

# JAVA RMI ve Hibernate teknolojileri kullanılarak çok amaçlı bir yazılım altyapısı hazırlanması

---

Hakan ALBAĞ  
Tahsin Barış AKAN

*Bitirme Projesi*

05.06.2006

# Giriş

---

- ▶ Ticari yazılımlarda ortak ihtiyaçlar
  - ▶ Birden fazla kullanıcı
  - ▶ Yoğun veritabanı işlemleri
  - ▶ Genişletilebilirlik
  - ▶ Kullanıcı dostu
- ▶ Geliştirme süreçleri ve maliyetler

# Amaç

---

- ▶ Ortak ihtiyaçlara sahip yazılım sistemlerinin geliştirilmesine hizmet edecek bir altyapı
  - ▶ Geliştirme sürecini hızlandırmak
    - ▶ Birden fazla kullanıcı tarafından uzaktan erişim
    - ▶ Veritabanı erişimleri
    - ▶ Kullanıcı / grup / yetki yönetimi
    - ▶ Standard arayüzler
    - ▶ Dinamik modüler yapı

# e-cza

---

- ▶ Ecza deposu stok otomasyon yazılımı



- ▶ Stok takibi
- ▶ İlaç – Firma tanımlama / alım / satım işlemleri
- ▶ Geçmiş satışlara yönelik stok analizi

# Java RMI

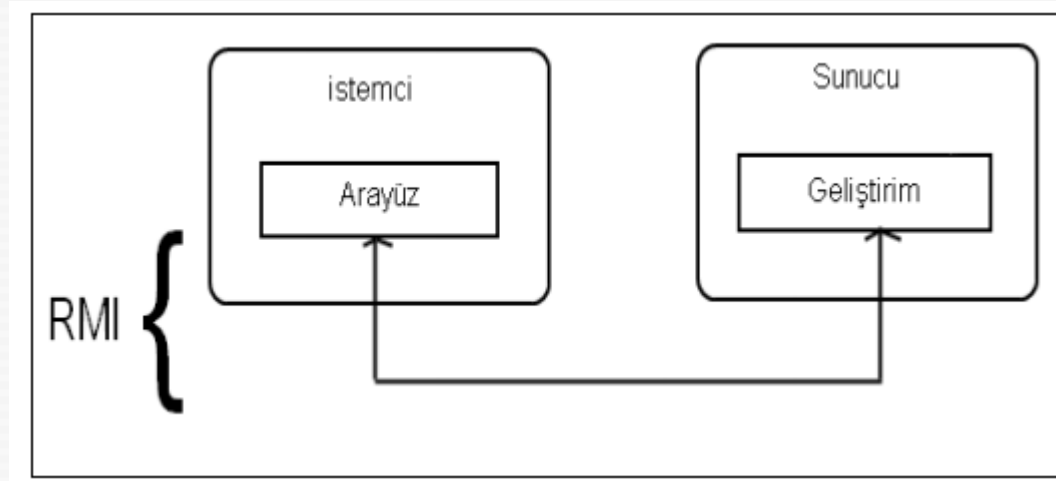
---

- ▶ Dağınık uygulamalarda uzaktaki nesnelere yerel ortamdaki esneklikte çalışmayı sağlar.
- ▶ Dağınık bir uygulamanın gerçekleştirdikleri:
  - ▶ Uzak nesnelere belirlemek
  - ▶ Uzak nesnelere iletişim kurmak
  - ▶ Nesnelere derlenmiş kodunun (bytecode) yüklenmesi

