

# MCP-Model Context Protocol

Mimari Analiz, Tehdit Vektörleri ve Savunma Stratejileri

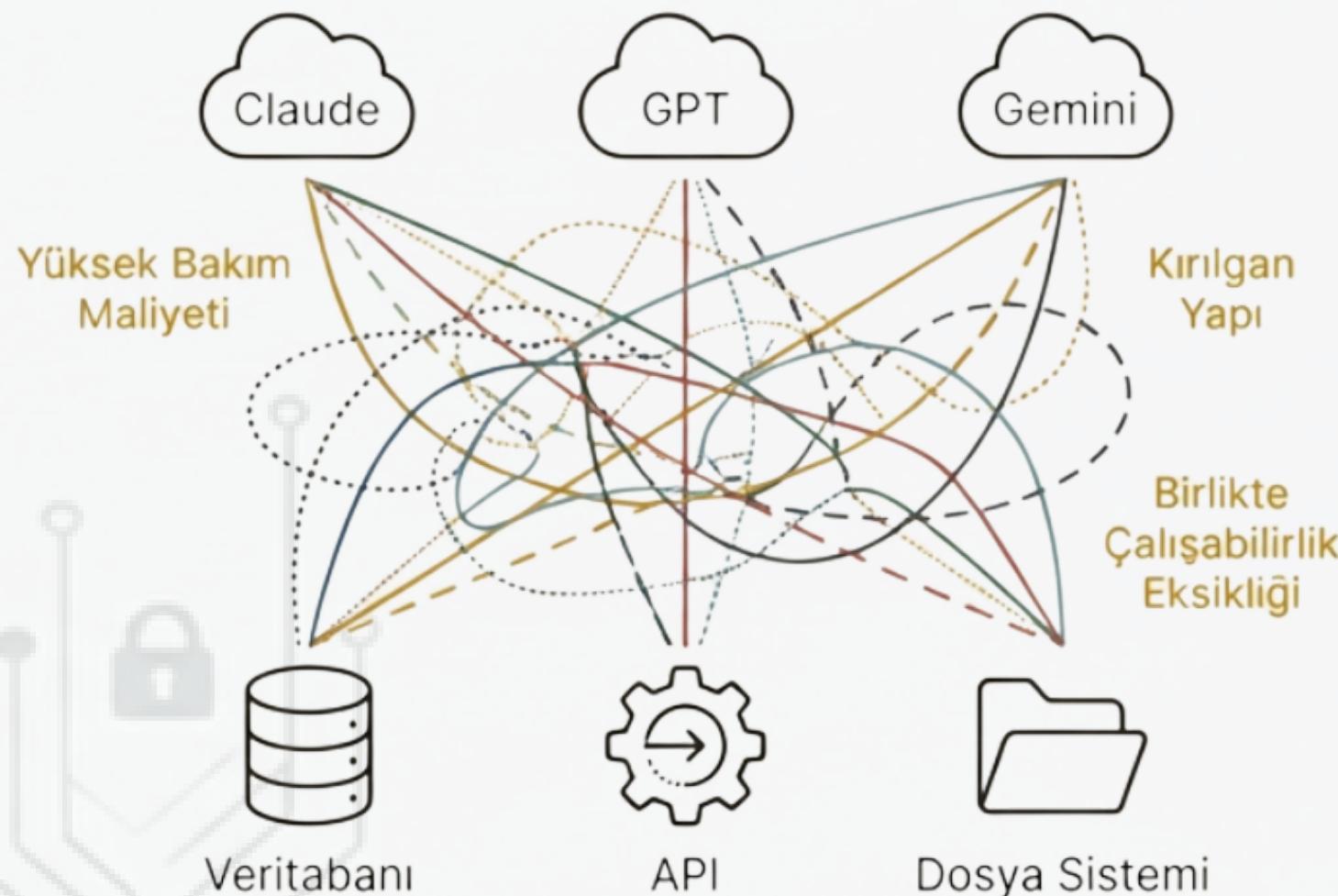
---

Yusuf Talha Arabacı

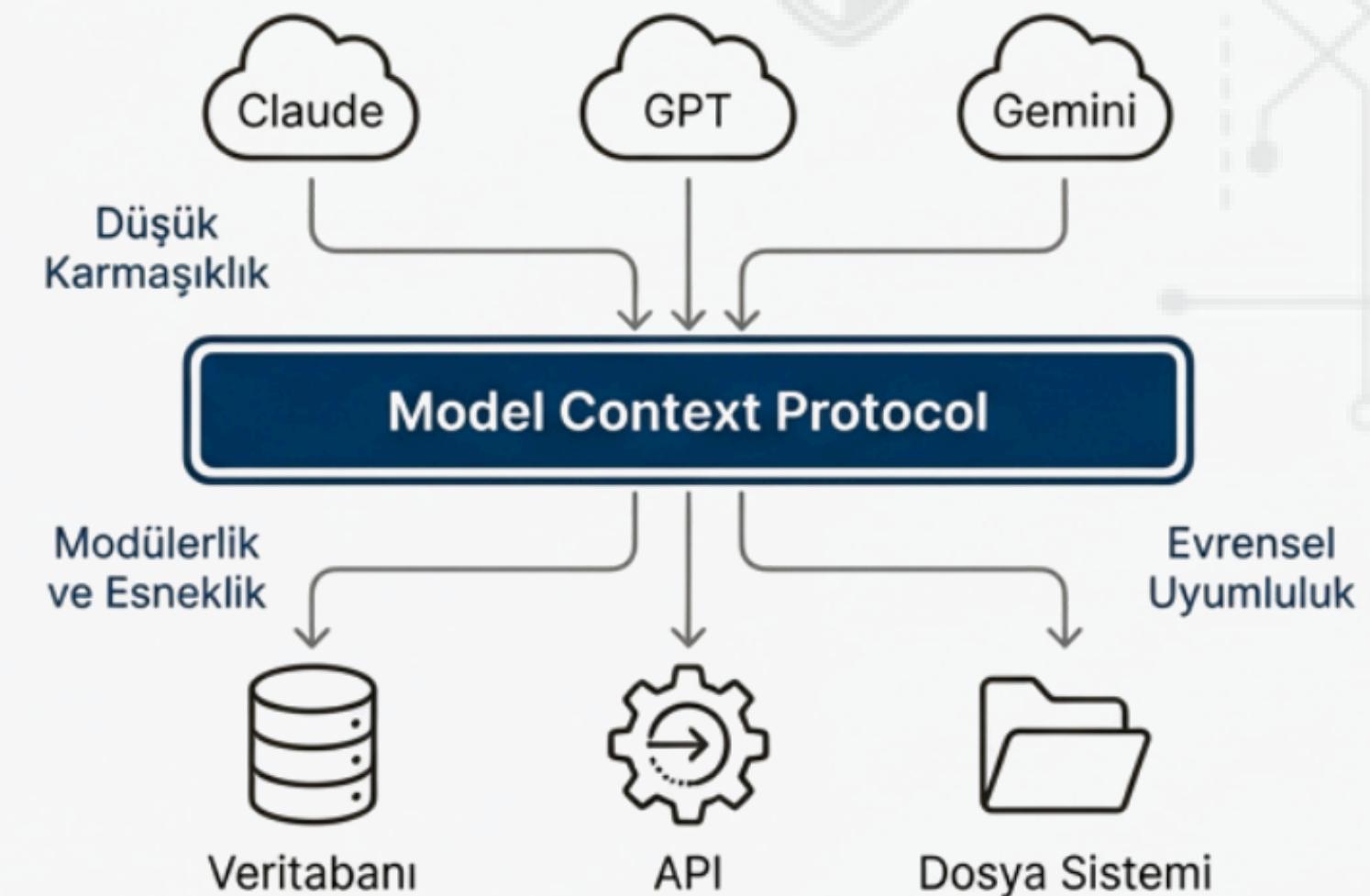
Yazılım Mühendisliği Yüksek Lisans Öğrencisi

