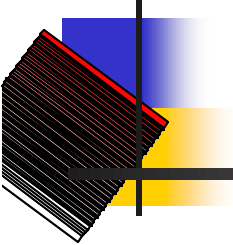
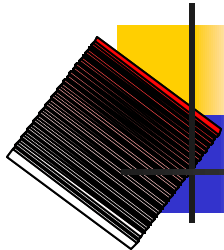


# The Beginner's Guide To EJB's with JBoss: Quick and Free



---

Cedrick W. Johnson  
Catylist, Inc.  
[javadude@cedrick.net](mailto:javadude@cedrick.net)



# Agenda

---

- ✍ Fuzzy Wuzzy Introductions
- ✍ EJB Lingo
- ✍ Entity Bean Defined
- ✍ Types of Entity Beans:
  - ✍ Fine
  - ✍ Coarse
  - ✍ O/E, etc....
- ✍ Differences between regular client DB access and Entity access
- ✍ CMP & BMP Side by side
- ✍ Entity Bean Intricacies
- ✍ Performance Increases
- ✍ Review
- ✍ Resources
- ✍ Demonstration



# About The Presenter

- ✍ Cedrick Johnson
- ✍ Cavenger Systems  
CTO, co-founder  
(old co.)
- ✍ Technology  
Evangelist and  
100% Geek





# Some EJB “Lingo”

---

## ✍ Session Beans

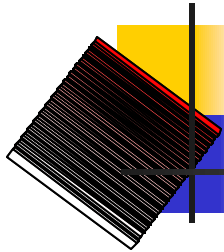
- ✍ A business process (verb)
- ✍ Stateful and Stateless

## ✍ Value Objects

## ✍ Local Interfaces (NEW IN 2.0!)

## ✍ Entity Beans

- ✍ A data object (noun)
  - ✍ CMP
  - ✍ BMP
  - ✍ Finder Methods
- ## ✍ EJB-QL (NEW IN 2.0 TOO!!!!)



# Entity Bean Defined

---

- ✍ A bean that is used to represent a ROW of data within a database
- ✍ BMP and CMP Entity Beans
  - ✍ BMP = harder, but allows for more control over your SQL code
  - ✍ CMP = easier, but can sometimes be TOO easy
- ✍ A REUSABLE “component” of your software application that allows for access to a data resource



# Types of Entity Beans: Fine-Grained

---

- ✍ Usually a client accesses these beans via each of the bean's accessor methods.
- ✍ Client calls each of the get/set methods for reading/manipulating data
- ✍ Fine for small applications or for beginners



# Types of Entity Beans: Coarse Grained

---

- ✍ A client accesses these beans by using a Value Object, an object that contains the fields that represent the EJB's accessor methods
- ✍ Client makes ONE call, gets the VO, then makes changes to the VO and resubmits
- ✍ More suitable for performance-critical applications (as we will see later)



# Comparison: Entity Beans vs. Traditional DB Access Methods

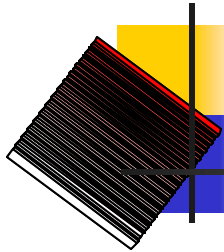
## Traditional

- ✍ Each client eats up a DB connection
- ✍ Business logic resides on either client or DB
- ✍ Single point of failure, little or no reusable components

## Entity

- ✍ Connections to the DB are handled by the container
- ✍ Business logic can now reside in an server-side EJB = reusability
- ✍ Clustering provides redundancy
- ✍ Somewhat complicated to begin learning/implementing properly





# CMP and BMP Differences

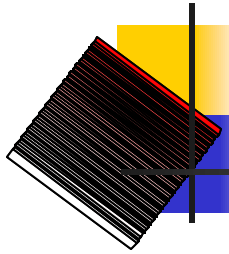
---

## Container

- ✍ No client SQL Code (faster to develop)
- ✍ Plug and play with database
- ✍ No DB access code to deal with

## Bean

- ✍ More control over queries
- ✍ Semi-Plug and play with DB
- ✍ Less new learning (reuse and extend old JDBC code/components)



## How is a EB mapped to a row?

---

- ✍ Container obtains all the rows in a database, then creates a Remote interface “handle” to that row in the database.
- ✍ 100 Rows, 100 “handles”
- ✍ 1000 Rows, 1000 “handles”



# How do you add new rows with Entity Beans?

- ✍ Clients can call the `create()` method on the Entity Bean's REMOTE interface... This tells the **container** to invoke `ejbCreate()` within your EJB to perform the operations needed.

```
MyEntityRemote.create(new Integer(1), "Hello");
```



Container

```
MyEntityEJBImpl.ejbCreate(new Integer(1), "Hello");
```



# How do you modify existing rows with Entity Beans?

- ✎ Usually you need to find the data, then set the attributes on the retrieved row. Container manages the update
- ✎ For better performance, use a Value Object

```
MyEJB.ejb = home.findByPrimaryKey(1);  
.ejb.setName("Alfred");
```



# Modifying Existing Rows with a Value Object

---

```
public PersonVO getPerson() {
    PersonVO lPerson = null;
    MyEJBRemote ejb = null;

    try {
        ejb = ejbHome.findByPrimaryKey(1);
        lPerson = ejb.getPersonVO();
    } catch .....

    return lPerson;
}
```

