



JavaOneSM
Sun's 2000 Worldwide Java Developer Conference*



JavaOneSM
Sun's 2000 Worldwide Java Developer Conference

Enterprise JavaBeansTM 2.0 Technology: New Features

Linda DeMichiel
Ümit Yalçinalp
EJB Architects
Sun Microsystems, Inc.

Agenda

- Goals for EJB™ 2.0 Architecture
- New Architecture Features
- Status and Roadmap
- Extended Q&A at the BOF Tonight



What Is the Enterprise JavaBeans™ Architecture?

- A component architecture for the development and deployment of distributed business applications
- Goals
 - Simplify development and deployment of business applications
 - Separate business logic from system code
 - Portable components (Write Once, Run Anywhere™)
 - Allow multi-vendor interoperability



Goals for EJB™ 2.0 Architecture

- Java Community ProcessSM initiative JSR-000019
 - Integration with the Java™ Message Service API
 - Improved Support for Entity Beans with Container Managed Persistence
 - Support for Container Managed Relationships
 - Query Language for Finder Methods
 - Home Methods for Entity Beans
 - Network Interoperability



Expert Group for EJB™ 2.0 Specification

Allaire

Art Technology Group

BEA

Bluestone

Forte

Fujitsu

Gemstone

IBM

InLine Software

Inprise

IONA

iPlanet

Luna Information Systems

Oracle

Persistence

Progress Software

Secant

Siemens

Silverstream

Software AG

Sun Microsystems

Sybase

The Object People

Tibco

Vitria



Integration with Java™ Message Service API (JMS)

- Provide loose coupling among heterogeneous systems
 - Interoperate with legacy systems that use messaging for integration
 - Interoperate with B2B systems not based on the J2EE™ platform
- Provide asynchronous communication
- Provide support for disconnected use of enterprise beans



Message-driven Beans

- New enterprise bean type
 - Asynchronous
 - Activated upon message arrival
 - Stateless
 - No home or remote interface
 - Bean implements **`javax.jms.MessageListener`** interface
 - **`onMessage`** method contains business logic
 - Configured as listener for queue/topic

