RapidStart

BALTOJI KNYGA

# OpenClaw diegimas verslui

Nuo vietinio agento iki saugios, patikimos ir lengvai valdomos verslo infrastruktūros.

RAPIDSTART · 2026 M. RUGPJŪTIS

Architektūra · Diegimas · Sauga
Valdymas · Eksploatavimas

# Santrauka

OpenClaw yra patrauklus, nes yra neįprastai atviras. Jis suteikia savarankiškai talpinamą agento vykdymo aplinką, tačiau neįstumia organizacijos į vieną debesį, vieną modelio tiekėją, vieną tapatybės sistemą ar vieną veiklos modelį. Tokia lankstumas yra savybė — ir diegimo naštos šaltinis.

Verslui naudojant darbas greitai išeina už įdiegimo ribų. Organizacija turi sukurti patikimą šeimininką, apibrėžti tapatybės ir kredencialų ribas, prijungti modelius ir verslo sistemas, apriboti, ką agentai gali daryti, apsisaugoti nuo agentams būdingų grėsmių, tokių kaip užklausų injekcija ir perteklinė savarankiškumas, bei nustatyti stebėjimą, atnaujinimus ir atkūrimą.

Išvada yra praktiška: organizacijos, kurios pasirenka OpenClaw, turėtų traktuoti aplinkinę aplinką kaip sistemos dalį. Pagrįstas diegimo modelis aiškiai perkelia infrastruktūros, saugumo ir veiklos sprendimus, automatizuoja tai, ką galima pakartoti, dokumentuoja tai, kas lieka priklausoma nuo darbo krūvio, ir padaro aplinką atkuriamą.

**OpenClaw suteikia vykdymo aplinką. Paruošimas produkcijai priklauso nuo architektūros, kontrolės priemonių ir veiklos praktikų aplink ją.**

## Keturi pastebėjimai

01 OpenClaw yra patrauklus, kai svarbiau valdyti vykdymo aplinką, turėti modelių lankstumą ir išplėtimą nei visiškai valdomas SaaS veiklos modelis.

02 **Valdomos agentų platformos yra stipri alternatyva, kai organizacija nori, kad tiekėjas** valdytų didesnę platformos, vykdymo ir operacinę naštą.

03 Parengimas gamybai priklauso bent tiek pat nuo tapatybės, leidimų, apsaugos nuo užklausų injekcijos, stebimumo ir atkūrimo, kiek nuo paties agento vykdymo aplinkos.

04 Tikrasis pasiekimas nėra „mes paleidome OpenClaw.“ Tikrasis pasiekimas yra „mes galime jį diegti, valdyti, eksploatuoti ir prognozuojamai atkurti.“

# 01 Kodėl OpenClaw yra svarbus verslui

Svarbus pokytis versle su AI nėra vien tai, kad modeliai geriau atsako į klausimus. Svarbu, kad agentai gali likti pasiekiami, naudoti įrankius, palaikyti kontekstą ir atlikti veiksmus sistemose. OpenClaw sujungia savarankiškai talpinamą vartų sistemą, kanalus, sesijas, atmintį, įrankius, įgūdžius ir daugiagentinį maršrutizavimą vienoje vykdymo aplinkoje.[1]

| VERSLO NAUDA | KODĖL IT YRA SVARBUS |
|---|---|
| **Visada veikiantys agentai** | Nuolatinė vykdymo aplinka gali veikti aplink procesą ar komandą, o ne priklausyti nuo to, kad vienas darbuotojas atidarytų pokalbių langą. |
| **Veiksmai verslo sistemose** | Įrankiai ir MCP jungtys leidžia agentui pereiti nuo klausimų atsakymo prie skaitymo ir veiksmų CRM, ERP, Microsoft 365 ir API ribose. |
| **Modelių lankstumas** | Vykdymo aplinka gali būti sukonfigūruota keliems modelių tiekėjams ir persiuntimo (failover) scenarijams, o ne padaryti vieną modelį pačia programa.[1] |
| **Vykdymo aplinkos kontrolė** | Organizacija kontroliuoja, kur vykdymo aplinka gyvena, kaip ji plečiama ir kiek aplinkinės architektūros ji valdo. |

## Agentai, sukurti aplink darbą

Dauguma įmonių jau turi pakankamai informacijos; trukdis — ją surinkti iš skirtingų sistemų ir tada atlikti kitą žingsnį. Agentas gali tapti jungiamuoju sluoksniu, kuris surenka kontekstą, mąsto apie jį ir naudoja įrankius darbui pažengti. Tai skiriasi nuo paprasto asmeninio pokalbių roboto suteikimo kiekvienam darbuotojui.

