Distributed Processing with J2EE Technology

Presenter: Wickramanayake HMKSK
Technical Trainer
Virtusa (Pvt) Ltd
Email: kwickramanayake@virtusa.com
Version: 0.1
Last Updated: 05-Oct-2004
Objectives

- Differentiate between client-server and multi-tier architecture
- List the advantages and issues with distributed architectures
- List the advantages of J2EE
- Discuss the goals and scope of J2EE and EJB
- List two primary features of the EJB specification
- List two benefits of an EJB solution
Centralized Architecture

- Operation is centralized
- And non-distributed
Client-Server (Two-tier) Architecture

- Work is somewhat distributed
- Clients are fat

First Tier
- Fat Client
  (Business Logic)

Second Tier
- Server
  (Data)
Three-tier Architecture

- Business logic moves towards the EIS and is separated from client
- Clients can be thin!
- Many advantages!
Advantages of Three-tier Distributed Technology

- **Scalability**
  Ability to add additional computing resources to accommodate increasing client loads without changing the application.

- **Reliability and Availability**
  Separation of presentation and business logic tends to make the system more reliable and available.

- **Extensibility**
  Ability to add additional functionality without impacting existing functionality.

- **Software Resource Balancing**
  Ability of many machines or processes to request the resources or services of other machines or processes.

- **Security**
  Authentication and authorization issues – For a complete corruption, a hacker has to break three tiers.
Issues With Distributed Architecture

- **Additional complexity**

  Modern applications are inherently complex. In case of a failure, finding the exact point of failure may be difficult.

- **Multiple points of failure**

  Problems can occur at many locations. But a system may still work as opposed to a client-server architecture where a failure may severely affect the operation.

- **Network Bandwidth**

  An additional bandwidth is required for the operation and hence the system is required to have a good architecture.
Enterprise Complexity

- Business Intelligence
- Web Cache
- Portal
- Wireless & Mobile
- J2EE
- Management & Security
- Web Services
- B2B Integration
- Application Integration
- Any Data Source
What is J2EE?

J2EE Consists of four parts:

- Platform Specification
  - Defines J2EE requirements

- Reference Implementation
  - Operational J2EE platform

- Compatibility Test Suit
  - Validates J2EE platform compatibility

- J2EE Blueprints
  - Describes how to build (better) J2EE applications
J2EE Architecture Overview
What Is An Application Server?

- An application server is a software that provides infrastructure to support management of business logic and access to services.

- The application server is frequently viewed as part of a three-tier application, consisting of a graphical user interface (GUI) server, an application (business logic) server, and a database server.
What Is Business Logic?

- The code that implements the functionality of an application.
- The rules associated with the data in a database that typically encode business policies.
- An example is automatically adding late fees for overdue items.
An J2EE application server provides a service-oriented infrastructure to automatically support and manage components.

The enterprise developer can concentrate on application components, not the underlying services.

Separation of business logic and services provide for better reuse of business logic.
Enterprise JavaBeans

- Enterprise JavaBeans (EJB) is an architecture for component-based distributed computing
  - Customizable at deployment time
  - Deployed on a compatible application server
  - Portable to other application servers

- Enterprise Beans (EB) are components of distributed transaction-oriented enterprise applications.

EJB Developer Roles

- **Bean Provider**
  - Writes beans that perform business operations

- **Application Assembler**
  - Assemble beans together to form an application group and may also create client applications

- **Deployer**
  - Specifies the runtime-environment that is used by the bean provider and the application assembler in the deployment descriptor (a configuration file)

- **Container Developer**
  - Creates the container (Server Developer probably provides containers)

- **Server Developer**
  - Creates the EJB server

- **System Administrator**
Questions?