RapidStart

WHITEPAPER

Distribuire OpenClaw in azienda

Dall'agente locale a un'infrastruttura aziendale sicura, affidabile e gestibile.

Architettura · Distribuzione · Sicurezza
Governance · Esercizio

RAPIDSTART · AGOSTO 2026

SINTESI

Sintesi esecutiva

OpenClaw è interessante perché è insolitamente aperto. Offre un runtime per agenti self-hosted senza vincolare l'organizzazione a un unico cloud, a un unico fornitore di modelli, a un unico sistema di identità o a un unico modello operativo. Quella flessibilità è il punto di forza — ed è anche l'origine dell'onere di distribuzione.

Nell'uso aziendale il lavoro va rapidamente oltre l'installazione. L'organizzazione deve predisporre un host duraturo, definire i confini di identità e credenziali, collegare modelli e sistemi aziendali, limitare ciò che gli agenti possono fare, difendersi da minacce specifiche degli agenti come il prompt injection e l'autonomia eccessiva, e istituire monitoraggio, aggiornamenti e ripristino.

La conclusione è pratica: le organizzazioni che scelgono OpenClaw dovrebbero considerare l'ambiente circostante parte integrante del sistema. Un buon modello di distribuzione rende esplicite le decisioni di infrastruttura, sicurezza ed esercizio, automatizza ciò che può essere ripetuto, documenta ciò che resta specifico del carico di lavoro e rende l'ambiente ricostruibile.

> **OpenClaw fornisce il runtime. La maturità per la produzione nasce dall'architettura, dai controlli e dalle pratiche operative che lo circondano.**

Quattro conclusioni

01 **OpenClaw convince** quando la proprietà del runtime, la flessibilità sui modelli e l'estensibilità contano più di un modello operativo SaaS completamente gestito.

02 **Le piattaforme di agenti gestite sono valide alternative** quando l'organizzazione vuole che sia un fornitore a farsi carico di gran parte della piattaforma, del runtime e dell'onere operativo.

03 **La maturità per la produzione dipende** almeno tanto da identità, permessi, difese dal prompt injection, osservabilità e ripristino quanto dal runtime dell'agente in sé.

04 **Il vero traguardo non è** “abbiamo OpenClaw in funzione”. È “sappiamo distribuirlo, governarlo, gestirlo e ricostruirlo in modo prevedibile”.

01 Perché OpenClaw conta per le aziende

Il cambiamento importante nell'IA aziendale non è semplicemente che i modelli rispondono a domande migliori. È che gli agenti possono restare disponibili, usare strumenti, mantenere il contesto e compiere azioni su più sistemi. OpenClaw riunisce in un unico runtime un gateway self-hosted, canali, sessioni, memoria, strumenti, skill e routing multi-agente.[1]

BENEFICIO PER L'AZIENDA	PERCHÉ CONTA
Agenti sempre attivi	Un runtime persistente può operare attorno a un processo o a un team, invece di dipendere da un singolo dipendente che apre una finestra di chat.
Azione sui sistemi aziendali	Gli strumenti e le connessioni MCP consentono all'agente di passare dal rispondere a domande al leggere e agire su CRM, ERP, Microsoft 365 e API.
Flessibilità sui modelli	Il runtime può essere configurato su più fornitori di modelli e con schemi di failover, invece di fare di un singolo modello l'applicazione stessa.[1]
Controllo del runtime	L'organizzazione controlla dove risiede il runtime, come viene esteso e quanta parte dell'architettura circostante possiede.

Agenti costruiti attorno al lavoro

Gran parte delle aziende dispone già di informazioni in abbondanza; l'attrito sta nel comporle tra i sistemi e poi eseguire il passo successivo. Un agente può diventare il livello connettivo che raccoglie il contesto, ragiona su di esso e usa strumenti per far avanzare il lavoro. È cosa diversa dal dare semplicemente a ogni dipendente un chatbot personale.

> **Per gli agenti aziendali di lunga durata, il modello può essere trattato sempre più come infrastruttura sottostante all'agente, non come l'applicazione stessa.**

02 Dove si colloca OpenClaw

Le alternative a OpenClaw non sono concorrenti diretti: si collocano a livelli diversi. La domanda utile è quale modello operativo si adatti meglio al lavoro, ai requisiti di controllo, alle esigenze di integrazione e alla propensione ingegneristica dell'organizzazione.

