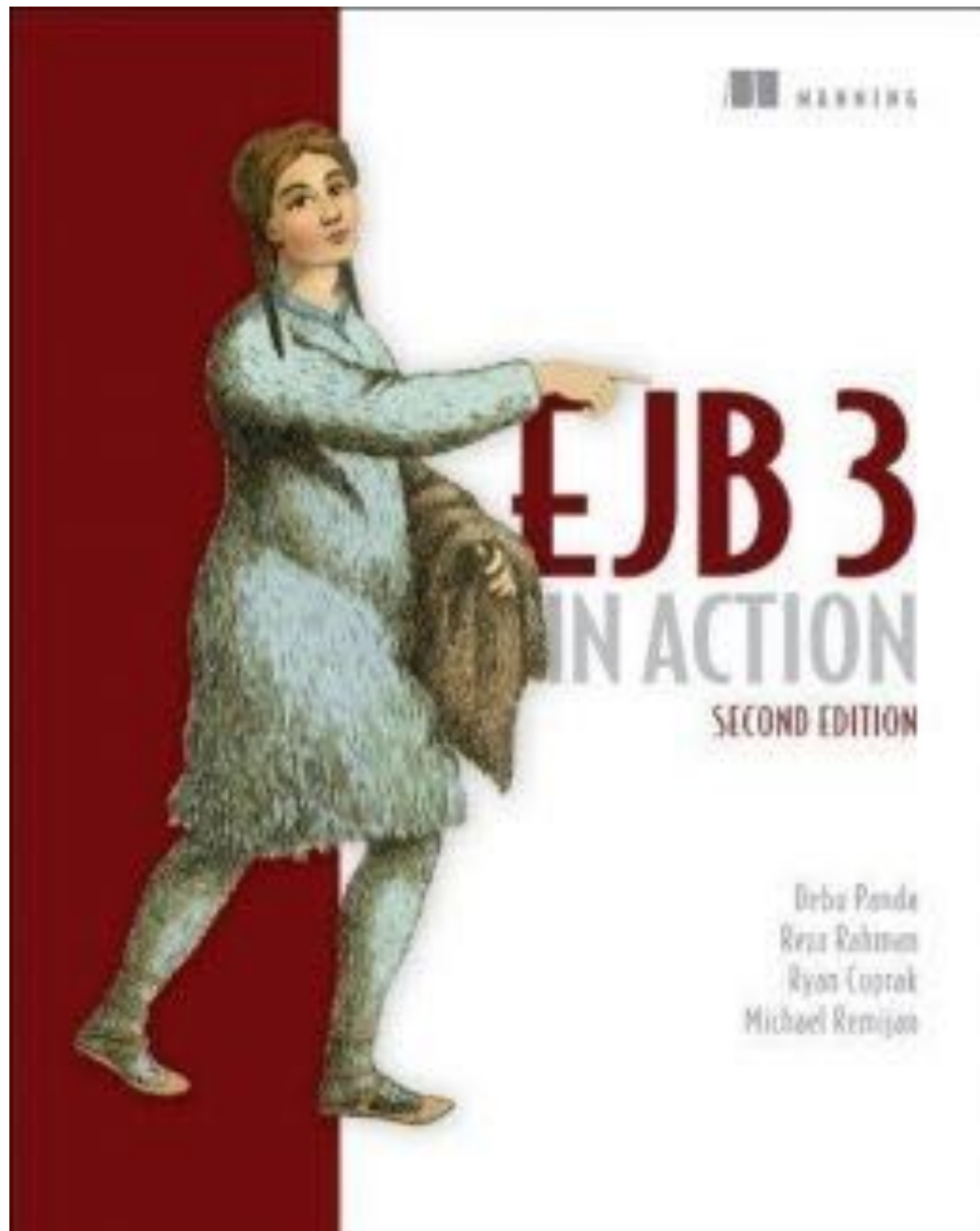




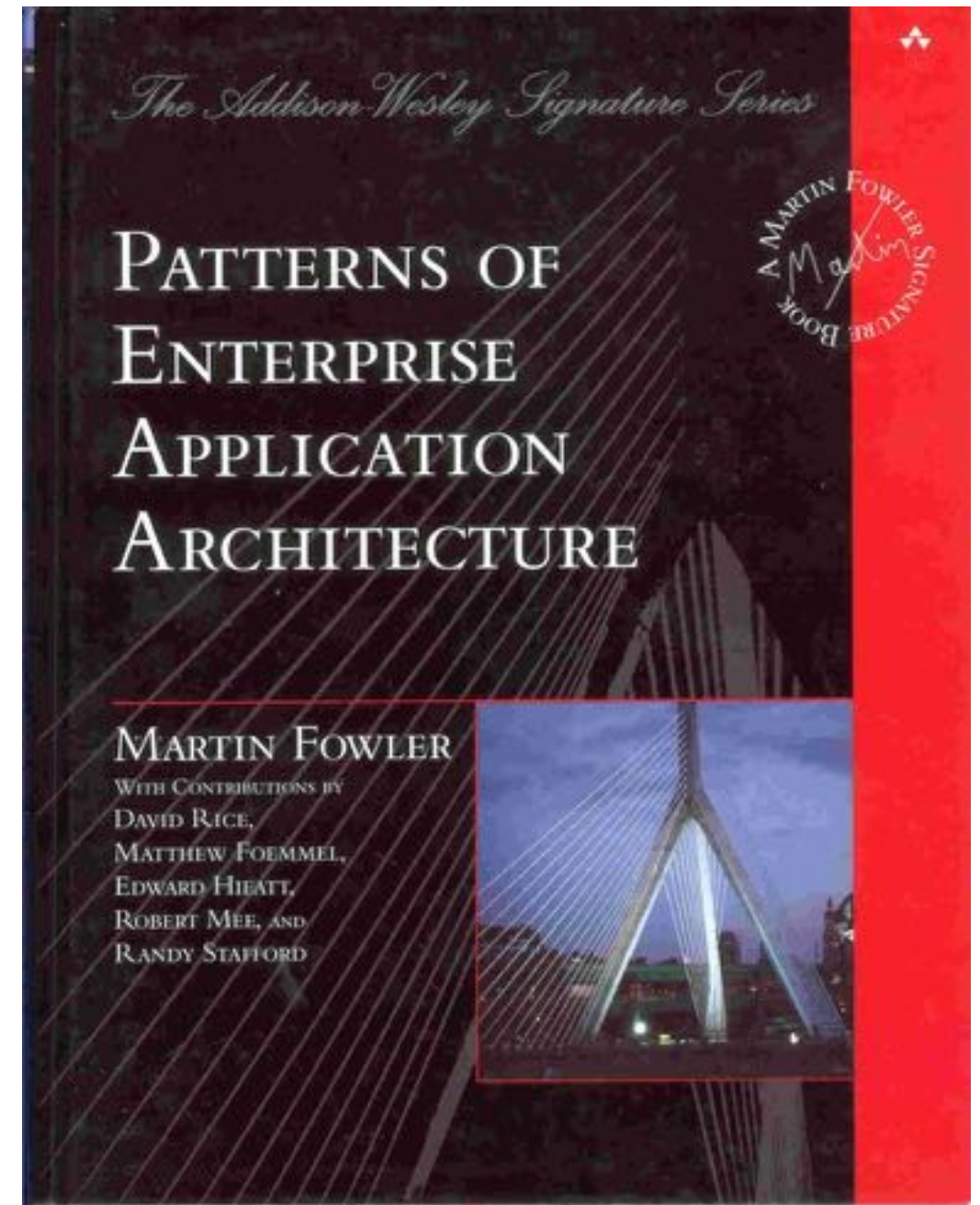
Java EE, EJB Basics

