

## EJB Entity Beans – to be or not to be



### Component Based Development

Johan Eltes

Enterprise Systems Architect

Callista Enterprise AB

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB



## Introduction

### ✂ My path to EJB

- ✂ 2 years Mainframe (OOA + ??D + Cobol)
  - CICS 87
  - TSO 89-90
- ✂ 10+ years of OOD/OOP
  - Smalltalk 88-93
  - Objective-C 93-96
  - Java 97 - 00
- ✂ 7 years distributed objects
  - NeXT PDO 93-96
  - Corba 96-98
  - DCOM/MTS 97-00
  - RMI/EJB 98-00

### ✂ EJB Experience

- ✂ 5 projects:
  - 2 BEA WebLogic
  - 3 WebSphere AE

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB



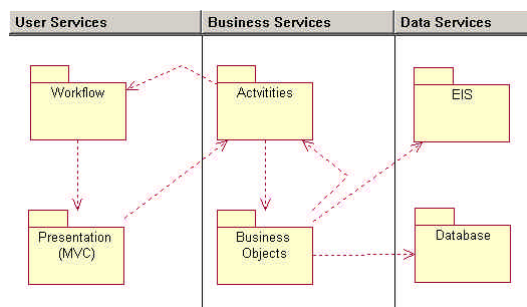
## Why talk 40 minutes about entity beans?

- ✗ Several architectural traps
  - ✗ Performance
  - ✗ Concurrency
  - ✗ Complexity
  - ✗ Use Model
  - ✗ Exception handling
- ✗ You rely on proprietary extensions
  - ✗ Associations
  - ✗ Locking strategies
  - ✗ Read-only markers
- ✗ A lot of inconcistent design guides, like...
  - ✗ Sun's Blueprint
  - ✗ IBM Redbooks

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB



## Reference Architecture



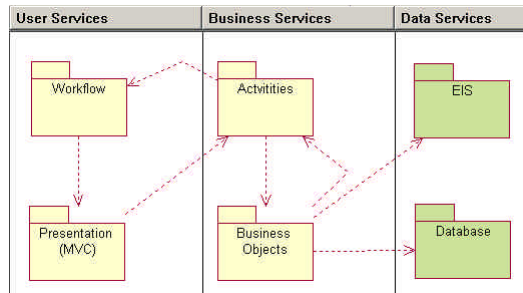
...for this presentation

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB



## What Is an EJB Entity Bean?

Scenario 1



[Sun J2EE Blueprint, section 5.2]

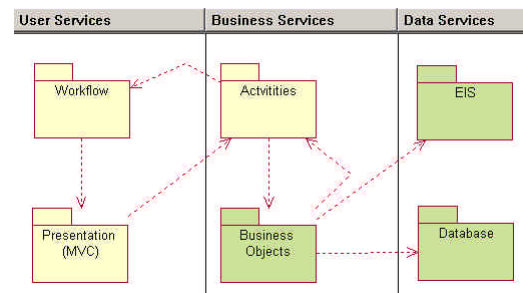
An entity bean represents an object view of business data stored in persistent storage or an existing application. The bean provides an object wrapper **around the data** to simplify the task of accessing and manipulating it.

