

| IBM Software Group

WebSphere Application Server v6.1: technical overview

26 Febbraio 2008, Università di Bologna: Seminario IBM



ON DEMAND BUSINESS™

Antonella Bertoletti
Senior Consulting I/T Specialist
WebSphere Technical Sales Team
IBM Software Group

© 2005 IBM Corporation

WAS family overview

Extended Deployment (XD)

On demand operating environment

- Operation Optimization
- Compute Grid
- Object Grid

Network Deployment

Application Server

Express

- Full J2EE 1.4
- Web based admin
- Messaging
- Web/EJB containers
- JDK
- PMEs

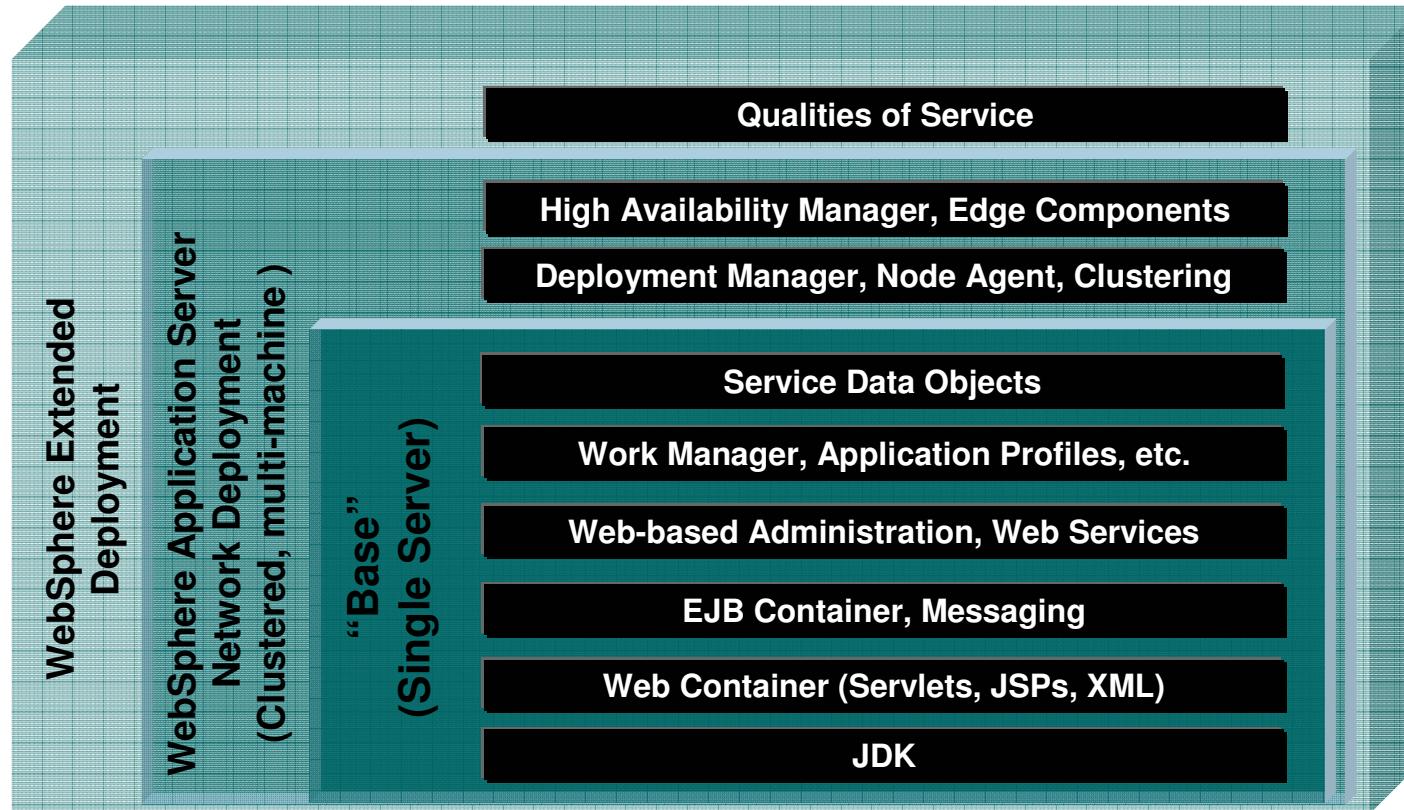
Like Express
without license
limitation

- Clustering
- Failover
- Workload Management
- Distributed Administration
- Web Services
 - Web Services Gateway



WebSphere Application Server v6.1 Basic Concepts

Packaging and Install (1/2)



Packaging and Install (2/2)

- Major distributed platforms supported are:
 - ▶ Windows® 2000, 2003 Server
 - ▶ Linux®/Intel®, Linux/PPC™, zLinux¹
 - ▶ Unix® platform: AIX®, Solaris™, HP-UX®
- Other supported platforms:
 - ▶ i5/OS™ and OS/400® on iSeries® platform
 - ▶ z/OS® ¹

¹ Not supported on WebSphere Application Server v6 - Express

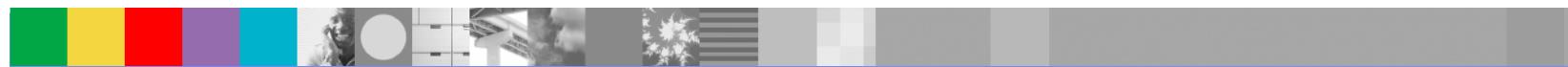
Web Services evolution

WAS V5.0.2/5.1	WAS V6.0	WAS V6.1
<p>JAX-RPC (JSR-101) 1.0</p> <ul style="list-style-type: none">▪ New standard API for programming Web services in Java <p>JSR-109 1.0</p> <ul style="list-style-type: none">▪ New J2EE deployment model for Java Web services <p>SAAJ 1.1</p> <p>WS-Security</p> <ul style="list-style-type: none">▪ Extensions added <p>WS-I Basic Profile 1.0</p> <ul style="list-style-type: none">▪ Profile compliance <p>UDDI4J version 2.0 (client)</p> <p>Apache Soap 2.3 enhancements</p> <p>The engine is a new high performance SOAP engine supporting both HTTP and JMS</p>	<p>JAX-RPC (JSR-101) 1.1</p> <ul style="list-style-type: none">▪ Additional type support▪ xsd:list▪ Fault support▪ Name collision rules▪ New APIs for creating Services▪ isUserInRole() <p>JSR-109 – WSEE 1.1</p> <ul style="list-style-type: none">▪ Moved to J2EE 1.4 schema types▪ Migration of web services client DD moving to appropriate container DDs▪ Handlers support for EJBs▪ Service endpoint interface (SEI) is a peer to LI/RI <p>SAAJ 1.2</p> <ul style="list-style-type: none">▪ APIs for manipulating SOAP XML messages <p>WS-Security</p> <ul style="list-style-type: none">▪ WSS 1.0 <p>WS-I Basic Profile 1.1</p> <ul style="list-style-type: none">▪ Attachments support <p>WS-TX AT (Atomic Transactions)</p> <p>JAXR support</p> <p>UDDI v3 support</p> <ul style="list-style-type: none">▪ Includes both the registry implementation and the client API library	<p>WS-BA (Business Activity)</p> <ul style="list-style-type: none">▪ Compensation framework for loosely coupled transactions <p>WS-I BSP (Basic Security Profile)</p> <ul style="list-style-type: none">▪ Interoperability over the wire (i.e. WebSphere client with .NET svr.)▪ Tightening of specification <p>WS-N (Notification)</p> <ul style="list-style-type: none">▪ Publish/Subscribe model <p>WS – Security enhancements</p> <p>Performance Enhancements</p> <ul style="list-style-type: none">▪ SAAJ changes (send XML docs. w/ attachments)▪ SOAP/JMS▪ New and faster parser (Banshee instead of B2B) <p>SOAP/JMS Enhancements</p> <ul style="list-style-type: none">▪ Caching enhancements<ul style="list-style-type: none">▪ Text message enhancements <p>WS-RF (Resource Framework)</p> <ul style="list-style-type: none">▪ Stateful web service resources <p>WS-Addressing</p> <ul style="list-style-type: none">▪ Endpoint ref. support for WS-Res.



Upgrade to Java SDK 5.0

- IBM's Virtual Machine for Java and JIT includes
 - ▶ Improved performance
 - ▶ Improved startup
 - ▶ Improved garbage collection
- No Sun intellectual property
- IBM's Virtual Machine for Java used on Windows®, Linux®, AIX®, i5/OS®, z/Linux® and z/OS®
- Sun's JVM used on Solaris and HP-UX



Portlet support

- WebSphere Application Server V6.1 includes support for running JSR168 compliant Portlets
 - ▶ Portlets container provides Portlet runtime environment and lifecycle management
 - ▶ Supports calling Portlets directory from a browser by URL, and from Servlet or JSP code
 - Example URL: <http://host:port/context/portlet-name/portletwindows>
- Does not include advanced capabilities of WebSphere Portal, such as Portlet aggregation, personalization, and collaboration



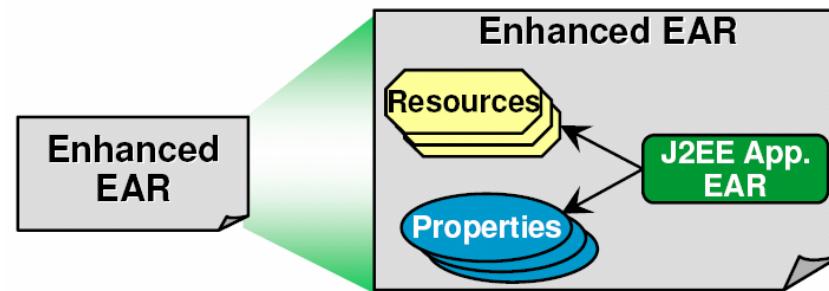
Enhanced Application Server ToolKit (ASTK)

- Provides WebSphere users with a set of basic Eclipse-based tools for assembling, deploying, debugging and profiling their WebSphere applications
 - ▶ No development support
- Deployment descriptor editors including WAS extensions (including PMEs that are now in the base) and bindings
 - ▶ Module/EAR creation and editing
- WebSphere Rapid Deployment Capabilities
 - ▶ Debug and Trace/Profiling Tools
 - ▶ Server Tools
 - Configuration Validation
 - Automated Table and Datasource creation
 - WebSphere Test Environment
 - ▶ Universal Test Client
 - ▶ EJB/Web Services Deployment Tools



Enhanced EAR

- EAR file that contain most of the application information needed to install in the Application Server
 - ▶ J2EE EAR, Deployment information and some application resources (JDBC) and properties (like setting class loader policy)



- Enhanced EAR support integrated with Rational Application Developer, Application Server Toolkit and WebSphere Application Server v6
- Benefits: Improved productivity
 - ▶ Application resources/properties come with the application
 - ▶ Application install process create the necessary resources within the server or cluster
 - ▶ Moving application from one server to another also moves the resources

WebSphere Configuration Archives

- Basically the same as a regular WebSphere Application Server configuration, with two main differences:
 - ▶ It may be a subset of a full configurations
 - ▶ Configuration information is virtualized to make it portable
 - Removes any specific information, like the host name
- WebSphere Configuration Archives are used to import/export configurations
- Allows simple creation of many servers with the same configuration



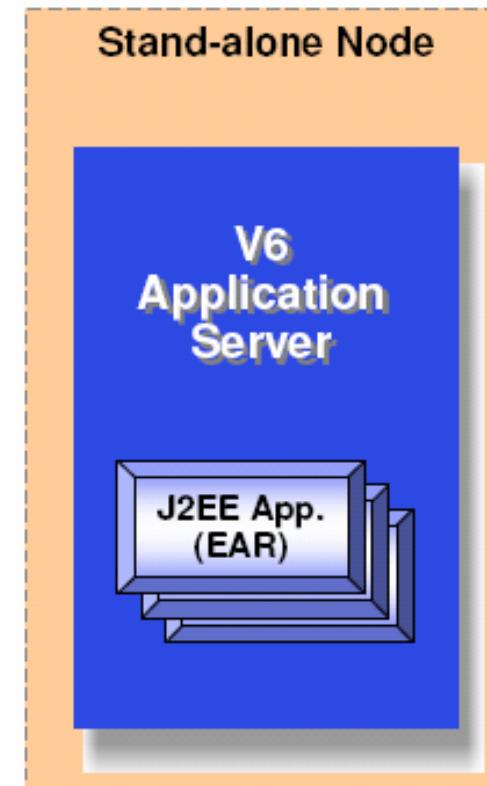
System Management

- Improved Administrative Console appearance and functionality
 - ▶ Console views change based on context
- Integrated Tivoli Performance Viewer
- Integrated IBM HTTP Server v6 management
- Support for extensible Server types
 - ▶ Web Server
 - ▶ Generic Server
- Console allows management of Web Servers:
 - ▶ Check the status of the Web Server
 - ▶ Generate the plug-in configuration file for that specific Web Server
 - No manual editing needed any longer
- In addition, if the Web Server runs on a managed node:
 - ▶ You can propagate the file to the node using the console
- Special treatment for the IBM HTTP Server (IHS)
 - ▶ Console allows stopping/starting server, editing httpd.conf file, displaying error.log, and propagating plug-in configuration even on an un-managed node



Terminology and Topology: Stand-alone Node

- Single node that runs one Application Server
- Independent of other nodes and other Application Servers
 - ▶ Cannot be managed by Deployment Manager

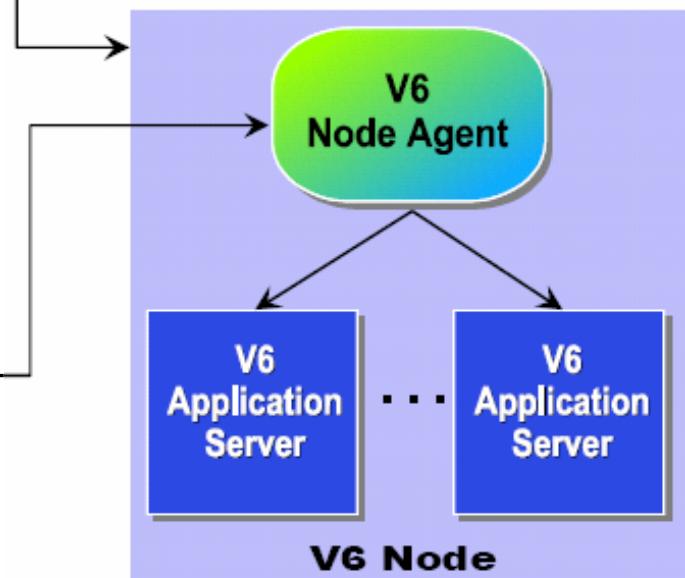


WebSphere Application Server Network Deployment



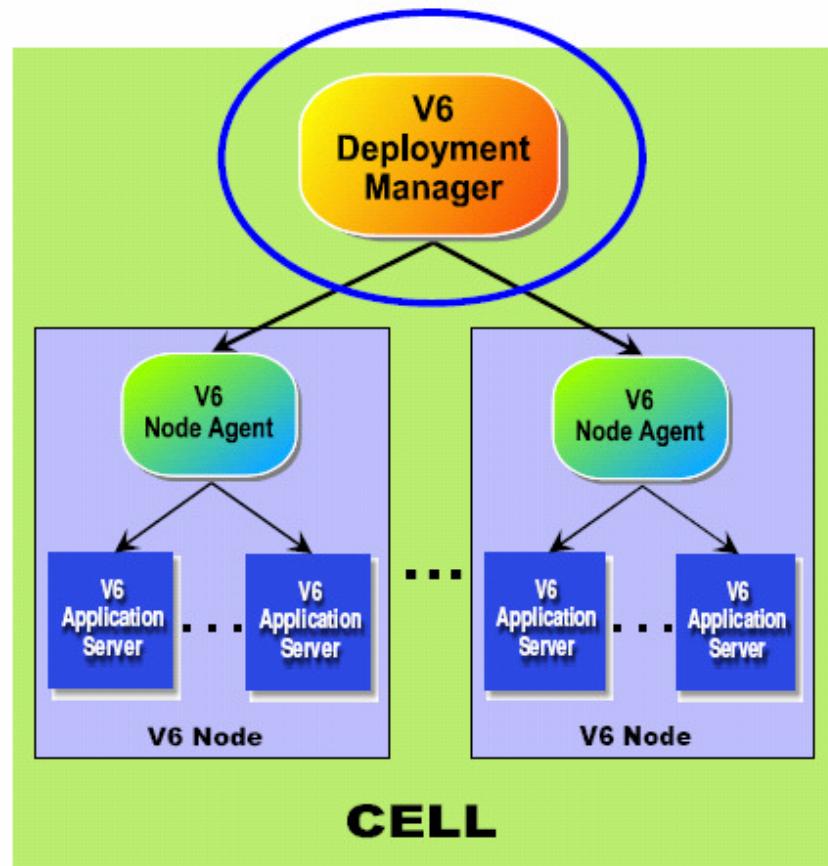
Terminology: Managed Node and Node Agent

- Managed Node
 - ▶ Grouping of Application Servers for configuration and operational management on a single machine
 - ▶ Contains one Node Agent
 - ▶ Can have multiple nodes on a single machine
- Node Agent
 - ▶ JVM process that manages the servers on a Node
 - ▶ Node Agents are created automatically when you add (federate) a Stand-alone Application Server node to a Cell