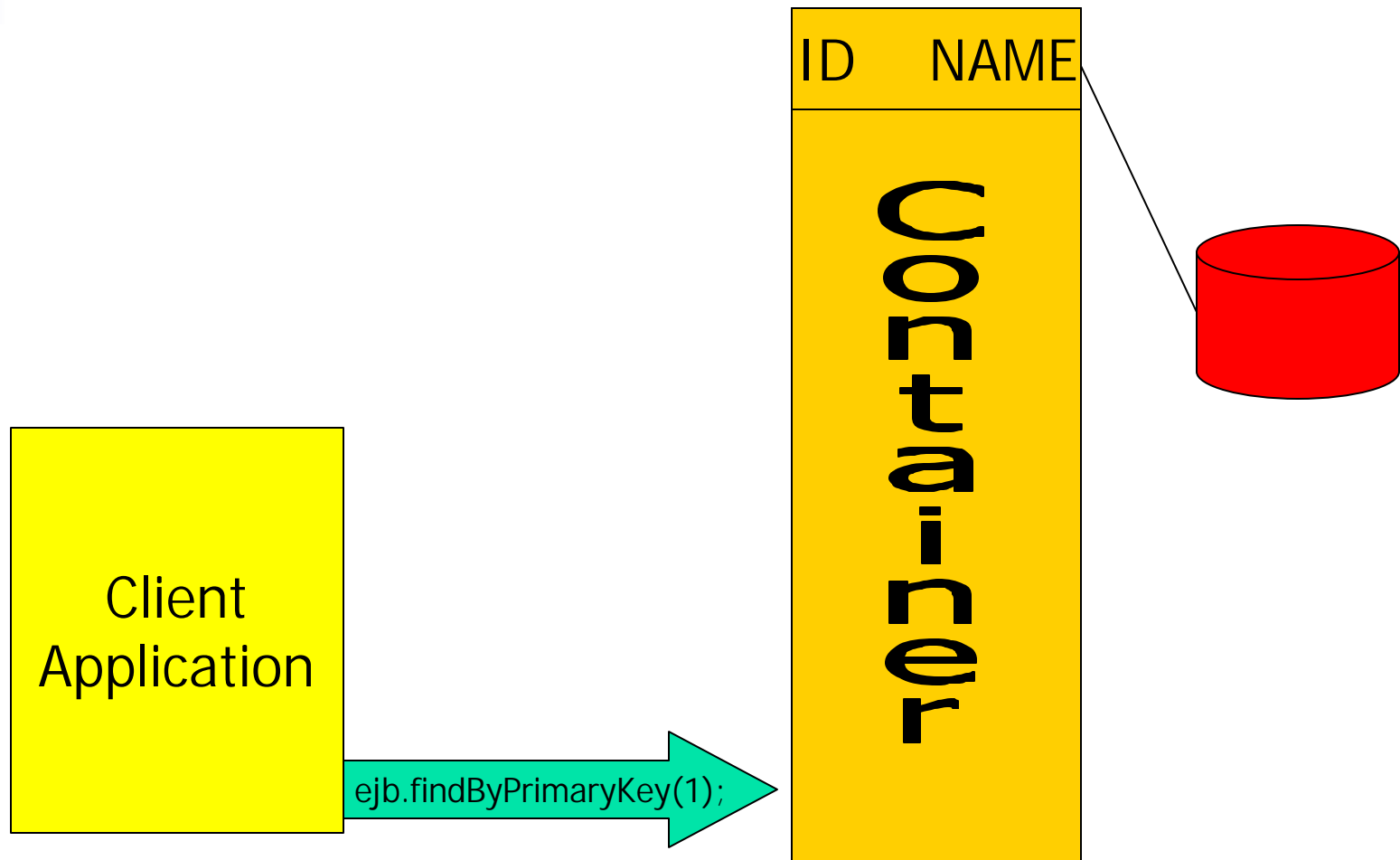


Are We Awake????

