

Façade Örüntüsü

Law of Demeter

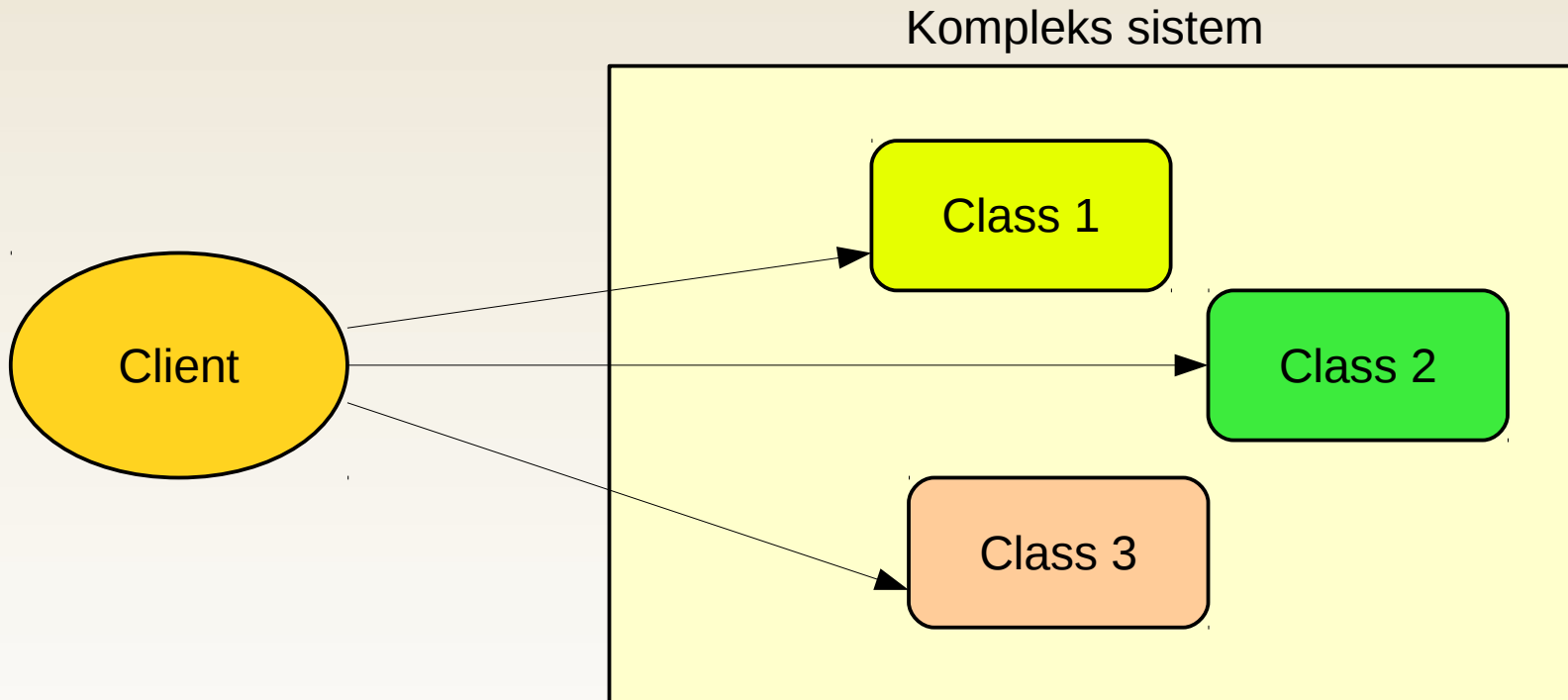
Her zaman için sadece kapı komşun ile konuş...

**Nesneler ve modüller arası iletişimi
mümkün olduğunca basitleştir...**

Alırım Façanı Aşağı!

