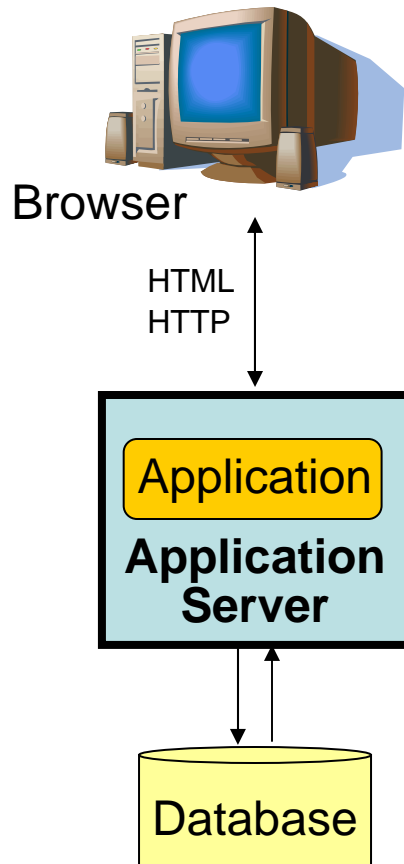


WebSphere's Advanced Technology Improves Business

Achieve Elastic Application Scalability

Why Was The Web Application Server Invented?

To support applications that can be accessed across the Internet with a browser



The application server is the infrastructure to run your application

Over Time, Application Servers Have Added Capabilities

- Clusters of servers to provide greater efficiency and availability
- Transaction integrity to protect stored data
- Advanced security features
- Batch processing facility for Java
- Native platform features to optimize standard Java functions

An Explosion Of Mobile, Social And Cloud Applications Is Driving **New** Demands On Middleware

- Customers now expect high reliability and responsiveness for Web applications
- Businesses need to cope with transaction-intensive applications requiring access to large amounts of data
- Developers need a server that is lightweight, quick, and easy to configure

WebSphere Meets The New Demands

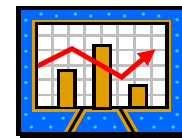
- WebSphere Application Server Network Deployment (ND) adds **intelligent management** for elastic scalability and reliability
- WebSphere Extreme Scale provides an **elastic grid** to further improve scalability for data-intensive applications
- WebSphere Liberty Profile includes a **lightweight development server** with fast startup time and easy configuration

... and these new capabilities come with exceptional price performance

Intelligent Management Capabilities Provide Elastic Scalability And Greater Availability

PERFORMANCE MANAGEMENT

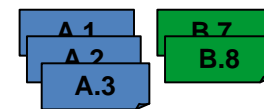
- ▶ Maintain response time objectives despite heavy demand or server failure
- ▶ Guarantee response time to selected classes of users or applications
- ▶ Elastic scalability adds resources as demand increases, reduces as demand subsides



*Self-Optimizing
Self-balancing*

APPLICATION EDITION MANAGEMENT

- ▶ Ensure uninterrupted service via rolling application updates



Self-Managing

HEALTH MANAGEMENT

- ▶ Create configurable “sense/respond” health conditions and actions that respond to problems before users experience an outage

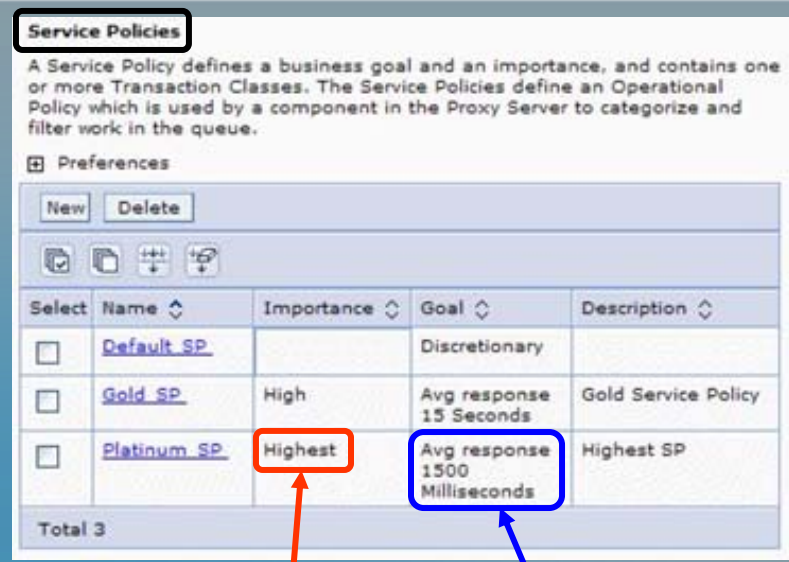


*Self-Protecting
Self-Healing*

Performance Management Uses Service Policies To Ensure Reliable Service

- A **Service Policy** specifies the **response time goal** for one application service and its **importance** relative to the others
- The **Dynamic Cluster** capability **allocates processor capacity** to workloads to implement and enforce service policies
- You can maintain response time objectives despite constantly **varying workload demands**
- This scheme can also ensure **differentiated response time objectives** for various classes of customers and applications
- The On Demand Router **sequences incoming requests** based on response time objectives to classify, prioritize, and intelligently route requests, ensuring that service policy goals are met

Service Policy Definition



The screenshot shows a web interface for defining service policies. At the top, there is a title 'Service Policies' and a descriptive paragraph. Below that is a 'Preferences' section with 'New' and 'Delete' buttons. The main part of the interface is a table with columns for 'Select', 'Name', 'Importance', 'Goal', and 'Description'. There are three rows of policies: 'Default_SP', 'Gold_SP', and 'Platinum_SP'. The 'Platinum_SP' row has 'Highest' in the 'Importance' column and 'Avg response 1500 Milliseconds' in the 'Goal' column. A red box highlights the 'Highest' text, and a blue box highlights the 'Avg response 1500 Milliseconds' text. Arrows point from these boxes to labels 'Relative importance' and 'Response time goal' respectively. At the bottom of the table, it says 'Total 3'.