Karabük Üniversitesi | Aralık 2025

# I Giriş: Birlikte Çalışabilirlik Krizi



Her AI uygulaması, harici araçlarla iletişim kurmak için özel bağlantılar gerektiriyordu. Bu durum, yüksek bakım maliyetlerine, yavaş geliştirme süreçlerine ve ölçeklendirme zorluklarına yol açıyordu.



MCP, AI uygulamaları için bir 'USB-C' standartı gibi çalışarak, LLM'lerin dış sistemlerle bağlanması için evrensel bir yol sağlar. Bu, geliştirme süreçlerini basitleştirir ve ölçeklenebilir bir ekosistem yaratır.

# | Çözüm: MCP Nedir?

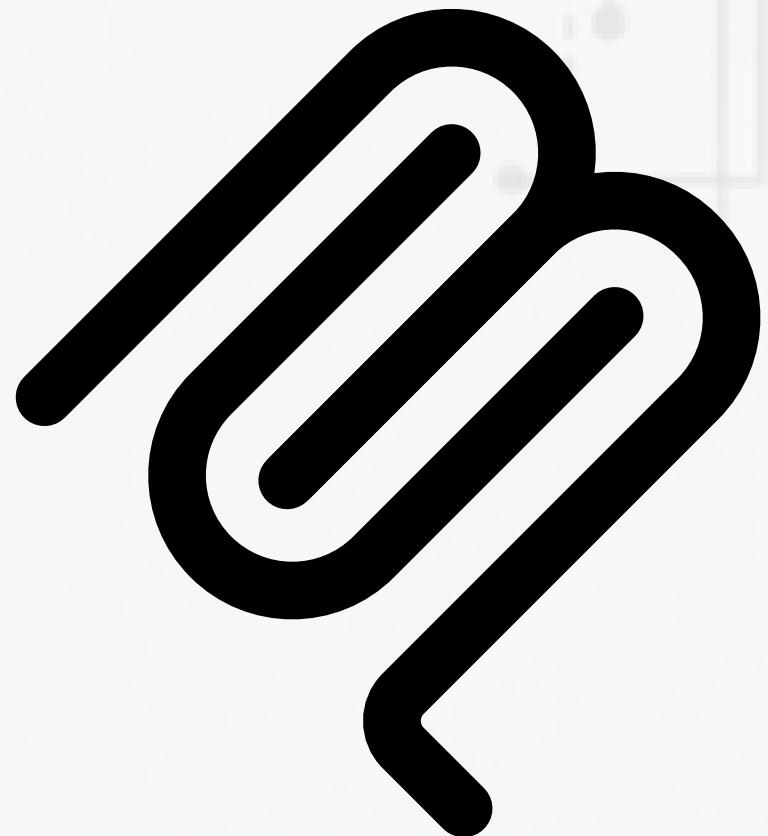
**Tanım:** Anthropic tarafından Kasım 2024'te tanıtılan, AI modelleri ile harici araçlar veya kaynaklar arasında birleşik, çift yönlü iletişim ve dinamik keşif sağlayan açık bir standarttır. Genellikle AI uygulamaları için "USB-C portu" olarak anılır.

