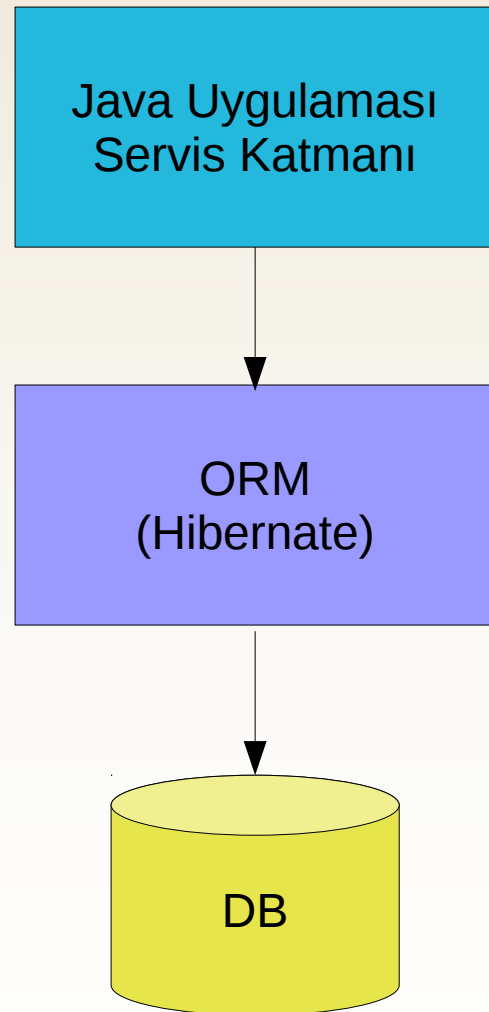


# JPA/Hibernate Persistence Framework'e Giriş

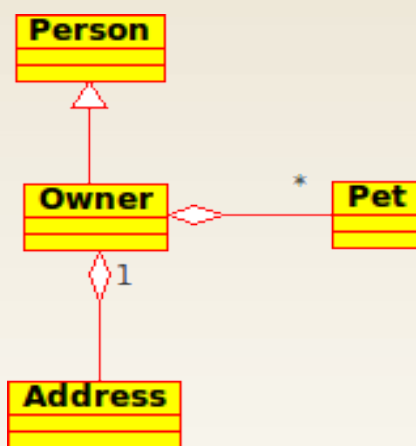


# ORM'ye Giriş



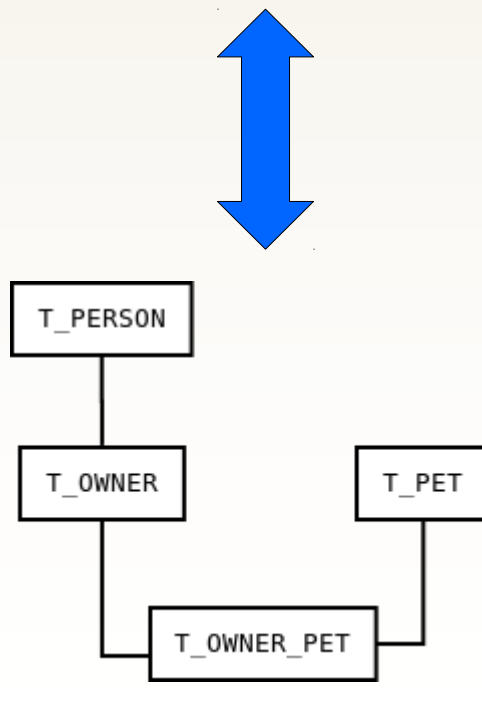
- ORM, **persistent verinin Java tarafında ele alınmasını sağlar**
- Hedef, geliştiricilerin **iş mantığına odaklanmalarını sağlamaktır**
- Hibernate/JPA bir **ORM çözümüdür**
- **Servis katmanı ile veritabanı arasında yer alır**

# ORM'ye Giriş



Domain Model / Nesne Model

- ORM çözümleri **nesne model ile veri modelini** ilişkilendirir
- Mümkün olduğunca **pesistence işlemlerini nesne model üzerinden** yapmayı hedefler



E-R Model / Data Model

# Paradigma Uyumsuzluğu

- Nesne model ve ilişkisel model birbirlerine çok benzer iki yapı olsalar da aslında aralarında **çok temel farklılıklar** vardır
- Bu farklılıklar iki dünyanın birbiri ile eşleştirilmesinde bir takım **problemler ve zorluk noktaları** doğurur

# Paradigma Uyumsuzluğu

- **Granülerite problemi**
  - Her sınıfa sadece tek bir tablo karşılık gelmeyebilir
  - Bire bir eşleşme olacak/olmalı diye bir kural yoktur
- **Yönlü ilişkilerde problem**
  - Nesne model'de ilişkilerin yön bilgisi mevcuttur
  - İlişkisel model'de ise yön bilgisi söz konusu değildir
- **Identity problemi**
  - Veritabanı kimliği ile nesne kimliği ve equality kavramları farklılık arz edebilir