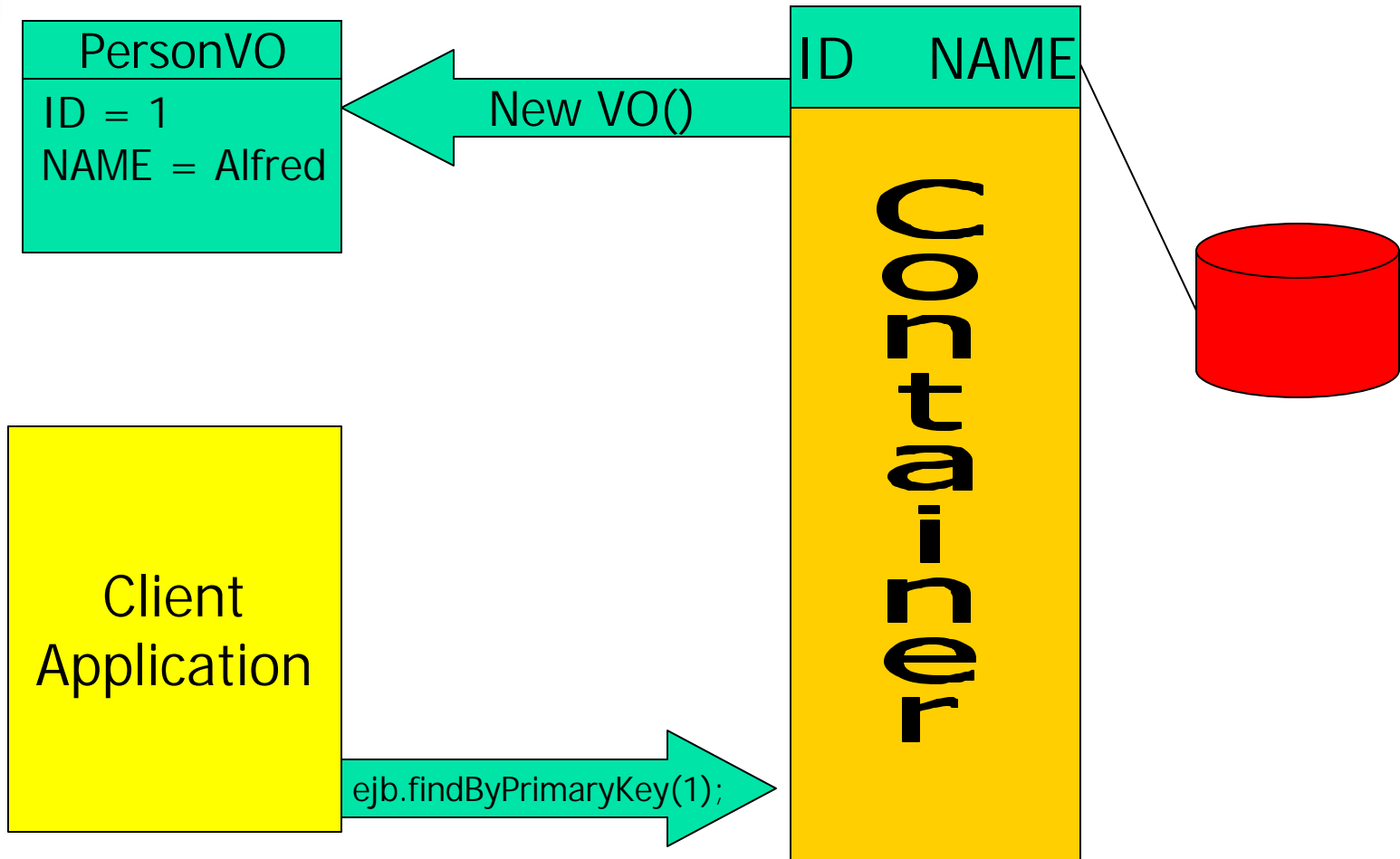
---



# Modifying Existing Rows with a Value Object

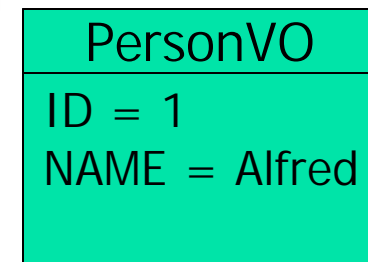


# Modifying Existing Rows with a Value Object

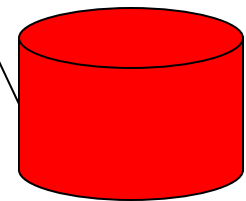




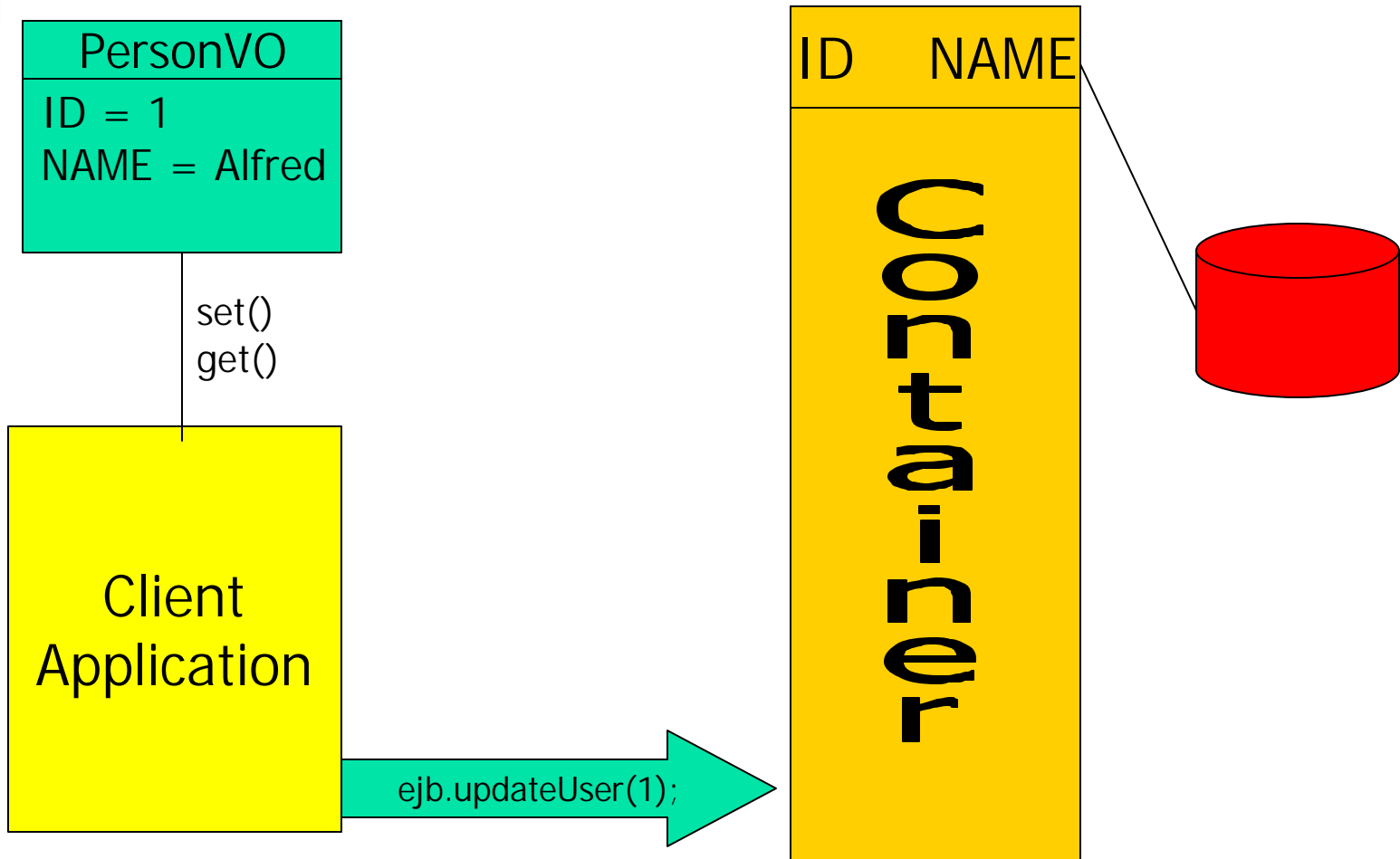
# Modifying Existing Rows with a Value Object



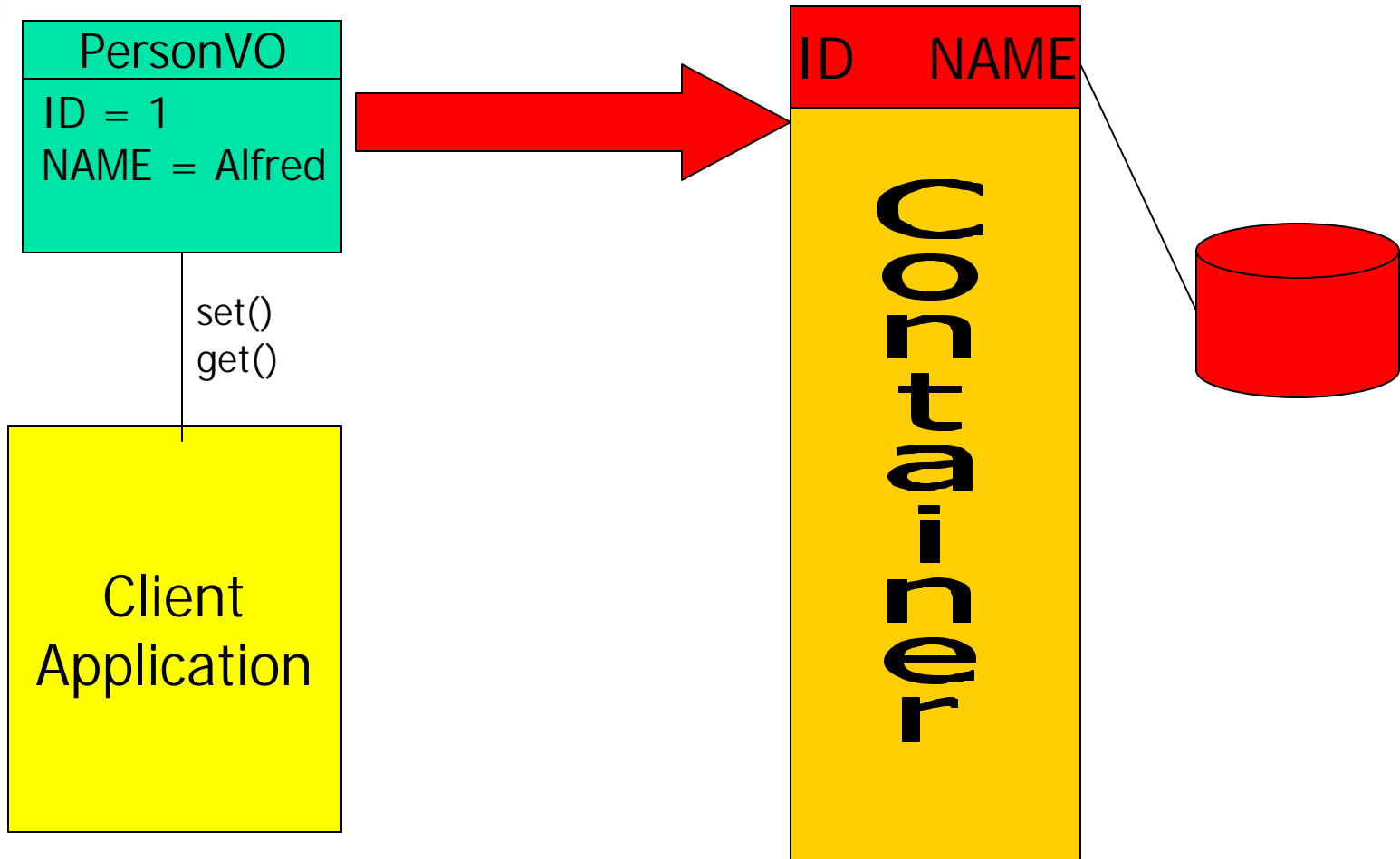
set()  
get()