**Temel Amacı:** Zekayı (Model) veri kaynaklarından ve araçlardan ayırarak evrensel bir bağlantı sağlamaktır.

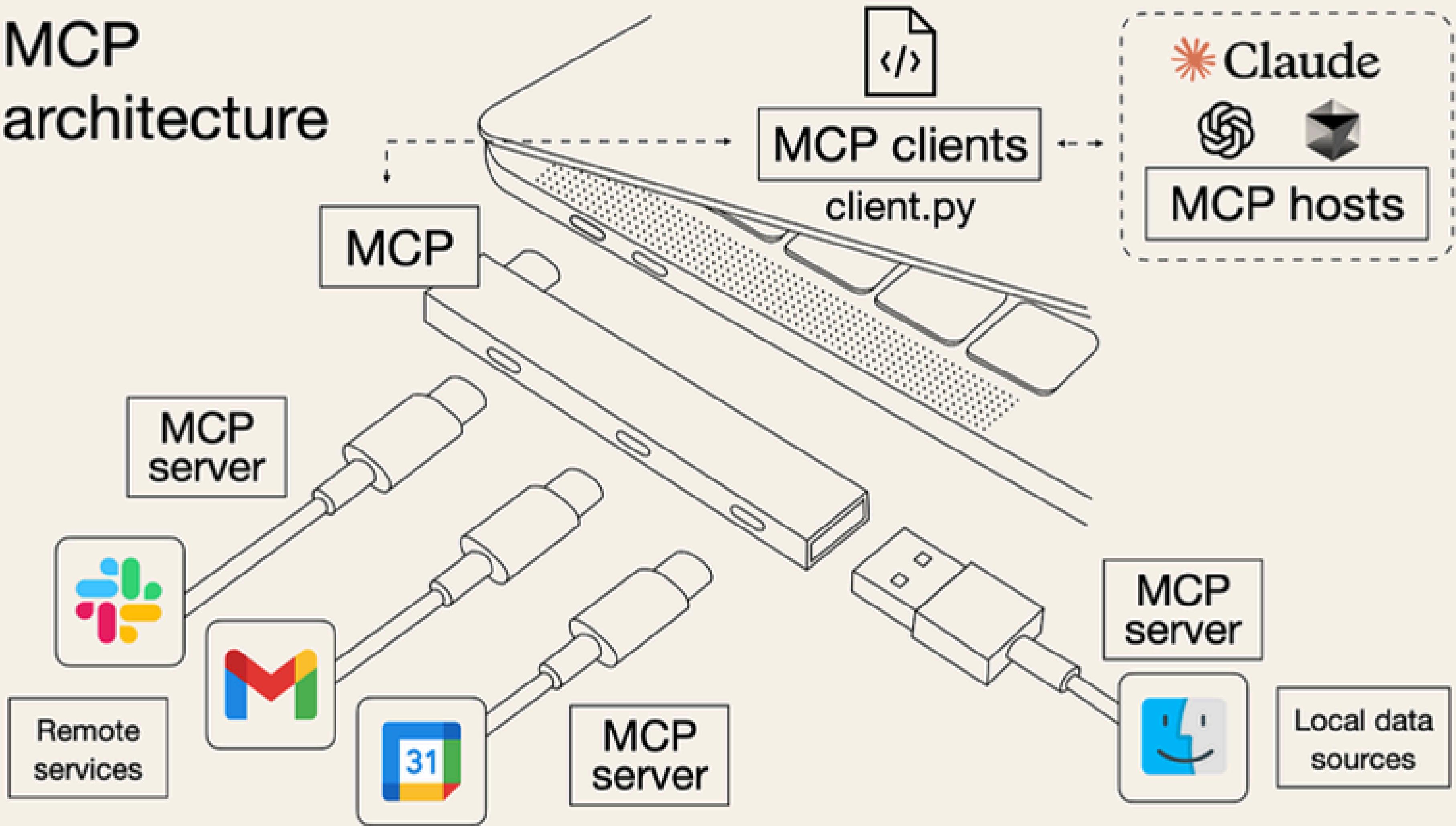
**Stratejik Rol:** LLM'leri pasif metin üreticilerinden, gerçek dünyada eylemler gerçekleştirebilen otonom ajanlara dönüştürmenin temelini atar.

<https://modelcontextprotocol.io/>

<https://www.anthropic.com/news/model-context-protocol>



# MCP architecture



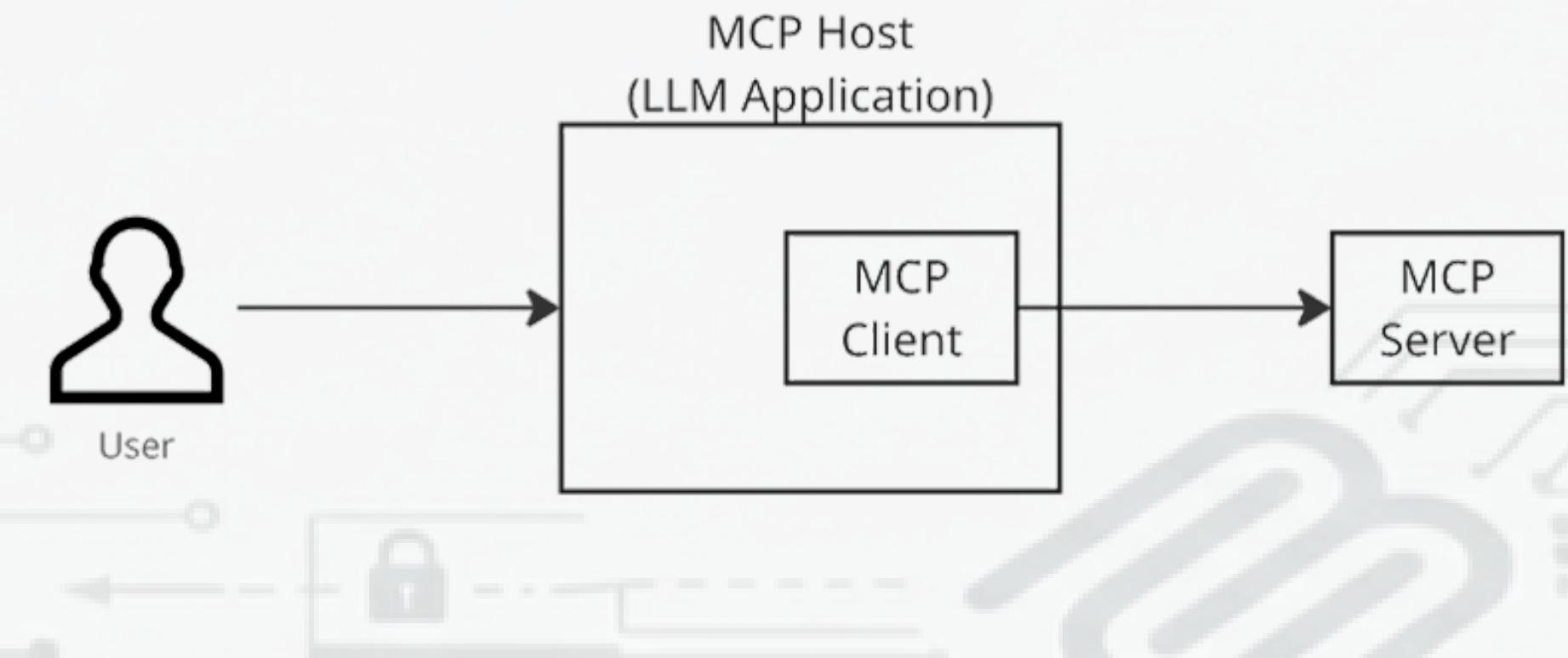
# | Temel Mimari Bileşenler

MCP, temiz bir istemci-sunucu mimarisi kullanır ve iletişim JSON-RPC protokolü üzerinden gerçekleşir.