Philippe Collet, contains 78,3% of slides from
Sébastien Mosser
Lecture #3

Bibliography

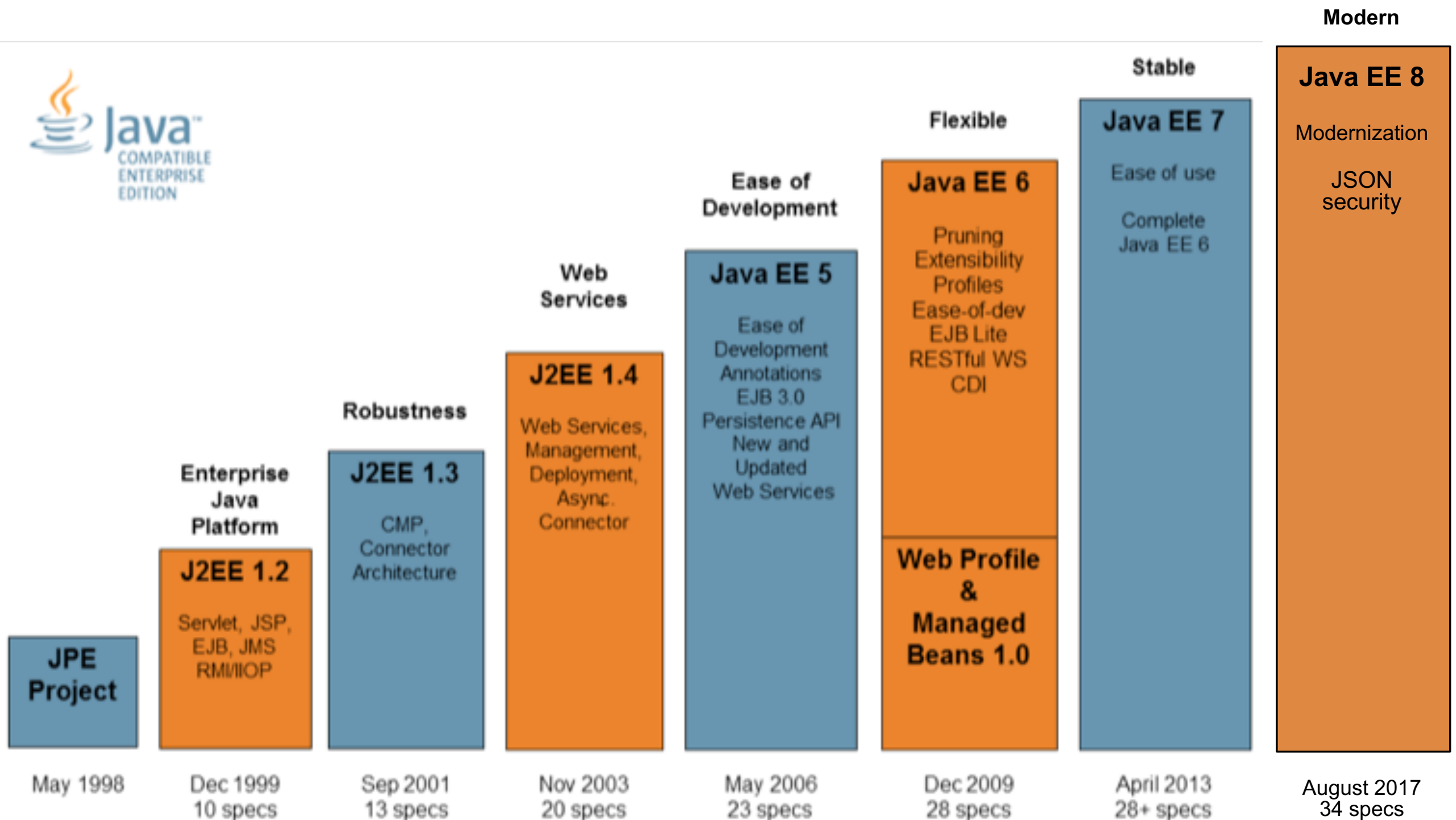


