

IBM Presents:

Noospherics Technologies – IBM Premier Business Partner

DPT Consulting – Enterprise Training Provider

Managing WebSphere / J2EE Projects

Agenda:

- Welcome
- Overview of Requirements in e-Business Projects
- Overview of WebSphere Platform and the J2EE Architecture
- Planning and Execution Of A WebSphere / J2EE Projects
- BREAK
- **Continued** Execution Of A WebSphere / J2EE Projects
- Managing A WebSphere Application In Production
- Closing Remarks – Next Steps

Noospherics Technologies

- IBM Premier Business Partner
- IBM Web Integrator Partner
- Provide consulting, training and project outsourcing
- Helping your business become an e-business
- Specialize in IBM WebSphere and J2EE technologies

- Enterprise / J2EE Application Architects
- IBM WebSphere-certified developers and administrators
- Sun Java-certified developers
- IBM-certified MQ/MQSI specialists

Noospherics Technologies

- Comprehensive training and mentoring
- Courses:
 - Managing WebSphere/J2EE Projects
 - J2EE Development with VisualAge and WebSphere
 - WebSphere Administration
 - WebSphere 4.0 Migration
 - Enterprise Development with JMS and MQSeries

DPT Consulting Group (overview)

- 14 Years of Experience Delivering Technology Training and Mentoring
- Helps Clients to Develop and Implement the People Strategy Necessary for E-Business Transformation
- Accredited by ACCET and Courses Recommended for College Credit by ACE
- Vendor Certified Instructors
- Training offered in:
 - Web Development (J2EE, .Net, WebSphere, WebLogic, IPlanet, HP/Bluestone)
 - Databases (Oracle, SQL Server, UDB/DB2)
 - Client/Server and Mainframe Technologies
 - Messaging (MQSeries)

DPT Consulting Group (services)

- Delivery of Instructor Led Training
- Development of E-Business People Strategy
- Customized Training Roadmaps for Re-Skilling
- Skills Based Management
- Mentoring Services
- Blended Curriculum Design (integration of ILT and WBT)
- Curriculum Design, Development and Customization

Noospherics and DPT Partnership

- Provide Total Solution for E-Business Technology and People Strategy
 - Can Provide the Right Blend of Consulting/Mentoring/Training Services to Match Each Organization's Goals and Objectives
- Project Management, Mentoring and ReSkilling Blended as ONE Solution

Introductions

- Name
- Company
- Interest in using WebSphere / J2EE
- Type of Application

Agenda:

- Welcome
-  **Overview of Requirements in e-Business Projects**
- Overview of WebSphere Platform and the J2EE Architecture
- Planning and Execution Of WebSphere / J2EE Projects
- BREAK
- **Continued Execution Of A WebSphere / J2EE Projects**
- Managing A WebSphere Application In Production
- Closing Remarks – Next Steps

Overview of Requirements in e-Business Projects

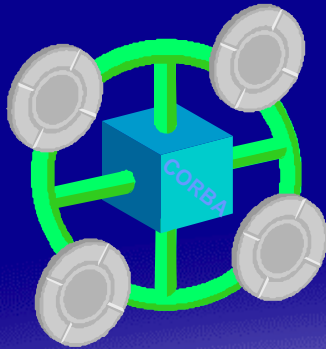
- Agility – able to change process fast
- Scalability – millions of potential users
- Security – transactions over the web
- Reliability – automatic fail-over, redundancy
- Maintainability

Agenda:

- Welcome
- Overview of Requirements in e-Business Projects
-  Overview of WebSphere Platform and the J2EE Architecture
- Planning and Execution Of WebSphere / J2EE Project
- BREAK
- **Continued** Execution Of A WebSphere / J2EE Projects
- Managing A WebSphere Application In Production
- Closing Remarks – Next Steps

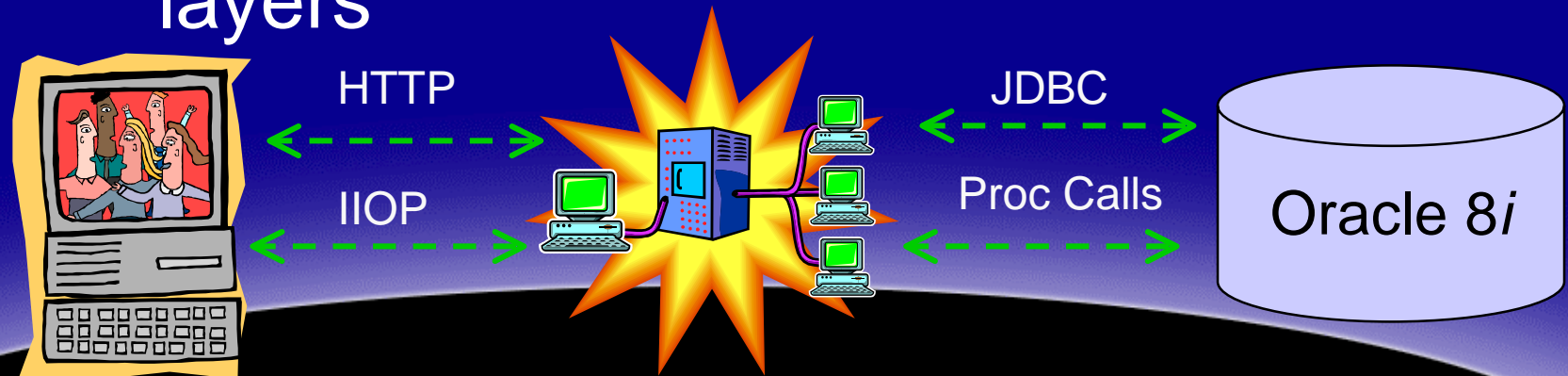
Two Recent Trends in Software Development

- Component based application development
- Three-tier/multi-tier application model



Multi-tiered Architecture

- Separates business logic from User Interface and system services
- Business logic in the middle layers
- Loose coupling in between the layers



Multi-tiered Architecture Benefits

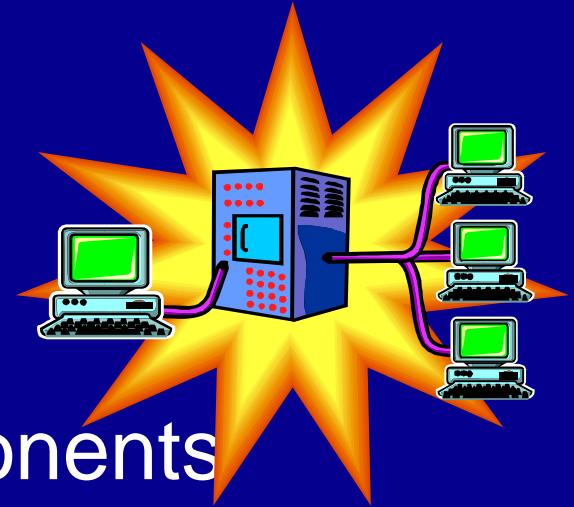
- Thin clients
- Enhanced portability
- Easier to change and maintain
- More scalable and flexible
- Reduced development time
- More effective use of resources

Complexities in the Architecture

- Distribution and integration
- Communication between layers and components
- Synchronization and transaction management
- Resource management