OPZIONE	CATEGORIA	PUNTI DI FORZA	CONTESTO IDEALE
Copilot Studio	Piattaforma di agenti gestita, low-code	Authoring, connettori, governance e pubblicazione nativi Microsoft	Quando l'organizzazione vuole una piattaforma gestita e una proprietà minima del runtime.[2]
Microsoft Foundry Agent Service	Hosting gestito di agenti	Agenti basati su prompt e ospitati, con endpoint gestiti, scalabilità, identità e osservabilità	Quando serve codice personalizzato ma si preferiscono hosting gestito da Microsoft e controlli enterprise.[3]
OpenClaw	Runtime di agenti self-hosted	Proprietà del runtime, canali, strumenti, skill, flessibilità sui modelli, architettura aperta	Quando l'organizzazione vuole possedere il runtime e le scelte architetturali circostanti.[1]
LangGraph / CrewAI	Framework per sviluppatori	Massimo controllo su applicazioni di agenti scritte in codice e sulla loro orchestrazione	Quando il team costruisce un'applicazione di agenti su misura anziché distribuire un runtime.
Zapier / simili	Automazione IA gestita	Connettività SaaS rapida e automazione dei flussi di lavoro	Quando l'ampiezza dell'automazione SaaS conta più della proprietà del runtime.

Piattaforme gestite: quando conviene possedere meno runtime

A volte una piattaforma enterprise gestita è la scelta migliore. Copilot Studio è forte quando l'authoring low-code, l'esercizio gestito da Microsoft e la governance nativa sono requisiti primari. OpenClaw è più convincente quando l'organizzazione dà valore alla proprietà del runtime, a un'ampia estensibilità, alla flessibilità sui modelli e alla possibilità di plasmare l'architettura circostante. La scelta non riguarda semplicemente quale prodotto sia migliore: riguarda quale modello operativo l'organizzazione voglia possedere.

Hosting gestito e framework per sviluppatori

I servizi di hosting gestito per agenti sono utili quando un team di sviluppo vuole codice personalizzato mentre il fornitore gestisce gran parte del livello di hosting, identità e osservabilità. I framework per sviluppatori come LangGraph e CrewAI si collocano più in basso nello stack e sono preferibili quando il team intende costruire da sé l'applicazione dell'agente. OpenClaw si distingue come runtime self-hosted già esistente, con un proprio gateway, canali, skill e modello operativo.[3][4][5]

03 Quando OpenClaw è la scelta sbagliata

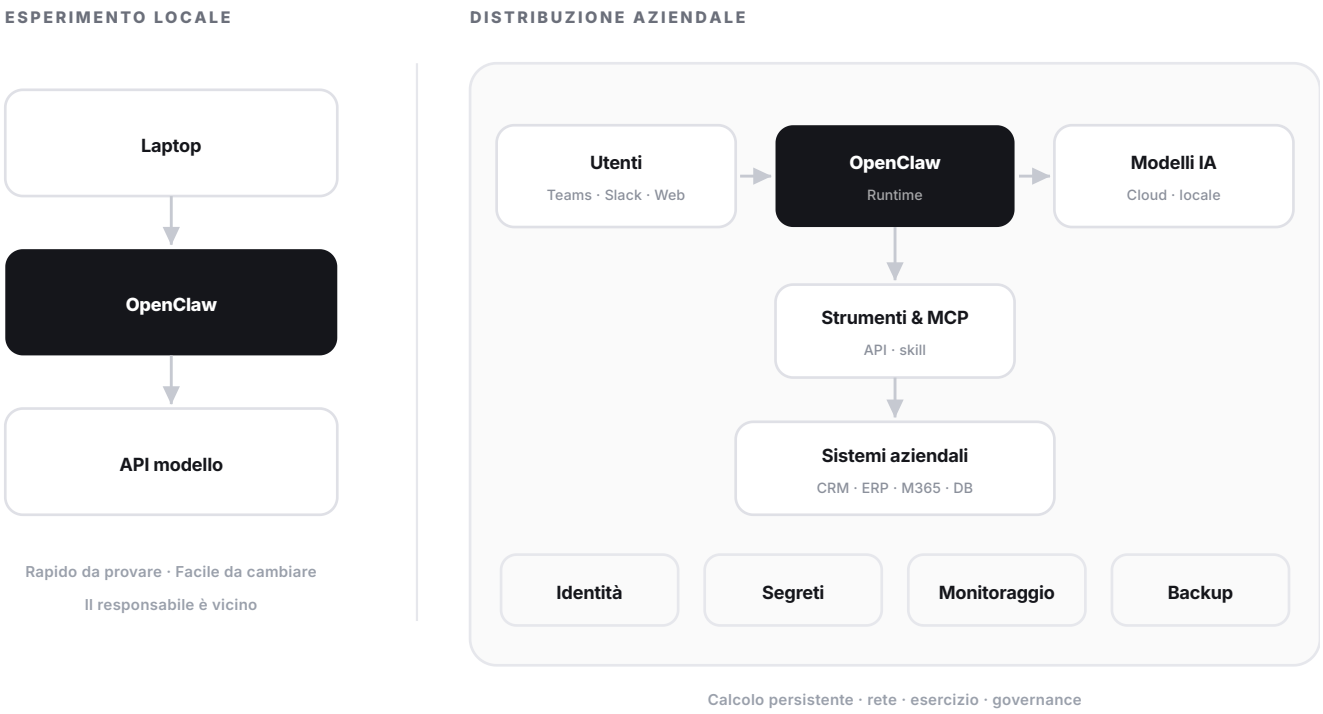
Una decisione architetturale credibile include un criterio di esclusione. OpenClaw non è la risposta migliore solo perché è aperto o perché il runtime è potente.

