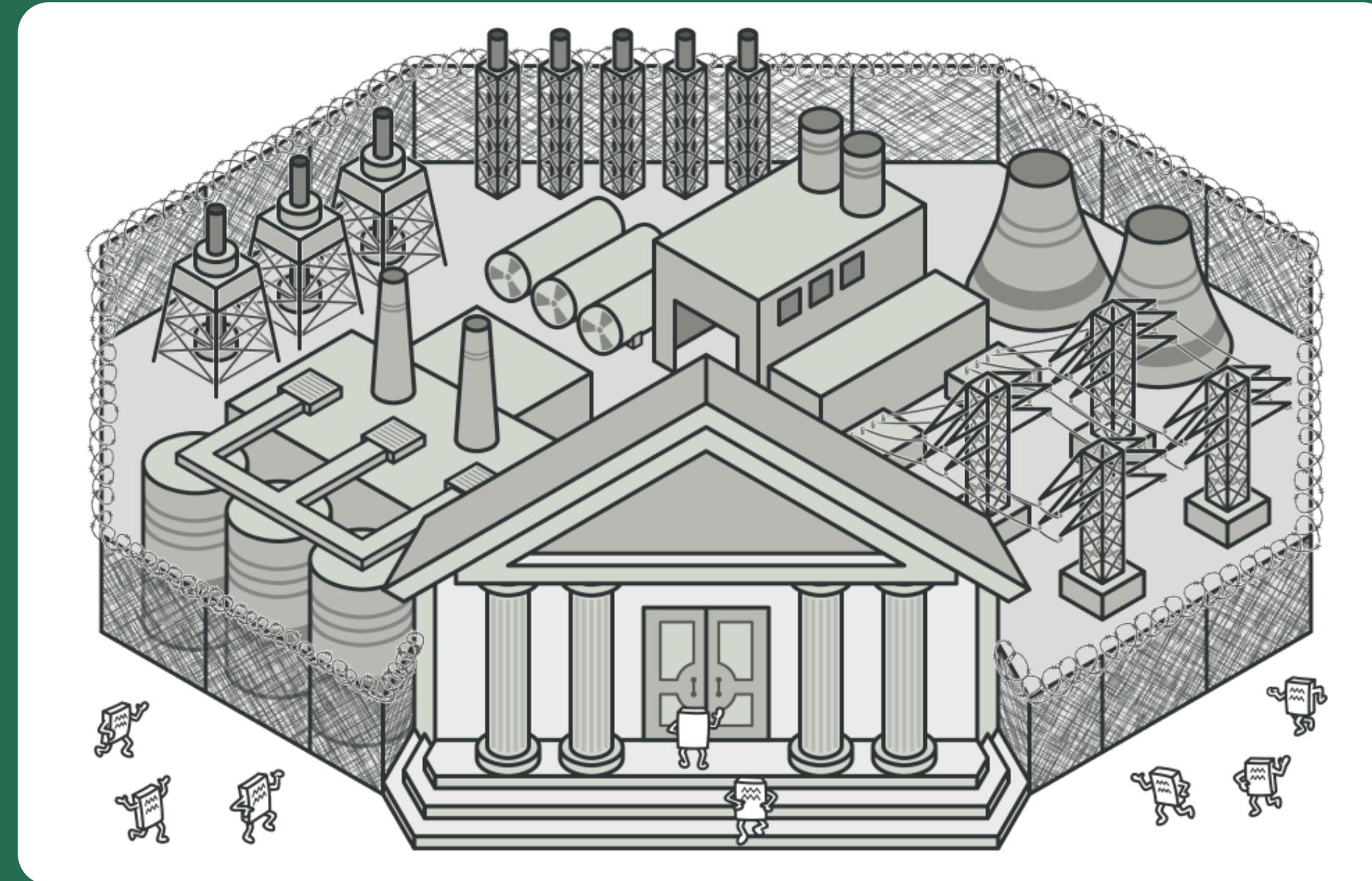


الگوی FACADE



PUBLISHED BY : FATEMEH AHMADZADEH