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB



## What Is an EJB Entity Bean?

Scenario 2

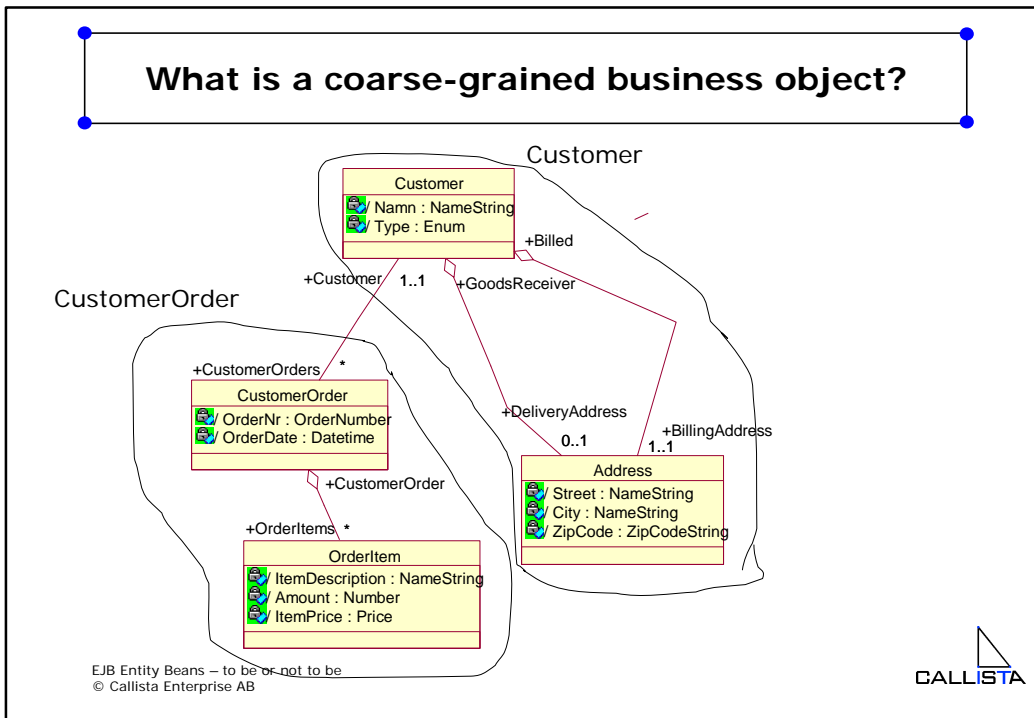
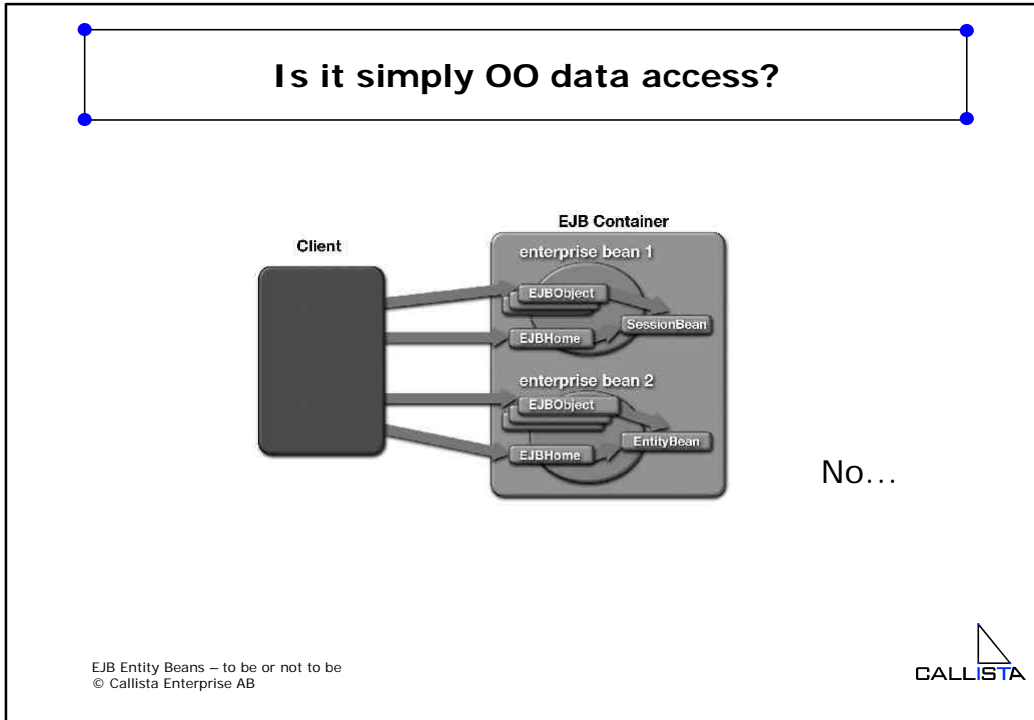


[Sun J2EE Blueprint, section 5.3.1]

...Therefore modelling every object representing a row in the database as an entity bean is not recommended. An entity bean is better suited to represent a **coarse-grained business object** that provides more complex behaviour than only get and set methods for its fields.