# Java RMI (2)

---

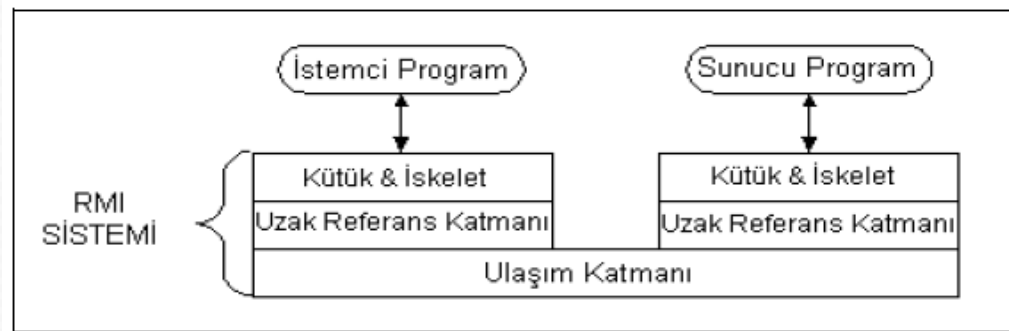
- ▶ RMI basit bir kavram üzerinde yükselir:
  - ▶ "Davranışın tanımı ve davranışın geliştirilmesi ayrı kavramlardır."



# Java RMI (3)

---

## ► RMI Katmanları



### ► Kütük ve iskelet :

İstemci çağrılarını keserek RMI servisine yönlendirir.

### ► Uzak referans katmanı:

Uzak nesnelere yapılan referans istemlerini yönlendirir.

### ► Ulaşım Katmanı:

Temel bağlantı özelliklerini sunar.

# Hibernate

---

- ▶ Java için ORM / Object-Relational Mapping kitaplığı
  - ▶ Veritabanı işlemlerini nesneye dayalı yürütmek
  - ▶ Temel işlemler için programcı yükünü azaltmak
  - ▶ Cep bellek mekanizması

