RapidStart

TEKNİK RAPOR

# Kurumsal Ortamda OpenClaw'u Devreye Almak

Yerel ajandan güvenli, güvenilir ve yönetilebilir kurumsal altyapıya.

RAPİDSTART · AĞUSTOS 2026

Mimari · Dağıtım · Güvenlik
Yönetişim · Operasyon

ÖZET

# Yönetici Özeti

OpenClaw, alışılmadık ölçüde açık olduğu için ilgi çekiyor. Kurumu tek bir buluta, tek bir model sağlayıcıya, tek bir kimlik sistemine ya da tek bir işletim modeline mahkûm etmeden, kendi barındırdığınız bir ajan çalışma zamanı sunuyor. Bu esneklik hem işin en güçlü yanı hem de dağıtım yükünün kaynağı.

Kurumsal kullanımda iş, kurulumun çok ötesine hızla geçer. Kurumun kalıcı bir barındırma ortamı oluşturması, kimlik ve kimlik bilgisi sınırlarını tanımlaması, modelleri ve iş sistemlerini bağlaması, ajanların neler yapabileceğini sınırlaması, istem enjeksiyonu ve aşırı yetki gibi ajana özgü tehditlere karşı savunma kurması ve izleme, güncelleme ve kurtarma süreçlerini oturtması gerekir.

Sonuç pratiktir: OpenClaw'u seçen kurumlar, onu çevreleyen ortamı sistemin bir parçası olarak ele almalıdır. Sağlam bir dağıtım deseni; altyapı, güvenlik ve operasyon kararlarını açıkça ortaya koyar, tekrarlanabilir olanı otomatikleştirir, iş yüküne özgü kalanı belgeler ve ortamı yeniden kurulabilir hâle getirir.

> **Çalışma zamanını OpenClaw sağlar. Üretime hazır olmak ise onun etrafındaki mimariden, kontrollerden ve operasyon pratiklerinden gelir.**

## Dört sonuç

01 **OpenClaw güçlü bir tercihtir** — çalışma zamanı sahipliği, model esnekliği ve genişletilebilirlik, tümüyle yönetilen bir SaaS işletim modelinden daha önemli olduğunda.

02 **Yönetilen ajan platformları güçlü alternatiflerdir** — kurum, platform, çalışma zamanı ve operasyon yükünün daha büyük bölümünü bir sağlayıcının üstlenmesini istediğinde.

03 **Üretime hazır olmak**, en az ajan çalışma zamanının kendisi kadar kimliğe, izinlere, istem enjeksiyonu savunmalarına, gözlemlenebilirliğe ve kurtarmaya bağlıdır.

04 **Asıl dönüm noktası** “OpenClaw çalışıyor” demek değildir. “Onu öngörülebilir biçimde dağıtabiliyor, yönetebiliyor, işletebiliyor ve yeniden kurabiliyoruz” demektir.

# 01 OpenClaw Kurumlar İçin Neden Önemli

Kurumsal yapay zekâdaki asıl değişim, modellerin daha iyi sorulara yanıt vermesi değildir. Asıl değişim, ajanların sürekli erişilebilir kalabilmesi, araç kullanabilmesi, bağlamı koruyabilmesi ve sistemler arasında eylem gerçekleştirebilmesidir. OpenClaw; kendi barındırdığınız bir ağ geçidini, kanalları, oturumları, belleği, araçları, becerileri ve çok ajanlı yönlendirmeyi tek bir çalışma zamanında bir araya getirir.[1]

| KURUMSAL FAYDA | NEDEN ÖNEMLİ |
|---|---|
| **Sürekli çalışan ajanlar** | Kalıcı bir çalışma zamanı, tek bir çalışanın sohbet penceresi açmasına bağlı kalmadan bir sürecin ya da ekibin çevresinde çalışabilir. |
| **İş sistemlerinde eylem** | Araçlar ve MCP bağlantıları, ajanın soru yanıtlamaktan çıkıp CRM, ERP, Microsoft 365 ve API'ler üzerinde okuma ve eylem yapmasına olanak tanır. |
| **Model esnekliği** | Çalışma zamanı, tek bir modeli uygulamanın kendisi hâline getirmek yerine birden çok model sağlayıcıya ve yük devretme desenine göre yapılandırılabilir.[1] |
| **Çalışma zamanı denetimi** | Çalışma zamanının nerede barındığına, nasıl genişletildiğine ve çevresindeki mimarinin ne kadarına sahip olduğuna kurum karar verir. |

## İşin etrafında kurulan ajanlar

Çoğu şirkette bilgi zaten fazlasıyla vardır; asıl zorluk, bu bilgiyi sistemler arasında bir araya getirmek ve ardından bir sonraki adımı hayata geçirmektir. Bir ajan; bağlamı toplayan, bunun üzerinde akıl yürüten ve işi ilerletmek için araçları kullanan bağlayıcı katman hâline gelebilir. Bu, her çalışana kişisel bir sohbet botu vermekten farklı bir şeydir.

