

```

ManufacturerServlet
- service(httpReq, httpResp) {
    String manufacturerNo = httpReq.getParameter("manufacturerNo");

    Manufacturer manufacturer = manufacturerDao
        .getManufacturer(manufacturerNo);
    httpReq.setAttribute("manufacturer", manufacturer);
    httpReq.getRequestDispatcher("/manufacturer.jsp").forward(
        httpReq, httpResp);
}

```

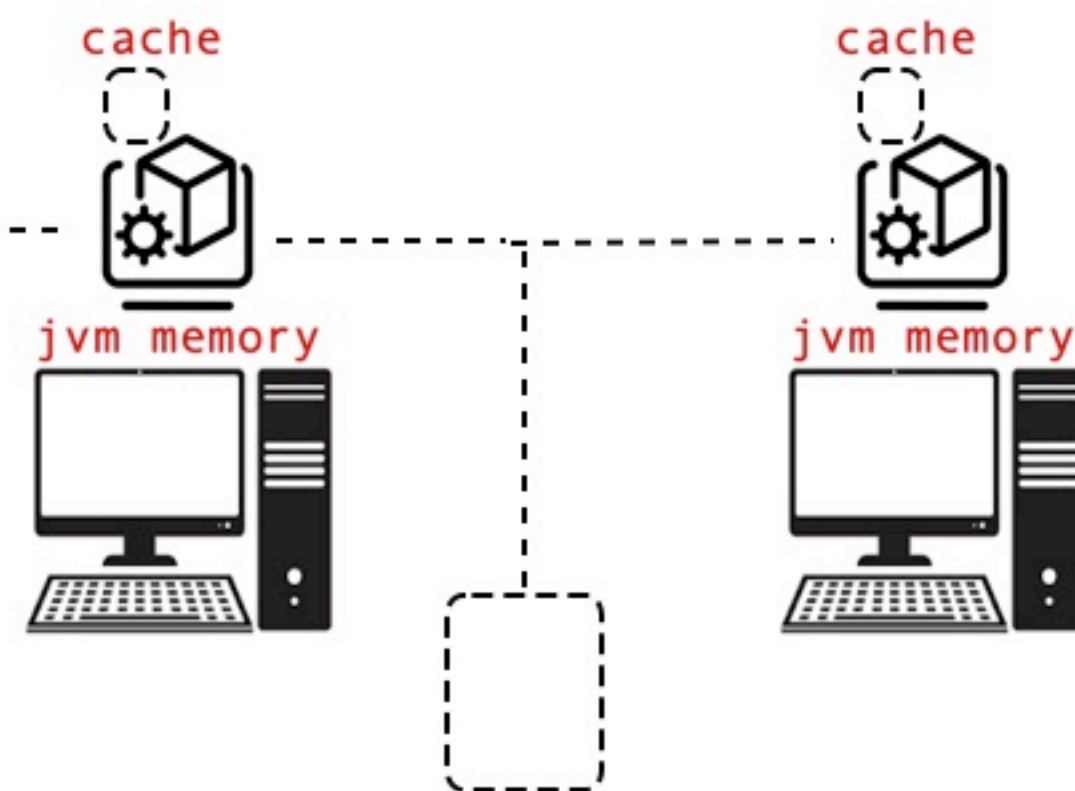
```

ManufacturerDao
- getManufacturer(String manufacturerNo) {
    SessionFactory sf = SFR.getSf();
    Session session = null;
    try {
        session = sf.openSession();
        return session.get(Manufacturer.class, manufacturerNo);
    }finally {
        if(session != null) {
            session.close();
        }
    }
}

```

Here manufacturer data is either  
 1. Long-lived data  
 2. moderate data  
 that stays same for longer amount of time.