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB





## What do we need, to represent a C-G BOs?

- ⌘ Associations (for traceability)
  - ⌘ *Concept of independent and dependent entities (for traceability)*
- ⌘ Implement all BO-bound business rules in the BO (for architectural quality)
  - ⌘ *Provide both class- and instance behaviour*
  - ⌘ *Class Invariants / OCL or simply "Validation rules"*
- ⌘ Minimize interaction (for performance)
  - ⌘ *"Do-a-lot" type of methods*
  - ⌘ *Getting and assigning structural data (XML or serialized Beans)*
  - ⌘ *BO level Optimistic locking*
  - ⌘ *Isn't this what coarse-grained really means?*

## Issues with associations

- ⌘ Not defined by the spec
  - ⌘ *Proprietary solutions*
  - ⌘ *Not even portable within vendor releases (WebSphere)*
  - ⌘ *Poor exception reporting from integrity constraints*
- ⌘ Solution
  - ⌘ *Code by hand or home-brewn generators*
  - ⌘ *or don't do anything (client will manage foreign keys as any other data)*

## Issues with business rules

```
public interface CustomerOrder extends EJBObject {  
    public setOrderDate(Timestamp date);  
    public setReadyToShip(Timestamp date);  
    public addItem(String productId, int qty);  
}
```

### Constraints:

- a) Order must have at least one item
- b) ready-to-ship date must be future to order date

```
public class CustomerOrderBean extends EntityBean {  
    public void ejbStore() {  
        // I can implement the constraints  
        // but not throw an application exception  
    }  
}
```

## Issues with business rules

### ☞ Solutions

- ☞ Provide a `validate()` method
- ☞ or Use bulk-assignment (serialized bean or XML)

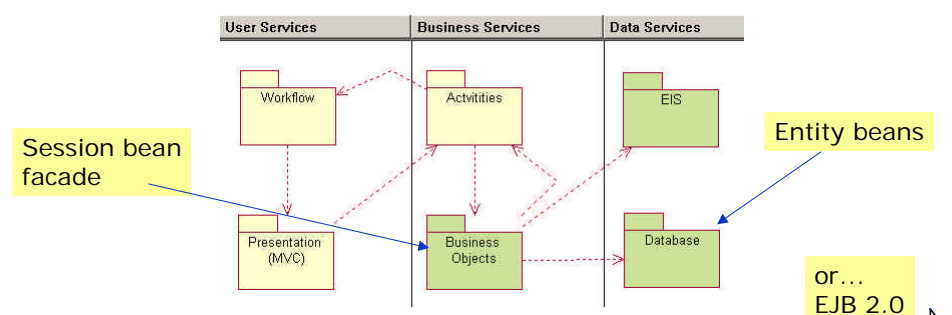
```
public interface CustomerOrder extends EJBObject {  
    public CustomerOrderData getOrderData();  
    public void setOrderData(CustomerOrderData orderData);  
}
```

OR... (not allowed in EJB 2.0)

```
public interface CustomerOrder extends EJBObject {  
    public setOrderDate(Timestamp date);  
    public setReadyToShip(Timestamp date);  
    public addItem(String productId, int qty);  
    public void validate() throws ....;  
}
```

## More issues with business rules

- ✗ Provide both class- and instance behaviour
  - ✗ Finders can only find object references, not data
  - ✗ Can't represent class methods, like `int Person::averageAge()`
- ✗ Solution



EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB

## Minimize interaction

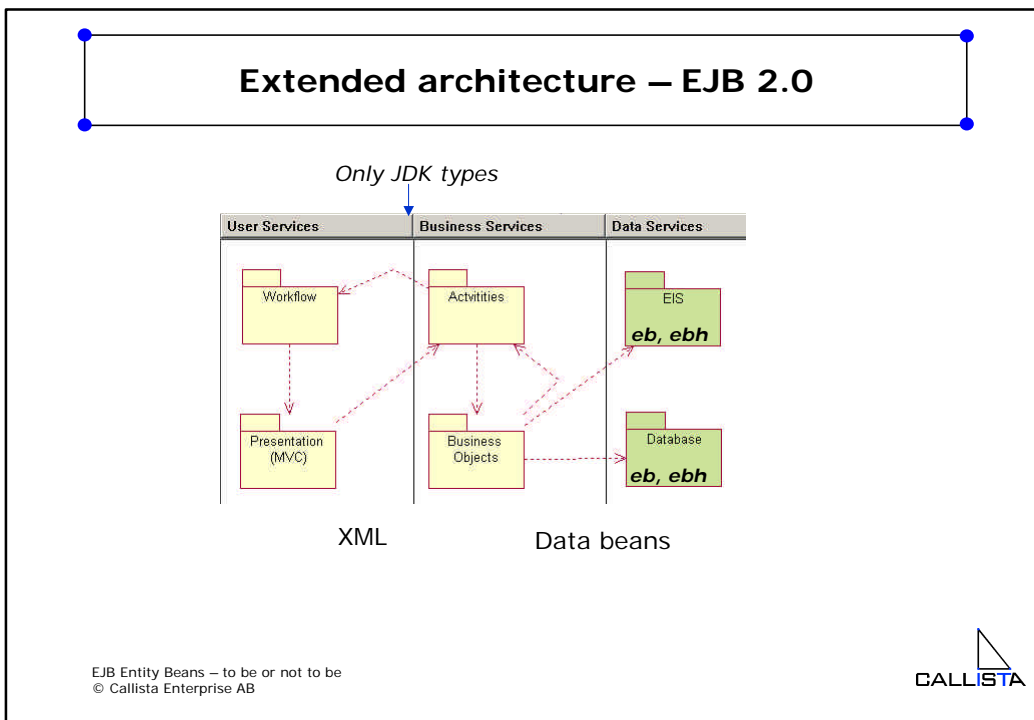
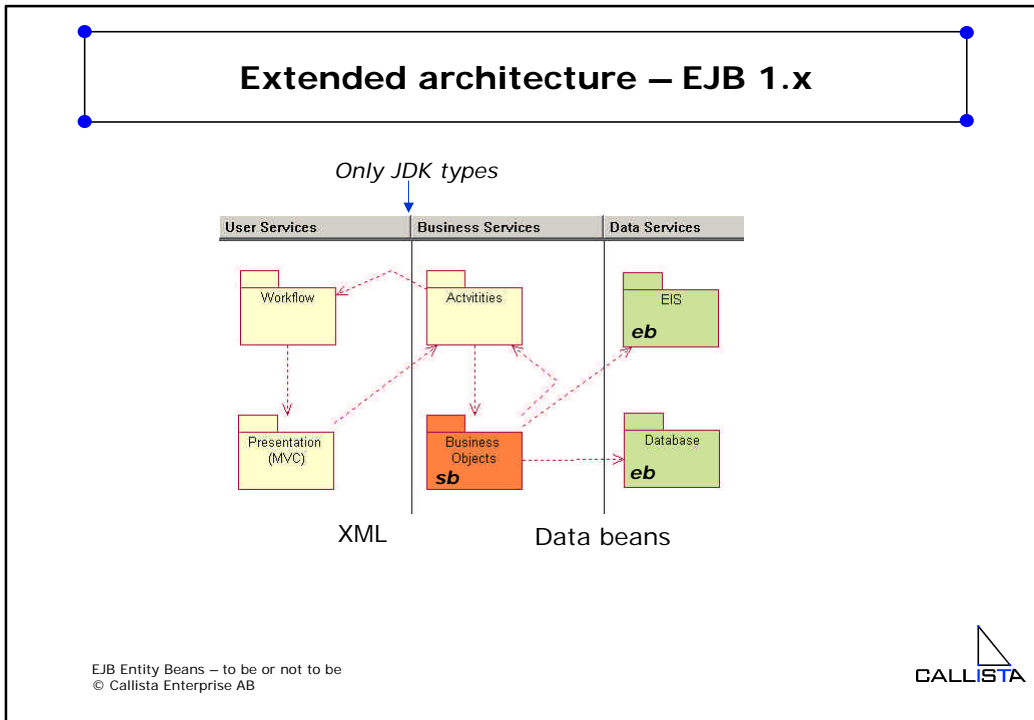
- ✗ Property Getters and setters are too fine grained
  - ✗ Entity beans are distributed objects
  - ✗ Potentially, every call is a save
- ✗ Solution
  - ✗ Use BO session bean facade → *or... EJB 2.0 (business methods in home interface)*
  - ✗ Provide bulk retrieval
  - ✗ Serialization is slow! Implement informal serialization interface.
  - ✗ Use XML encoding

```

public class CustomerOrderData implements Serialization, XmlEncoding {
    private void readObject(ObjectInputStream in) {
        this.fromXmlString(in.readUTF());
    }

    private void writeObject(ObjectOutputStream out) {
        out.writeUTF(this.toXmlString());
    }
}
    
```

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB



## Data finders

- ✍ EJB Entity Home today
  - ✍ Return IOR
  - ✍ Return Enumeration / Collection of IOR
- ✍ What we often need
  - ✍ Return data