Server-Side Tasks

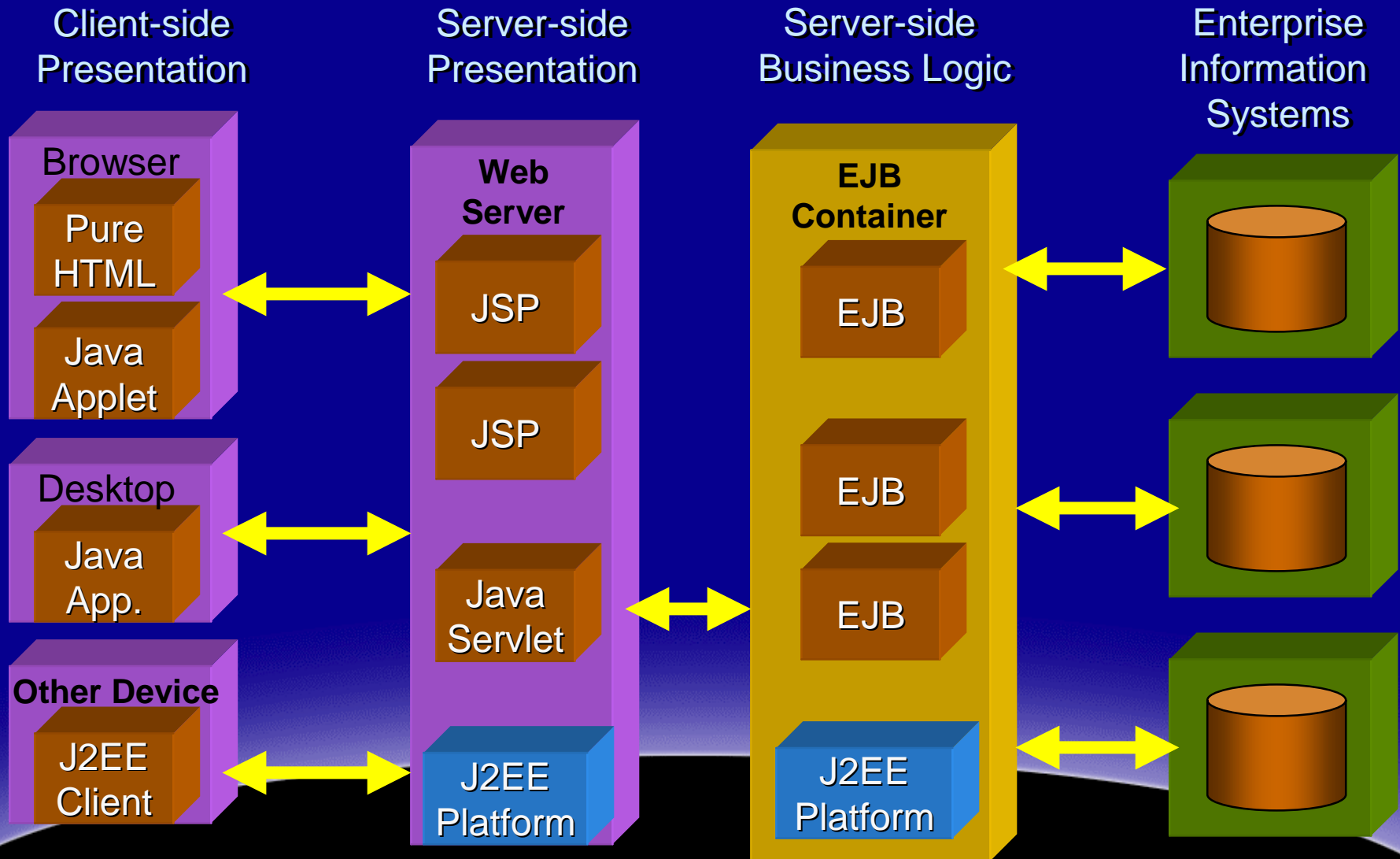
- Handle multiple clients
- Scalability issues
- Transactions
- Shared resources
- Persistence
- Larger life cycle of components



Application Servers

- Provide infrastructure support and services to business components
- Provide framework for communication between components and layers
- Manage the life cycle of business and other components
- Manage distribution, scaling, replication, recovery and so on

J2EE Application Architecture



J2EE Application Services

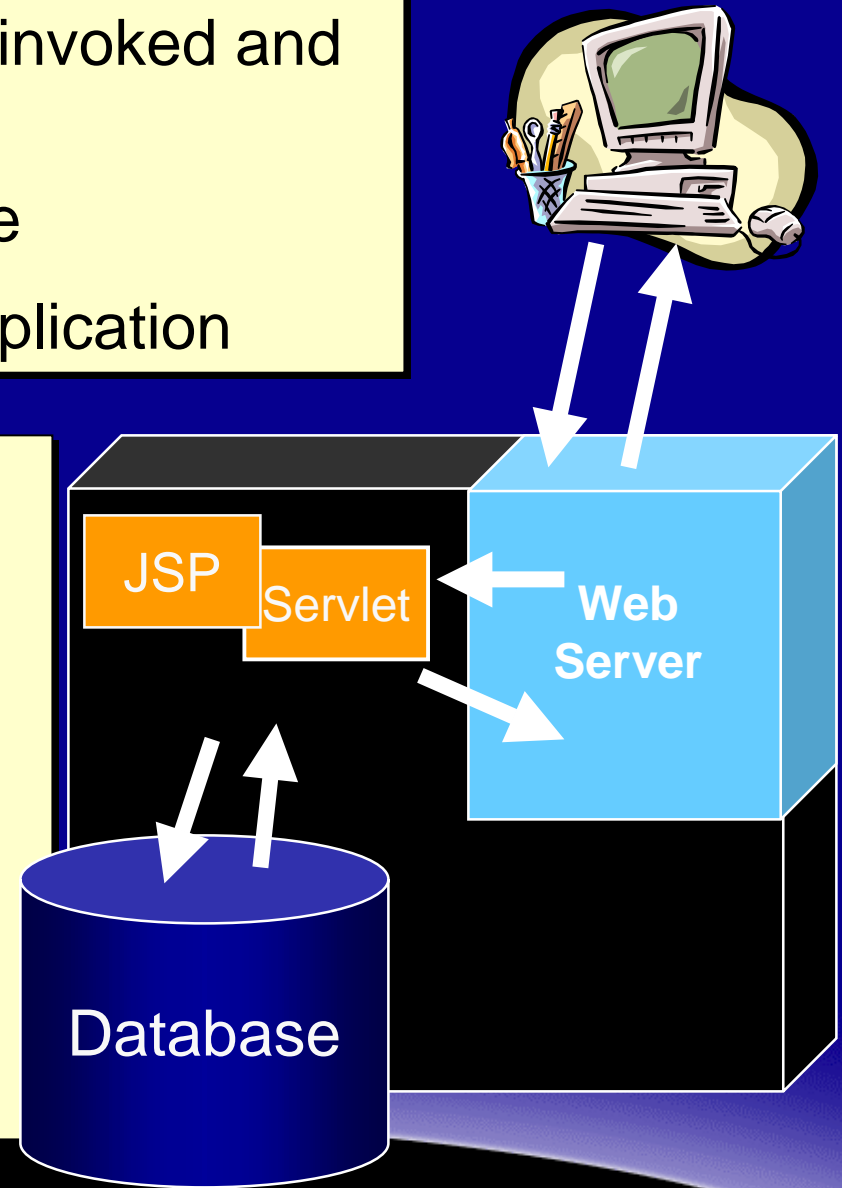
- Servlets - Controls app flow
- JSP - view (user interface)
- JDBC - database access
- EJB - back-end logic/persistence
- JNDI - naming service
- JMS - message service
- XML - back-end integration
- JTA - transaction services
- RMI-IIOP - remote object services
- Java Mail - email services

Servlets

- Stand alone classes that can be invoked and executed on the server
- Provide request/response service
- Control page interaction in an application

JSPs

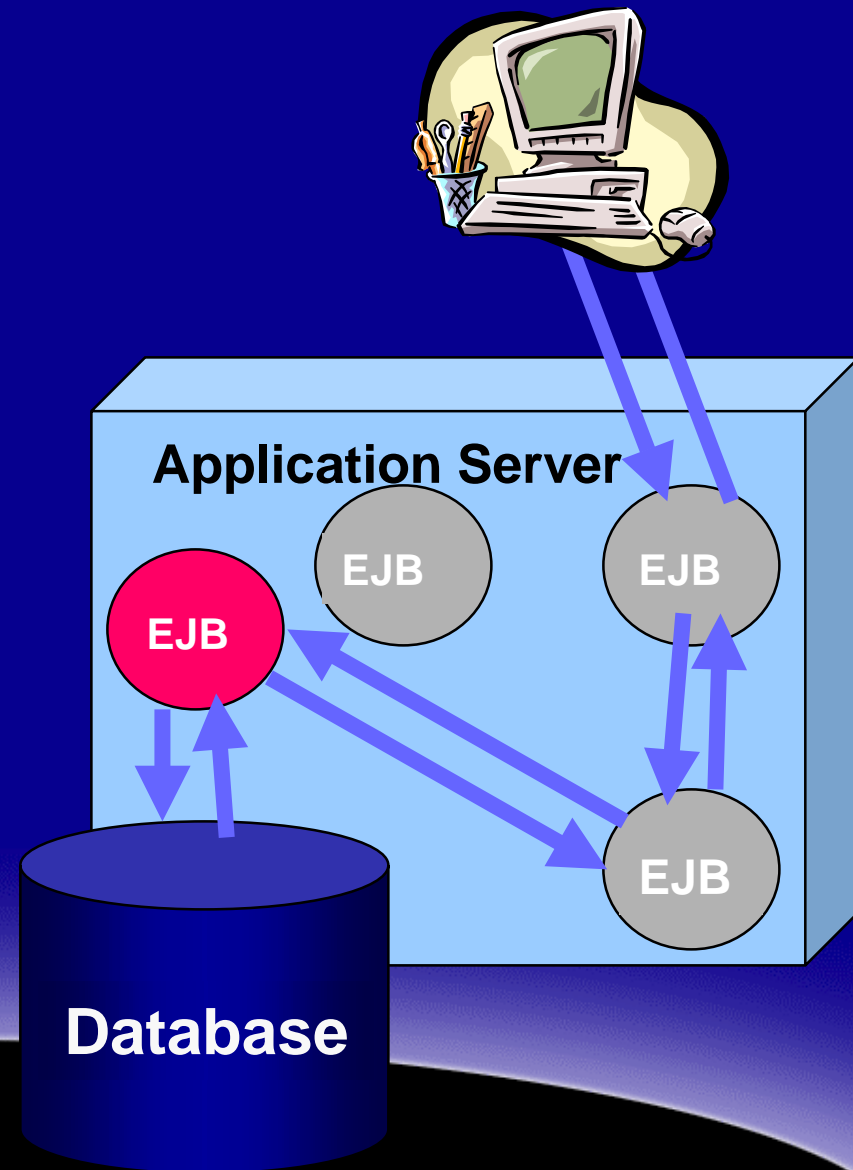
- Provide dynamic content
- Executed as servlets by a web server
- Use special tags to hold code
- Separate page design from programming