> **Uzun ömürlü kurumsal ajanlarda model, giderek daha çok ajanın altındaki altyapı olarak ele alınabilir — uygulamanın kendisi olarak değil.**

# 02 OpenClaw Nereye Oturuyor

OpenClaw'un çevresindeki seçenekler birebir muadili değildir; farklı katmanlarda dururlar. Asıl yararlı soru, kurumun işine, denetim gereksinimlerine, entegrasyon ihtiyaçlarına ve mühendislik iştahına hangi işletim modelinin en iyi uyduğudur.

| SEÇENEK | KATEGORİ | GÜÇLÜ OLDUĞU ALAN | EN UYGUN OLDUĞU DURUM |
|---|---|---|---|
| **Copilot Studio** | Yönetilen low-code ajan platformu | Microsoft'a özgü geliştirme, bağlayıcılar, yönetişim ve yayınlama | Kurum, yönetilen bir platform ve asgari çalışma zamanı sahipliği istediğinde.[2] |
| **Microsoft Foundry Agent Service** | Yönetilen ajan barındırma | Yönetilen uç noktalar, ölçekleme, kimlik ve gözlemlenebilirlikle birlikte istem tabanlı ve barındırılan ajanlar | Özel kod gerektiğinde, ancak Microsoft tarafından yönetilen barındırma ve kurumsal kontroller tercih edildiğinde.[3] |
| **OpenClaw** | Kendi barındırdığınız ajan çalışma zamanı | Çalışma zamanı sahipliği, kanallar, araçlar, beceriler, model esnekliği, açık mimari | Kurum, çalışma zamanına ve çevresindeki mimari kararlara sahip olmak istediğinde.[1] |
| **LangGraph / CrewAI** | Geliştirici framework'leri | Kodla yazılan ajan uygulamaları ve orkestrasyon üzerinde azami denetim | Ekip, bir çalışma zamanı dağıtmak yerine kendine özgü bir ajan uygulaması geliştiriyorsa. |
| **Zapier / benzerleri** | Yönetilen yapay zekâ otomasyonu | Hızlı SaaS bağlantısı ve iş akışı otomasyonu | SaaS otomasyonunun kapsam genişliği, çalışma zamanı sahipliğinden daha önemli olduğunda. |

## Yönetilen platformlar: daha az çalışma zamanı sahipliğinin daha iyi olduğu durumlar

Bazen yönetilen bir kurumsal platform daha doğru tercihtir. Low-code geliştirme, Microsoft tarafından işletilen operasyon ve yerleşik yönetişim birincil gereksinimse Copilot Studio güçlüdür. Kurum; çalışma zamanının sahipliğine, geniş genişletilebilirliğe, model esnekliğine ve çevredeki mimariyi şekillendirme yetkisine değer veriyorsa OpenClaw daha ikna edicidir. Karar yalnızca hangi ürünün daha iyi olduğu değildir; kurumun hangi işletim modeline sahip olmak istediğidir.

## Yönetilen barındırma ve geliştirici framework'leri

Yönetilen ajan barındırma hizmetleri, geliştirme ekibi özel kod yazmak isterken barındırma, kimlik ve gözlemlenebilirlik katmanının daha büyük bölümünü sağlayıcının işletmesi istendiğinde işe yarar. LangGraph ve CrewAI gibi geliştirici framework'leri yığında daha alt katmanda yer alır ve ekip ajan uygulamasını kendisi geliştirmeye niyetliyse daha uygundur. OpenClaw ise kendi ağ geçidi, kanalları, becerileri ve işletim modeliyle hâlihazırda var olan, kendi barındırdığınız bir çalışma zamanı olarak ayrışır.[3][4][5]

# 03 OpenClaw'un Yanlış Tercih Olduğu Durumlar

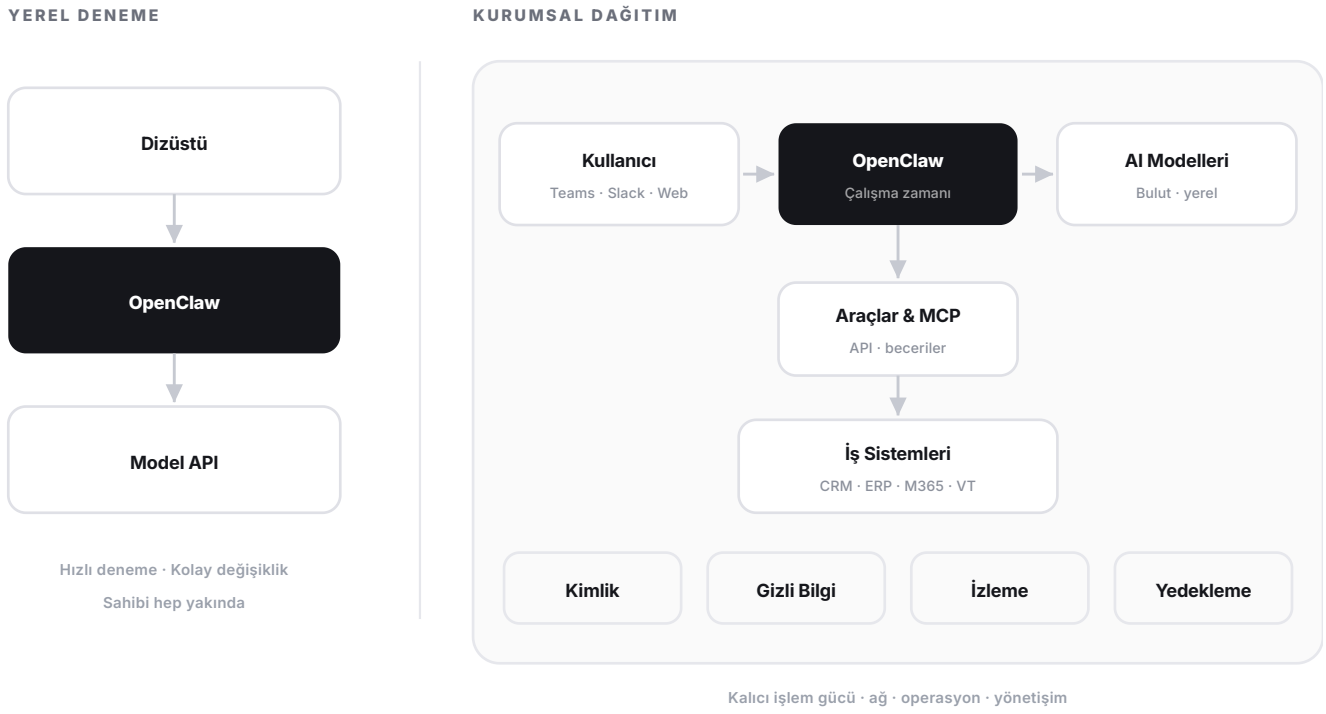
İnandırıcı bir mimari karar, eleyici koşulları da içerir. OpenClaw, yalnızca açık olduğu ya da çalışma zamanı güçlü olduğu için en iyi yanıt değildir.