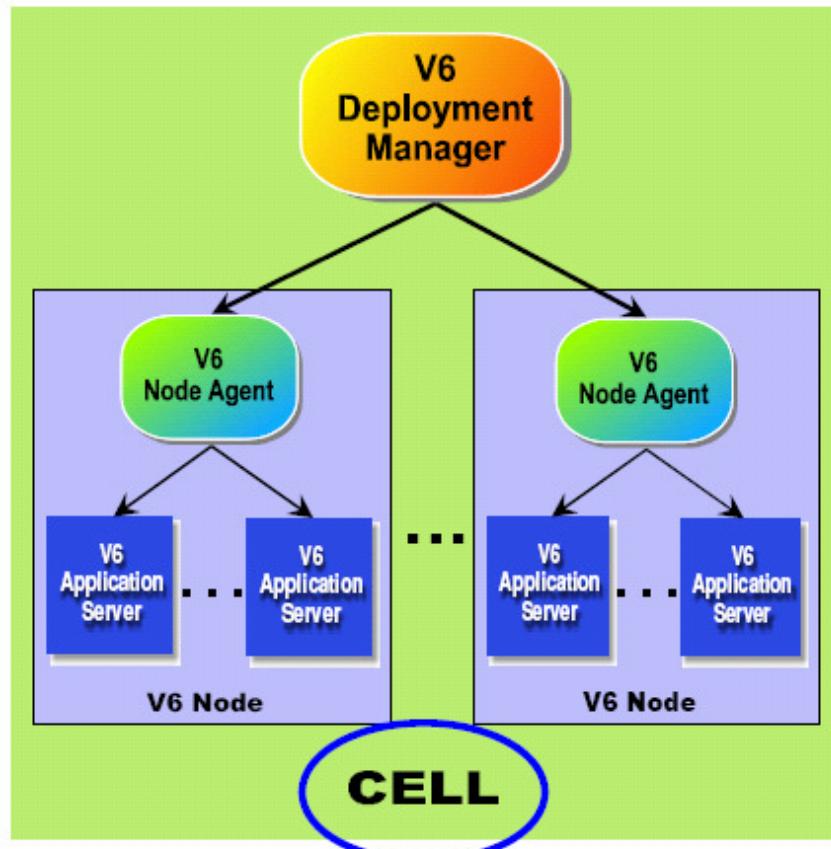


Terminology: Deployment Manager (DMgr)

- Manages one or more nodes in a distributed topology
- Communicates with the Node Agents for the configuration and operational management for that nodes



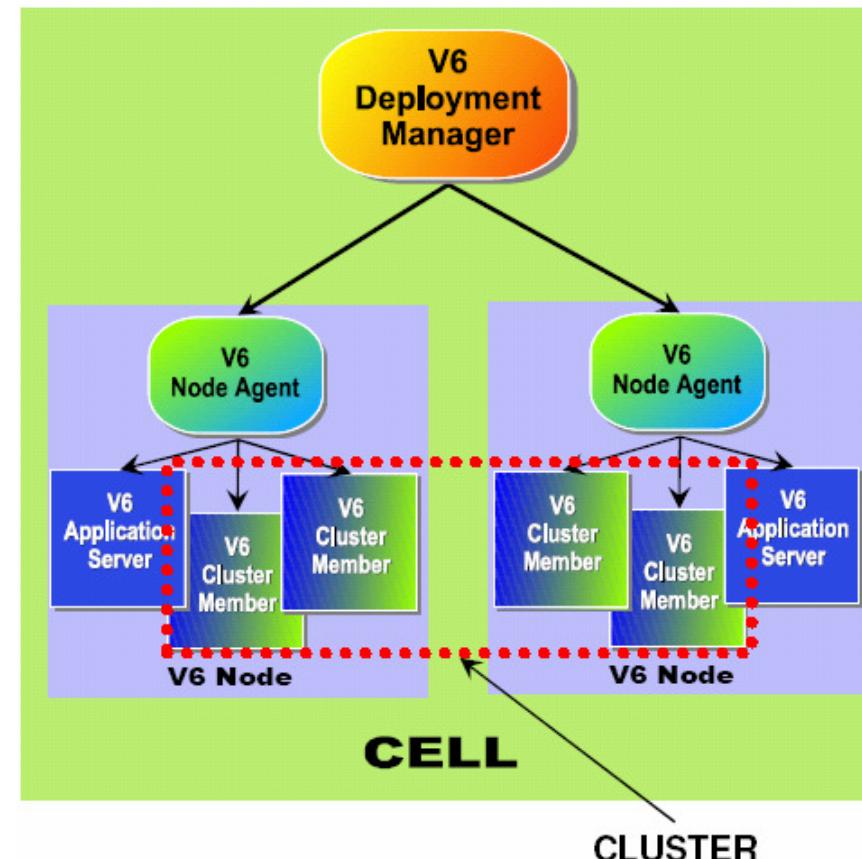
Terminology: Cell



- Network of multiple Nodes and one Deployment Manager over one or more machines
- Provides a single logical administration domain
- DMgr and the Nodes can be one the same or different machine

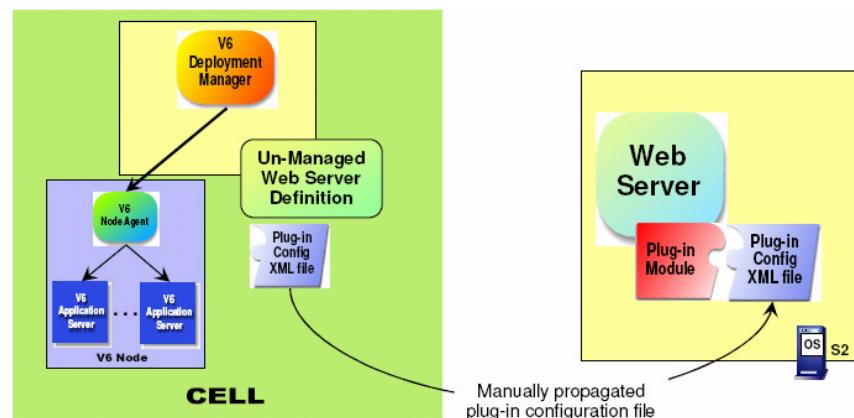
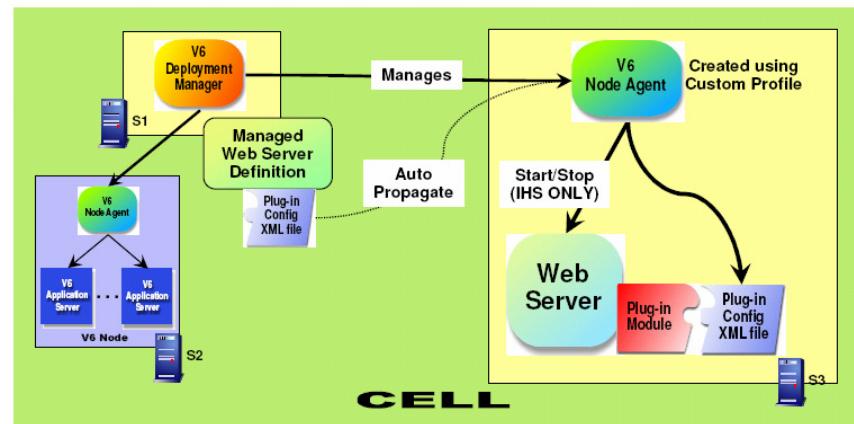
Application Server clusters

- A cluster is a grouping of application servers, called cluster members, running the same set of J2EE applications
- Cluster provides scalability and failover capabilities
- A Cell can have zero or more clusters
- Can span machine boundaries
 - ▶ Vertical – Same machine
 - ▶ Horizontal – Different machine
- Can span different OS types – like AIX and Windows
 - ▶ Exception – cluster members cannot span distributed and z/OS



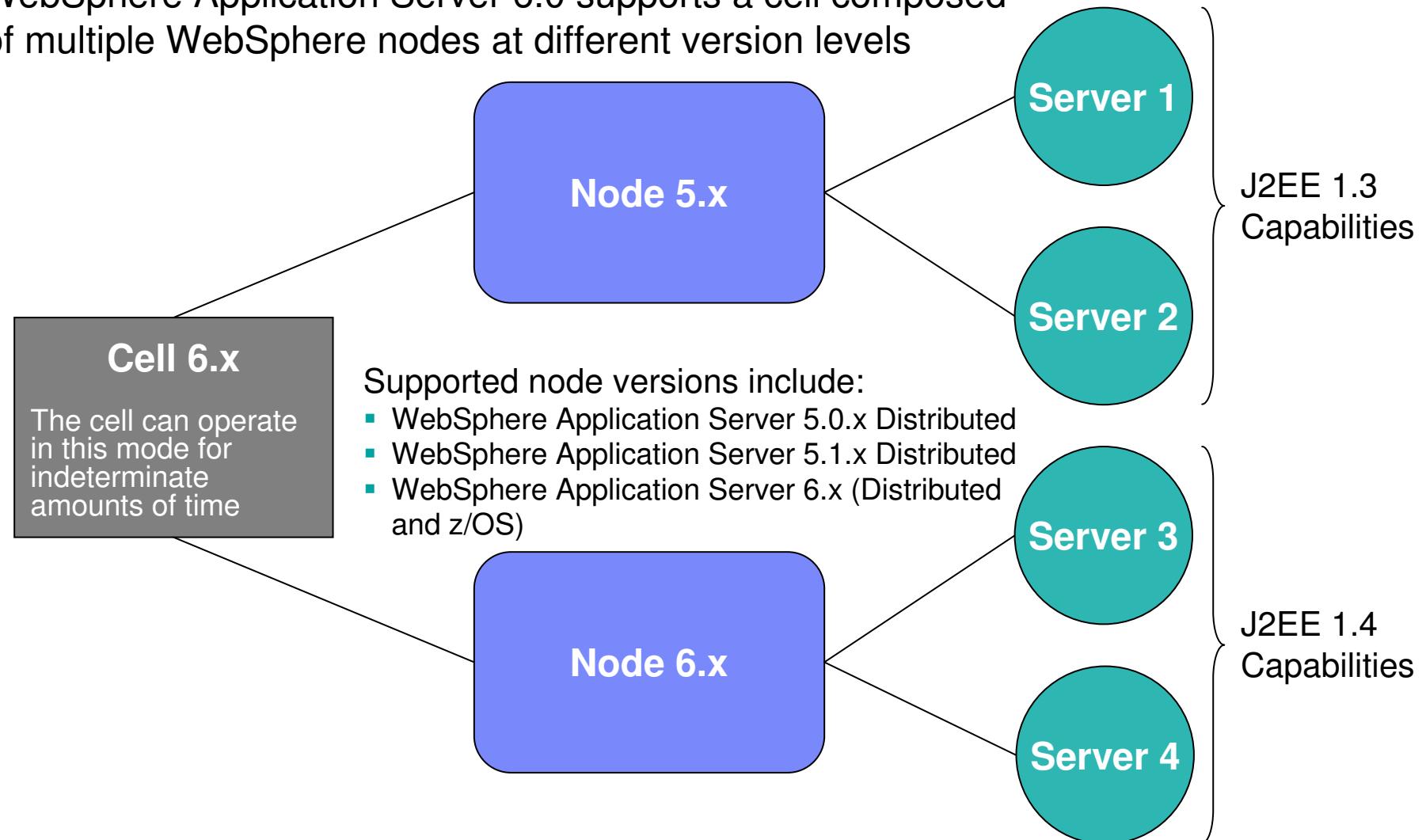
Web Servers

- Managed Web server:
 - ▶ Gives you the ability to start and stop the Web server from the WebSphere Application Server console
 - ▶ Automatically push the plug-in configuration file to the Web server
- Unmanaged Web servers
 - ▶ This is the only option in a stand-alone server environment
 - ▶ Common option for Web servers installed outside a firewall
 - ▶ Requires that each time the plug-in configuration file is regenerated, it is copied from the machine where WebSphere Application Server is installed to the machine where the Web server is running

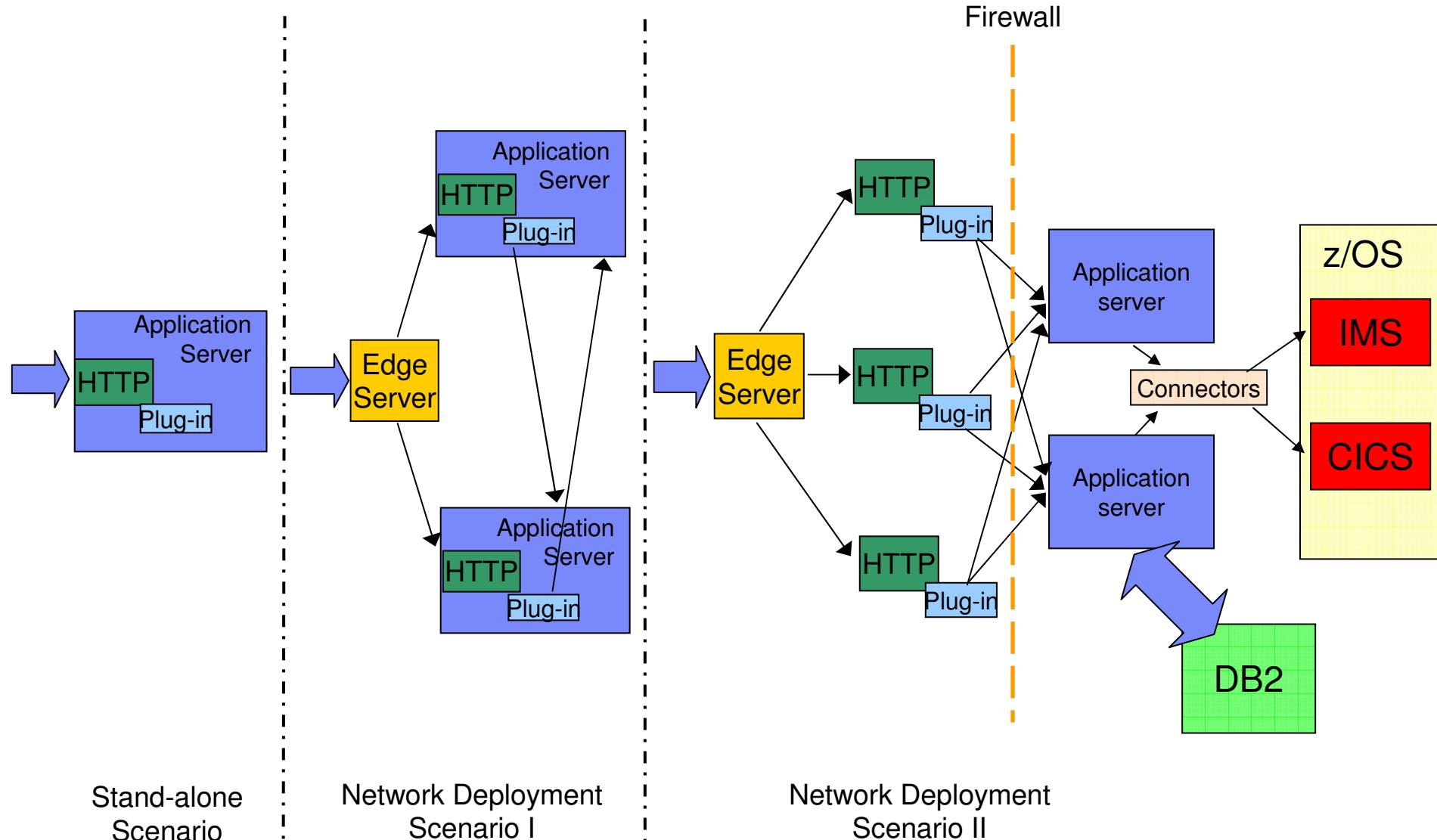


Mixed Version Cells

WebSphere Application Server 6.0 supports a cell composed of multiple WebSphere nodes at different version levels