CRITERIO DI ESCLUSIONE	RISPOSTA MIGLIORE
Serve un runtime gestito dal fornitore, con SLA	Una piattaforma gestita come Copilot Studio o Foundry Agent Service può essere più adatta sul piano operativo.
Il problema è l'automazione deterministica di flussi di lavoro	Power Automate, Logic Apps, Zapier o un altro strumento di automazione possono essere più semplici e più facili da governare.
Nessuno è responsabile dell'esercizio del cloud	Un runtime self-hosted comporta responsabilità di patching, monitoraggio, ripristino e sicurezza. Se nessuno se ne fa carico, è meglio non creare quell'obbligo.
Requisiti di controllo regolamentari stringenti che il team non è in grado di soddisfare	Scegliere una piattaforma i cui controlli gestiti, certificazioni, modello di supporto e superfici di audit corrispondano al requisito.
L'organizzazione non ha bisogno di possedere il runtime né di flessibilità sui modelli	I vantaggi principali di OpenClaw potrebbero non giustificare l'onere operativo.
Il caso d'uso è un singolo assistente circoscritto	Un agente gestito o una funzionalità applicativa integrata può risultare sensibilmente più semplice di un runtime generico.

> **L'open source non è una strategia. Il self-hosting ha valore solo quando il controllo che offre vale la responsabilità che crea.**

04 Dall'agente locale all'infrastruttura aziendale

OpenClaw rende deliberatamente semplice la prima esperienza. È un'ottima cosa per lo sviluppo e la sperimentazione. Una distribuzione aziendale parte un livello prima: l'organizzazione ha bisogno anzitutto di un ambiente duraturo in cui il runtime possa vivere.



Una VM cloud è spesso lo schema più semplice, ma il calcolo persistente introduce subito scelte su regione, sistema operativo, storage, amministrazione, comportamento al riavvio, patching, monitoraggio e ripristino.

> **La domanda dello sviluppatore è "Riesco a farlo funzionare?". La domanda dell'azienda è "Riusciamo a creare un ambiente in cui funzioni in modo affidabile, continuo e sotto controlli noti?"**

05 Una baseline di distribuzione in produzione

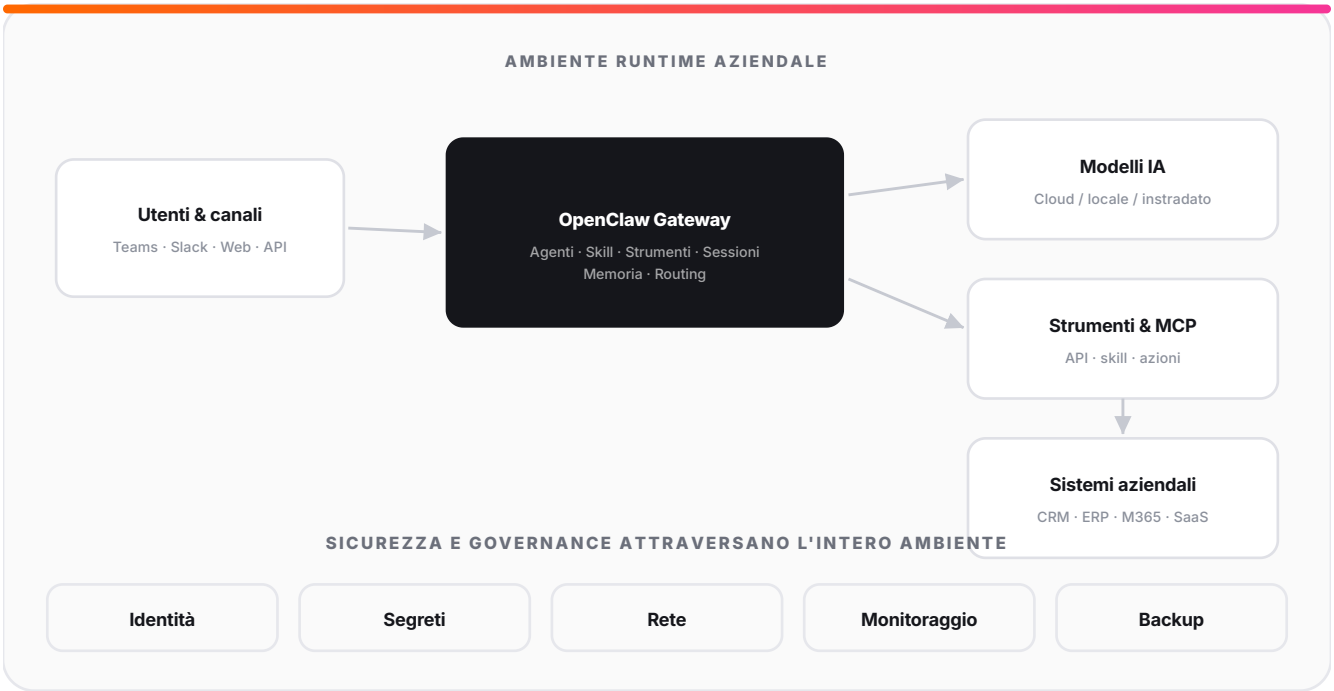
Una guida utile alla distribuzione non dovrebbe fermarsi alle domande. Questa baseline raccoglie i controlli e le scelte operative che di norma andrebbero compiuti in modo deliberato prima di considerare un ambiente OpenClaw come infrastruttura aziendale. I prodotti e i fornitori specifici possono variare; i requisiti no.