Enterprise JavaBeans

EJBs:

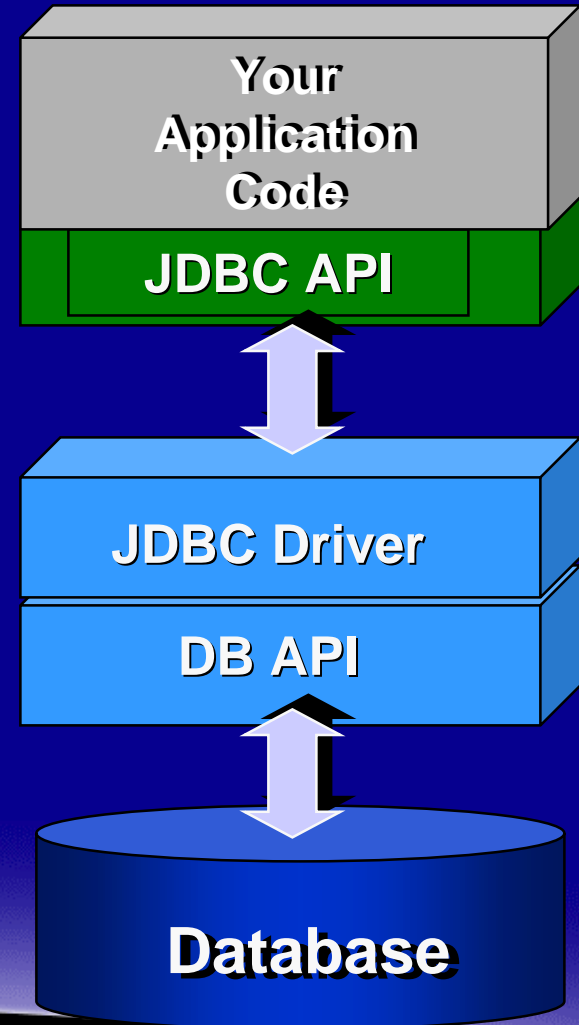
- Server side business components
- Have well-defined interfaces and different levels of state
- Can be used in any compliant App. Server without any code changes or recompilation



Java Database Connectivity

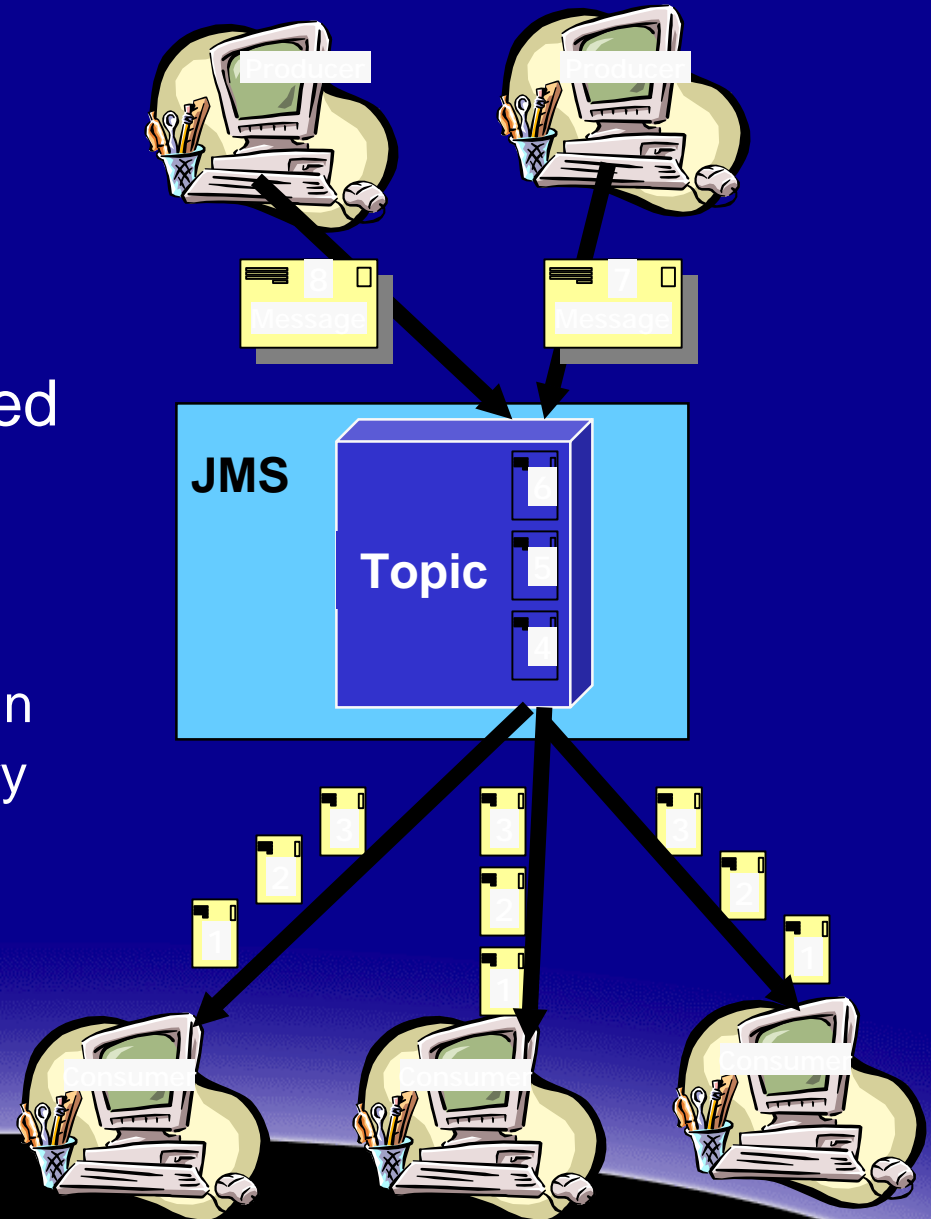
JDBC is:

- a standard Java interface for accessing heterogeneous databases
- a specification that defines four different driver types for connecting to databases



Java Message Service

- JMS is a Java API for accessing message-oriented middleware.
- The interface supports:
 - the Point-To-Point domain
 - the Publish/Subscribe domain
 - guaranteed message delivery
 - transactional sessions