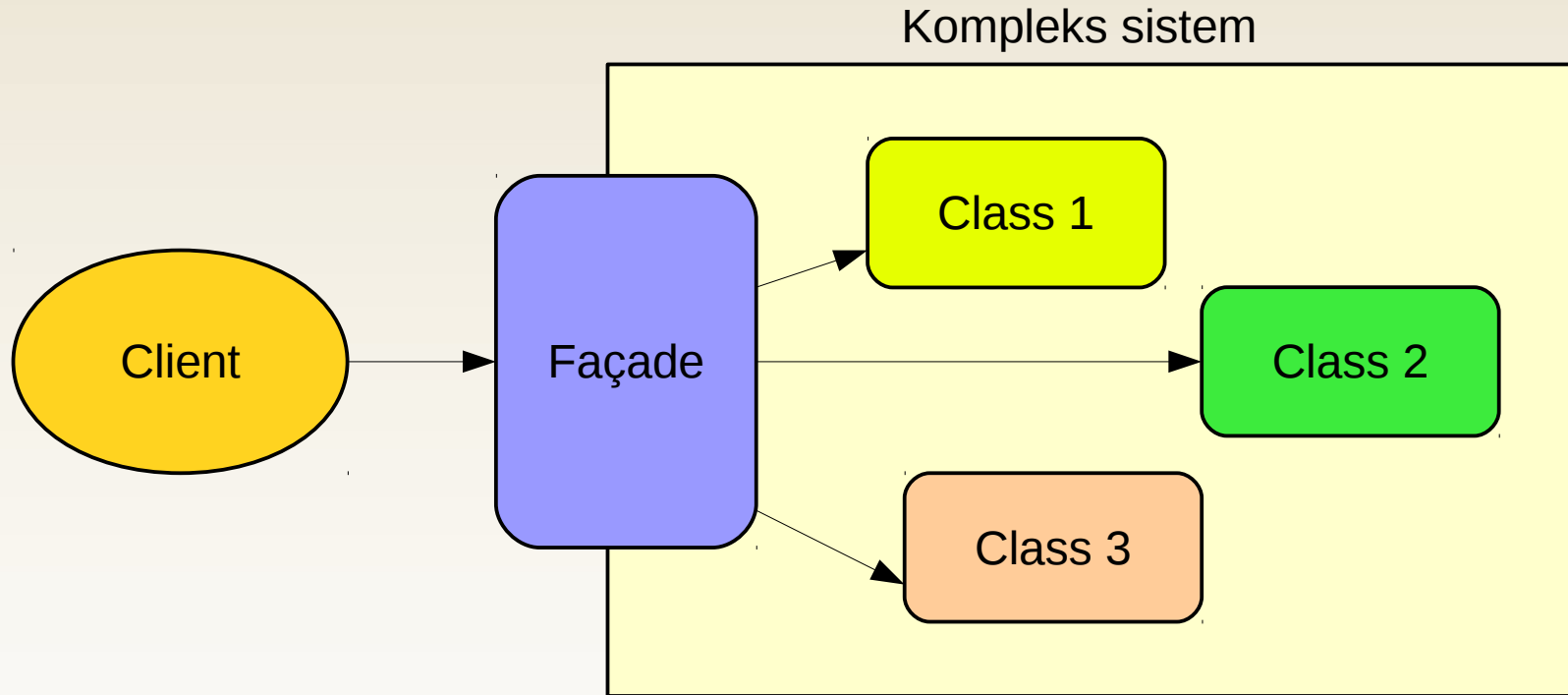


- Faça aslında İtalyanca **faccia** kelimesinden geliyor
- **Ön yüz, çehre** anlamındadır
- Türkçe'ye girişi de Beyoğlu civarında yaşayan İtalyan kökenli levantenlerin vasıtasıyla oluyor
- Neredeyse 400-500 yıllık geçmişe sahip bir kelime dilimizde.
- Deyim TDK'ya göre **birisini mahcup etmek, bozmak** anlamına geliyor



Bazı sistemlerde istemci uygulamanın, modülün veya sınıfın hedef kod üzerinde bir senaryoyu işletebilmesi için çoğunlukla birden fazla metot çağırısı yapması, bu metot çağrılarını da belirli bir sırada gerçekleştirmesi, metotların input parametrelerini uygun biçimde hazırlaması, dönen değerleri işleyerek müteakip metotların input parametrelerine dönüştürmesi konusudur.