DECISIONE	BASELINE CONSIGLIATA	PERCHÉ
Host del runtime	Calcolo persistente di proprietà dell'organizzazione	Mantiene il runtime indipendente dal dispositivo di un dipendente e dà all'organizzazione un confine operativo definito.
Identità organizzativa	Identità del carico di lavoro dedicata, dove supportata[13]	Evita di ancorare l'accesso di produzione alle credenziali personali di chi ha installato l'agente.
Accesso alle risorse	Accesso basato sull'identità dove la piattaforma lo supporta[13]	Riduce la dipendenza da segreti di lunga durata e migliora revoca e tracciabilità.
Segreti	Archiviazione gestita dei segreti; nessun file di segreti di produzione in locale sulle postazioni degli sviluppatori[15]	Tratta token e chiavi come asset gestiti, con controlli di accesso, rotazione e procedure di ripristino.
Accesso ai modelli	Fornitori e modelli approvati, con politiche documentate su regione, quote e fallback[10][12]	Rende la scelta del modello una politica operativa anziché una preferenza dello sviluppatore.
Superficie di interazione umana	Canali aziendali approvati e adeguati al caso d'uso	Mantiene l'accesso all'agente entro superfici di comunicazione che l'organizzazione può governare.
Esercizio	Stato di salute, log e alerting centralizzati, più procedure documentate di riavvio e aggiornamento	Rende visibili i guasti e offre agli operatori una risposta ripetibile.
Distribuzione	Configurazione scriptata e ripetibile, anziché VM assemblate a mano come esemplari unici[14]	Rende realistici la ricostruzione, la distribuzione di una seconda istanza e il ripristino.

Questi sono principi architetturali di base, non una politica universale. Topologia di rete, permessi specifici, retention, soglie di approvazione, obiettivi di ripristino e fornitori di servizi approvati dipendono comunque dal carico di lavoro e dall'organizzazione.

06 Architettura di riferimento per OpenClaw in azienda

Il diagramma è uno schema di riferimento, non un'architettura imposta da OpenClaw.



Il confine del runtime

L'ambiente circostante — identità, segreti, rete, monitoraggio e ripristino — è ciò che trasforma un runtime flessibile in un sistema governabile. L'identità stabilisce chi o che cosa è l'agente. I segreti stabiliscono quali credenziali può usare. La rete definisce che cosa può raggiungere. Il monitoraggio determina se gli operatori sono in grado di vedere i guasti. Backup e ripristino determinano se l'ambiente può essere riportato in funzione.

OpenClaw lascia all'organizzazione la libertà di plasmare quel confine; un modello di distribuzione Microsoft compie in anticipo un sottoinsieme definito di quelle scelte.

07 Identità, segreti e rete

IDENTITÀ

Rendere l'agente un attore dell'organizzazione

Non ancorare un agente di produzione alle credenziali personali del dipendente che lo ha installato. Usare un'identità del carico di lavoro dedicata e organizzativa dove la piattaforma di destinazione lo consente. Per alcuni sistemi esterni possono restare necessarie credenziali applicative, ma devono essere eccezioni gestite, con titolarità chiara del ciclo di vita, privilegio minimo e procedure di revoca.