[EJB3 in Action, 2nd ed., 2014]

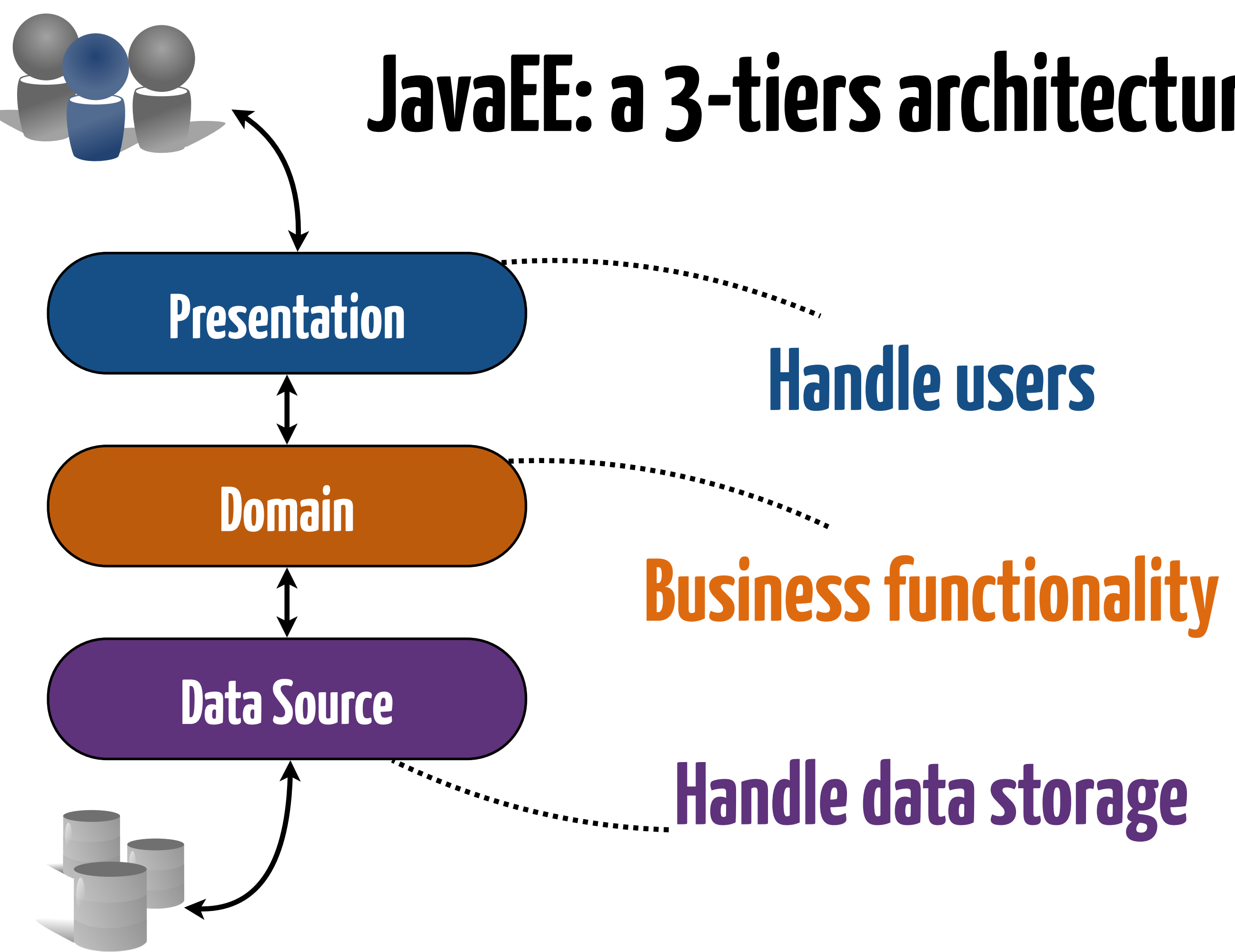


[PoEAA, 2002]