Scalability and availability



Stand-alone
Scenario

Network Deployment
Scenario I

Network Deployment
Scenario II

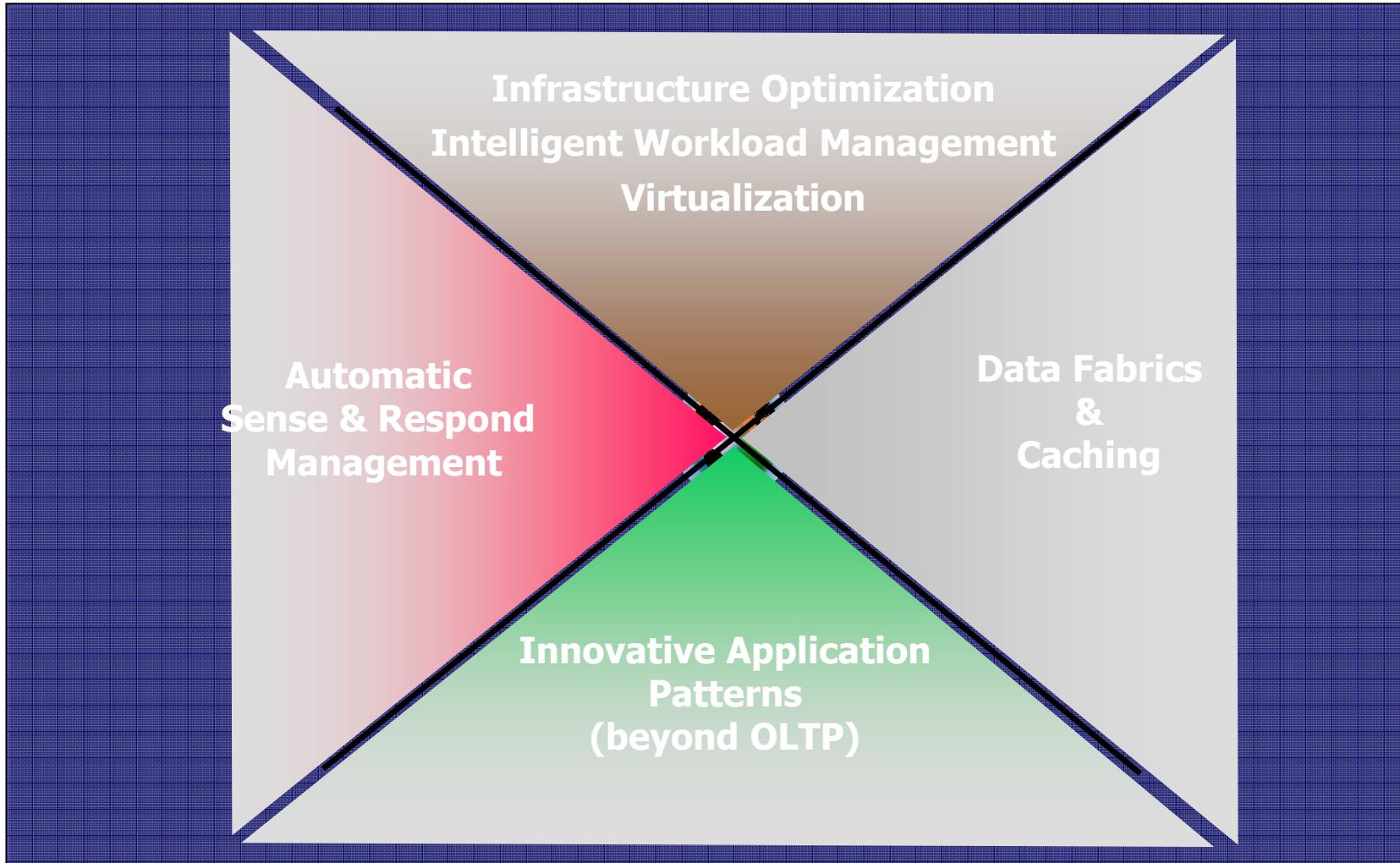


WebSphere Extended Deployment



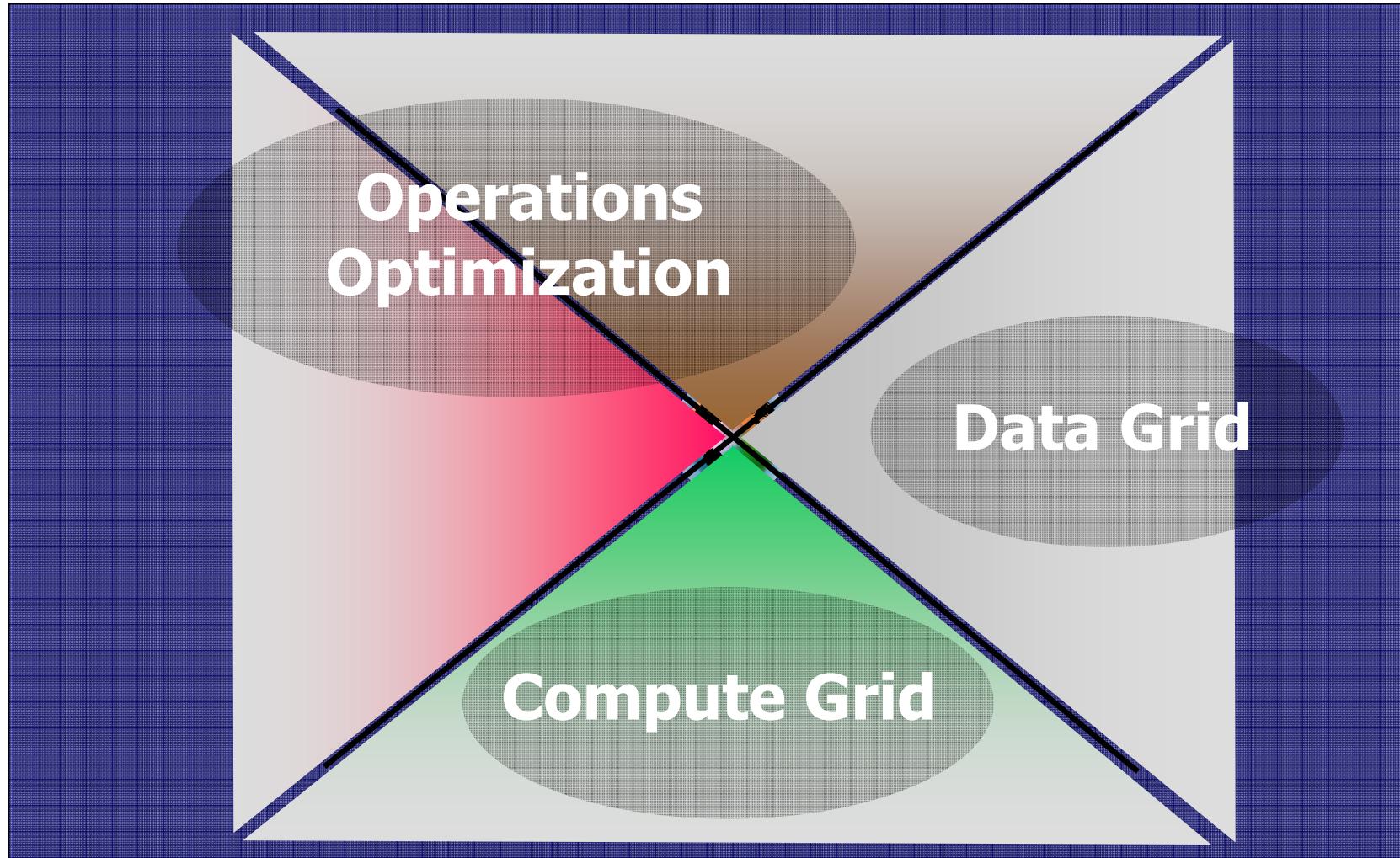
What is WebSphere XD?

Software to virtualize, control, and turbo-charge your application infrastructure



WebSphere XD Packaging Structure

Available as a single, integrated package, or as 3 individual components



New Install Approach

- WebSphere Extended Deployment (XD) 6.1 Supports centralized installation from Deployment Manager to Nodes in the cell.
- Single Install to the Deployment Manager
 - ▶ Includes all code for all of the scenarios described for all platforms
- “PUSH” of install package from DMGR to endpoints
 - ▶ Select a set of hosts and push XD to those endpoints
 - ▶ Installs appropriate endpoint code based on type of endpoint
 - ▶ Agent-less
- Centralization of Patch management



WebSphere Extended Deployment - Operation Optimization -



XD OO Customer Value/Scenarios

- Save Money through Server Consolidation
 - ▶ This is where the XDVA is useful
 - ▶ Consolidate multiple under-utilized servers into a shared environment that is smaller than the original environment
 - ▶ Typically seeing 20%-40% total savings, including the cost of XD
- Improve Environment Manageability
 - ▶ This is not about saving hardware/software cost
 - ▶ Enable the Middleware Operations team to run the environment easier
 - ▶ Focus on application versioning, monitoring, and XD admin upgrades to WAS
- Improve Availability, Scalability and Resiliency
 - ▶ Make their application more stable and more scalable
 - ▶ Focus here is on the adaptability of the XD environment, traffic management, better hardware utilization, service policy



Middleware Virtualization

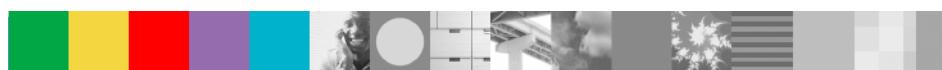
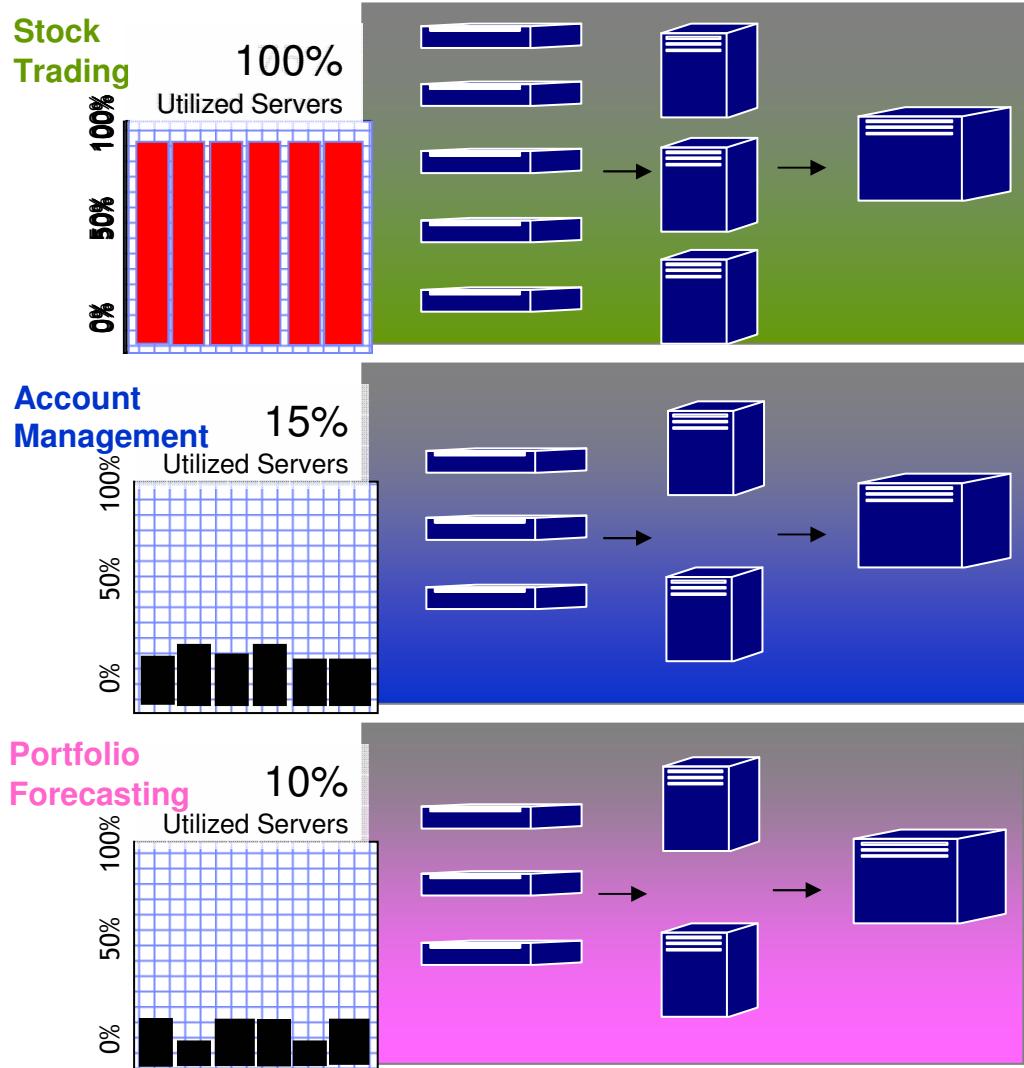
- Traditionally Server Topology is fixed
- XD treats the cell as a virtualized resource pool
 - ▶ It is a collection of machines that will host the application(s)
 - ▶ Nodes within a cell are tagged with “capabilities”. Custom “capabilities” may be defined.
- A Dynamic Cluster is a virtual cluster of servers hosting the application. The membership of the Dynamic Cluster is managed automatically
 - ▶ The active size of the Dynamic Cluster is managed automatically based on service policy and current conditions
 - ▶ Cluster members are placed on nodes which meet a set of specified capabilities (such as node is a member of a specific nodegroup)
 - ▶ Limits can be placed on the size of the dynamic cluster (min-max from 0-n)
 - ▶ Applications are assigned to Dynamic Clusters
- Isolation policies can be configured when dynamic cluster members are co-located on a node
- The combination of Node Capabilities and Dynamic Clusters provides the virtualization construct in XD



WXD OO: Middleware virtualization

Conventional Distributed Environment

- Environment
 - ▶ Multiple business critical applications
 - ▶ Hundreds of application servers
- Challenges
 - ▶ Underutilized servers
 - ▶ Inability to share resources across server pools – especially during peaks
 - ▶ Inconsistent quality of service for business critical applications
 - ▶ Human-intensive monitoring and management environment



WXD OO: Middleware virtualization

WXD Environment

- Virtualized
 - ▶ Pooled resources
 - ▶ Virtualized applications
- Goals based
 - ▶ Operational policies are attached to Application to reflect operational goals and importance of application
 - ▶ Autonomic managers monitor environment for maximum utilization using business goals
- Results
 - ▶ Reduce total cost of ownership (doing more with same/less)
 - ▶ Increase stability and repeatability of environment

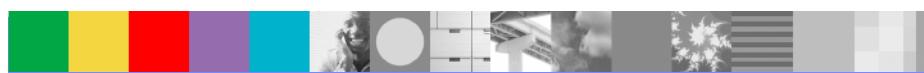
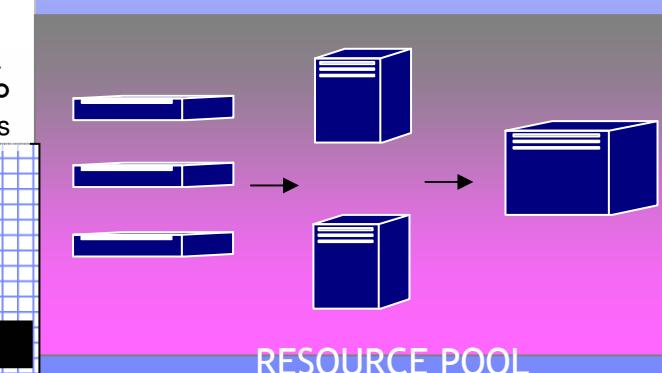
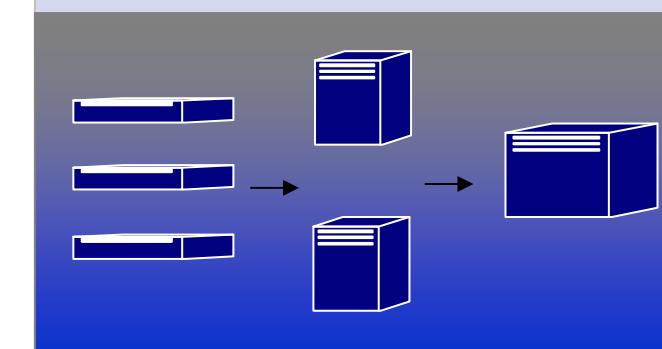
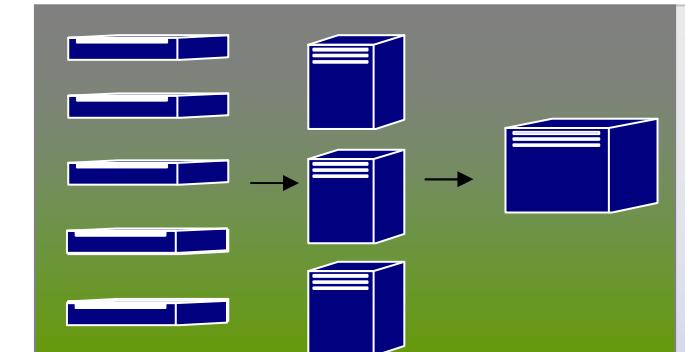
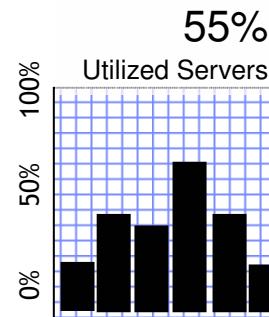
Stock
Trading

Customer
Support

Account
Management

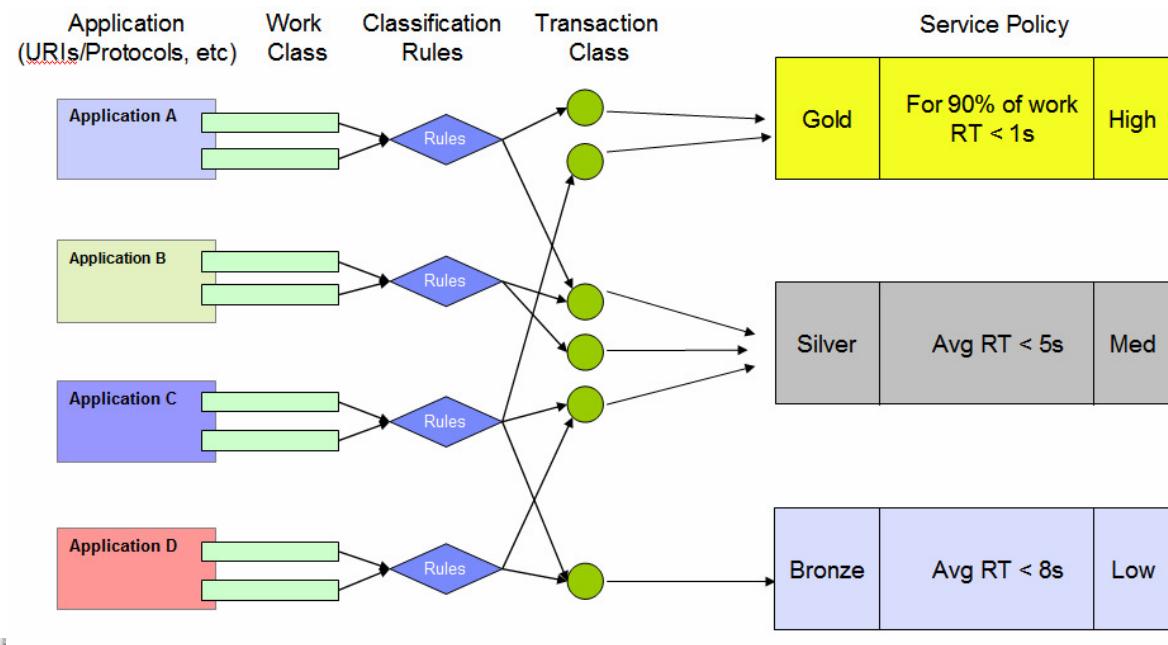
Risk
Management

Portfolio
Forecasting



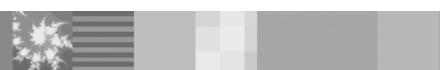
Defining SLAs through Service Policy

- Service Policy is definition of a performance goal used by XD to decide how to manage resources in the server environment
- Defined in terms of the end user result the customer wishes to achieve
- Comprised of three parts:
 - ▶ A set of classification rules to decide which policy applies to a given request
 - ▶ A performance goal the user desires to be achieved (i.e. 500ms average response time)
 - ▶ An importance level to inform XD of the relative priority of different classes of work



Policies – Applying Business Goals to Applications

Application Solutions	Service Policies	Goals	Priorities
Stock Trading	Gold	RT < 1sec	Very High
Account Management	Silver	RT < 2sec	Medium
Portfolio Forecasting	Bronze	RT < 5sec	Medium
Customer Support	Idle	Best Effort	Low
Risk Management			



Techniques to Meet the SLAs

- XD two primary techniques to meet Service Policy objectives
 - ▶ Traffic Shaping and Application Placement
- Traffic Shaping
 - ▶ Based on the notion that not all requests are equal and serving work first-come-first-serve is not necessarily the best approach
 - ▶ Controls Traffic in a number of ways
 - Prioritization – Processed in order of importance
 - Flow Control – Using queuing, the rate of work being sent to the server cluster is controlled
 - Traffic Spraying
 - Weighted Least Outstanding Requests
 - Dynamic Weights
 - Overload Protection – Control total amount of outstanding work for each class of service
- Application Placement
 - ▶ The ability to adjust the size of a Dynamic Cluster in real-time
 - ▶ Controls how much capacity is online for an application at any moment in time.
 - ▶ Provides integration with Tivoli Intelligent Orchestrator to enable new hardware to be provisioned into XD's Node Group.