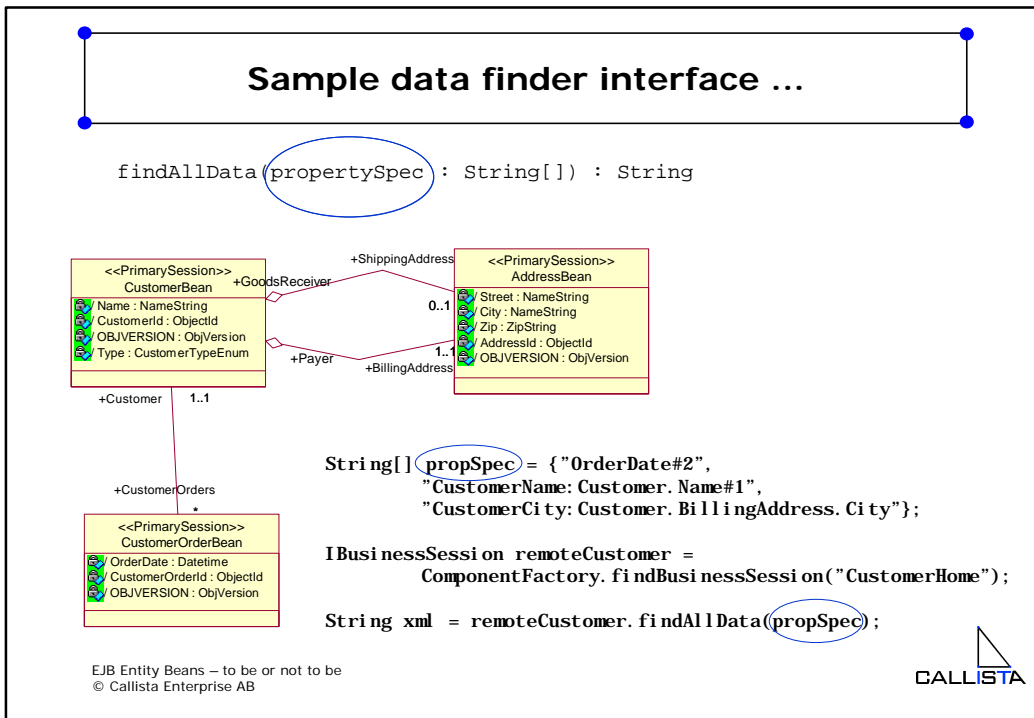
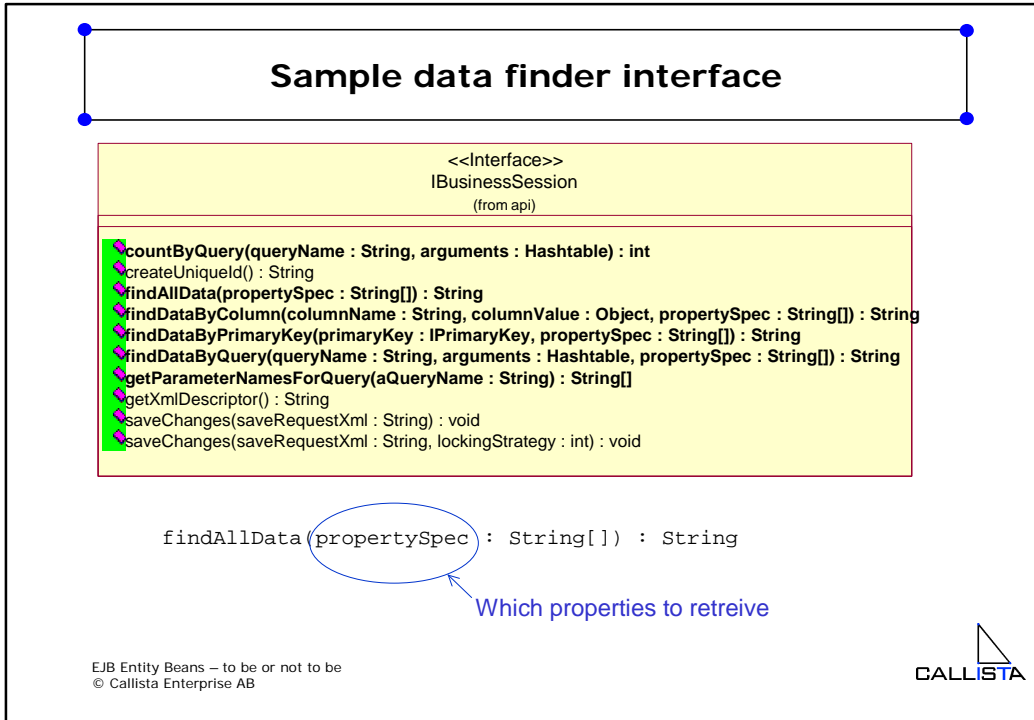
```
public interface CustomerOrder extends EJBHome {  
    Enumeration findToBeProcessed(int customerType);  
}
```