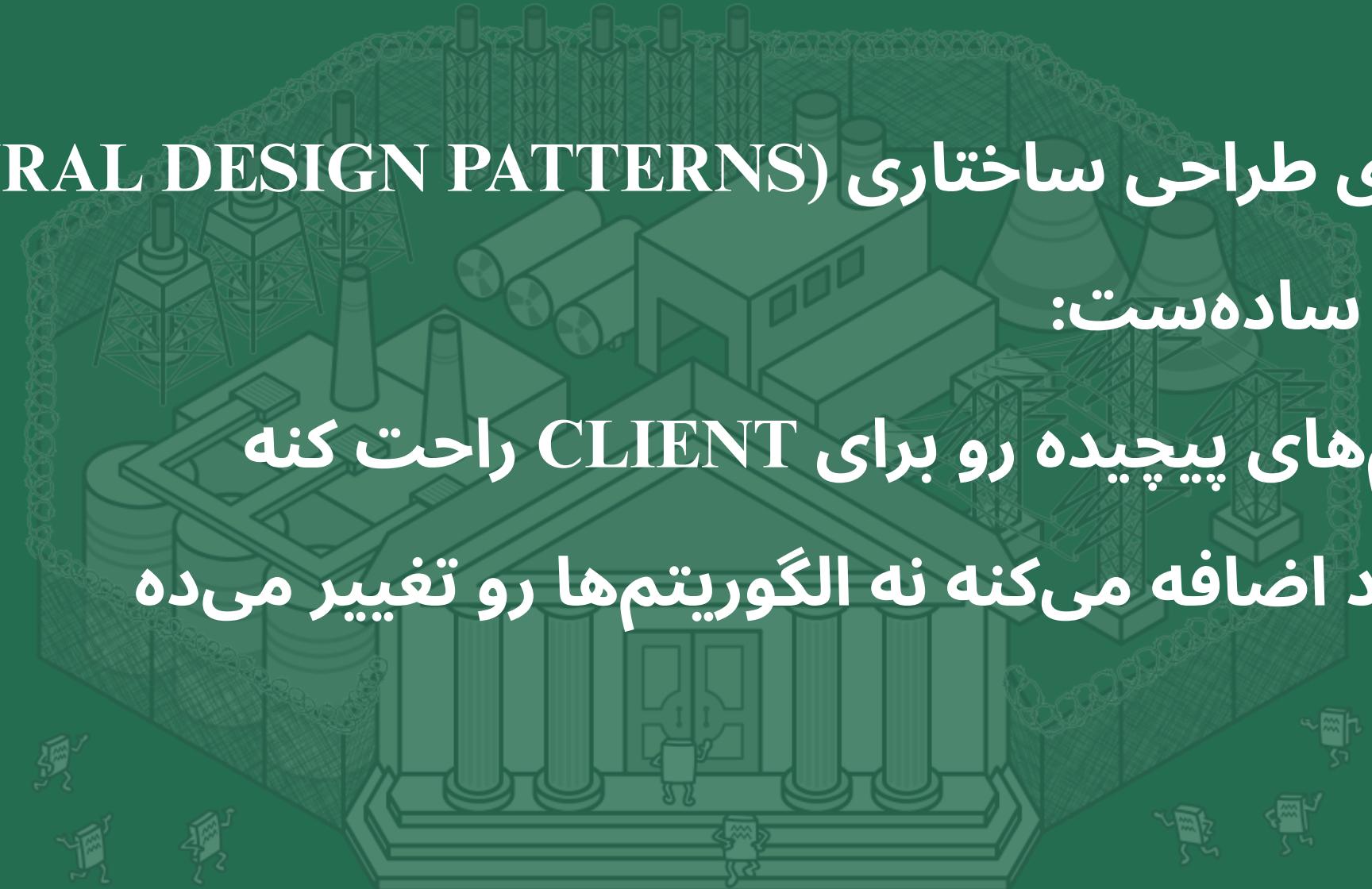
معرفی الگوی FACADE

یکی از الگوهای طراحی ساختاری (STRUCTURAL DESIGN PATTERNS)