> **Ilgalaikiams verslo agentams modelis vis labiau gali būti traktuojamas kaip infrastruktūra po agentu — o ne pats programos sluoksnis.**

# 02 Kur telpa OpenClaw

Alternatyvos aplink OpenClaw nėra tiesioginiai lygiaverčiai; jos yra skirtinguose sluoksniuose. Naudingas klausimas — kuris veiklos modelis geriausiai atitinka organizacijos darbą, valdymo reikalavimus, integracijos poreikius ir inžinerinę nuovoką.

| PARINKTIS | KATEGORIJA | KUR IT YRA STIPRUS | GERIAUSIA IT |
|---|---|---|---|
| **Copilot Studio** | Valdomas low-code agentų platforma | Microsoft-natūralus kūrimas, jungtys, valdymas ir publikavimas | Kai organizacija nori valdomos platformos ir minimalios vykdymo aplinkos nuosavybės.[2] |
| **Microsoft Foundry Agent Service** | Valdomas agentų talpinimas | Patarimai ir talpinami agentai su valdomais galiniais taškais, masteliavimu, tapatybe ir stebimumu | Kai reikalingas pritaikytas kodas, bet pageidaujamas Microsoft valdomas talpinimas ir įmonės kontrolė. [3] |
| **OpenClaw** | Savarankiškai talpinamas agentų vykdymo modulis | Vykdymo nuosavybė, kanalai, įrankiai, įgūdžiai, modelio lankstumas, atvira architektūra | Kai organizacija nori valdyti vykdymo aplinką ir su tuo susijusius architektūros pasirinkimus.[1] |
| **LangGraph / CrewAI** | Kūrėjų karkasai | Maksimalus valdymas per koduotus agentų sprendimus ir orkestraciją | Kai komanda kuria individualų agento sprendimą, o ne diegia paruoštą vykdymo aplinką. |
| **Zapier / panašūs** | Valdomas AI automatizavimas | Greita SaaS ryšio ir darbo srautų automatizacija | Kai svarbiau SaaS automatizacijos aprėptis nei vykdymo aplinkos nuosavybė. |

## Valdomos platformos: kai mažesnė vykdymo aplinkos nuosavybė yra pranašesnė

Kartais valdomas įmonės platformos sprendimas yra geresnis pasirinkimas. Copilot Studio yra stiprus, kai pagrindiniai reikalavimai yra mažo kodo kūrimas, Microsoft valdomos operacijos ir gimtoji valdymo sistema. OpenClaw yra patrauklesnė, kai organizacija vertina vykdymo aplinkos nuosavybę, platų išplėtimą, modelių lankstumą ir galimybę formuoti aplinkinę architektūrą. Pasirinkimas nėra vien tik kuris produktas geresnis; tai kuris veiklos modelis organizacija nori prisiimti.

## Valdomas talpinimas ir kūrėjų karkasai

Valdomos agentų talpinimo paslaugos yra naudingos, kai kūrėjų komanda nori pritaikyto kodo, o tiekėjas operuoja didesne talpinimo, tapatybės ir stebimumo sluoksniu. Kūrėjų karkasai, tokie kaip LangGraph ir CrewAI, yra žemesni sluoksnyje ir tinkamesni, kai komanda ketina kurti agento programą pati. OpenClaw skiriasi kaip esama savarankiškai talpinama vykdymo aplinka su savo vartais, kanalais, įgūdžiais ir veiklos modeliu.[3][4][5]