SEGRETI

Eliminare prima i segreti non necessari

Il segreto migliore è quello di cui la distribuzione non ha bisogno. L'accesso basato sull'identità può ridurre il numero di credenziali in chiaro da conservare. Dove i segreti restano necessari, vanno collocati in un archivio gestito, con accesso limitato all'identità del runtime e procedure documentate di rotazione e risposta agli incidenti. In produzione, evitare credenziali scritte nel codice e file di segreti locali agli sviluppatori.

RETE

La raggiungibilità è una politica

Un progetto di produzione dovrebbe definire in modo deliberato l'amministrazione in ingresso e la raggiungibilità in uscita, invece di ereditare i comodi valori predefiniti dello sviluppo. È preferibile non avere percorsi pubblici in ingresso non necessari; usare una via amministrativa controllata, come connettività privata, un servizio bastion, una VPN o equivalente. L'accesso in uscita dovrebbe essere limitato agli endpoint dei modelli, ai canali, alle sorgenti dei pacchetti e ai sistemi aziendali di cui il carico di lavoro ha effettivamente bisogno.

> **L'identità definisce chi è l'agente. I segreti definiscono quali porte può aprire. La rete definisce quali porte può raggiungere.**

08 L'integrazione con i sistemi aziendali è dove valore e rischio si incontrano

Il valore di OpenClaw cresce nettamente quando l'agente può operare su CRM, ERP, suite di collaborazione, database, API e server MCP. Cresce altrettanto la conseguenza di una decisione sbagliata. La disponibilità di uno strumento non è una politica.

CAPACITÀ	ESEMPIO	CONTROLLO CONSIGLIATO
Lettura	Recuperare il contesto di clienti, ordini o documenti	Privilegio minimo; l'ambito dei dati segue l'esigenza dell'utente o del processo.
Analisi	Riassumere, classificare, confrontare, rilevare eccezioni	Conservare il contesto di origine ed evitare di trattare l'interpretazione generata come dato autorevole privo di provenienza.
Raccomandazione	Redigere una risposta o proporre l'azione successiva	Definire quando è richiesta una revisione umana prima dell'esecuzione.
Scrittura	Aggiornare il CRM o creare un'attività	Limitare oggetti e campi scrivibili e usare un'identità attribuibile.
Esecuzione	Attivare un'API, un flusso di lavoro o un processo	Definire precondizioni, limiti, idempotenza e gate di approvazione.
Azione esterna	Inviare email, pubblicare all'esterno, modificare uno stato visibile al cliente	Richiedere approvazioni più stringenti, evidenze di audit e capacità di revoca rapida.

> **L'obiettivo non è la massima autonomia ovunque. L'obiettivo è l'autonomia giusta per il lavoro da svolgere.**

09 Minacce specifiche degli agenti: prompt injection e confused deputy

I soli permessi non rendono sicuro un agente. Un agente può essere legittimamente autorizzato a usare uno strumento e nondimeno essere manipolato perché usi quell'autorità in modo scorretto. Il NIST tratta il prompt injection diretto e indiretto come rischi per i sistemi di IA generativa, comprese le istruzioni nascoste in contenuti che un'applicazione integrata con un LLM recupera in seguito.[7] OWASP colloca Prompt Injection ed Excessive Agency tra i principali rischi delle applicazioni LLM.[8]

MINACCIA	COME SI MANIFESTA	SCHEMA DI CONTROLLO
Prompt injection diretto	Un utente istruisce l'agente a ignorare le politiche o a usare impropriamente uno strumento	Applicazione delle politiche a livello di sistema, strumenti in allowlist, identità con ambito ristretto, gate di approvazione per le azioni rilevanti.
Prompt injection indiretto	Istruzioni malevole nascoste in una pagina web, un documento, un'email o dati recuperati	Trattare i contenuti recuperati come dati non attendibili; separare i contenuti dalle istruzioni di controllo; limitare le chiamate agli strumenti e il traffico in uscita.
Comportamento da confused deputy	L'agente dispone di un'autorità legittima, ma una fonte non attendibile lo induce a esercitarla per lo scopo sbagliato	Legare le azioni a un richiedente o a un processo autenticato, validare intento e contesto, richiedere approvazione quando l'impatto è rilevante.
Autonomia eccessiva	L'agente dispone di più strumenti, ambito o autonomia di quanto il compito richieda	Privilegio minimo, schemi degli strumenti ristretti, confini espliciti di scrittura ed esecuzione, credenziali a vita breve dove possibile.
Gestione impropria dell'output	L'output generato viene passato direttamente a codice, SQL, shell, HTML o a un altro sistema	Validare e sanificare gli output; usare interfacce degli strumenti tipizzate anziché l'esecuzione di comandi in forma libera, dove possibile.