# Modifying Existing Rows with a Value Object



# Modifying Existing Rows with a Value Object

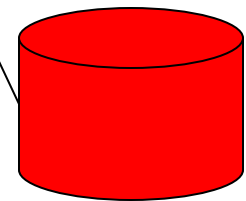


# Modifying Existing Rows with a Value Object

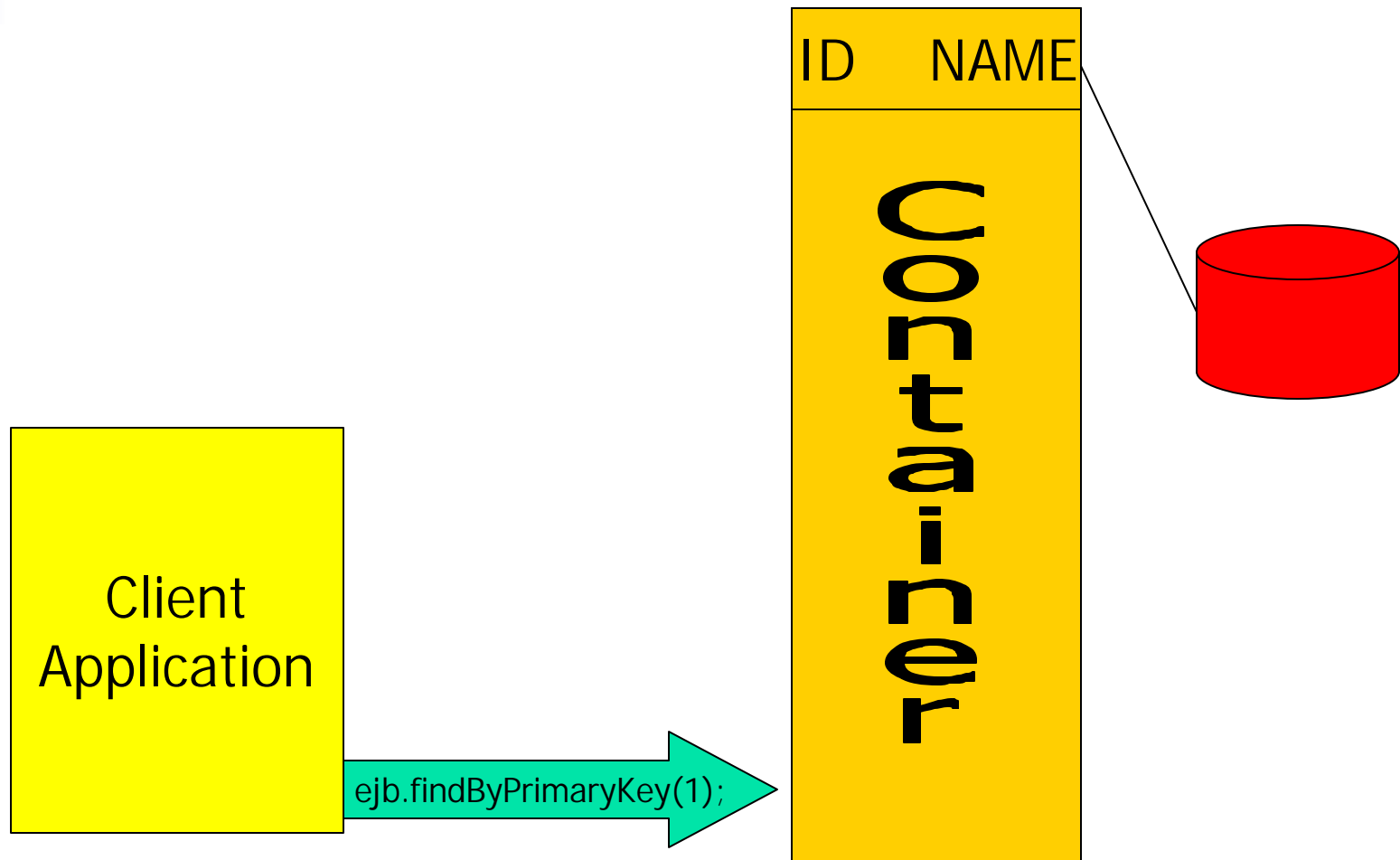
Values are now persisted in the database



