ORM & J2EE Persistence Basics
Session Bean Basics

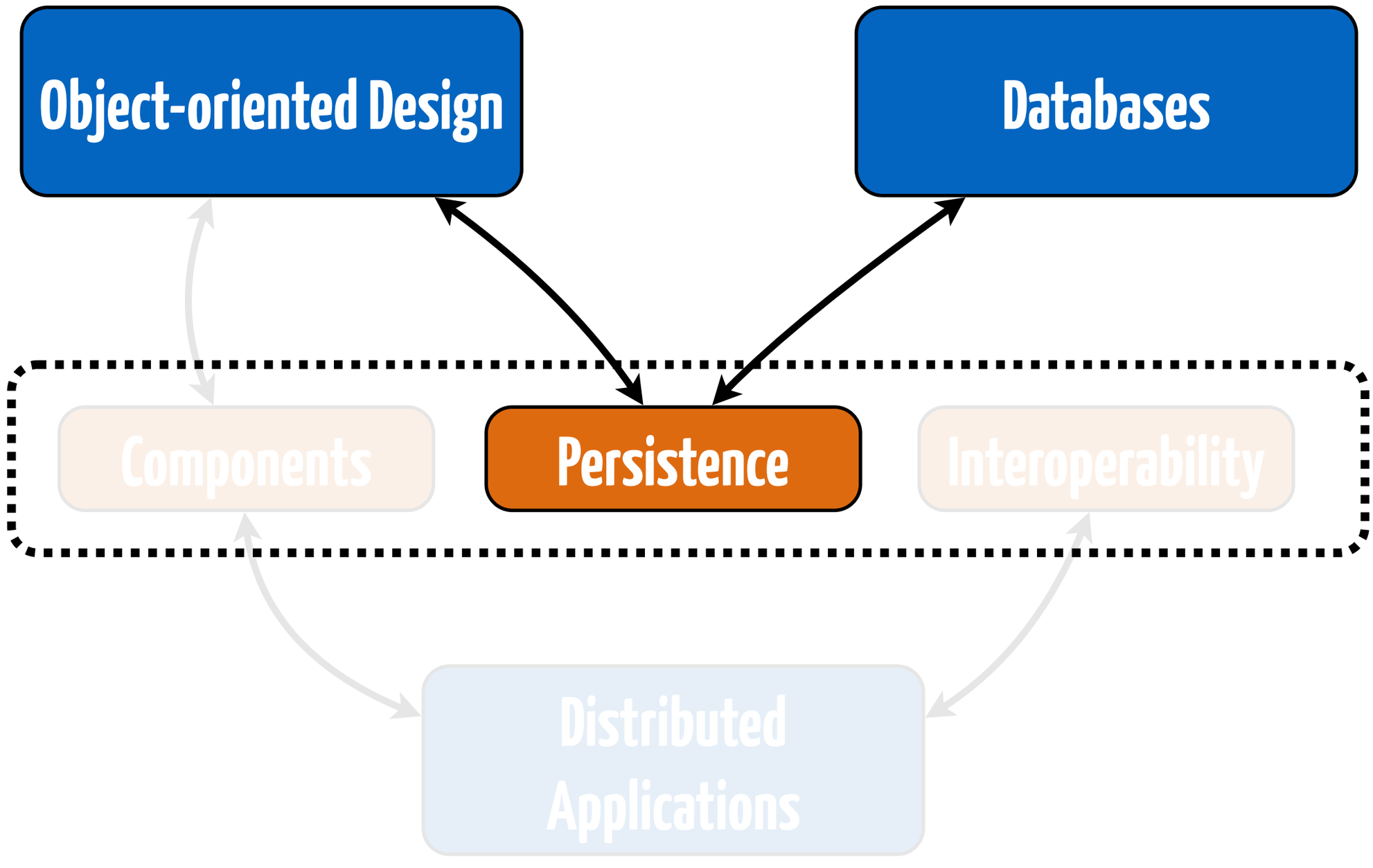
pictures: sxc.hu, pixabay

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

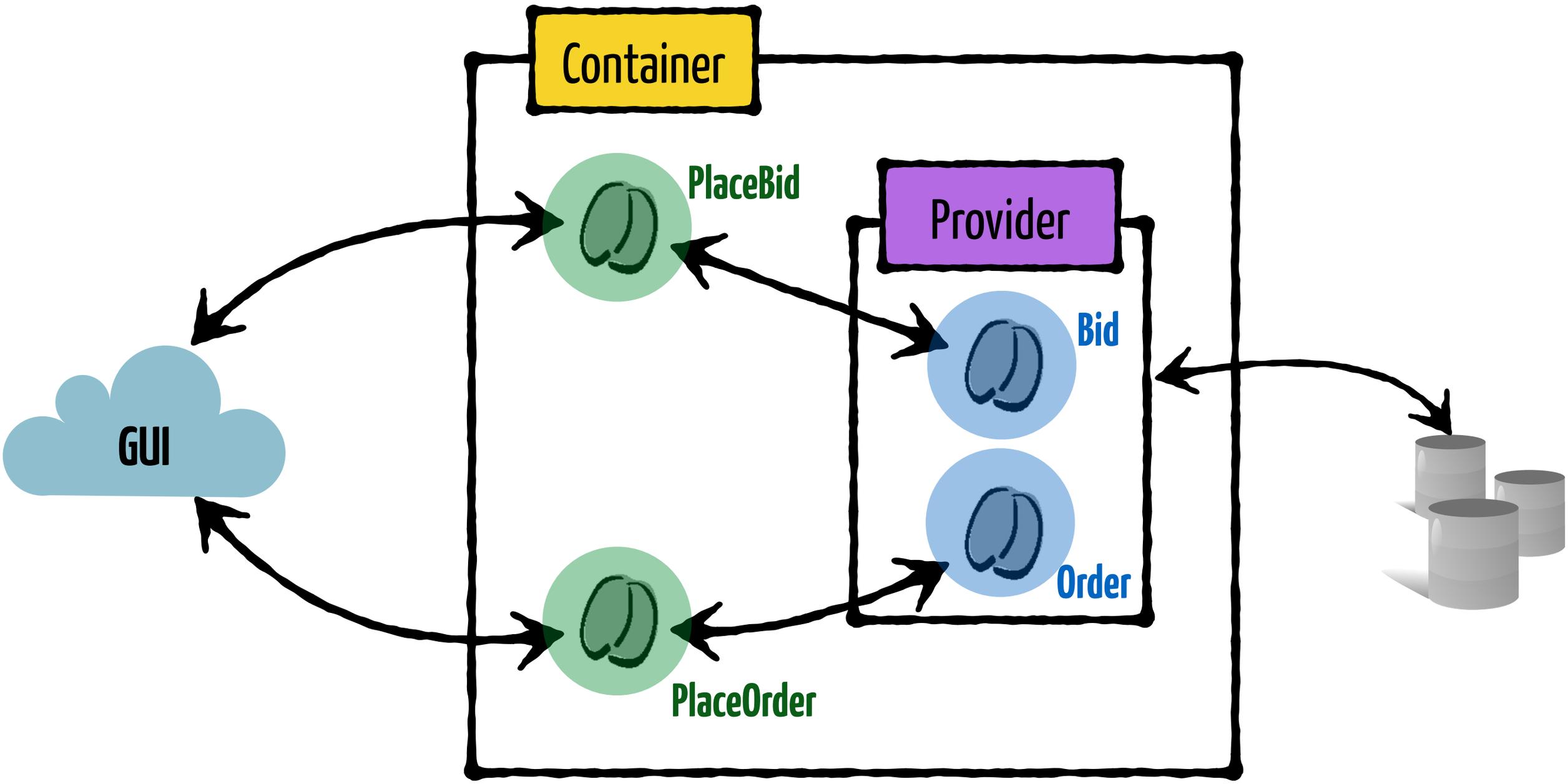
Lecture #4

Object-Relational Mapping 101

Applications Server: Dependencies



Session & Entity - Reminder

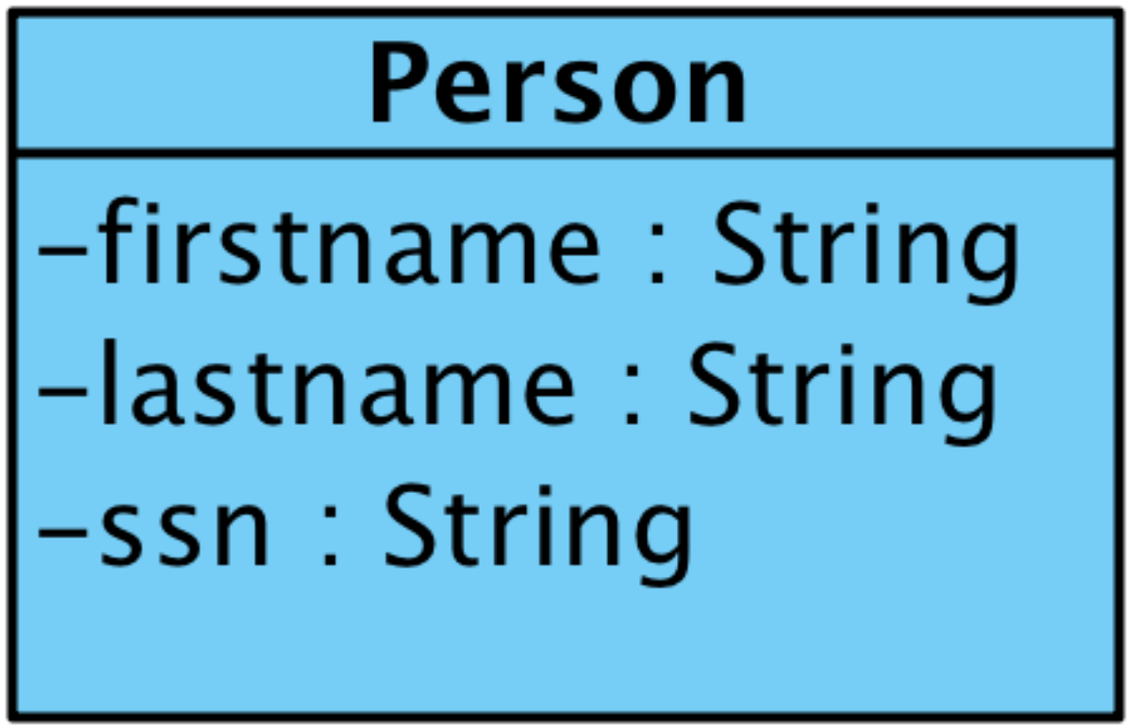


Impedance Mismatch

Object-Oriented	Relational
Classes	Relation (table)
Object	Tuple (row)
Attribute	Attribute (column)
Identity	Primary Key
Reference	Foreign Key
Inheritance	N/A