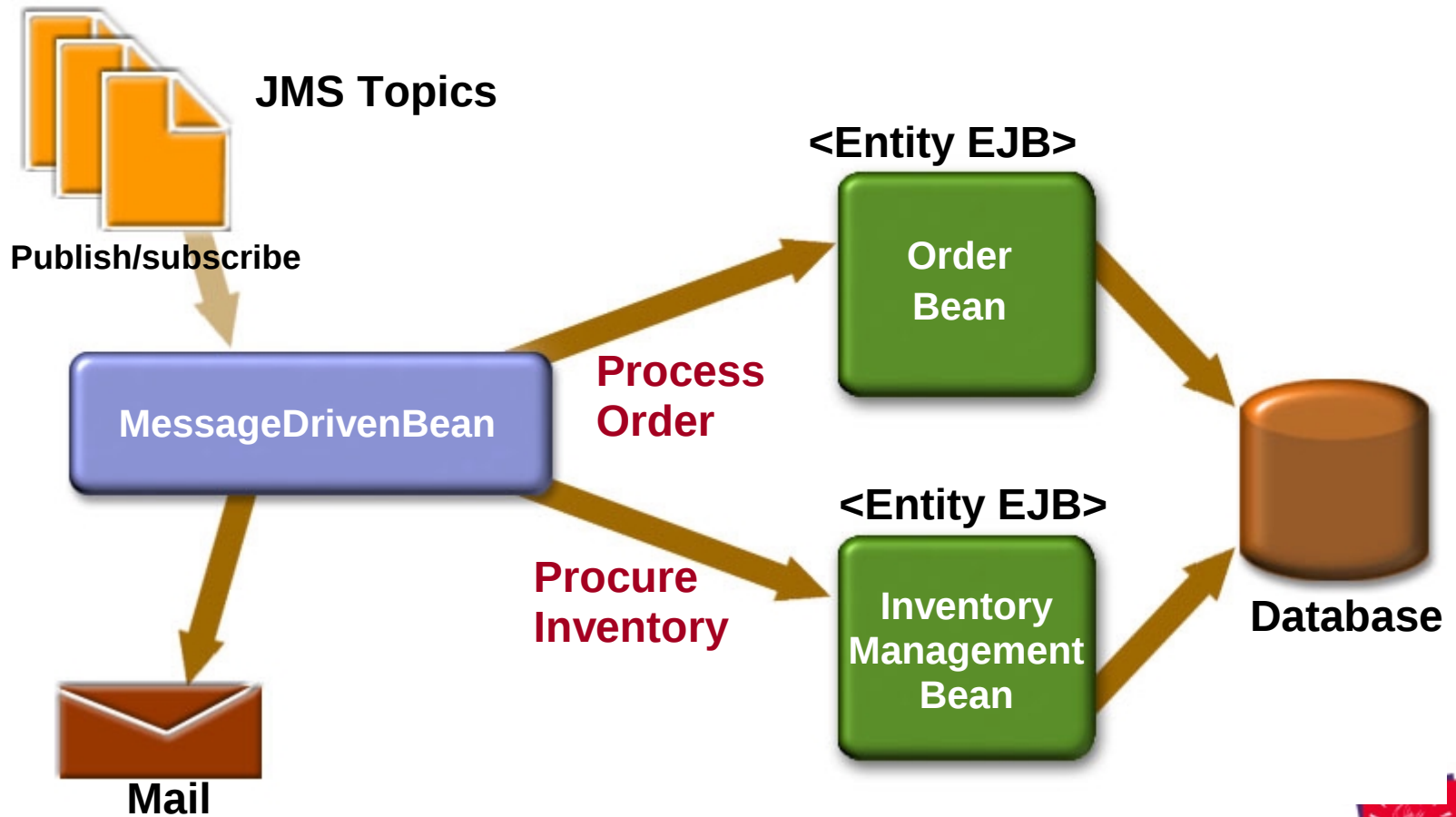


JMS and EJB Architecture

- JMS APIs for sending messages available to all enterprise beans
 - Use in point-to-point configurations
 - Reliable queuing
 - Use within pub/sub configurations
 - Beans publishing events