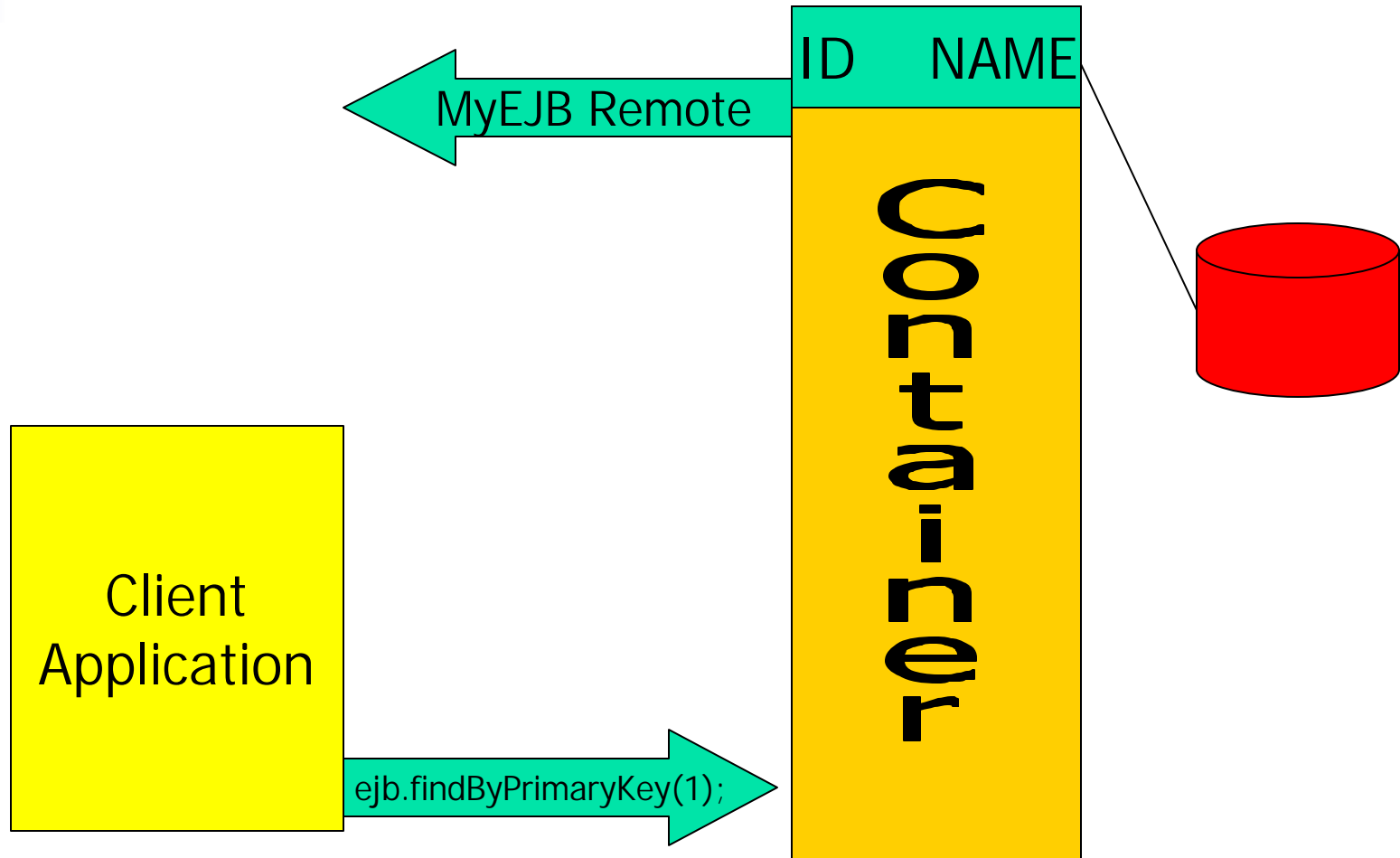
ID	NAME
103-04500	UCC+000



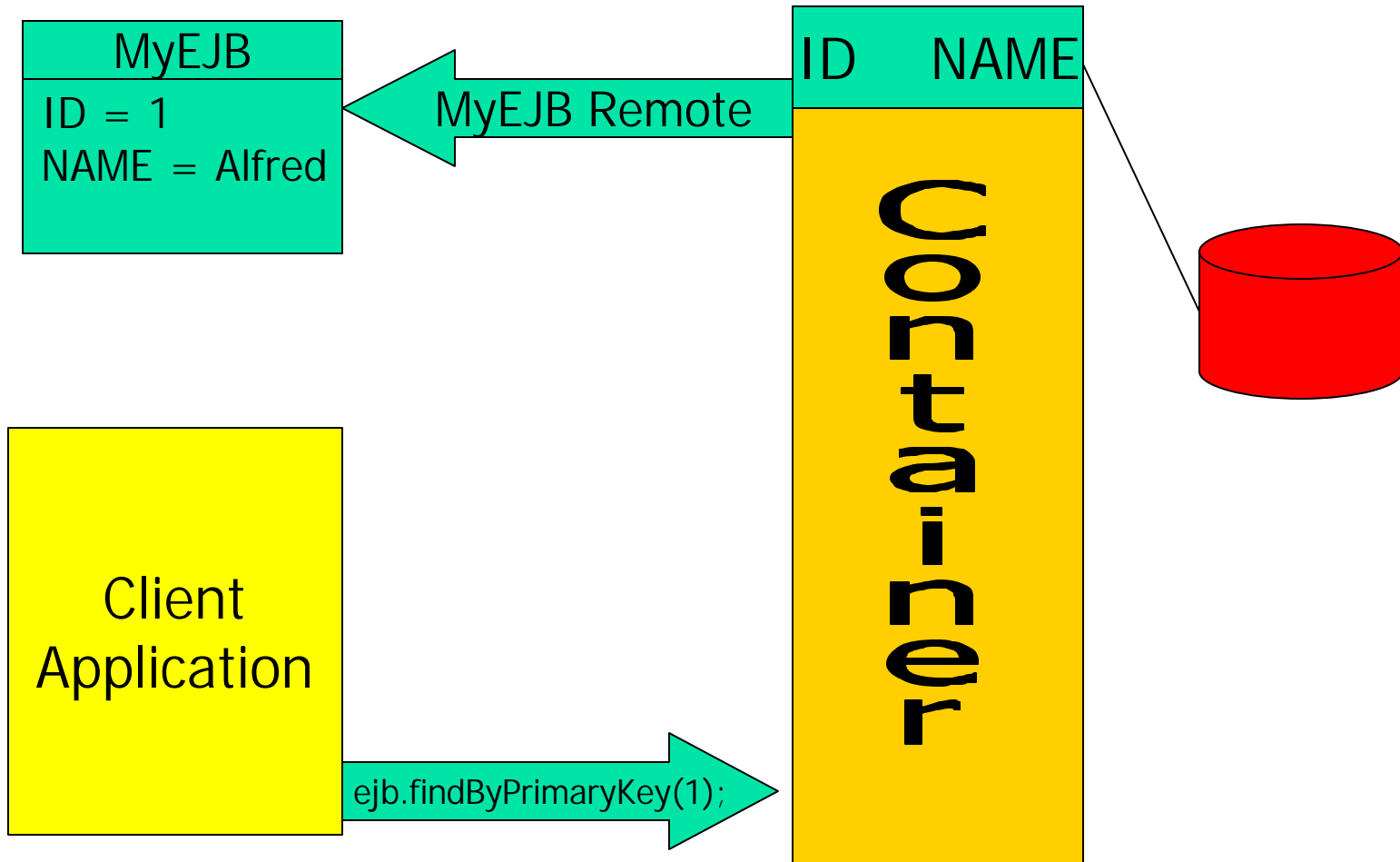
# Modifying Existing Rows with a Remote Interface



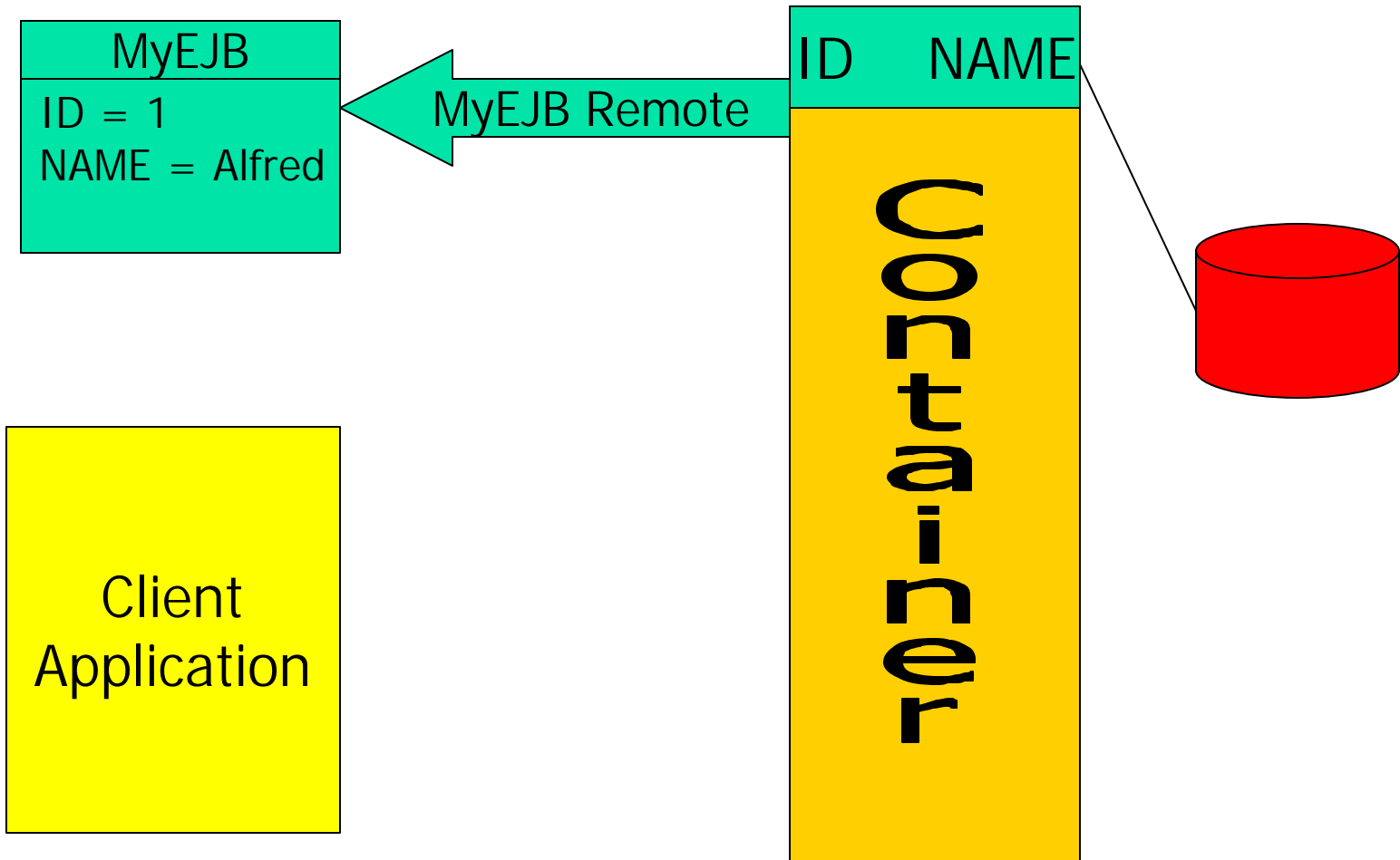
# Modifying Existing Rows with a Remote Interface