# 03 Kada OpenClaw yra neteisingas pasirinkimas

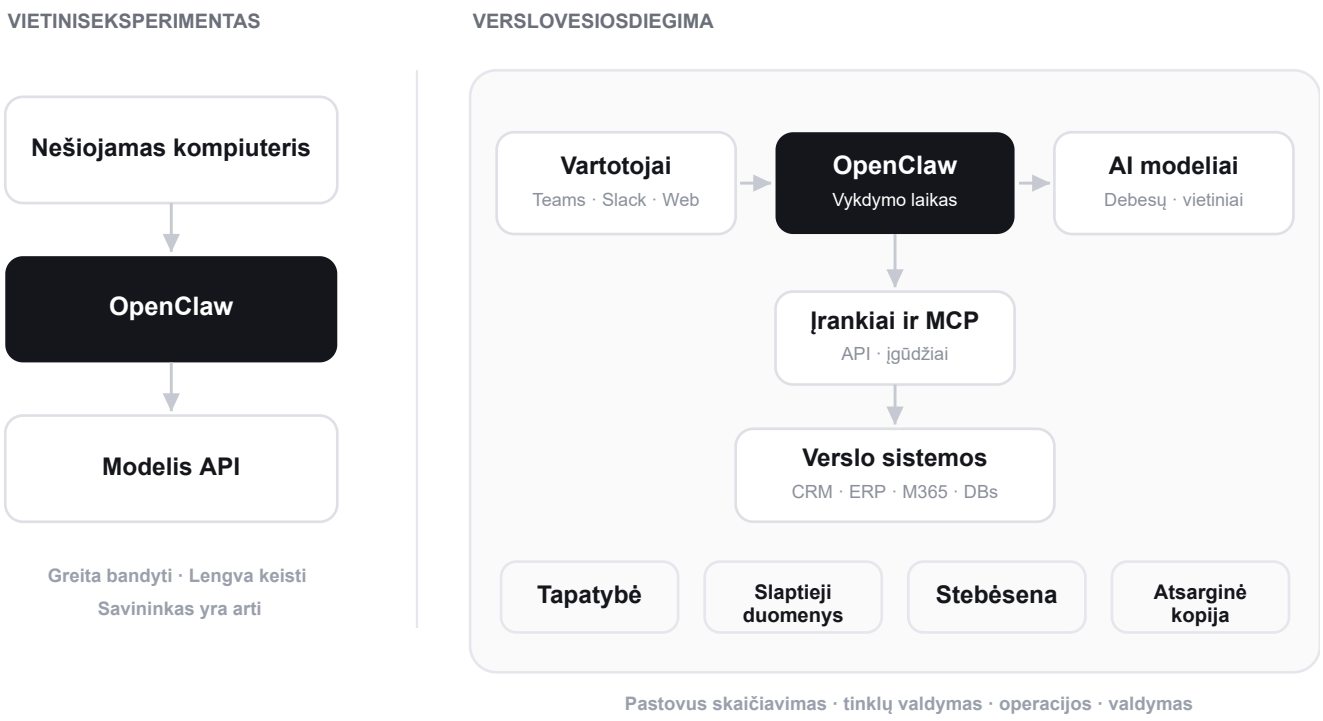
Patikimas architektūros sprendimas įtraukia ir diskvalifikatorių. OpenClaw nėra geriausias atsakymas vien todėl, kad jis atviras arba jo vykdymo aplinka yra galinga.

| DISKVALIFIKATORIUS | GERESNIS SPRENDIMAS |
|---|---|
| **Jūs norite tiekėjo valdomos vykdymo aplinkos ir SLA** | Valdoma platforma, tokia kaip Copilot Studio arba Foundry Agent Service, gali būti geresnis operacinis sprendimas. |
| **Problema yra deterministinis darbo eigos automatizavimas** | Power Automate, Logic Apps, Zapier arba kitas darbo eigos įrankis gali būti paprastesnis ir lengviau valdomas. |
| **Niekas neatsako už debesijos operacijas** | Savarankiškai talpinama vykdymo aplinka sukuria pataisų, stebėjimo, atkūrimo ir saugumo atsakomybes. Jei niekas jų neprisiima, nesukurkite tokios prievolės. |
| **Griežti reguliavimo kontrolės reikalavimai, kurių jūsų komanda negali tenkinti** | Pasirinkite platformą, kurios valdomos kontrolės, sertifikatai, palaikymo modelis ir audito paviršiai atitinka reikalavimą. |
| **Organizacija nereikalauja vykdymo aplinkos nuosavybės ar modelių lankstumo** | Pagrindinės OpenClaw pranašumai gali neatsverti eksploatavimo naštos. |
| **Naudojimo atvejis yra vienas siauras asistentas** | Valdomas agentas arba įterpta programos funkcija gali būti reikšmingai paprastesnė nei bendros paskirties vykdymo aplinka. |

> **Atvirasis kodas nėra strategija. Savarankiškas talpinimas yra vertingas tik tada, kai kontrolė, kurią jis suteikia, verta atsakomybės, kurią jis sukuria.**

# 04 Nuo vietinio agento iki įmonės infrastruktūros

OpenClaw tyčia palengvina pirmąją patirtį. Tai puiku kūrimui ir eksperimentams. Verslo diegimas prasideda viena sluoksiu anksčiau: organizacijai pirmiausia reikia patvarios aplinkos, kurioje vykdymo aplinka gali gyventi.




Debesies VM dažnai yra paprasčiausias modelis, tačiau nuolatinis skaičiavimas iš karto įveda pasirinkimus dėl regiono, operacinės sistemos, saugyklos, administravimo, paleidimo elgesio, pataisų, stebėjimo ir atkūrimo.