> **Un'architettura di agenti sicura deve governare sia ciò che l'agente è autorizzato a fare, sia quali informazioni possono influenzare quella decisione.**

10 Modelli, affidabilità ed esercizio

Strategia sui modelli

Un utente locale può scegliere il proprio modello preferito. Un'azienda ha bisogno di una politica: fornitori approvati, area geografica, condizioni sulla privacy, latenza, quote, comportamento di fallback e controlli di costo. Microsoft Foundry può fornire un livello governato di accesso ai modelli mentre OpenClaw resta il runtime.[3][10][11] Compiti diversi possono giustificare modelli diversi; il processo aziendale non dovrebbe essere cablato a un'unica generazione di modelli.

Affidabilità e osservabilità

Quando un processo dipende da un agente, gli operatori devono sapere se il gateway è in salute, se i canali sono connessi, se l'accesso ai modelli sta fallendo, se le credenziali sono scadute e quali azioni rilevanti l'agente ha tentato. OpenClaw offre diagnostica del gateway e probe sui canali, ma il modello operativo circostante richiede comunque titolarità, alerting e procedure di risposta.[6]

Dimensionamento e costi: impegnarsi sui driver, non su una precisione fittizia

DRIVER DI COSTO	INDICAZIONI PRATICHE
VM / calcolo	CPU e memoria dipendono dai canali, dalla concorrenza, dagli strumenti locali e dal fatto che i modelli girino in remoto o in locale. Partire da una VM general purpose contenuta e collaudare sotto carico il carico di lavoro reale prima di scalare.
Utilizzo dei modelli	Di solito è il costo variabile dominante quando i modelli sono remoti. Tracciare token e richieste per agente e per attività, impostare quote e instradare il lavoro semplice verso modelli meno costosi dove opportuno.
Storage / log	Stato delle conversazioni, log e artefatti si accumulano. Definire la retention e monitorare la crescita dello storage, invece di considerare la VM come infinita.
Esercizio	Il costo nascosto è la titolarità umana: patching, risposta agli incidenti, test degli aggiornamenti e revisione degli accessi. L'automazione riduce questo costo più di quanto non faccia limare una piccola quota della spesa per la VM.

I prezzi esatti non sono riportati intenzionalmente, perché regione cloud, tipo di calcolo e tariffe dei modelli cambiano. Il modello di distribuzione dovrebbe rendere quelle variabili osservabili e sostituibili, non nasconderle.

11 Matrice di maturità per la produzione

È un punto di controllo interno concreto. Un'azienda non ha bisogno di controlli identici per ogni carico di lavoro, ma le decisioni devono essere esplicite.

CAPACITÀ	SPERIMENTAZIONE	DISTRIBUZIONE AZIENDALE	POSTURA PRONTA PER LA PRODUZIONE
Calcolo	Laptop / ad hoc	Host persistente	Dimensionamento, titolarità, riavvio e ripristino noti
Identità	Credenziali personali	Schema di identità dedicata	Privilegio minimo, ciclo di vita, tracciabilità
Segreti	Ambiente / configurazione	Archiviazione protetta	In vault e ruotabili; con accesso controllato
Rete	Impostazioni predefinite comode	Ingresso / uscita limitati	Percorso amministrativo documentato e segmentazione
Modelli	Modello preferito	Fornitore / modello approvato	Regione, quote, fallback, politica di costo
Integrazioni	Token degli sviluppatori	Accesso aziendale con ambito definito	Confini espliciti di lettura / scrittura / esecuzione
Difese dalle injection	Di norma assenti	Guardrail su prompt e strumenti	Gestione dei contenuti non attendibili, controlli di approvazione e sul traffico in uscita
Monitoraggio	Ispezione manuale	Controlli di stato / log	Alert, titolarità, procedure di risposta
Aggiornamenti	Ad hoc	Pianificati	Disciplina di test, rollback e rilascio
Ripristino	Di norma assente	Backup di VM / configurazione	Procedura di ricostruzione e ripristino collaudata
Documentazione	Appunti personali	Registro della distribuzione	Definizione riproducibile dell'ambiente

> **Un agente in esecuzione è un traguardo tecnico. Un agente governabile è una capacità aziendale.**

12 Ciclo di vita e riproducibilità sono un unico argomento

Un ambiente di produzione non si conclude con l'installazione. Va provisionato, messo in sicurezza, configurato, collegato, validato, gestito, aggiornato e prima o poi ricostruito.

Quei passaggi del ciclo di vita diventano affidabili solo quando le scelte di distribuzione vengono documentate, invece di restare nella memoria di un singolo tecnico.