| ELEYİCİ KOŞUL | DAHA İYİ YANIT |
|---|---|
| **Sağlayıcı tarafından yönetilen bir çalışma zamanı ve SLA istiyorsunuz** | Copilot Studio veya Foundry Agent Service gibi yönetilen bir platform operasyonel olarak daha uygun olabilir. |
| **Sorun, deterministik iş akışı otomasyonundan ibaret** | Power Automate, Logic Apps, Zapier ya da başka bir iş akışı aracı daha basit ve yönetmesi daha kolay olabilir. |
| **Bulut operasyonlarının sahibi yok** | Kendi barındırdığınız bir çalışma zamanı; yama, izleme, kurtarma ve güvenlik sorumlulukları doğurur. Bunların sahibi yoksa, bu yükümlülüğü yaratmayın. |
| **Ekibinizin karşılayamayacağı katı düzenleyici kontrol gereksinimleri** | Yönetilen kontrolleri, sertifikaları, destek modeli ve denetim yüzeyleri bu gereksinimi karşılayan bir platform seçin. |
| **Kurumun çalışma zamanı sahipliğine veya model esnekliğine ihtiyacı yok** | OpenClaw'un başlıca avantajları, getirdiği operasyon yükünü haklı çıkarmayabilir. |
| **Kullanım senaryosu tek ve dar kapsamlı bir asistan** | Yönetilen bir ajan veya uygulamaya gömülü bir özellik, genel amaçlı bir çalışma zamanından belirgin biçimde daha basit olabilir. |

> **Açık kaynak bir strateji değildir. Kendi barındırmak, yalnızca sağladığı denetim yarattığı sorumluluğa değdiğinde anlamlıdır.**

# 04 Yerel Ajandan Kurumsal Altyapıya

OpenClaw ilk deneyimi bilinçli olarak kolaylaştırır. Bu, geliştirme ve denemeler için mükemmeldir. Kurumsal bir dağıtım ise bir katman öncesinden başlar: kurumun önce, çalışma zamanının içinde yaşayabileceği kalıcı bir ortama ihtiyacı vardır.




Bulut VM'i çoğu zaman en basit desendir; ancak kalıcı işlem gücü hemen ardından bölge, işletim sistemi, depolama, yönetim, yeniden başlatma davranışı, yama, izleme ve kurtarma tercihlerini gündeme getirir.

> **Geliştiricinin sorusu "Çalıştırabilir miyim?"dir. Kurumun sorusu ise "Güvenilir biçimde, kesintisiz ve bilinen kontroller altında çalıştığı bir ortam kurabilir miyiz?"dir.**