> **Kūrėjo klausimas yra „Ar aš galiu tai paleisti?“ Verslo klausimas yra „Ar galime sukurti aplinką, kurioje tai veikia patikimai, nuolat ir pagal nustatytas kontrolės sąlygas?“**

# 05 Produkcijos diegimo pagrindas

Naudingas diegimo vadovas neturėtų apsiriboti klausimais. Ši bazė fiksuoja valdiklius ir veiklos pasirinkimus, kurie paprastai turi būti priimti apsvarstytai prieš traktujant OpenClaw aplinką kaip verslo infrastruktūrą. Tikslios prekės ir tiekėjai gali skirtis; reikalavimai — ne.

| SPRENDIMAS | REKOMENDUOJAMAS PAMATAS | KODĖL |
|---|---|---|
| **Vykdymo hostas** | Pastovas, organizacijos nuosavybės skaičiavimas | Laiko vykdymą nepriklausomai nuo darbuotojo įrenginio ir suteikia organizacijai apibrėžtą veikimo ribą. |
| **Organizacinė tapatybė** | Skirta darbo krūvio tapatybė, kai tai palaikoma[13] | Vengia gamybos prieigos pririšimo prie asmeninių prisijungimų to, kuris įdiegė agentą. |
| **Išteklių prieiga** | Prieiga, pagrįsta tapatybe, kai platforma tai palaiko[13] | Sumažina priklausomybę nuo ilgalaikių slaptumų ir pagerina atšaukimo galimybes bei audituojamumą. |
| **Slaptieji duomenys** | Valdomas slaptumų saugojimas; jokių gamintojo lokalių gamybos slaptumo failų[15] | Elgiasi su žetonais ir raktai kaip su valdomais ištekliais, turinčiais prieigos kontrolę, keitimo ir atkūrimo procedūras. |
| **Modelio prieiga** | Patvirtinti tiekėjai ir modeliai su dokumentuota regiono, kvotos ir atsarginės politikos aprašymais[10][12] | Paverčia modelio pasirinkimą operacine politika, o ne kūrėjo preferencija. |
| **Žmogiškoji sąsaja** | Patvirtinti verslo kanalai, tinkami naudojimo atvejui | Laiko agentų prieigą komunikacijos paviršiuose, kuriais organizacija gali valdyti. |
| **Operacijos** | Centralizuota sveikatos būklės, žurnalo ir įspėjimų sistema bei dokumentuotos perkrovimo ir atnaujinimo procedūros | Padaro gedimus matomus ir suteikia operatoriams pakartojamą reagavimo tvarką. |
| **Diegimas** | Skriptinė, kartojama konfigūracija vietoj rankomis sukurtų unikalių VM[14] | Padaro atkūrimą, antros instancijos diegimą ir atkūrimą realiais. |

Tai yra baziniai architektūros principai, o ne visuotinė politika. Tinklo topologija, specifiniai leidimai, saugojimo terminai, patvirtinimo ribos, atkūrimo tikslai ir patvirtinti paslaugų teikėjai vis tiek priklauso nuo darbo krūvio ir organizacijos.

# 06 Pavyzdinė OpenClaw architektūra verslui

Diagramoje pateiktas pavyzdinis modelis, o ne privaloma OpenClaw architektūra.




## Veikimo lauko riba

Aplinkinė aplinka — tapatybė, paslaptys, tinklo ryšiai, stebėjimas ir atkūrimas — yra tai, kur verslas paverčia lankstų vykdymo aplinką valdomą sistema. Tapatybė nulemia, kas arba kas yra agentas. Paslaptys nulemia, kokius kredencialus jis gali naudoti. Tinklai apibrėžia, ką jis gali pasiekti. Stebėjimas nustato, ar operatoriai mato gedimus. Atsarginė kopija ir atkūrimas nulemia, ar aplinka gali būti atkurta.

OpenClaw suteikia organizacijai laisvę formuoti tą ribą; Microsoft diegimo modelis iš anksto apibrėžia tam tikrą tų pasirinkimų poaibį.

# 07 Tapatybė, slaptieji duomenys ir tinklai

TAPATYBĖ

## Padarykite agentą organizacijos veikėju

Neprisegkite gamybos agento prie asmeninių sąskaitų duomenų, kuriuos naudojo įrengusį darbuotoją. Naudokite specialią organizacinę darbo krūvio tapatybę, kai taikinio platforma tai palaiko. Programėlės kredencialai vis tiek gali būti būtini kai kurioms išorinėms sistemoms, tačiau jie turėtų būti valdomi kaip išimtys su aiškia gyvenimo ciklo atsakomybe, mažiausio privilegijų principu ir atšaukimo procedūromis.

SLAPTUKAI

## Pirmiausia pašalinkite nereikalingus slaptukus

Geriausias slaptumas yra tas, kurio diegimui nereikia. Tapatybė pagrįstas prieigos valdymas gali sumažinti saugotinų žaliavinių kredencialų skaičių. Ten, kur slaptukai lieka būtini, laikykite juos valdomame slaptumų saugykloje, apribokite prieigą pagal vykdymo tapatybę ir nustatykite dokumentuotas rotacijos bei incidentų reagavimo procedūras. Venkite kietai įkoduotų kredencialų ir kūrėjo vietinių slaptumų failų gamybinėje aplinkoje.