Message-driven Bean Scenario

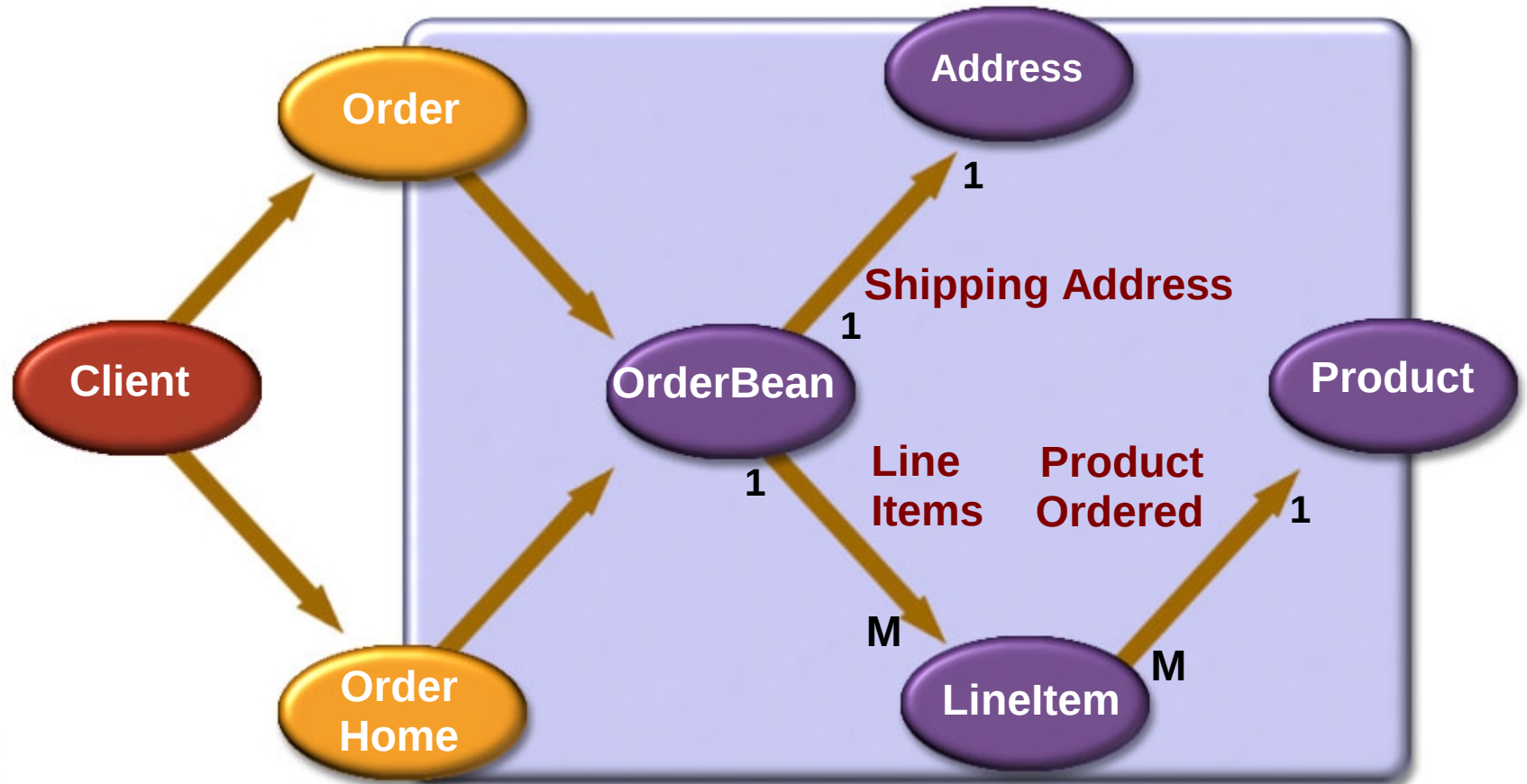


Goals for Container Managed Persistence

- Rich, flexible modeling
- Declarative, to enable portability
- Optimizable, scalable
- Pluggable



Abstract View of an Entity Application



Building Blocks

- Entity bean classes
 - Coarse-grained business objects
- Dependent object classes
 - Finer-grainer, dependent objects
- Dependent value classes
 - Provide client view of persistent state
 - Allow "legacy" classes to be included in persistent state



The Glue: Relationships

- 1-1, 1-N, M-N associations
 - Among beans; beans and dependent objects; dependent objects
 - Allow transparent navigation to related objects (beans or dependent classes)
 - Provide for unidirectional or bidirectional navigability
 - Foundation for path expressions in queries
- Allow creation of network of beans and dependent objects



Three Viewpoints on Container Managed Persistence

- Client view
- Bean Provider view
- Persistence Manager view



Client View

- The same as in bean-managed persistence
- Determined by home and remote interfaces of entity bean
- Entity bean can be evolved independently of client view

Bean Provider's View: Abstract Schema

- Logical abstraction over persistent state
- Declaratively defined in deployment descriptor
 - **cmp-fields** capture persistent state
 - **cmr-fields** capture persistent relationships
- Embodied in method-based API
- Foundation for query definition as well



Programming Model

- Persistent state and relationships accessed through methods:
 - get/set accessor methods for cmp/cmr fields
 - `java.util.Collection` API for multi-valued relationships
 - `create<type>` methods are factories for dependent objects
- Entity beans, dependent objects, accessor and create methods are abstract:
 - Concrete classes and methods are generated at deployment time
 - Persistence Manager provides implementation