# 05 Üretim Dağıtımı İçin Temel Standart

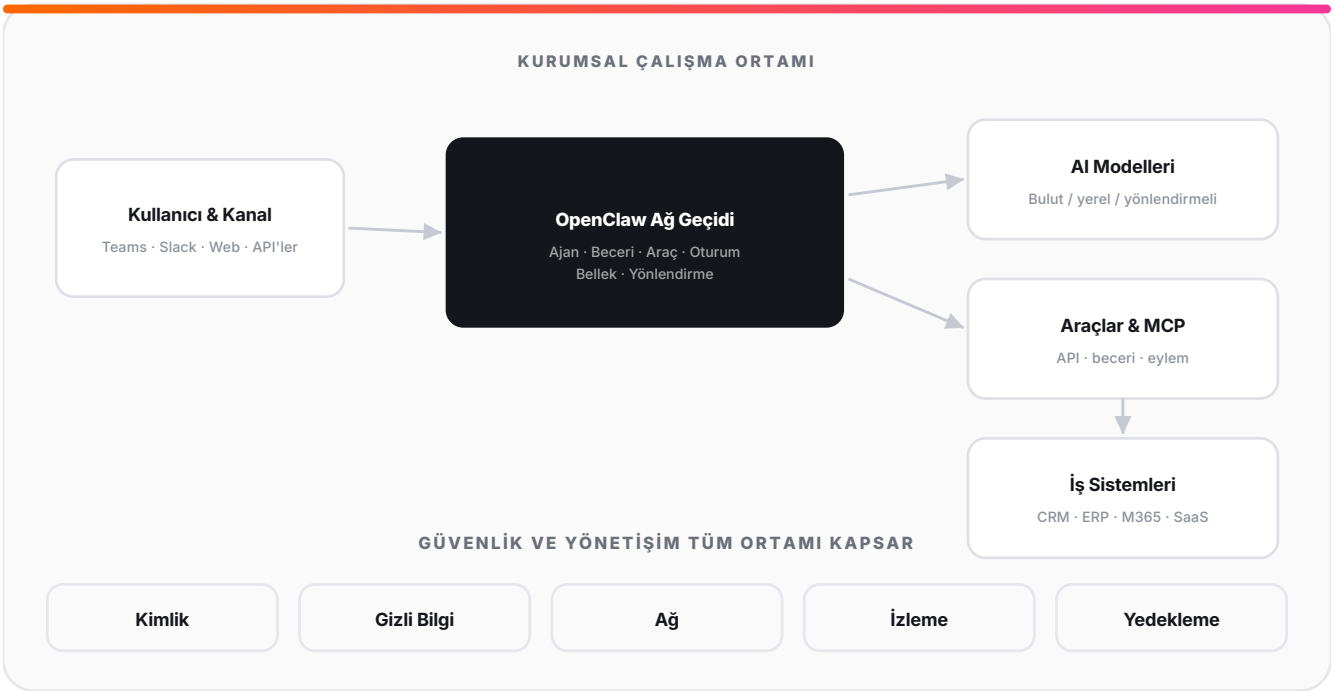
Yararlı bir dağıtım kılavuzu soru sormakla yetinmemelidir. Bu temel standart, bir OpenClaw ortamı kurumsal altyapı olarak ele alınmadan önce normalde bilinçli biçimde verilmesi gereken kontrol ve operasyon kararlarını toplar. Ürünler ve sağlayıcılar değişebilir; gereksinimler değişmez.

| KARAR | ÖNERİLEN TEMEL STANDART | GEREKÇE |
|---|---|---|
| **Çalışma zamanı sunucusu** | Kalıcı, kuruma ait işlem gücü | Çalışma zamanını çalışan cihazlarından bağımsız tutar ve kuruma tanımlı bir işletim sınırı verir. |
| **Kurumsal kimlik** | Destekleniyorsa ayrılmış bir iş yükü kimliği[13] | Üretim erişiminin, ajanı kuran kişinin kişisel kimlik bilgilerine bağlanmasını önler. |
| **Kaynak erişimi** | Platform destekliyorsa kimlik tabanlı erişim[13] | Uzun ömürlü gizli bilgilere bağımlılığı azaltır; iptal edilebilirliği ve denetlenebilirliği artırır. |
| **Gizli bilgiler** | Yönetilen gizli bilgi deposu; geliştirici bilgisayarında üretim gizli bilgi dosyası bulundurulmaz[15] | Token'ları ve anahtarları; erişim denetimi, rotasyon ve kurtarma prosedürleri olan yönetilen varlıklar olarak ele alır. |
| **Model erişimi** | Bölge, kota ve yedek model politikası belgelenmiş, onaylı sağlayıcılar ve modeller[10][12] | Model seçimini bir geliştirici tercihi olmaktan çıkarıp operasyonel bir politika hâline getirir. |
| **İnsan arayüzü** | Kullanım senaryosuna uygun, onaylı kurumsal kanallar | Ajan erişimini, kurumun yönetebildiği iletişim yüzeylerinin içinde tutar. |
| **Operasyon** | Merkezî sağlık izleme, günlükleme ve uyarıların yanı sıra belgelenmiş yeniden başlatma ve güncelleme prosedürleri | Arızayı görünür kılar ve operatörlere tekrarlanabilir bir müdahale yolu verir. |
| **Dağıtım** | Elle kurulmuş, her biri kendine özgü VM'ler yerine betiklenmiş, tekrarlanabilir yapılandırma[14] | Yeniden kurmayı, ikinci bir örneğin dağıtımını ve kurtarmayı gerçekçi kılar. |

Bunlar temel mimari ilkelerdir, evrensel bir politika değildir. Ağ topolojisi, belirli izinler, saklama süreleri, onay eşikleri, kurtarma hedefleri ve onaylı hizmet sağlayıcılar yine de iş yüküne ve kuruma bağlıdır.

# 06 Kurumsal OpenClaw İçin Referans Mimari

Diyagram bir referans desendir; OpenClaw'un dayattığı bir mimari değildir.