From J2EE to JavaEE



JavaEE: a 3-tiers architecture

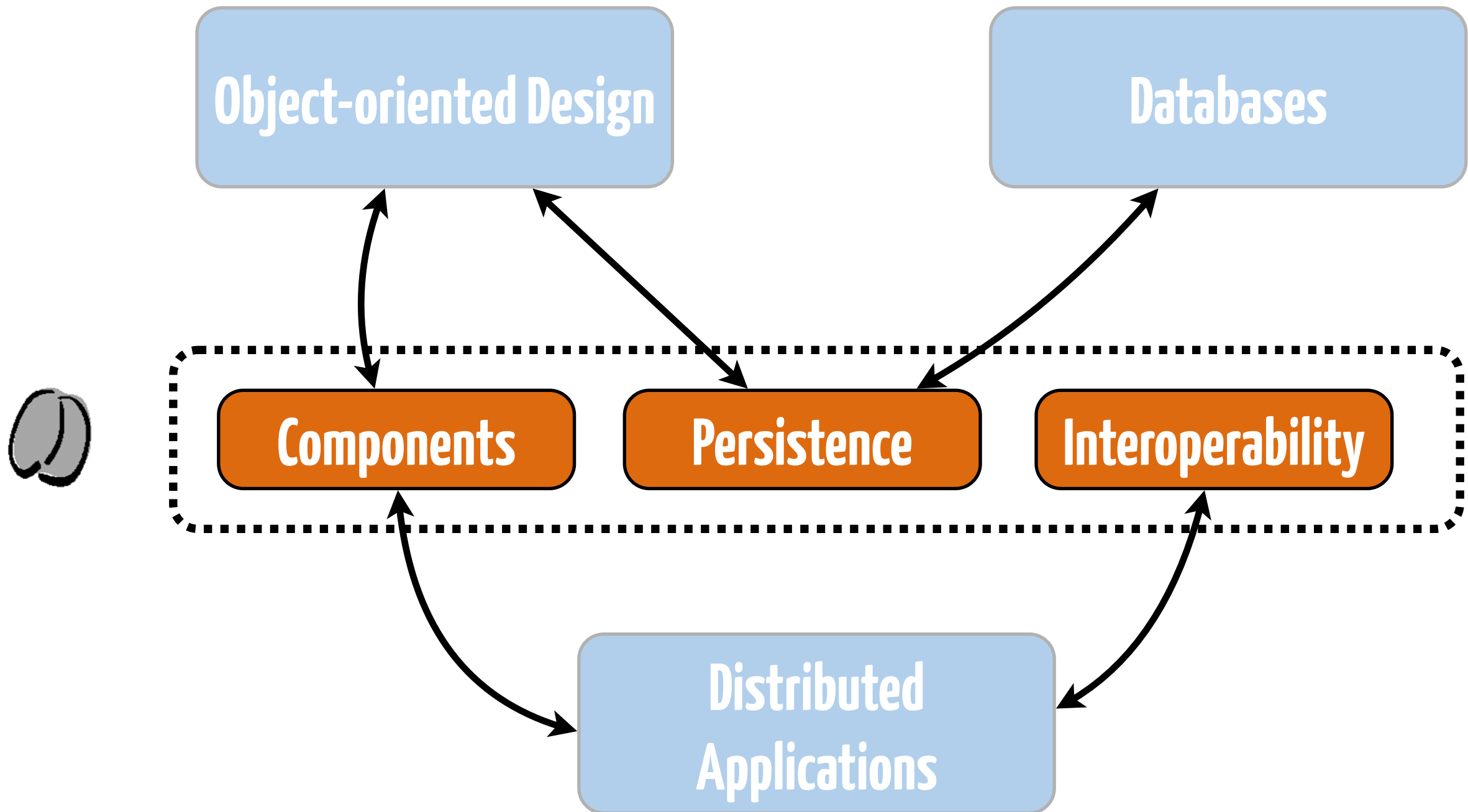




Enterprise Java Beans

101

Applications Server: Dependencies



EJB as **Components**