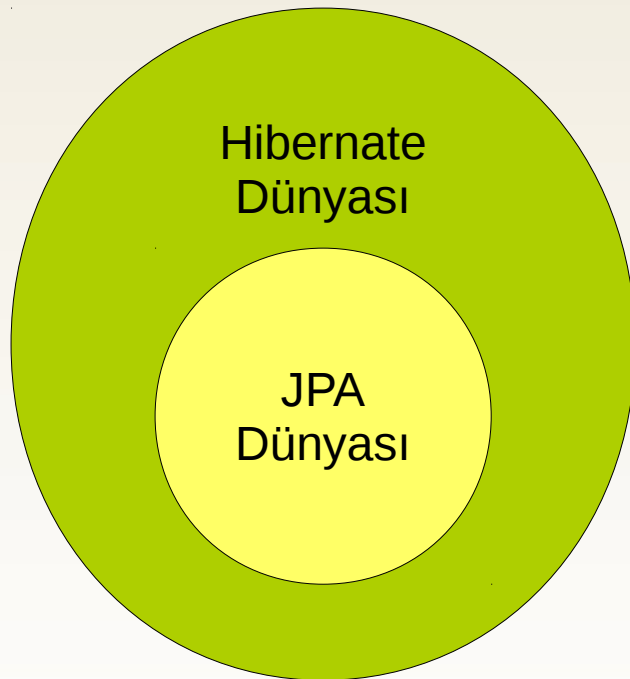
# Paradigma Uyumsuzluğu

- **Nesne ağında dolaşım problemi**
  - Nesne model, nesne ağı üzerinde adım adım dolaşmaya uygundur
  - İlişkisel model de ise ihtiyaç duyulan veri, tek bir seferde tabular olarak elde edilip işlenir
- **Altsınıf problemi ve polymorphic sorgular**
  - Bir inheritance hiyerarşisi veritabanında farklı tablo yapıları ile ele alınabilir
  - Üst sınıflar veya interface'ler üzerinden yapılan nesne sorgulamaları ilişkisel model üzerinde ele alınabilmelidir

# ORM Çözümünü Oluşturan Temel Kısımlar

- Herhangi bir ORM çözümü temel olarak **dört kısımdan** oluşur
  - Metadata tanımlama/mapping kabiliyeti
  - CRUD işlemleri için bir API
  - Nesnelerin değişikliklerini takip eden, lazy ilişkileri yöneten ve diğer optimizasyon fonksiyonlarını ele alan bölüm (Persistence Context)
  - Sorgulama için bir Query/Criteria API

# Hibernate ve JPA İlişkisi



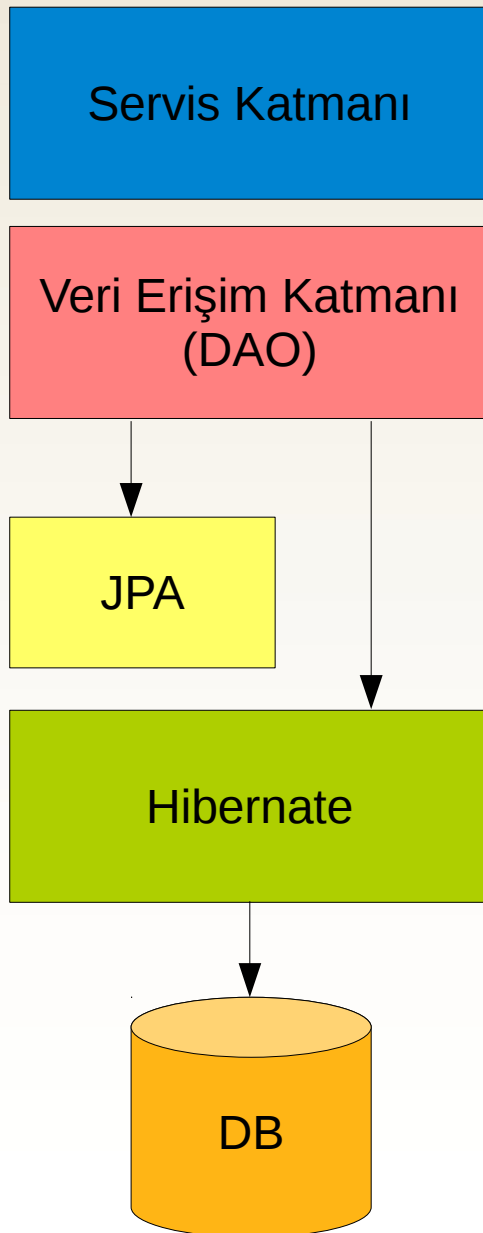
- JPA EJB3'ün **ORM persistence** spesifikasyonudur
- Hibernate bir **JPA implementasyonudur**.
- Diğerleri; EclipseLink, OpenJPA, DataNucleus, ObjectDB
- Hibernate, JPA'nın sunduğu bütün kabiliyetleri sunmaktadır. **İlave özelliklere de sahiptir**



# JPA-Hibernate Versiyon İlişkisi

JPA API	Hibernate ORM
1.0	3.2+
2.0	3.5+
2.1	4.3+
2.2	5.3+

# Katmanlı Mimari ve Hibernate/JPA



- **Veri erişim (DAO) katmanı JPA ve Hibernate ile ilişki halindedir**
- **Hibernate kullanımı JPA API üzerinden yada doğrudan native Hibernate API üzerinden olabilir**

# JPA API Üzerinden Çalışmanın Faydaları

- Metadata'nın **otomatik tespit** edilmesi konfigürasyon sürecini kolaylaştırır
- **Standart** bir konfigürasyon söz konusudur
- Konfigürasyon dosyasının **yeri sabitlenmiştir**
- Persistence işlemleri **standard veri erişim arayüzü** üzerinden gerçekleştirilir

# İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