TINKLERYS

## Pasiekiamumas yra politika

Gamybinis dizainas turėtų sąmoningai apibrėžti įeinamą administravimą ir išeinamą pasiekiamumą, o ne paveldėti patogias kūrimo numatytąsias reikšmes. Teikite pirmenybę nebūtinam viešam įeinamam keliui; naudokite kontroliuojamą administracinį maršrutą, pvz., privačią jungtį, bastiono paslaugą, VPN ar lygiavertį sprendimą. Išeinamoji prieiga turėtų būti apribota iki modelių galinių taškų, kanalų, paketų šaltinių ir verslo sistemų, kurių darbo krūvis iš tikrųjų reikia.

> **Tapatybė apibrėžia, kas yra agentas. Slaptukai apibrėžia, kuriuos vartus jis gali atrakinti. Tinklų ryšiai apibrėžia, kurių vartų jis gali pasiekti.**

# 08 Verslo sistemų integracija — tai, kur susitinka nauda ir rizika

OpenClaw vertė smarkiai kyla, kai agentas gali dirbti per CRM, ERP, bendradarbiavimo paketus, duomenų bazes, API ir MCP serverius. Taip pat auga ir neteisingo sprendimo pasekmė. Įrankių prieinamumas nėra politika.

| GALIMYBĖ | PAVYZDYS | REKOMENDUOJAMA KONTROLĖ |
|---|---|---|
| **Skaityti** | Gauti kliento, užsakymo arba dokumento kontekstą | Mažiausios privilegijos; duomenų apimtis seka vartotojo arba proceso poreikį. |
| **Analizuoti** | Santraukuoti, klasifikuoti, palyginti, aptikti išimtis | Išlaikyti šaltinio kontekstą ir vengti sugeneruotos interpretacijos traktavimo kaip autoritetingų duomenų be kilmės įrodymo. |
| **Rekomenduoti** | Parengti atsakymą arba pasiūlyti kitą veiksmą | Apibrėžti, kada prieš vykdymą reikalingas žmogaus peržiūra. |
| **Rašyti** | Atnaujinti CRM arba sukurti užduotį | Apriboti rašomus objektus ir laukus bei naudoti priskiriamą tapatybę. |
| **Vykdyti** | Paleisti API, darbo eigą arba procesą | Apibrėžti prielaidas, ribas, idempotentiškumą ir patvirtinimo vartus. |
| **Išorinė akcija** | Siųsti el. laišką, skelbti išorėje, keisti su klientu susijusią būseną | Reikalauti griežtesnio patvirtinimo, audito įrodymų ir greitos atšaukimo galimybės. |

> **Tikslas nėra visur suteikti maksimalų autonomiškumą. Tikslas — suteikti tinkamą autonomiją konkrečiam darbui.**

# 09 Agentui būdingi grėsmės veiksniai: injekcija ir sutrikdytasis įgaliotinis

Vien privilegijos nepadaro agente saugaus. Agentas gali turėti teisėtą leidimą naudoti įrankį ir vis tiek būti manipuliuojamas netinkamai panaudoti tą autoritetą. NIST aptaria tiesioginę ir netiesioginę užklausų injekciją kaip generatyvių AI sistemų rizikas, įskaitant atakas, įterptas į turinį, kurį vėliau gauna LLM integruota programa.[7] OWASP identifikuoja Prompt Injection ir Excessive Agency tarp pagrindinių LLM taikomųjų programų rizikų.[8]

| GRĖSMĖ | KAIP IT ATRODO | VALDYMO ŠABLONAS |
|---|---|---|
| **Tiesioginė užklausos injekcija** | Vartotojas nurodo agentui ignoruoti politiką arba netinkamai naudoti įrankį | Sisteminio lygmens politikos vykdymas, leidžiamų įrankių sąrašas, apribotos tapatybės, patvirtinimo vartai reikšmingiems veiksmams. |
| **Netiesioginė užklausos injekcija** | Piktavališki nurodymai paslėpti interneto puslapyje, dokumente, el. laiške arba atkurtuose duomenyse | Laikykite atkurtą turinį kaip nepatikimus duomenis; atskirkite turinį nuo valdymo nurodymų; suvaržykite įrankių iškvietimus ir eigos išėjimą. |
| **Suklaidintas atstovo elgesys** | Agentas turi teisėtą įgaliojimą, tačiau nepatikimas šaltinis verčia jį panaudoti šį įgaliojimą netinkamam tikslui | Pririškite veiksmus prie autentifikuoto užklausėjo arba proceso, patvirtinkite ketinimą ir kontekstą, reikalaukite patvirtinimo, kai poveikis yra esminis. |
| **Per didelė autonomija** | Agentas turi daugiau įrankių, aprėpties arba autonomijos, nei reikalauja užduotis | Mažiausios privilegijos, siauri įrankių schemos, aiškios rašymo ir vykdymo ribos, trumpalaikiai kredencialai, kai įmanoma. |
| **Neteisingas išvesties tvarkymas** | Sugeneruota išvestis perduodama tiesiogiai į kodą, SQL, apvalkalą, HTML arba kitą sistemą | Patikrinkite ir išvalykite išvestis; kiek įmanoma, naudokite tipizuotus įrankių sąsajų tipus, o ne laisvos formos komandų vykdymą. |

