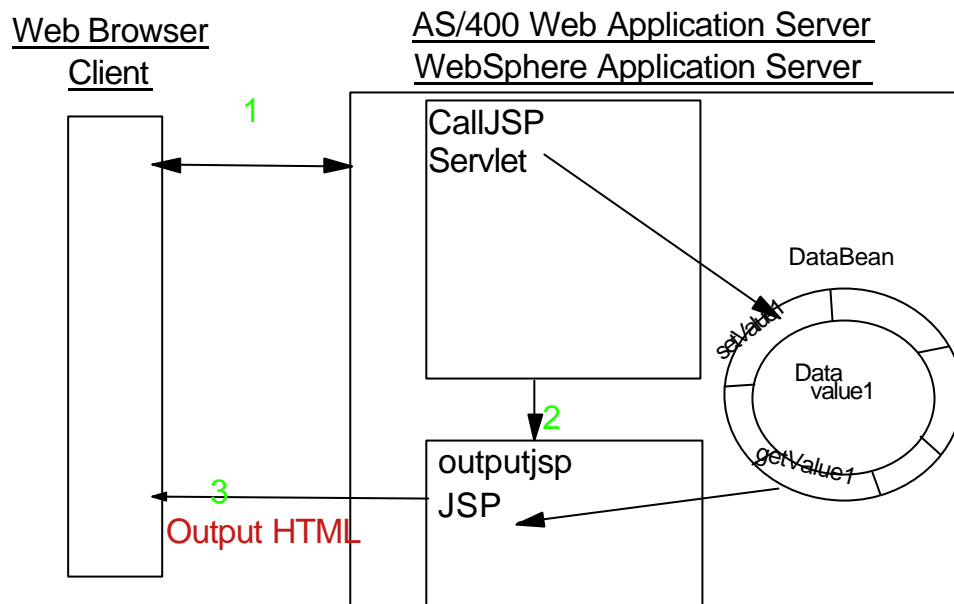


Simple JSP Example



© 2002 IBM Corporation

Enterprise JavaBeans definition

Requirements:

- Make it easier to write business applications
- Remove the need to understand low-level APIs, security, and transactions,

What are Enterprise JavaBeans?

- Architecture for defining how to build components for business applications
- Components that can be used to build distributed applications
- RMI Object with some attributes configurable at runtime
- Separate from, but complementary to, Java Beans

© 2002 IBM Corporation

What are EJBs?

EJB = Enterprise JavaBeans

- Component model targeted at server business logic development
- Interface that insulates the application programmer from complexities and platform dependencies of transaction systems, databases, security, etc.
- Sun initiative with broad industry momentum
- Specification, but no reference implementation
- Defines a component interface, but not the underlying implementation or packaging
- Container and server providers allowed to differentiate their offerings

EJB goals

Standard server-side component model

Binary portability "write once, run anywhere"

Make applications easy to write:

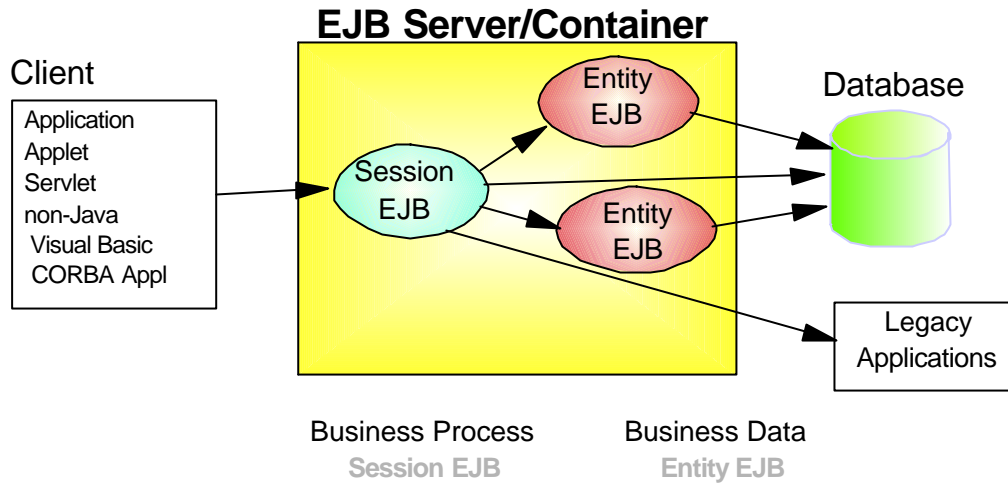
- Application assembly with visual tools
- Complexity of security, transactions, and persistence hidden