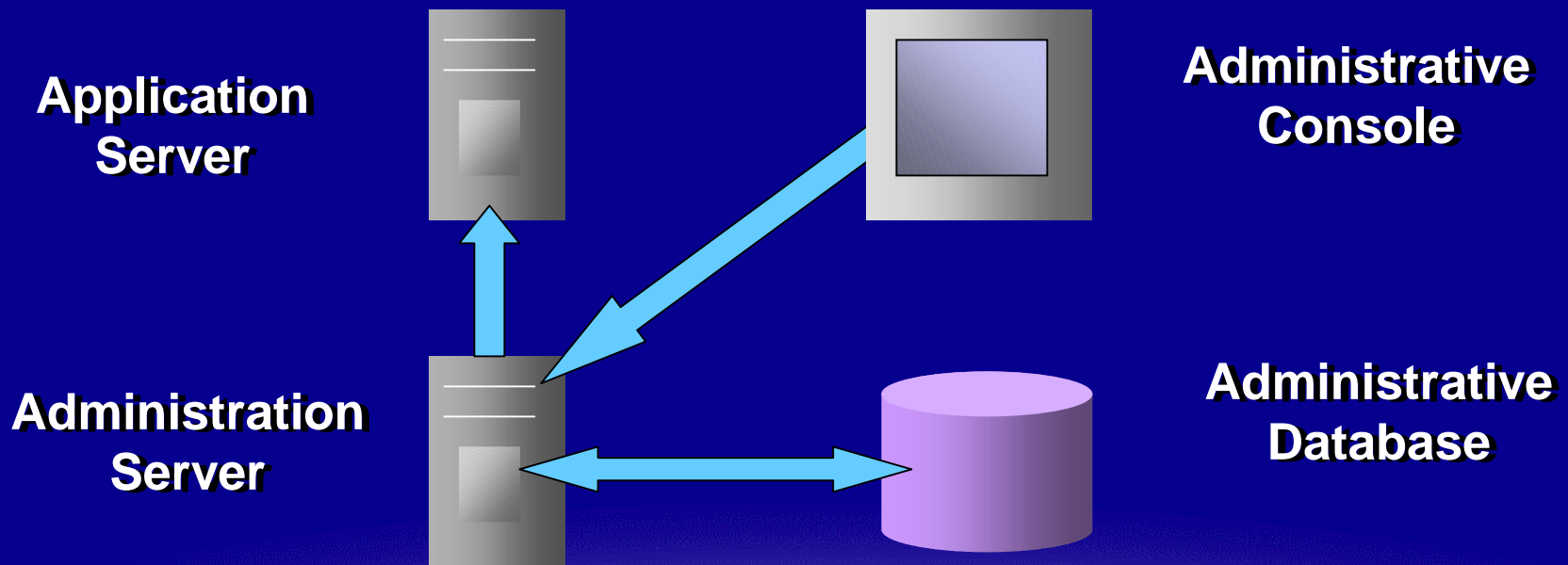


WebSphere Application Server: e-business Application Deployment Platform

- **WebSphere** is an Environment for open distributed computing
 - Built on open standards-based technology
 - Combines the portability of server-side business applications with Java^(TM) technologies
- **Scalable, robust**
 - Design and deploy portable Java-based Web applications
 - Enables interaction with enterprise databases and transaction systems

WebSphere Application Server Architecture Overview

WebSphere Components



WebSphere Architecture Overview

- Administration Server

- Responsible for runtime management, security, transaction coordination, workload management
- Runs on all nodes in a domain
- Controls interaction between each node

- Application Server

- Contains all code: servlets, JSPs, EJBs
- Can define multiple Application Servers
- Each App server has its own JVM (Java Virtual Machine)

WebSphere Architecture Overview

- Administrative Database

- Stores all runtime configuration information in a single persistent repository
- WebSphere supports DB2, Oracle, Sybase (WS Advanced Edition)

- Administrative Console

- Graphical user interface (GUI) for administration of a WebSphere domain
- Can run locally on one of the Administration Server nodes or on a remote node that attaches to a running Administration Server

Advanced Edition

Extends WebSphere functionality across multiple machines

- Complete support for high-performance, scalable Web-driven transactions
- Supports the development and use of EJBs
- Focuses on JSPs and EJBs that access relational databases for persistent state data
- Supports the replication of EJB server models - easy to clone servers across multiple nodes

Advanced Edition Services

- **Naming Service**
 - Uses JNDI to locate objects as if on the same machine
 - Can use different services behind JNDI:
 - Lightweight Directory Access Protocol (LDAP)
 - Domain Name System (DNS)
- **Security Service**
 - Security mechanisms include:
 - Access control to HTML, JSPs, EJBs and business methods
 - Permission-based authorization
 - Security Realms, SSL, Digital Certificates

Advanced Edition Services

- **Workload Management (WLM)**
 - Improves the scalability by grouping multiple EJB servers into EJB server groups or *clusters*
 - Allows the modeling of application server processes
 - *Clones* -instances of a model (app server) can be created on a single machine, or across multiple machines in a cluster
 - Provides workload distribution and failover
 - Ensures that the workload is evenly distributed across the EJB servers in the EJB server groups

WebSphere Application Server

Other Tools Included:

- **IBM DB2**--Distributed relational database, which can be used as a resource manager in conjunction with TXSeries and Component Broker
- **IBM HTTP Server**--A powerful Web server based on the popular Apache Server. Provides enhanced SSL for secure transactions.

IBM WebSphere Product Family

WebSphere Studio

- Workbench
- Wizards
- Content Authoring
- Site Development
- Content Management

VisualAge

- Application Programming
- Component Development
- Team Development

Standard Edition

- DB Connection Manager
- Java Servlet Run-time and services

Advanced Edition

- Clustering, scaling
- Multi-process Servlets, EJS/EJB
- DB Connection Manager
- Java Servlet Run-time and services

Enterprise Edition

- Composed Biz Components
- Transaction processing
- Clustering, scaling
- Multi-process Servlets, EJS/EJB
- DB Connection Manager
- Java Servlet Run-time and services

- Caching

EDGE Server

- Load Balancing

- Filtering

- Distributed Files

HTTP Server

NT AIX Solaris Linux

NetWare OS/2 OS/400 OS/390

T
I
V
O
L
I

Build

Run

Manage

Agenda:

- Welcome
- Overview of Requirements in e-Business Projects
- Overview of WebSphere Platform and the J2EE Architecture
-  **Planning and Execution Of A WebSphere / J2EE Project**
- **BREAK**
- **Continued Execution Of A WebSphere / J2EE Projects**
- **Managing A WebSphere Application In Production**
- **Closing Remarks – Next Steps**

Planning and Execution Of A WebSphere / J2EE Project

- Staff Transition Guidelines
- Project Management Guidelines
- Architecture Guidelines
- Software Development Process Guidelines
- Phases of an OO Project
- Reuse and Struts
- Java Unit Testing

Staff Transition Guidelines

- Long term enterprise-wide planning
- Clearly defined set of standards
 - organizational
 - operational
 - architectural
- Long term planning to minimize the impact of transition to the bottom line

Solution?

Just in time training and consulting methodology.

Staff Transition Guidelines - Training

- Assess current skill level
- Identify training needs and goals
- Create training roadmaps for each skill group
- Identify training resources
- Create timeline for achieving training goals

