



جاودان

WHERE ENGINEERING MEETS INTELLIGENCE.

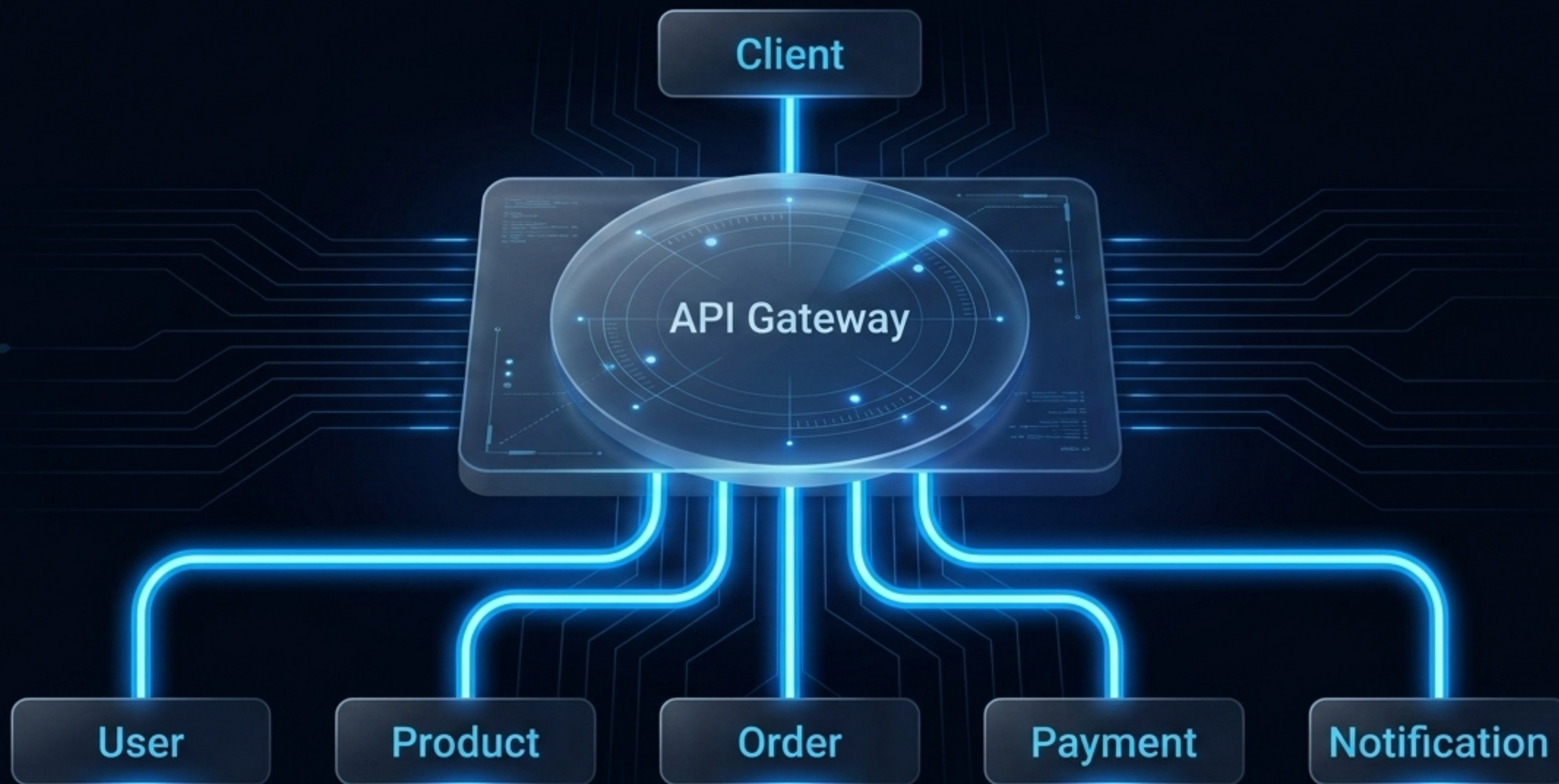
قبل از API Gateway چه مشکلی داشتیم؟





دروازه هوشمند ورود به معماری Microservices

یک لایه بین Client و Microservice ها که مانند پذیرش یک بیمارستان یا برج مراقبت فرودگاه عمل می‌کند.