> **Saugaus agento architektūra turi reguliuoti tiek tai, ką agentui leidžiama daryti, tiek kokia informacija gali įtakoti tą sprendimą.**

# 10 Modeliai, patikimumas ir operacijos

## Modelių strategija

Vietinis vartotojas gali pasirinkti mėgstamą modelį. Verslui reikia politikos: patvirtintų tiekėjų, geografijos, privatumo sąlygų, vėlinimo, kvotų, atsarginio elgesio ir kaštų valdymo. Microsoft Foundry gali suteikti valdomą modelių prieigos sluoksnį, kol OpenClaw lieka vykdymo laiku.[3]

[10][11] Skirtingos užduotys gali pateisinti skirtingus modelius; verslo procesas neturėtų būti griežtai pririštas prie vienos modelio kartos.

## Patikimumas ir stebėsenos galimybės

Kai procesas priklauso nuo agente, operatoriams reikia žinoti, ar vartai veikia, ar kanalai prijungti, ar modelių prieiga nepavyksta, ar kredencialai nepasibaigė ir kokių reikšmingų veiksmų agentas bandė atlikti. OpenClaw teikia vartų diagnostiką ir kanalų sondas, bet supantis valdymo modelis vis tiek reikalauja atsakomybės, įspėjimų ir reagavimo procedūrų.[6]

## Mastelio parinkimas ir sąnaudos: įsipareigokite pagrindiniams veiksniams, o ne apgaulingam tikslumui

| SĄNAUDŲ VEIKĖJAS | PRAKTINĖS RODO SMIOS |
|---|---|
| **VM / skaičiavimai** | CPU ir atmintis priklauso nuo kanalų, konkurencijos, vietinių įrankių ir to, ar modeliai veikia nuotoliniu būdu ar vietoje. Pradėkite nuo kuklaus universalios paskirties VM ir ištestuokite esamą darbo krūvį prieš plečiant. |
| **Modelių naudojimas** | Dažniausiai vyraujanti kintamoji kaina, kai modeliai yra nuotoliniai. Sekite žetonus ir užklausas pagal agentą ir užduotį, nustatykite kvotas ir nukreipkite paprastus darbus į pigesnius modelius, kai tinkama. |
| **Saugykla / žurnalai** | Pokalbių būsena, žurnalai ir artefaktai kaupiasi. Apibrėžkite saugojimo terminus ir stebėkite saugyklos augimą, o ne laikykite VM kaip begalinės. |
| **Operacijos** | Paslėptosios sąnaudos yra žmogiškoji atsakomybė: pataisymai, incidentų reagavimas, atnaujinimų testavimas ir prieigos peržiūros. Automatizacija sumažina šias sąnaudas labiau nei nedidelis VM išlaidų sumažinimas. |

Tikslios kainos specialiai čia neįvardijamos, nes debesies regionas, skaičiavimo tipas ir modelių įkainiai keičiasi. Diegimo šablonas turėtų padaryti tuos kintamuosius stebimais ir keičiamesiais, o ne juos slėpti.

# 11 Parengties eksploatuoti matrica

Tai praktiškas vidinis barjeras. Verslui nereikia vienodų valdiklių kiekvienam darbo krūviui, bet sprendimai turėtų būti aiškūs.