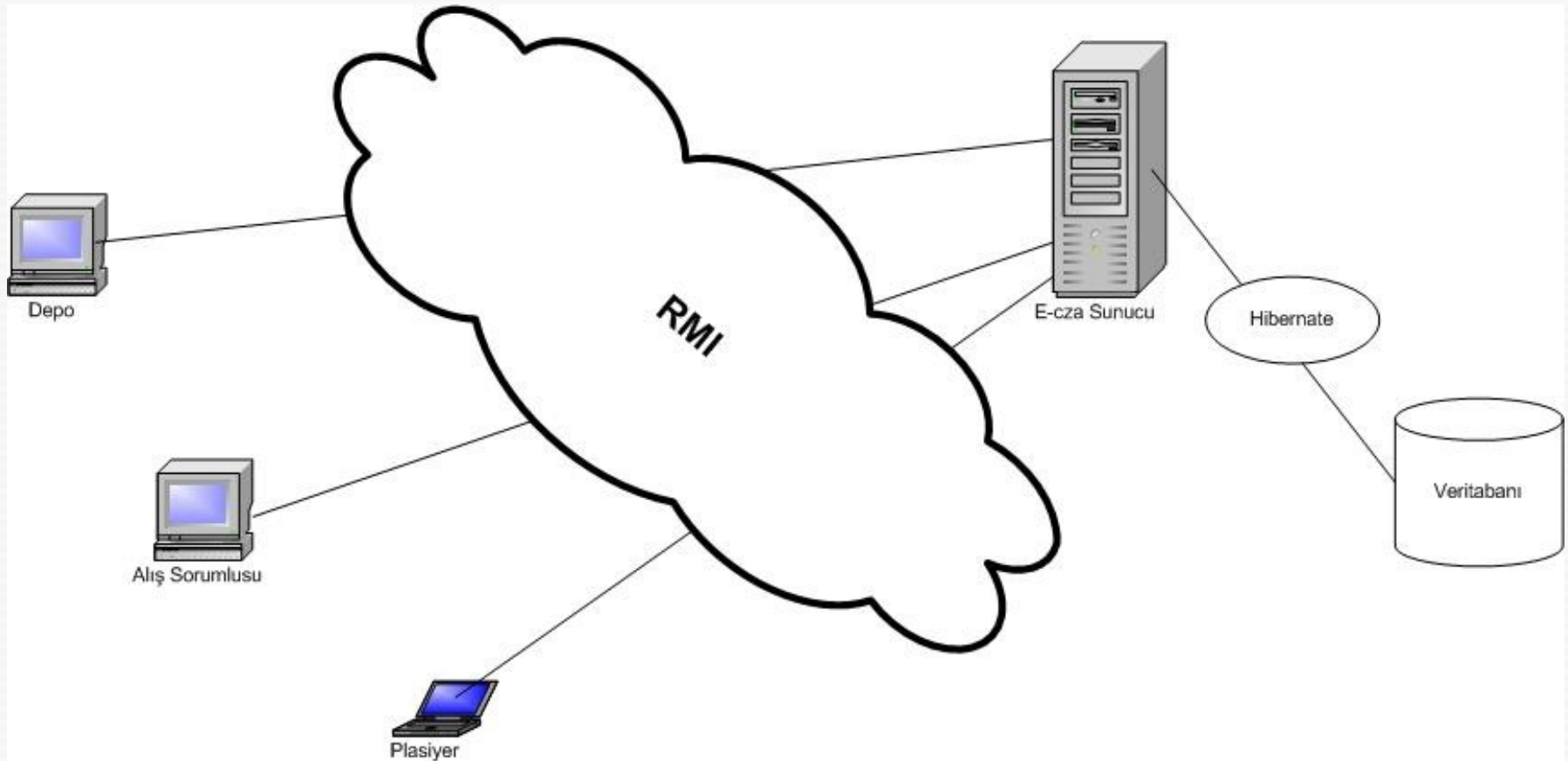


# Hibernate (2)

---

- ▶ Varlık nesneleri (POJO)
- ▶ XML eşleme dosyaları (.hbm.xml)
  - ▶ Java tipi – SQL tipi
  - ▶ Java sınıfı – SQL tablosu
  - ▶ 1-1,1-n ,m-n ilişkiler
- ▶ Konfigurasyon
  - ▶ SQL diyalekti
    - ▶ JDBC ( mySQL,MS SQL Server,Oracle,vs.)

# Sistem - Genel yapı



# İstemci & Sunucu

---

- ▶ İstemci

- ▶ Arayüz

- ▶ Yerel veri işleme

- ▶ RMI çağruları

- ▶ Sunucu

- ▶ Uzak nesnelerin yönetimi

- ▶ Tüm veritabanı işlemleri

# Varlık nesneleri

---

- ▶ POJO ( Plain old java objects)
  - ▶ İlac, Firma, Kampanya gibi temel nesnelere
  - ▶ İstemci de Sunucu da ihtiyaç duyar
  
- ▶ Veri modelinden nesne modeline geçiş
  - ▶ Eclipse Hibernate Tools

# GenericDAO

---

- ▶ DAO – Data Access Object
  - ▶ Şablon sınıf
  - ▶ Tüm varlık nesnelere için veri erişimi
  - ▶ Standard veritabanı işlemleri
  - ▶ SQL

```
T getByID(Integer ID)
List<T> findAll()
List<T> findByCriteria(Criterion ... criteria)
List findByQuery(String sql)
```

# GenericDAO (2)

---

- ▶ İstemciden veritabanı erişimi
  - ▶ Doğrudan veritabanı bağlantısı
  - ▶ Sorunlar
    - ▶ Hibernate cep bellek
    - ▶ Düşük performans
    - ▶ Mimari
- ▶ Çözüm : RemoteGenericDAO

# RemoteGenericDAO

---

- ▶ Veri tabanı erişiminin tek yerden yapılması

GenericDAO → RMI istemcisi

RemoteGenericDAO → Veritabanı erişimi

- ▶ Hibernate cep bellek kullanımında
- ▶ Şablon sınıf ama nesnelere açılıştan yaratılmalı