Routing: اولین مسئولیت هر Gateway

Gateway مانند تابلو راهنمای فرودگاه عمل می‌کند و تصمیم می‌گیرد درخواست به کدام سرویس ارسال شود.



اعمال سیاست‌های امنیتی در یک نقطه. اگر کاربر معتبر نباشد، درخواست وارد سیستم نمی‌شود.

چه کسی هستید؟



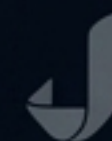
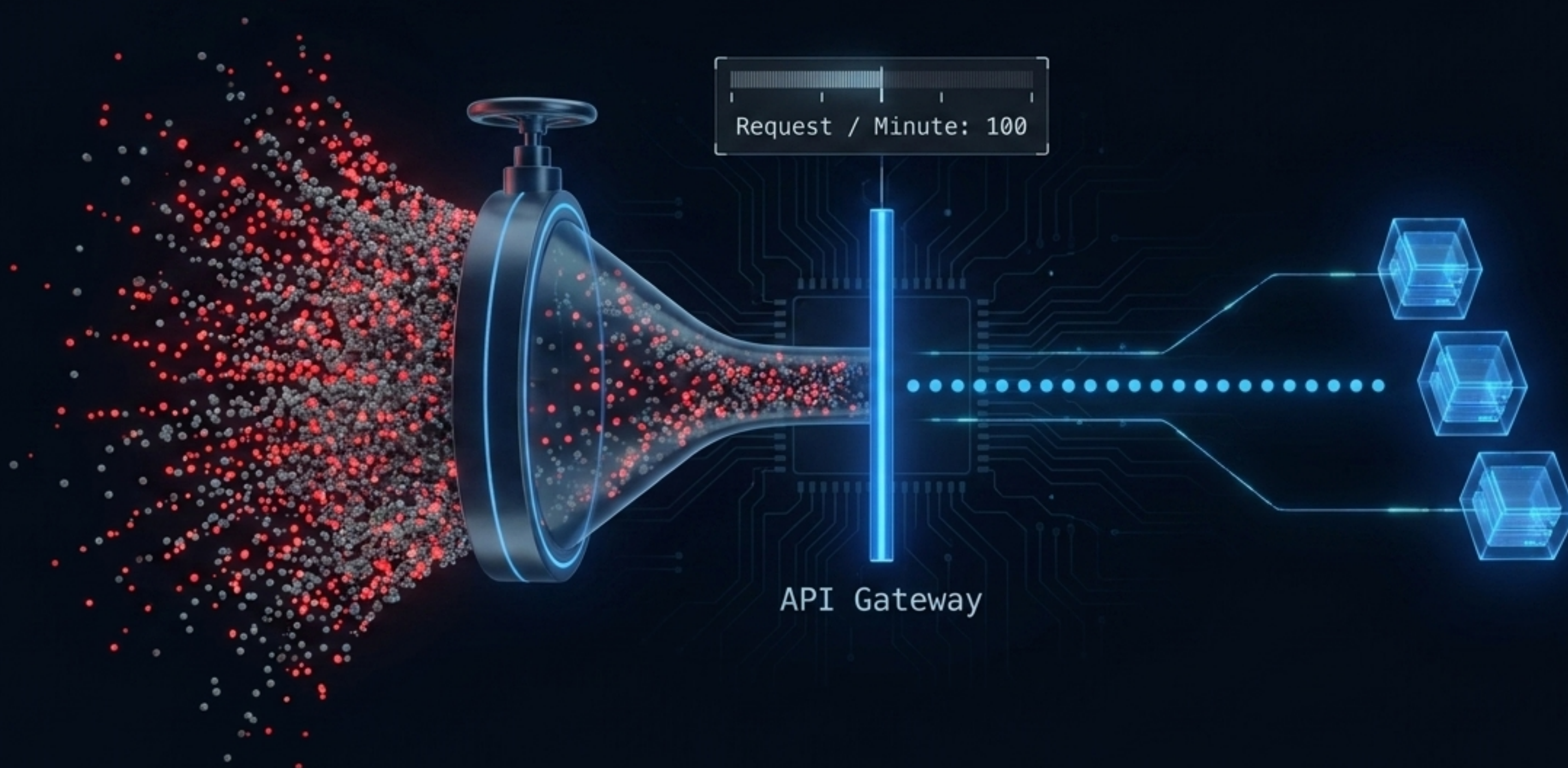
چه دسترسی‌هایی دارید؟





محافظت در برابر فشار بیش از حد

جلوگیری از Down شدن سیستم و کندی پاسخ‌ها در زمان هجوم درخواست‌ها.



