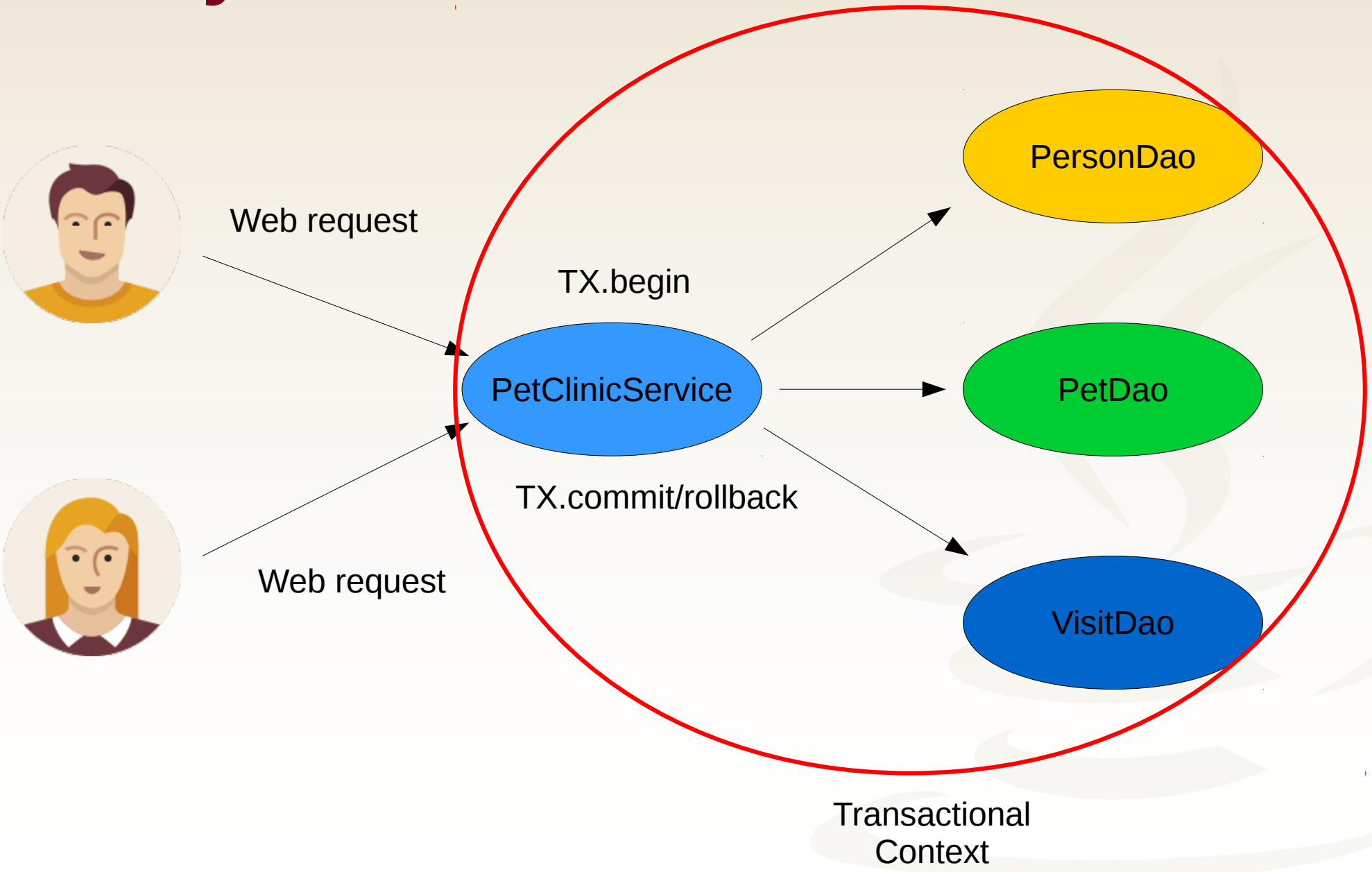
Contextual Session Kabiliyeti

Contextual Session Kabiliyeti

JAVA Eğitimleri

- Transaction yönetimi **servis katmanı**nın görevidir
- TX demarcation servis katmanında **dekleratif** veya **programatik** olarak gerçekleştirilir
- Servis katmanında bir transaction başlatıldıktan sonra veri erişim katmanında **DAO nesneleri ile persistence işlemleri** gerçekleştirilir
- Ardından servis metodunun başarılı sonlanıp sonlanmamasına göre **transaction commit** veya **rollback** edilir

