Ignify Consulting

Enterprise JavaBeans

Author: Rizwan Shaikh

rizwan@ignify.com

Ignify Consulting
13304 Alondra Blvd #201.
Cerritos CA 90703
www.ignify.com

March 2003

Overview:

- Enterprise JavaBeans
  - This technology defines a model for the development and deployment of reusable Java™ *server* components.
  - *Components* are pre-developed pieces of application code that can be assembled into working application systems.
  - Java technology currently has a component model called JavaBeans™, which supports reusable *development* components.
  - The EJB architecture *logically* extends the JavaBeans component model to support server components.

Overview: ...contd.

- Enterprise JavaBeans

  - It is a component model for enterprise applications.
  - Combines server-side components with distributed object technologies such as CORBA and Java RMI.

The convergence of these two technologies: the traditional transaction processing monitors and distributed object services is represented by Component Transaction Monitors (CTMs).

Ignify Consulting
excellence in technology

Some Concepts Defined…..

Distributed Objects:

Distributed computing allows:
- a business system to be more accessible.
- parts of the systems to be located on different locations.

The most recent development in distributed computing is *distributed objects*. Distributed object technologies such as CORBA, RMI and Microsoft's DCOM allow objects running on one machine to be used by client applications on different computers.

Ignify Consulting
excellence in technology

Distributed object computing extends an object-oriented programming system by allowing objects to be distributed across a heterogeneous network, so that each of these distributed object components interoperate as a unified whole. These objects may be distributed on different computers throughout a network, living within their own address space outside of an application, and yet appear as though they were local to an application.

Ignify Consulting
excellence in technology

Server-Side Components:

- Server components are application components that run in an application server.
- A server-side component model defines an architecture for developing *distributed-business objects.* They combine the accessibility of distributed object systems with the fluidity of objectified business logic.
- Server-side component model supports attribute-based programming, which allows the runtime behavior of the component to be modified when it is deployed without having to change the programming code in the component. This makes it flexible, extensible and reusable.

Ignify Consulting
excellence in technology

Component Transaction Monitors (CTMs):

- These are distributed object application servers. Application servers are a combination of servers with different technologies, including web-servers,
- ORBs, databases and so forth.

- CTMs evolved as a hybrid of traditional TP monitors and ORB technologies.

- They implement robust server-side component models. They provide an infrastructure that can automatically manage transactions, object distribution, concurrency, security, persistence and resource management

Ignify Consulting
excellence in technology

- **TP Monitors**: Transaction Processing Monitors are powerful, high speed *server platforms* for mission-critical applications.

- TP monitor systems **automatically manage** the entire environment that a business system runs in, including transaction resource management and fault tolerance.

- The business logic is made up of procedural applications that are accessed through network messaging or RPC.

- *Messaging* allows a client to send a message directly to the TP monitor requesting that the application be run using certain parameters.

- *RPC* is a distributed mechanism that allows a client to invoke procedures or applications in a TP monitor as if the procedures were run locally.

Ignify Consulting
excellence in technology

ORB: Object Request Broker is an *application server* that facilitates the connectivity between the client application and distributed objects.

These are communication back bones that are used to access and interact with remote objects. When a distributed object's method is invoked it is an object instance and not an application procedure.

The **responsibility** for concurrency, transactions, resource management and fault tolerance falls **on the developer** of the distributed object application using an ORB.

Ignify Consulting
excellence in technology

Enterprise JavaBeans: Defined

Sun Microsystems' definition of Enterprise JavaBeans is:

The EJB architecture is a component architecture for the development and deployment of component-based distributed business applications. Applications written using the EJB architecture are scalable, transactional and multi-user secure. These applications can be written once and then deployed on any server platform that supports the EJB specifications.

Ignify Consulting
excellence in technology

Distributed Object Architecture:

Distributed object systems are the foundation for modern 3-tier architecture.

In 3-tier architecture the presentation logic resides on the client (1st tier), the business logic on the middle tier (2nd tier) and other resources such as the databases reside into the backend (3rd tier).

More complex architectures are often used in which there are many tiers: different objects reside on different servers and interact to get the job done. Creating these n-tier architectures with EJB is particularly easy.