Example

```
public abstract class OrderBean implements
    javax.ejb.EntityBean {
    // OrderBean accessors.
    public abstract Address getShippingAddress();
    public abstract java.util.Collection getLineItems();
    public abstract LineItem createLineItem();
    ...
    // business method
    public void addLineItem(Product p, int quantity) {
        LineItem l = createLineItem();
        l.setQuantity(quantity);
        l.setProduct(p);
        getLineItems().add(l);
    }...}
```



Persistence Manager's View

- Implements abstract schema:
 - Entity bean and dependent object classes
 - get/set methods; create methods
 - Collection classes used in relationships
- Implements bean's lifecycle contract: `ejbLoad`, `ejbStore`,...
- Maps query language to target datastore
- Manages all persistent state and relationships, including referential integrity at runtime



Advantages

- Rich, flexible modeling
 - Supports complex coarse-grained to fine-grained object modeling
 - Provides modeling of network of related beans and dependent objects
- Portable
 - Captures abstract schemas in deployment descriptor
 - Foundation for declarative query language
 - Bean is independent of target deployment environment



Advantages

- Optimizable, Scalable
 - Supports lazy loading
 - Supports dirty detection/efficient updates
 - Enables different caching strategies
 - Enables different concurrency control strategies
 - Enables query optimization
- Pluggable
 - Allows different vendors products to plug in
 - Supports variety of relational and non-relational persistent backend systems, deployment tools



Goals for EJB™ Query Language (EJB QL)

- Portable definition of finder methods for entity beans with container managed persistence
- Rich for querying complex relationships
- Declarative language, independent of persistence manager and data store



EJB™ QL features

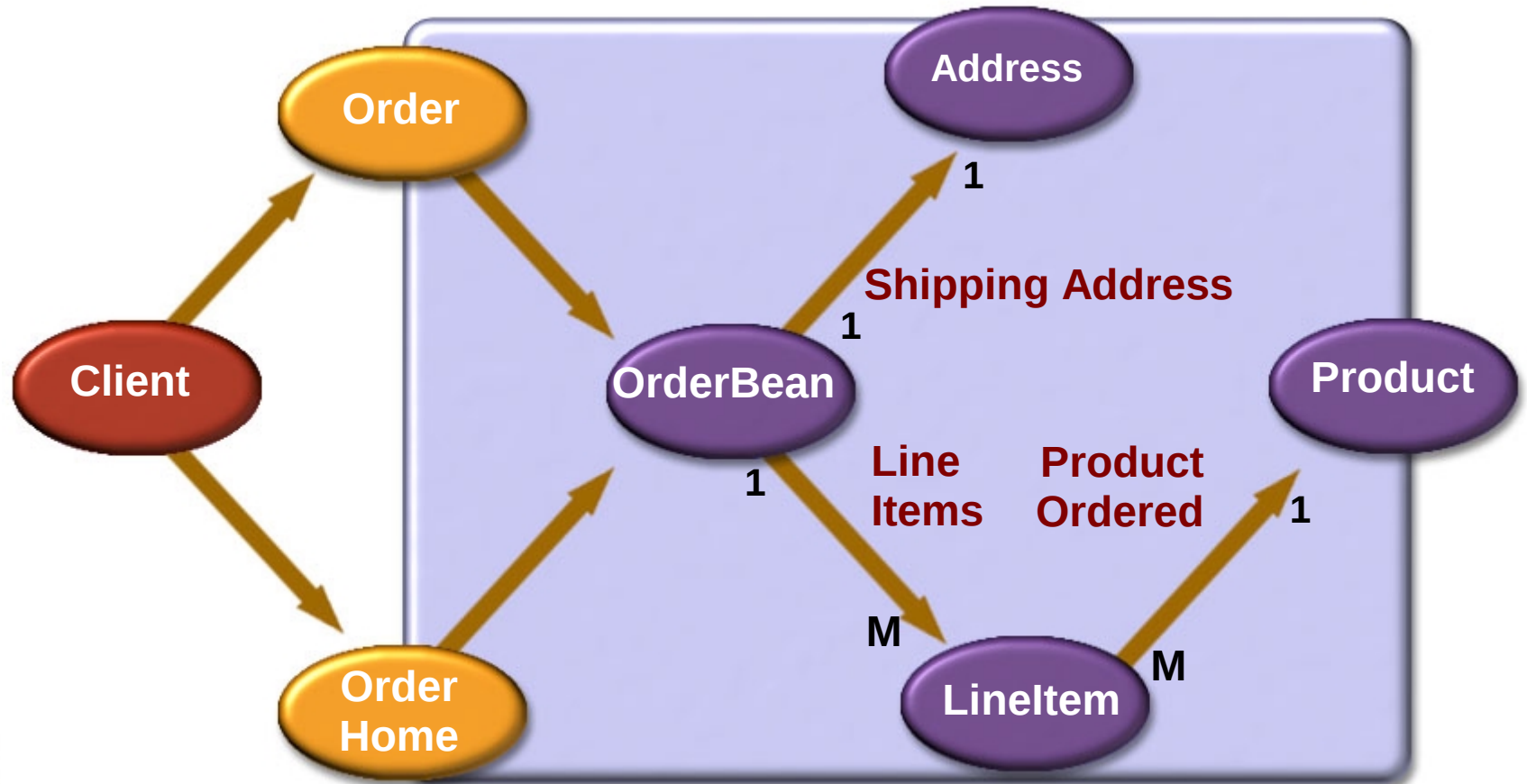
- Queries defined at logical, abstract schema level in deployment descriptor
- SQL-like typed language with SELECT, FROM, and WHERE clauses
- Query domain consists of all entity beans and dependent objects in ejb-jar file
- Based on navigability over abstract schemas
- Supports parameterized queries
- Special operators to support queries over remote entity beans



