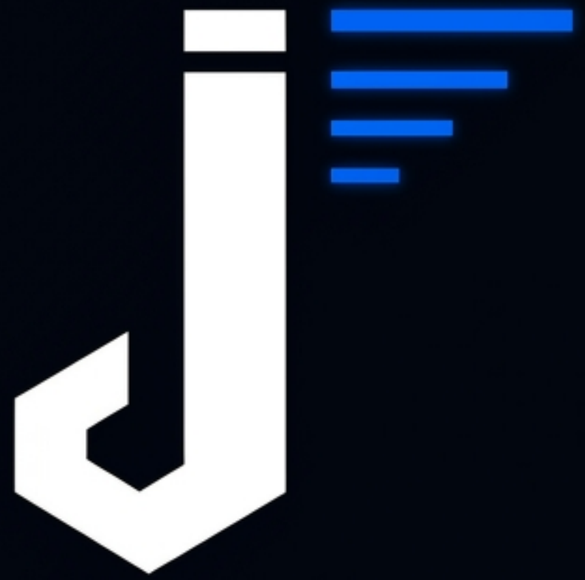


اصلاً API یعنی چی؟

فرض کن وارد رستوران شدی. تو غذا می‌خوای، آشپز غذا رو درست می‌کنه، اما گارسون سفارش رو بین شما رد و شما رد و بدل می‌کنه.

API هم دقیقاً نقش همین گارسون رو داره. (یعنی دو نرم‌افزار از طریق API با هم حرف می‌زنن).

چرا چند نوع API داریم؟

همه برنامه‌ها نیاز یکسانی ندارن.

حجم اینترنت
(مصرف بهینه داده).

سرعت براشون مهمه
(مثل اسنپ برای موقعیت لحظه‌ای راننده).

سادگی
(مثل اپلیکیشن هواشناسی).

انعطاف
(مثل فروشگاه اینترنتی).

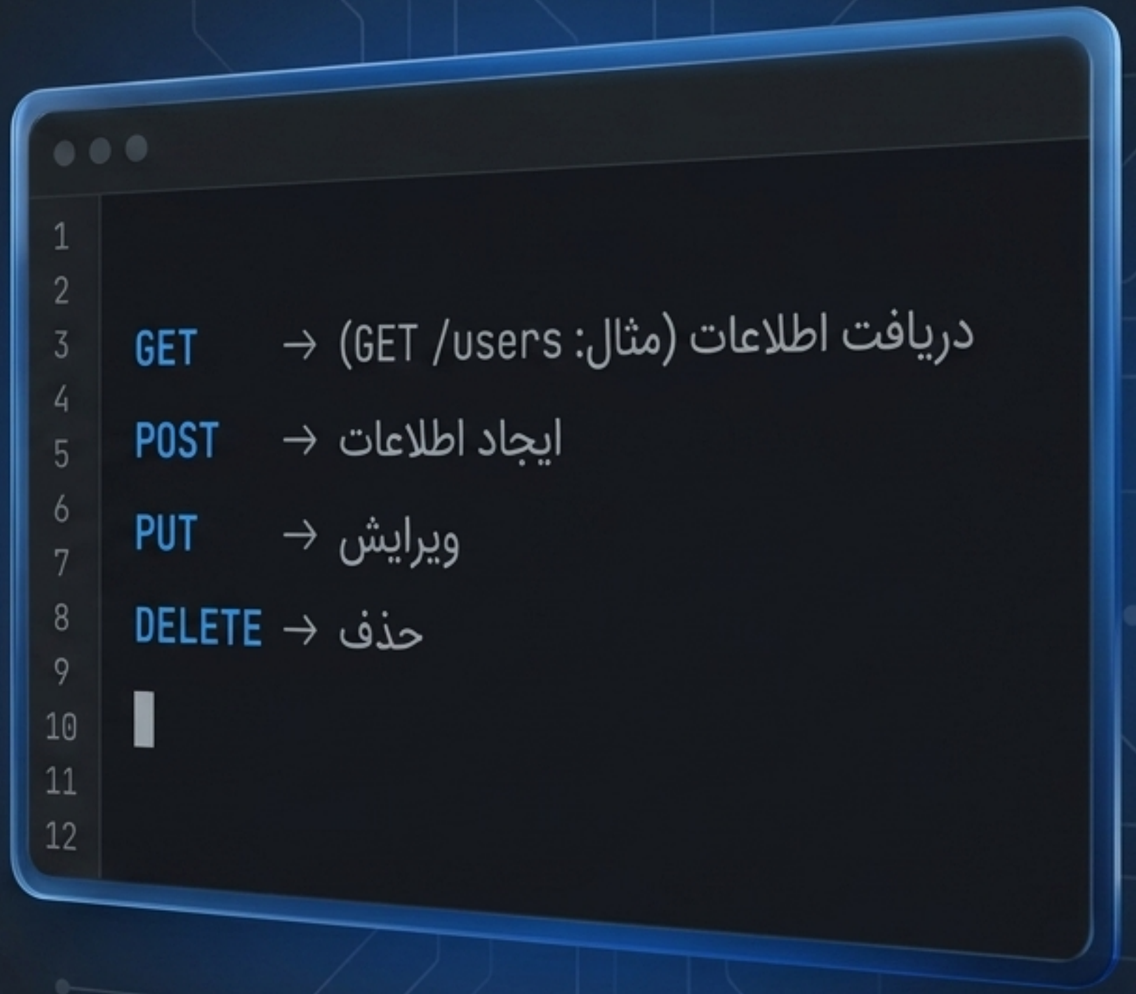
به همین دلیل، مدل‌های مختلفی برای طراحی API به وجود اومده.