Ignify Consulting
excellence in technology

Component Models:

The term “component model” has many different interpretations.

The JavaBeans is intended to be used for *intra*process purposes, while EJB is designed to be used for *inter*process components . In other words, JavaBeans was not intended for distributed components.

A component model defines a set of interfaces and classes in the form of Java packages that must be used in a particular way to isolate and encapsulate a set of functionality. Once a component is defined, it becomes an independent piece of software that can be distributed and used in other application. A component is developed for a specific purpose but not a specific application.

Ignify Consulting
excellence in technology

EJB Architecture:

Enterprise JavaBeans server-side components come in two fundamentally different types:

- **Entity Beans** and
- **Session Beans.**

Entity beans are *persistent objects* that are maintained in a permanent data store. A primary key identifies each instance of an entity bean. They model the real world objects.

Session beans are an extension of the client application and are responsible for process or tasks. These are *transient objects* and exists only for the duration of a single client/server session.

Ignify Consulting
excellence in technology

Every Enterprise bean is present in a home called the **EJB container.**

An EJB container manages the enterprise beans contained within it. For each enterprise bean, the container is responsible for:

- registering the object.
- providing a remote interface for the object.
- creating and destroying object instances.
- checking security of the object.
- managing the active state for the object and
- coordinating distributed transactions.

Optionally the container can also maintain all persistent data within the object.

Ignify Consulting
excellence in technology

Implementation of EJB using Classes and Interfaces:

- Remote Interface
  Defines bean's business methods.
- Home interface
  Defines bean's life cycle methods: methods for creating new beans, removing beans and finding beans.
- Bean class
  Implements the bean's business methods.
- Primary key
  Provides a pointer into the database.

Ignify Consulting
excellence in technology

Deployment Descriptors and JAR Files:

Deployment descriptors allow us to customize behavior of enterprise beans at runtime without having to change the software itself.
They allow to set the runtime attributes of the server-side components.

A JAR (Java ARchive) file packages the EJB. It includes bean classes, remote interfaces, home interfaces and primary keys, for each bean.

Ignify Consulting
excellence in technology

Primary Services:

There are many value-added services available for distributed applications. Among them **six** primary services are required to create an effective CTM.

These services are automatically managed by EJB servers.

EJB servers provide an environment that supports the execution of applications developed using EJB technology. It manages and coordinates the allocation of resources to the application.

These services are……..

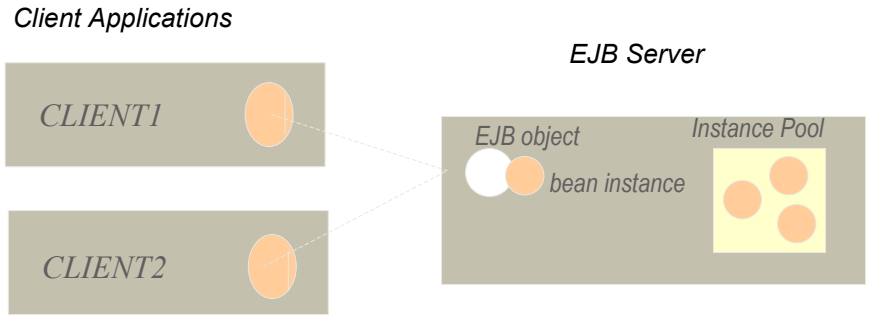
Ignify Consulting
excellence in technology

Primary Services:

- Concurrency
- Transaction
- Persistence
- Distributed Objects
- Naming
- Security