distributed caching

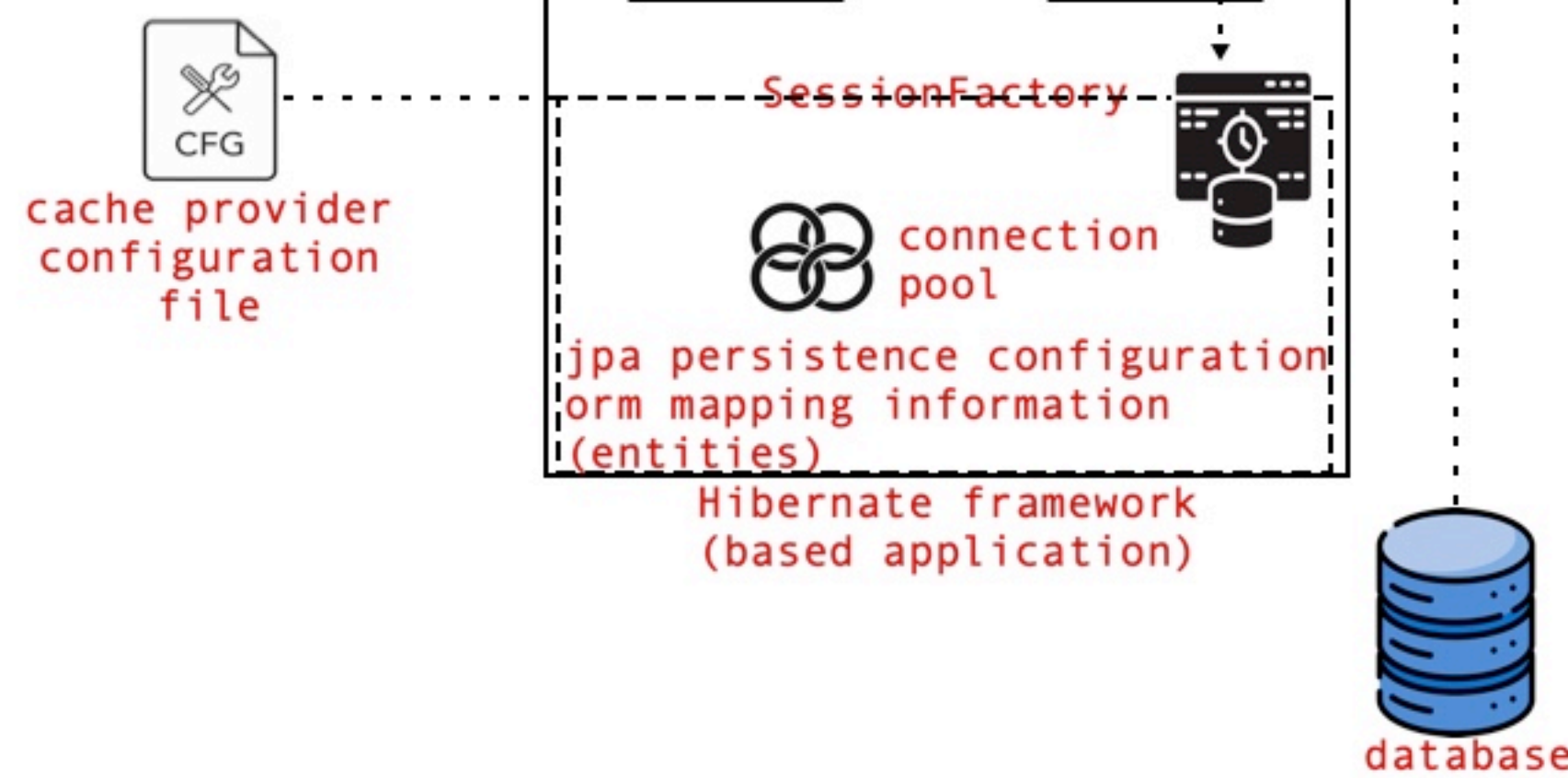


[different users are querying same manufacturer]

```

class ...Dao {
    ... fetch...() {
        Session session = sessionFactory.openSession();
        Manufacturer manufacturer = session.get(Manufacturer.class, mno);
        // house keeping
        return manufacturer;
    }
}

```



```

@Entity
@Table(name="manufacturer")
@Cacheable
@org.hibernate.annotations.Cache(usage=CacheConcurrency
    Strategy.READ_WRITE)

```

```

class Manufacturer {
    @Id
    @GeneratedValue(...)
    @Column(name="manufacturer_no")
    int manufacturerNo;
}

```

Cache Concurrency Strategy:  
 READ\_ONLY  
 READ\_WRITE  
 NONSTRICT\_READ\_WRITE  
 TRANSACTIONAL