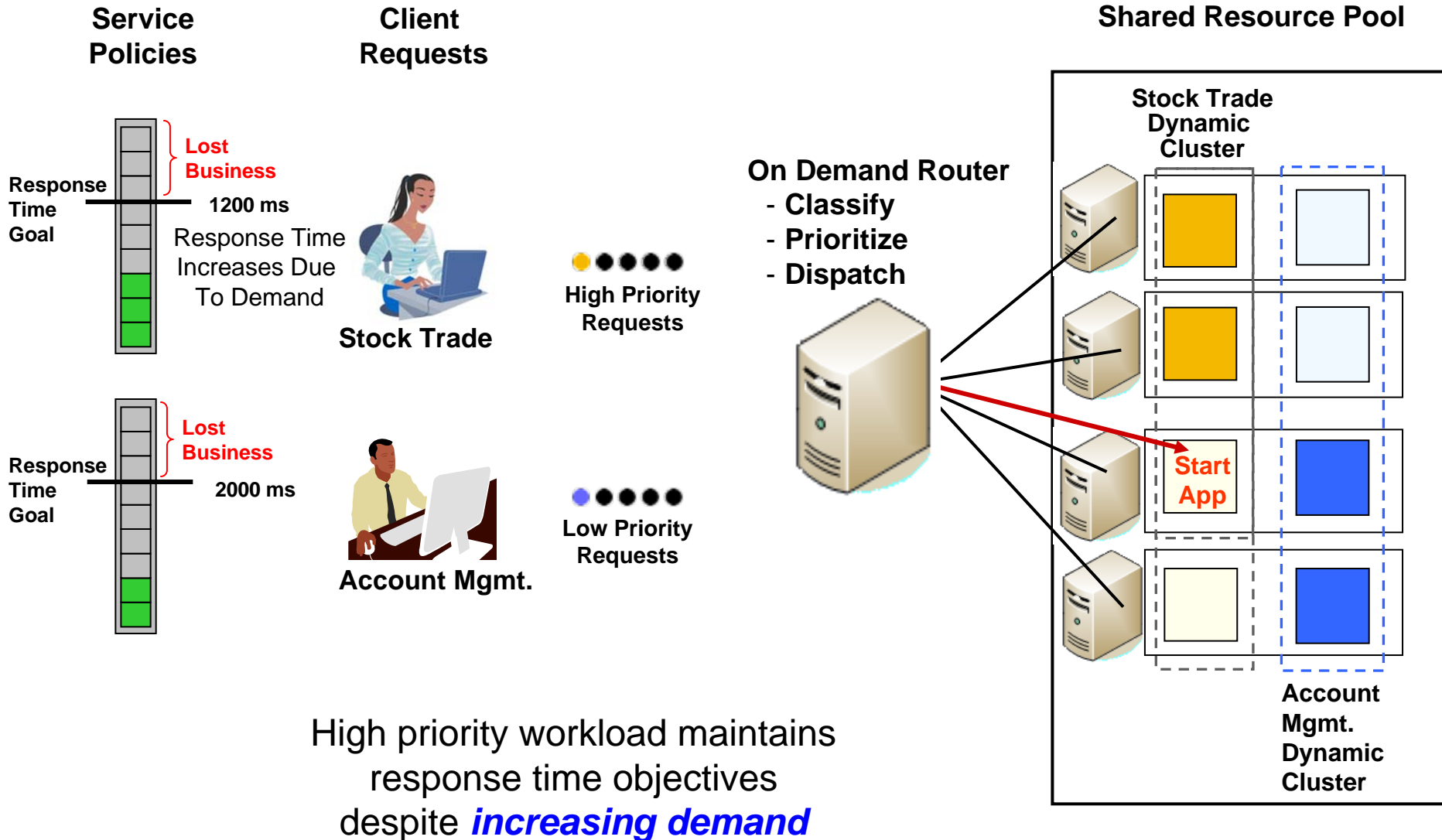
Select	Name	Importance	Goal	Description
<input type="checkbox"/>	Default_SP		Discretionary	
<input type="checkbox"/>	Gold_SP	High	Avg response 15 Seconds	Gold Service Policy
<input type="checkbox"/>	Platinum_SP	Highest	Avg response 1500 Milliseconds	Highest SP