La prova della ricostruzione

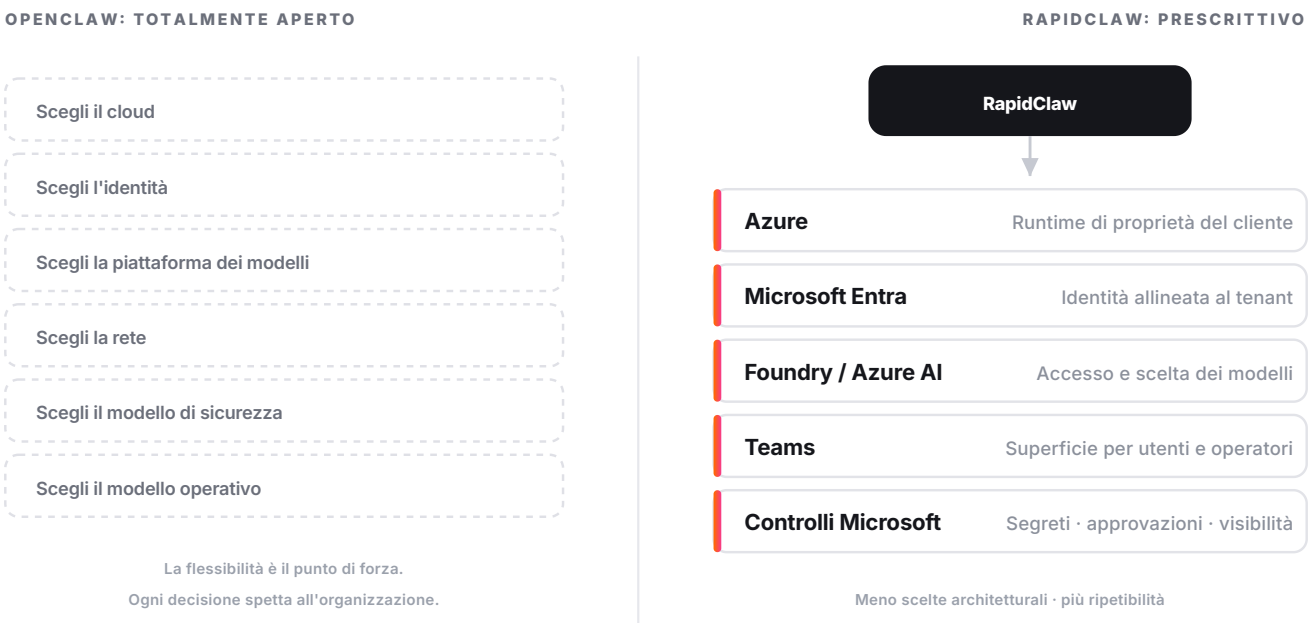
Se la VM sparisse stanotte, un altro amministratore riuscirebbe a ricreare l'ambiente domani? L'azienda potrebbe distribuire una seconda istanza per un altro team o un'altra regione? Saprebbe spiegare quale configurazione è intenzionale e quale impostazione è un residuo storico? Infrastructure-as-code, configurazione scriptata, registri di distribuzione e percorsi di aggiornamento controllati trasformano una macchina che funziona in una capacità ripetibile.

> **Una VM che funziona è un'istanza. Un ambiente riproducibile è una capacità.**

13 Da OpenClaw a RapidClaw

Tutto quanto detto finora vale a prescindere dal cloud o dal fornitore. OpenClaw lascia deliberatamente aperta l'architettura circostante. RapidClaw comincia dove quell'apertura diventa un onere di distribuzione per le organizzazioni standardizzate su Microsoft.

RapidClaw non è un fork di OpenClaw. OpenClaw resta il runtime. RapidClaw è un modello prescrittivo di distribuzione ed esercizio costruito attorno a esso per le organizzazioni Microsoft. In pratica significa che una distribuzione guidata mette in piedi l'intero ambiente dentro il tenant Azure del cliente in circa venti minuti, invece di una VM assemblata a mano la cui configurazione resta in capo a chi l'ha costruita.[9]



AMBITO DECISIONALE	SCELTA DI RAPIDCLAW
Confine cloud	Runtime Azure di proprietà del cliente[9][14]
Piano dell'identità	Identità organizzativa allineata a Microsoft Entra[9][13]
Piattaforma dei modelli	Percorso Microsoft Foundry / Azure AI per impostazione predefinita[3][10][12]
Superficie di interazione umana	Microsoft Teams dove appropriato
Controlli	Schemi allineati a Microsoft per segreti, accessi, visibilità e governance[9][11][15]
Distribuzione	Configurazione guidata e ripetibile anziché un assemblaggio manuale una tantum[9][14]

14 Che cosa cambia RapidClaw — e che cosa lascia aperto

RapidClaw è deliberatamente stretto su una dimensione e aperto su un'altra. Standardizza le scelte di distribuzione Microsoft che non dovrebbero essere reinventate ogni volta, lasciando OpenClaw aperto agli agenti, alle skill, agli strumenti e alle connessioni ai sistemi aziendali adatti al carico di lavoro.

