti vengono suddivisi tra entità specializzate – come un agente dedicato all'analisi dei dati e un altro alla sintesi dei risultati – migliorando sia l'efficienza che la precisione.

Il futuro degli Agenti AI sarà plasmato da tre pilastri fondamentali:

- **la standardizzazione**, che ne faciliterà l'integrazione e l'interoperabilità;
- **le ottimizzazioni computazionali**, essenziali per ridurre l'impatto economico e ambientale;
- **una governance etica solida**, indispensabile per garantire che l'autonomia crescente di questi sistemi non comprometta i principi di equità e responsabilità.

Se ben bilanciati, innovazione e sostenibilità possono trasformare gli Agenti AI in alleati preziosi per un progresso inclusivo, rendendo la digitalizzazione e l'automazione non solo strumenti di efficienza, ma anche opportunità condivise per aziende, comunità e individui. Immaginiamo, ad esempio, un agente AI che coordina una rete di sensori in una smart city, ottimizzando il consumo energetico e migliorando la qualità della vita senza lasciare indietro le aree meno sviluppate: questo è il potenziale che si può sbloccare con una visione strategica.

Per sfruttare al massimo le capacità degli Agenti AI, le aziende e i policy maker – intesi come individui o gruppi che influenzano la formulazione, l'implementazione e la valutazione delle politiche pubbliche a vari livelli di governo e in diversi settori – dovranno agire in modo coordinato e lungimirante.

I policy maker includono un'ampia gamma di attori: funzionari pubblici e amministratori che gestiscono le regolamentazioni, legislatori che definiscono le leggi, leader politici che orientano le priorità strategiche, esperti e consulenti che forniscono dati e analisi, rappresentanti di organizzazioni internazionali che promuovono standard globali, gruppi di interesse e lobby che rappresentano esigenze settoriali, oltre ad accademici e ricercatori che contribuiscono con studi e innovazioni. Questi soggetti giocano un ruolo cruciale nel garantire che l'adozione degli Agenti AI avvenga in modo responsabile, equo e sostenibile, plasmando un ecosistema tecnologico che non amplifichi le disuguaglianze esistenti, ma le mitighi.

Azioni strategiche per le Aziende e I Policy Maker

Ecco le azioni strategiche che aziende e policy maker dovrebbero intraprendere per massimizzare il potenziale degli Agenti AI:

1. ### Investire in Formazione e Competenza

   Promuovere programmi di formazione mirati è essenziale per dotare professionisti e aziende delle competenze necessarie a sfruttare gli Agenti AI. Discipline come il prompt engineering – l'arte di progettare istruzioni efficaci per gli LLM – e la gestione avanzata di questi modelli permettono di adattare rapidamente gli agenti a scenari diversi, da quello industriale a quello creativo. Ad esempio, un programma di upskilling potrebbe insegnare a un team di marketing come usare un agente AI per generare campagne personalizzate, aumentando il ritorno sull'investimento e la reattività al mercato.

2. ### Adottare Soluzioni Open-Source

   Incoraggiare l'uso di LLM open-source può abbattere le barriere economiche e tecniche, democratizzando l'accesso alla tecnologia. Questo approccio non solo riduce i costi per le

piccole e medie imprese (PMI) e le startup, ma stimola anche l'innovazione diffusa, consentendo a sviluppatori indipendenti di personalizzare soluzioni per esigenze locali. Pensiamo a una startup che utilizza un modello open-source per creare un assistente virtuale multilingue per mercati emergenti, un'opportunità che sarebbe stata impensabile con soluzioni proprietarie costose.

3. ### Promuovere la Standardizzazione
   La collaborazione con consorzi industriali e progetti open-source per definire protocolli di comunicazione standardizzati è un passo cruciale verso l'interoperabilità degli Agenti AI. Standard condivisi permetterebbero, ad esempio, a un agente logistico di dialogare senza problemi con un agente di gestione inventario, migliorando l'efficienza complessiva dei sistemi. Questo approccio non solo semplifica l'integrazione tecnologica, ma accelera l'adozione su larga scala, rendendo gli agenti più plug-and-play in contesti diversi.

4. ### Ottimizzare le Risorse Computazionali
   Investire in tecniche come la quantizzazione, il pruning e i modelli distillati è fondamentale per ridurre i costi computazionali ed energetici degli Agenti AI. Queste ottimizzazioni permettono di comprimere modelli complessi senza sacrificarne le prestazioni, rendendoli eseguibili su hardware meno potente. Ad esempio, un agente ottimizzato potrebbe gestire l'analisi di dati in tempo reale su un dispositivo edge in una fabbrica, abbassando l'impronta carbonica e i costi operativi, un aspetto critico per la sostenibilità a lungo termine.

5. ### Implementare Governance Etica
   Sviluppare framework di governance chiari è indispensabile per garantire un uso responsabile degli Agenti AI. Questo significa promuovere la trasparenza nelle decisioni – ad esempio, spiegando come un agente approva un prestito – e prevenire bias che potrebbero penalizzare gruppi specifici, oltre a definire responsabilità precise in caso di errori o danni. Un caso pratico potrebbe essere un agente diagnostico in sanità: senza una governance etica, un errore di valutazione potrebbe avere conseguenze gravi, e la colpa resterebbe ambigua tra sviluppatore, operatore e sistema.

6. ### Favorire l'Inclusività
   Promuovere politiche che garantiscano un accesso equo alle tecnologie AI è vitale per evitare che le differenze di efficienza tra grandi compagnie e realtà più piccole amplifichino le disuguaglianze economiche. Iniziative come finanziamenti pubblici per PMI o la creazione di "AI commons" – risorse condivise come dataset e modelli pre-addestrati – possono livellare il campo di gioco. Un esempio potrebbe essere un programma che fornisce a cooperative agricole strumenti AI per ottimizzare la produzione, riducendo il divario con i grandi produttori.