Façade



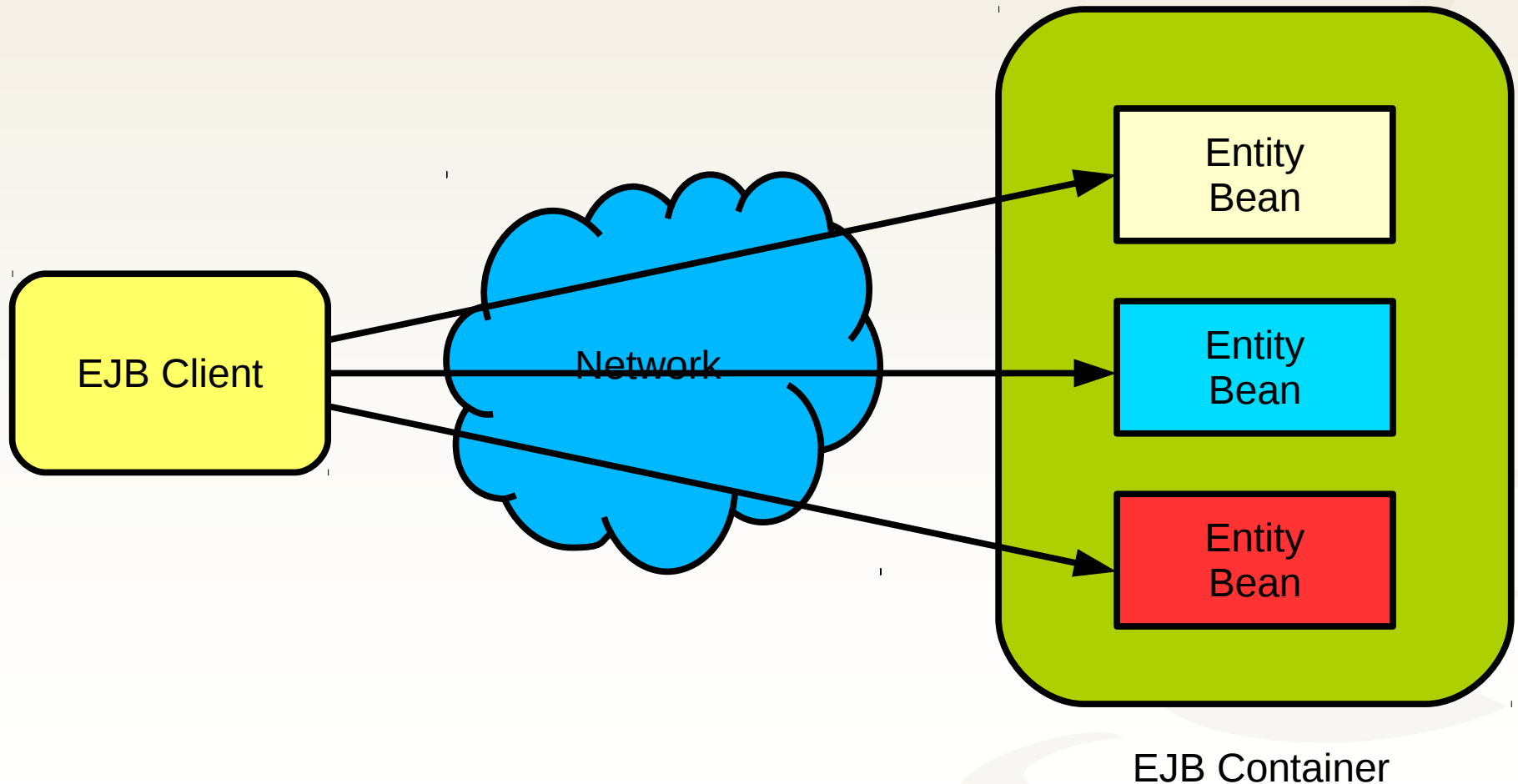
Façade, mevcut sistemi kullanan **yeni ve basit bir arayüz** sunar