Ignify Consulting
excellence in technology

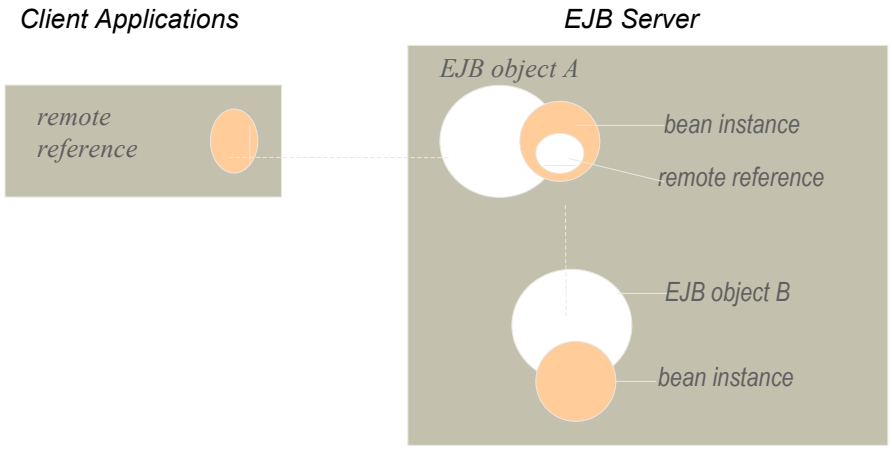
Concurrency:



Client sharing access to the EJB object.

Ignify Consulting
excellence in technology

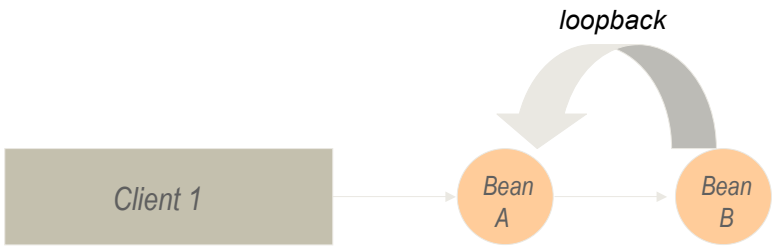
Reentrance:



Bean access each other through EJB objects

Ignify Consulting
excellence in technology

Loop Back



A loopback scenario

Ignify Consulting
excellence in technology

Transaction:

A unit of work or set of tasks that are executed together.
Each transaction is atomic i.e. All tasks in the transaction are carried out in a bunch.

Transactions are managed automatically.
EJB also provides for managing transactions explicitly.

Ignify Consulting
excellence in technology

Persistence:

- Container-Managed Persistence

  The EJB container is responsible for synchronizing an entity bean's instance fields with the data in the database, when a bean's state is automatically managed by a persistence service.

- Bean-Managed Persistence

  When beans are designed to manage their own state.

Distributed Objects:

The distributed services available: CORBA, Java RMI and DCOM. Each uses a different RMI network protocol.

- **CORBA**: Internet Inter ORB Protocol (IIOP).
- **Java RMI**: Java Remote Method Protocol (JRMP).
- **DCOM**: Object Remote Procedure Call (ORPC).

If the client view is supported by the EJB Server, any distributed object protocol can be used.

Naming:

- Object Binding
  - Association of a distributed object with an identifier.
  - A pointer or an index to a specific distributed object.

- Lookup API
  - Interface to the naming system.
  - Allows client to connect to a distributed service.
    Request a remote reference to a specific object.

  EJB mandates the use of JNDI as a lookup API.

Ignify Consulting
excellence in technology

Security:

- Authentication
  -Validates the identity of the user
- Access Control
  -Allows users access to resources for which they have permission
- Secure Communication
  -Communication between the client and the server is secured here by encryption.

Ignify Consulting
excellence in technology

Sources:

- http://members.tripod.com/gsraj/misc/ejbmts/ejbmtscomp.html
  - Comparison between MTS and EJB
  - Comparison between CORBA, DCOM and Java RMI
- www.ejbtut.com
  - Complete tutorial for EJB
- http://java.sun.com/products/ejb/faq.html
  - Sun's EJB site for FAQs.
  - Covers entire EJB from basics to deployment
  - Enterprise JavaBeans 2nd Ed. – Richard Monson Haefel O'Reilly Publication.

Ignify Consulting
excellence in technology

- **Los Angeles Office**
  - 13304 Alondra Blvd #201, Cerritos CA 90703
  - Email: sales@Ignify.com
  - Tel: 562-404-8089
- **India Office**
  - 7 Madhuban, North Main Road, Koregaon Park, Pune
  - Tel: +91-20-612-0778
  - Email: India@Ignify.com
- **San Francisco Bay Area**
  - 4800 Great America Pkwy, Suite 310, Santa Clara, CA 95054
  - Tel: 408-480-3289

- **Global Website**: http://www.Ignify.com

Ignify Consulting
excellence in technology