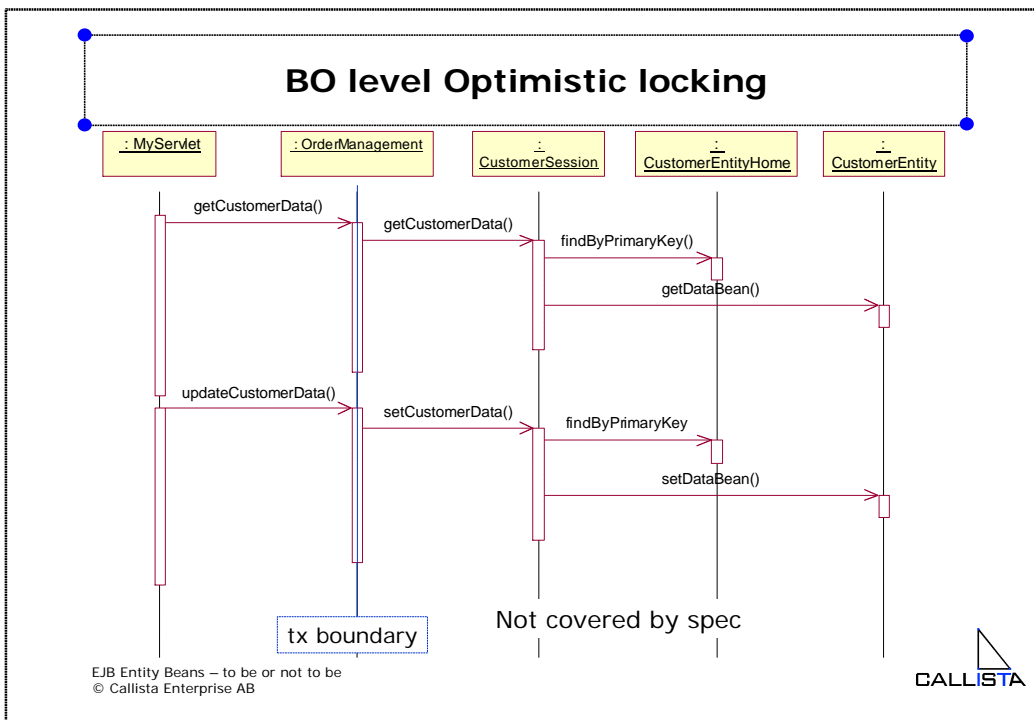
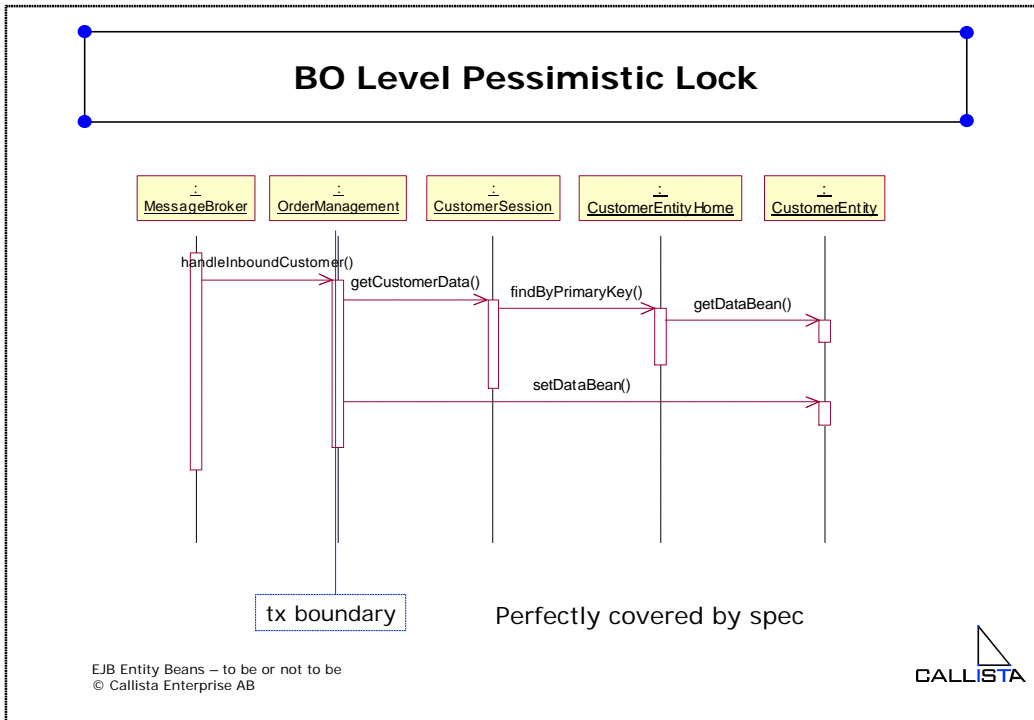
EJB 1.x always enumerate IORs