Interface **reuse**

Encapsulate

application

Behavior

Book seller

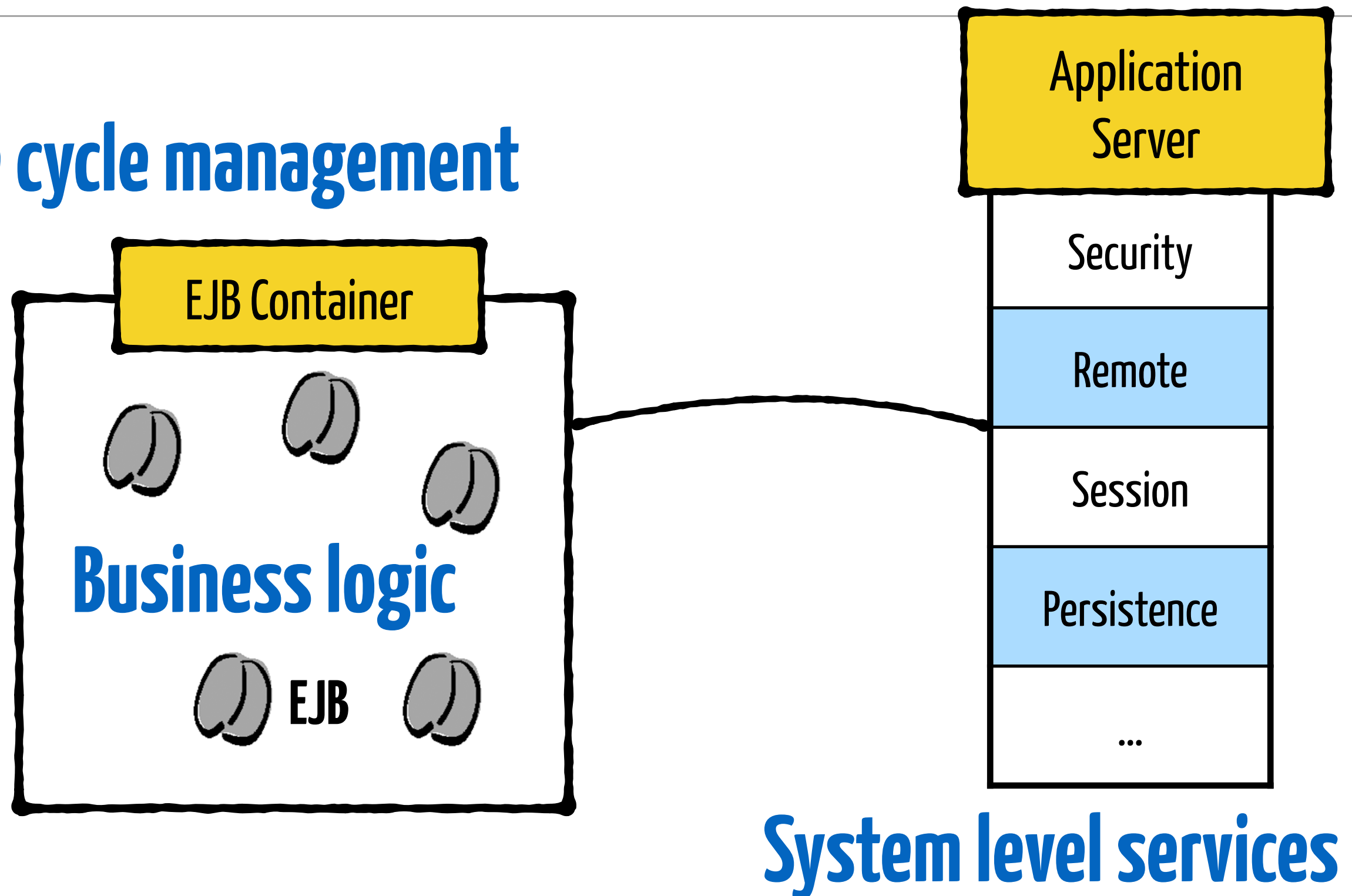


Credit Card
Payment

Music seller

JavaEE as the **specification** for