بهترین نقطه برای مشاهده کل سیستم. در زمان Debug کردن، این لاگ‌ها بسیار ارزشمند هستند.



Error Rate:

0.02%

Status Code:

200 OK

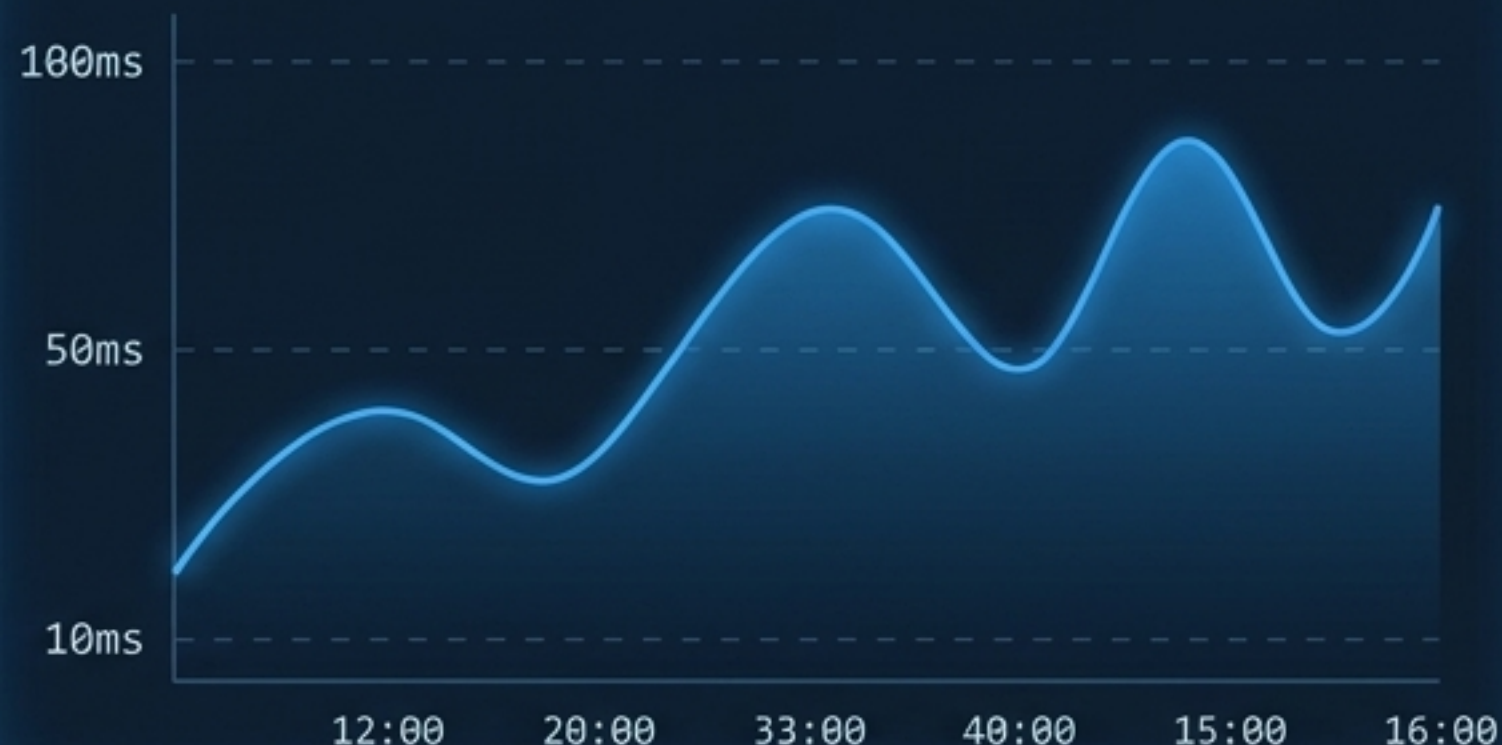
API Usage:

84%

Uptime:

99.99%

Response Time

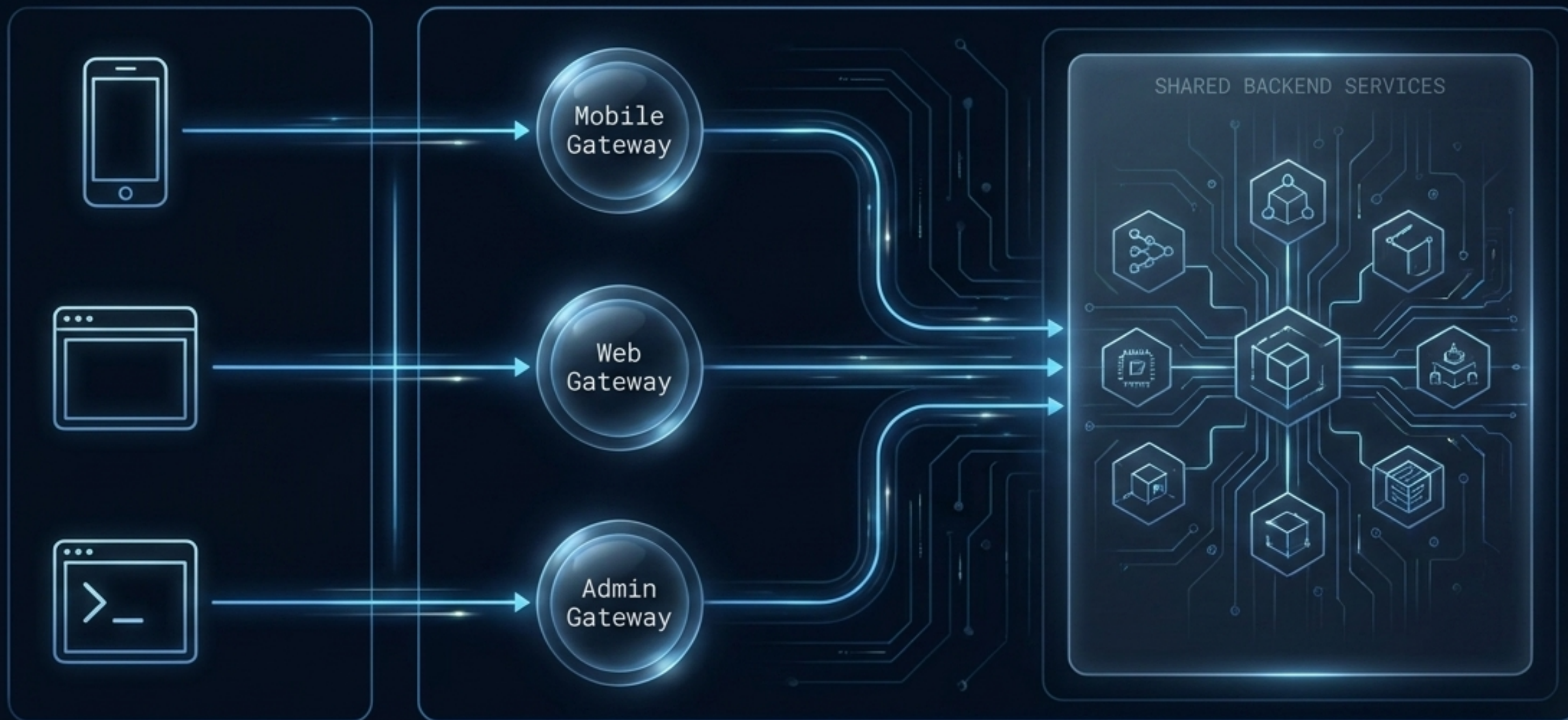


```
[TIME: 14:32:01] [IP: 192.168.1.1] [USER: admin_44] [ENDPOINT: /api/v1/orders] [STATUS: 200] [LATENCY: 42ms]
[TIME: 14:32:03] [IP: 192.168.1.5] [USER: operator_12] [ENDPOINT: /api/v1/products] [STATUS: 200] [LATENCY: 38ms]
[TIME: 14:32:05] [IP: 192.168.1.1] [USER: admin_44] [ENDPOINT: /api/v1/users] [STATUS: 200] [LATENCY: 45ms]
[TIME: 14:32:08] [IP: 192.168.1.9] [USER: guest_88] [ENDPOINT: /api/v1/status] [STATUS: 200] [LATENCY: 20ms]
```


پاسخ تمام سرویس ها : Aggregation Pattern جمع آوری شده و یک Response واحد برمی گردد.