Methods	~ Stored Procedure

[EJB3 in Action]

Example of **Domain Model**



first_name	last_name	ssn
Sébastien	MOSSER	16118325358
...		

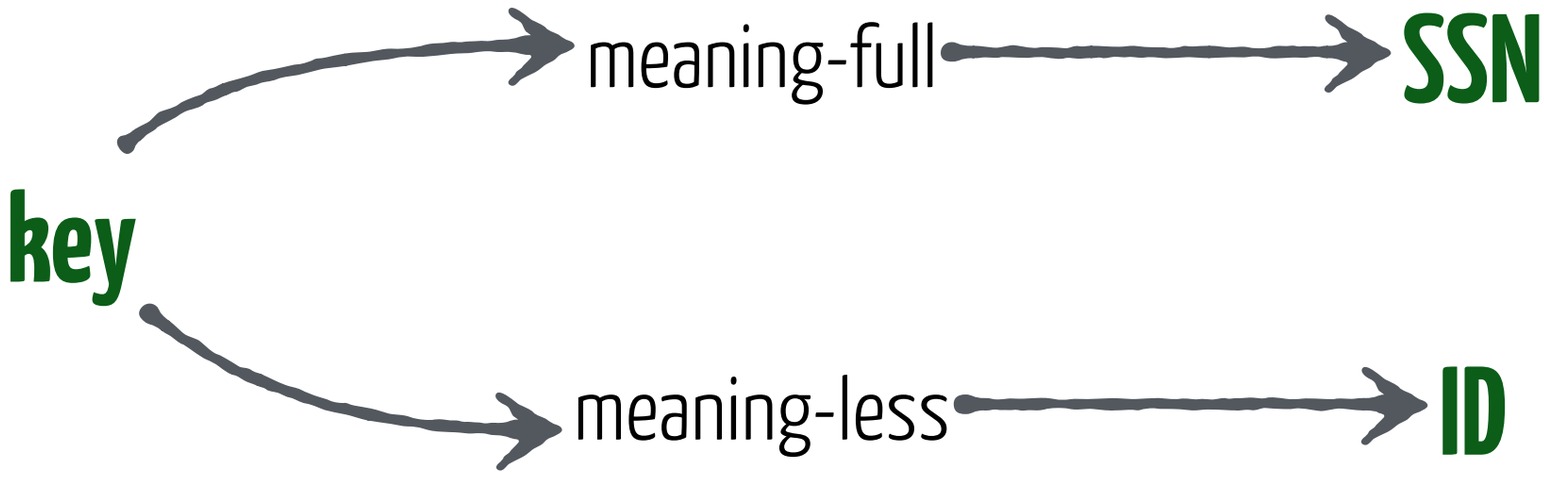
[PoEAA]

Problem: Domain Object Identity

```
Person p1 = new Person("X","Y","DDMMYYXXXXX")
Person p2 = new Person("X","Y","DDMMYYXXXXX")
```

How to support persons **uniqueness**?

Candidates for **Key** role



- Necessary condition:
  - **Key must be unique**
- Nice-to-have condition:
  - **Key should be immutable**
- **Compound** versus **Simple**