İstemci işini **bu arayüz üzerinden** görür

Asıl **işin neredeyse tamamı** yine arkadaki sistem tarafından yapılmaktadır

EJB ve Session Façade

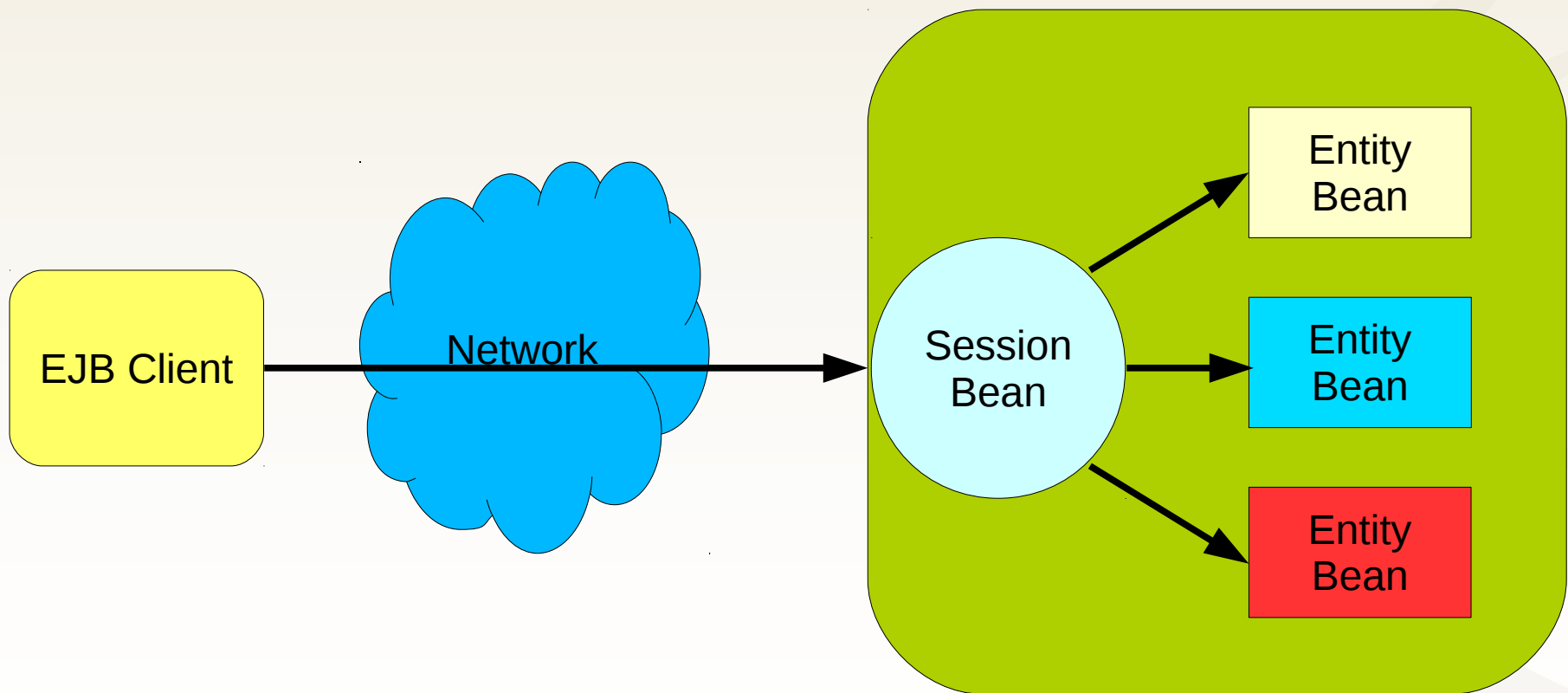
Client'in bir senaryoyu hayata geçirmek için network üzerinden teker teker entity Bean instance'larına erişmesi performans açısından oldukça kötü sonuçlar doğuracaktır