الگوی BFF: نیازهای نسخه موبایل و وب یکسان نیست.
هر Frontend، یک API مخصوص به خود دارد.





Gateway فقط یک Load Balancer یا Router ساده نیست.

	Load Balancer	API Gateway
توزیع درخواست	✓	✓
Routing هوشمند	✗	✓
احراز هویت	✗	✓
محدودیت نرخ (Rate Limiting)	—	✓
مدیریت API	—	✓

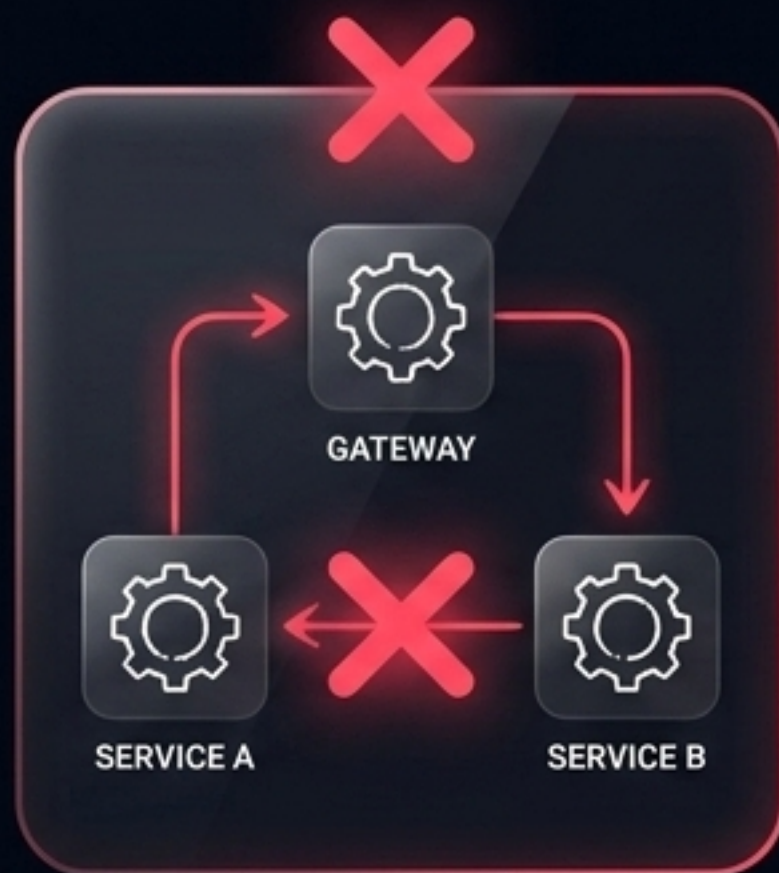


بزرگ‌ترین اشتباهات و ضدالگوها

این موارد بارها توسط توسعه‌دهندگان به عنوان ضدالگو مطرح شده‌اند.