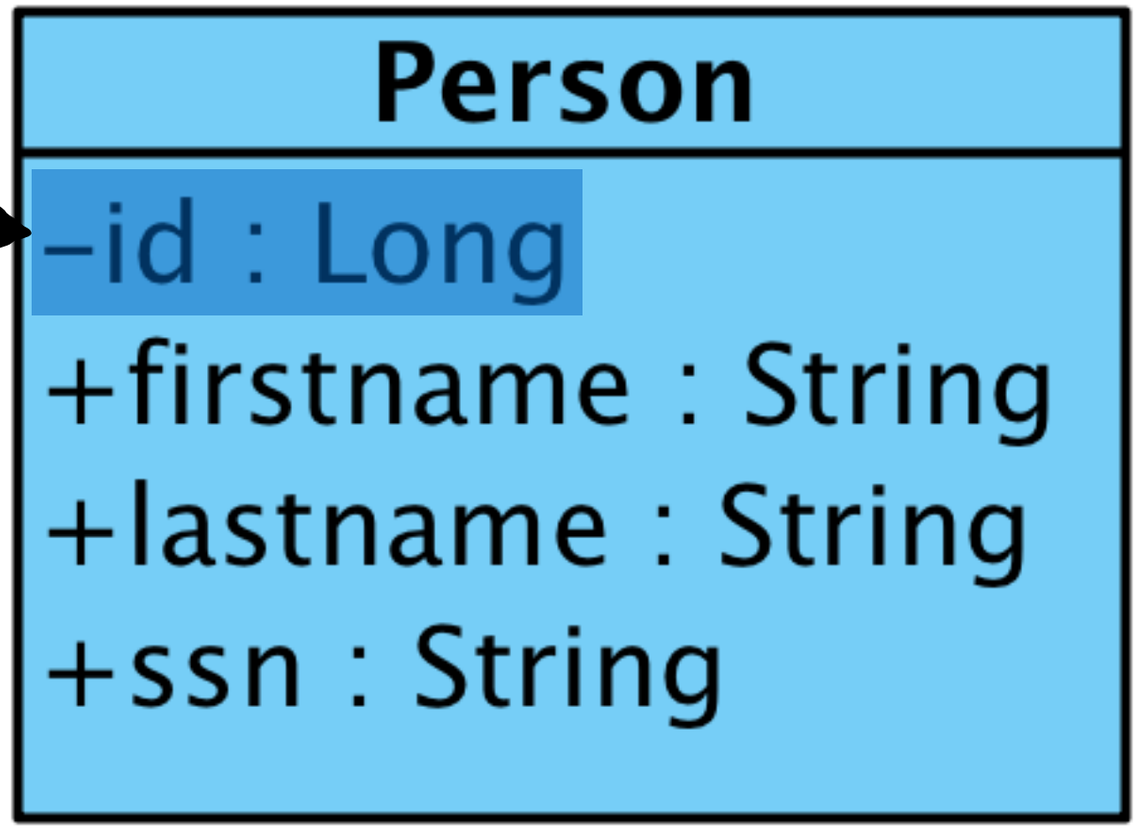
[PoEAA]

Pattern: **Identity Field**

Not visible at
the business level

auto-generated

GUID DIY

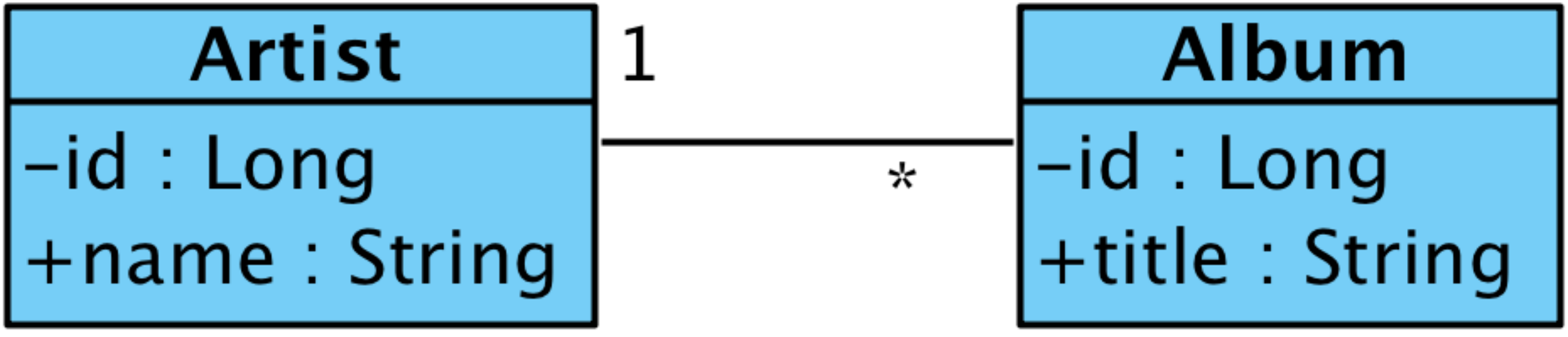


[PoEAA]

QUESTION

When are 2 customers equals?

Problem: Representing associations



artists

id	name
1	Linkin Park
	…

albums

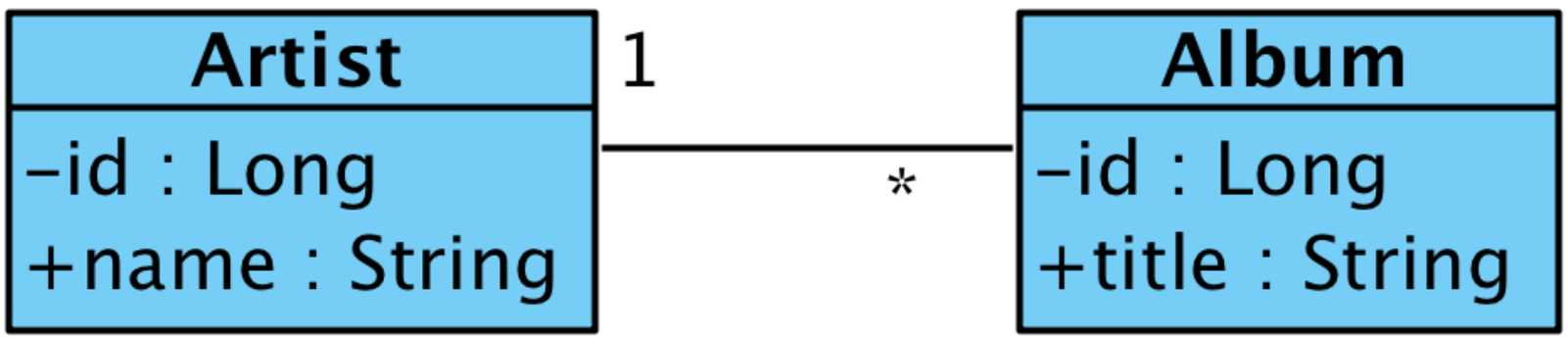
id	title
1	A Thousand Suns
2	Minutes to Midnight
	…

[PoEAA]

QUESTION

How to bind customers to orders?

Solution #1: Association Table [M-N]



artists

id	name
1	Linkin Park
...	

albums

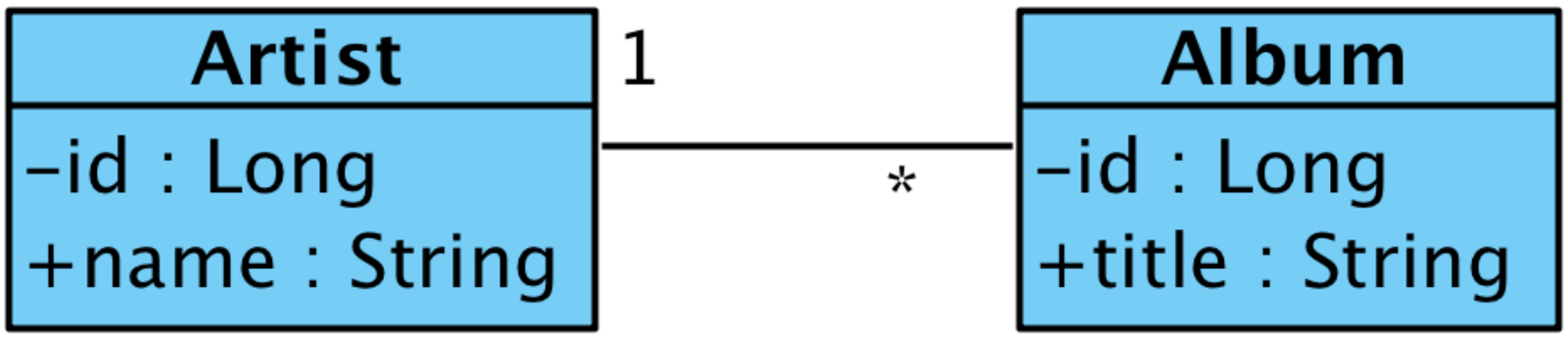
id	title
1	A Thousand Suns
2	Minutes to Midnight
...	

artists_to_albums

artist_id	album_id
1	1
1	2
...	