Standard interfaces for tool suppliers

Interoperate with non-Java components (using IIOP and COM bridges)

Co-exist with and reuse of existing server components, such as LDAP directories and XA Resource Managers

Enterprise JavaBeans Architecture

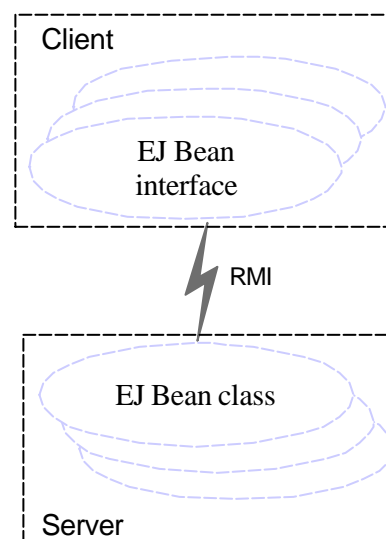


© 2002 IBM Corporation

EJB programming model

Enterprise Bean provider

- Develops domain-specific components that can be reused
- Provides:
 - ◆ Interfaces for application assembler
 - ◆ Class to implement business logic
 - ◆ Descriptor for deployment
- May rely on EJB container to provide system services



© 2002 IBM Corporation

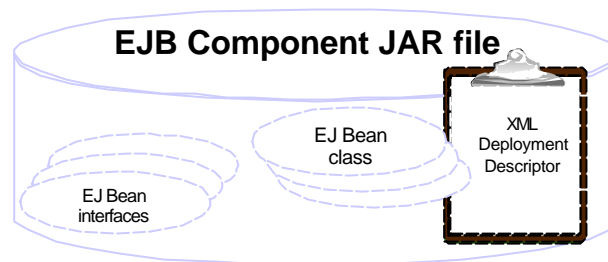
Deployment

EJB components are packaged in Jar files

Contains home and remote interfaces, bean classes, deployment descriptors and manifests

A bean's default settings that may be customized are set in the deployment descriptor

Settings for transactions, security, and user defined settings



Types of Enterprise JavaBeans

■ Session Beans

- ▶ Transient objects representing business tasks
- ▶ Typically reads and writes data to a database for a client
- ▶ Represent an action being taken on behalf of a single client
- ▶ May maintain state across method calls
- ▶ Examples include OrderPlacement and OrderEntryClerk