معماری REST (استاندارد جهانی)

در REST هر چیزی یک «**منبع**» (Resource) محسوب می‌شود (مثل: کاربران، محصولات، سفارش‌ها).

مزایا: ساده، یادگیری راحت، مناسب اکثر پروژه‌ها.

معایب: گاهی اطلاعات اضافی برمی‌گرداند.

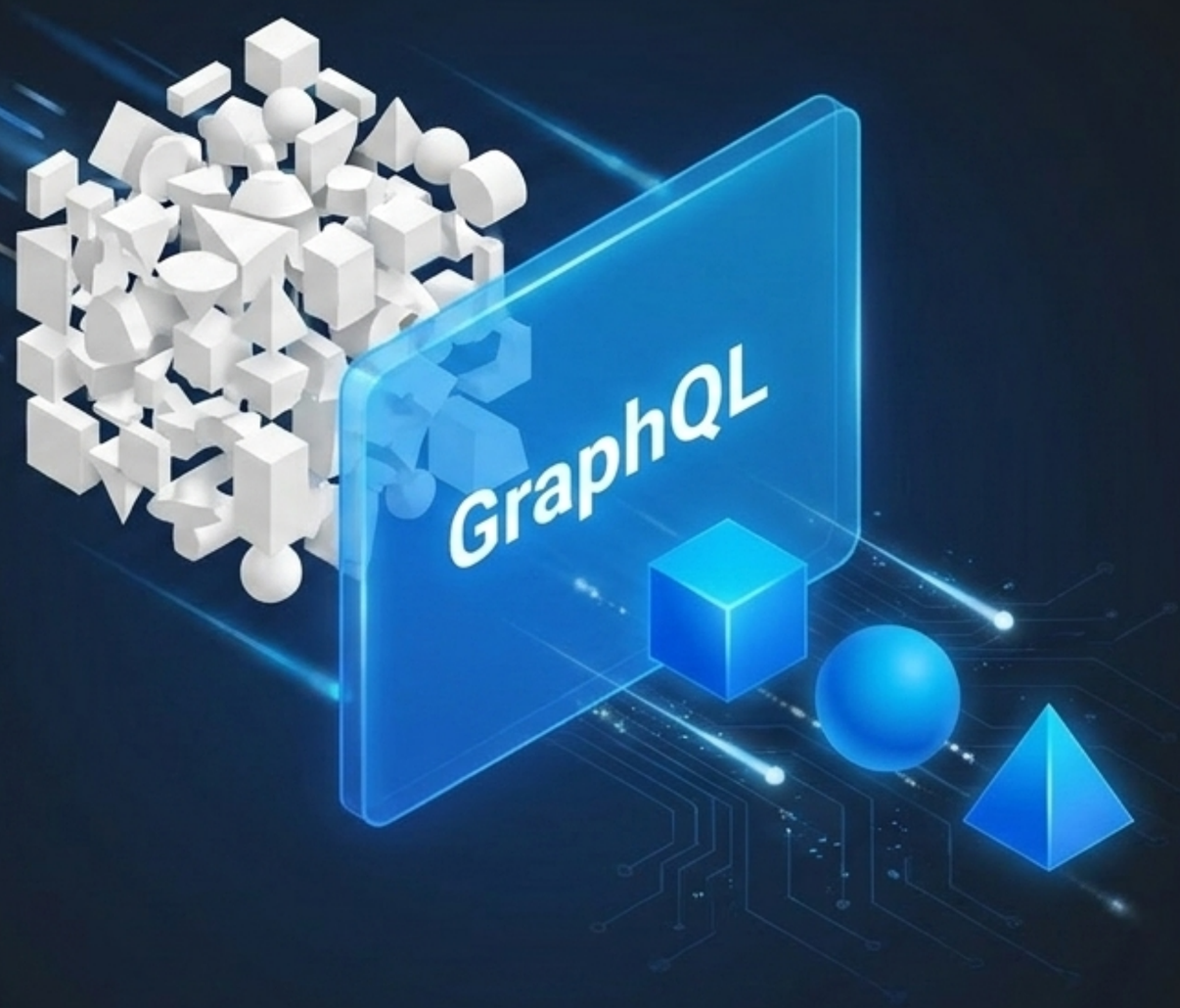


معماری GraphQL (دقت مطلق)

اینجا کلاینت خودش دقیقاً مشخص می‌کند چه اطلاعاتی می‌خواهد. نه کمتر، نه بیشتر.

مزایا: انعطاف بالا، دریافت دقیق اطلاعات، مناسب اپلیکیشن‌های بزرگ.

معایب: پیاده‌سازی سخت‌تر، مدیریت امنیت پیچیده‌تر.



تفاوت ساختاری: مشکل Over-fetching

در REST ممکن است ۱۰ فیلد دریافت کنی، در حالی که فقط ۳ مورد لازم داری (مصرف اینترنت بیشتر، سرعت کمتر).

اما در GraphQL فقط همان ۳ مورد برمی‌گردد (اینترنت کمتر، سرعت بیشتر، بدون داده اضافه).

معماری gRPC (نهایت سرعت) By Google

هدف اصلی: سرعت بسیار بالا. به جای JSON از فرمت فشرده‌تری به نام Protocol Buffers استفاده می‌کند.

- ارتباط بین سرورها
- سیستم‌های بانکی
- بازی‌های آنلاین
- سرویس‌های ابری

مزایا: بسیار سریع و حجم کم داده. معایب: یادگیری سخت‌تر است.

ماتریس مقایسه جامع معماری‌ها

	REST	GraphQL	gRPC
فرمت دیتا	JSON	اختصاصی Query	Protocol Buffers
بهترین کاربرد	محبوب و عمومی	مناسب داده‌های پیچیده	مناسب Microservices
ویژگی کلیدی	ساده	انعطاف بالا	سرعت بالا

دروازه ورودی API Gateway