# Modifying Existing Rows with a Remote Interface

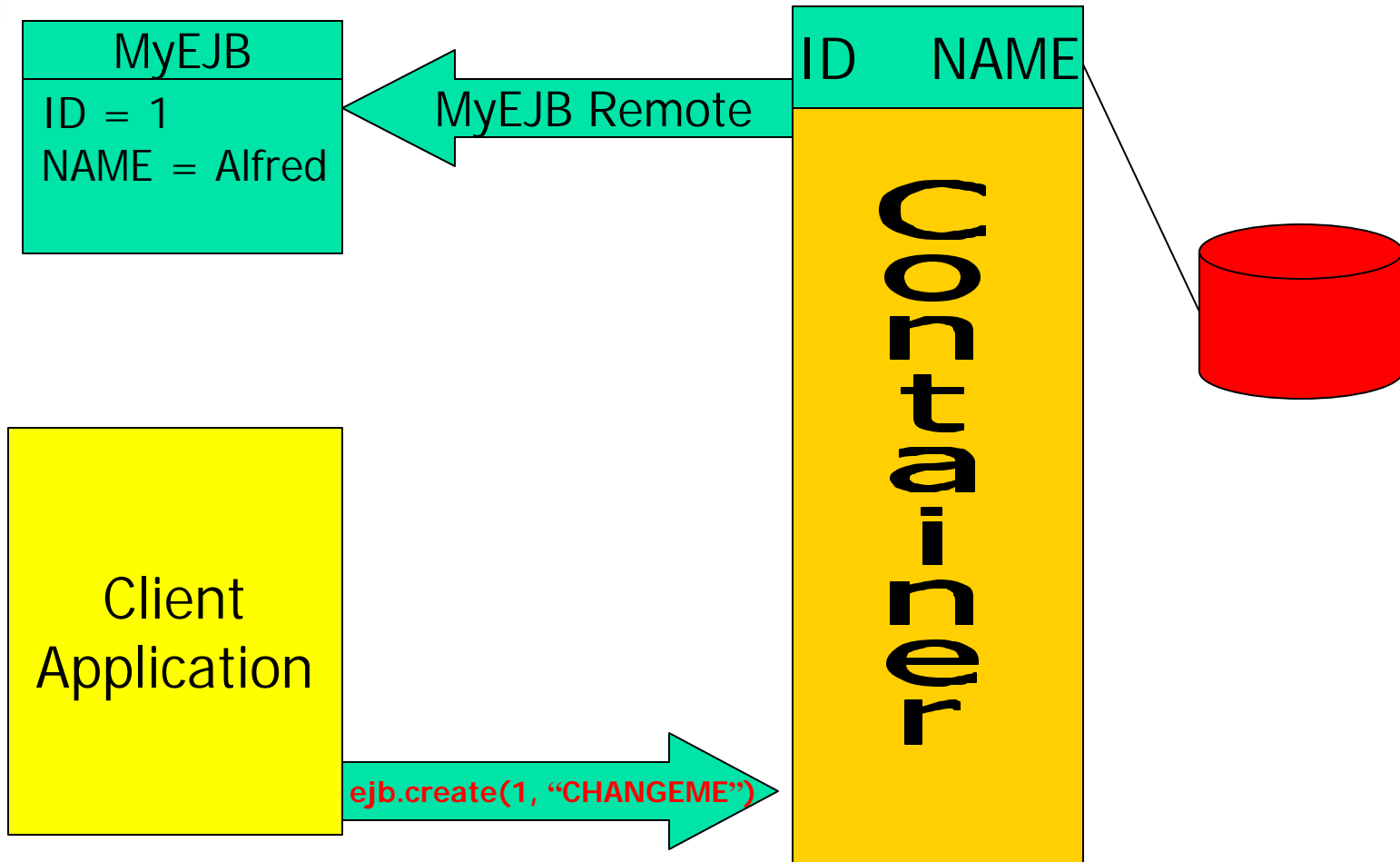


# Modifying Existing Rows with a Remote Interface

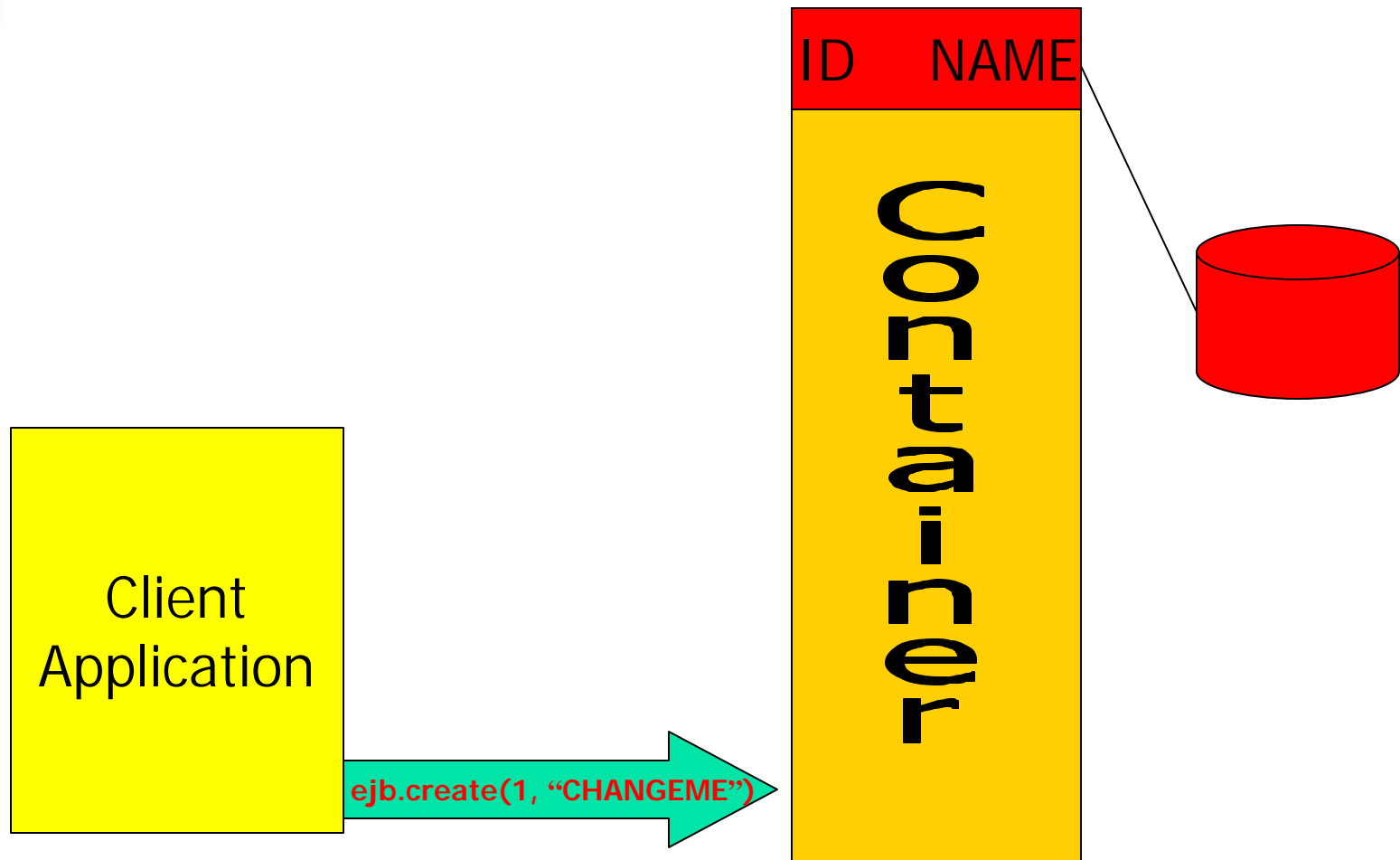




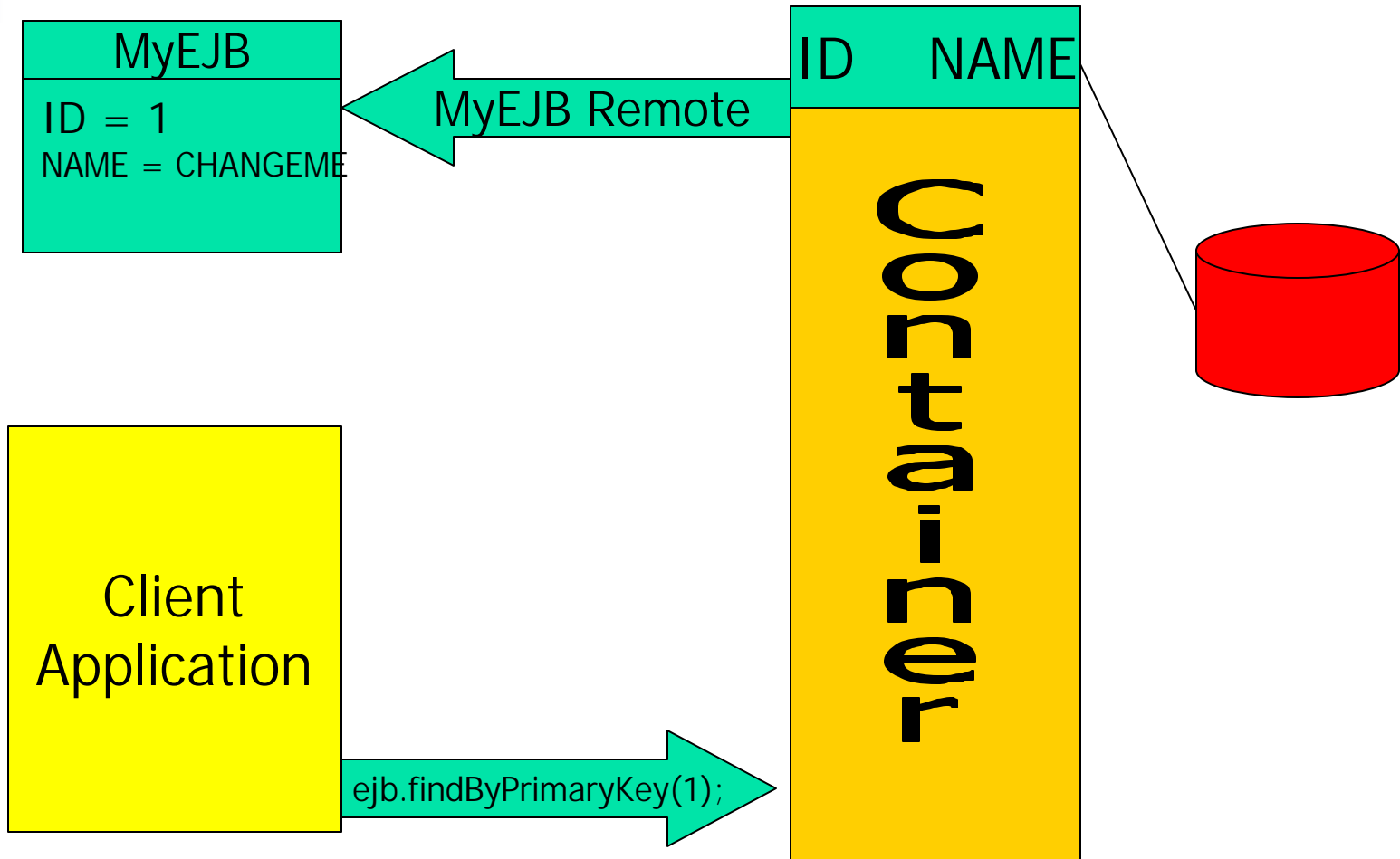
# Modifying Existing Rows with a Remote Interface

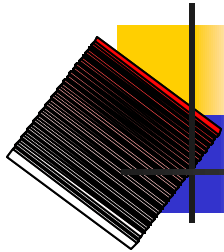


# Modifying Existing Rows with a Remote Interface



# Modifying Existing Rows with a Remote Interface

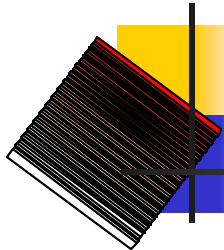