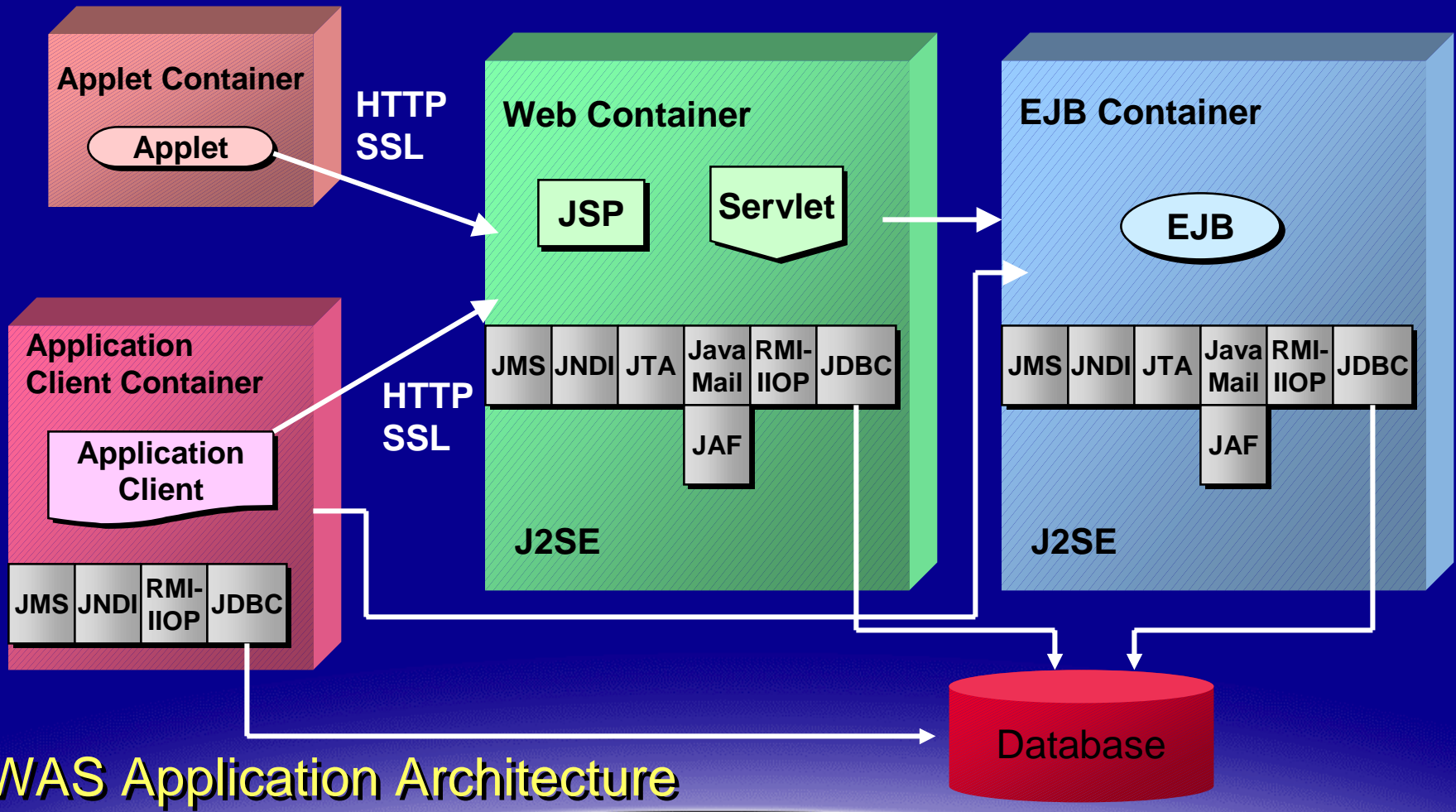
Project Management Guidelines

- Processes
- Requirements
- Quality Management
- Estimation
- Risk Management
- Reviews and Meetings

Architecture Guidelines: How Best To Use J2EE Features

- A well-defined Application Architecture provides the reuse, extensibility and scalability inherent in J2EE
- The MVC structural model provides clear separation of roles, separating the user interface (view) from the application components (model)
- Layering is essential with this model to promote reuse and easy-to maintain code

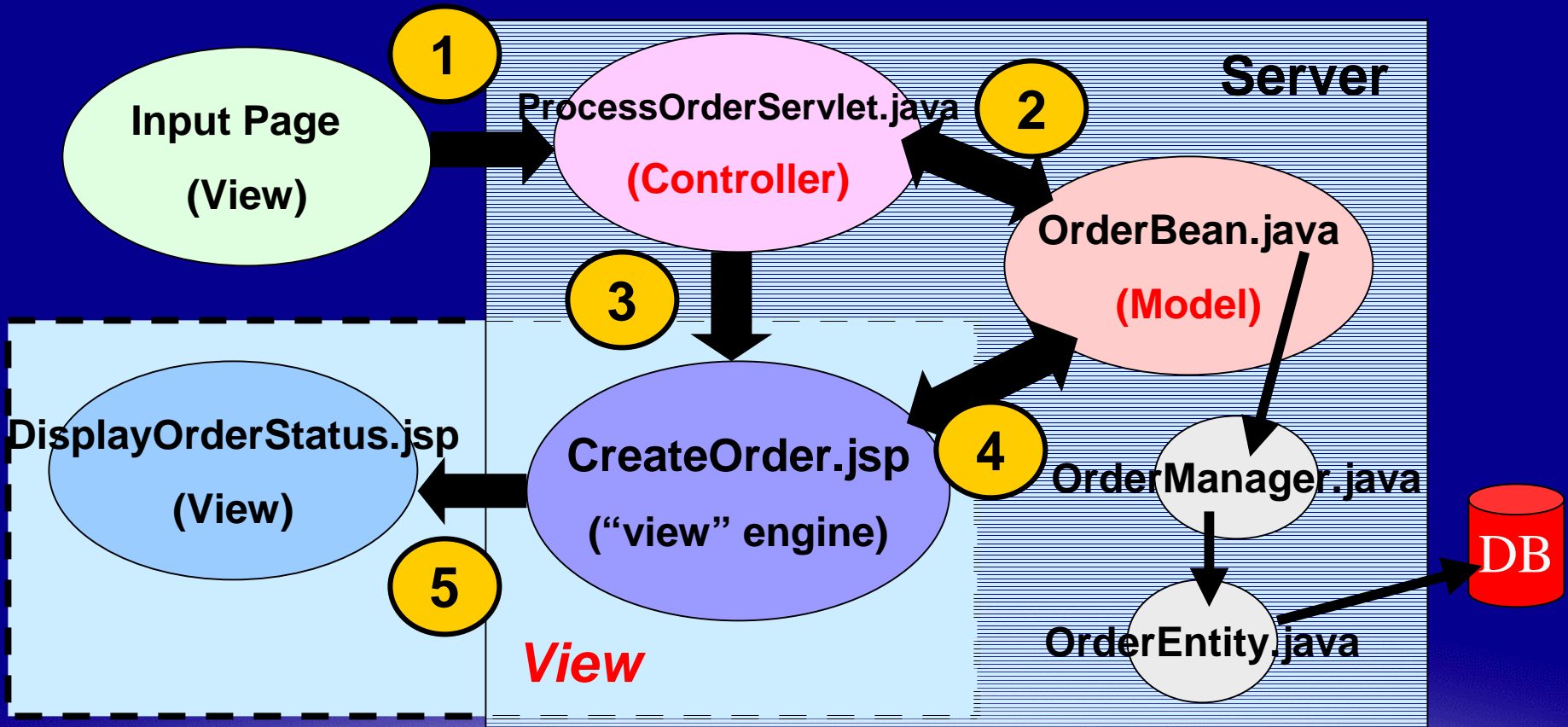
Architecture Guidelines: How Best To Use J2EE Features



WAS Application Architecture

Architecture Guidelines: How Best To Use J2EE Features

WebSphere MVC Programming Model



Web Tier (Supplier)

View - CreateOrder.jsp

- Displays products
- Allows customer to submit an order

Controller - ProcessOrderServlet.java

- Receives order - creates Java model object
- Calls OrderManager session bean to submit the order
- Receives Order confirmation from Session Bean
- Confirms to the customer that the order has been entered.

Model – Order.java

Web Tier (Supplier) continued

View – DisplayAllOrders.jsp

- Lists Order Numbers of all orders made by user
- Each Order Number is a hyperlink to execute DisplayOrderStatus.jsp

View - DisplayOrderStatus.jsp

- Includes QueryBean Custom Tag for JDBC query on the database to fetch the order status and other required information