[PoEAA]

Solution #2: **Foreign Key** [1-N]



artists

id	name
1	Linkin Park
...	

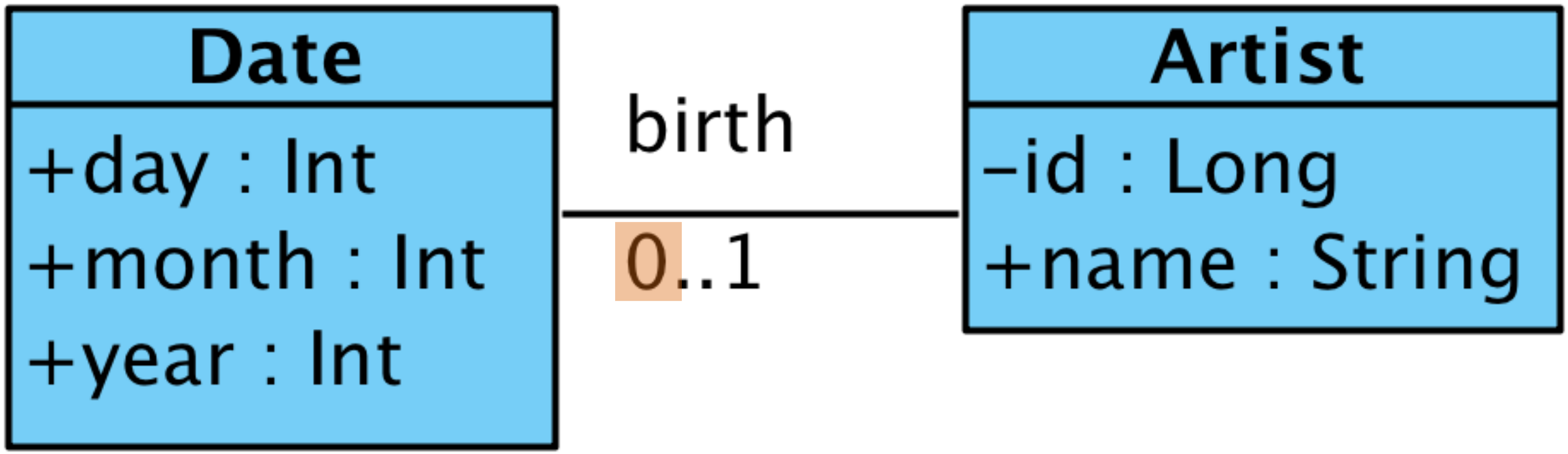
albums

id	title	artist_id
1	A Thousand Suns	1
2	Minutes to Midnight	1
...		

or **[1-N] ≡ [M-N]** when N =1

[PoEAA]

Solution #3: **Relation Merge** [1-1]



artists

id	name	birth_day	birth_month	birth_year
1	Linkin Park	-1	-1	-1
...				