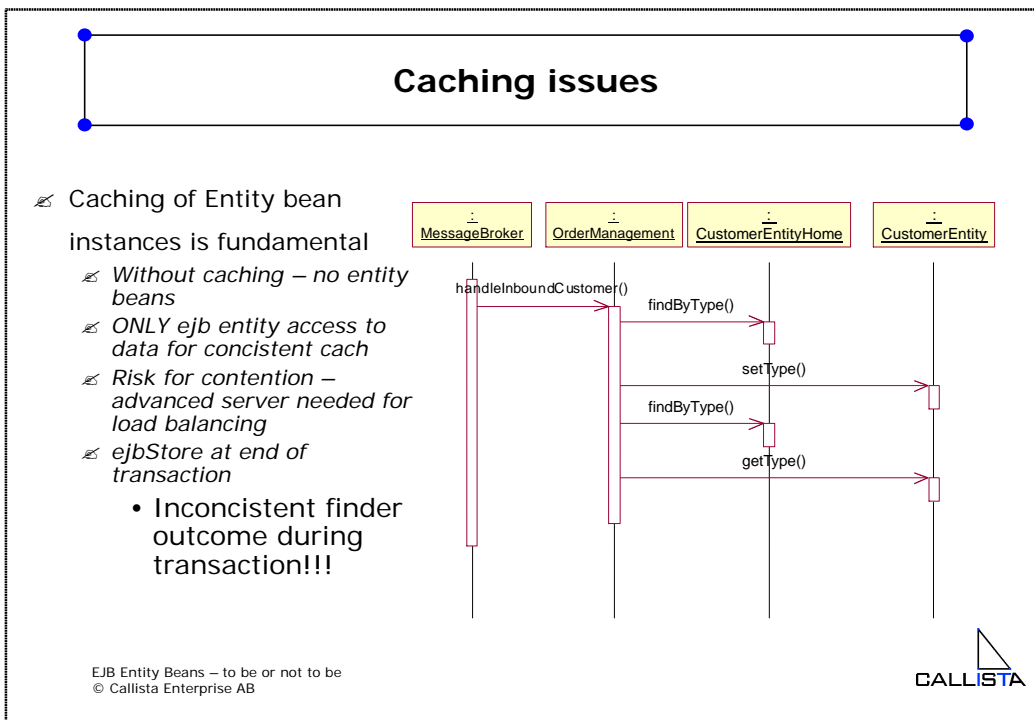
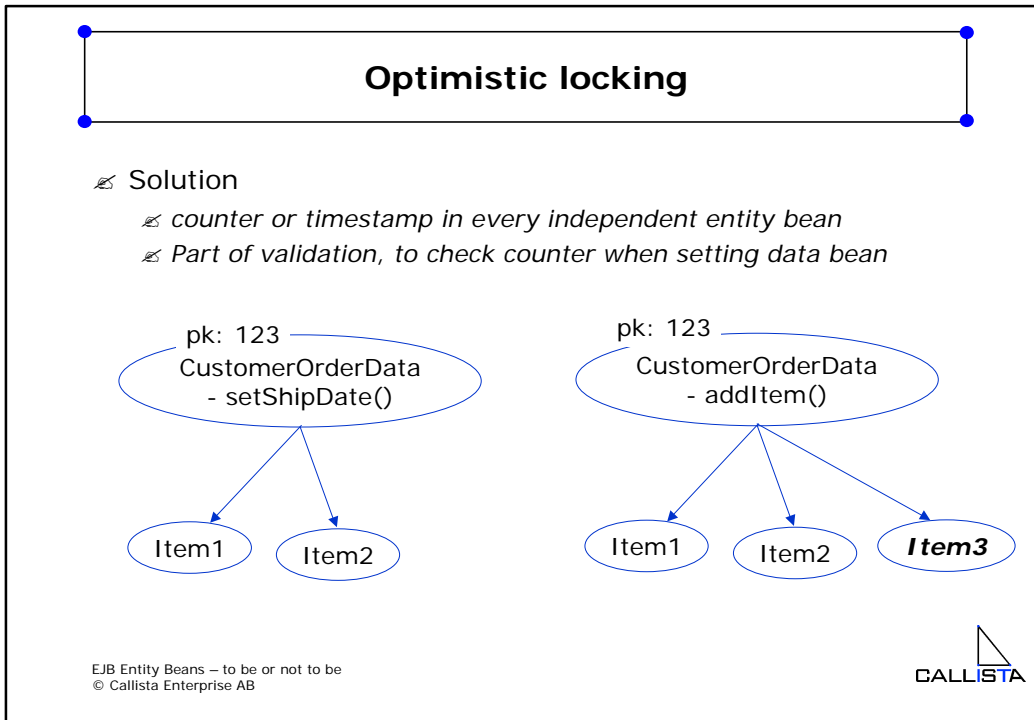
EJB 2.0 Home business method returns anything, e.g. data beans