# Kullanıcı Arayüzü

---

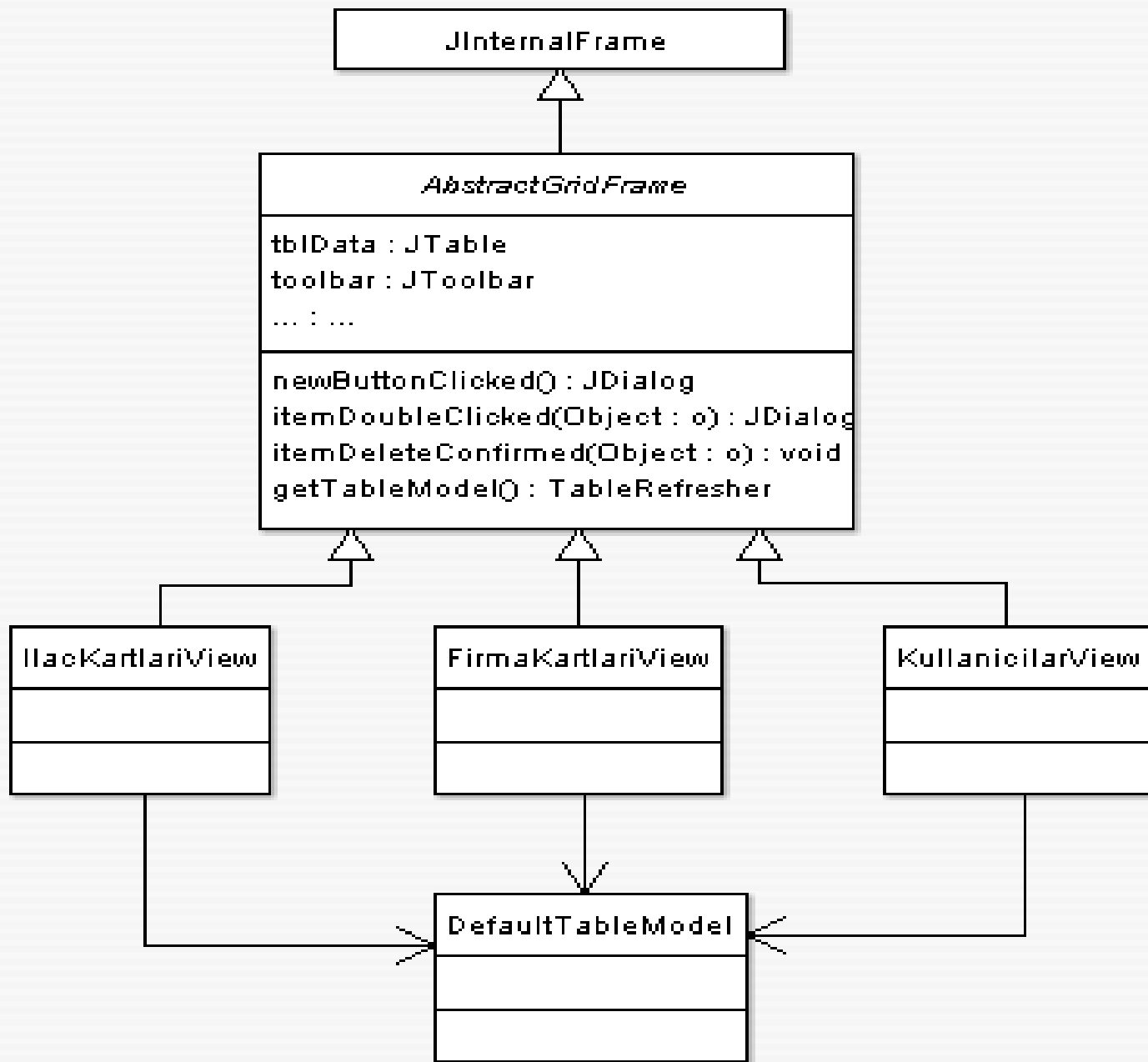
- ▶ Java Swing
- ▶ Arayüz Kalıtımı
- ▶ Dinamik menu elemanları
  - ▶ Genişletilebilirlik
  - ▶ Erişim denetimi



# Arayüz kalıtımı

---

- ▶ Ortak görünüm ve işlemler
  - ▶ Tablo yapısı
  - ▶ Ekleme / Silme / Güncelleme
  