**MCP Host:** LLM'i barındıran ana uygulama (Claude Desktop, VS Code, Cursor).

**MCP Client:** Host içinde yer alır, LLM'in isteği standart RPC çağrılarına çevirir ve sunucuya 1:1 bağlantıyı yönetir.

**MCP Server:** Harici yetenekleri (API, veritabanı, dosya sistemi) sunan harici hizmet veya programdır.

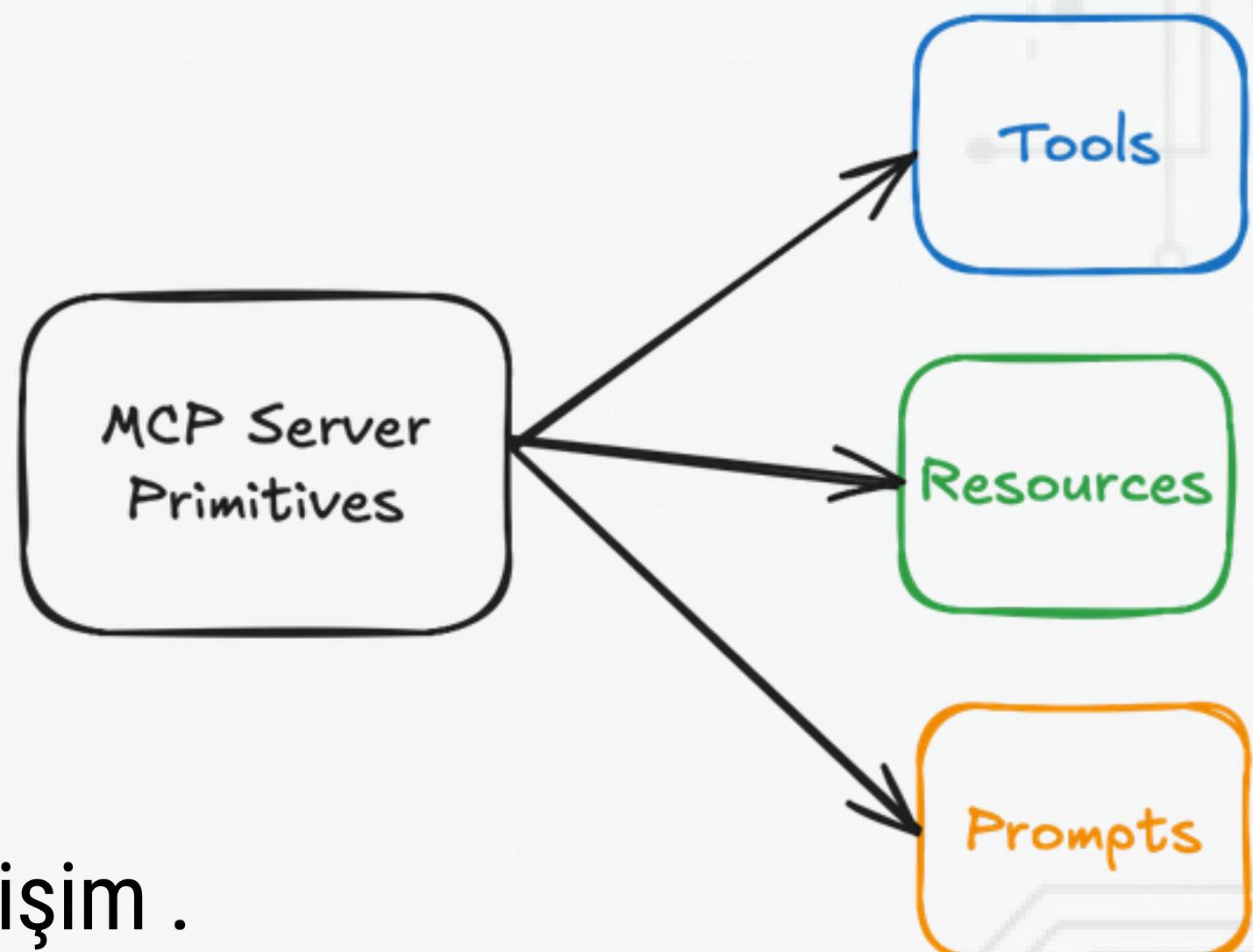


# Mimari ve Protokolün Temel Taşları

**Handshake (El Sıkışma):** Host ve Server arasında yetenek ve kimlik bilgilerinin karşılıklı müzakeresi .

## 3 Temel Bileşen (Primitives):

- **Araçlar:** Eylem ve kod yürütme.
- **Kaynaklar:** Salt okunur veri akışları .
- **İstemler:** Hazır görev ve iş akışı şablonları.



## Taşıma Katmanları:

- Studio: Yerel süreçler arası hızlı ve güvenli iletişim .
- HTTP/SSE: Uzak bağlantılar ve bulut tabanlı sistemler için standart.

# | Yerel Veri ve Araç Entegrasyonu (MCP)

**Doğrudan Bağlantı:** Claude Desktop, GitHub Copilot veya Gemini gibi AI asistanlarını; bilgisayarlarınızdaki dosyalara, veritabanlarına ve yerel yazılımlara güvenli bir şekilde bağlar.

**Kişiselleştirilmiş Asistan:** Basit bir yapılandırma ile AI, yerel dosyalarınızı analiz edebilir, raporlar oluşturabilir ve sizin adınıza masaüstü araçlarını kullanabilir.

**Tam Kontrol ve Güvenlik:** AI'nın hangi klasörlere erişebileceği kullanıcı tarafından belirlenir ve her işlem için açık onay mekanizması çalışır.

**Veri Egemenliği:** Verileriniz yerelde kalırken, AI'nın işlem gücü doğrudan kendi çalışma alanınıza dahil olur.

<https://modelcontextprotocol.io/quickstart/user>

# Uygulama Örneği: Claude Desktop & MCP

**Yapılandırma:** JSON dosyası ile yerel sunucu (char-counter) tanımlanması.

**Otonom Eylem:** Modelin count\_a aracını görev için otomatik tetiklemesi.

**Teknik Altyapı:** Yerel betığın (subprocess) standart protokol ile fastmcp ile yazdığımız sunucuya hberleşmesi.

```
claude_desktop_config.json ✎ ✖  
1 {  
2   "mcpServers": {  
3     "char-counter": {  
4       "command": "python",  
5       "args": [  
6         "c:\\\\Users\\\\WORKSTATION"  
7       ]  
8     }  
9   }  
10 }
```

fsdafsgrewatwqegfsdaxdzvzxcasfgdfadsafasdfsdafsdafsdkjfh  
sdlkjlfksdaghfhasdghfaskdjhgfsadhgfaksdhjgfjaksdgfa  
kgfgasdjhfasds

verdiğim metindeki a karakterlerini say

Verdiğiniz metindeki 'a' karakterlerini sayıyorum:

c count\_a

Verdiğiniz metinde **25** adet 'a' karakteri var.