## Data finders, continued

- ✍ Standardization for data bean (or XML) finders needed
  - ✍ Tremendous productivity gain
  - ✍ Reduced tool vendor lock-in
  - ✍ Primary focus of .NET ADO+ ✍
- ✍ "Workaround"
  - ✍ Large projects: Build custom framework
  - ✍ Session bean methods until 2.0, then as home business methods
  - ✍ Implement using JDBC
  - ✍ Restrict usage to none-transactional requests, if Entity Beans are used for DML, due to cach/isolation complexity







## Cach rules when using entity beans

- ✍ Never code JDBC-DML
- ✍ Close down EJB server when running batches
- ✍ Train programmers to understand the impact of a cach during transaction.
  - ✍ *Design/code reviews*

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB



## Entity bean checklist – part 1(3)

Subject	Architects Task	Risk (if not considered)	EJB 2.0 Provides...
Use Model	Select ONE use model – Coarse grained business objects or pure persistence (recommended).	Architecture, Time plan	Focus on pure persistence
Business Rules	Decide how to implement object constraints. Use some kind of setAll() pattern, using data beans or long parameter list to be able to validate before assigning. If use model is pure persistence, put constraints in session bean.	Architecture, Reuse, unstable interfaces	An additional restriction: app exception and entity state.
Associations	Define style guides, generators	Vendor lock-in	Portable solution

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB



## Part 2 (3)

Subject	Architects Task	Risk (if not considered)	EJB 2.0 Provides...
Object model	Use session beans as business object model. Implement using entity beans.	Architecture, Reuse	Entity Home business methods
Optimistic locking	Select a strategy, based on a per entity basis.	Data quality, User satisfaction	Recommendation to provide mechanism
Data Finders	Define framework or generators to handle 80-90% of data retrieval tasks. Use JDBC.	Time plan, Performance	

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB

## Part 3 (3)

Subject	Architects Task	Risk (if not considered)	EJB 2.0 Provides...
Caching	Create awareness about impact on finders, since cach and db not in synch during transaction	Invalid data	EJB QL implementers may make this transparent
Data beans serialization	Decide if optimisation needed (custom serialization)	Performance	

EJB Entity Beans – to be or not to be  
 © Callista Enterprise AB

## Summary

- ✂ We need EJB 2.0!
- ✂ In the meanwhile, think twice before heading entity beans
  - ✂ *Type of applications? Fighting them, or using them?*
  - ✂ *B2B – rather consider pure session bean or mixed architecture*
  - ✂ *C2B – Entity beans does the job! No opt.lock, simple bus.logic, cach needed, few updates - many reads*
  - ✂ *Enterprise Applications – stick to session beans*

johan.eltes@callista.se

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB



## Links

- ✂ Suns J2EE Design Blueprint
  - ✂ <http://java.sun.com/j2ee/download.html#blueprints>
- ✂ IBM redbook "Design and Implement Servlets/JSPs/EJBs for IBM WebSphere"
  - ✂ <http://www.redbooks.ibm.com/redpieces/pdfs/sg245754.pdf>
- ✂ EJB 2.0 Draft
  - ✂ <http://java.sun.com/products/ejb/2.0.html>
- ✂ EJB 1.1 Spec
  - ✂ <http://java.sun.com/products/ejb/docs.html>
- ✂ Shareware Session Bean Business Object Framework
  - ✂ <http://www.callista.se/Enterprise/enstart.htm>
- ✂ This presentation (IE4+)
  - ✂ <http://www.callista.se/euroforum/presentation/>

EJB Entity Beans – to be or not to be  
© Callista Enterprise AB