EJB ve Session Façade

Client senaryoyu işletmek için session bean'a bir metot çağırısı yapar

Session Bean arka tarafta farklı entity bean instance'lara erişerek istenen fonksiyonalliteyi Yerine getirir



EJB Container

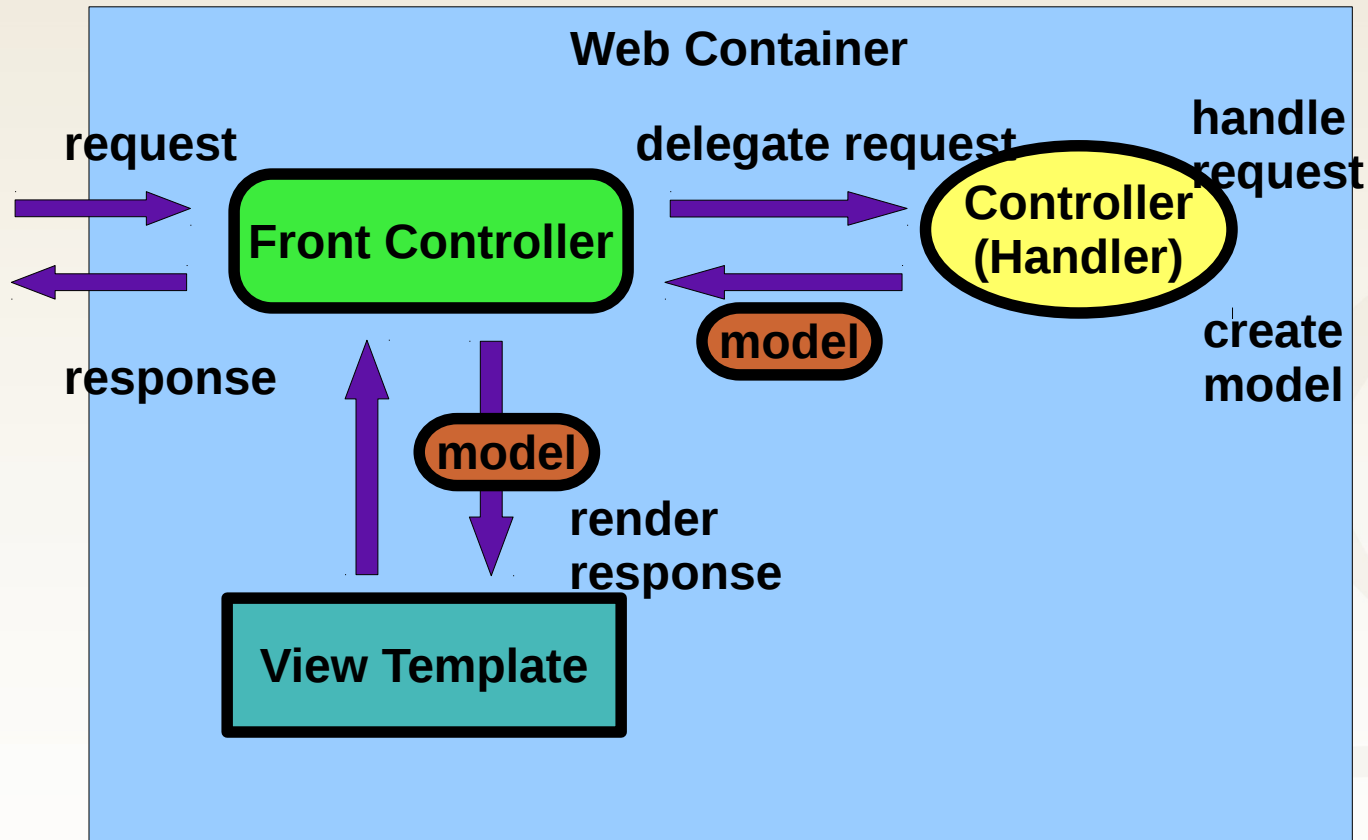
Web Uygulamaları ve Front Controller



Web uygulamasına gelen bütün requestler Front Controller tarafından ele alınır

Bu sayede uygulamanın ihtiyaç duyduğu güvenlik, internationalization, view render gibi işlemler tek ve merkezi bir noktadan halledilmiş olur