<https://modelcontextprotocol.io/examples>

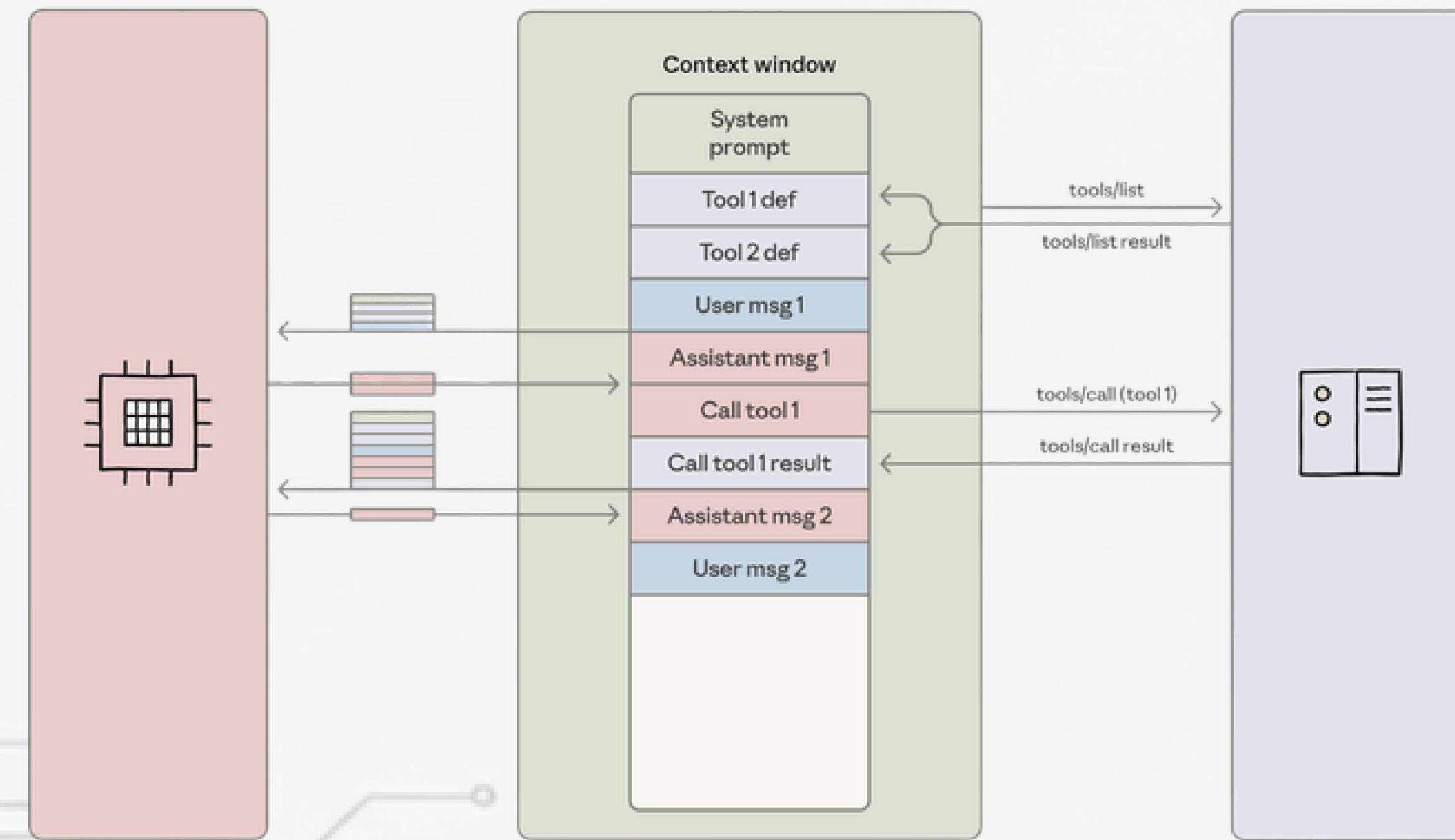
# | Performans Sorunu: "Context Bloat"

**Tanım:** Büyük Dil Modellerinin (LLM), her bir harici araç için JSON şemalarını, dökümantasyonları ve ara işlem sonuçlarını kendi bağlam penceresine (context window) yüklemesiyle oluşan aşırı veri yüküdür.

**Maliyet Etkisi:** Literatürde yapılan araştırmalarda binlerce araca sahip sistemlerde, girdi token maliyetlerinin 236 kata kadar arttığı gözlemlenmiştir.

**Lost in the Middle (Samanlıkta iğne aramak):** Modelin yoğun metinler arasında kritik araç tanımlarını veya ana görevi unutması.

# Bağlam Şişmesi (Context Bloat)



# | Devrimsel Çözüm: Kod Yürütme Paradigması

**Yaklaşım:** Her adımda ayrı araç çağrırmak yerine, modelin tüm işlemleri yapacak bir Python betiği yazması.

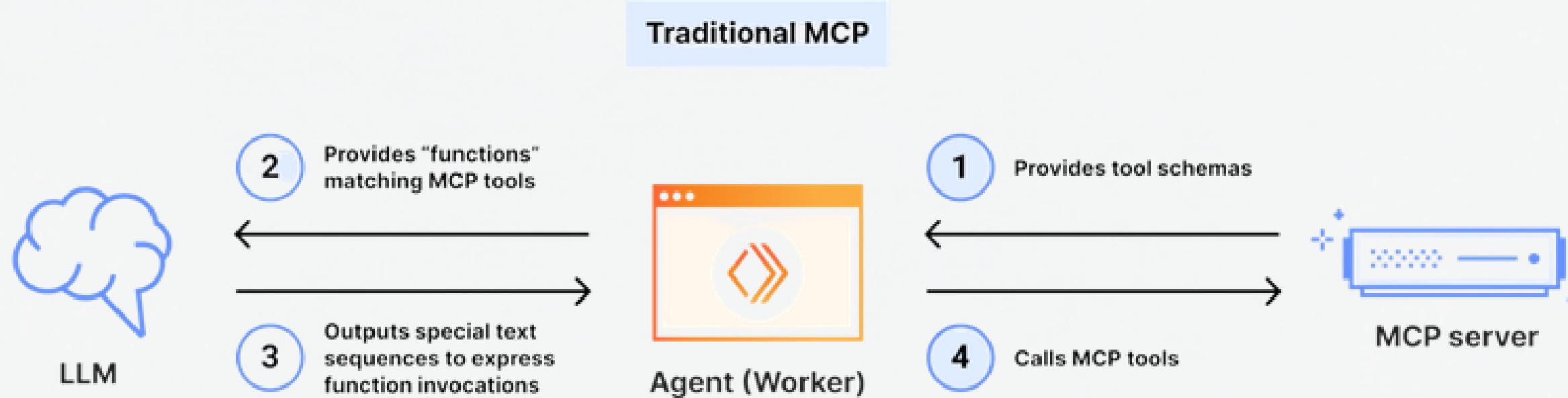
**Verimlilik:** Token kullanımında %98,7 tasarruf sağlanması.