# EJB Performance

---

- ✍ YES! EJB's are slow compared to other methods of access
- ✍ BUT: Optimization is key to achieving maximum performance and scalability
- ✍ If you use EJB's, you must architect systems from the ground up with proper design decisions, else it WILL be slow.
- ✍ Use Value Objects
- ✍ Use Local Interfaces
- ✍ Lazy Instantiation
- ✍ Session or Message Driven Bean façades, etc.

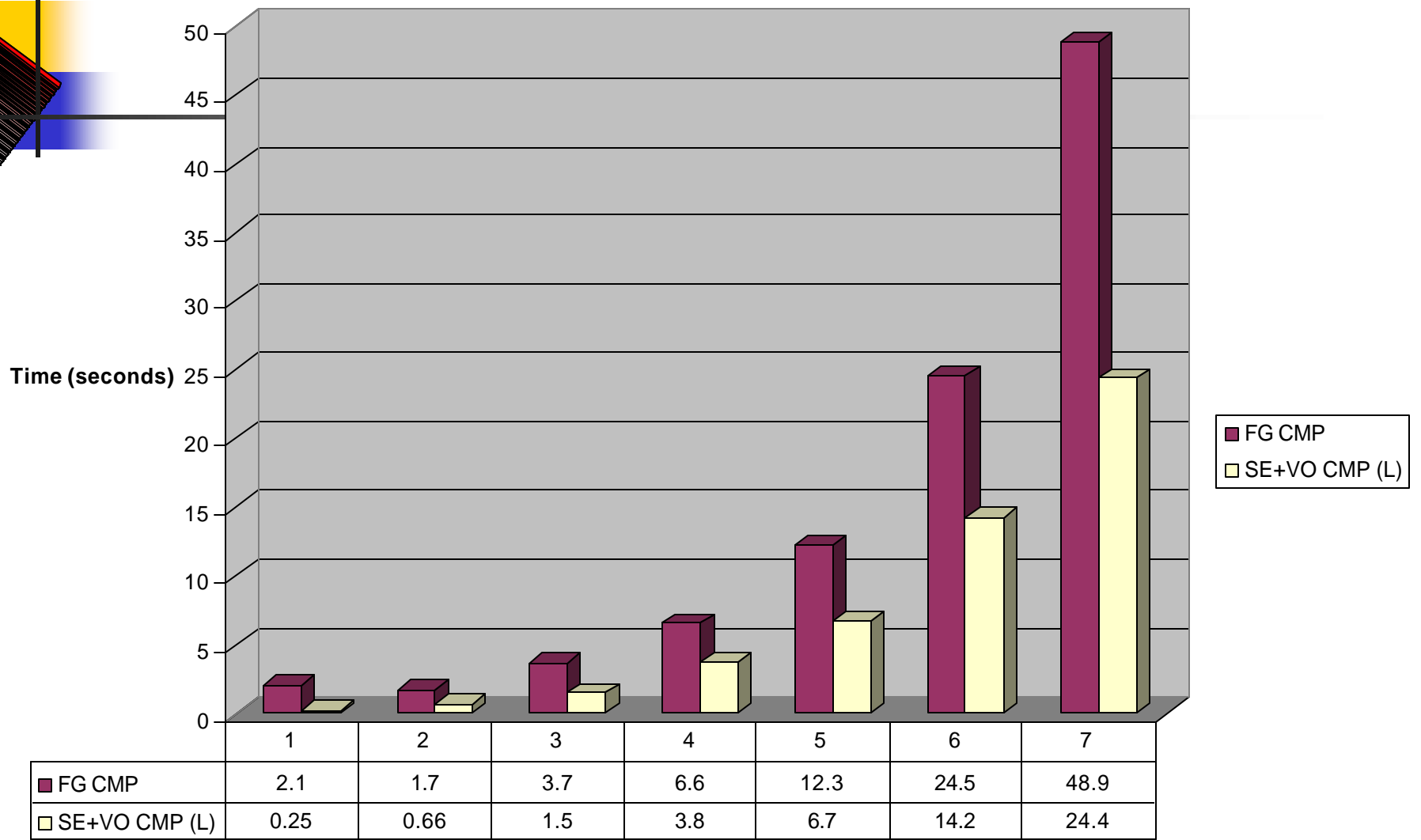
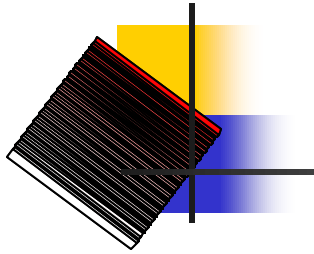