## Çalışma zamanı sınırı

Çevredeki ortam — kimlik, gizli bilgiler, ağ, izleme ve kurtarma — bir kurumun esnek bir çalışma zamanını işletilebilir bir sisteme dönüştürdüğü yerdir. Kimlik, ajanın kim ya da ne olduğunu belirler. Gizli bilgiler, hangi kimlik bilgilerini kullanabileceğini belirler. Ağ, nereye erişebileceğini tanımlar. İzleme, operatörlerin arızayı görüp göremeyeceğini ortaya koyar. Yedekleme ve kurtarma ise ortamın geri getirilip getirilemeyeceğini belirler.

OpenClaw bu sınırı şekillendirme özgürlüğünü kuruma bırakır; bir Microsoft dağıtım deseni ise bu tercihlerin tanımlı bir bölümünü önceden yapar.

# 07 Kimlik, Gizli Bilgiler ve Ağ

KİMLİK

## Ajanı kurumsal bir aktör hâline getirin

Üretimdeki bir ajanı, onu kuran çalışanın kişisel kimlik bilgilerine bağlamayın. Hedef platform destekliyorsa ayrılmış bir kurumsal iş yükü kimliği kullanın. Bazı dış sistemler için uygulama kimlik bilgileri hâlâ gerekli olabilir; ancak bunlar, yaşam döngüsü sahibi net olan, en az ayrıcalık ilkesine uyan ve iptal prosedürleri tanımlanmış yönetilen istisnalar olmalıdır.

GİZLİ BİLGİLER

## Önce gereksiz gizli bilgileri ortadan kaldırın

En iyi gizli bilgi, dağıtımın hiç ihtiyaç duymadığıdır. Kimlik tabanlı erişim, saklanması gereken ham kimlik bilgisi sayısını azaltabilir. Gizli bilgilerin gerekli kaldığı yerlerde bunları yönetilen bir gizli bilgi deposuna koyun, erişimi çalışma zamanı kimliğiyle sınırlayın ve belgelenmiş rotasyon ile olay müdahale prosedürleri oluşturun. Üretimde koda gömülü kimlik bilgilerinden ve geliştirici bilgisayarındaki gizli bilgi dosyalarından kaçının.

AĞ

## Erişilebilirlik bir politikadır

Üretim tasarımı, geliştirme ortamının pratik varsayılanlarını devralmak yerine gelen yönetim erişimini ve giden erişilebilirliği bilinçli olarak tanımlamalıdır. Gereksiz hiçbir genel gelen yol bırakmayın; özel bağlantı, bastion hizmeti, VPN veya eşdeğeri gibi denetimli bir yönetim yolu kullanın. Giden erişim; iş yükünün gerçekten ihtiyaç duyduğu model uç noktaları, kanallar, paket kaynakları ve iş sistemleriyle sınırlı olmalıdır.

> **Kimlik, ajanın kim olduğunu tanımlar. Gizli bilgiler, hangi kapıları açabileceğini tanımlar. Ağ ise hangi kapılara ulaşabileceğini tanımlar.**

# 08 İş Sistemleri Entegrasyonu Değerle Riskin Buluştuğu Yerdir

Ajan; CRM, ERP, işbirliği paketleri, veritabanları, API'ler ve MCP sunucuları üzerinde çalışabildiğinde OpenClaw'un değeri hızla artar. Yanlış bir kararın sonuçları da öyle. Bir aracın kullanılabilir olması politika değildir.

| YETENEK | ÖRNEK | ÖNERİLEN KONTROL |
|---|---|---|
| **Okuma** | Müşteri, sipariş veya belge bağlamını getirme | En az ayrıcalık; veri kapsamı kullanıcının veya sürecin ihtiyacını izler. |
| **Analiz** | Özetleme, sınıflandırma, karşılaştırma, istisnaları tespit etme | Kaynak bağlamını saklayın ve üretilen yorumu, kökeni belirtilmeden yetkili veri gibi ele almayın. |
| **Öneri** | Yanıt taslağı hazırlama veya sonraki eylemi önerme | Yürütmeden önce hangi durumlarda insan incelemesinin zorunlu olduğunu tanımlayın. |
| **Yazma** | CRM kaydını güncelleme veya görev oluşturma | Yazılabilir nesne ve alanları sınırlayın ve kime ait olduğu izlenebilen bir kimlik kullanın. |
| **Yürütme** | Bir API'yi, iş akışını veya süreci tetikleme | Ön koşulları, limitleri, idempotentliği ve onay kapılarını tanımlayın. |
| **Dışa dönük eylem** | E-posta gönderme, dışarıya paylaşım yapma, müşteriye görünen durumu değiştirme | Daha güçlü onay, denetim kanıtı ve hızlı iptal yeteneği zorunlu tutun. |

> **Amaç her yerde azami özerklik değildir. Amaç, iş için doğru düzeyde özerkliktir.**