On Demand Router (1/2)

- The On Demand Router (ODR) is a component that logically replaces and extends the functionality of the ND HTTP Plug-in
- The ODR provides the standard functionality of a proxy server with added On Demand features
 - ▶ Request classification and prioritization
 - ▶ Request queuing
 - ▶ Routing and load balancing
 - ▶ Weighted round robin dispatching with Dynamic WLM weights
 - ▶ Dynamic routing table updates with multiple WebSphere backend cells
 - ▶ HTTP Session affinity
 - ▶ SSL ID Affinity
 - ▶ WPF Partition Affinity

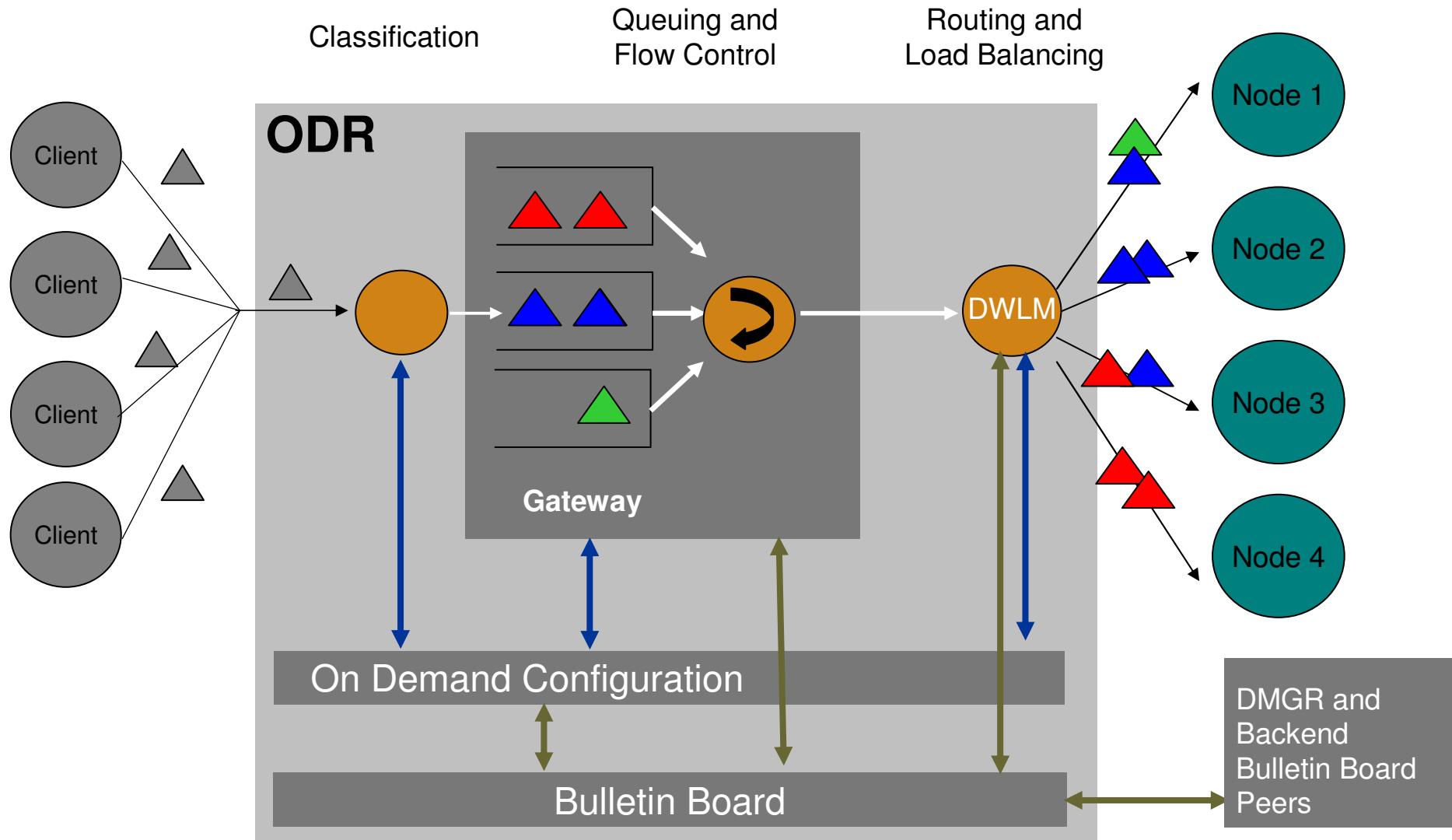


On Demand Router (2/2)

- The ODR does not need a plugin config file for request routing and load balancing (unlike an http server)
- The ODR uses On Demand Configuration component (ODC) to handle retrieval and distribution of back-end routing information
- Each ODR can be configured with one or more cell destinations
- Each ODR will automatically update its routing tables as applications are installed or removed from the back ends cells
- In the event that the DMGR is down or unavailable, the ODR continues to use its current configuration data
- The current configuration data is persisted to disk so that the ODR can be stopped and started even when the DMGR is not available



ODR functional components



Dynamic WLM

- Can be enabled or disabled for static & dynamic clusters
 - ▶ Dynamically adjusts routing weights for cluster members
 - ▶ Weights are integers: The higher a server's weight relative to other servers, the higher the share of request to the given server
 - ▶ Load balancer controller runs periodically [20 sec]
 - ▶ Goal → equalize the average response time across servers in clusters
- Balances workload across cluster members
 - ▶ Load balancer executor has failover capability by redirecting client requests if one or more servers are unable to process them
 - ▶ Goal is to equalize service times among nodes serving a dynamic cluster
 - ▶ Web module & EJB service times
- Supports DWLM for
 - ▶ Heterogeneous nodes
 - ▶ Multi-tiered requests
 - ▶ Changes due to Placement Control

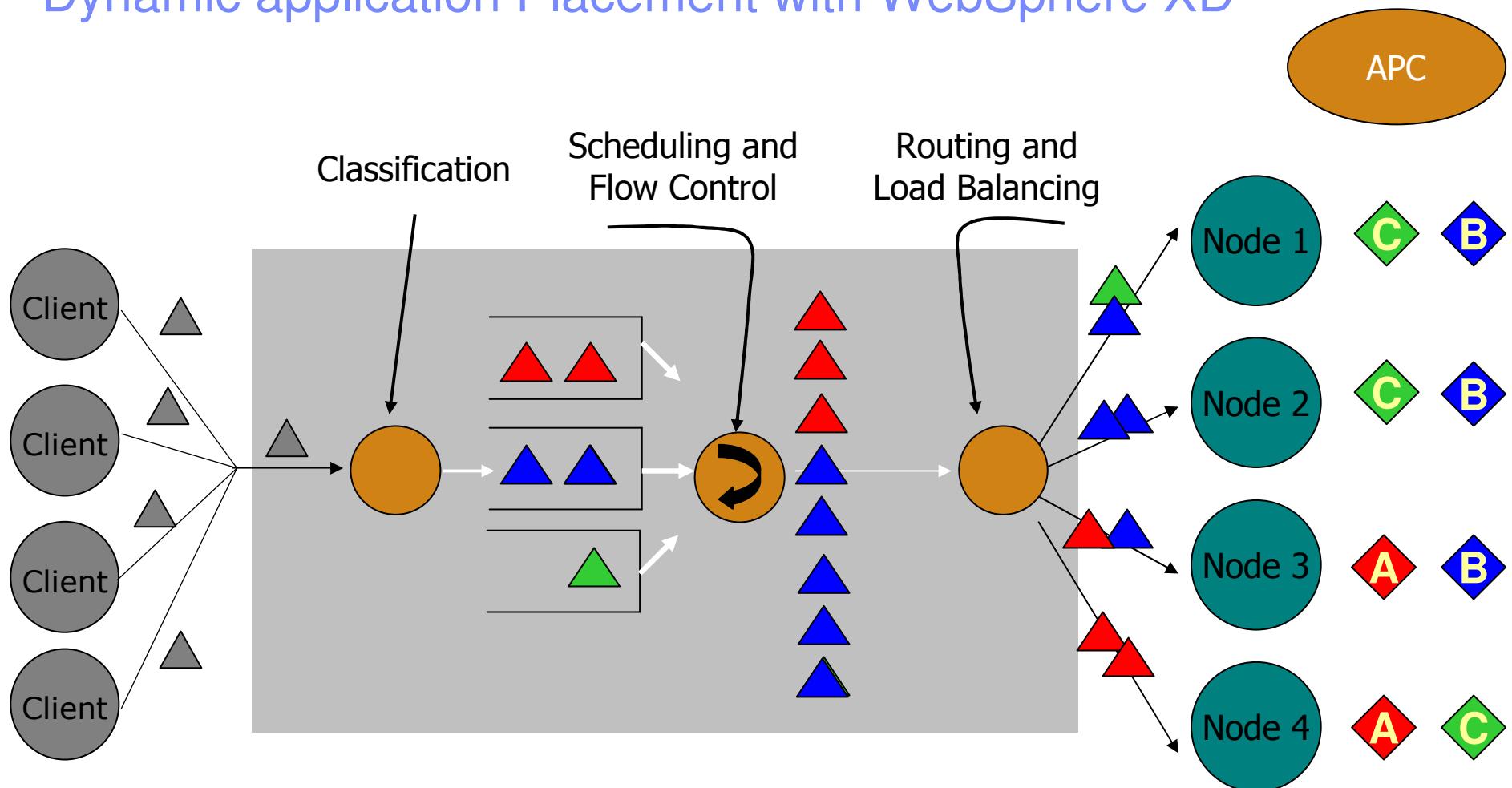


Application Placement

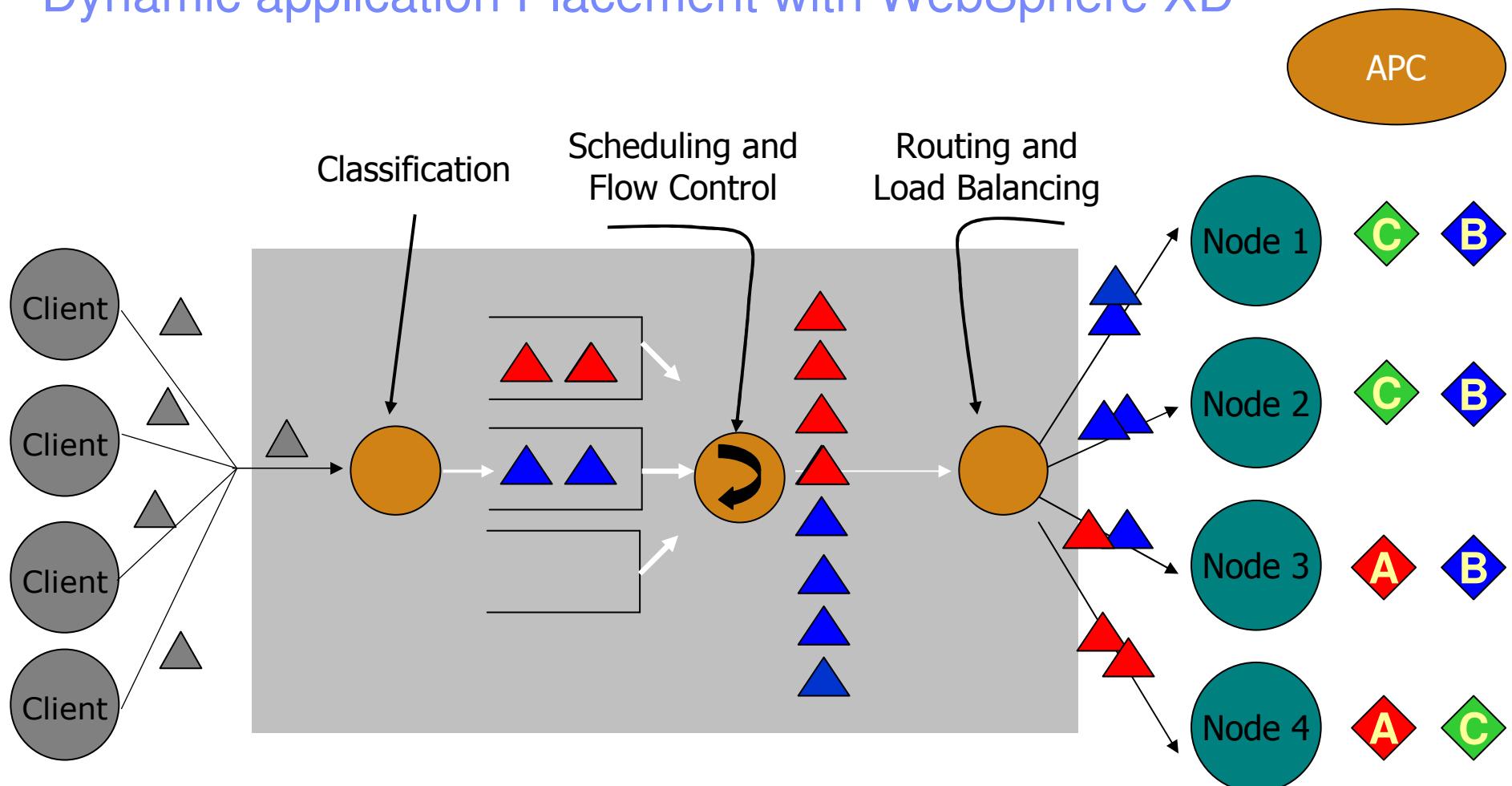
- Application Placement controls size of dynamic clusters
- One Application Placement Controller (APC) per Cell
 - ▶ HA Managed Singleton
- APC decides how many servers should run and where
 - ▶ For each Dynamic Cluster
 - ▶ Respecting vertical stacking constraints (load-independent capacity, e.g., memory)
 - ▶ Dividing up load-dependent capacity (CPU) according to demand
 - ▶ Load dependent demand determined by ARFM



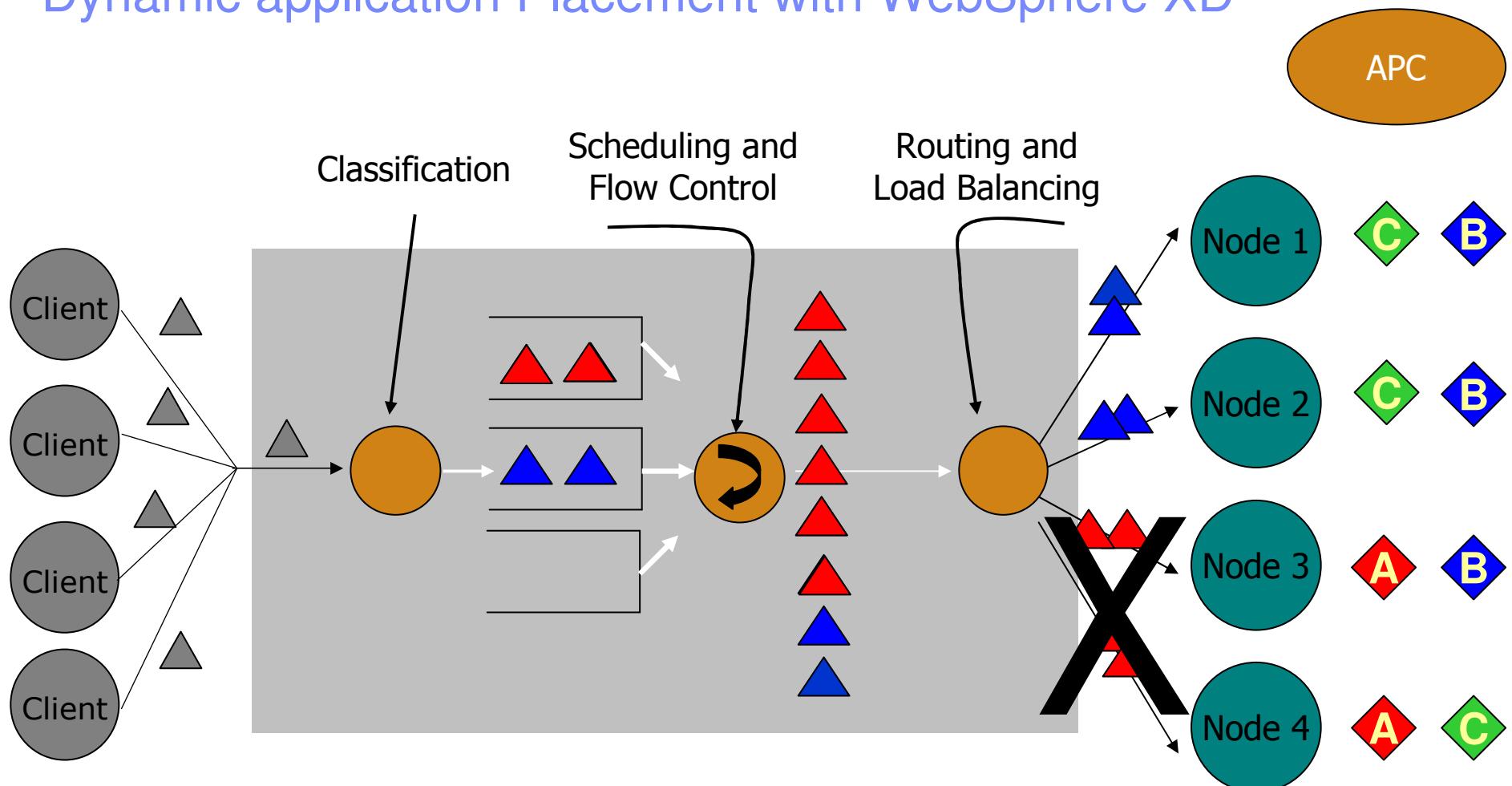
Dynamic application Placement with WebSphere XD