Contextual Session Kabiliyeti



Contextual Session Kabiliyeti

- Hibernate ile çalışırken servis katmanında bir transaction başlatmak için **Session'ı burada açmak** ve daha sonra **persistence işlemlerini aynı session üzerinden gerçekleştirmek** için veri erişim katmanındaki DAO nesnelerine aktarmak gerekmektedir
- Bunun için en uygun yol Java'daki **ThreadLocal** veri yapısını kullanmaktır
- Aynı Session nesnesi **ThreadLocal bir değişken içerisinde** farklı katmanlardaki nesneler arasında rahatlıkla paylaşılabilir

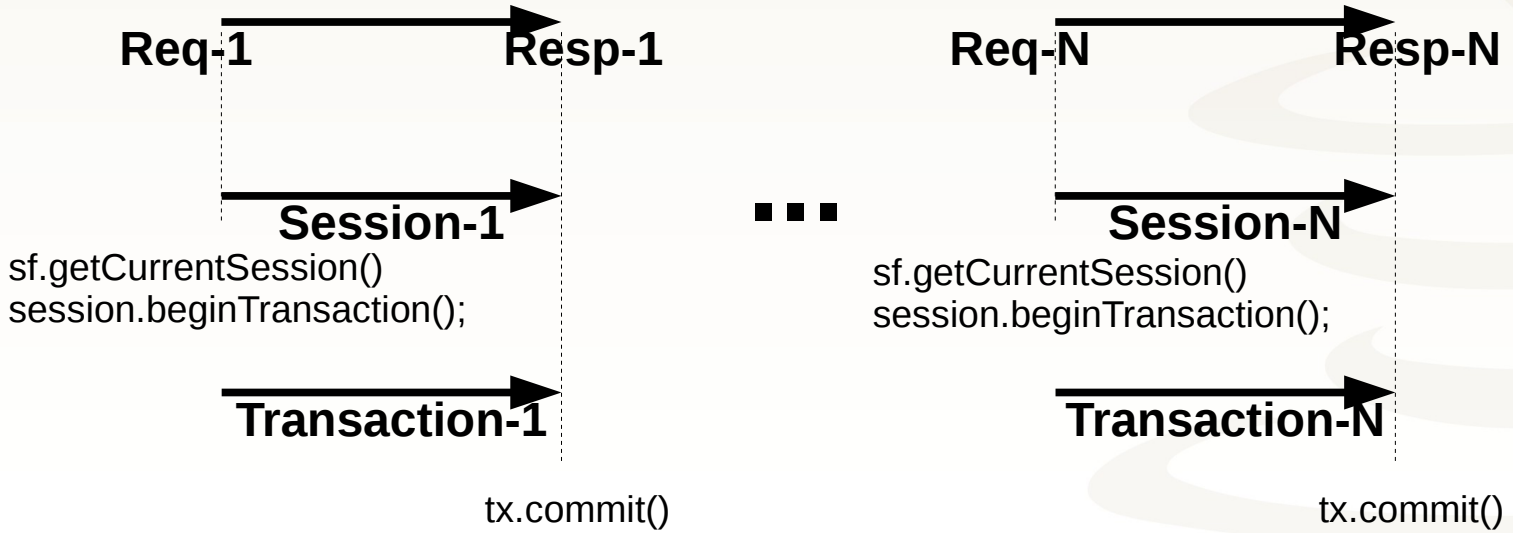
Contextual Session Kabiliyeti

JAVA Eğitimleri

- Hibernate 3 ile birlikte gelen **contextual session** kabiliyeti de bunu sağlamaktadır
- SessionFactory üzerinden bir Session açıldıktan sonra bu **ThreadLocal bir değişkene set** edilir
- Ardından SessionFactory üzerinden istenildiği vakit bu **ThreadLocal değişkende tutulan Session'a erişilebilir**
- Web uygulamalarında **HTTP request'i boyunca aynı thread** kullanıldığı için bu yöntem rahatlıkla kullanılabilir

Contextual Session Kabiliyeti

- Aktive edilmesi için **hibernate.current_session_context_class** konfigürasyon property değeri **thread** olmalıdır
- sf.**getCurrentSession**() thread'e bind edilmiş bir Session varsa bunu döner, yoksa yeni bir Session oluşturulur
- tx.commit()/rollback() Session'ı otomatik olarak kapatır



Contextual Session Kabiliyeti ve Transaction'lar

```
Session s = null;
Transaction tx = null;
try {

    s = sf.getCurrentSession();
    tx = s.beginTransaction();

    //...

    tx.commit();
} catch (RuntimeException ex) {

    tx.rollback();

} finally {
    //s.close();
}
```

getCurrentSession() metodu Thread'a bind edilmiş mevcut bir Session varsa onu döndürür yoksa yeni bir Session Yaratır

Sorgulama dahil bütün işlemler aktif bir TX içerisinde yer almalıdır

TX commit/rollback işlemi aynı zamanda Session'ı da sonlandırır

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com