هدفش خیلی سادهست:

کار با سیستم‌های پیچیده رو برای CLIENT راحت کنه

نه منطق جدید اضافه می‌کنه نه الگوریتم‌ها رو تغییر می‌ده



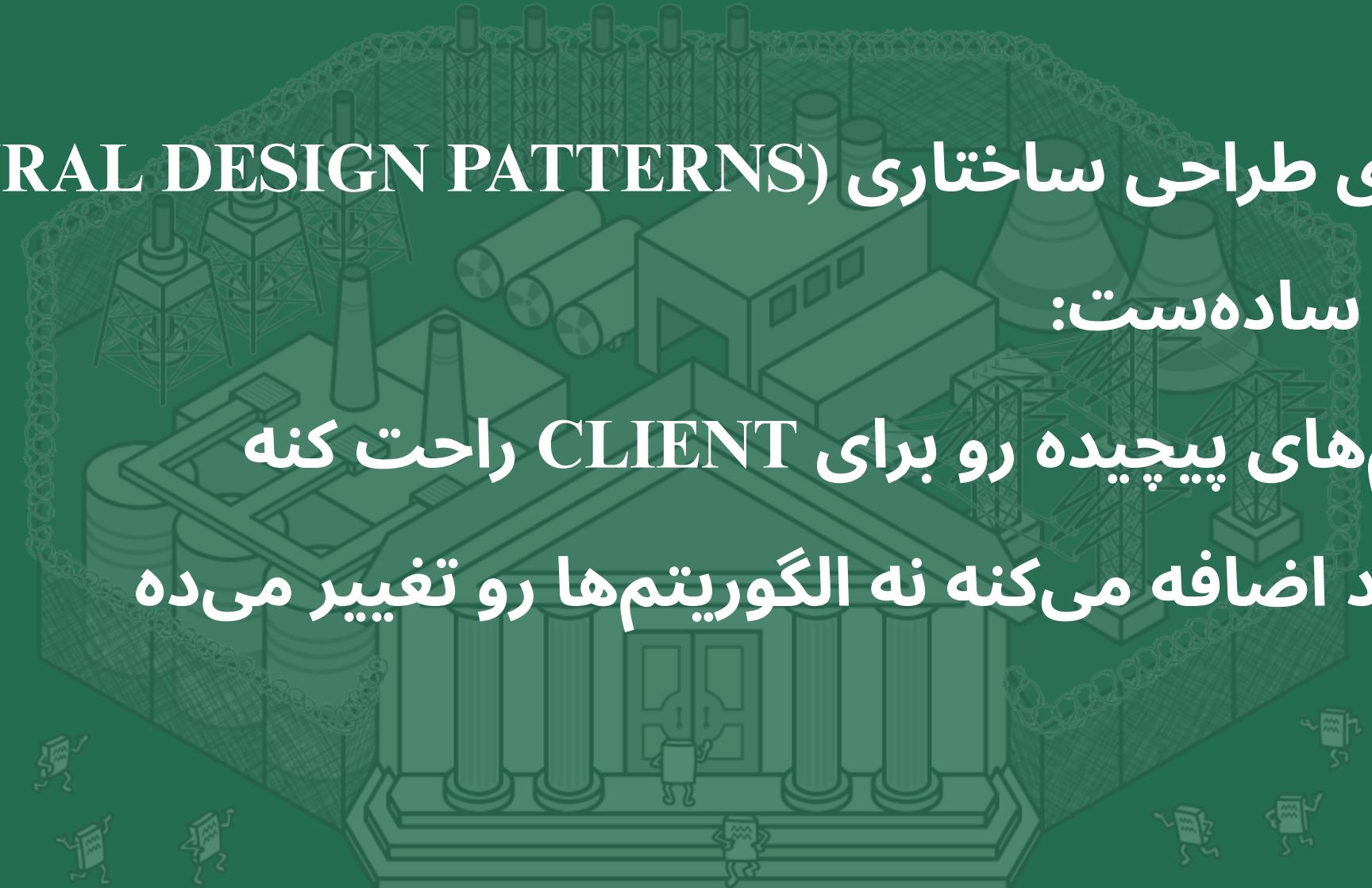
معرفی الگوی FACADE

یکی از الگوهای طراحی ساختاری (STRUCTURAL DESIGN PATTERNS)