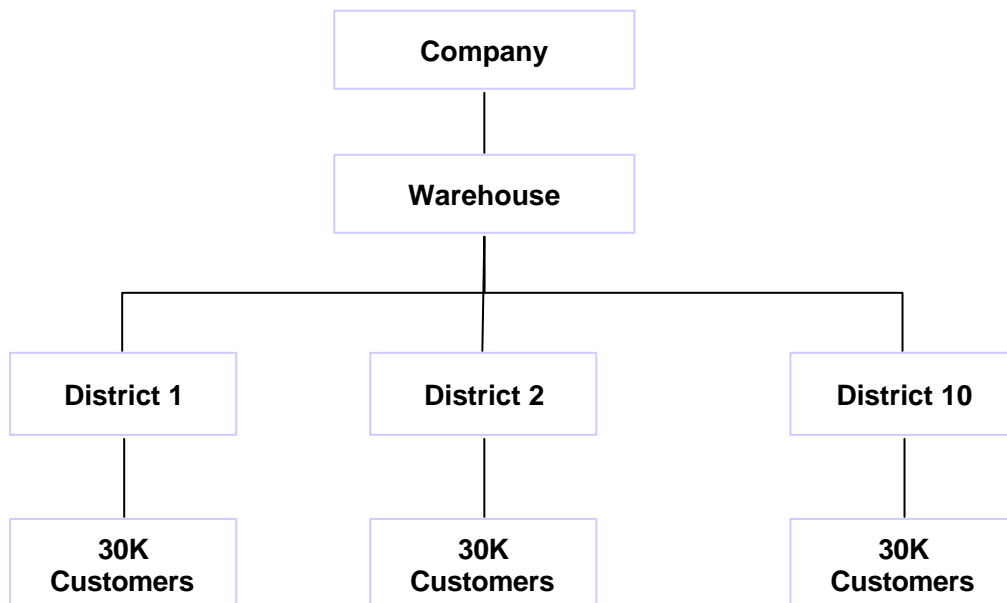
■ Entity Beans

- ▶ Persistent objects representing business data
 - Entity bean class represents a table
 - Instance of an entity bean represents a row of a table
- ▶ Each entity EJB is uniquely identifiable (primary key)
- ▶ Persistence managed by the container or the bean itself
- ▶ Examples include Stock and Order

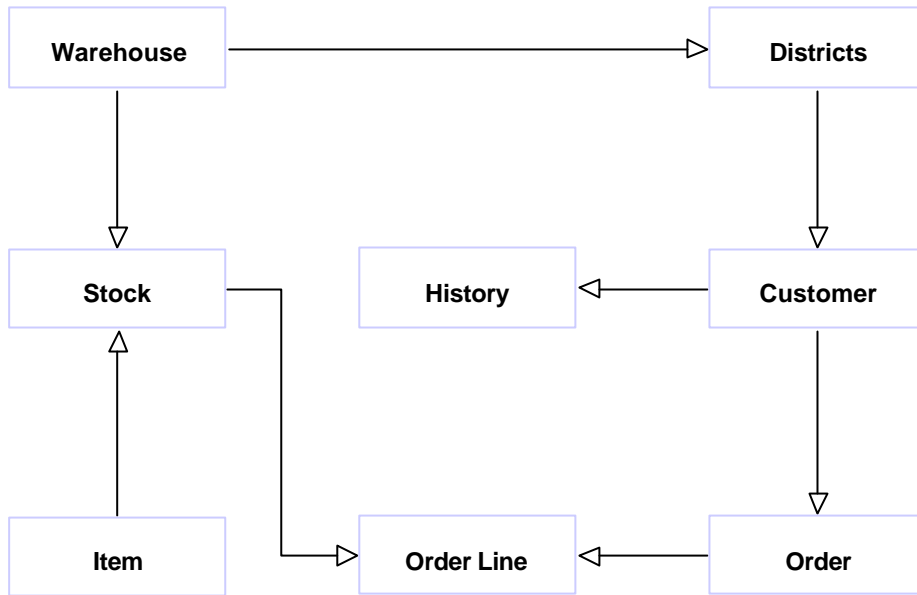
Bean Persistence Choices

- **Applies only to entity beans**
- **Container-managed persistence**
 - ▶ The container provider manages reading and writing bean attributes to database
 - ▶ The bean provider sets up a mapping from bean attributes to database columns used by the container
 - ▶ This has the advantage of less programming for the bean provider and the bean code is independent of persistence details
- **Bean-managed persistence**
 - ▶ The bean provider manages reading and writing bean attributes to the database
 - ▶ This has the advantage of greater flexibility when mapping to databases and the bean can interface to existing applications

Order entry application



Order entry application



Application objects

District	
	ivDistrictID : String
	ivWarehouseID : String
	ivNextOrderNumber : int
	ivDistrictName : String
	ivDistrictTax : float
	ivYTDBalance : float