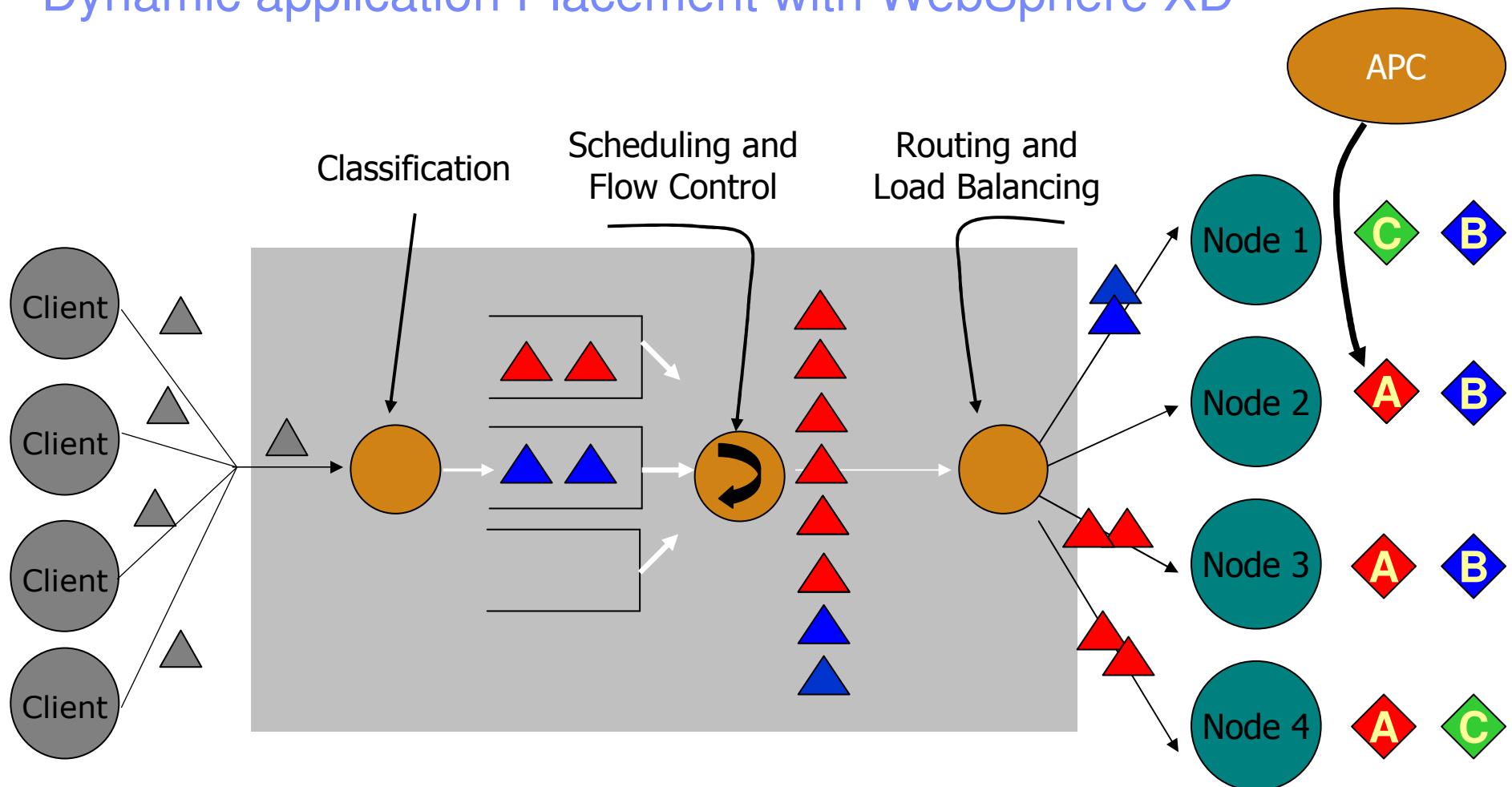
Dynamic application Placement with WebSphere XD



Dynamic application Placement with WebSphere XD

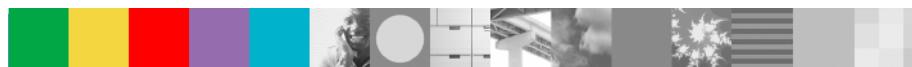
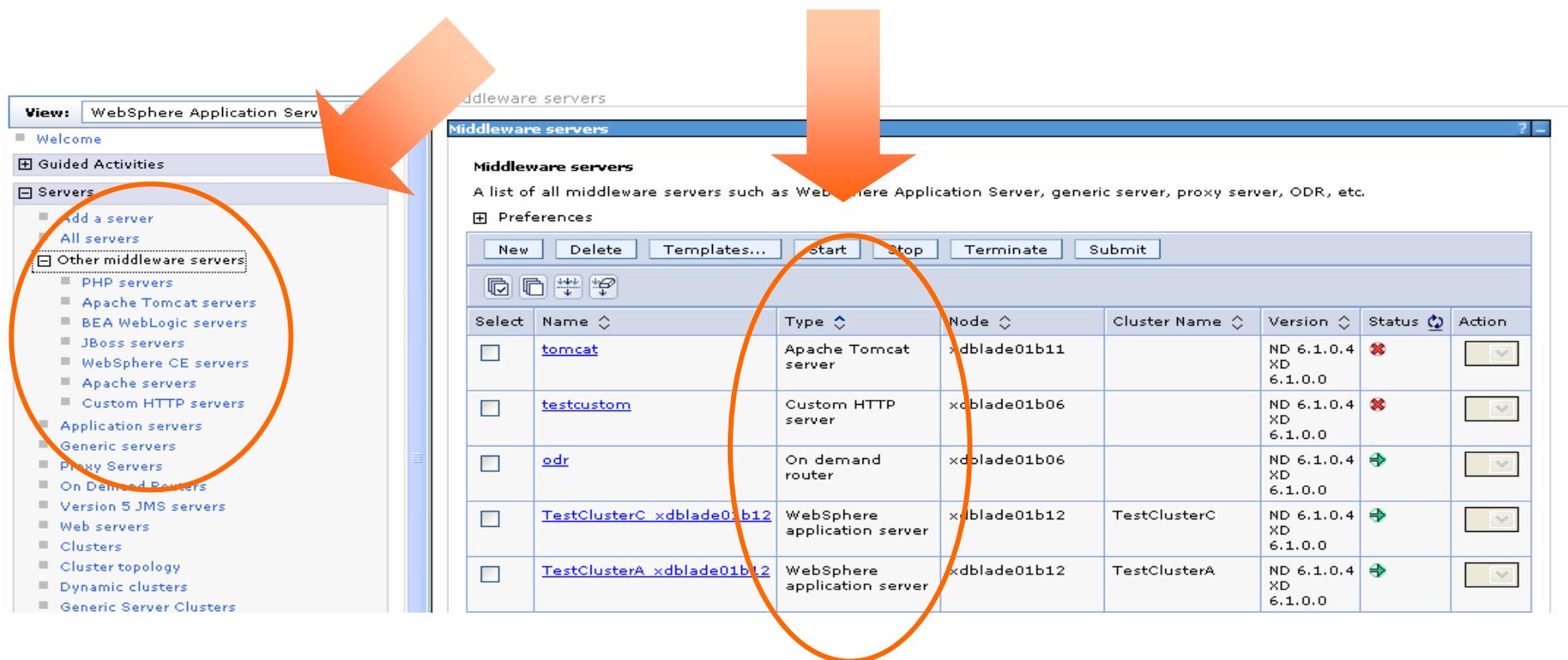


Dynamic application Placement with WebSphere XD



Unified Administration across Middleware

Common collection views allow administrators to interact with their servers regardless of type



1st Class Support for Non-WebSphere Platforms

Three categories of support for middleware server types...

Complete Lifecycle Management

- Create/remove server instances
- Govern all aspects of server configuration
- Provide operational control
- Deploy applications
- Server health and performance is monitored and visualized.



Assisted Lifecycle Management

- Provides specific templates for creating representations of existing servers and applications
- Servers can be controlled operationally
- Administrative utilities are provided to manage the external configuration and runtime
- Server health and performance is monitored and visualized.

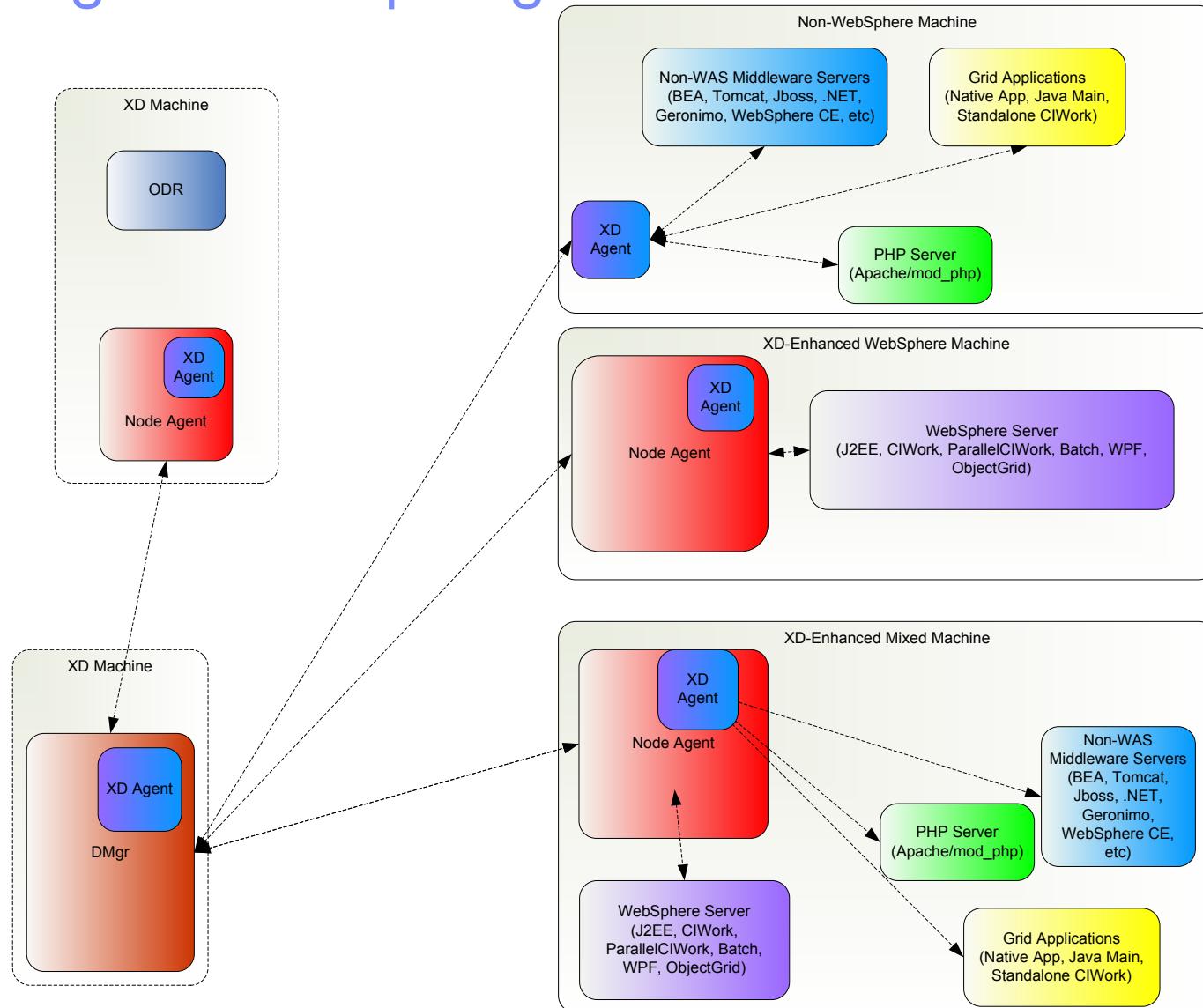


Generic Lifecycle Management

- Provides generic templates for the user to manually define servers and operational commands.
- Control server operations and monitor health and performance



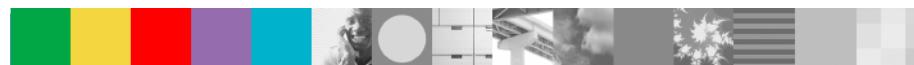
XD Management Topologies



Server Maintenance Mode

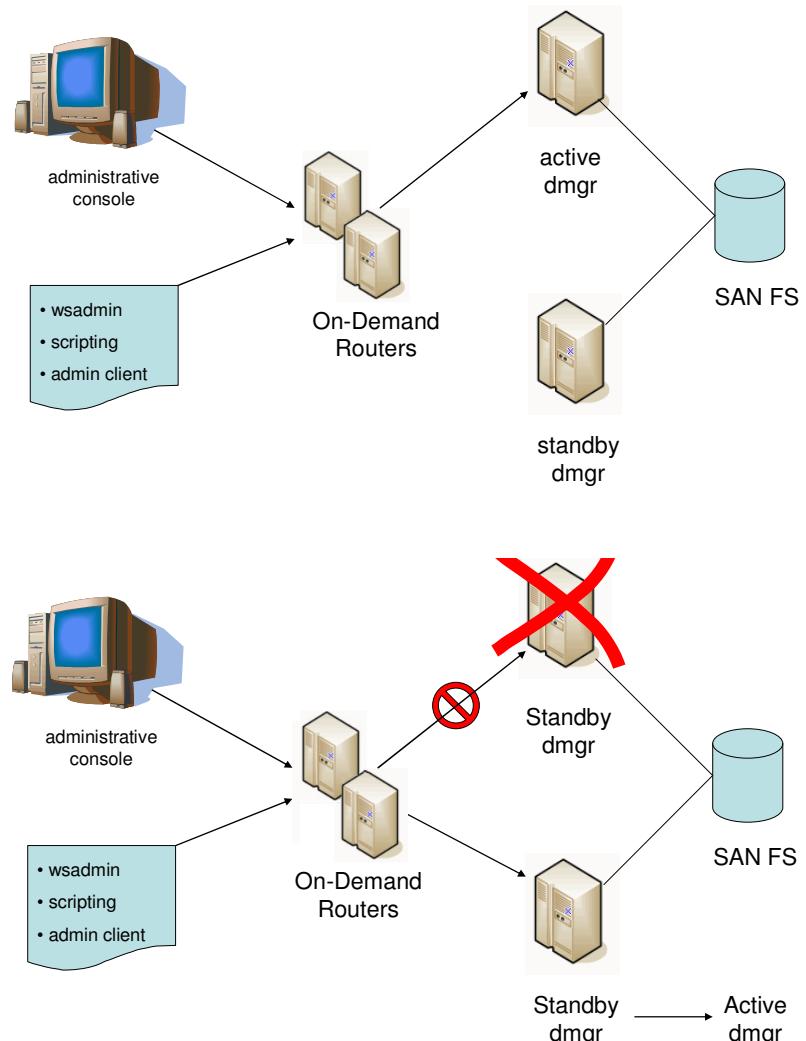
- XD provides the capability to isolate a running server (of any type) from production traffic. This allows for problem determination to be performed on the server or other maintenance without disruption to production traffic.
- If the server is a member of a dynamic cluster, a new cluster member will first be started before the server is placed into maintenance mode in order to assure the minimum policy on the dynamic cluster is met.

Select	Name	Type	Node	Maintenance mode	Status	Action	Maintenance mode	
<input type="checkbox"/>	AccountManagement_DC_N2Node01	WebSphere application server	N2Node01	AccountManagement_DC	ND 6.1.0.7 WXDOP 6.1.0.0 XD 6.1.0.0			
<input checked="" type="checkbox"/>	AccountManagement_DC_N3Node01	WebSphere application server	N3Node01	AccountManagement_DC	ND 6.1.0.7 WXDOP 6.1.0.0 XD 6.1.0.0			
<input type="checkbox"/>	FinancialAdvice_DC_N2Node01	WebSphere application server	N2Node01	FinancialAdvice_DC	ND 6.1.0.7 WXDOP 6.1.0.0 XD 6.1.0.0			
<input type="checkbox"/>	FinancialAdvice_DC_N3Node01	WebSphere application server	N3Node01	FinancialAdvice_DC	ND 6.1.0.7 WXDOP 6.1.0.0 XD 6.1.0.0			



Highly Available Deployment Manager Configuration

- Each deployment manager on a separate machine
 - ▶ Only one is active
 - ▶ Others are standby
- Shared file system required for dmgrs to share configuration repository
 - ▶ File system with recoverable locks required - e.g. SAN FS, DFS v4
- JMX traffic proxied through XD On-demand Router (ODR)
 - ▶ SOAP connector only
- HA ODRs recommended
 - ▶ (they're recommended for production XD configurations anyway)
- hadmgrConfig command line utility provided to perform configuration



Monitoring Operations

- XD provides a set of views for understanding the dynamic goals directed environment the application is being hosted in. The administrative console is enhanced with Operations and Reporting tabs off the detail view of servers, clusters, applications and service policies
- Operations tab provides insight into the stability of the resource, how work for the resource is actively being managed, outstanding tasks that need operators to act upon, and where the resource is currently placed (running).
- Reporting tab allows for in depth charts to be viewed to understand the performance of the environment

Middleware servers > TestClusterA_xdbblade01b13

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Select	Application Server	Cluster	Stability	Weight	CPU Utilization
<input type="checkbox"/>	myPHPserver		Stable	1	0%
<input type="checkbox"/>	singletonServer		Stable	0	44.91%
<input type="checkbox"/>	TestClusterA_xdbblade01b13	TestClusterA			
<input type="checkbox"/>	nodeagent				

Total Filtered total:

Middleware servers > TestClusterA_xdbblade01b13

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Select	Application Server	Cluster	Stability	Weight	CPU Utilization
<input type="checkbox"/>	TestClusterA_xdbblade01b13		Stable	1	0%
<input type="checkbox"/>	nodeagent		Stable	0	44.91%
<input type="checkbox"/>	od1				

All Applications > A

Use this page to configure an enterprise application. Click the links to access the application or its modules.

On demand routers

On demand routers associated with this resource.

- od1 (EastCell, blade1) 1642 requests in last 10 mins
- od2 (EastCell, blade2) 2034 requests in last 10 mins

Applications

Applications associated with this resource.

- TestAppA (EastCell) 3876 requests in last 10 mins

Service policies

Service policies associated with this resource.

- Platinum (EastCell) 1678 requests in last 10 mins
- Gold (EastCell) 2011 requests in last 10 mins
- Default_SF (EastCell) 87 requests in last 10 mins

Reports Preferences

Open new chart tab

A

Preferer

10.00
8.00
6.00
4.00
2.00
0.00

Time: 12-08-55, 12-07-15, 12-07-35, 12-07-55, 12-08-05, 12-08-20

1/23/07 12:08:29

Add data Remove View Table

Select	Pattern	Data Set Type	Data Set	Data Metric	Data Filter	Scale
<input type="checkbox"/>	J2EE Module	A:MicroWebApp.war	Concurrent Requests	1.0		
<input type="checkbox"/>	J2EE Module	A:MicroWebApp.war	Avg. Throughput	1.0		
<input type="checkbox"/>	J2EE Module	A:MicroWebApp.war	Avg. Response Times (ms)	10.0		



Monitoring XD Itself

- An XD Summary View is introduced for operators to monitor and receive visual alerts to when the stability of XD becomes questionable or unstable.
 - Core runtime components can be tracked (location and stability)
 - State and stability of ODRs
 - Coregroup stability
 - Node state and stability
- A reporting summary view provides operators with the ability to configure sets of in-depth charts into groups that can be pulled up and viewed at any time for a real-time snapshot of the environment's performance.