# 09 Ajana Özgü Tehditler: Enjeksiyon ve Şaşırtılmış Vekil

Yalnızca izinler bir ajanı güvenli kılmaz. Bir ajan, bir aracı kullanmaya meşru biçimde yetkili olabilir ve yine de bu yetkiyi yanlış kullanacak şekilde manipüle edilebilir. NIST; doğrudan ve dolaylı istem enjeksiyonunu, bir LLM entegre uygulamanın sonradan getirdiği içeriğe gömülü saldırılar dâhil olmak üzere üretken yapay zekâ sistemleri için risk olarak ele alır.[7] OWASP ise istem enjeksiyonunu (Prompt Injection) ve aşırı yetkiyi (Excessive Agency) önde gelen LLM uygulama riskleri arasında sayar.[8]

| TEHDİT | NASIL GÖRÜNDÜĞÜ | KONTROL DESENİ |
|---|---|---|
| **Doğrudan istem enjeksiyonu** | Kullanıcı, ajana politikayı yok saymasını veya bir aracı kötüye kullanmasını söyler | Sistem düzeyinde politika uygulama, izin listesindeki araçlar, kapsamı daraltılmış kimlikler, sonuç doğuran eylemler için onay kapıları. |
| **Dolaylı istem enjeksiyonu** | Bir web sayfasına, belgeye, e-postaya veya getirilen veriye gizlenmiş kötü niyetli talimatlar | Getirilen içeriği güvenilmeyen veri olarak ele alın; içeriği kontrol talimatlarından ayırın; araç çağrılarını ve dışa veri çıkışını sınırlayın. |
| **Şaşırtılmış vekil davranışı** | Ajanın yetkisi meşrudur, ancak güvenilmeyen bir kaynak bu yetkiyi yanlış amaçla kullanmasına yol açar | Eylemleri kimliği doğrulanmış bir talep sahibine veya sürece bağlayın, niyeti ve bağlamı doğrulayın, etkisi ciddi olduğunda onay zorunlu tutun. |
| **Aşırı yetki** | Ajanın, görevin gerektirdiğinden daha fazla aracı, kapsamı veya özerkliği vardır | En az ayrıcalık, dar araç şemaları, açık yazma ve yürütme sınırları, mümkün olduğunda kısa ömürlü kimlik bilgileri. |
| **Hatalı çıktı işleme** | Üretilen çıktı doğrudan koda, SQL'e, kabuğa, HTML'e veya başka bir sisteme aktarılır | Çıktıları doğrulayın ve temizleyin; mümkün olduğunda serbest biçimli komut yürütme yerine tipli araç arayüzleri kullanın. |

> **Güvenli bir ajan mimarisi, hem ajanın neyi yapmasına izin verildiğini hem de bu kararı hangi bilginin etkileyebileceğini yönetmelidir.**

# 10 Modeller, Güvenilirlik ve Operasyon

## Model stratejisi

Yerel bir kullanıcı favori modelini seçebilir. Bir kurumun ise politikaya ihtiyacı vardır: onaylı sağlayıcılar, coğrafya, gizlilik koşulları, gecikme, kotalar, yedek model davranışı ve maliyet kontrolleri. OpenClaw çalışma zamanı olarak kalırken Microsoft Foundry, yönetişimi sağlanmış bir model erişim katmanı sunabilir.[3][10][11] Farklı görevler farklı modelleri gerektirebilir; iş süreci tek bir model kuşağına sabitlenmemelidir.

## Güvenilirlik ve gözlemlenebilirlik

Bir süreç ajana bağımlı hâle geldiğinde operatörlerin şunları bilmesi gerekir: ağ geçidi sağlıklı mı, kanallar bağlı mı, model erişiminde hata var mı, kimlik bilgilerinin süresi dolmuş mu ve ajan sonuç doğuran hangi eylemleri denedi. OpenClaw ağ geçidi tanılamaları ve kanal yoklamaları sunar; ancak çevresindeki işletim modeli yine de sahiplik, uyarı ve müdahale prosedürleri gerektirir.[6]

## Boyutlandırma ve maliyet: sahte kesinliğe değil, maliyet etkenlerine odaklanın

| MALİYET ETKENİ | PRATİK YÖNLENDİRME |
|---|---|
| **VM / işlem gücü** | CPU ve bellek; kanallara, eşzamanlılığa, yerel araçlara ve modellerin uzakta mı yoksa yerelde mi çalıştığına bağlıdır. Mütevazı, genel amaçlı bir VM ile başlayın ve ölçeklendirmeden önce gerçek iş yükünü yük testine tabi tutun. |
| **Model kullanımı** | Modeller uzakta çalıştığında genellikle baskın değişken maliyettir. Token ve istekleri ajan ve görev bazında izleyin, kotalar tanımlayın ve uygun olan yerlerde basit işleri daha ucuz modellere yönlendirin. |
| **Depolama / günlükler** | Konuşma durumu, günlükler ve üretilen dosyalar birikir. VM'i sınırsızmış gibi görmek yerine saklama sürelerini tanımlayın ve depolama büyümesini izleyin. |
| **Operasyon** | Gizli maliyet insan sahipliğidir: yama, olay müdahalesi, güncellemelerin test edilmesi ve erişim gözden geçirmeleri. Otomasyon bu maliyeti, VM harcamasından kırpılacak küçük tutarlardan çok daha fazla azaltır. |

Bulut bölgesi, işlem türü ve model ücretleri değiştiği için burada bilinçli olarak kesin fiyat verilmemiştir. Dağıtım deseni bu değişkenleri gizlemek yerine gözlemlenebilir ve değiştirilebilir kılmalıdır.