- ▶ AbstractGridFrame sınıfı
  - ▶ Tüm tablo yapılı ekranlar için soyutlama sınıfı



# Dinamik menu ve erişim

---

- ▶ Menu elemanları veritabanında
  - ▶ Menu grupları (Modüller)
    - ▶ Menu elemanlar ve çalıştıracakları java sınıfları

1. Yeni modüller eklenebilir
2. Kullanıcı erişim kontrolü

# Kullanıcı yönetimi

---

- ▶ Çekirdek modüle dahil / Sistem menüsü
- ▶ Kullanıcı ve Grup tanımı
  - ▶ Her kullanıcı bir gruba dahil olmak zorunda
  - ▶ Erişim denetimi gruplar üzerinden yapılır

# Stok Modülü

---

- ▶ Bir ezca deposu stok yapısına göre

- ▶ İlaç , firma , kampanya

- ▶ Stok hareketleri

- ▶ Alış hareketleri

- ▶ Satış Hareketleri

- ▶ Satın alma analiz

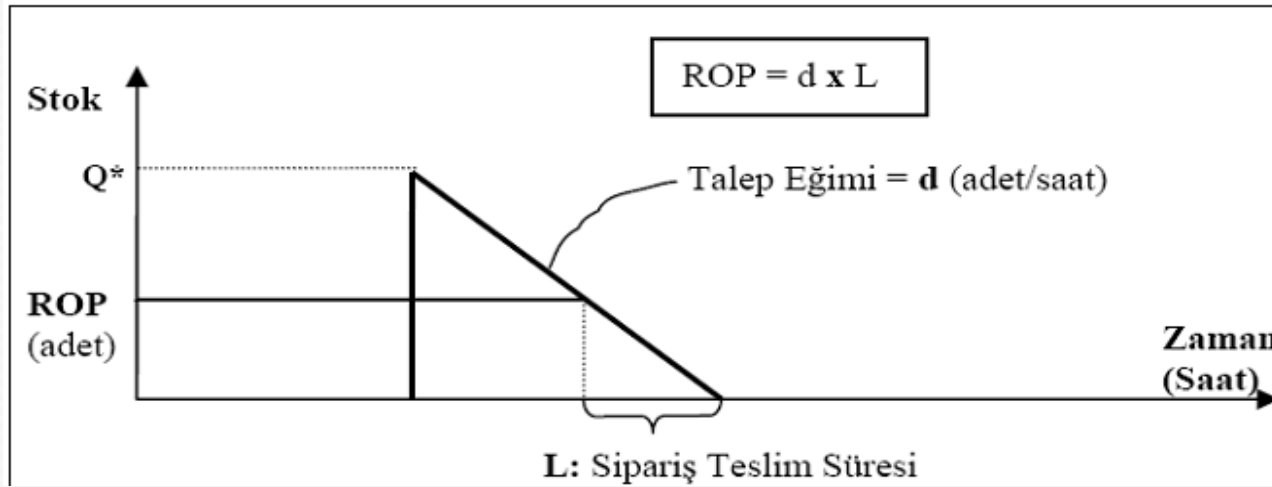
- Belirli tarih aralığında tükenmesi beklenen ilaçlar

- Genel
      - Dönemsel

# Stok Analiz

- ▶ Geçmiş satışlar üzerinden analiz hesaplama

$$\text{ROP(adet)} = (\text{Birim zamandaki talep}) * (\text{temin süresi})$$



# Stok Analiz (2)

---

- ▶ Analiz için gerçekleştirilen adımlar
  - ▶ Normal dağılıma göre veri üretme
  - ▶ İleriye yönelik test verisi
  - ▶ Dönemsel veri üretme

# Sonuçlar & Yorumlar

---

- ▶ Hibernate kullanımı
  - ▶ Uygun şartlar & doğru kullanım
  
- ▶ Web Servis ↔ RMI
  - ▶ Performans
  - ▶ Geliştirme kolaylığı
  - ▶ Birlikte işlerlik (interoperability)



# Teşekkürler

---

- ▶ Demo ve sorular