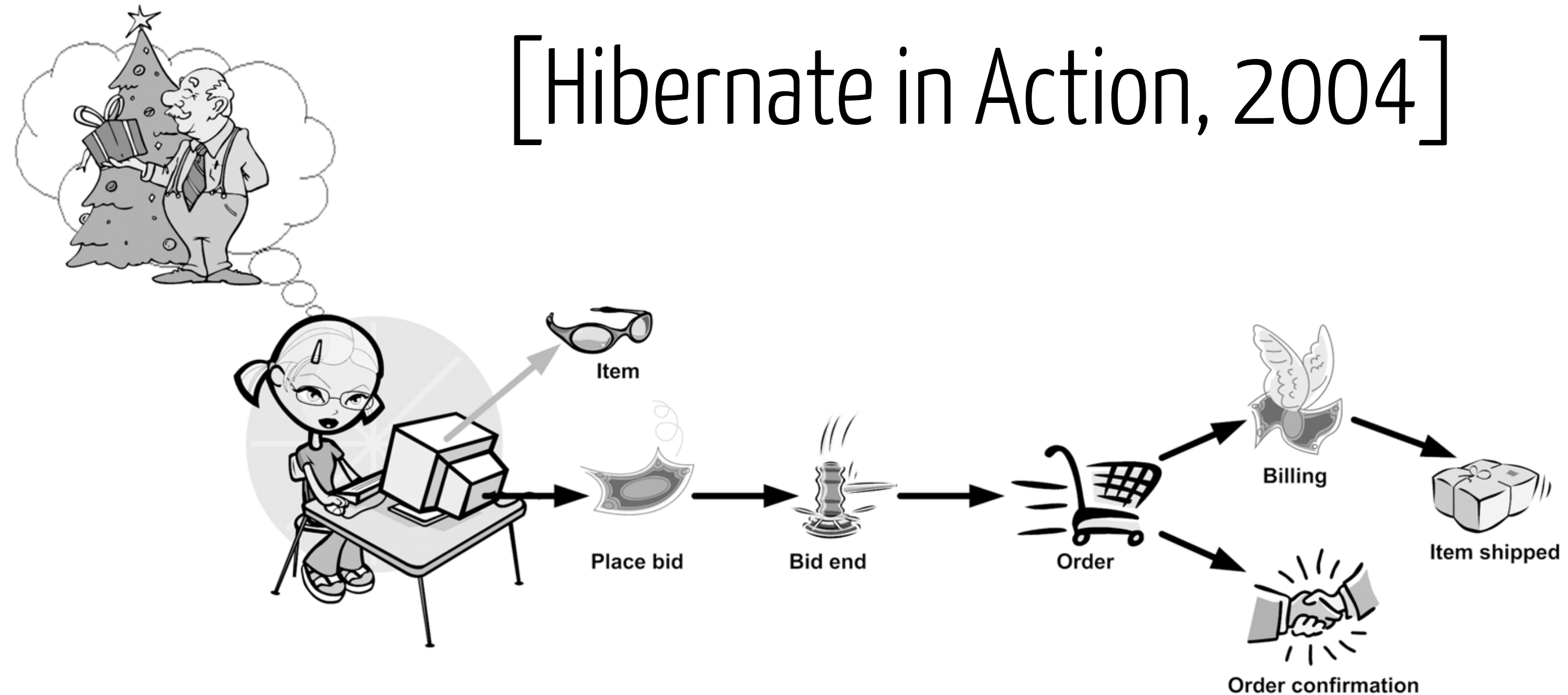
Life cycle management



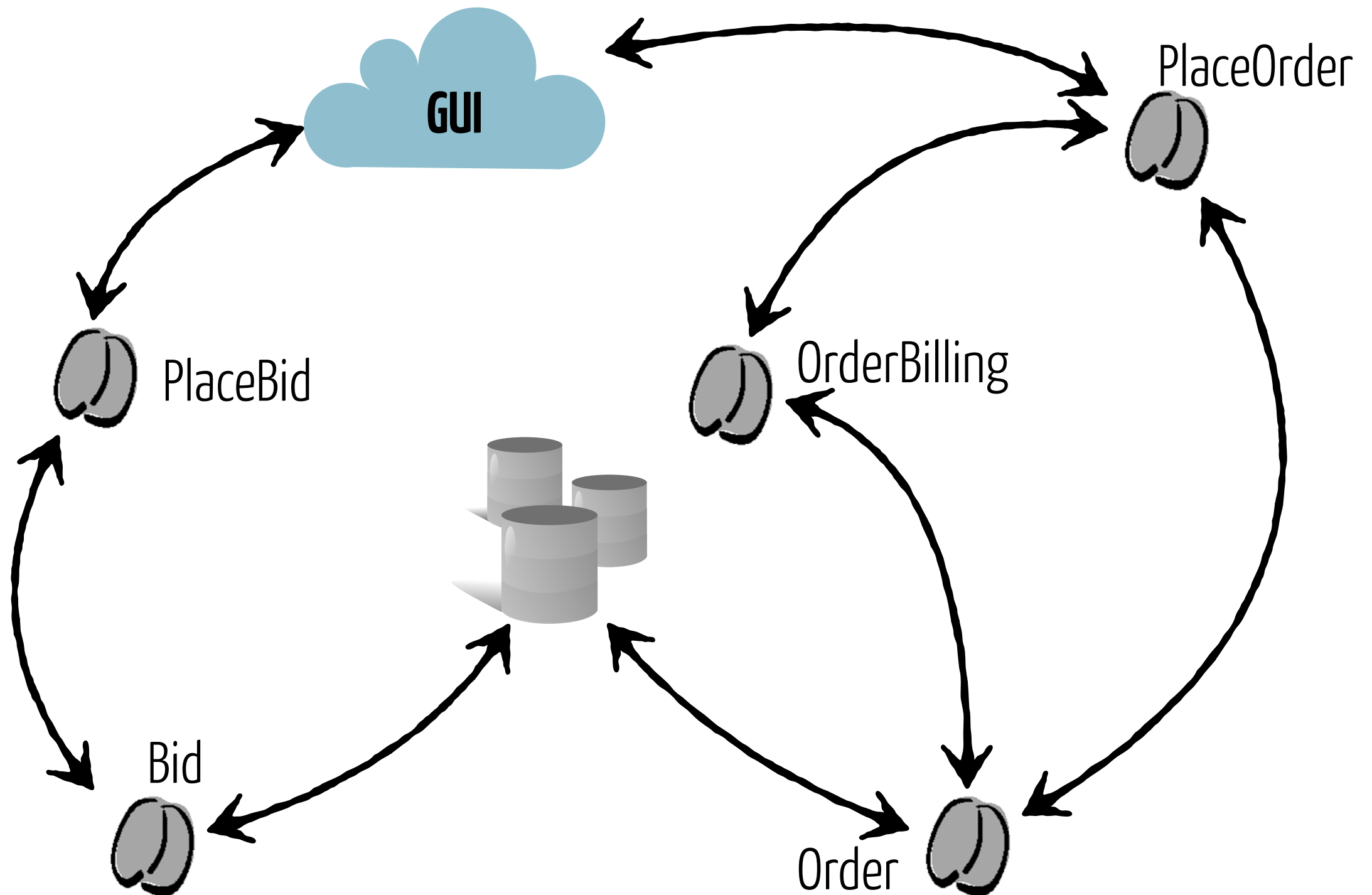
System level services

The ActionBazaar example

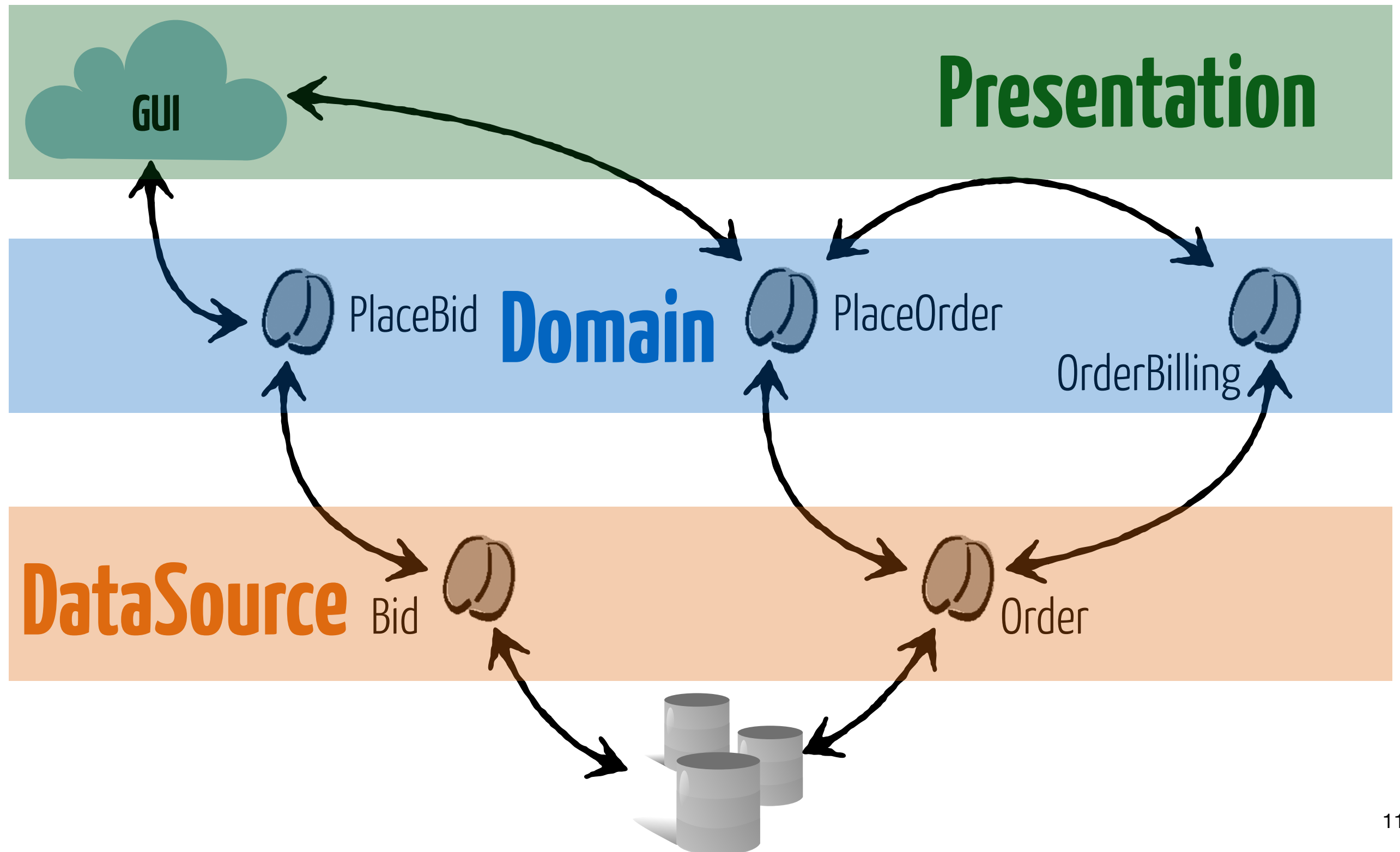
[Hibernate in Action, 2004]



Identifying Components



3-tiers Architecture



Rule of Thumb

Domain Beans interfaces as **Verbs**

DataSource Beans as **Nouns**

What is a **POJO**?