Web Uygulamaları ve Front Controller

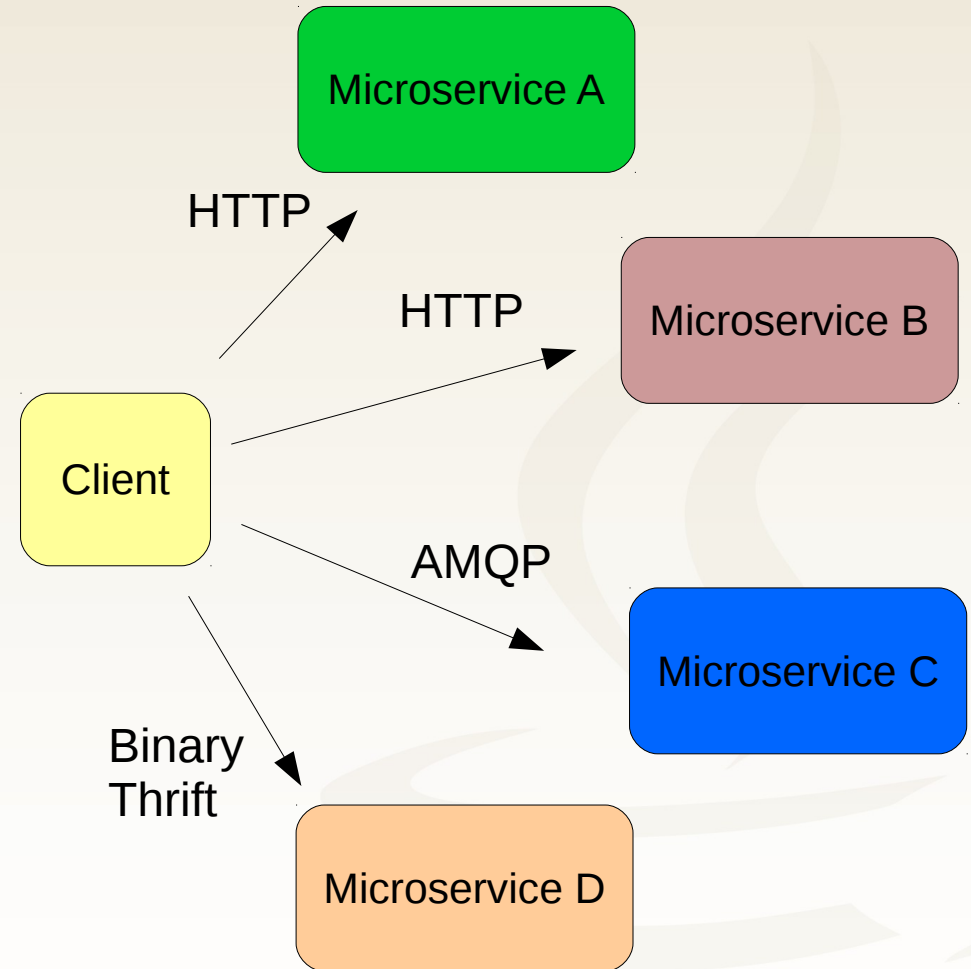


Front Controller genellikle bir Servlet olur, uygulamaya gelen bütün request'leri handle eder

Request'in hangi komuta karşılık geldiğini çözümlemesi için request'in path'inden, parametrelerinden vs yararlanır