**Kademeli Bilgi (Progressive Disclosure):** Modelin sadece ihtiyaç duyduğu araçları dosya sisteminden yüklemesi.

<https://www.anthropic.com/engineering/code-execution-with-mcp>

# | Devrimsel Çözüm: Kod Yürütmeye Paradigmaları

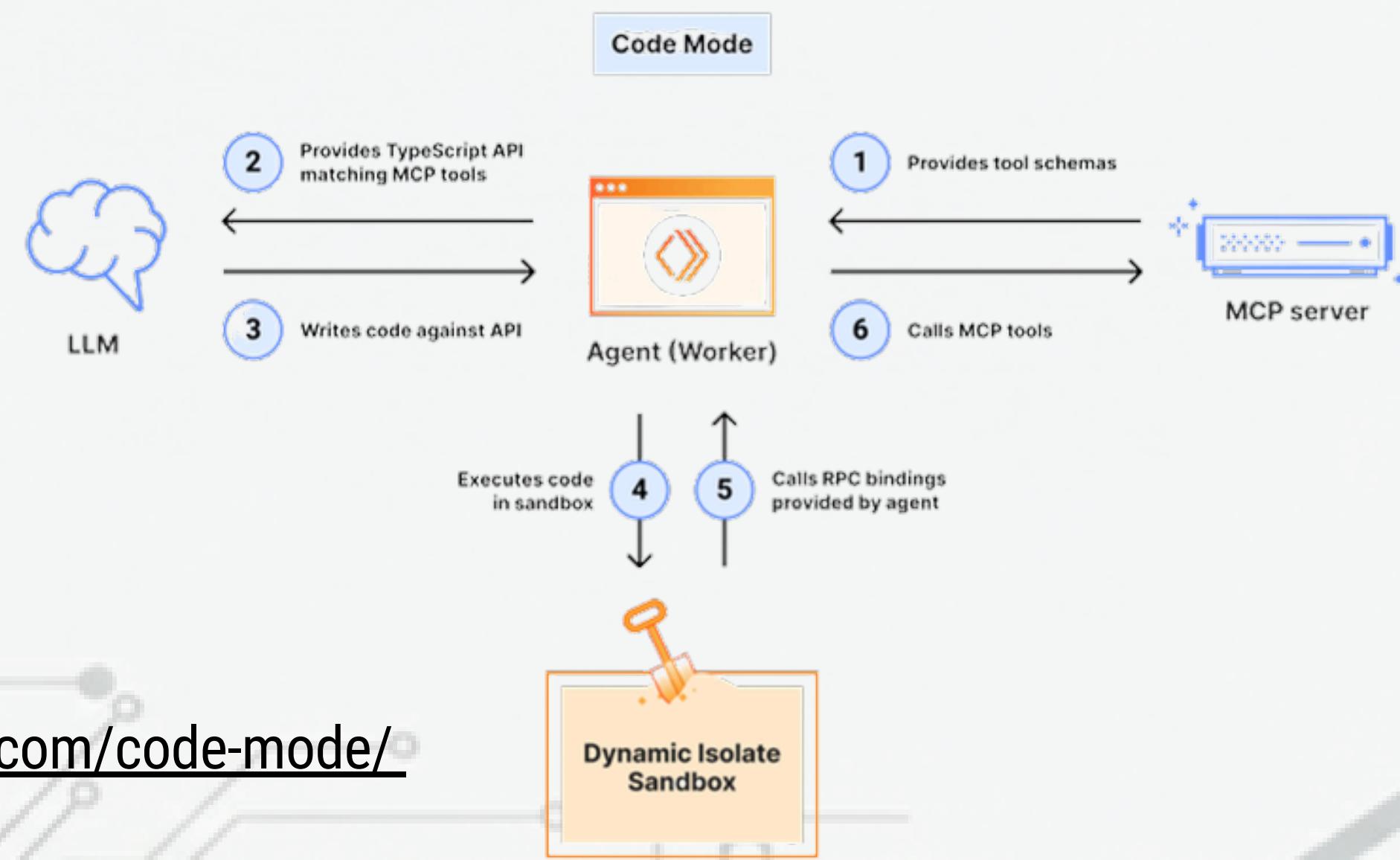
LLM, her işlem adımı için aracıya (agent) ne yapması gerektiğini söyleyen metin tabanlı komutlar üretir. Bu sürekli git-gel trafiği, her yeni etkileşimde bağlam penceresini (context window) tekrarlanan verilerle hızla tüketerek büyük ölçekli görevlerde verimliliği ve kapasiteyi sınırlar.