Total 3

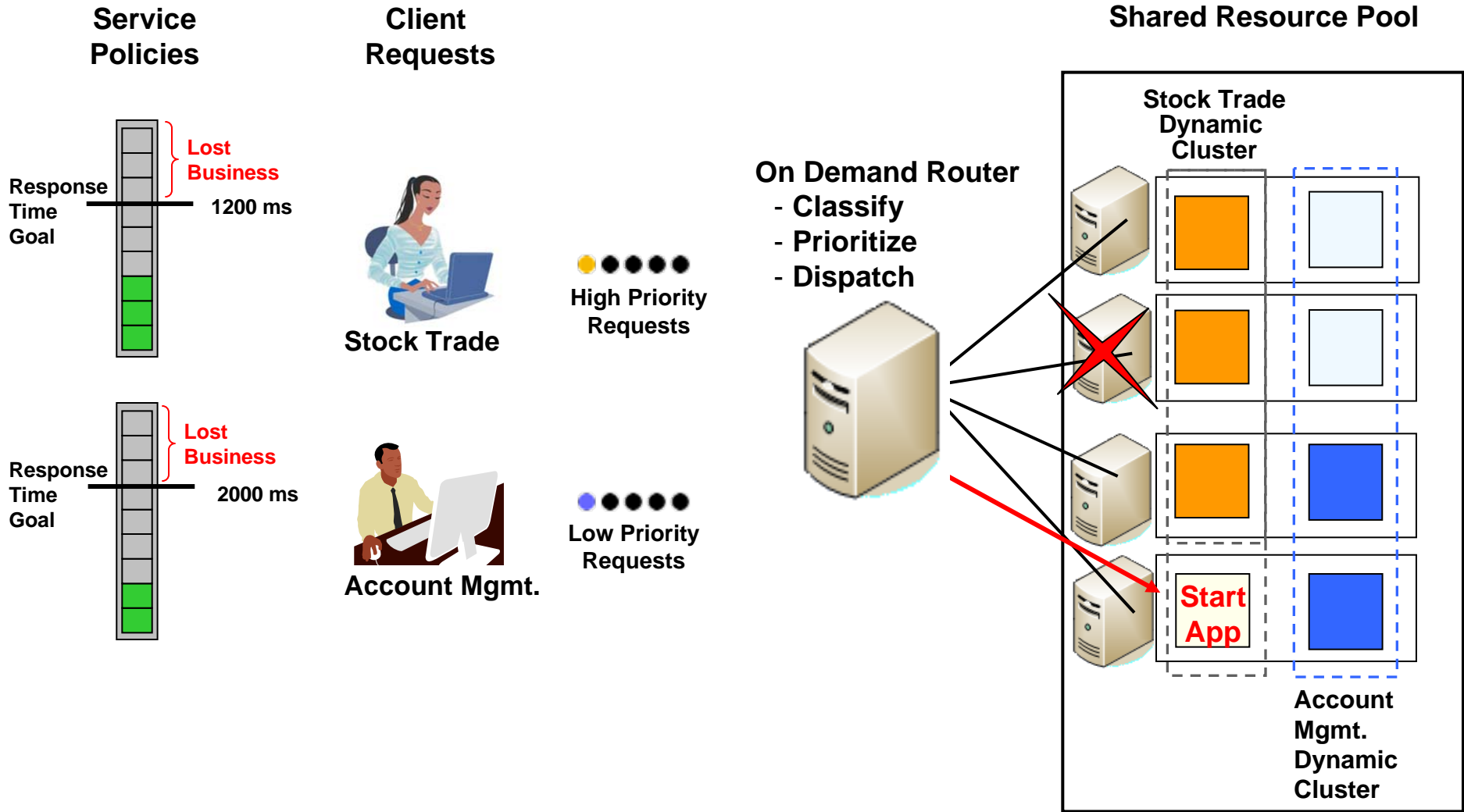
Relative
importance

Response
time goal

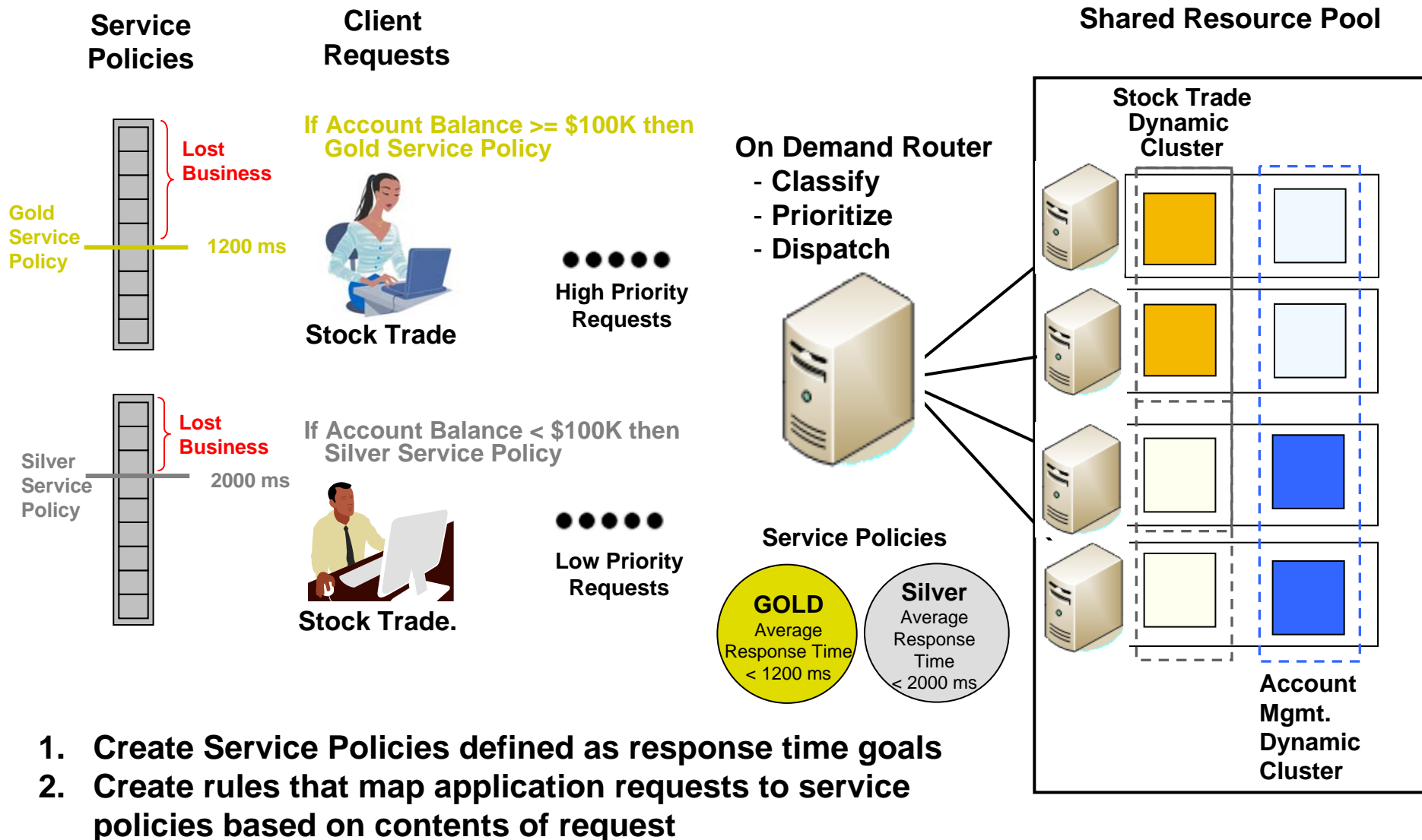
WebSphere ND Manages Response Time Objectives Defined In Service Policies