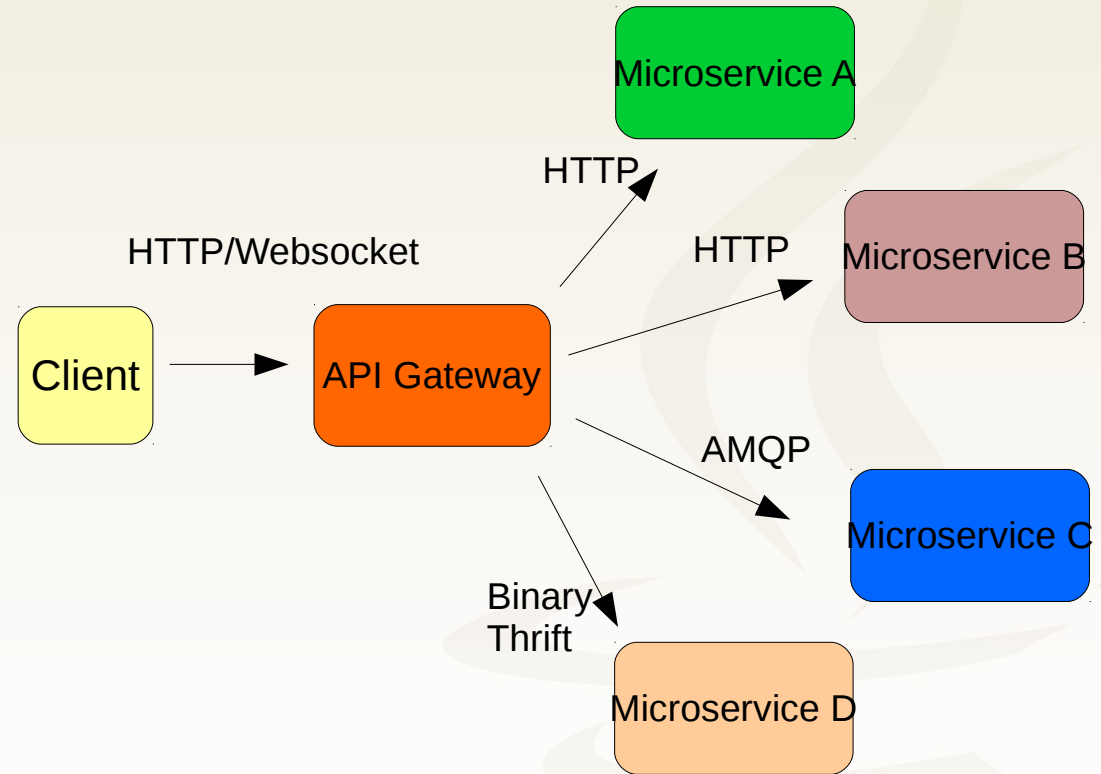
Microservice ve API Gateway

- İstemci uygulamaların **doğrudan her bir mikroservis bileşeni** ile etkileşimi bazı problemler doğurabilir
 - Mikroservis'in sunduğu API ile istemcinin **ihtiyaçlarının örtüşmemesi**
 - Bir takım mikro servislerin HTTP dışında web üzerinden iletişime uygun olmayan **binary protokoller kullanmaları**
 - Mikroservisleri istemciden **bağımsız refactor** etme imkanının ortadan kalkması