An ordinary Java Object

Not bound to any restriction such as

Extending some classes

Implementing some interfaces

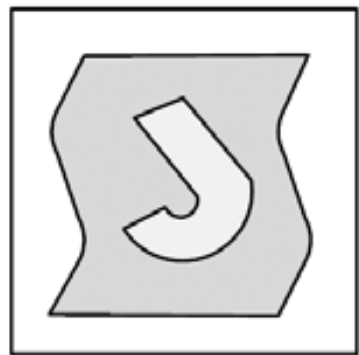
Containing some annotations

What is a **Bean**?

A POJO that
is **serializable**
has **no-argument constructor**

allows access to properties using **getters / setters** with a **simple naming convention**

What is a **Enterprise Java Bean**?



POJO



Annotation



EJB

EJB v3 = "**Plain Old Java Object on Steroids**"

POJO on Steroids



```
public interface HelloUser {  
    public void sayHello(String n);  
}
```

```
@javax.ejb.Stateless
```

```
public class HelloUserBean implements HelloUser{  
    public void sayHello(String n) {  
        System.out.println("Hello, " + n + "!");  
    }  
}
```

Different **Flavors**

Domain

Session Bean

Message-driven
Bean

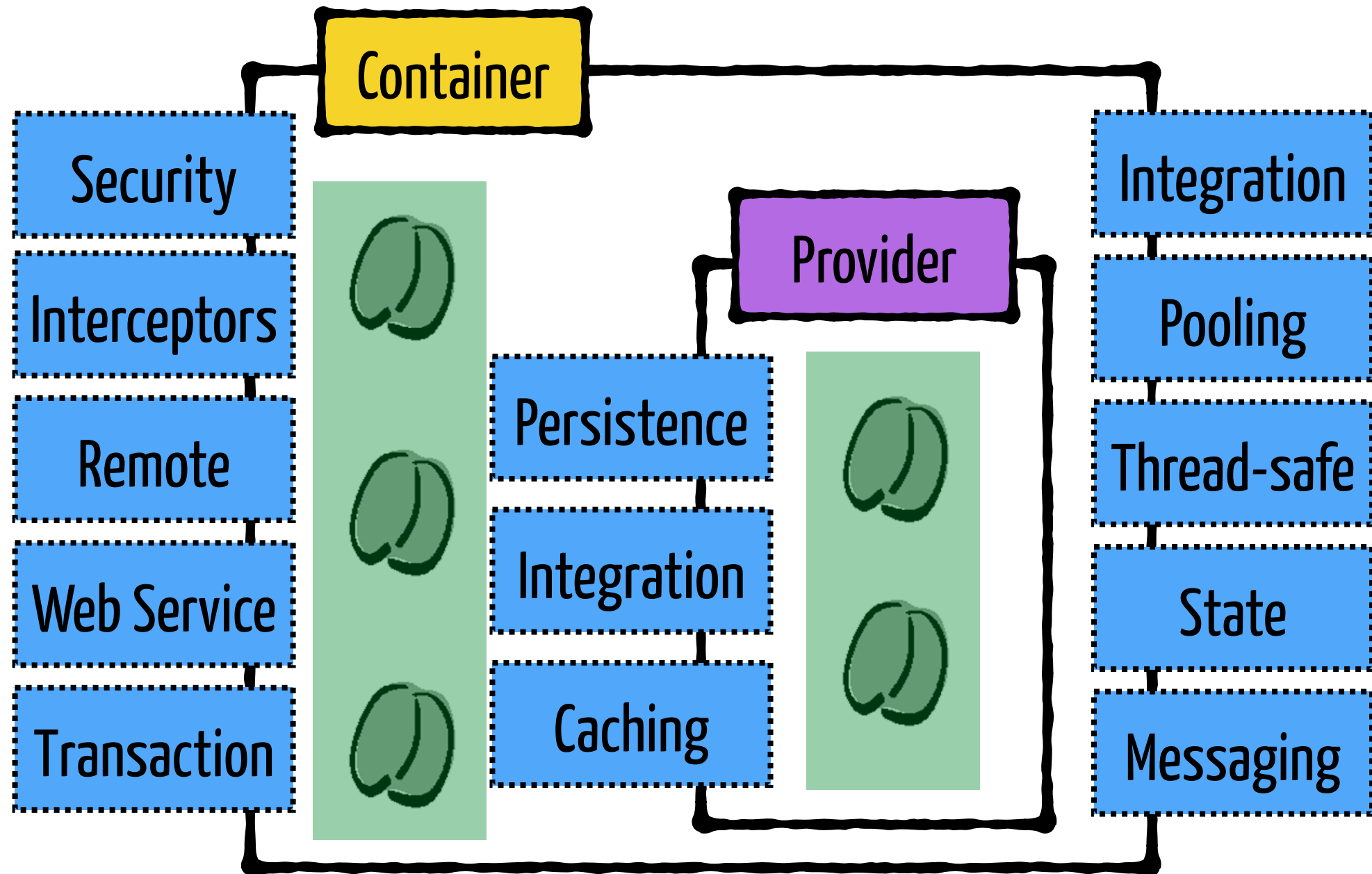
DataSource

Entity Bean

Provider

Container

EJBs **Services**: focus on **Business**!



Session & Entity