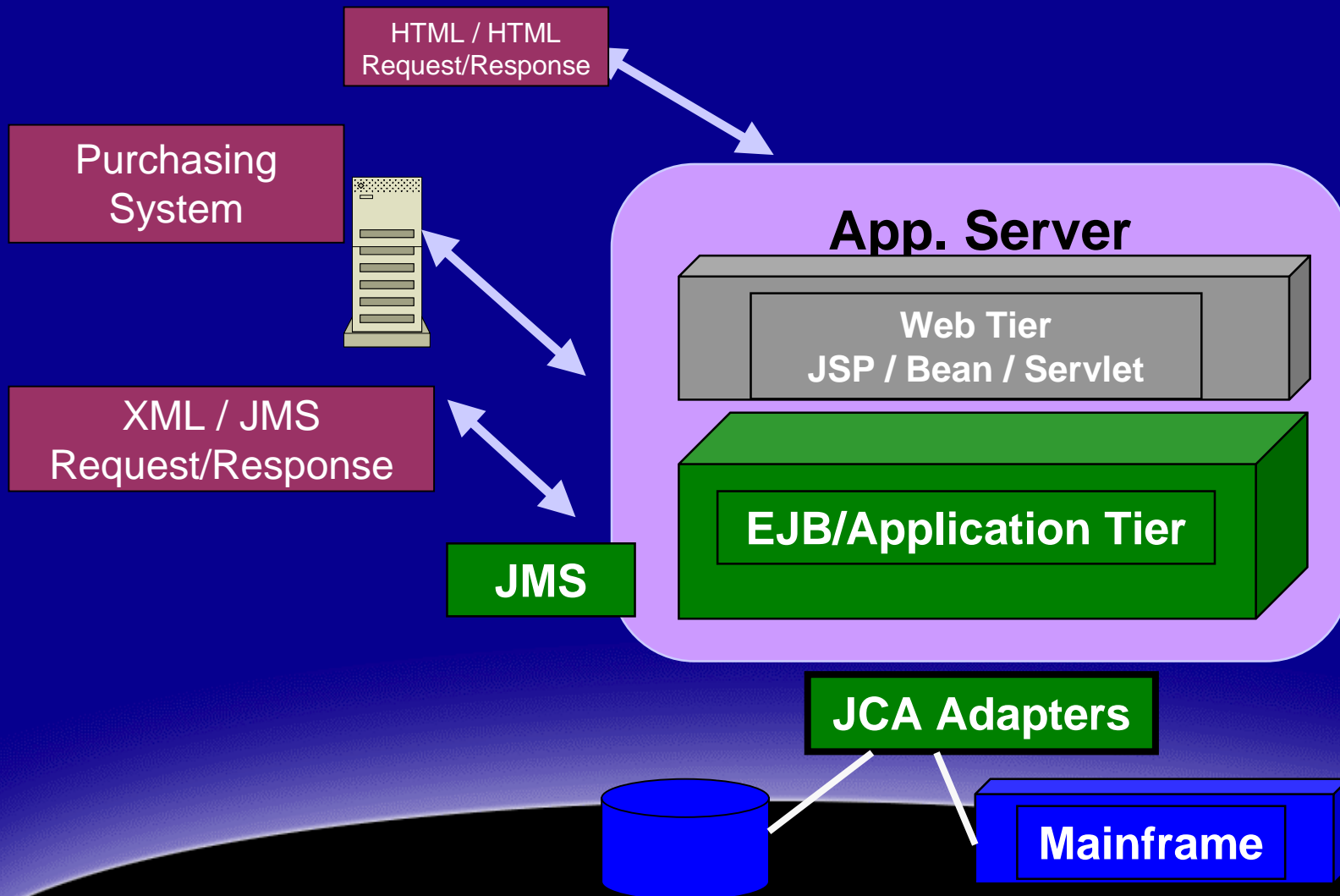
Model – OrderStatus.java

- Display object - contains only the fields of an order required for the DisplayOrderStatus page

Application (EJB) Tier

- OrderManager – session bean
 - Handles creation of orders
 - Returns the delivery status of an order
- MessageManager – session bean
 - Handles sending and receiving messages and translation between Java Objects and XML messages
- OrderEntity bean
 - Supports transactional creation of new order in database.

WebSphere Application Server and J2EE Integration



Software Development Process Guidelines

Modified Rational Unified Process (RUP)

Architecture-centric

Flexible

Layered

De-coupled layers

Technology neutral

Software Development Process Guidelines

Modified Rational Unified Process...

- Model-based-UML
- Strong Requirements analysis
- Strong Use Case analysis
- Early involvement of Business Customer
- Early testing based on Use Cases
 - Iterative -regular incremental iterations
 - Early testing
 - *NEVER* wait until the end
- Early aggressive Risk Management – Risk Lists and Risk Models

Software Development Process Guidelines

Best of XP

- Iterative detailed design
- Continual unit testing during and before coding
- Weekly system tests
- Automated unit tests – Junit, JUnitPerf, Cactus (formerly J2EEUnit)
- Automated builds and system tests run nightly- ANT
- Direct involvement of business customer

Software Development Process Guidelines

WebSphere Development Process

- Developer daily test of VAJ WebSphere Test Environment
- Weekly tests on WebSphere Development Environment
- At least once a month – full stress testing on WebSphere production Environment
 - No surprises
 - Firewalls
 - Policy director
 - Full clone and cluster simulations

Code Management

1. Shared Application Code Repository (VisualAge Team Repository)
 - Defined Project/Package structure for maintainability
 - Defined ownership at Project, Package and Class level for better security, source code control
 - Defined version control process like defined version stamps (description-date-owner) at Project, Package and Class level, Structured Comments for better build management

Configuration Management

2. Externalize all environment-specific implementations outside the code into flat files like stage-specific properties
3. Implementation of SCM Tool (PVCS/ Rational Clear Case) for build management
 - Static content – html, jpeg, gif, etc.
 - JSP Pages
 - Properties Files
4. Defined Structure for build containing all the build components integrated for different stages

Build Management

5. Maintain versions of the Application database in relation to build version
6. Application Server Configurations should be implemented as a build component and should be SCM controlled
 - Stage-specific – Virtual Host, Data Source, etc.
 - Application-specific – EJBContainer, ServletEngine, WebApp
7. Defined path for code migration – Development Stage to Testing and Quality Assurance Stage to Production

Phases of an OO Project

- Pre-Inception
- Inception
- Elaboration
- Construction
- Transition

Project Phases

Inception

- Identify Actors and Use Cases
- Document Use Cases
- Perform Risk Assessment
- Update Project Estimates

Elaboration

- Begin a Software Requirements Specification (SRS Create Use Case Reports
- Find and Build reusable objects
- Begin development of Object Model (UML Diagrams)
- Begin development of the Data Model

Construction

- Identify Remaining Requirements
- Develop and test the software
- Complete the data model
- Update estimates
- Update risk assessment

Transition

- Transition product to user community
- Monitor operation
- Develop new releases
- Correct problems
- Convert and migrate databases
- Train users and operators
- Finalize user documentation

Elaboration and Construction

- J2EE Implementation with WebSphere takes advantage of the WebSphere architecture
- The design uses the three major WebSphere Services:
 - Security
 - Naming
 - Workload Management
- Don't re- implement services that WebSphere already does well

Security Service

- Applications are moving toward a single-sign-on security model
- WebSphere implements it's own authentication
- Security implementation should be handled by a wrapper and accessed by all applications

Naming Service

- WebSphere Implementation of JNDI
- Use WebSphere name-caching and provide a wrapper for all applications to use
- Data Source lookup
- EJB Home Interface lookup

Transition

- Deployment To System Test Environment
- Testing
- Deployment To Production Environment
- Tuning
- Maintenance
- Product Improvement - begins inception of the next cycle

WebSphere Deployment Preparation and Testing

- WebSphere Load testing and tuning is iterative, just as all parts of RUP!
- Iterative cycle
 - Deploy applications to WebSphere
 - Monitor WebSphere – Garbage collection, Objects, Container behavior, Threads, heap, Logs, etc.
 - Run load tests
 - Tune WebSphere and Application
 - Repeat iterations

WebSphere Tuning

- The four parts
 - Application tuning (Java Code)
 - Application Server tuning (WebSphere)
 - Operating System tuning
 - Data Base tuning
- Tuning Tools
 - Log Analyzer (WebSphere)
 - Resource Analyzer (WebSphere)
 - OS and DB platform specific tools

RUP and WebSphere

- Note that most of Inception and Elaboration are technology-independent models
- These object and component models can apply to CORBA or J2EE models (or WebServices)
- WebSphere technology needs to be considered in these phases
- WebSphere **MUST** be considered in final Elaboration and throughout Construction and Transition

Reuse

What is reuse?

- To use “something” again for a reason other than what it was originally developed or created for
- To “not have to reinvent the wheel”
- A well-designed component with a single conceptual idea and well-defined set of public interfaces
- Can be reused without modifications

Reuse

What can be reused?

- WebSphere component groupings or frameworks
- Procedures and methods
- Templates
- Modeling
- Code snippets
- Project plans
- Project estimates, actual results
- Business model

Reuse

Why is reuse important?

- Quality applications through the use of proven components
- Shorten time to market through faster development
- Standardization from an enterprise business model
- Reduced development costs

Creation of a new reusable component involves 9 steps.