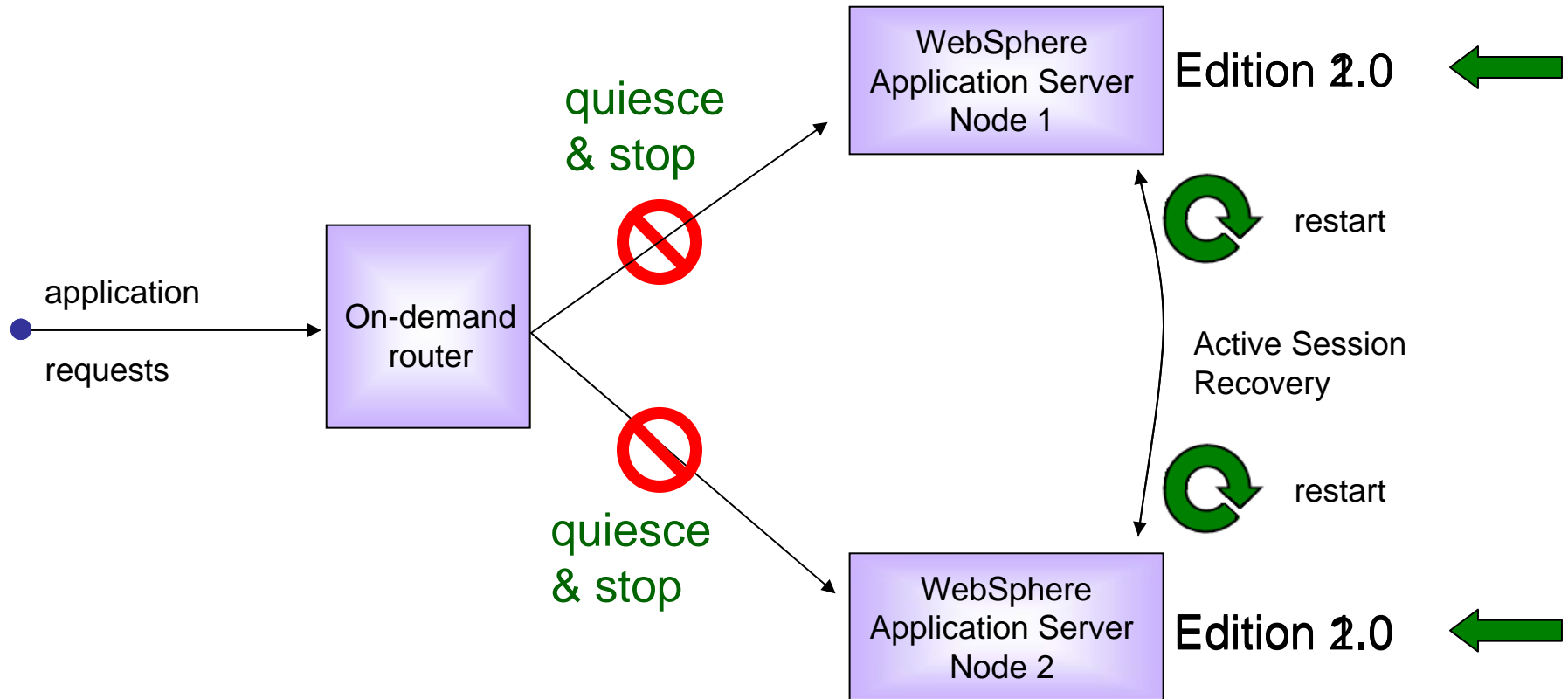
WebSphere ND Maintains Response Time Objectives Despite Server Failure!



Guarantee Faster Response Times For Selected Customers And Mission-Critical Applications

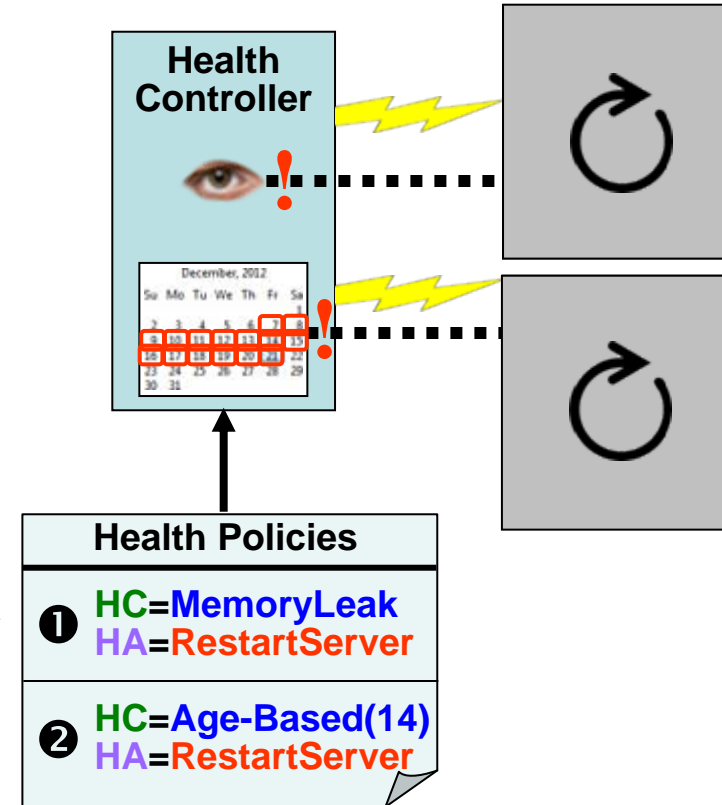


Edition Management: Deploy A New Application Version With No Interruption To Customers