Name	Scope	Stability	Current location
ARFMController	N1Cell01	∅	N1Cell01/N1ODR01/odr
Application Placement Controller	N1Cell01	✓	N1Cell01/N1CellManager01/dmgr
Async PMI Bridge	N1Cell01	✓	N1Cell01/N1ODR01/nodeagent
DWLM Controller	Tomcat_DC (N1Cell01)	✓	N1Cell01/N1ODR01/nodeagent
Health Controller	N1Cell01	✓	N1Cell01/N1CellManager01/dmgr
Node Detect Bridge	N1Cell01	✓	N1Cell01/N1ODR01/nodeagent
Work Profiler Controller	N1Cell01	✓	N1Cell01/N1CellManager01/dmgr

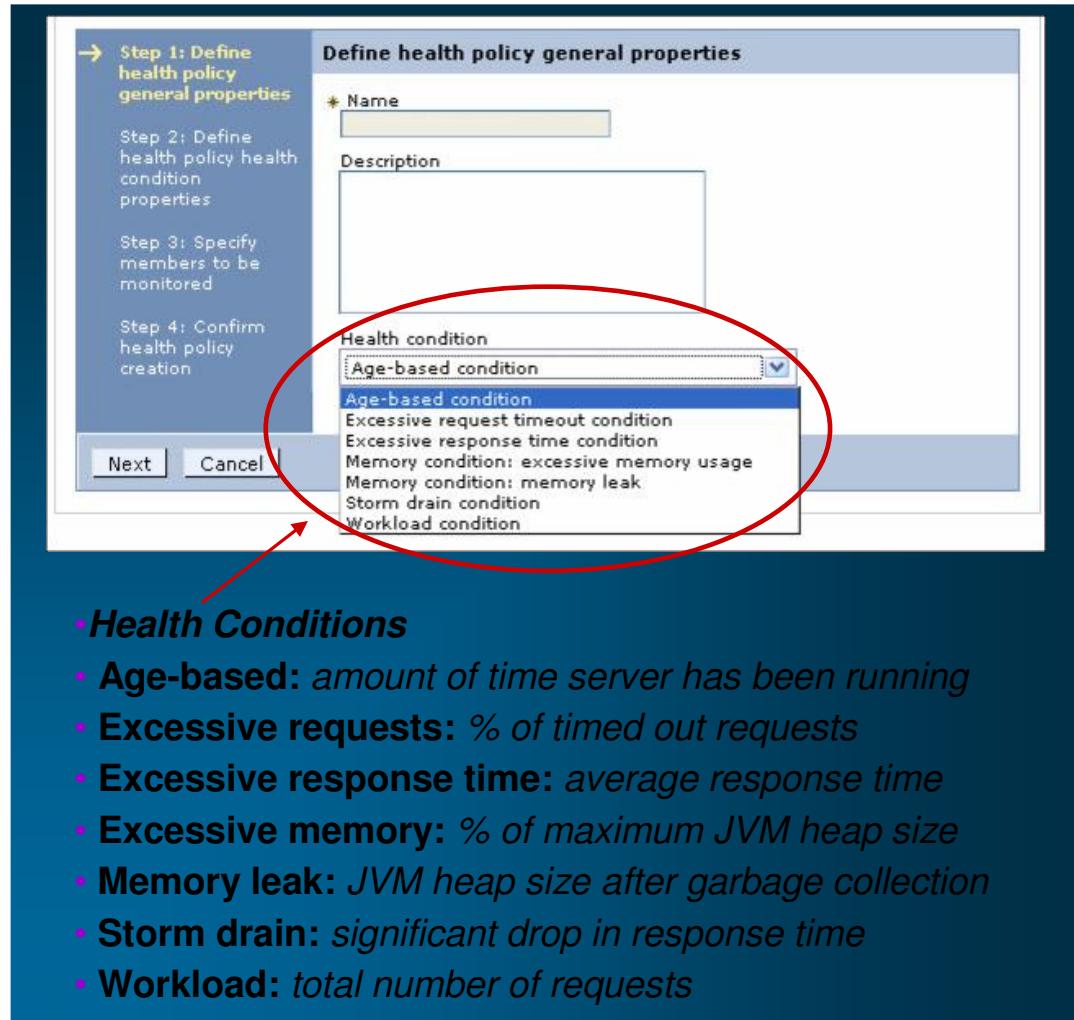
Health Monitoring and Management (HMM)

- Monitor environment for common health problems and take corrective action
- Conditions include:
 - ▶ Memory Leaks or Excessive Memory Usage
 - ▶ Hung Servers
 - ▶ Excessive Requests Timeouts or Response Time
 - ▶ Storm Drains
 - ▶ Extended Service Policy Violation
 - ▶ Server Age and Max Requests
- When detected an action plan can be put into effect automatically
 - ▶ Notify administrators (including via email)
 - ▶ Capture diagnostics information (java thread or heap dump)
 - ▶ Place the server into Maintenance Mode
 - ▶ Restart server
- Custom conditions, actions and action plans
- Server restarts are smart and done in a way to prevent outage and service policy violations



Health Management – Health Policies

- Health policies can be defined for common server health conditions
- Health conditions are monitored and corrective actions taken automatically
 - ▶ Notify administrator
 - ▶ Capture diagnostics
 - ▶ Restart server
- Application server restarts are done in a way that prevent outages and service policy violations



Custom Health Conditions

- XD enables customers to create expressions defining what “unhealthy” means in their environment
- Custom expressions can be built using operands which represent metrics from the On Demand Router, base PMI metrics (WAS only), MBean operations and attributes (WAS only), and/or URI return codes. Complex expressions using a mix of operands is supported.
- Other middleware server types can leverage the ODR metrics and URI return code operands
- Create the health policy by using the `createHealthPolicy AdminTask` command

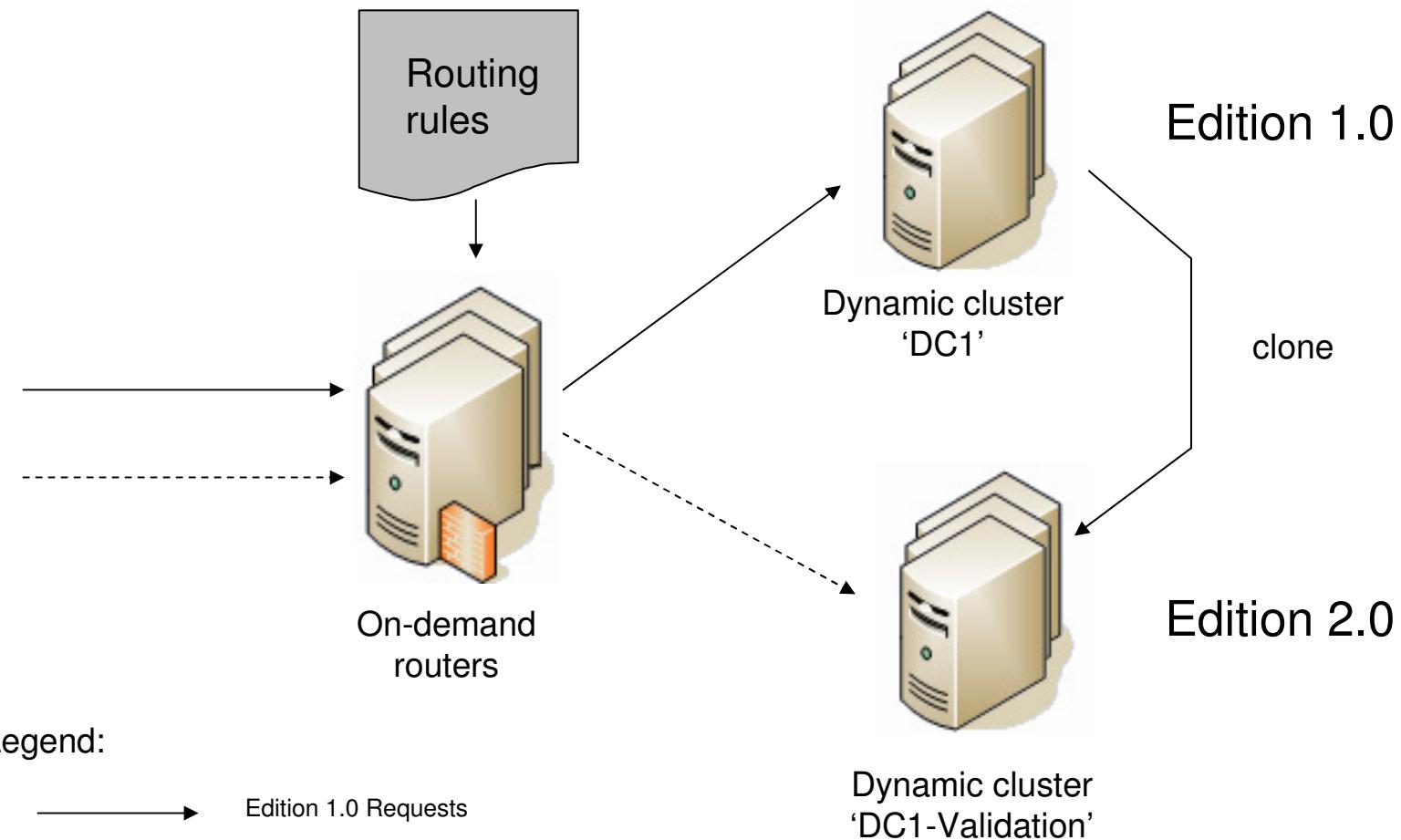


Application Edition Management

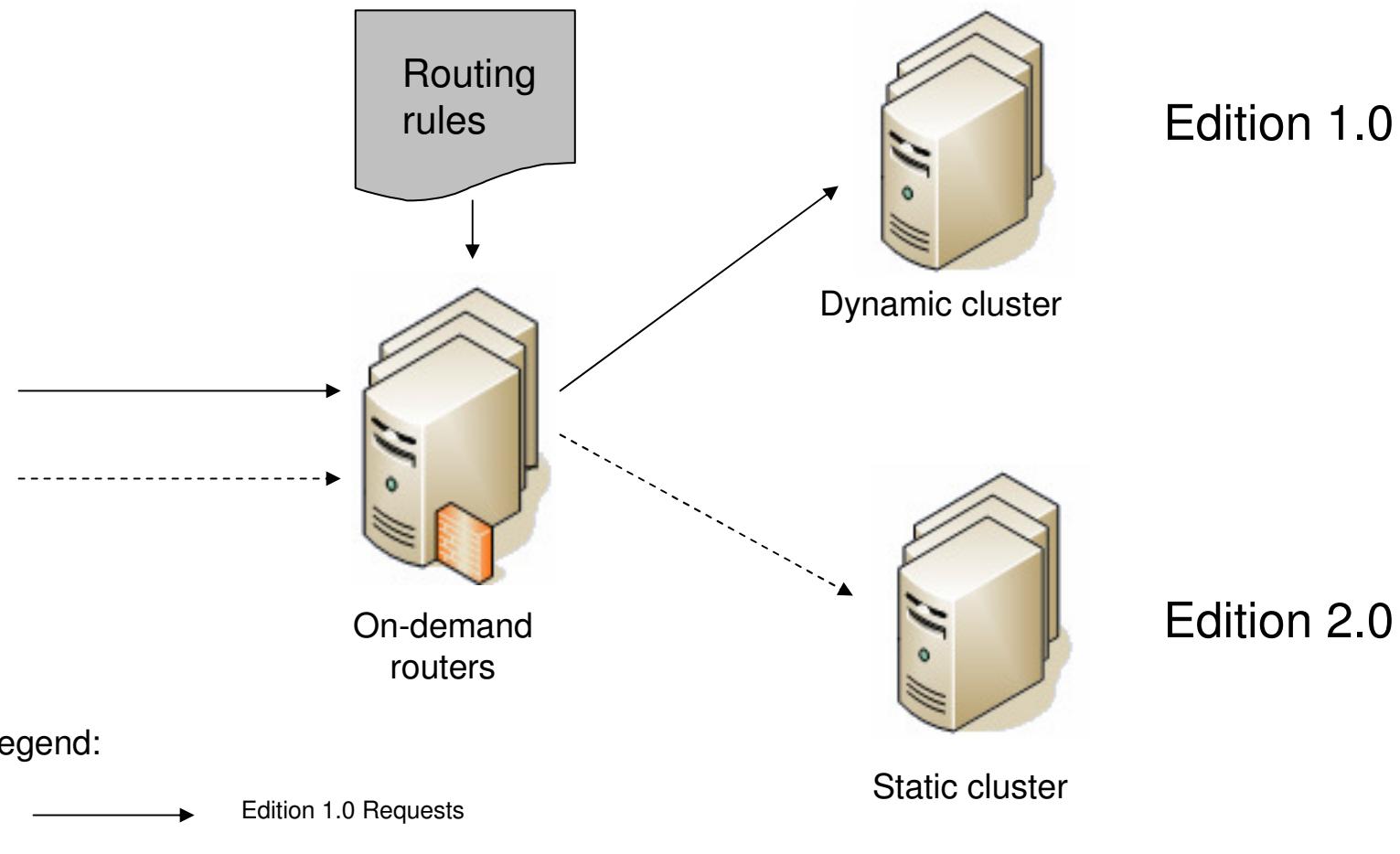
- WebSphere XD supports managing multiple editions of an application in a WebSphere cell
 - ▶ Only on Complete Lifecycle Management servers
 - ▶ Interruption-free rollout of application updates (grouped or atomic rollout)
 - ▶ Ability to “roll back” to a previous application version
 - ▶ “Validation mode” to verify functionality using a subset of users
- An *edition* is a distinct instance of a J2EE or PHP application (similar to *version*)
- Each edition is identified by a label
- An edition is a *deployment* version of an application
 - ▶ May be a distinct *build* version
 - ▶ May be the same build version with different deployment bindings (e.g., resource-ref)
 - ▶ May be both
- Useful for maintaining work classes from one edition to another.



Operational Scenarios: Validation Mode



Operational Scenarios: Concurrent Activation



Edition Control Center: Manage Editions

[Edition Control Center > Manage Editions](#)

Manage editions of an application. The deployment targets for each edition were specified during the application install process. After install, an edition is initially in the inactive state. Inactive editions cannot be started. Activating an edition makes it eligible to be started. Validating an edition puts it into a special "validation mode" that configures the edition to run on a clone of its original deployment target. Validation mode requires assignment of a routing policy to the edition to control who may access it. Rolling out an edition performs an interruption-free upgrade of one edition to another on the same deployment target. Rolling out an edition that is in validation mode performs an interruption free upgrade of the edition on the deployment target from which the validation mode target was cloned. After the rollout, the clone is deleted. Deactivation makes an edition ineligible to be started. An edition must be stopped before deactivation.

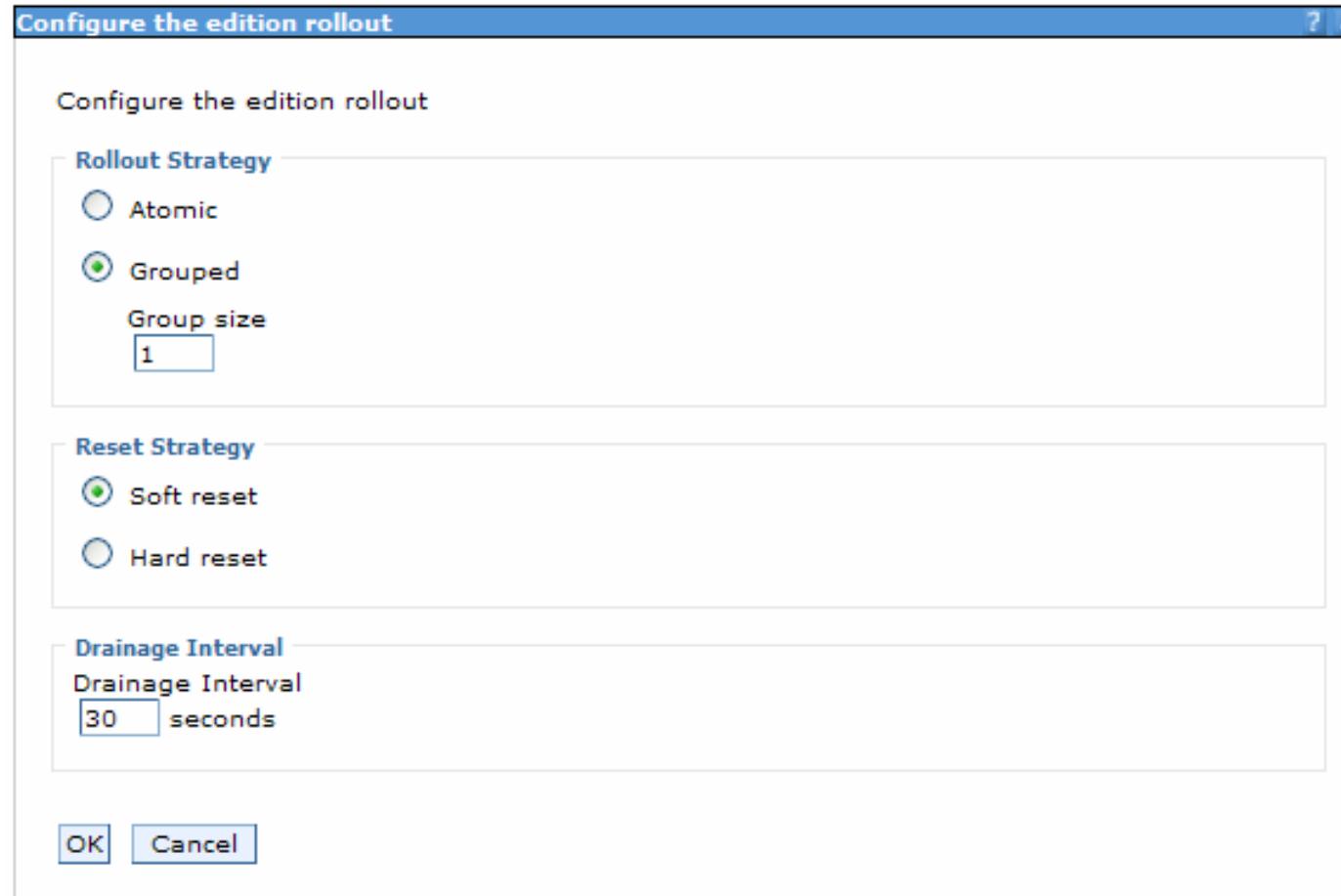
Application=**XDStockTradeEdition**

[+ Preferences](#)

Activate Validate Rollout Deactivate				
Select	Editions ▲	Description	Target ◊	State ▲
<input type="checkbox"/>	1.0		WebSphere:cell=wsbeta156Cell01,cluster=StockTrade_DC	INACTIVE
<input type="checkbox"/>	1.0.1		WebSphere:cell=wsbeta156Cell01,cluster=StockTrade_DC	ACTIVE
Total 2				



Edition Rollout



Configuring Routing Policy

- On-Demand Router can be configured route requests to a particular application edition
 - ▶ By IP address
 - ▶ By user or group
 - ▶ By HTTP cookie
 - ▶ Any combination of the above

The screenshot shows the 'Routing Policies' tab of the IBM On-Demand Router configuration interface. At the top, there are tabs for Configuration, Local Topology, Service Policies, and Routing Policies. Below the tabs, a message says 'Define routing policies for an application and all of its editions:'. There are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

The main area is titled 'Work Classes for HTTP Requests' with 'Default_HTTP_WC' selected. It includes 'New' and 'Delete' buttons. A section titled 'If HTTP request matches' shows an 'HTTP patterns' field containing '/LongRunningJobSchedulerWebSvcRouter/* (LongRunningJobSchedulerWebSvcRouter.war)'. An 'Edit HTTP Patterns' button is also present.