# EJB Performance

---

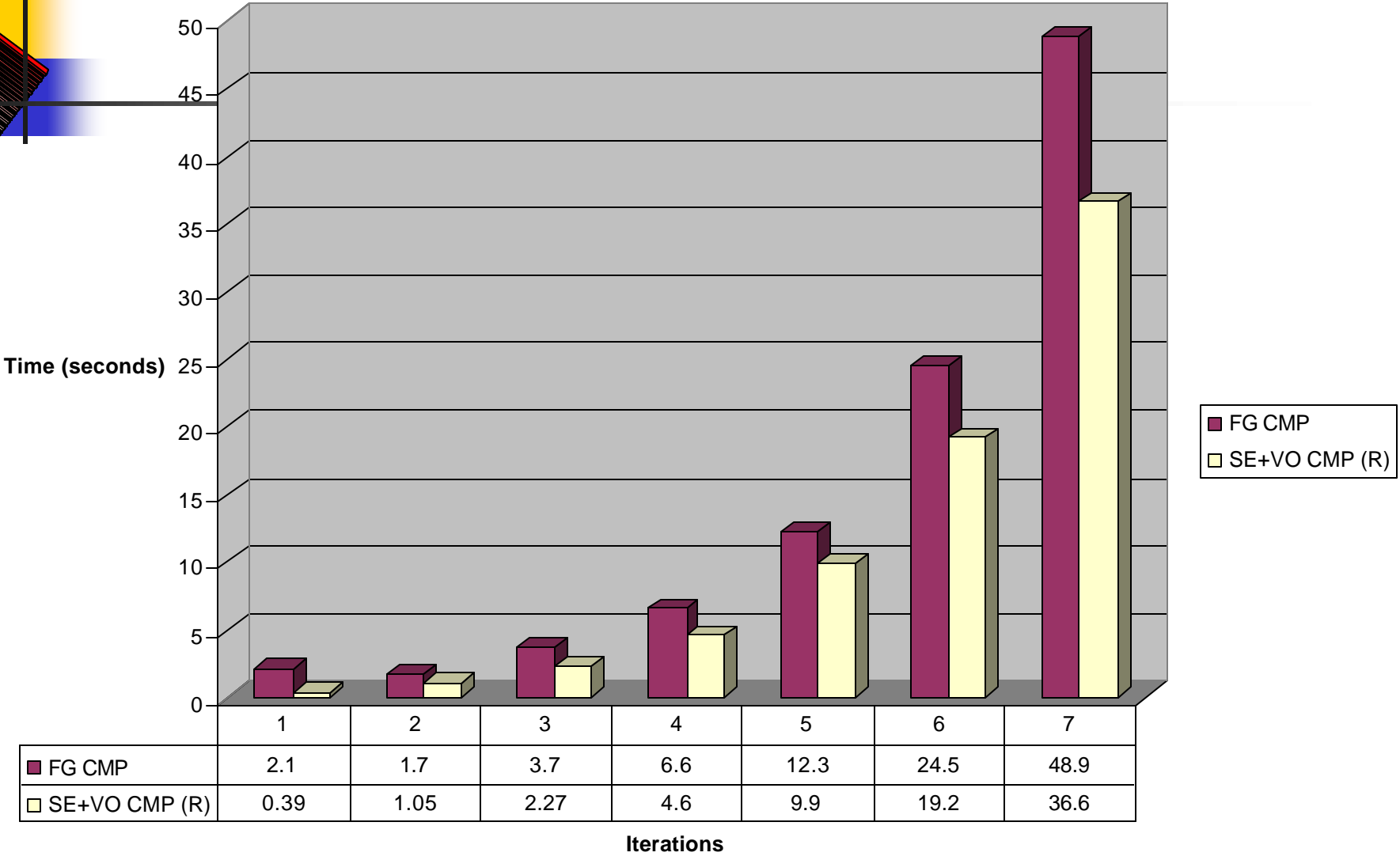
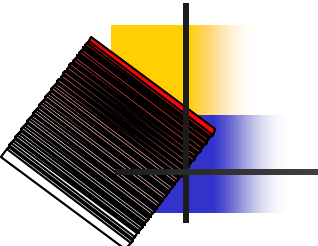
- ✍ Serialization
  - ✍ Use the **transient** keyword for fields not being serialized
  - ✍ Smaller transport (value) objects that transmit only data you need
- ✍ “Coarse Grained” Network calls
- ✍ Garbage Collection
  - ✍ Null out references to objects that are no longer needed
- ✍ Cached Row Sets and a Updater EJB
  - ✍ Client gets a row set, disconnects (Disconnected DS) performs operations, then “publishes” that RS to a listener EJB which performs the necessary DB updates/checking

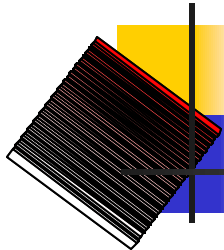
**CMP Optimized SE+VO (Local Interface) vs CMP FG Entity**



**Iterations**

**CMP Optimized Remote Interface SE+VO vs. FG Entity**



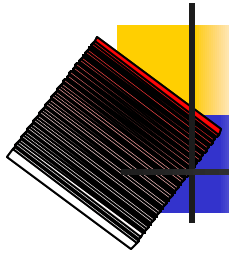


# Review

---

- ✍ EJB is NOT all there is to J2EE
  - ✍ Evaluate project needs
  - ✍ Not needed for small applications, usually
- ✍ Investigate and learn!!
- ✍ Why do we use EJB?
- ✍ What is an entity bean
- ✍ Entity Bean Types
- ✍ Differences between DAO and EJB data access
- ✍ CMP & BMP: Side by Side
- ✍ Performance Issues





# Resources To Get You Started

---

- ✍ JBuilder 5/6 (NOW WITH UML!!!)
  - ✍ <http://www.borland.com/jbuilder>
- ✍ Sybase EAServer 4.1 Developer Edition
  - ✍ <http://www.sybase.com/products/easerver>
- ✍ JBoss Open Source App Server
  - ✍ <http://www.jboss.org>
- ✍ Orion App Server
- ✍ BEA Weblogic App Server
  - ✍ <http://www.bea.com>
- ✍ Eclipse IDE
  - ✍ [www.eclipse.org](http://www.eclipse.org)
  
- ✍ Your local EJB nut
- ✍ The presenter



# Resources To Get You Started (Continued)

---

- ✍ The ServerSide
  - ✍ <http://www.theserverside.com>
- ✍ The Middleware Company
  - ✍ <http://www.middlewarecompany.com>
- ✍ JGuru Forums
  - ✍ <http://www.jguru.com>
- ✍ AspectJ
  - ✍ <http://www.aspectj.org>
- ✍ EJB Performance Measurements – November 2001 CJUG  
Presenter- Maciej Zawadski
  - ✍ <http://www.urbancode.com>



# Demonstration

---

JBoss 3.0 Application Server  
JBoss Druid - CMP Generator  
Eclipse IDE  
No cards up my sleeve