Improve Application Reliability With Health Management

- WebSphere ND v8.5 includes a policy-driven **Health Controller** to continuously monitor applications and take corrective measures
- **Health Policies** specify **Health Conditions (HC)**, and **Health Actions (HA)** to be taken when these conditions are detected
- In this example:
 - Policy 1 Restart Server** when a **Memory Leak** causes the heap to exceed a threshold
 - Policy 2 Restart Server** when it has reached a configured **Age** (e.g. 14 days)
- WS ND 8.5 can provide **35 flexible Health Management policies.**



WebSphere Meets The New Demands

- WebSphere Application Server ND adds **intelligent management** for elastic scalability and reliability
- WebSphere Extreme Scale provides an **elastic grid** to further improve scalability for data-intensive applications
- WebSphere Liberty Profile includes a **lightweight development server** with fast startup time and easy configuration

... and these new capabilities come with exceptional price performance

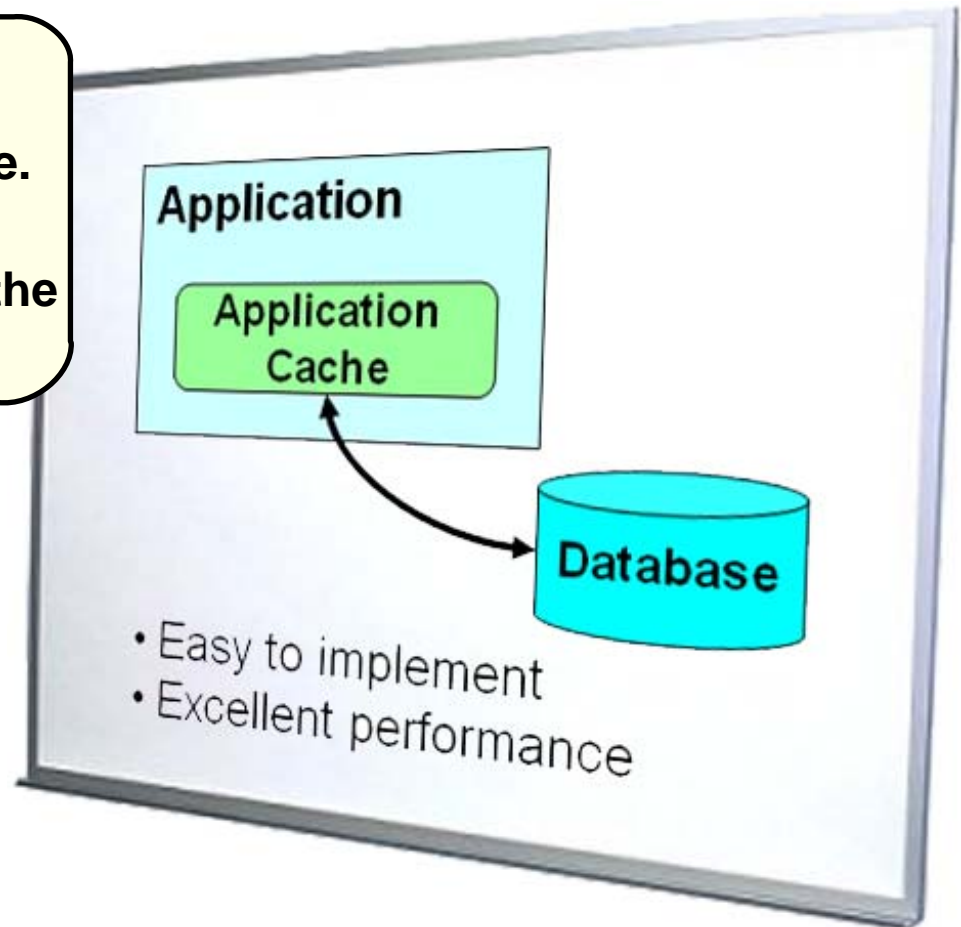
Some Performance Problems Require A Different Solution

Our customer account app runs too slowly. Too many apps are using the database.

An in-memory cache could greatly reduce requests to the database.



Developer



Cons on Local Cache:

- × Difficult for multiple applications to share the cache
- × Limited to one location
- × Cache size is limited to the amount of memory available on the computer
- × Cache reliability depends on application and system reliability

It's a great idea, but we have the same problem with a LOT of applications, and a local cache is a limited solution.

We need an enterprise-wide caching solution!