Below this, a section titled 'Then apply the following routing rules' contains a table:

Add Rule	Delete Rule	Move Up	Move Down
Select	Order	Routing Rule	
<input type="checkbox"/>	1	If clientipv4 = '192.168.0.4' Then Permit routing to Select edition name LongRunningScheduler Apply	

There is also a 'Rule Builder' button.

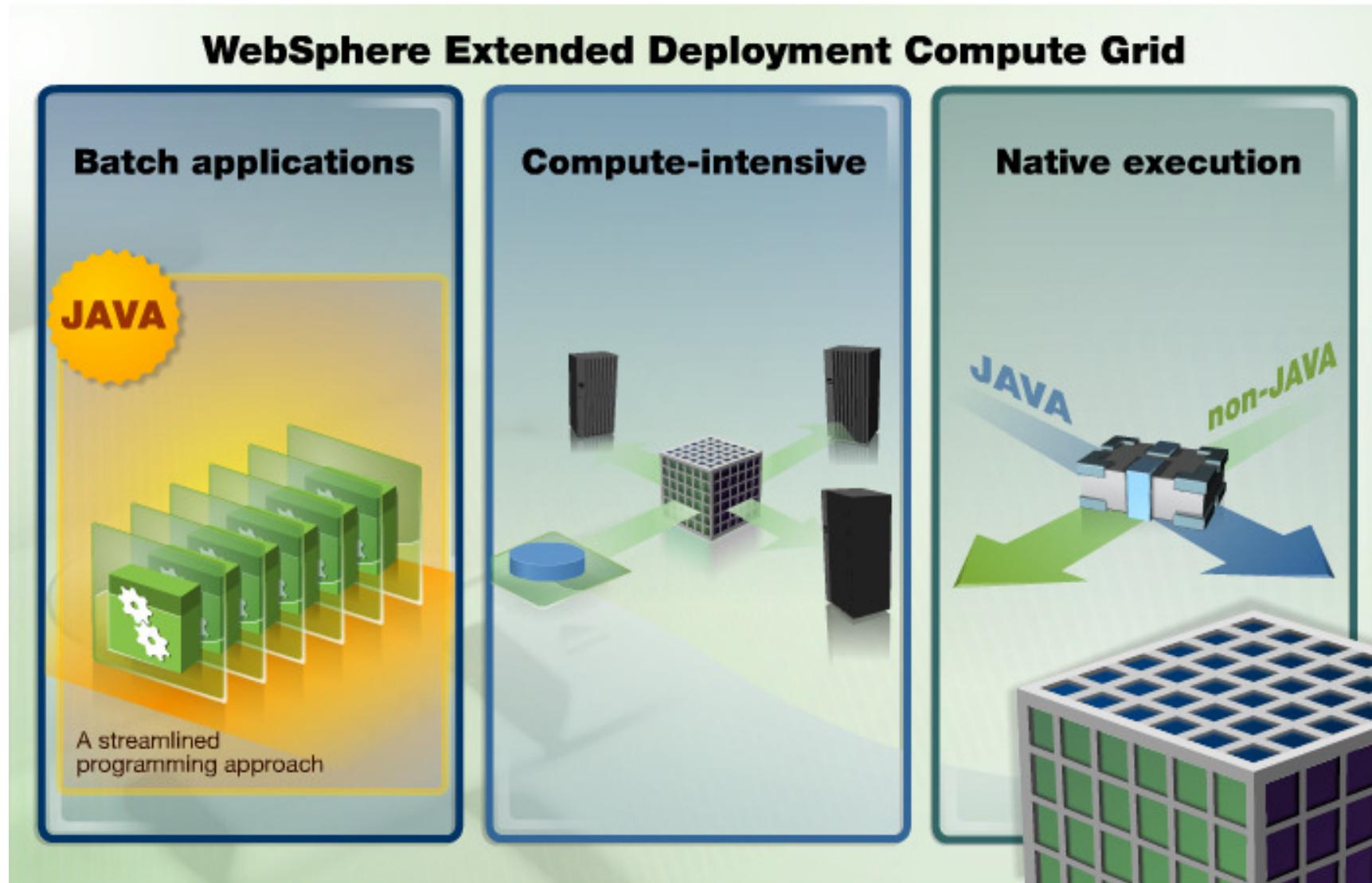
At the bottom, there is a section titled 'If no routing rules apply' with fields for 'Select action' (set to 'Permit routing to') and 'Select edition name' (set to 'LongRunningScheduler').



WebSphere Extended Deployment *- Compute Grid -*

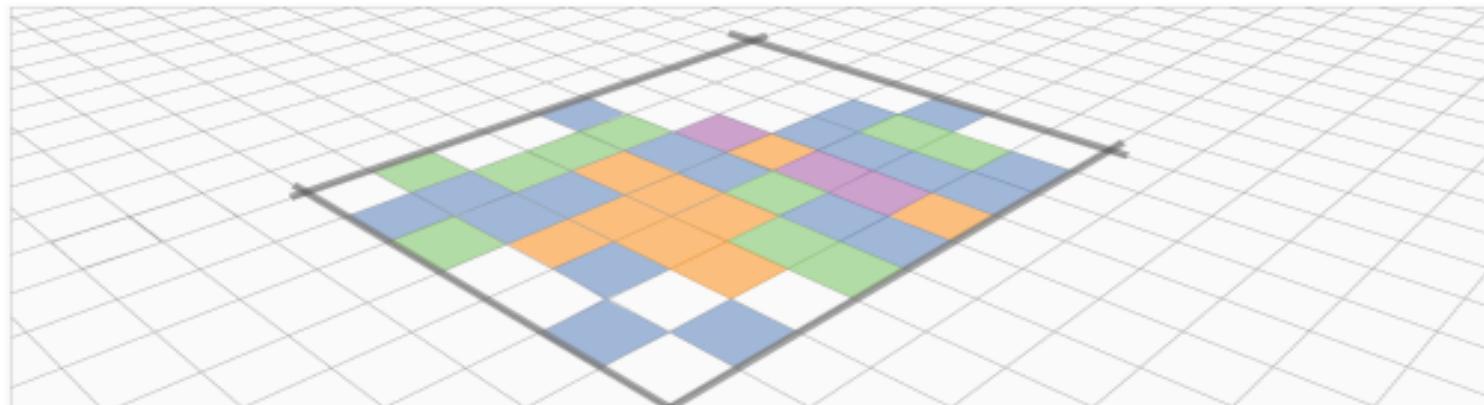


XD CG Customer Value/Scenarios



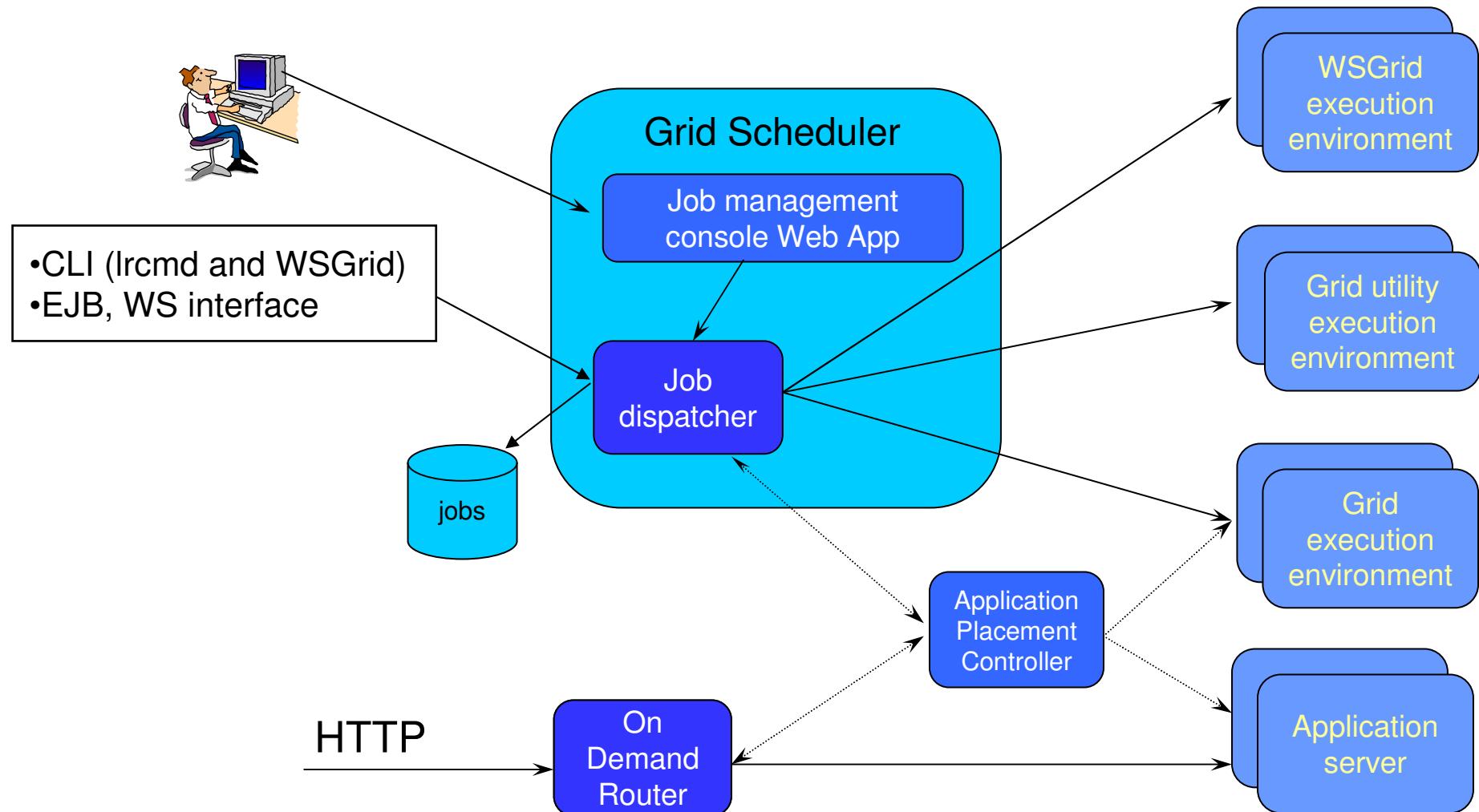
Compute Grids

- J2EE Applications are typically online, request-response oriented in nature
- Many problems lend themselves to more of a submit-and-execute style that can run for a while in an unattended manner
 - ▶ Portfolio, Weather Forecasting
 - ▶ Modeling
 - ▶ End of day/month/quarter transaction processing
- Traditionally these computing environments have been separate from the online world
- XD introduces these types of applications to your Java/J2EE based infrastructure

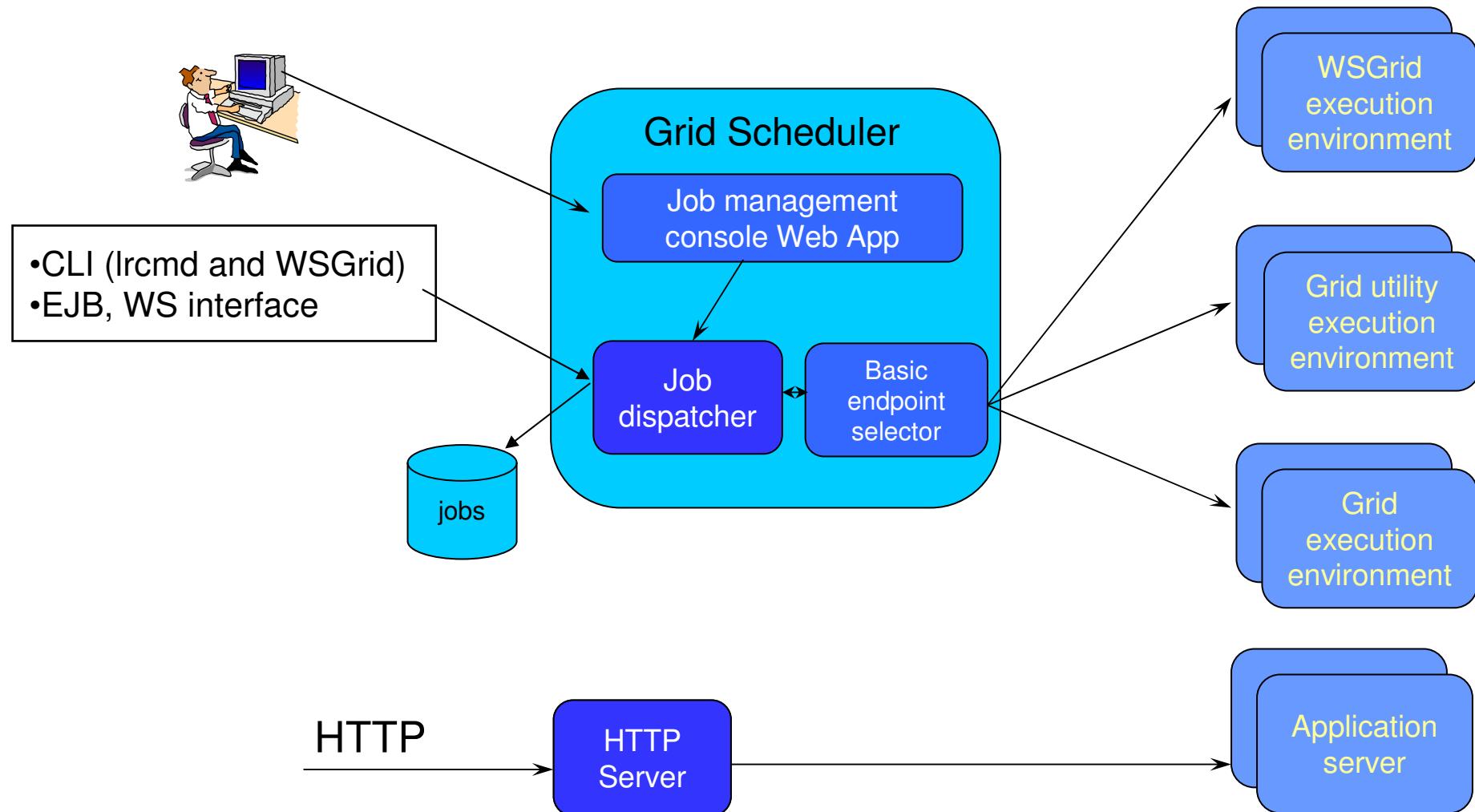


WebSphere Compute Grid – mixed workloads (LR + OLTP)

General Architecture (with OO)



General Architecture (without OO)



Grid Programming Models

- Two styles of grid applications are supported
- Compute Intensive Applications
 - ▶ A job that uses the CPU for an extended time to perform some computation
 - ▶ Implementable as components on WebSphere or standalone programs in Java or native languages
- Java Transactional Batch
 - ▶ Container manages transactions, data streams, steps, checkpointing, and job lifecycle
 - ▶ Customer provides logic to process one record in the data stream
 - ▶ Data streams are an abstraction that supports data from any source, including database, network, files, etc.
 - ▶ Implemented in Java on WebSphere Today
- Applications are packaged and deployed as regular J2EE EAR files