Agendo su questi fronti, aziende e policy maker possono trasformare gli Agenti AI in strumenti di progresso condiviso, non solo per i giganti tecnologici, ma per l'intera società. Bilanciando innovazione, sostenibilità e inclusività, sarà possibile costruire un futuro in cui l'automazione non sia sinonimo di esclusione, ma di opportunità diffuse. La chiave sta nell'agire ora, con visione e responsabilità, per plasmare una tecnologia che rifletta i valori di equità e progresso che vogliamo vedere nel mondo di domani.

9. Fonti e possibili approfondimenti

Il panorama dei **S**istemi **M**ulti-**A**gente (MAS) è ampio e in continua evoluzione. Per approfondire i concetti trattati nei capitoli precedenti e orientarsi tra architetture, pattern decisionali e applicazioni reali, si propone una raccolta selezionata di articoli, white paper e pubblicazioni tecniche. Le fonti riportate offrono una panoramica completa: si spazia dalle linee guida progettuali alla classificazione dei pattern architetturali, dalle sfide specifiche dei MAS basati su LLM fino agli impieghi industriali concreti in settori come la finanza, la manifattura e la sanità. Questi materiali rappresentano un punto di partenza utile per chi desidera consolidare le proprie conoscenze e approfondire l'applicazione dei MAS in contesti reali o sperimentali.

Titolo	Link	Contenuto
Building effective agents	https://www.anthropic.com/engineering/building-effective-agents	Linee guida per progettare agenti efficaci: semplicità architetturale, robustezza, riduzione delle dipendenze e attenzione al controllo umano nei task critici.
Challenges in Multi-Agent Systems: Navigating Complexity in Distributed AI	https://smythos.com/ai-agents/multi-agent-systems/challenges-in-multi-agent-systems/#:~:text=Transparency%20is%20another%20crucial%20element,world%20applications	Esamina le principali sfide nei MAS: trasparenza, sincronizzazione, coerenza dell'output e difficoltà nel debug di sistemi distribuiti complessi.
Choosing the Right Agentic Architecture for Your System	https://okareo.com/blog/posts/agentic-architecture	Presenta una metodologia decisionale per selezionare l'architettura agentica ideale in base a requisiti come supervisione, efficienza e trasparenza.
Multi-Agent System Patterns in Financial Services: Architectures for	https://community.aws/content/2uDxjoo105xRO6Q7mfkogmOYTVp/multi-agent-system-patterns-in-financial-services-architectures-for-next-generation-ai-solutions	Descrive l'uso di architetture multi-agente nella finanza, evidenziando pattern ibridi, scalabilità, resilienza e routing per il rilevamento frodi e l'automazione dei pagamenti.

Next-Generation AI Solutions		
What is agentic architecture?	https://www.ibm.com/think/topics/agentic-architecture#:~:text=	Definisce l'architettura agentica e le sue applicazioni aziendali. Spiega l'evoluzione da workflow statici a sistemi intelligenti e dinamici
LLM Multi-Agent Systems: Challenges and Open Problems	https://ar5iv.labs.arxiv.org/html/2402.03578#:~:text=Hierarchical%20Structure,the%20sequential%20nature%20of%20decision	Survey tecnica sui problemi aperti nei sistemi multi-agente basati su LLM, inclusi design gerarchici, sequenze decisionali e coordinamento adattivo.
Exploring Agentic Workflow Patterns	https://dev.to/dpaluy/exploring-agentic-workflow-patterns-312a#:~:text=Description%3A%20Agents%20iterate%20over%20a,improve	Esamina vari pattern architetturali agentici, come sequenziale, iterativo, parallelo, evidenziandone applicazioni pratiche e strategie di progettazione flessibile.
Patterns in multi-agent systems	https://www.linkedin.com/pulse/patterns-multi-agent-systems-atul-yadav-x5uue/	Discussione pratica dei pattern multi-agente comuni (gerarchico, peer-to-peer, centralizzato) e loro impatti su coordinazione, latenza e resilienza.
Exploring the Applications of Multi-Agent Systems in Real-World Scenarios	https://smythos.com/ai-agents/multi-agent-systems/applications-of-multi-agent-systems/#:~:text=Multi,These%20interconnected%20systems%20are%20transforming	Esamina le principali sfide nei MAS: trasparenza, sincronizzazione, coerenza dell'output e difficoltà nel debug di sistemi distribuiti complessi.
Framework di agenti AI: scegliere la base giusta per la tua azienda	https://www.ibm.com/it-it/think/insights/top-ai-agent-frameworks#:~:text=Da%20un%20singolo%20agente%20di,framework%20per%20gli%20agenti%20AI	Guida alla scelta dei migliori framework di sviluppo per agenti AI. Include panoramiche su LangChain, ReAct, AutoGPT e altri modelli emergenti.
How multi-agent systems transform modern manufacturing efficiency	https://www.byteplus.com/en/topic/495450?title=how-multi-agent-systems-transform-modern-manufacturing-efficiency	Analizza come i MAS migliorano l'efficienza produttiva nelle fabbriche intelligenti, automatizzando il coordinamento tra macchine, linee di assemblaggio e manutenzione predittiva.
Large Language Model-Enabled Multi-Agent Manufacturing Systems	https://arxiv.org/html/2406.01893v2#:~:text=Large%20Language%20Model,manufacturing%2C%20making%20them%20more%20adaptable	Esplora l'impiego di LLM nei MAS per la manifattura. Evidenzia l'adattabilità e la comunicazione tra agenti LLM per ottimizzare la produzione.