هدفش خیلی سادهست:

کار با سیستم‌های پیچیده رو برای CLIENT راحت کنه

نه منطق جدید اضافه می‌کنه نه الگوریتم‌ها رو تغییر می‌ده



درد مشترک ما دولوپرها

تا حالا شده برای یک کار ساده مجبور باشی:

◦ چند تا کلاس مختلف بسازی؟

◦ کلی تنظیم انجام بدی؟

◦ متدها رو با ترتیب خاص صدا بزنی؟

◦ یک اشتباه کوچیک = کلی باگ 😬



ایدهی اصلی FACADE

«Facade میگه:

«تو جزئیات رو فراموش کن، فقط این متده را صدا بزن»

مثل یک ریموت کنترل:

◦ دکمه هاش کمه

◦ ولی پشتیش کلی سیستم کار من کنه



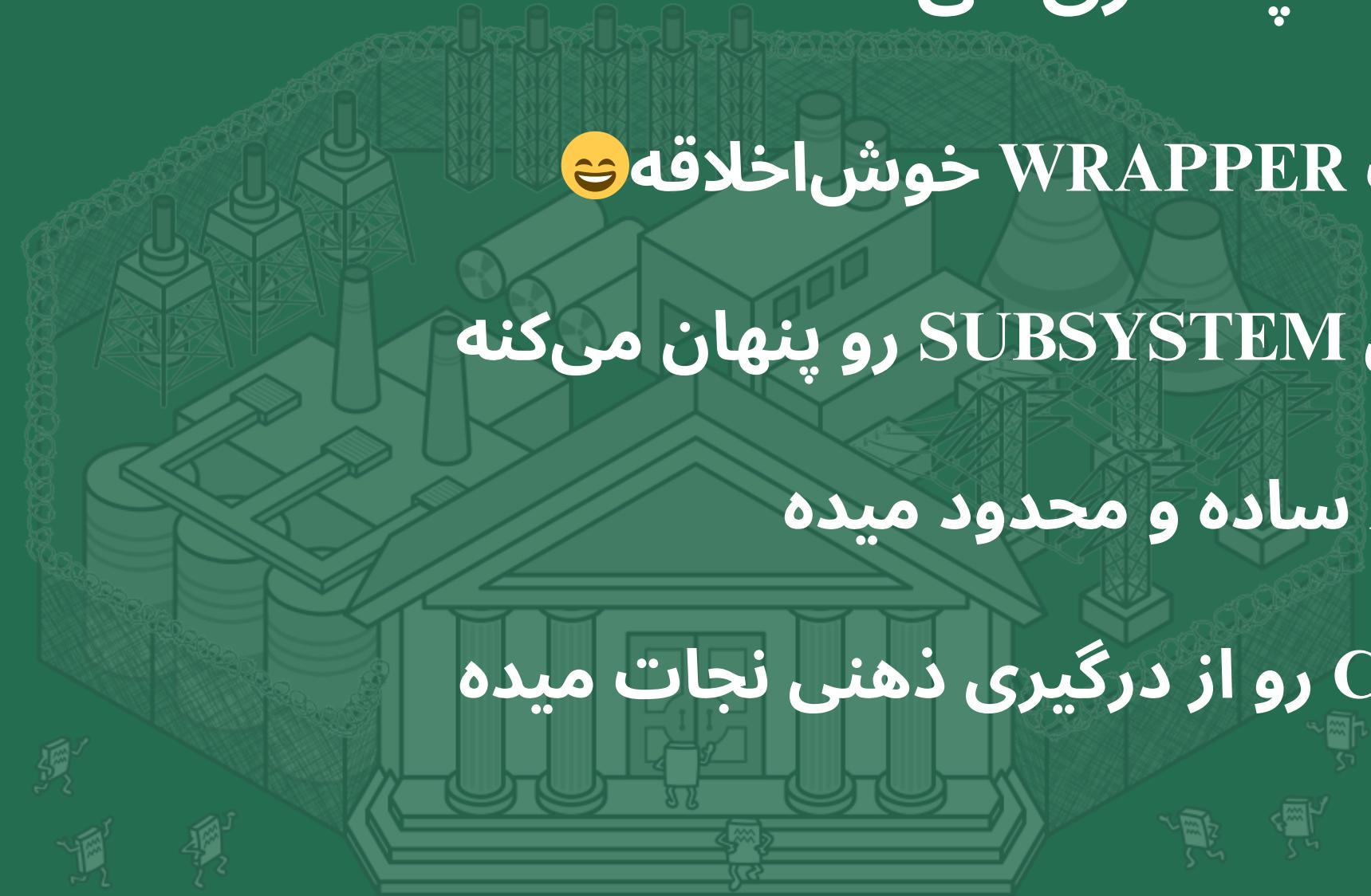
دقیقاً چه کاری می‌کنه؟ FACADE

یک WRAPPER خوش‌اخلاقه 😊

✓ پیچیدگی SUBSYSTEM را پنهان می‌کنه

✓ یک API ساده و محدود می‌دهد

✓ رو از درگیری ذهنی نجات می‌دهد



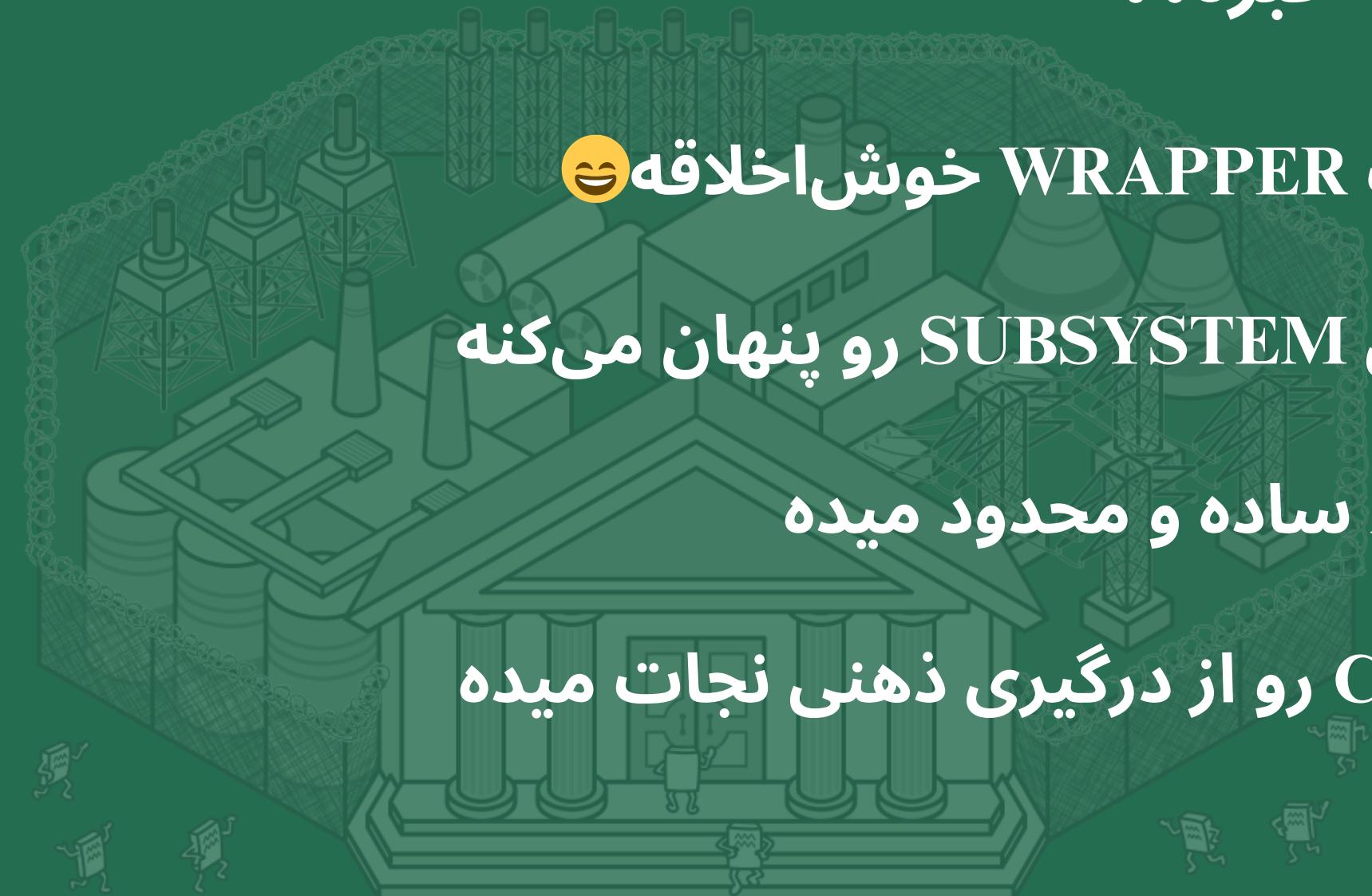
پشت صحنه چه خبره؟؟

یک WRAPPER خوش اخلاقه 😊 FACADE

✓ پیچیدگی SUBSYSTEM را پنهان می‌کنه

✓ یک API ساده و محدود می‌دهد

✓ رو از درگیری ذهنی نجات می‌دهد CLIENT



FACADE دقیقاً چه کاری می‌کنه؟

این الگو :

- کلاس‌های لازم رو می‌شناسه
- ترتیب درست فرآخوانی رو مدیریت می‌کنه
- وابستگی‌ها رو خودش هندل می‌کنه

فقط استفاده می‌کنه ✨ CLIENT 



چرا FACADE همه‌چیز رو نمی‌ده؟

عمدتاً همه‌ی قابلیت‌ها رو نمایش نمیده

چرا؟

چون هدفتش سادگیه، نه کامل بودن

در صورت نیاز خاص:

می‌توانی مستقیم بری سراغ SUBSYSTEM



مثال واقعی: سفارش تلفنی

برای خرید تلفنی:

- با انبار تماس نمی‌گیری
- با حسابداری حرف نمی‌زنی
- با پیک هماهنگ نمی‌کنی

☞ فقط با اپراتور صحبت می‌کنی



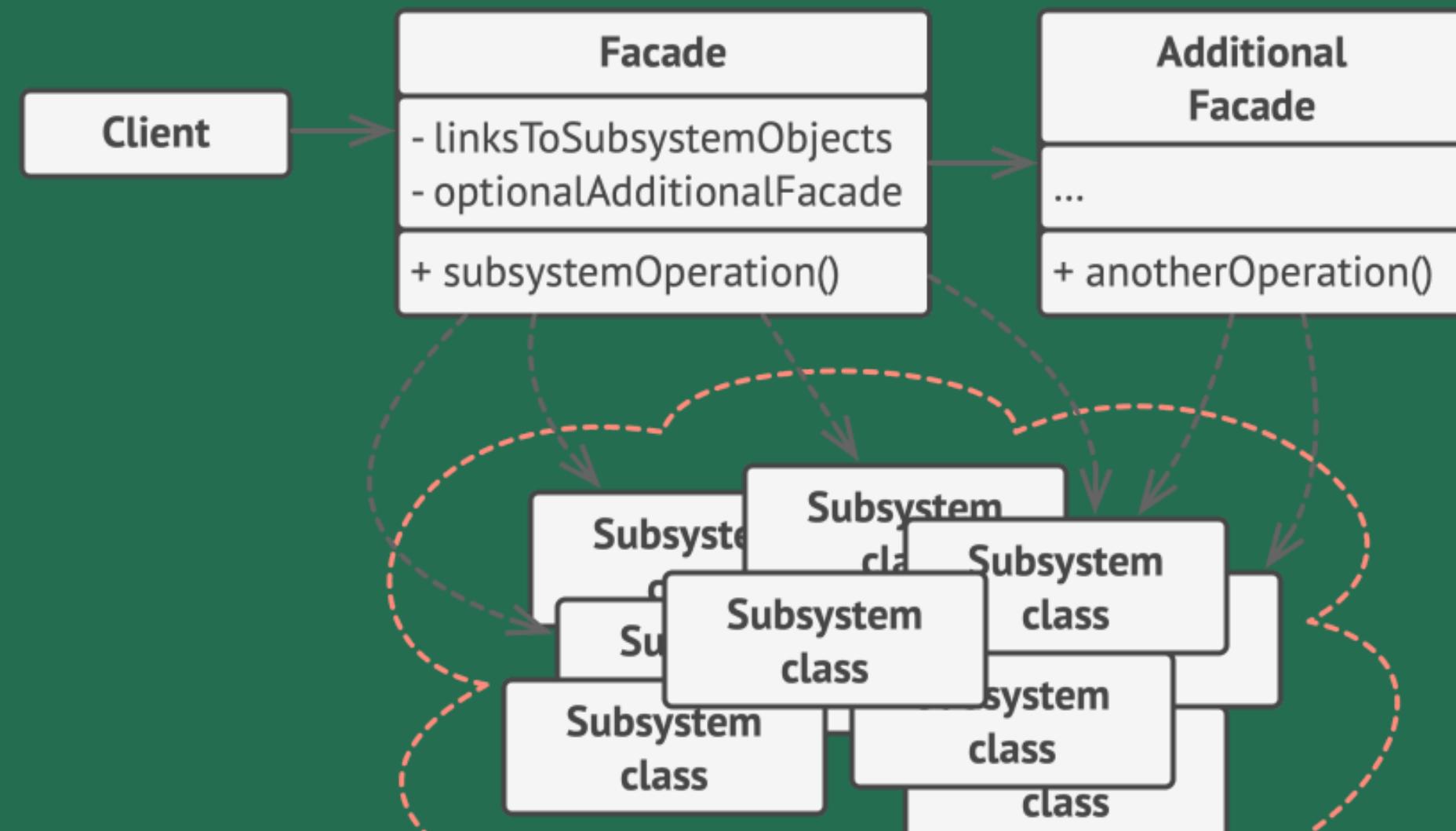
اپراتور = FACADE

اپراتور:

سفارش رو می‌گیره
موجودی رو چک می‌کنه
پرداخت رو انجام می‌ده
ارسال رو هماهنگ می‌کنه
تأیید نهایی می‌ده
✿ برای تو: یک مرحله ساده