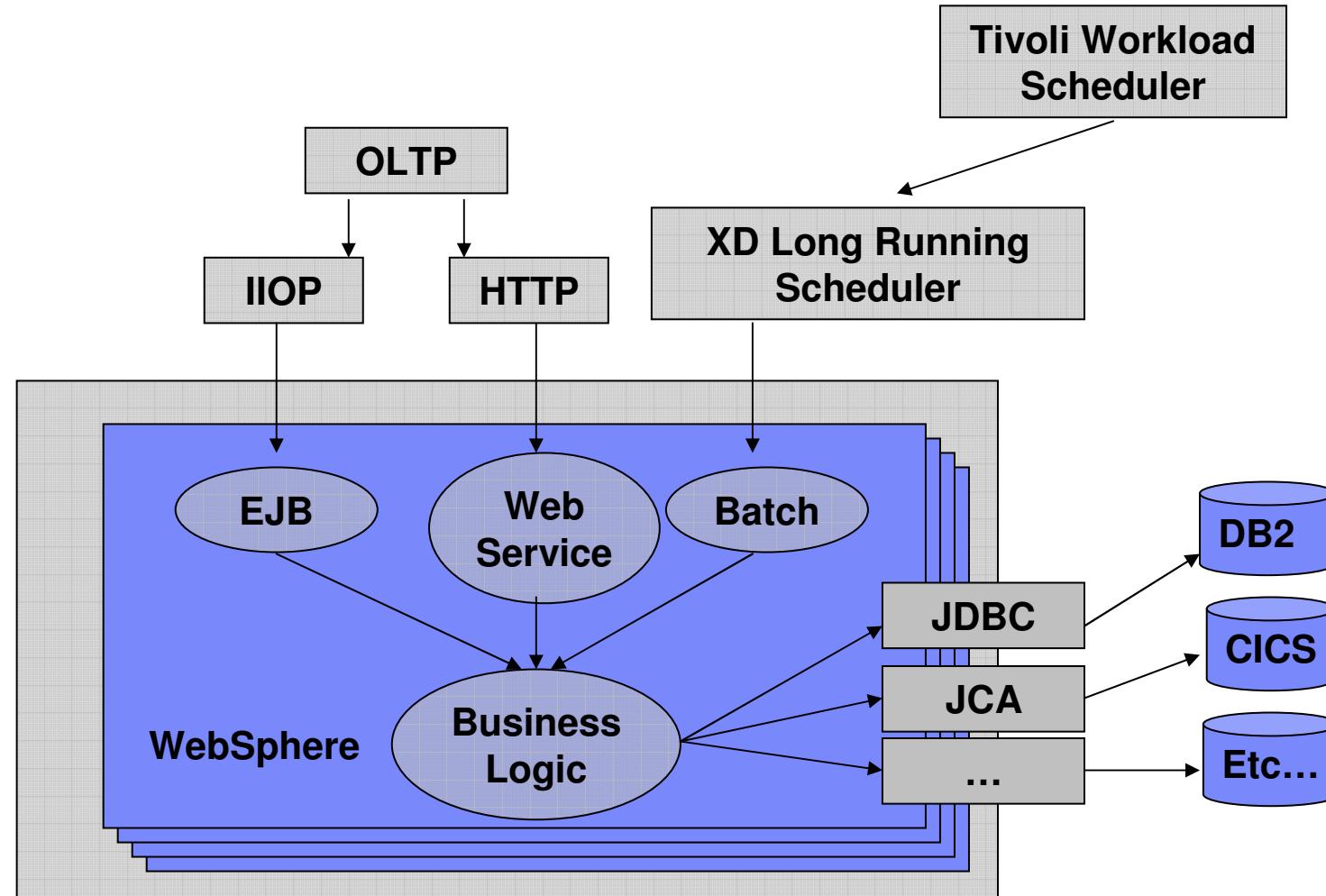


Native Jobs: XD Jobs Beyond WebSphere

- Support for non-J2EE programs
 - ▶ background command execution
 - ▶ Java main, scripts, compiled programs
 - ▶ Execute in own Unix process
- Execute outside WebSphere Application Server
 - ▶ under control of Middleware Agent
 - ▶ supported on distributed platforms only

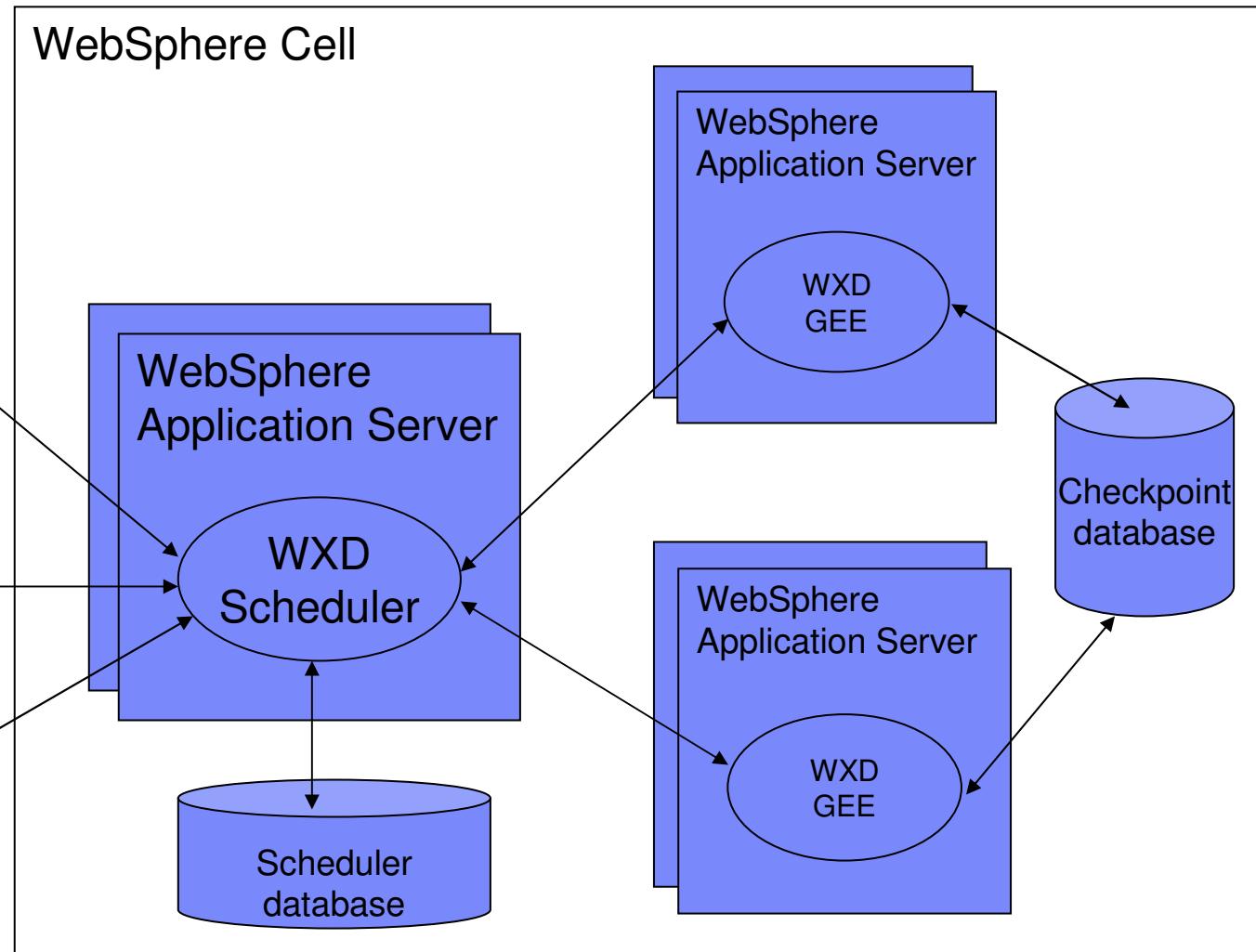
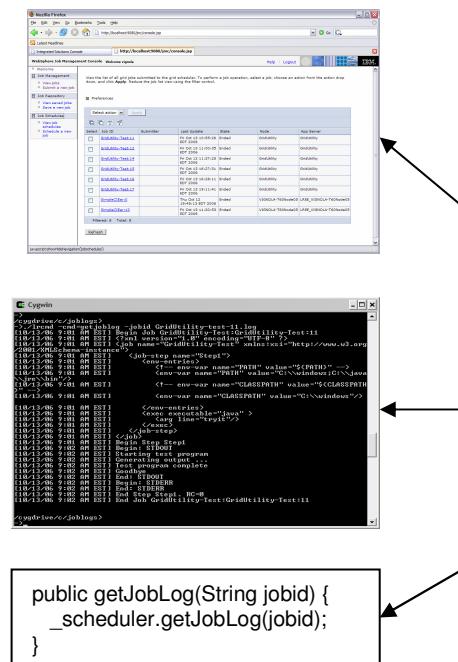


Running Mixed Workloads- OLTP and Batch



WebSphere XD Environment – Topography

- console
 - command line
 - APIs



Job Management Console - Submit Job

- simple one-click job submission
- job definition source from file system or repository
- optionally review/modify substitution properties

Specify job

Specify the path of the job definition to submit as a new job. The job definition might originate from the local file system or from the grid scheduler's job repository. Users in the Iadmin role can save a job definition from the local file system to the job repository.

Local file system
* Specify path
C:\WebSphere\AppServer4\longRunning\posting Sample

Job repository
* Specify job name

Modify substitution properties

Specify information for scheduling a grid job.

Specify job properties

Specify values for substitution properties for this job.

Name	Value
checkpoint	timebased
checkpointInterval	15
postingsDataStream	\${was.install.root}\${file.separator}ter
wsbatch.count	5



Job Classes

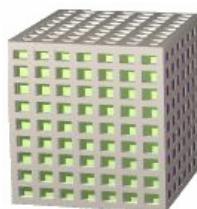
- Administrative control over resource consumption
- Defined through Grid Scheduler configuration in WAS admin console
- Named policies that control
 - ▶ maximum execution time
 - ▶ maximum number of concurrent jobs per endpoint (logical batch container)
 - ▶ job log retention (age, space)
 - ▶ execution record retention (age, number)



WebSphere Extended Deployment - Object Grid -

What is ObjectGrid?

A flexible framework for realizing high performance, scalable and data-intensive applications



It can be used as a very powerful cache that scales from simple in-process topologies to powerful distributed topologies.

It can be used as a form of in memory database to manage application state (and it scales to 1000's of servers).
Sometimes referred to as Distributed Application State Management.



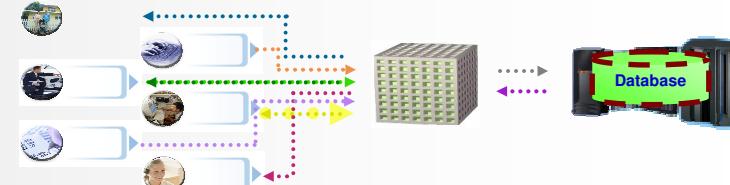
It can be used as a platform for building powerful Data Grid applications.

So what can you do with ObjectGrid?

Five Exemplar Usage Scenarios

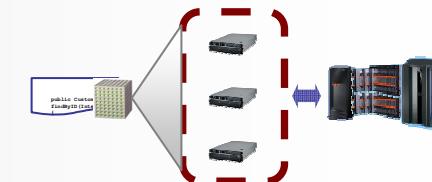
Database Cache

Reduce database contention



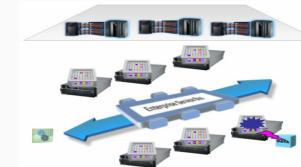
Client / Server

Clients cache a subset of data



Peer-to-Peer / Shared

Integration with SOA



Real Time Data / Event Mining

Continuous queries to support real-time decision making



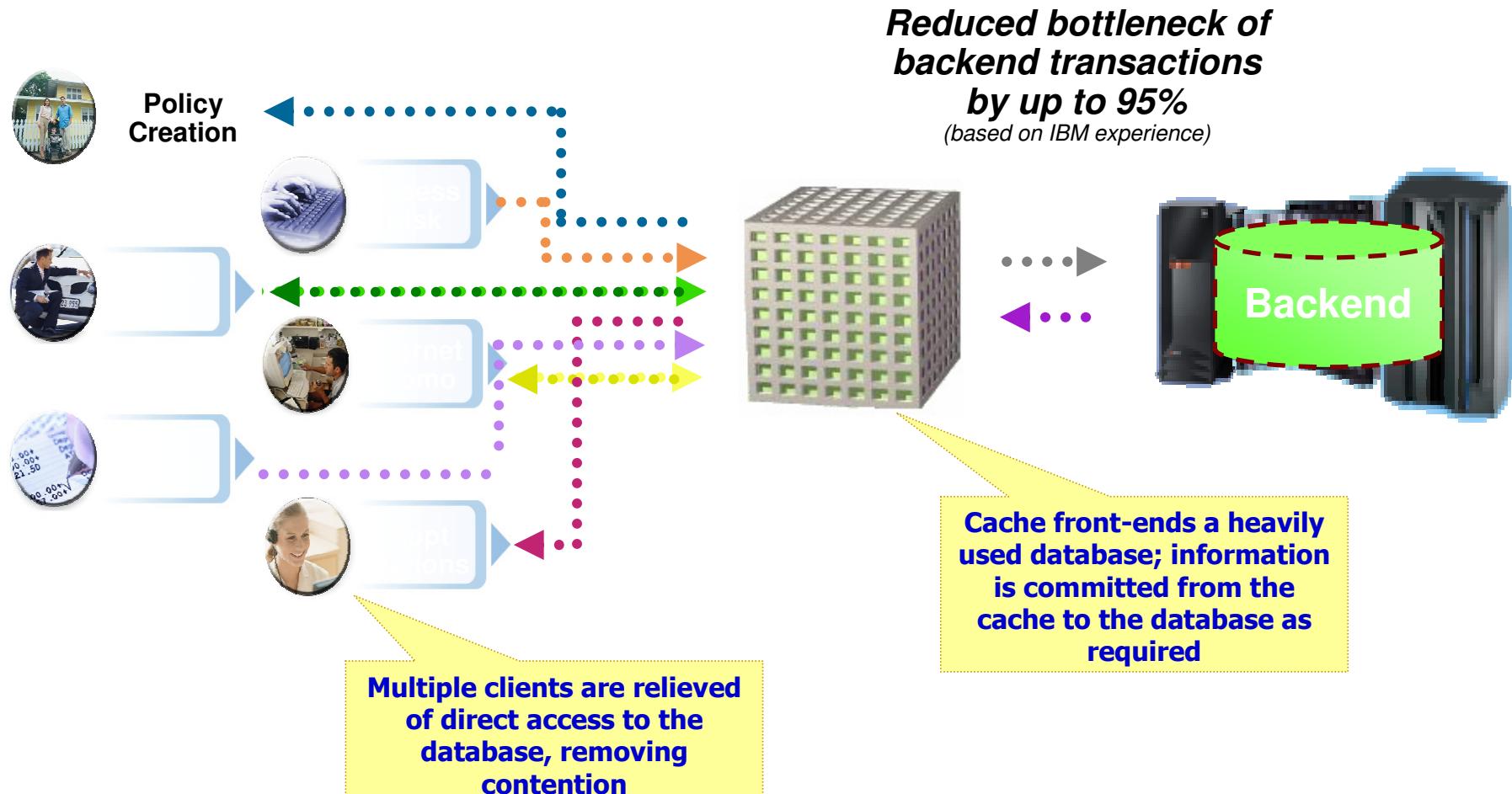
Ultra-scale Data Grid

Run code everywhere and aggregate the results



ObjectGrid: Backend Cache

For 3-tier architectures, significantly speedup backend access...



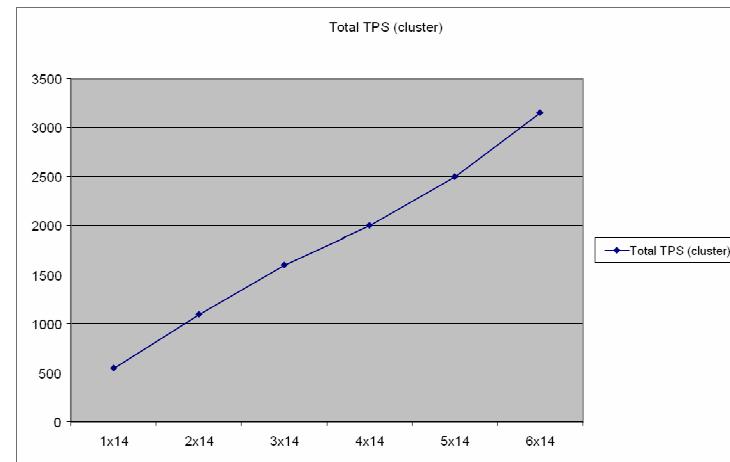
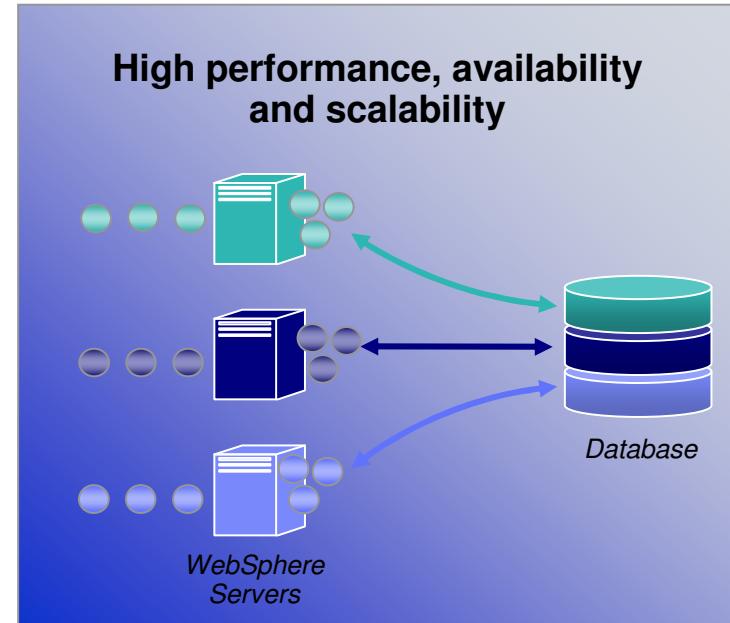
ObjectGrid

- What is it?
 - ▶ A powerful cache that scales from simple in-process to 1000 server farms
 - ▶ A form of in memory database to manage application state
 - ▶ A platform for building Data Grid applications
- Constructed from a portable, pluggable core with extensions
- Transactional
- High Performance Distribution features, including
 - ▶ Replication, Write-through support, Partitioning, Clustering, Client-Server access model
- Advanced Features
 - ▶ Hierarchical keyword-based invalidation
 - ▶ Dynamic indexing support
 - ▶ Spring Integration
 - ▶ Query: EJB-QL Style; Continuous and Temporal
- Runs in any J2SE environment (1.4.x) or higher
 - Such as Tomcat



Asymmetric Clustering: Partitioning Facility

- High Performance Computing Capabilities
 - ▶ Dynamic data partitioning and re-partitioning
 - ▶ High end caching
 - ▶ Workload management
 - ▶ Autonomic high availability management
 - ▶ Virtual Data Source Support
- Consider Partitioning for:
 - ▶ Write-intensive applications that traditionally do not scale well because of contention for data
 - ▶ Applications that must process high-speed message feeds or require other types of sequenced request processing
 - ▶ Applications with a need for singletons that must be made highly available
- Results
 - ▶ Consistently low response times
 - ▶ 99.999% availability (Class 5)
 - ▶ Linear scalability on commodity hardware



Questions