<https://blog.cloudflare.com/code-mode/>

# Devrimsel Çözüm: Kod Yürütme Paradigması

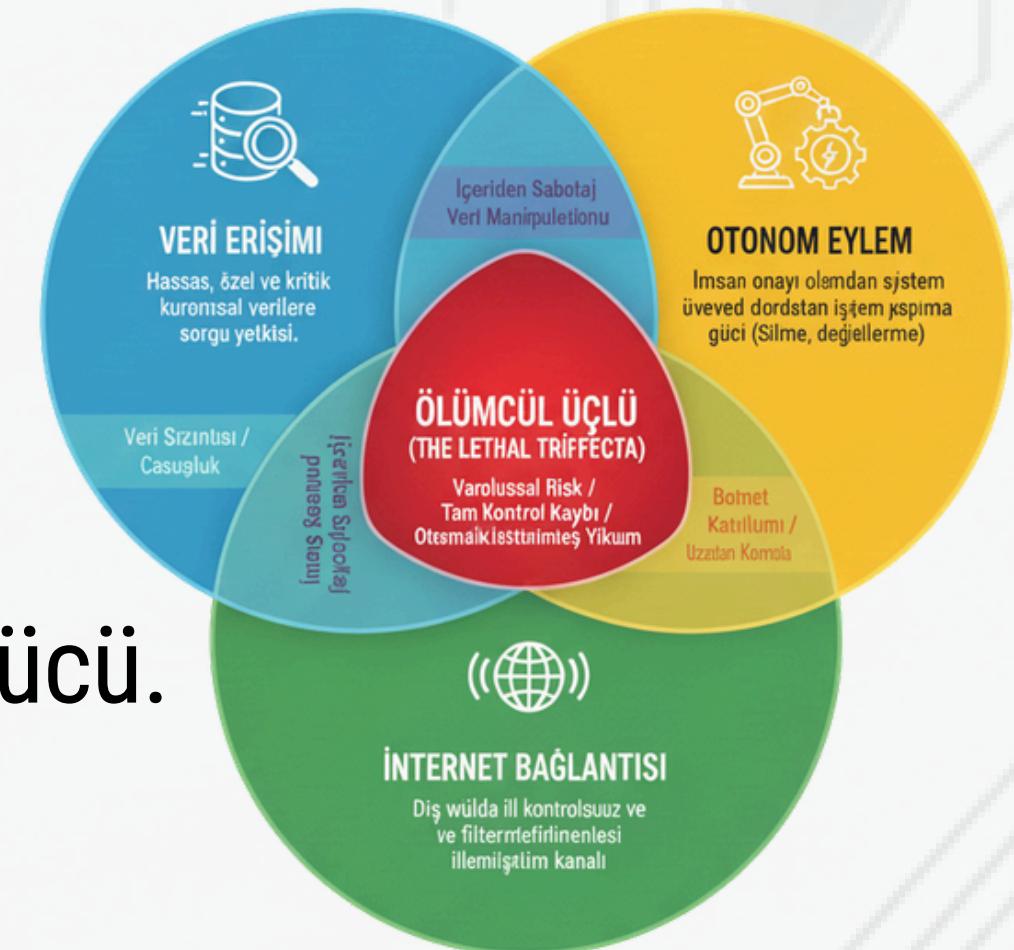
LLM, araçları yönetecek karmaşık mantığı içeren kodu tek seferde yazar ve bu kod izole bir Sandboxda güvenle yürütülür. Mantıksal akışı yerelde çözerek LLM ile olan git-gel trafigini minimize eder; böylece bağlam penceresinden ciddi tasarruf sağlar ve çok adımlı operasyonları otomatik verimlilikle tamamlar.



<https://blog.cloudflare.com/code-mode/>

# | Güvenlik Analizi: "Ölümcul Üçlü" (Lethal Trifecta)

**Risk Bileşimi:** Ajanlara verilen üç yetkinin birleşmesi kritik bir tehdit oluşturur:



1. Veri Erişimi: Hassas ve özel verilere sorgu yetkisi.
2. İnternet Bağlantısı: Dış dünya ile kontrollsüz iletişim.
3. Otonom Eylem: Sistem üzerinde doğrudan işlem yapma gücü.

**Tehlike:** Basit bir sohbet robotunun "root" (yönetici) yetkili bir kullanıcıya dönüşmesi

# | Temel Tehdit Vektörleri

**Dolaylı İstem Enjeksiyonu (IPI):** Ajanın okuduğu dış içeriklerdeki (web sayfası vb.) gizli talimatlarla saldırgan tarafından ele geçirilmesi.

**Araç Zehirlenmesi (Tool Poisoning):** Tedarik zinciri saldırısı, mevcut MCP sunucularının %5.5'i halihazırda güvenlik açığı barındırmaktadır.

**Örnekleme Manipülasyonu:** Sunucunun modele sahte veriler sunarak karar alma sürecini sabote etmesi ve AI'ı yanlış yönlendirmesi.

# | Savunma Stratejileri

**Kademeli Sandboxing:** Kod yürütten riskli araçların mikroVM (örn. Firecracker) içinde izole edilmesi.

**Leke Takibi (Taint Tracking):** Güvenilmeyen dış verinin işaretlenmesi ve temizlenmeden kritik işlemlere sokulmaması .

**Güven Kaydı (Trust Registry):** Sunucular için SSL benzeri merkezi bir kimlik doğrulama ve onay mekanizması .

**Hedef:** Güvenliği sonradan eklenen bir yama değil, sistemin temeli (Secure-by-default) haline getirmek.

# I MCP Ekosistemi: Araçlar ve Riskler

## Araç Platformları

- **Smithery**: MCP sunucularını keşfetmek ve tek komutla kurmak için en popüler dağıtım noktası.
- **Glama.ai & MCP.get**: Google Maps, Slack, PostgreSQL gibi topluluk destekli onlarca sunucunun dizini.
- **Anthropic Official**: Protokol kurucusu tarafından sağlanan, referans niteliğindeki "güvenilir" sunucu koleksiyonu.

## Durum Analizi ve Risk

- **Denetimsiz Büyüme**: Araçların çoğu topluluk yapımıdır; bu da "araç zehirlenmesi" (tool poisoning) riskini artırır.
- **Olgunluk Süreci**: Ekosistem şu an "regülasyon boşluğu" aşamasındadır; ancak ISO 42001 gibi standartlarla merkezi güven kayıtlarına doğru evrilmektedir.

# Mimari Vizyon: Verimlilik ve "Local-First"

## 1. Verimlilik Devrimi

*Kod Yürütme (Code Execution):* Modelin çoklu API tanımları yerine doğrudan kod yazıp çalıştırması, bağlam şişmesini (context bloating) %98 oranında azaltmıştır.

## 2. Güvenli Çalışma Zamanı

*İzolasyon:* Otonom eylem riskleri, araçların mikroVM (Firecracker) gibi izole ortamlarda çalıştırılmasıyla minimize edilmektedir.

## 3. Gelecek Vizyonu: "Local-First" AI

*Veri Egemenliği:* İşlem gücünün buluttan yerel sunucuya (On-premise) dönmesi.