# 11 Üretime Hazırlık Matrisi

Bu, pratik bir iç kontrol kapısıdır. Bir kurumun her iş yükü için birebir aynı kontrollere ihtiyacı yoktur; ancak kararlar açıkça verilmiş olmalıdır.

| YETENEK | DENEME | KURUMSAL DAĞITIM | ÜRETİME HAZIR DURUŞ |
|---|---|---|---|
| **İşlem gücü** | Dizüstü / gelişigüzel | Kalıcı sunucu | Bilinen boyutlandırma, sahiplik, yeniden başlatma ve kurtarma |
| **Kimlik** | Kişisel kimlik bilgileri | Ayrılmış kimlik deseni | En az ayrıcalık, yaşam döngüsü, denetlenebilirlik |
| **Gizli bilgiler** | Ortam değişkeni / yapılandırma | Korumalı depolama | Kasada saklanan ve döndürülebilir; erişimi denetimli |
| **Ağ** | Pratik varsayılanlar | Kısıtlı gelen / giden trafik | Belgelenmiş yönetim yolu ve segmentasyon |
| **Modeller** | Tercih edilen model | Onaylı sağlayıcı / model | Bölge, kota, yedek model, maliyet politikası |
| **Entegrasyonlar** | Geliştirici token'ları | Kapsamı tanımlı kurumsal erişim | Açık okuma / yazma / yürütme sınırları |
| **Enjeksiyon savunmaları** | Genellikle yok | İstem ve araç koruma bariyerleri | Güvenilmeyen içeriğin ele alınması, onay ve dışa çıkış kontrolleri |
| **İzleme** | Elle inceleme | Sağlık kontrolleri / günlükler | Uyarılar, sahiplik, müdahale prosedürleri |
| **Güncellemeler** | Gelişigüzel | Planlı | Test, geri alma ve sürüm disiplini |
| **Kurtarma** | Genellikle yok | VM / yapılandırma yedeği | Test edilmiş yeniden kurma ve kurtarma prosedürü |
| **Dokümantasyon** | Kişisel notlar | Dağıtım kaydı | Yeniden üretilebilir ortam tanımı |

> **Çalışan bir ajan teknik bir dönüm noktasıdır. İşletilebilir bir ajan ise kurumsal bir yetkinliktir.**

# 12 Yaşam Döngüsü ve Tekrarlanabilirlik Tek Bir Meseledir

Bir üretim ortamı kurulumla bitmez. Sağlanması, güvenliğinin alınması, yapılandırılması, bağlanması, doğrulanması, işletilmesi, güncellenmesi ve eninde sonunda yeniden kurulması gerekir.


Bu yaşam döngüsü adımları, ancak dağıtım kararları tek bir mühendisin hafızasında yaşamak yerine kayda geçirildiğinde güvenilir hâle gelir.

## Yeniden kurma testi

VM bu gece yok olsa, başka bir yönetici ortamı yarın yeniden oluşturabilir mi? Şirket, başka bir ekip veya bölge için ikinci bir örnek dağıtabilir mi? Hangi yapılandırmanın bilinçli, hangi ayarın geçmişten kalma bir kalıntı olduğunu açıklayabilir mi? Kod olarak altyapı, betiklenmiş yapılandırma, dağıtım kayıtları ve denetimli güncelleme yolları; çalışan bir makineyi tekrarlanabilir bir yetkinliğe dönüştürür.

> **Çalışan bir VM bir örnektir. Yeniden üretilebilir bir ortam ise bir yetkinliktir.**

# 13 OpenClaw'dan RapidClaw'a

Buraya kadar anlatılan her şey, bulut ya da sağlayıcıdan bağımsız olarak geçerlidir. OpenClaw çevredeki mimariyi bilinçli olarak açık bırakır. RapidClaw ise bu açıklığın, Microsoft üzerinde standartlaşmış kurumlar için bir dağıtım yüküne dönüştüğü noktada başlar.

**RapidClaw, OpenClaw'un bir fork'u değildir.** Çalışma zamanı yine OpenClaw'dur. RapidClaw, Microsoft kullanan kurumlar için onun etrafında kurulmuş, tercihleri net bir dağıtım ve işletim desenidir. Pratikte bu şu anlama gelir: rehberli dağıtım, yapılandırması onu kuran kişinin belleğinde kalan elle kurulmuş bir VM yerine, ortamın tamamını müşterinin kendi Azure kiracısında yaklaşık yirmi dakikada ayağa kaldırır.[9]

OPENCLAW: TAMAMEN AÇIK

- Bulutu seç
- Kimliği seç
- Model platformunu seç
- Ağı seç
- Güvenlik desenini seç
- İşletim modelini seç

Esneklik, işin özelliğidir.
Her kararın sahibi kurumdur.