Item	
	ivItemID : String
	ivItemName : String
	ivItemPrice : float
	ivItemInfo : String

Customer	
	ivCustomerID
	ivFirstName
	ivLastName
	ivInitials
	ivPhone
	ivCreditLimit
	ivBalance
	ivYTDBalance

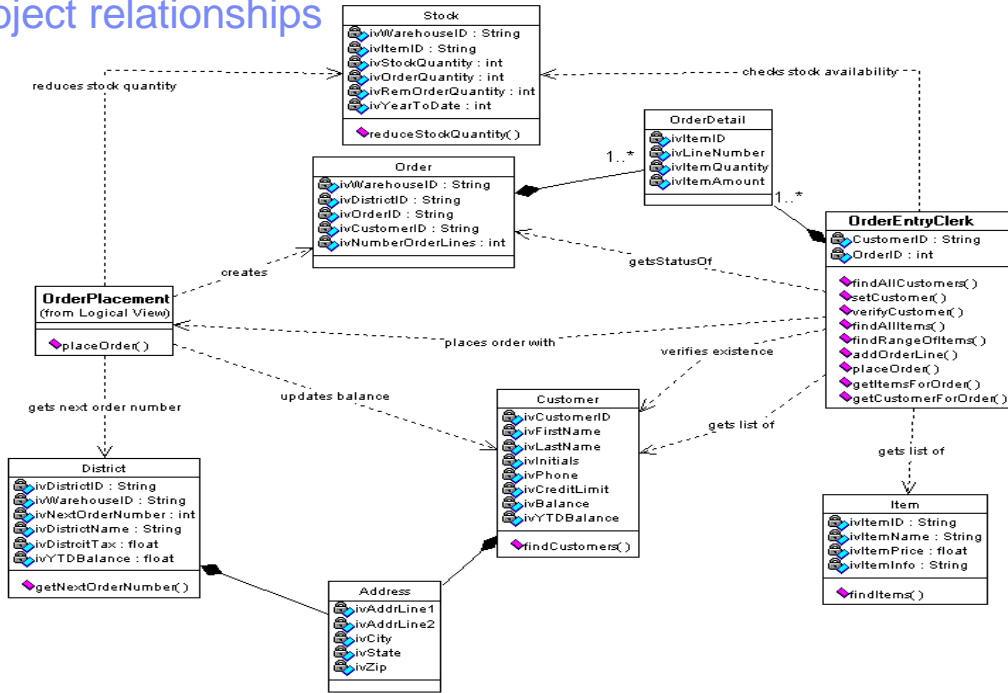
Address	
	ivAddrLine1
	ivAddrLine2
	ivCity
	ivState
	ivZip

Stock	
	ivWarehouseID : String
	ivItemID : String
	ivStockQuantity : int
	ivOrderQuantity : int
	ivRemOrderQuantity : int
	ivYearToDate : int