Microservice ve API Gateway

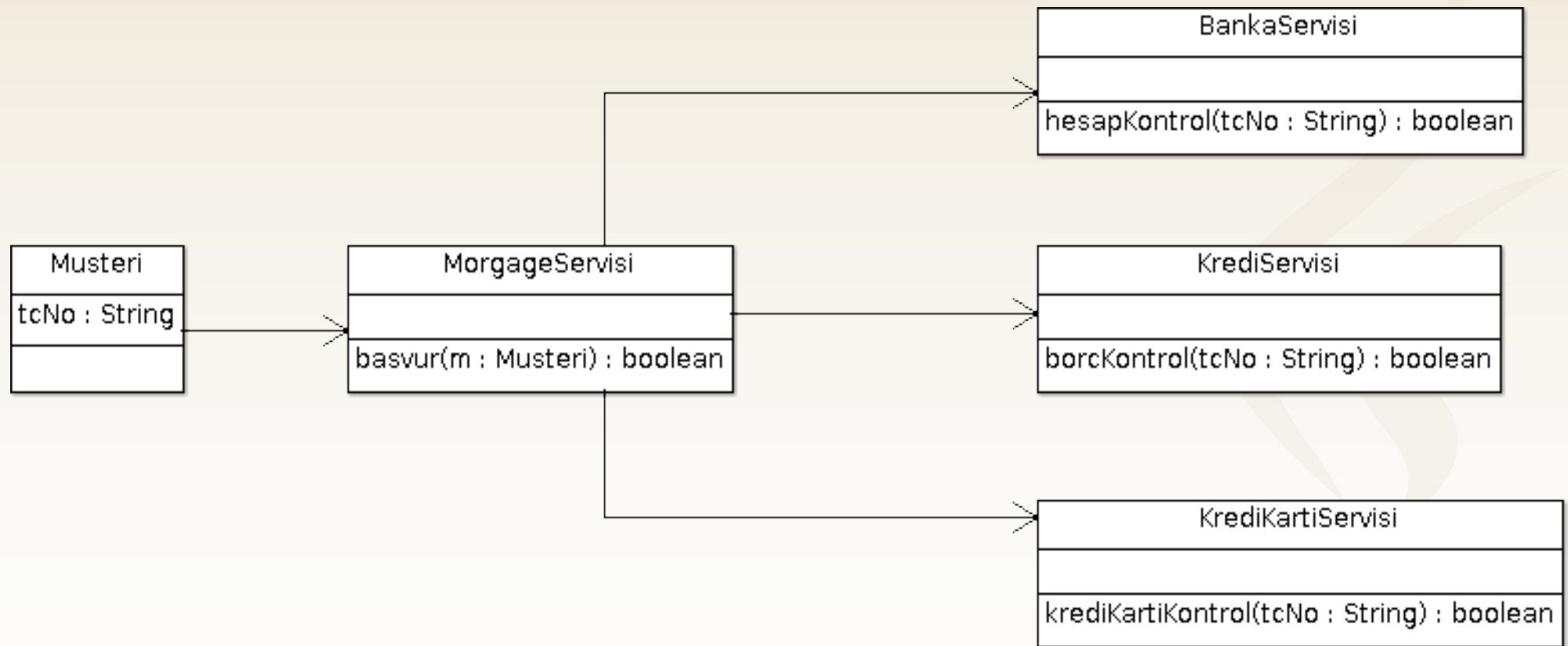
- İstemcilerin doğrudan mikroservis bileşenleri ile konuşması yerine **iletişim API Gateway üzerinden** sağlanır
- Routing, service composition, protokol dönüşümü, güvenlik** gibi konular API Gateway tarafından ele alınır
- Farklı istemcilere özel **farklı türde client API'ler** sunulabilir



LAB ÇALIŞMASI: Façade

- Online morgage başvurusu yapmayı sağlayan bir uygulama geliştirilmesi istenmektedir
- Morgage başvurusu yapan müşterilerin taleplerine verilecek cevap, üç ayrı alt sistemden müşteri ile ilgili bilgilerin değerlendirilmesine göre olumlu veya olumsuz olarak dönmektedir
- Bu alt sistemlerden biri müşterinin banka hesabı olup olmadığına, banka hesabı varsa bakiyesine bakar
- Diğer kredi kartlarının ödeme durumlarını, borçlarını kontrol eder
- Bir diğeri ise müşterinin morgage dışında başka bir kredi daha kullanıp kullanmadığına bakar
- Karar bu üç sisteme sırası ile erişip, uygun metotları çağırarak elde edilecek sonuca göre verilir

LAB ÇALIŞMASI: Façade



Façade Örüntüsünün Sonuçları

- Sistemin kullanımı **daha basit ve kolay** olur
- İstemcinin kullanması ve bağımlı olması gereken **nesnelerin sayısını azaltır**
- Modüller arasındaki **bağımlılık ve etkileşim daha basit ve yönetilebilir** hale gelir

İletişim



www.harezmi.com.tr

www.java-egitimleri.com



info@harezmi.com.tr

info@java-egitimleri.com



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)