CONFINE	SIGNIFICATO
RapidClaw standardizza	Percorso di distribuzione su Azure; integrazione con l'identità Microsoft; percorso dei modelli Foundry / Azure AI; superficie di integrazione Teams; configurazione e modello operativo ripetibili.
Il cliente continua a decidere	Quali agenti costruire; quali sistemi aziendali e server MCP esporre; i permessi di lettura, scrittura ed esecuzione; la politica di approvazione; la retention; i controlli per i carichi di lavoro regolamentati; la titolarità aziendale.
RapidClaw non fa	Creare un fork di OpenClaw o sostituirlo; rendere di prima classe ogni possibile architettura di cloud e runtime; eliminare la necessità di una governance aziendale.

> **Il valore di una distribuzione prescrittiva non sta nell'eliminare ogni decisione. Elimina le decisioni che le organizzazioni Microsoft non dovrebbero dover prendere da zero ogni volta.**

Conclusione

L'architettura totalmente aperta di OpenClaw è gran parte della sua attrattiva. La parte difficile inizia quando quella flessibilità deve diventare infrastruttura aziendale affidabile: calcolo persistente, identità, segreti, rete, politiche sui modelli, confini di integrazione, difese dalle injection, osservabilità, aggiornamenti e ripristino.

Per le organizzazioni che già operano su Microsoft, RapidClaw è la nostra risposta a questo divario di distribuzione: mantenere OpenClaw come runtime, compiere una volta sola le scelte architetturali Microsoft e trasformarle in un modello operativo ripetibile.

RIFERIMENTI

[1] Documentazione OpenClaw, docs.openclaw.ai, consultata ad agosto 2026.
docs.openclaw.ai

[2] Microsoft Learn, "What is Microsoft Copilot Studio?", aggiornato il 3 agosto 2026.
learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-what-is-copilot-studio

[3] Microsoft Learn, "What is Microsoft Foundry Agent Service?", aggiornato il 19 agosto 2026.
learn.microsoft.com/en-us/azure/foundry/agents/overview

[4] Documentazione di LangGraph, LangChain, consultata ad agosto 2026.
docs.langchain.com/oss/python/langgraph/overview

[5] Documentazione di CrewAI, consultata ad agosto 2026.
docs.crewai.com/en/introduction

[6] Documentazione OpenClaw, runbook e diagnostica del gateway, consultata ad agosto 2026.
docs.openclaw.ai/gateway

[7] NIST AI 600-1, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile", luglio 2024, §2.9 Information Security.
nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[8] "OWASP Top 10 for LLM Applications 2026" — Prompt Injection ed Excessive Agency, OWASP GenAI Security Project, agosto 2026.
genai.owasp.org/resource/owasp-genai-llm-top-10-2026

[9] RapidStart, "RapidClaw for OpenClaw — Governed AI in Azure", consultato ad agosto 2026.
www.rapidstart.com/en/products/rapidclaw-for-openclaw

[10] Microsoft Learn, "Built-in policies for model deployment in Microsoft Foundry portal", aggiornato il 18 agosto 2026.
learn.microsoft.com/en-us/azure/foundry/how-to/model-deployment-policy

[11] Microsoft Learn, "Role-based access control for Microsoft Foundry", aggiornato il 3 luglio 2026.
learn.microsoft.com/en-us/azure/foundry/concepts/rbac-foundry

[12] Microsoft Learn, "Deployment types for Microsoft Foundry Models", aggiornato il 6 agosto 2026.
learn.microsoft.com/en-us/azure/foundry/foundry-models/concepts/deployment-types

[13] Microsoft Learn, "Managed identities for Azure resources", documentazione di Microsoft Entra.
learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/overview

[14] Microsoft Learn, "What is an Azure landing zone?", Cloud Adoption Framework, aggiornato il 26 agosto 2026.
learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone

[15] Microsoft Learn, "Understanding autorotation in Azure Key Vault", aggiornato il 10 aprile 2026.
learn.microsoft.com/en-us/azure/key-vault/general/autorotation

OpenClaw è il runtime. RapidClaw è il modello Microsoft di distribuzione che lo circonda.

Chi è RapidStart

RapidStart realizza applicazioni aziendali e infrastrutture per agenti Microsoft-first, progettate per rendere più semplici la distribuzione, l'esercizio e l'adozione di tecnologie aziendali avanzate.