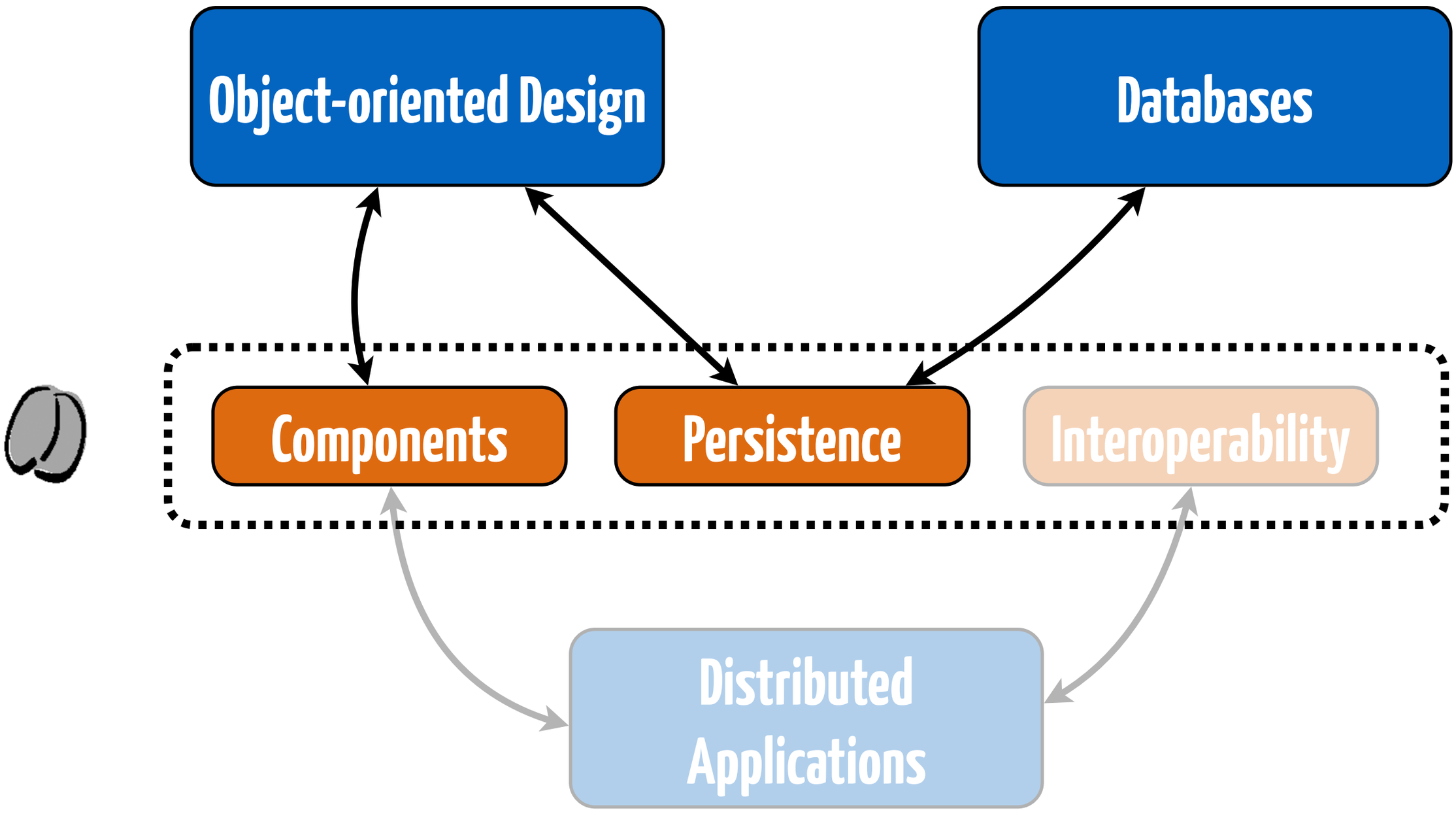
or **[1-1]** ≡ **[1-N]** when N =1

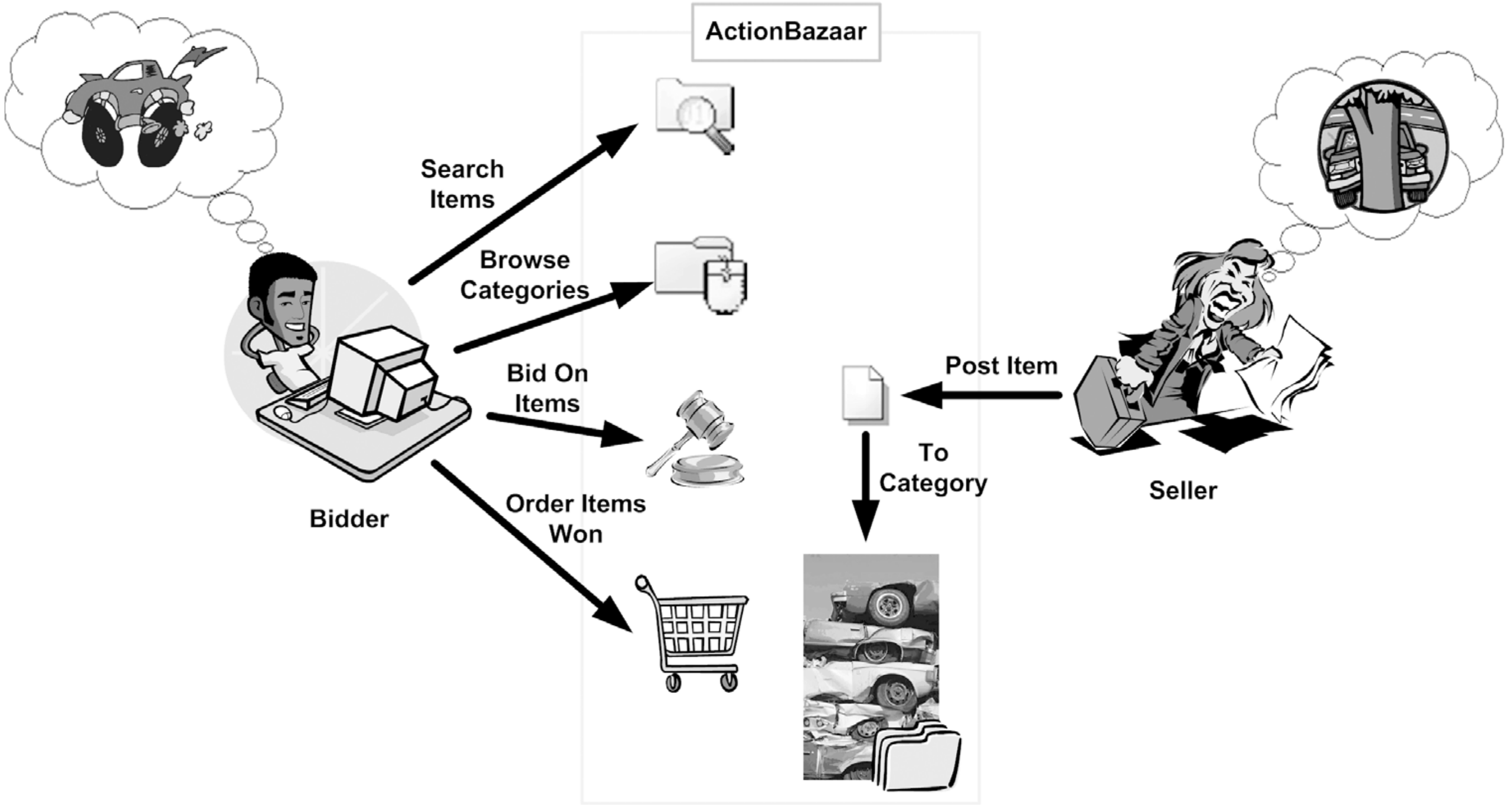
[PoEAA] or **[1-1]** ≡ **[M-N]** when M = 1 and N =1