RAPIDCLAW: NET TERCİHLER


Daha az mimari karar · daha çok tekrarlanabilirlik

| KARAR ALANI | RAPİDCLAW'UN TERCİHİ |
|---|---|
| **Bulut sınırı** | Müşteriye ait Azure çalışma zamanı[9][14] |
| **Kimlik düzlemi** | Microsoft Entra ile hizalı kurumsal kimlik[9][13] |
| **Model platformu** | Varsayılan olarak Microsoft Foundry / Azure AI yolu[3][10][12] |
| **İnsan arayüzü** | Uygun olduğunda Microsoft Teams |
| **Kontroller** | Microsoft ile hizalı gizli bilgi, erişim, görünürlük ve yönetişim desenleri[9][11][15] |
| **Dağıtım** | Tek seferlik elle kurulum yerine rehberli, tekrarlanabilir yapılandırma[9][14] |

# 14 RapidClaw Neyi Değiştirir — ve Neyi Açık Bırakır

RapidClaw bir boyutta bilinçli olarak dar, diğerinde açıktır. Yeniden icat edilmesine gerek olmayan Microsoft dağıtım tercihlerini standartlaştırırken, OpenClaw'un kendisini iş yüküne uygun ajanlara, becerilere, araçlara ve iş sistemi bağlantılarına açık bırakır.

| SINIR | ANLAMI |
|---|---|
| **RapidClaw'un standartlaştırdıkları** | Azure dağıtım yolu; Microsoft kimlik entegrasyonu; Foundry / Azure AI model yolu; Teams entegrasyon yüzeyi; tekrarlanabilir kurulum ve operasyon deseni. |
| **Müşterinin karar vermeyi sürdürdükleri** | Hangi ajanların geliştirileceği; hangi iş sistemlerinin ve MCP sunucularının açılacağı; okuma, yazma ve yürütme izinleri; onay politikası; saklama süreleri; düzenlemeye tabi iş yükü kontrolleri; iş sahipliği. |
| **RapidClaw'un yapmadıkları** | OpenClaw'u fork etmek veya onun yerini almak; olası her bulut ve çalışma zamanı mimarisini birinci sınıf desteklemek; kurumsal yönetişim ihtiyacını ortadan kaldırmak. |

> **Tercihleri net bir dağıtımın değeri, her kararı ortadan kaldırması değildir. Değeri, Microsoft kullanan kurumların her seferinde sıfırdan vermek zorunda kalmaması gereken kararları ortadan kaldırmasıdır.**

## Sonuç

OpenClaw'un sonuna kadar açık mimarisi, çekiciliğinin önemli bir parçasıdır. Zor kısım, bu esnekliğin güvenilir bir kurumsal altyapıya dönüşmesi gerektiğinde başlar: kalıcı işlem gücü, kimlik, gizli bilgiler, ağ, model politikası, entegrasyon sınırları, enjeksiyon savunmaları, gözlemlenebilirlik, güncellemeler ve kurtarma.

Hâlihazırda Microsoft üzerinde çalışan kurumlar için RapidClaw, bu dağıtım boşluğuna verdiğimiz yanıttır: çalışma zamanı olarak OpenClaw'u koruyun, Microsoft mimari kararlarını bir kez verin ve bunları tekrarlanabilir bir işletim desenine dönüştürün.

KAYNAKLAR


**[1]** OpenClaw Documentation, docs.openclaw.ai, erişim: Ağustos 2026.
docs.openclaw.ai

**[2]** Microsoft Learn, "What is Microsoft Copilot Studio?", 3 Ağustos 2026 tarihinde güncellendi.
learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-what-is-copilot-studio

**[3]** Microsoft Learn, "What is Microsoft Foundry Agent Service?", 19 Ağustos 2026 tarihinde güncellendi.
learn.microsoft.com/en-us/azure/foundry/agents/overview

**[4]** LangGraph belgeleri, LangChain, erişim: Ağustos 2026.
docs.langchain.com/oss/python/langgraph/overview

**[5]** CrewAI belgeleri, erişim: Ağustos 2026.
docs.crewai.com/en/introduction

**[6]** OpenClaw Documentation, ağ geçidi runbook'u ve tanılama bölümü, erişim: Ağustos 2026.
docs.openclaw.ai/gateway

**[7]** NIST AI 600-1, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile", Temmuz 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 ve Excessive Agency, OWASP GenAI Security Project, Ağustos 2026.
genai.owasp.org/resource/owasp-genai-llm-top-10-2026

**[9]** RapidStart, "RapidClaw for OpenClaw — Governed AI in Azure", erişim: Ağustos 2026.
www.rapidstart.com/en/products/rapidclaw-for-openclaw

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

**[11]** Microsoft Learn, "Role-based access control for Microsoft Foundry", 3 Temmuz 2026 tarihinde güncellendi.
learn.microsoft.com/en-us/azure/foundry/concepts/rbac-foundry

**[12]** Microsoft Learn, "Deployment types for Microsoft Foundry Models", 6 Ağustos 2026 tarihinde güncellendi.
learn.microsoft.com/en-us/azure/foundry/foundry-models/concepts/deployment-types

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

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

**[15]** Microsoft Learn, "Understanding autorotation in Azure Key Vault", 10 Nisan 2026 tarihinde güncellendi.
learn.microsoft.com/en-us/azure/key-vault/general/autorotation


# Çalışma zamanı OpenClaw'dur. RapidClaw ise onun etrafındaki Microsoft dağıtım desenidir.

## RapidStart Hakkında

RapidStart; gelişmiş kurumsal teknolojinin dağıtımını, işletilmesini ve benimsenmesini kolaylaştırmak üzere tasarlanmış, Microsoft odaklı kurumsal uygulamalar ve ajan altyapısı geliştirir.