**PROXY KALIBI**

• **Intent**: Provide a surrogate or placeholder for another object to

control access to it.

• **Amaç**: Ona olan erişimi kontrol etmek amacıyla bir başka nesne için

geçiş ya da ara nesne (gerçeği yerine geçen (sahte) nesne) sağla.

• Proxy, dilimize “vekil” olarak çevrilebilir.

**PROBLEM**

• Farklı sebeplerden dolayı bir nesneye olan erişimi kontrol etmek

gerekebilir.

• Bu sebepler, güvenlik, maliyet, teknik kısıtlar vs. olabilir.

• **Güvenlik**: Nesnenin kendisi de nesneye ulaşmak da tehlikeli olabilir.

• **Maliyet**: Nesneyi oluşturmayı ihtiyaç anına kadar geciktirmek

• **Teknik kısıtlar**: Nesne erişim dışında olabilir.

• Nesne**, uzak nesnedir** (**remote** **object**), dışarıdan yalıtılmıştır, nesne threadsafe

değildir, vs.

• **Katma değerli hizmetler**: Nesneye erişim öncesinde ve sonrasında

farklı işler yapmak gerekebilir.

• Erişim isteklerini saymak, düzenlemek ve bir kuyruğa sokmak vs.

• Nesnenin işi uzun sürmektedir ve bu sebeple **eşzamanlı** (**synchronous**) yerine **eşzamanlı** **olmayan** (**asynchronous**) erişim söz konusudur, vs.

• Nesneye erişimin kontrol sebebi olan durumları, nesne üzerinde gidermeye çalışmak nesnenin birlikteliğini düşürür.

• Bu kısıtları istemciden de yalıtmak gereklidir.

• İstemci olabildiğince kısıtlardan haberdar olmamalı, sanki asıl nesne ile çalışıyormuş gibi, normal çalışmasına devam etmelidir.

• Zaten belki de istemci önce asıl nesneye doğrudan ulaşıyordur ama

sonrasında yukarıda sayılan sebepler devreye girmiştir ve istemcide bir

değişiklik yapmadan önlem almak gerekmektedir.



**SONUCLAR**

• Hangi sebeple olursa olsun, ne zaman bir nesne, alt sistem ya da sisteme erişimi kontrol, onu saklama ihtiyacı varsa **Proxy** kullanılabilir.

• İstemci, saklanan nesne ya da sistem varmış gibi çalışmaya devam eder.

• Bu sayede istemci, vekilin sakladığı yerdeki değişimlerden de yalıtılır

• Erişim kontrol ihtiyacının mahiyetine göre vekil nesne farklı işlevler görür:

• **Remote** proxy (Uzak vekil): Dağıtık nesneler (distributed object) ve web servisleri,

• **Virtual** proxy (Sahte vekil): Geç yüklemeleri sistemlerde, ön bellek (cache) gerçekleştirmelerinde,

• **Protection** proxy (Koruyan vekil): Güvenlik (security), erişim kısıtı vb. durumlarda.

• Vekil, ne tür bir vekalet yaptığına göre istekleri gerçek nesneye iletir.

**KULLANIMLAR**

• JPA, Hibernate ya da EntityFramework gibi ORM frameworklerinde, özellikle geç yükleme (lazy loading) yapısını gerçekleştirmek amacıyla.

• Sonradan yüklemenin uygulandığı tüm torba (collection) yapılarında ve alanları sonradan yüklenecek entityler için vekil torbalar ve vekil entity nesneleri oluşturulur.

• Aspect-Oriented Programming’de (AOP), metotları aspectler ile örmek (weaving) için.

**Diger Kalıplarla İliski**

• **Proxy** ile **Adapter** arasındaki fark, ilkinin sakladığı nesne ile aynı arayüzde olmasına karşın Adapter farklı arayüzdedir.

• **Proxy** ile **Façade** arasındaki fark, ilkinin sakladığı nesne ile aynı arayüzde olmasına karşın Façade farklı arayüzleri basitleştirerek bir yerde toplamaktadır.

**UYGULAMA**

Aşağıdaki teknolojilerde Proxy kalıbının kullanımını araştırın:

• SOAP Web servisleri

• Corba

• RMI ve EJB

• .Net Remoting

2. Kullanıcı sayısı üzerinden sınırlı lisansa sahip bir yazılımı kullananların

sayısını bir kuyruk ile takip edip, yeni istekleri değerlendiren bir lisans

proxysi gerçekleştirin.

• (License proxy that keeps track of the number of users in a queue so

that users up to the limit can use the software and others wait for next

available license.)