| GALIMYBĖ IT | EKSPERIMENTAS | VERSLO DIEGIMAS | PRODUKCIJAI PARUOŠTAS POŽIŪRIS |
|---|---|---|---|
| **Skaičiavimai** | Nešiojamas kompiuteris / ad hoc | Nuolatinis serveris | Žinomas dydis, nuosavybė, paleidimas ir atkūrimas |
| **Tapatybė** | Asmeniniai prisijungimo duomenys | Atskiras tapatybės modelis | Mažiausios teisės, gyvavimo ciklas, audituojamumas |
| **Slaptieji duomenys** | Aplinka / konfigūracija | Apsaugotas saugojimas | Saugomas seifuose ir keičiama; prieiga kontroliuojama |
| **Tinklas** | Patogūs numatytieji nustatymai | Ribotas įeinamasis / išeinamasis srautas | Dokumentuotas administratoriaus kelias ir segmentacija |
| **Modeliai** | Pageidaujamas modelis | Patvirtintas teikėjas / modelis | Regionas, kvota, atsarginis variantas, kaštų politika |
| **Integracijos** | Vystytojo žetonai | Apribota verslo prieiga | Aiškios skaitymo / rašymo / vykdymo ribos |
| **Injekcijos apsauga** | Dažniausiai nėra | Prašymas ir įrankių ribos | Nepatikimo turinio tvarkymas, patvirtinimas ir išeigos kontrolės |
| **Stebėsena** | Rankinė peržiūra | Sveikatos patikrinimai / žurnalai | Įspėjimai, atsakomybė, reagavimo procedūros |
| **Atnaujinimai** | Atsitiktiniai | Planuoti | Bandymų, atsitraukimo ir leidimo disciplina |
| **Atkūrimas** | Dažniausiai nėra | VM / konfigūracijos atsarginė kopija | Išbandyta perdarymo ir atkūrimo procedūra |
| **Dokumentacija** | Asmeniniai užrašai | Diegimo įrašas | Atkartojama aplinkos apibrėžtis |

> **Veikiantis agentas yra techninis pasiekimas. Veikiantis verslui agentas yra verslo galimybė.**

# 12 Gyvavimo ciklas ir atkartojamumas yra vienas argumentas

Gamybinė aplinka nėra baigta ją įdiegus. Ji turi būti paruošta, apsaugota, sukonfigūruota, prijungta, patvirtinta, eksploatuojama, atnaujinama ir galiausiai atstatoma.




Šie gyvenimo ciklo žingsniai tampa patikimi tik tada, kai diegimo sprendimai yra užfiksuoti, o ne gyvena vieno inžinieriaus atmintyje.

## Atkūrimo testas

Jei VM šį vakarą išnyktų, ar kitas administratorius galėtų atkurti aplinką rytoj? Ar įmonė galėtų įdiegti antrą instanciją kitai komandai ar regionui? Ar ji galėtų paaiškinti, kuri konfigūracija yra sąmoninga, o kuri — istorinis likutis? Infrastruktūra kaip kodas, skriptinė konfigūracija, diegimo įrašai ir kontroliuojami atnaujinimo keliai paverčia veikiančią mašiną kartojama galimybe.

**Veikianti VM yra instancija. Kartojama aplinka yra galimybė.**

# 13 Nuo OpenClaw iki RapidClaw

Viskas iki šio taško taikoma nepriklausomai nuo debesies ar tiekėjo. OpenClaw sąmoningai palieka supančią architektūrą atvirą. RapidClaw prasideda ten, kur ta atvirumas tampa diegimo našta organizacijoms, standartizuotoms pagal Microsoft.

RapidClaw nėra OpenClaw šakutė. OpenClaw lieka vykdymo laiku. RapidClaw yra nuomonėmis grįstas diegimo ir eksploatavimo modelis aplink jį Microsoft organizacijoms. Praktikoje tai reiškia, kad vadovaujamasis diegimas visą aplinką paleidžia kliento pačių Azure nuomininke per maždaug dvidešimt minučių, o ne kaip rankomis surinktą VM, kurio konfigūracija priklauso tam, kas jį surinko.[9]




| SPRENDIMO SRITIS | RAPIDCLAW NUOMONĖ |
|---|---|
| **Debesų riba** | Kliento valdomas Azure vykdymo laikas[9][14] |
| **Tapatybės sluoksnis** | Microsoft Entra suderinta organizacinė tapatybė[9][13] |
| **Modelių platforma** | Pagal numatytuosius nustatymus Microsoft Foundry / Azure AI kelias[3][10][12] |
| **Žmogiškoji sąsaja** | Microsoft Teams kai tinkama |
| **Valdikliai** | Microsoft suderintos paslaptys, prieiga, matomumas ir valdymo modeliai[9][11][15] |
| **Diegimas** | Vestuotas, kartojamas konfigūravimas, o ne vienkartinis rankinis surinkimas[9][14] |

# 14 Ką keičia RapidClaw — ir ką palieka atvirą

RapidClaw sąmoningai siauras viename matmenyje ir atviras kitame. Jis standartizuoja Microsoft diegimo pasirinkimus, kurių nereikėtų išrasti iš naujo, tuo pačiu palikdamas OpenClaw atvirą agentams, įgūdžiams, įrankiams ir verslo sistemų ryšiams, kurie atitinka darbo krūvį.