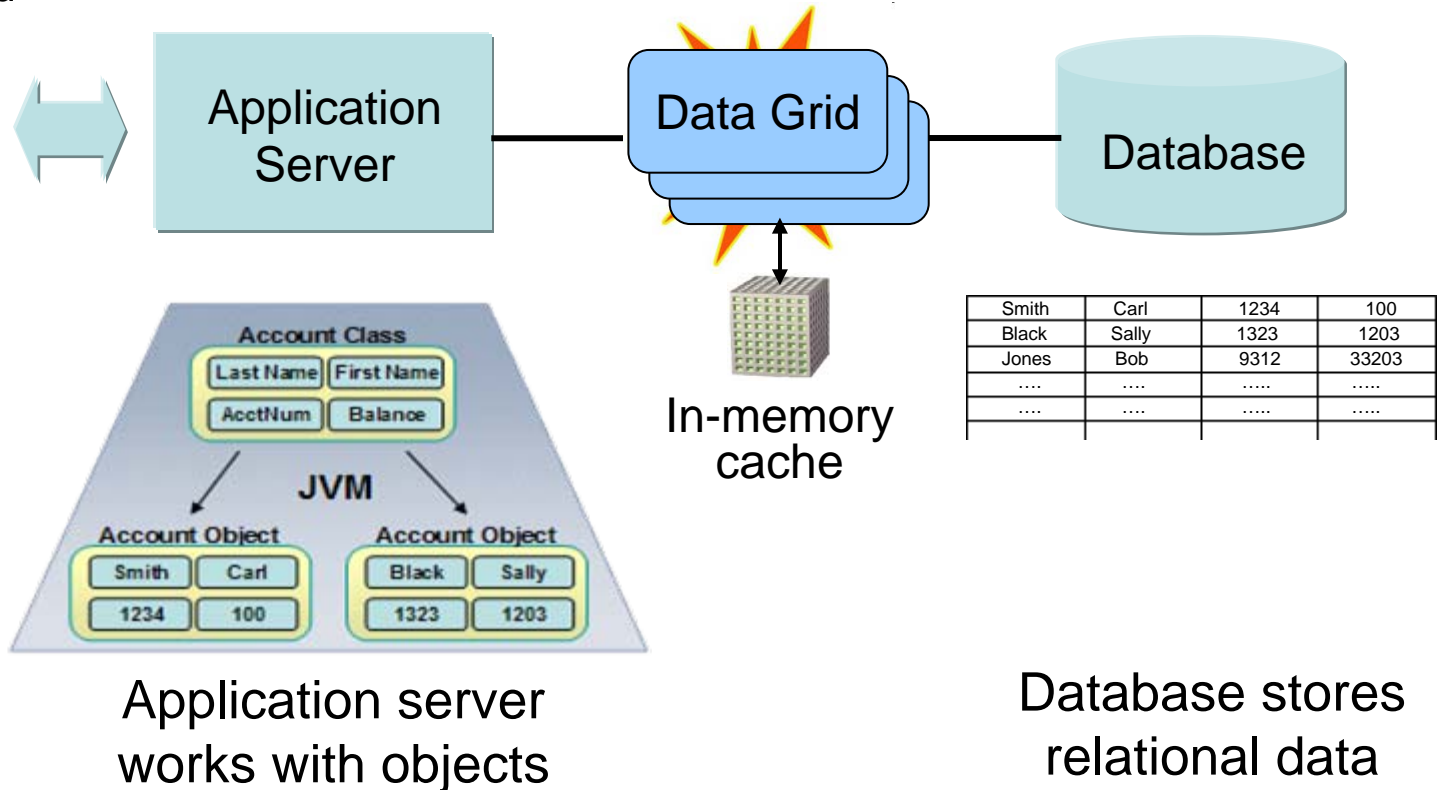
IT Architect

Extreme Demand Workloads Illustrate Best Use Case For Data Grids

Extreme Demand



Response time delays

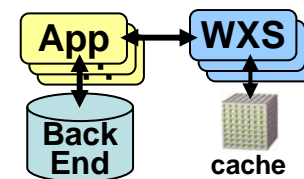


Extreme performance with transactional in-memory cache (data grid) to store data in Java object form

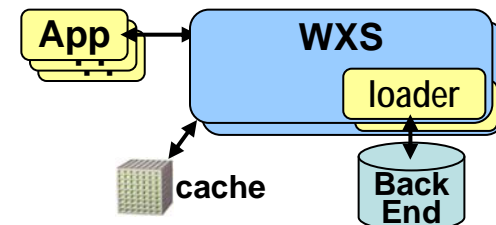
Four Use Patterns For Elastic Data Grid

Demonstrate A Variety Of Valuable Applications

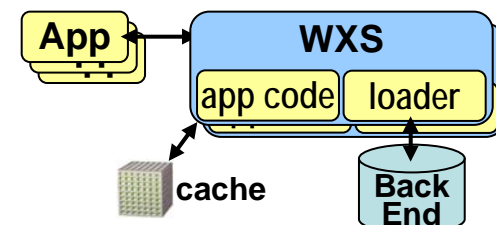
1. Side Cache pattern eliminates redundant requests for the same data
 - ▶ After reading data from the back end, keep a copy in the grid (a “side cache”) to satisfy future reads
 - ▶ Application manages updates to data source and WXS cache



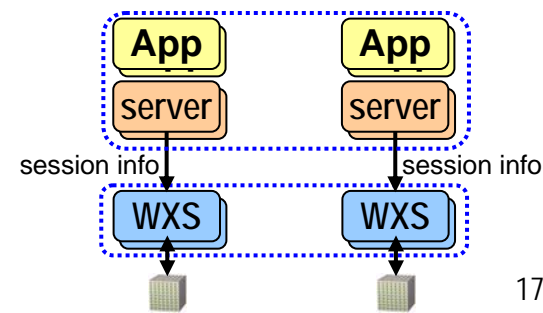
2. Inline cache pattern lets WXS manage the cache contents and back end updates
 - ▶ Similar to Side Cache pattern, except WXS manages cache and updates to back-end automatically



3. Extreme Transaction Processing (XTP) pattern moves application code onto data grid to minimize latency
 - ▶ Allows processing a data set in parallel, across servers, and combining the results
 - ▶ Data grid becomes System of Record with write-behind store