1. Gathering requirements

2. Identifying reusability

3. Determining commonality

4. Updating requirements

5. Designing and scheduling

6. Developing component

7. Verifying component

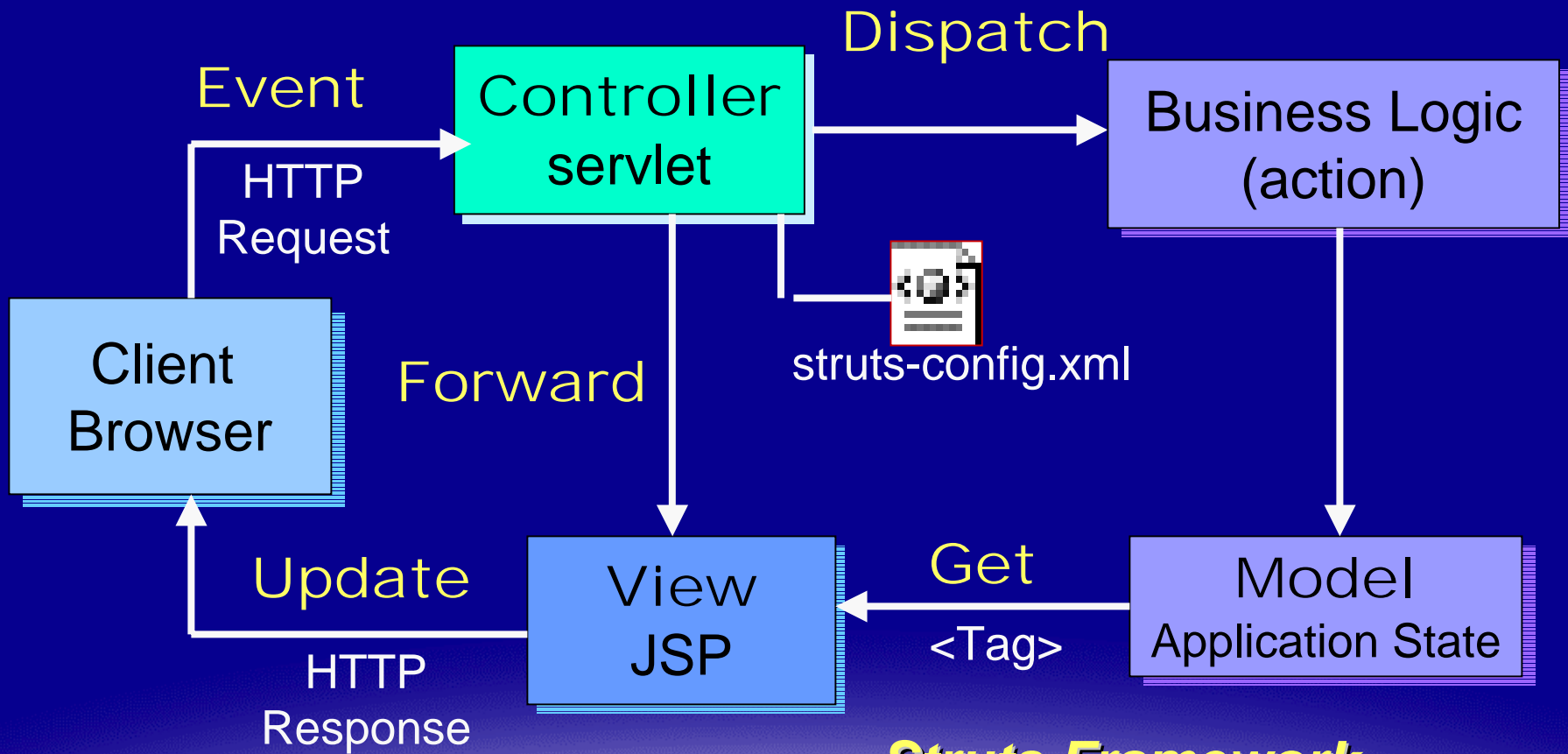
8. Component Available

9. Supporting Component

Struts

- An open source framework for building web applications with Java servlets and JSPs
- Based on the Model-View-Controller (MVC) design paradigm
- Uses powerful Custom Tag Library

Struts



Struts Framework

Struts - How It Works

- **Client browser**

An HTTP request from the client browser creates an event- the Web container responds with an HTTP response

- **Controller**

A command design pattern implemented as a servlet - The controller receives the request from the browser, and makes the decision where to send the request

- **Business logic**

The business logic updates the state of the model and helps control the flow of the application

- **Model state** - business objects update the application state

Action classes (*a session or request*) represent the Model state - the JSP file reads information from the Action classes using JSP tags

- **View** - The view is simply a JSP file. Tags are one of the things that make Struts unique

Java Unit Testing

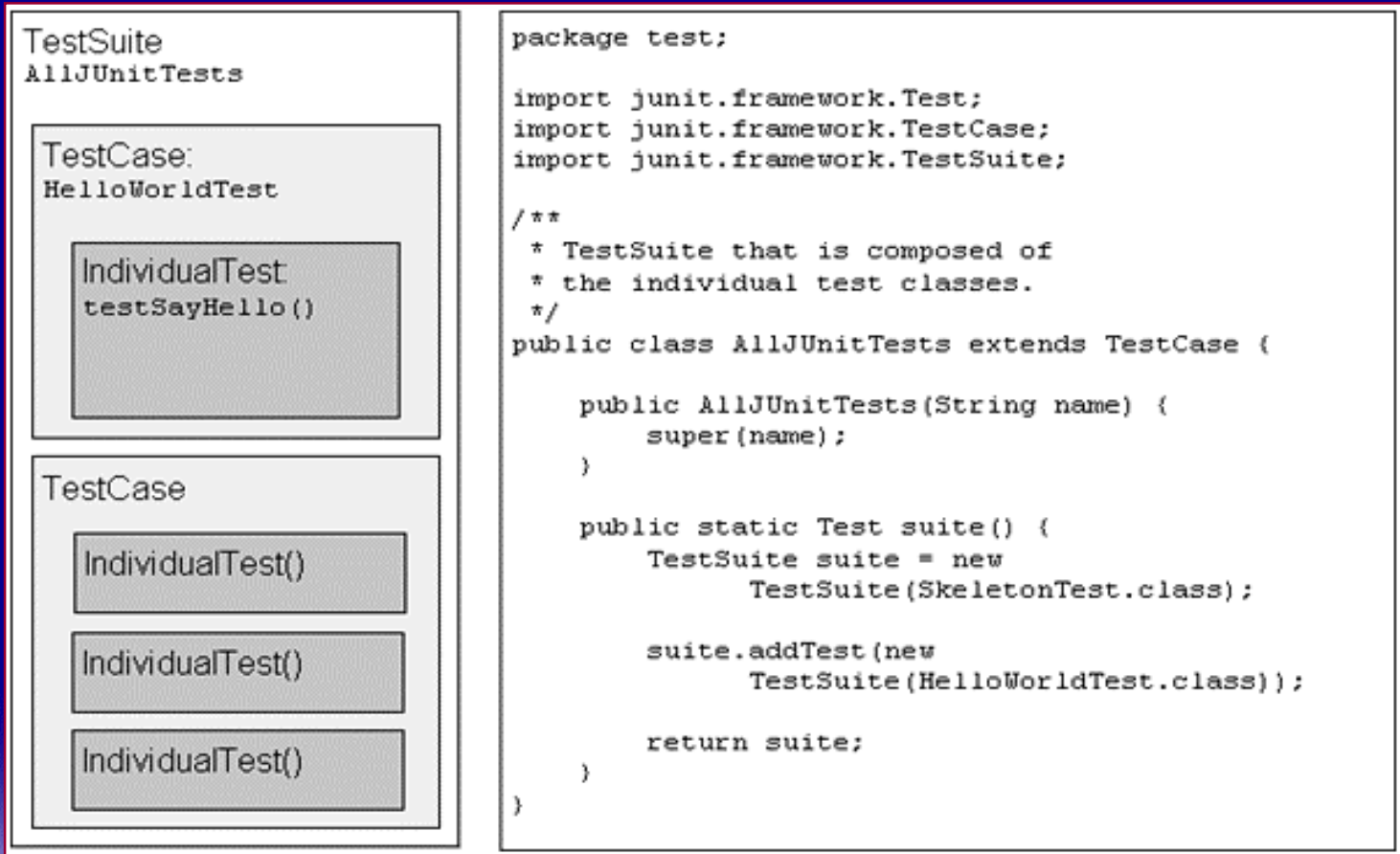
- Continual unit testing is a key to rapid and effective development
- Must be done regularly – even daily
- Extreme Programming (XP) has promoted the use of ongoing testing to keep projects on track with continual feedback.
- Code then has months of accumulated testing - tests are designed along with the code

J Unit (JUnit)

- **JUnit is an open-source tool that provides a test framework for automated testing and validation**
 - **Allows separation of test code from product code**
 - **Easy to integrate into build process**
 - **Spend less time debugging- verify that changes to code actually work**
 - **Designed around two key design patterns: the *Command* pattern and the *Composite* pattern**

J Unit (JUnit)

TestSuite Layout



Cactus (formerly J2EEUnit)

- **An extension to the JUnit testing framework for unit testing server-side java code**
- **Uses Redirector Proxies to launch tests that are on the server side**
 - ***A Servlet Redirector*** for unit testing servlet methods
 - ***A JSP Redirector***

JUnitPerf

- **JUnitPerf is a collection of JUnit *test decorators* to test the performance of existing JUnit tests:**
 - **TimedTest**
 - **ThreadedTest**
 - **LoadTest**
- **Not intended to be a full-fledged load testing or performance profiling tool or to replace the use of these tools**
- **Should be used to write localized performance unit tests to help developers refactor responsibly**

ANT

- **Ant is becoming the de facto standard for implementing the build process**
- **Written in java – allows multiple platforms**
 - **Java classes are used to extend build features instead of using shell-based commands**
- **Uses an XML file for configuration of the build process**
 - **learning how to configure a build is easier**