ساختار FACADE



در دنیای نرم افزار به چه معناست

مشتری : CLIENT

اپراتور : FACADE PATTERN

بخش ها : SUBSYSTEMS CLASSES



مثال نرم افزاری ساده



مثال نرم افزاری ساده



مثال معروف: VIDEO CONVERTER



ها: SUBSYSTEM

VIDEO LAYER

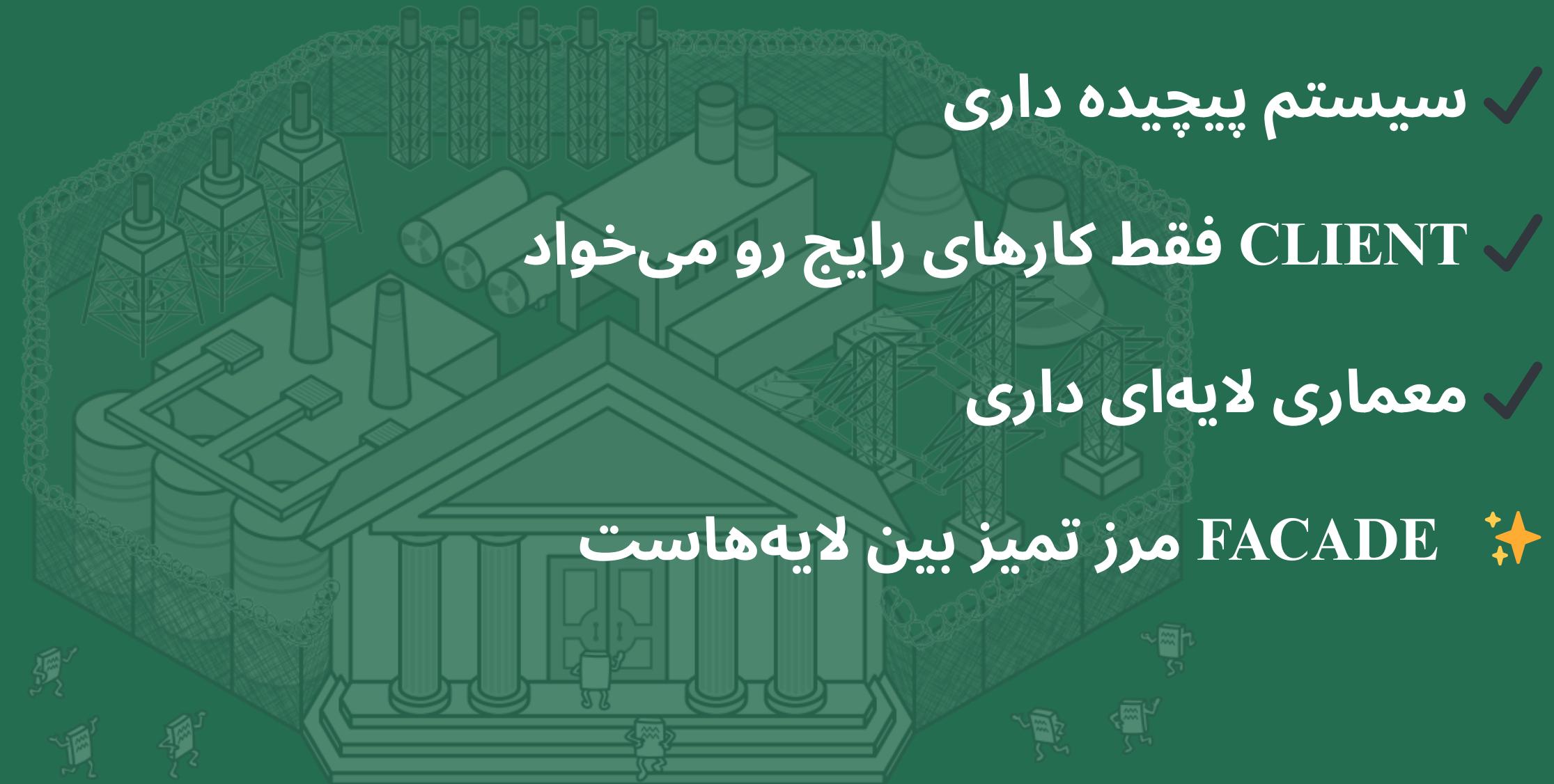
AUDIO LAYER

ها: FACADE

VIDEOFACADE •

AUDIOFACADE •

چه زمانی FASCADE انتخاب خوبیه؟



FACADE + SINGLETON *

اغلب Singleton می شه چون:

- یک نمونه کافیه
- دسترسی مت مرکز من ده
- رفتار سیستم یکنواخته



ADAPTER یا FACADE



Facade

Adapter

ساده‌سازی کل سیستم

تبديل Interface

چند کلاس

معمولًاً یک کلاس

توصیه دوستانه

یعنی: FACADE

کد تمیزتر 

ذهن آروم تر  

خوشحال تر CLIENT 

اگه سادگی مهمه، رفیق خوییه 