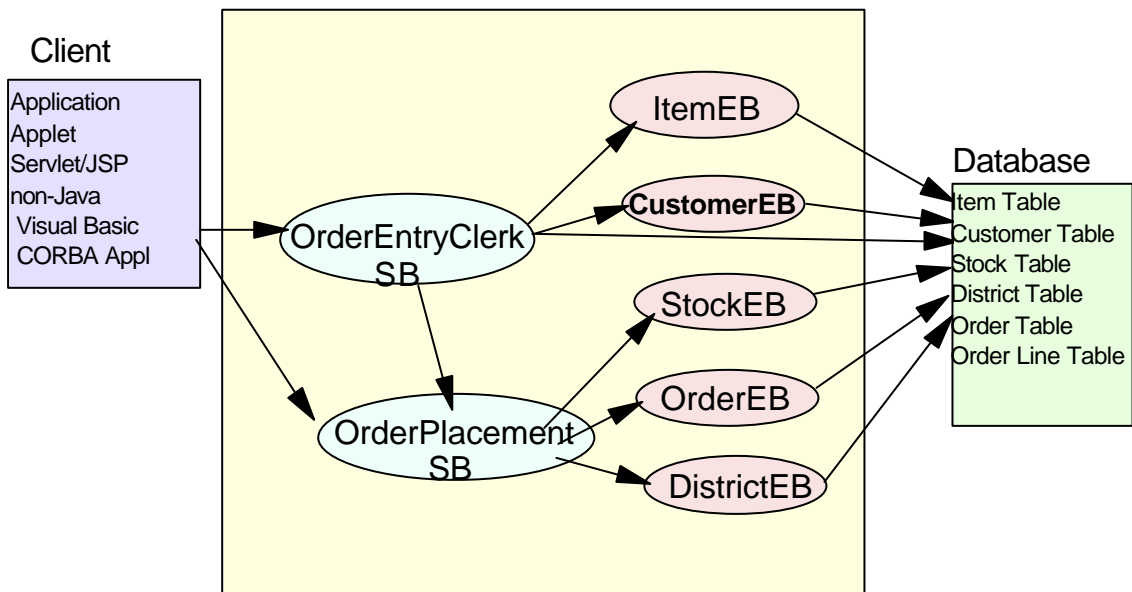
Order	
	ivWarehouseID : String
	ivDistrictID : String
	ivOrderID : String
	ivCustomerID : String
	ivNumberOrderLines : int