Make your beans **persistent**

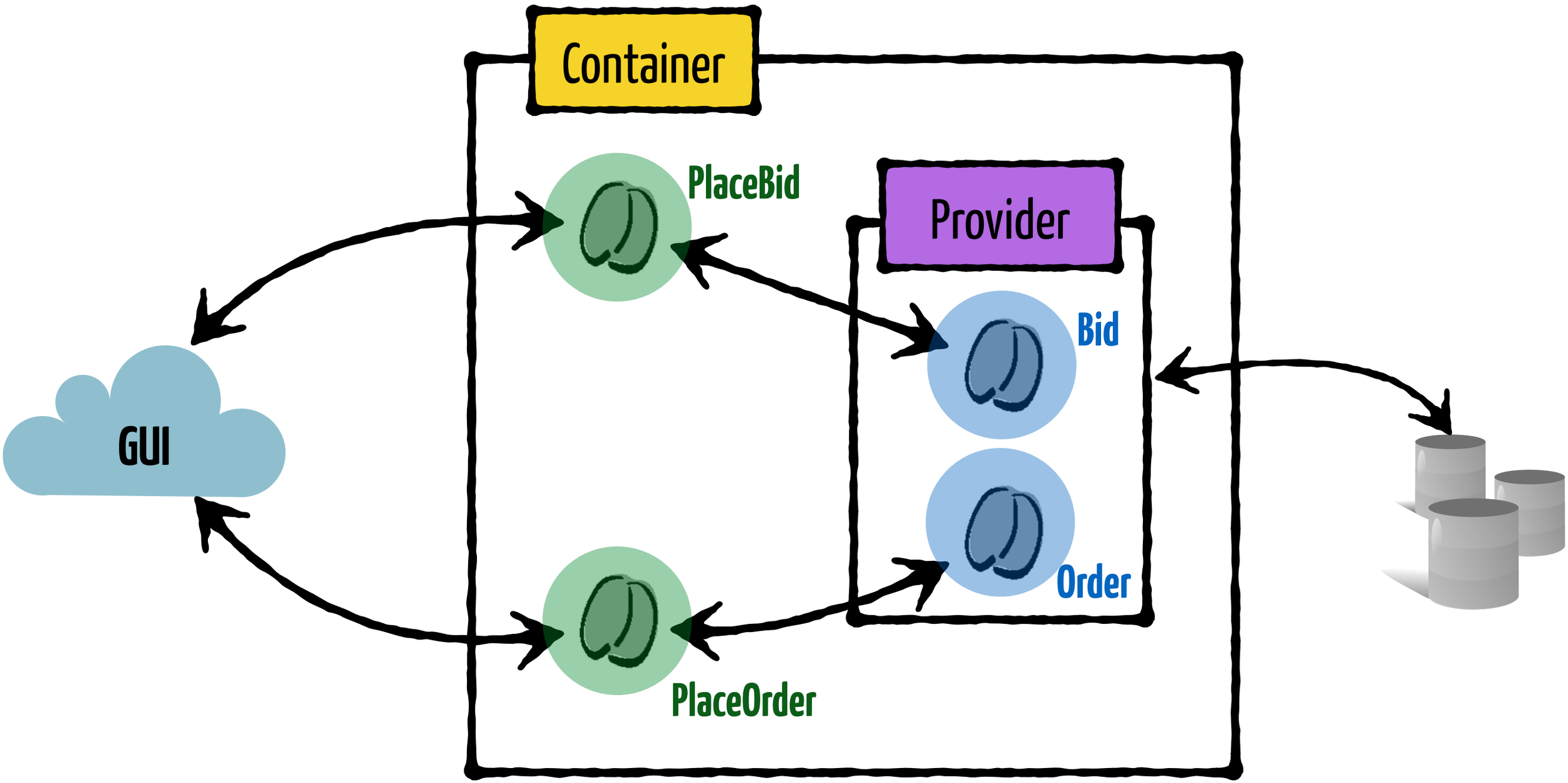
101

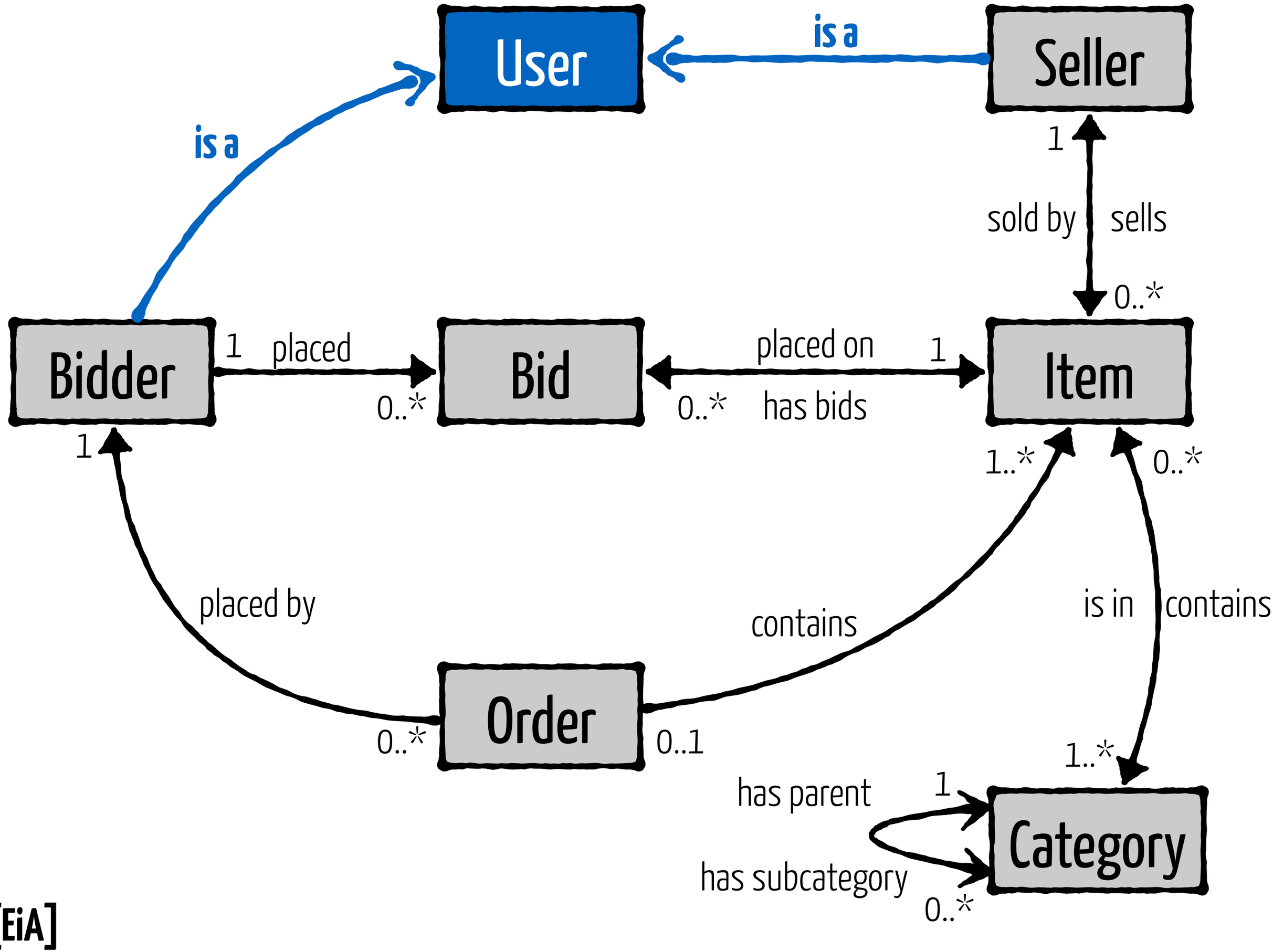
Applications Server: Dependencies



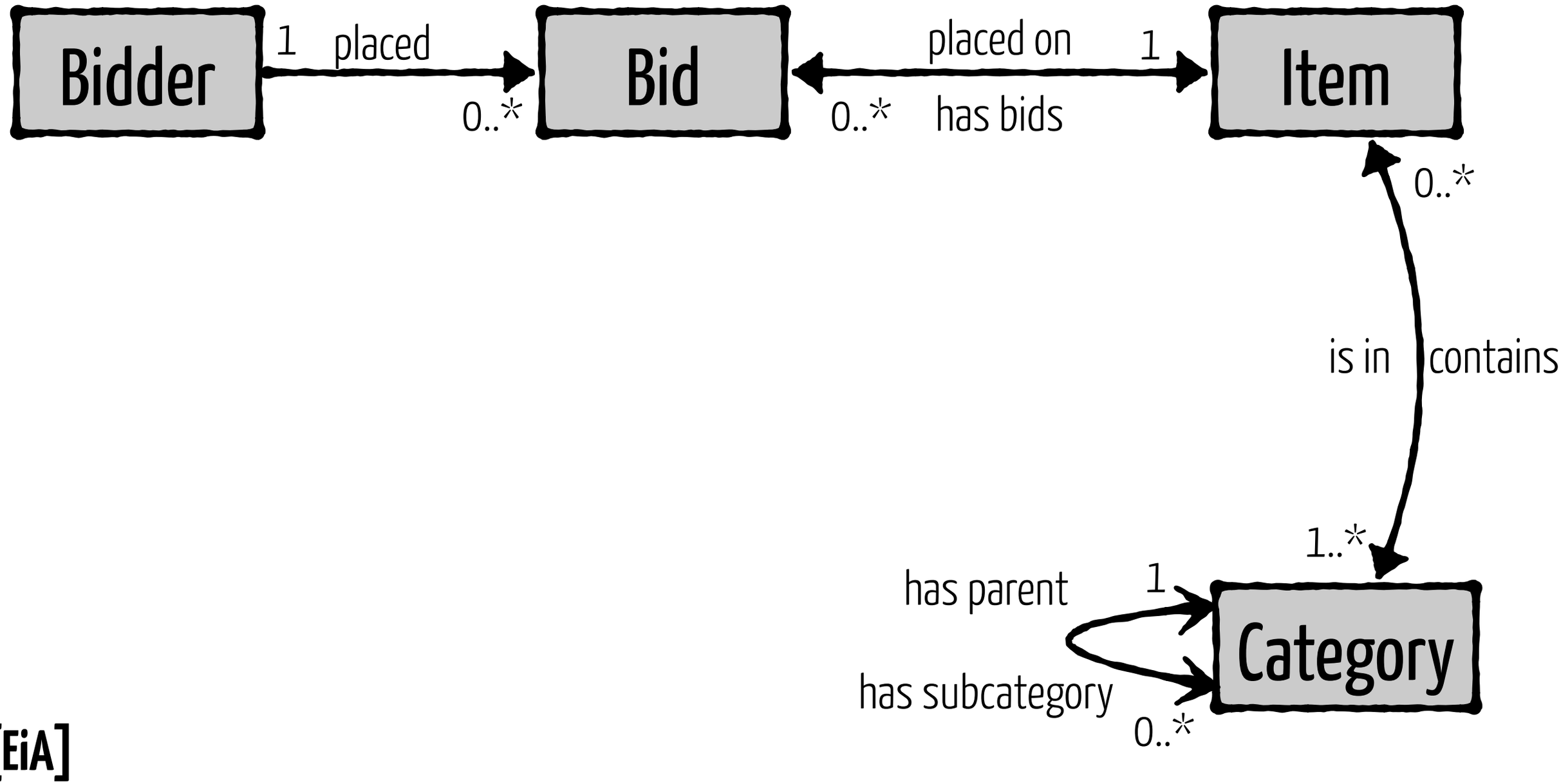

ActionBazaar
Search Items
Browse Categories
Bid On Items
Order Items Won
Bidder
Post Item
To Category
Seller