فرض کن یک ساختمان بزرگ داری. به جای اینکه هر کسی مستقیم وارد هر اتاق شود، اول از نگهبانی عبور می‌کند.

تمام درخواست‌ها پیش از ورود به سیستم، ابتدا وارد Gateway می‌شوند.

- امنیت بیشتر
- احراز هویت
- مدیریت ساده‌تر
- امنیت بیشتر
- به قوردرهنا
- محدود کردن تعداد درخواست‌ها
- ثبت لاگ

معماری سه‌لایه Gateway

۱. دریافت (Gateway درخواست کلاینت را می‌گیرد).

۲. بررسی (احراز هویت و امنیت چک می‌شود).

۳. ارسال (به سرویس مناسب در بک‌ند فرستاده می‌شود).

چالش جدید: عصر هوش مصنوعی

مشکل یکپارچه‌سازی ابزارها

قبلاً برای اینکه یک مدل هوش مصنوعی به داده‌های ما دسترسی پیدا کند، باید برای هر ابزار یک اتصال جداگانه می‌نوشتیم. هر کدام روش و پیچیدگی خودش را داشت.

پروتکل MCP (زبان مشترک هوش مصنوعی)

Model Context Protocol

یک استاندارد برای ارتباط مدل‌های هوش مصنوعی با ابزارها و داده‌ها.

MCP یک زبان مشترک ایجاد می‌کند تا مدل هوش مصنوعی بتواند به شکل استاندارد به فایل‌ها، دیتابیس و سرویس‌ها وصل شود.

جمع‌بندی کاربردی (کدام تکنولوژی برای کجا؟)

REST	ساده و مناسب اکثر پروژه‌ها	(مثال: دیجی‌کالا)
GraphQL	دریافت دقیق اطلاعات موردنیاز	(مثال: اینستاگرام)
gRPC	ارتباط سریع بین سرویس‌ها	(مثال: بازی‌های آنلاین)
API Gateway	مدیریت و امنیت متمرکز	(مثال: اسنپ)
MCP	استاندارد اتصال منابع به هوش مصنوعی	(مثال: ChatGPT)

جاودان