Query Methods

- Two kinds of query methods
 - ejbFind
 - ejbSelect
- Persistence Manager provides implementation based on declarative query specification

Abstract View of an Entity Application



Query Method ejbFind

- No SELECT clause
- Finds entity objects of given bean type

Examples: Find orders for a specific product:

```
FROM l IN lineitems, p FOR  
  l.product WHERE p.name = ?1
```

```
FROM l IN lineitems,  
  P FOR l=>product WHERE P IN  
  (ProductEJB >> findByName(?1))
```



Query Method `ejbSelect`

- `ejbSelect`
 - Internal finder method for use within entity bean class
 - Requires `SELECT` clause
 - **`ejbSelectWithEntity`** for finding related entities, dependent objects and values for a specific instance
 - **`ejbSelect`** for finding related entities and dependent values for entity bean type



ejbSelect Example

- Semantics depend on type of query method
 - EjbSelect: Find all products which have been ordered
 - ejbSelectWithEntity: Find all products which are ordered in *this* order.

```
SELECT p FROM l IN lineitems,  
p FOR l=>product
```

Home Methods

- Additional methods on Home interface for Entity Beans
 - Business methods that are *not* specific to an entity bean instance
 - Provided by the Bean Provider
 - Useful for bulk updates or other aggregate operations



Home Method Example

```
public interface OrderHome extends  
    javax.ejb.EJBHome {  
    public float taxRate(String statename)  
        throws RemoteException;  
  
    ...  
}
```

Current Status and Roadmap

- Public Draft available:
 - <http://java.sun.com/products/ejb/>
- Public Release target: Q4 2000
- Final Release as part of J2EE™ 1.3 platform
 - Specification
 - Reference Implementation
 - Compatibility Test Suite



Related Activities at the JavaOneSM Conference

- EJBTM 2.0 Architecture BOF: 9:00 pm Hall E
- J2EETM Platform and CORBA Interoperability: 2:45 Esplanade
- EJBTM Architecture-based Component ISV Panel: 8pm Hall E
- B2B Demo: J2EETM Platform in eCommerce Marketplaces at POD113





JavaOneSM
Sun's 2000 Worldwide Java Developer Conference*