Session & Entity




User
is a
Seller
is a
1
sold by
sells
0..*
Bidder
1
placed
0..*
Bid
placed on
1
0..*
has bids
Item
1
1..*
0..*
placed by
contains
is in
contains
Order
0..*
0..1
1..*
has parent
1
Category
has subcategory
0..*

[EiA]


Bidder
1
placed
0..*
Bid
placed on
0..*
has bids
1
Item
0..*
is in
contains
1..*
has parent
1
has subcategory
0..*
Category

[EiA]

Category

```
@Entity
public class Category {

   public Category() { … }

   protected String name;

   public String getName() {
      return this.name;
   }

   public void setName(String n) {
      this.name = n.toUpperCase();
   }
}
```

property-based access

(JPA)

Simple Primary Key: @Id

```
@Entity
public class Category {
   // …

   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   public Long id;

}
```

Identifiers must define an "equals" method

Equality Relation definition

- equals is reflexive
- equals is symmetric
- equals is transitive
- equals is consistent
- equals uses null as absorbing element

It's complicated!

Relationships

Type of Relationship	Annotation
1-1	`@OneToOne`
1-n	`@OneToMany`
n-1	`@ManyToOne`
n-m	`@ManyToMany`

[EiA]

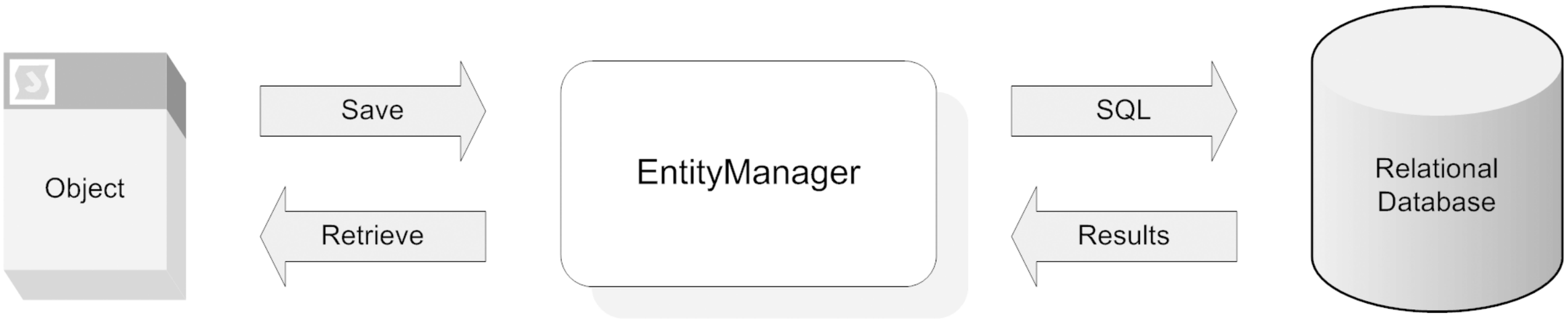
Class/table mapping

```
@Entity
public class Bid {
   @Id
   protected String bidId;
   @ManytoOne
   protected Item item;
}

@Entity
public class Item {
   @Id
   protected String itemId;
   @OneToMany(mappedBy="item")
   protected Set<Bid> bids;
}
```