OrderDetail	
	ivItemID
	ivLineNumber
	ivItemQuantity
	ivItemAmount

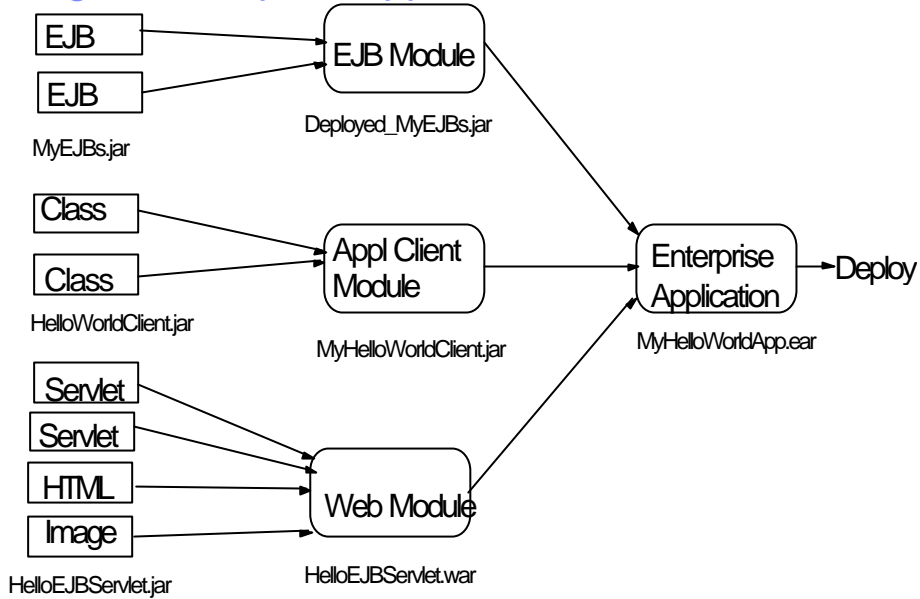
Object relationships



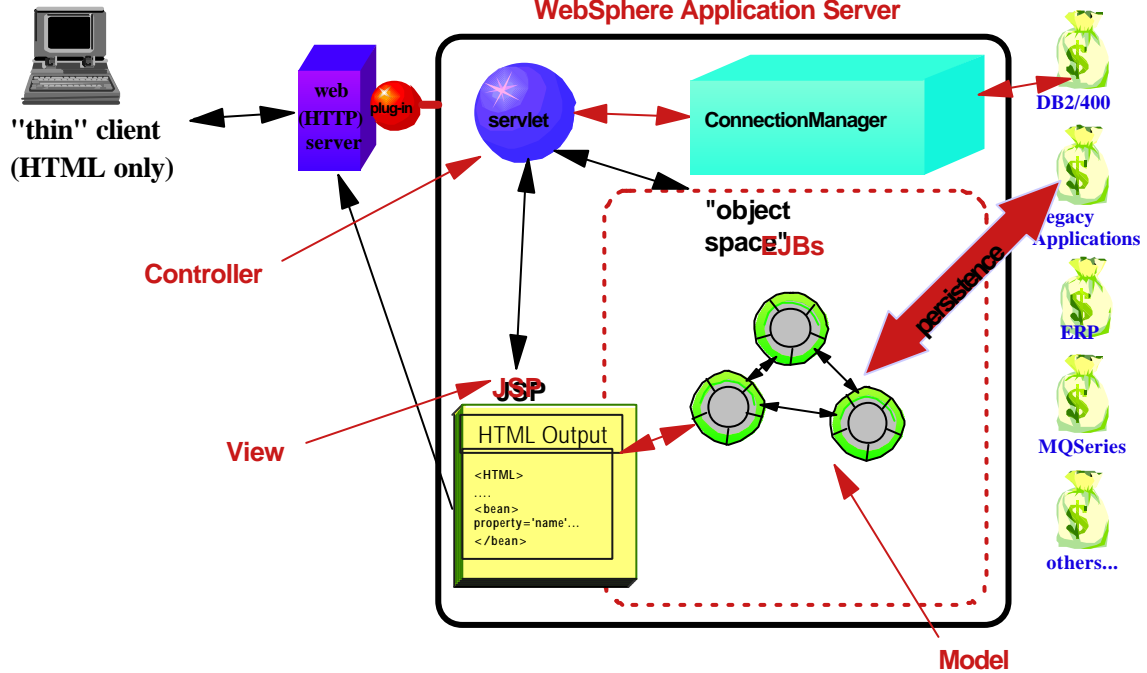
Order Entry Application EJBs



Creating an Enterprise application

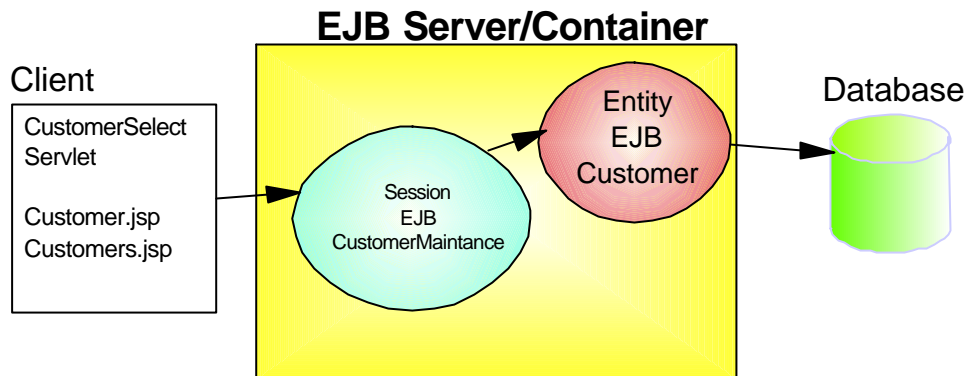


Model-View-Controller Architecture



ABC Company example

- Customer inquiry application using Servlet-JSP-EJB
- Customer inquiry using JSP and RPG *svrpgm



Evaluation

- **Advantages**
 - ▶ 100% flexibility
 - ▶ Benefit from Object Technology
 - ▶ Real internet/intranet applications
- **Disadvantages**
 - ▶ Much more complex
 - ▶ No 5250 support

More information...

■ Redbooks

- ▶ SG24-6600 WebSphere Development Studio Client for iSeries
- ▶ SG24-6804 iSeries Access for Web V5R2 and WebSphere Host Publisher V4.0
- ▶ Various WebSphere and WebSphere Studio redbooks