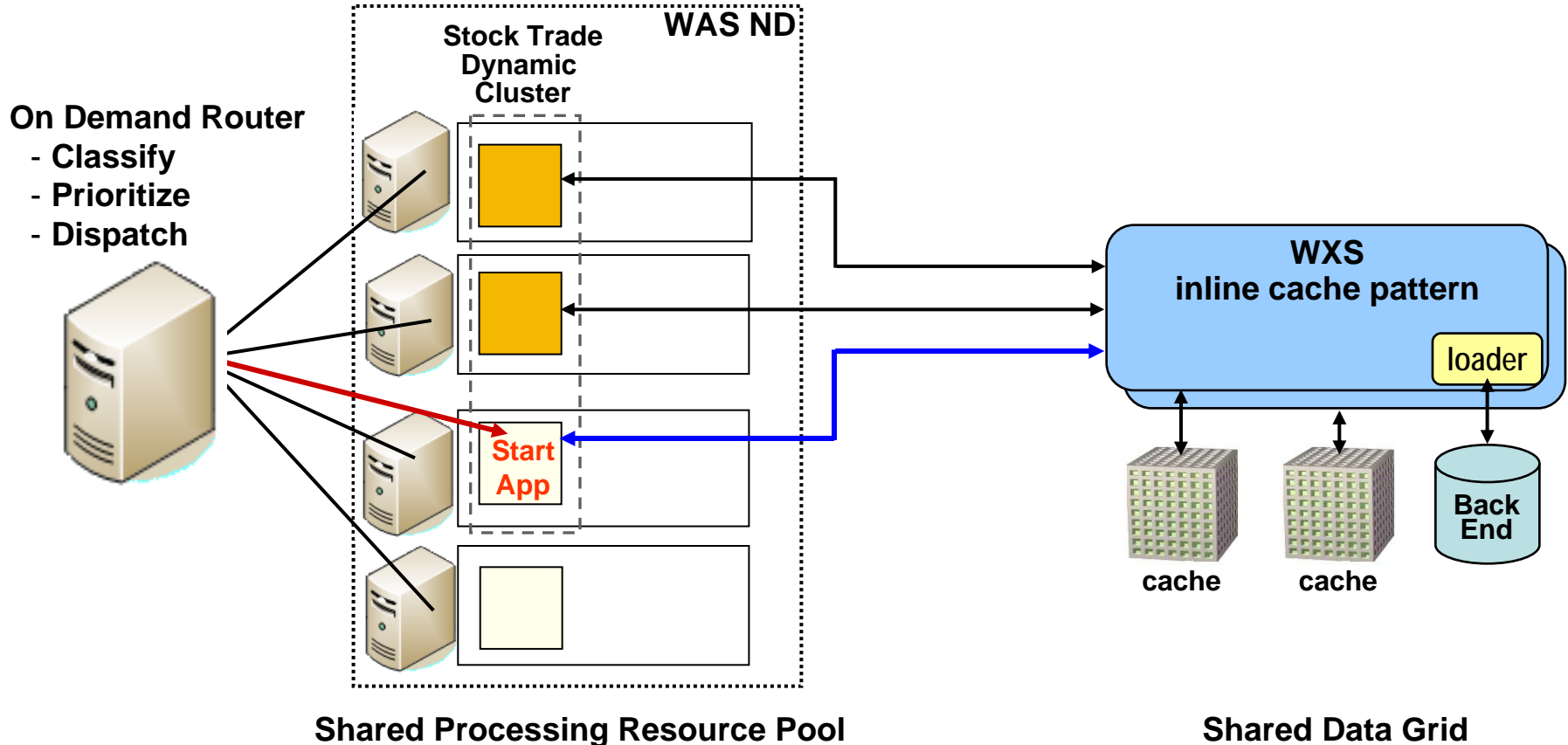
4. HTTP Session Store pattern - store HTTP session info in a data grid across user requests
 - ▶ Allows users to remain logged on with state saved (e.g. shopping cart)



WebSphere Extreme Scale Creates Synergy With WebSphere ND

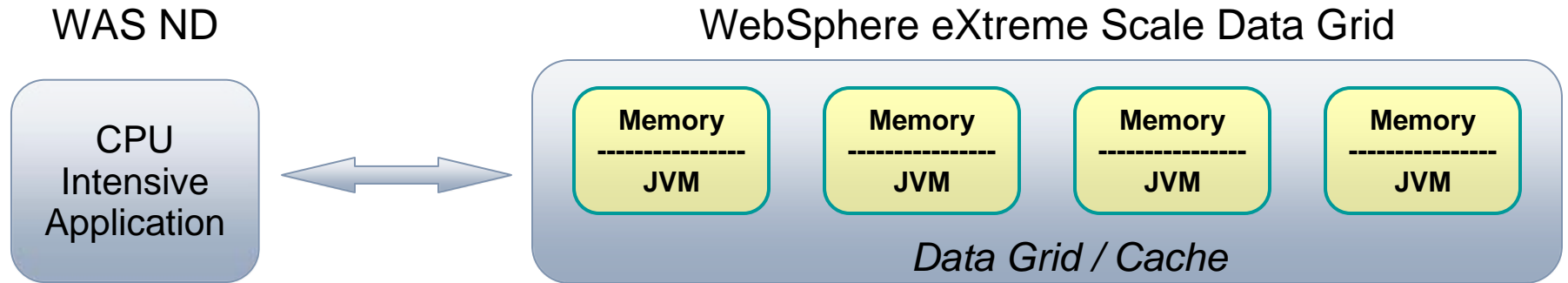
WebSphere ND 8.5: Elastic Processing Capacity