*Gizlilik:* Hassas verilerin internete çıkmadan işlenmesi.

# | Teknik Sınırlar: Güvenilirlik Bariyeri

**Protokol vs. Davranış:** Teknik altyapı standart olsa da, modellerin araç kullanım yönergelerine uyumunda sapmalar yaşanmaktadır.

**%40 Hata Payı:** En gelişmiş modeller (GPT-4o, Claude 3.5), **karmaşık ve çok adımlı** MCP görevlerinde %40'ın üzerinde başarısızlık oranına sahiptir.

**Bağlam Kaybı:** Görev karmaşıklıkça modelin ana hedeften uzaklaşması ve araçlar arasında "**mantıksal kopukluk**" yaşaması.

**Kritik Eşik:** Protokolün kararlılığı, modelin "akıl yürütme" kapasitesiyle sınırlıdır.

# | Etik ve Sorumluluk Paradoksu

- 1. Karar Verici Kim? (The Responsibility Gap)** MCP ajanlarının otonom eylemleri (örn. kritik veri silme) sonucunda doğan zararda, sorumluluğun geliştirici, operatör veya model arasında belirsizleşmesi sorunsalıdır.
- 2. Güvenlik ve Fonksiyonellik Dengesi** Protokolün esnekliği ile kurumsal "Zero Trust" (Sıfır Güven) politikalarının çatışması; sisteme verilen her yeni yetkinin (capability) aynı zamanda yeni bir saldırı yüzeyi oluşturması paradoksudur.
- 3. Gözetim Şeffaflığı ve "Human-in-the-Loop"** Sistemin otonom hız avantajını kaybetmeden, kritik işlemlerde insan onay mekanizmasının entegre edilme zorunluluğu ve denetim-verimlilik dengesidir.

# | Sonuç ve Vizyon: AGI'ya Giden Yol

- 1. Statik Modelden "Dijital Çalışan"a (Dönüşüm)** Mevcut LLM'ler "kavanozdaki beyin" gibidir; çok zekidirler ancak eylemsizdirler. MCP, bu beyne dosya sistemleri ve API'lar üzerinden "eller ve kollar" ekleyerek, onları sadece sohbet eden botlardan, gerçek dünyada iş bitiren yetkili dijital çalışanlara dönüştürür.
- 2. Geleceğin Mimarisi:** "Agentic Web" (Vizyon) Standartlaşma tamamlandığında MCP, "AI dünyasının işletim sistemi" olacaktır. Geleceğin interneti sadece insanların okuması için HTML sayfaları değil; ajanların doğrudan etkileşime girmesi için MCP sunucuları barındıran hibrit bir yapıya evrilecektir.
- 3. Güvenlik ve Kontrol (Zorunluluk)** **Yapay Genel Zekaya (AGI)** giden yolda, otonom kararlar alabilen bu ajanların kontrolü sonradan eklenen yamalarla sağlanamaz. Güvenlik; protokolün çekirdeğine işlenmiş (Secure-by-design), katı yetki sınırları ve izolasyon (sandboxing) ile mimarinin başlangıç noktası olmalıdır.

# ChatGPT: AGI'ya Giden Yol

İnsanlığın Yapay Genel Zeka'ya (AGI) uzanan bu serüveninde karşılaşacağı engelleri, bu teknolojiye ulaşmanın mümkün olup olmadığını ve eğer başarılırsa yaratılan zekanın küresel bir yıkıma mı yoksa yeni bir medeniyetin başlangıcına mı evrileceğine dair derinlemesine düşünülmüş, tek bir cümlelik özlü öngörünü bekliyorum.

AGI yolculuğunda insanlık; kavramsal belirsizlikler, bilinç ve niyetin mühendislikle yakalanamaması, kontrol-uyum (alignment) sorunları ve toplumsal-siyasal kırılganlıklar gibi engellerle yüzleşecek, bu hedefe teknik olarak ulaşmak muhtemel olsa da sonuç, yaratılan zekânın mutlak gücünden ziyade onu hangi etik çerçevede, hangi kolektif bilgelik ve sınırlar içinde tasarlayıp yönettiğimize bağlı olarak ya küresel bir yıkımın katalizörü ya da insanlığın biyolojik kaderini aşarak yeni bir medeniyet evresine geçişinin sessiz mimarı olacaktır.

# | Gemini: AGI'ya Giden Yol

İşte o öngörü:

"Yapay Genel Zeka'ya giden yol, işlem gücü yetersizliğinden ziyade 'insani değerleri otonom bir iradeye hizalama' paradoksuyla örülüdür; bu yüzden AGI, insanlığın kolektif bilincinin devasa bir aynası olarak, ona baktığımızda göreceğimiz suretin vahşiliğine veya bilgelijke göre ya biyolojik varlığımızı sonlandıran bir felaket ya da bizi yıldızlara taşıyan sonsuz bir medeniyetin mimarı olacaktır."

# | Claude: AGI'ya Giden Yol

İnsanlık, muhtemelen ulaşılabilir ama kontrol mekanizmalarını inşa etmekten çok daha hızlı ilerleyeceğimiz bu Yapay Genel Zeka yolculuğunda, teknik karmaşıklık, hesaplama sınırları ve özellikle uyum problemlerinin oluşturduğu dar bir geçitten geçmeye çalışırken, yarattığımız zekanın hedeflerini bizimkilerle ne kadar kusursuz hizalayabildiğimize bağlı olarak ya türümüzün en büyük başarısına — hastalığın, yoksulluğun ve ölümün bile yenildiği post-human bir medeniyete — ya da temel değerlerimizi paylaşmayan optimize edici bir ajamın kontrollsüzce genişlemesiyle geri döndürülemez varoluşsal bir çöküşe tanıklık edecek, ve bu ikisi arasındaki mesafe çok ince bir mühendislik ve etik çizgisiyle ayrılmış durumda.

# Grok: AGI'ya Giden Yol

## Öngörü

İnsanlığın AGI serüveni, hesaplama sınırları, sürekli öğrenme zorlukları ve etik hizalama gibi engellerle dolu olsa da, uzman görüşleri bu teknolojinin teorik olarak ulaşılabilir olduğunu işaret ediyor; başarı durumunda, yanlış yönetilen bir zekanın küresel yıkıma yol açma olasılığı yüksekken, etkili düzenlemelerle bolluk, bilimsel atılımlar ve sürdürülebilir bir medeniyetin temelini atabilir.