قرار دادن منطق کسب‌وکار
داخل Gateway



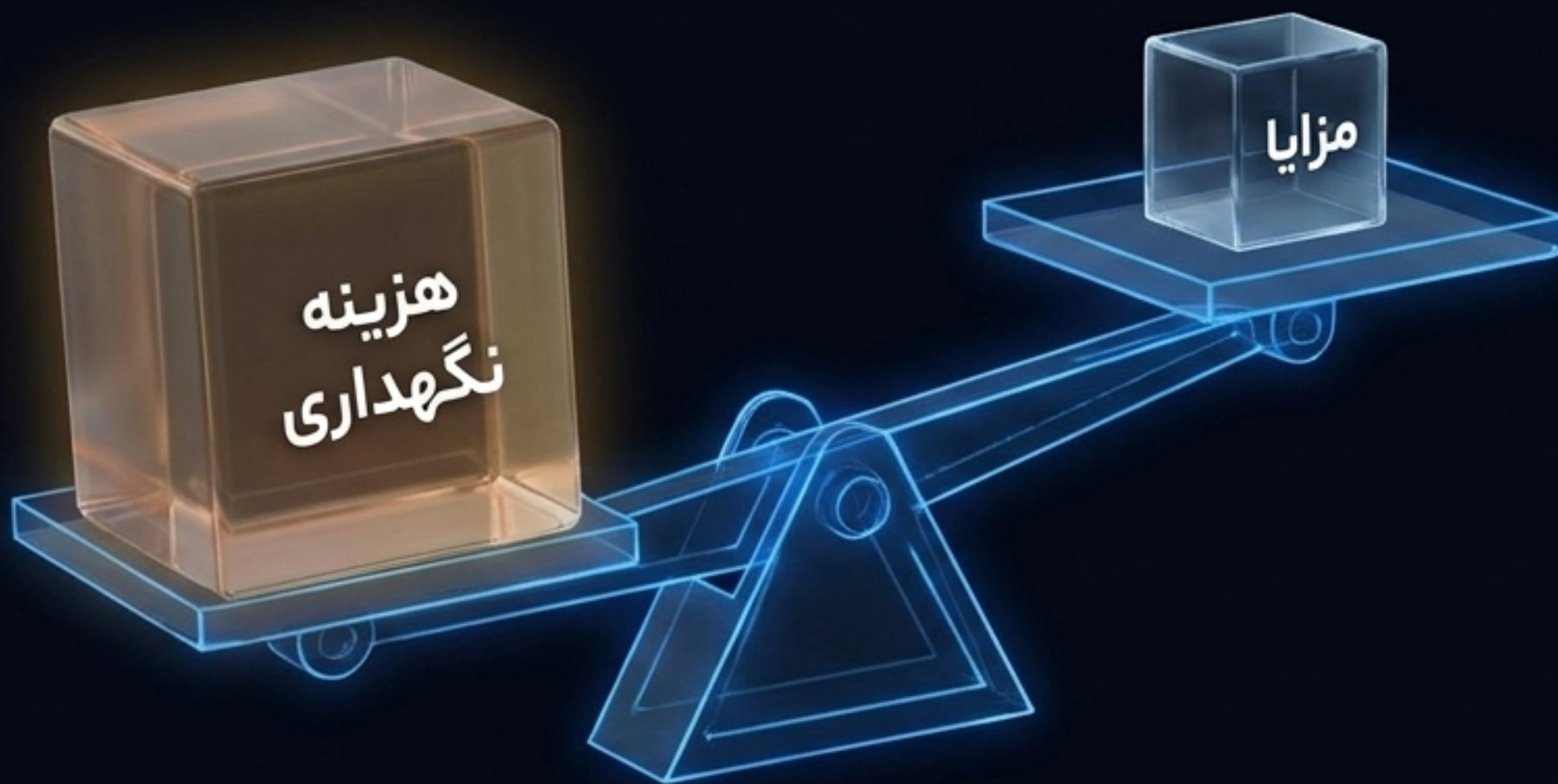
عبور ارتباط بین سرویس‌ها
از Gateway



ساخت یک Gateway بسیار
بزرگ و متمرکز

آیا همیشه به API Gateway نیاز داریم؟ خیر.

خیر. ممکن است هزینه نگهداری Gateway بیشتر از مزایای آن باشد.



در این شرایط استفاده نکنید:

- سیستم کوچک است
- فقط چند API دارید
- تنها یک Client وجود دارد

محبوب‌ترین ابزارهای معماری

هر کدام برای سناریوهای متفاوتی مناسب هستند.

Kong

**NGINX
Gateway**

**Spring Cloud
Gateway**

**AWS API
Gateway**

**Azure API
Management**

**Envoy
Gateway**

Traefik



اصل طلایی معماری

Gateway باید لایه‌ای سبک برای مدیریت درخواست‌ها باشد، نه محلی برای پیاده‌سازی منطق کسب‌وکار.

رعایت این اصل باعث می‌شود معماری در بلندمدت مقیاس‌پذیرتر، قابل نگهداری‌تر و پایدارتر باقی بماند.