WebSphere Extreme Scale: Elastic Data Grid



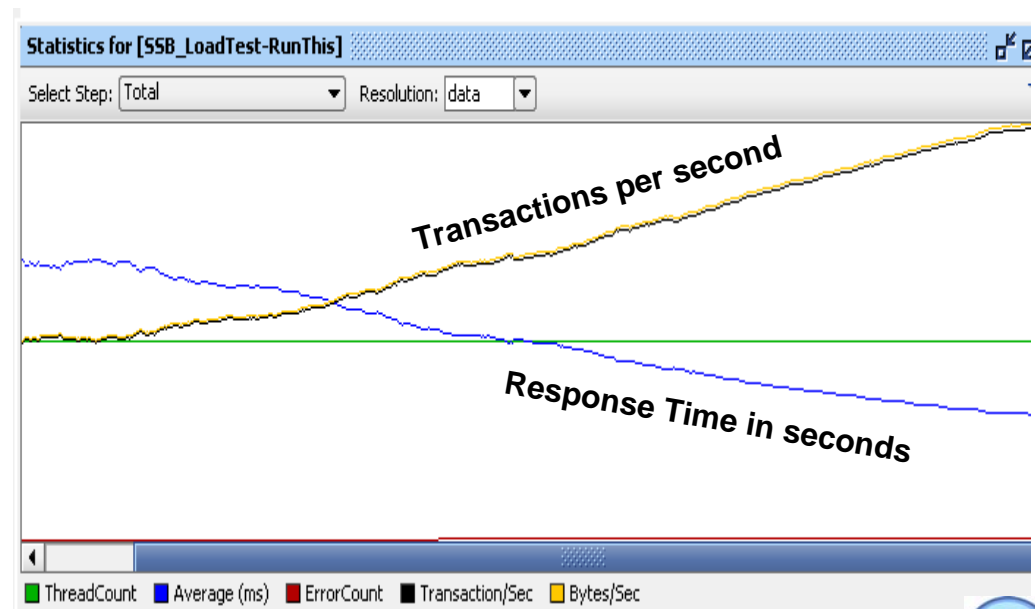
When a new application instance is started on WAS ND, it can immediately see and use the data cached in the shared data grid

Demo: WebSphere eXtreme Scale Caching Can Increase Application Performance



Demo Condition:

- CPU-intensive workload
- Response times reduce and transactions per second increase



WebSphere Meets The New Demands

- WebSphere Application Server ND adds **intelligent management** for elastic scalability and reliability
- WebSphere Extreme Scale provides an **elastic grid** to further improve scalability for data-intensive applications
- WebSphere Liberty Profile includes a **lightweight development server** with fast startup time and easy configuration

... and these new capabilities come with exceptional price performance

We really need autonomic and other WebSphere features for production, but our developers need something fast and easy!



Development Manager

WebSphere Liberty Profile has a small footprint, is easy to configure... and it's REALLY FAST!



IBM

Liberty Profile Has A Highly Composable Runtime With Just The Features You Need

WAS Extensions

AsyncBeans
wxs-session-1.0
wxs-1.0
...and more

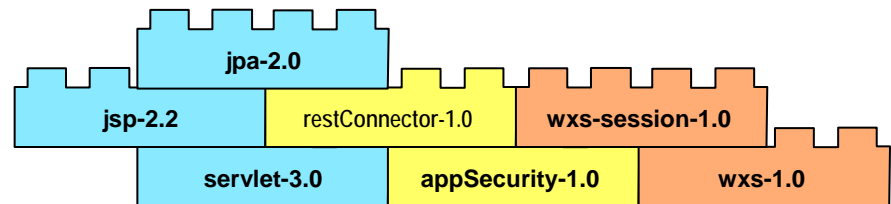
JEE

JPA 2.0
JSP 2.2
Servlet 3.0
...and more

Runtime services

restConnector-1.0
appSecurity-1.0
...and more

	WAS 8.5 ND	Liberty
App Server Download size	2048 MB ^{2,3}	117 MB ³
App Server Install size	1439 MB ^{2,3}	179 MB ³
Startup Time ¹	31 seconds	2 seconds
Physical memory (RAM) size ¹	282 MB	87 MB



Traditional Full WAS Profile:
Everything we can give you
is there all the time

WAS v8.5 Liberty Profile:
Bring in only the subset
your application requires

1. Stock trading application with off-platform database
2. Does not include WebSphere installer
3. Includes JVM

Demo: Compare Server And Application Installation

IBM and the competitor both claim fast and easy installation for their developer environments – just “unzip and go!”

In this demonstration, we will see what “unzip and go” REALLY means:

- ▶ First, for the IBM WebSphere Liberty Profile.
- ▶ Next, for the Competitor’s Zip Distribution.



WebSphere Application Server v8.5 Includes Liberty Profile

What Liberty Means For Developers

Incredibly fast start time

Small download size

Lightweight runtime with small memory footprint

Same reliable containers as WAS for same programming models

Popular Development Platforms:
Linux, Windows, OS X

Simplified server configuration: one XML file, or several, to make sharing and reuse easy

What Liberty Means For Production

Simple Configuration Model:
deploy using XML configuration from development environment

Simple Deployment Model: zip the server+application+configuration, then unzip to install and deploy

Adding features or updating the configuration does not require restarting the server

Elastic Caching is supported with WebSphere Extreme Scale

Liberty is shipping in IBM Products, included with Worklight, embedded in CICS Transaction Server 5.1, and more

Find WebSphere Liberty Profile Here

IBM Software Products:

- WebSphere Application Server ND 8.5 (all platforms)
- WebSphere Application Server 8.5 (all platforms)
- WebSphere Application Server for z/OS 8.5
- IBM Worklight v5.0 (included)
- CICS Transaction Server v5.1 (embedded)

IBM PureApplication System

Free Downloads from <http://wasdev.net>:

- WebSphere Application Server Liberty Profile
- WebSphere Application Server Developer Tools for Eclipse



WebSphere Meets The New Demands

- WebSphere Application Server ND adds **intelligent management** for elastic scalability and reliability
- WebSphere Extreme Scale provides an **elastic grid** to further improve scalability for data-intensive applications
- WebSphere Liberty Profile includes a **lightweight development server** with fast startup time and easy configuration

... and these new capabilities come with exceptional price performance

Summary

- WebSphere Application Server Network Deployment 8.5 has **best scalability** and is **more resilient** with **Intelligent Management**
- WebSphere Extreme Scale **increases scaling** and **supercharges performance** for transaction-intensive applications requiring access to large amounts of data
- WebSphere 8.5 includes Liberty Profile – a new **fast, lightweight development and production** environment