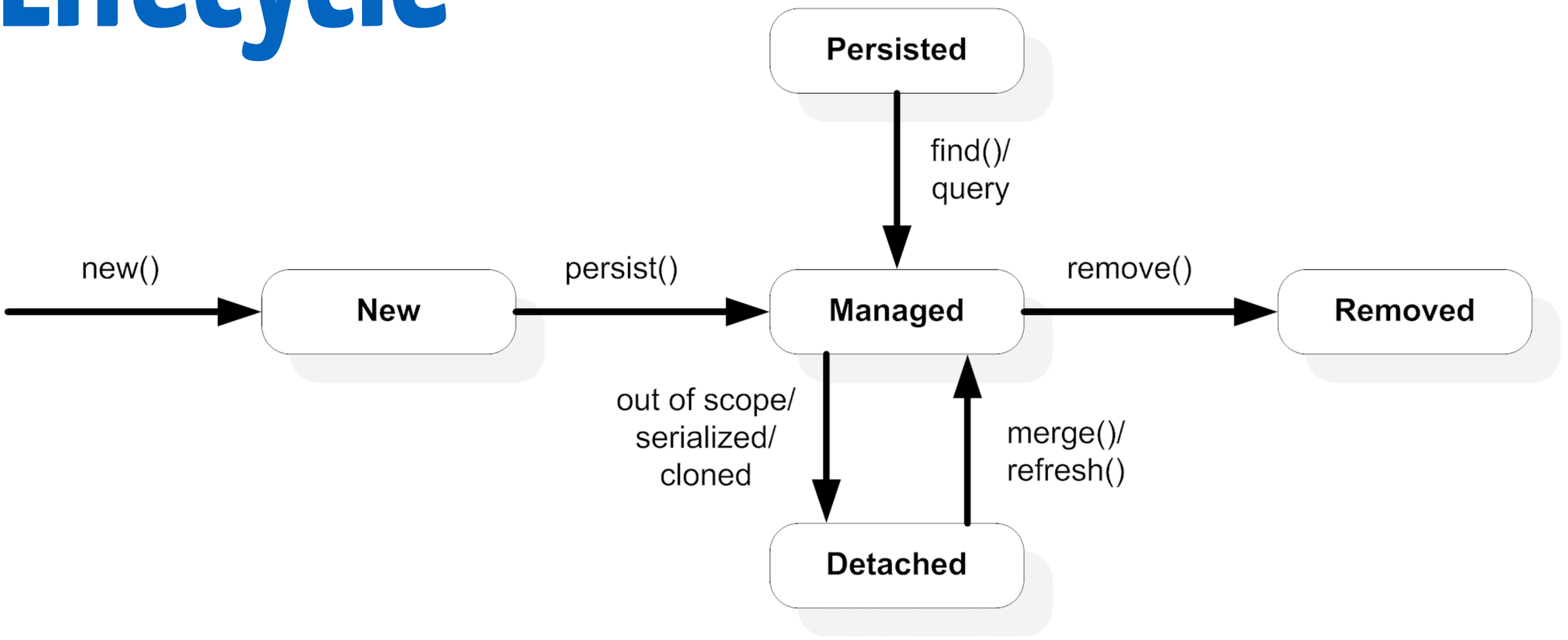
Id mapping

1-n mapping

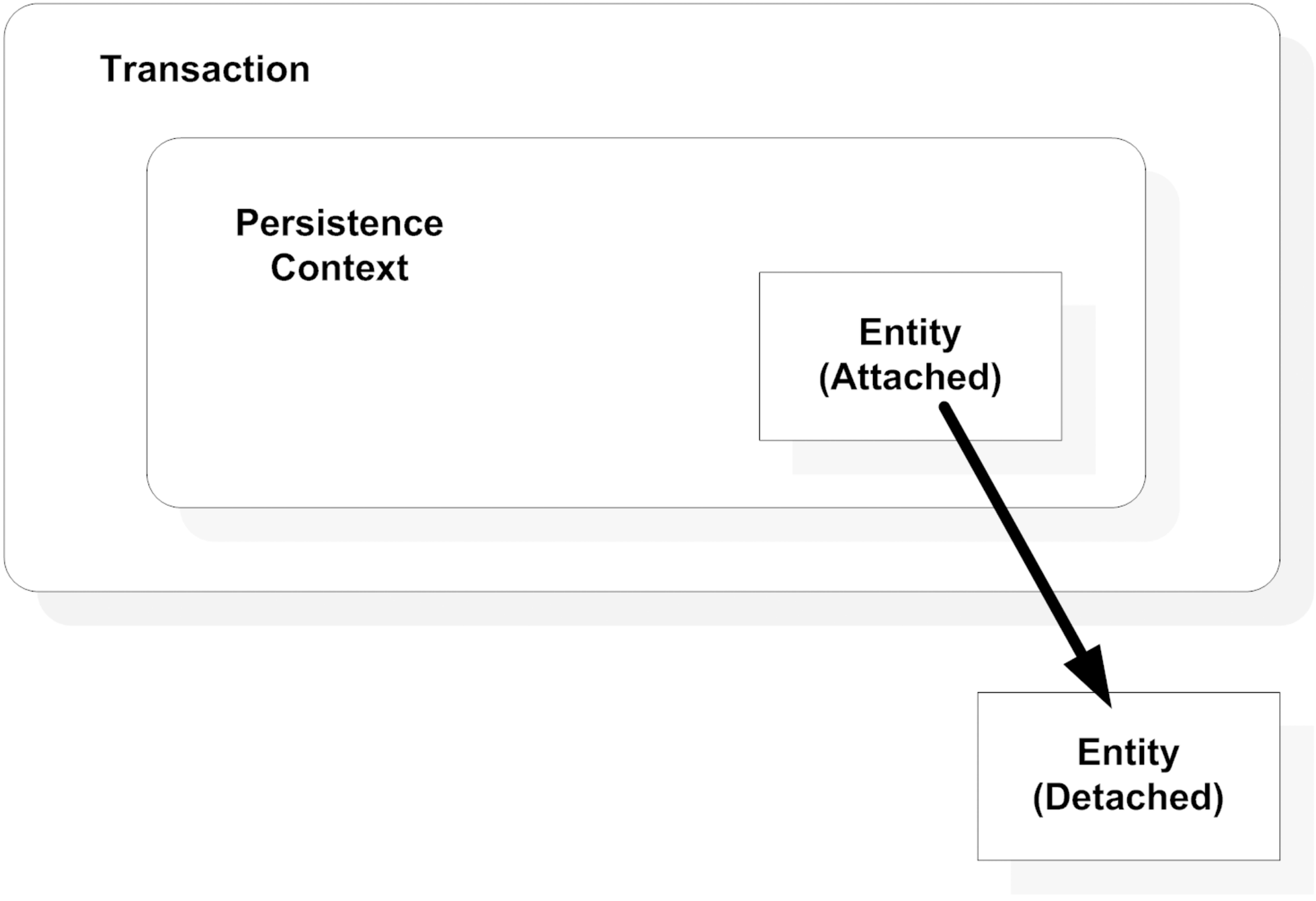
Entity Manager



Lifecycle



Handled by the provider + entity manager

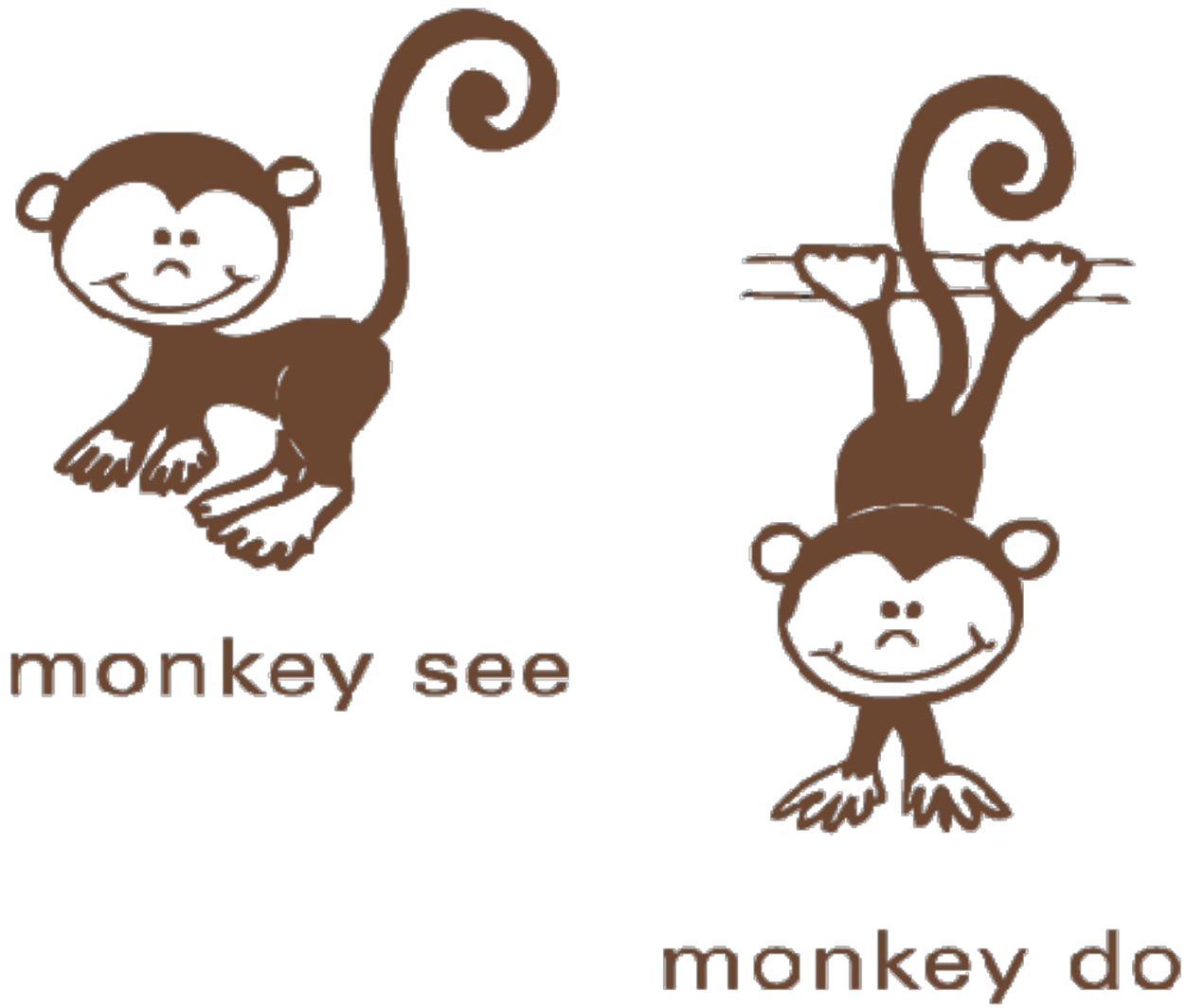

Transaction
Persistence Context
Entity (Attached)
Entity (Detached)

Persistence context is **Injected**

```
@PersistenceContext(unitName="admin")
EntityManager manager

@Resource
private UserTransaction transaction;

public void createAndStore() {

  AnEntityBean b = new AnEntityBean("Parameters");
  transaction.begin();
  try {
    manager.persist(b);
  } finally {
    transaction.commit();
  }

}
```

See [EiA], chapter 9

EJB are **standard**: Learn by **Example**!



http://infiltrated.net/index.php?option=com_content&view=article&id=43&Itemid=49

Starting persistence early?

https://github.com/collet/4A_ISA_TheCookieFactory/blob/develop/chapters/Persistence.md

First

More on a specific course later...