| RIBOS | PRASMĖ |
|---|---|
| **RapidClaw standartizuoja** | Azure diegimo kelias; Microsoft tapatybės integracija; Foundry / Azure AI modelio kelias; Teams integracijos paviršius; kartojamas nustatymas ir valdymo modelis. |
| **Klientas vis tiek sprendžia** | Kuriuos agentus kurti; kurias verslo sistemas ir MCP serverius atverti; skaitymo, rašymo ir vykdymo leidimai; patvirtinimo politika; saugojimas; reguliuojamo darbo krūvio kontrolės; verslo nuosavybė. |
| **Ko RapidClaw nedaro** | Nekuria OpenClaw atšakos ir jo nepakeičia; nesiekia vienodai palaikyti visų debesijos ir vykdymo aplinkų architektūrų; nepanaikina verslo valdymo poreikio. |

> **Nuomonės turinčio diegimo vertė nėra ta, kad jis panaikina visas sprendimo priėmimo situacijas. Jis panaikina sprendimus, kurių Microsoft organizacijoms nereikėtų kiekvieną kartą nuo nulio priimti.**

## Išvados

OpenClaw itin atvira architektūra yra svarbi jos patrauklumo dalis. Sudėtingoji dalis prasideda tada, kai ta lankstumas turi tapti patikima verslo infrastruktūra: pastovus skaičiavimas, tapatybė, slaptukai, tinklo ryšiai, modelio politika, integracijos ribos, injekcijos apsauga, stebėsena, atnaujinimai ir atkūrimas.

Organizacijoms, kurios jau veikia ant Microsoft, RapidClaw yra mūsų atsakymas į tą diegimo spragą: palikite OpenClaw kaip vykdymo laiką, nustatykite Microsoft architektūrinius pasirinkimus vieną kartą ir paverskite juos pakartojamu eksploatavimo modeliu.

**ŠALINIAI**


[1] OpenClaw Dokumentacija, docs.openclaw.ai, prieiga atlikta rugpjūtį 2026.
docs.openclaw.ai

[2] Microsoft Learn, „Kas yra Microsoft Copilot Studio?“ atnaujinta 3 2026.
learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-what-is-copilot-studio

[3] Microsoft Learn, „Kas yra Microsoft Foundry Agent Service?“ atnaujinta 19 2026.
learn.microsoft.com/en-us/azure/foundry/agents/overview

[4] LangGraph dokumentacija, LangChain, prieiga: 2026.
docs.langchain.com/oss/python/langgraph/overview

[5] CrewAI dokumentacija, prieiga: 2026.
docs.crewai.com/en/introduction

[6] OpenClaw Dokumentacija, Gateway vykdymo aprašas ir diagnostika, prieita 2026.
docs.openclaw.ai/gateway

[7] NIST AI 600-1, Dirbtinio intelekto rizikos valdymo sistema: Generatyvinio dirbtinio intelekto profilis, 2024 2.9, § Informacijos saugumas.
nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[8] OWASP Top 10 for LLM Applications 2026 — Užklausų injekcija ir perteklinė agentūra, OWASP GenAI Security Project, August 2026.
genai.owasp.org/resource/owasp-genai-llm-top-10-2026

[9] RapidStart, „RapidClaw for OpenClaw — Valdomas AI Azure,“ prieita 2026.
www.rapidstart.com/en/products/rapidclaw-for-openclaw

[10] Microsoft Learn, „Įtaisytos politikos modelio diegimui Microsoft Foundry portale,“ atnaujinta 18 2026.
learn.microsoft.com/en-us/azure/foundry/how-to/model-deployment-policy

[11] Microsoft Learn, „Roles pagrįsta prieigos kontrolė Microsoft Foundry,“ atnaujinta 3 2026.
learn.microsoft.com/en-us/azure/foundry/concepts/rbac-foundry

[12] Microsoft Learn, „Diegimo tipai Microsoft Foundry modeliams,“ atnaujinta: 6, 2026.
learn.microsoft.com/en-us/azure/foundry/foundry-models/concepts/deployment-types

[13] Microsoft Learn, „Valdomos tapatybės Azure ištekliams,“ Microsoft Entra dokumentacija.
learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/overview

[14] Microsoft Learn, „Kas yra Azure nusileidimo zona?“ Cloud Adoption Framework, atnaujinta 26 2026.
learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone

[15] Microsoft Learn, „Autorotacijos supratimas Azure Key Vault,“ atnaujinta: 10, 2026.
learn.microsoft.com/en-us/azure/key-vault/general/autorotation


# OpenClaw — vykdymo aplinka. RapidClaw — Microsoft diegimo modelis aplink ją.

## Apie RapidStart

RapidStart kuria Microsoft-pirmas verslo programas ir agentų infrastruktūrą, skirtą palengvinti pažangios verslo technologijos diegimą, valdymą ir priėmimą.