Break

- Have a cup of coffee
- Have a Danish
- Talk to us about how you plan to use WebSphere and J2EE

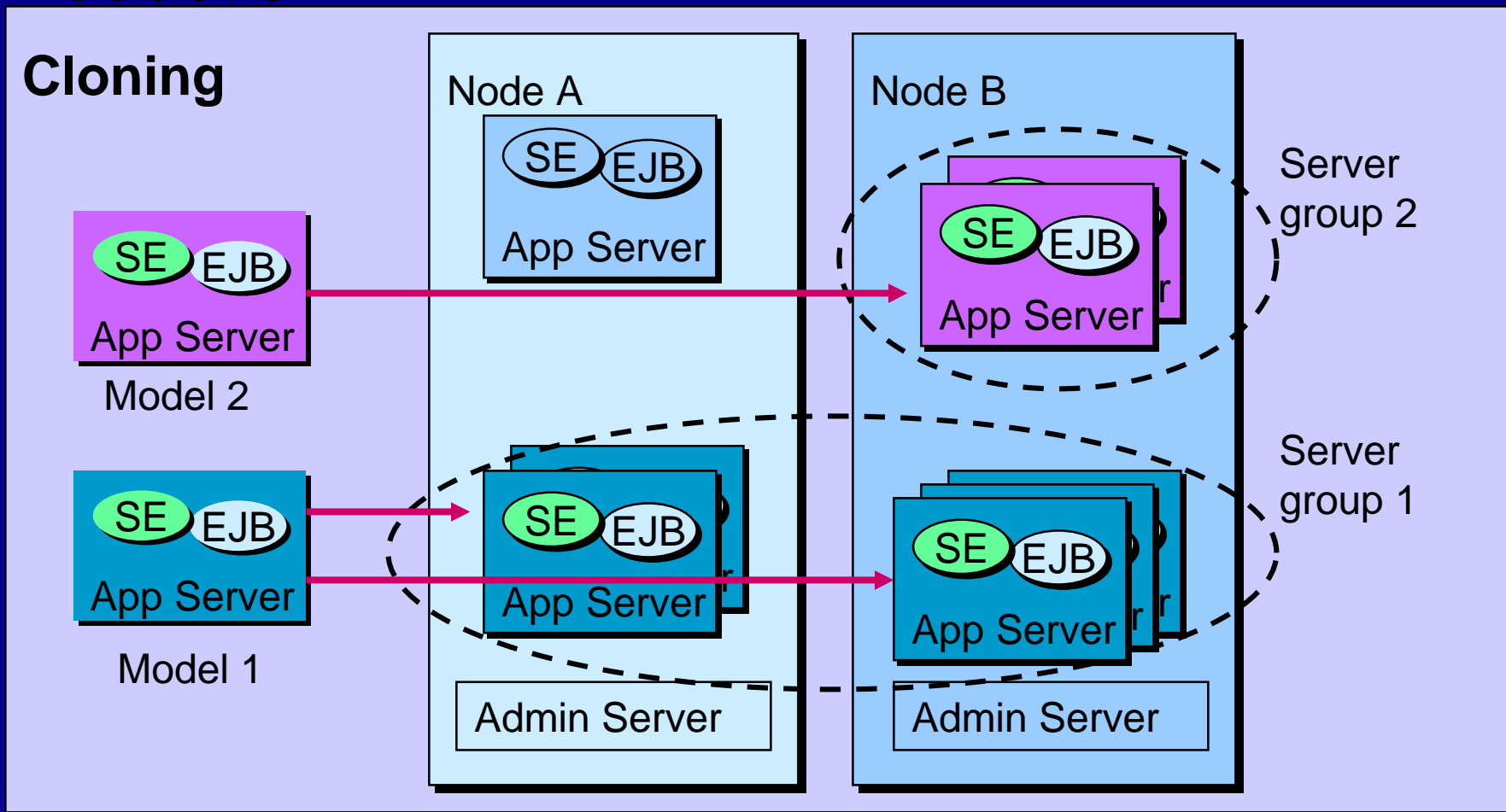
Agenda:

- Welcome
- Overview of Requirements in e-Business Projects
- Overview of WebSphere Platform and the J2EE Architecture
- Planning and Execution Of A WebSphere / J2EE Project
- BREAK
- Continued Execution Of A WebSphere / J2EE Projects
- **Managing A WebSphere Application In Production**
- Closing Remarks – Next Steps

Managing A WebSphere Application In Production

- **Cloning** allows for the creation of multiple copies of an object such as an application server
- Take a server that has been set up - create a model based on that setup - then create clones of that server

Managing A WebSphere Application In Production



Managing A WebSphere Application In Production

Two important benefits of cloning:

- Simplifies system administration - clones can be used to quickly create and maintain identical copies of a server configuration
- Organizes workload distribution - since by definition all the clones are identical, the workload distribution mechanisms can safely assume that any one of the clones is equally capable of servicing any request

Managing A WebSphere Application In Production

- **Vertical cloning** refers to the practice of defining multiple clones of an application server on the *same* physical machine
- A single JVM process cannot always fully utilize the CPU power of a large machine
- Vertical cloning provides a mechanism to create multiple JVM processes, which together can fully utilize all the processing power available

Managing A WebSphere Application In Production

- **Horizontal cloning** refers to the more traditional practice of defining clones of an app server on *multiple physical machines*
- Allows a single WebSphere application to span several machines while presenting a single system image (as though it were a single machine)
- Horizontal cloning provides both increased throughput and failover

Managing A WebSphere Application In Production

- The set of all the clones of one model of an application server constitutes a logical group called a server group or ***cluster***

Managing A WebSphere Application In Production

- The WAS XML Config Tool offers an alternative method for configuring and administering WebSphere
- XML Config is a time-saver for deploying multiple EJBs and servlets - avoiding lengthy menu-driven configuration

Managing A WebSphere Application In Production

Other uses of XML Config:

- Import or export complete or partial configuration into a node
- Create or restore a configuration backup
- Copy a configuration between nodes
- Move EJBs and servlets from one node to another

Managing A WebSphere Application In Production

- Messages, logs, and traces are important diagnostic tools for investigating the behavior of an application server
- They gather crucial information that helps facilitate the problem determination process
- Next - provide an overview of the these three mechanisms

Key Steps In Problem Determination

1. Validate the installation. (performed correctly?)
2. What is the topology? (platforms, number of servers, database, Web server, and setup)
3. What is the application doing? (app structure, components)
4. Develop simple test case to reproduce the problem.
5. Collect server logs and appropriate trace.

Key Steps In Problem Determination

6. Collect server configuration information.
7. Analyze logs, trace, and the test case.
8. Isolate to appropriate product component.
9. Collect additional, more targeted, traces as necessary.
10. Identify and correct the problem.

Avoiding Pitfalls in WebSphere / J2EE Projects

Avoiding the security risks of invoking servlets by class name

- WebSphere security is based on defining, and then securing, URIs (known as Web resources) for servlets.
- This allows an administrator to apply different security levels to different paths for accessing the same servlet.
- Now, suppose the WebSphere administrator drops in a servlet class to be invoked by its class name. Even if it is protected, users will be able to invoke the servlet by classname without any security checks.

Avoiding Pitfalls in WebSphere / J2EE Projects

To protect each servlet, the administrator needs to:

- Configure a Web resource based on the servlet class name
- Add the Web resource to the Web Path list of the Invoker servlet in the Web application
- Use the Configure Resource Security wizard to secure the Web resource.
- Secure the Invoker servlet itself.

Agenda:

- Welcome
- Overview of Requirements in e-Business Projects
- Overview of WebSphere Platform and the J2EE Architecture
- Planning and Execution Of A WebSphere / J2EE Project
- BREAK
- **Continued Execution Of A WebSphere / J2EE Projects**
- Managing A WebSphere Application In Production
- **Closing Remarks – Next Steps**

Next Steps

- Schedule a Free Needs Assessment Meeting
 - Determine E-Business and WebSphere Goals and Objectives
 - Assess High-level Organizational Skill Base
 - Discuss options available to your organization relative to both Technology and People Strategy

Services offered by Noospherics and DPT Consulting

- Technology Services
 - Application Architecture Assessment
 - Consulting – WebSphere Developers / Administrators
 - Project Outsourcing
- People Services
 - Organizational People Strategy Planning
 - Reskilling strategy
 - Mentoring vs. Consulting
 - Just in Time Training tied to Project Deliverables
 - J2EE, WebSphere and MQSeries Training